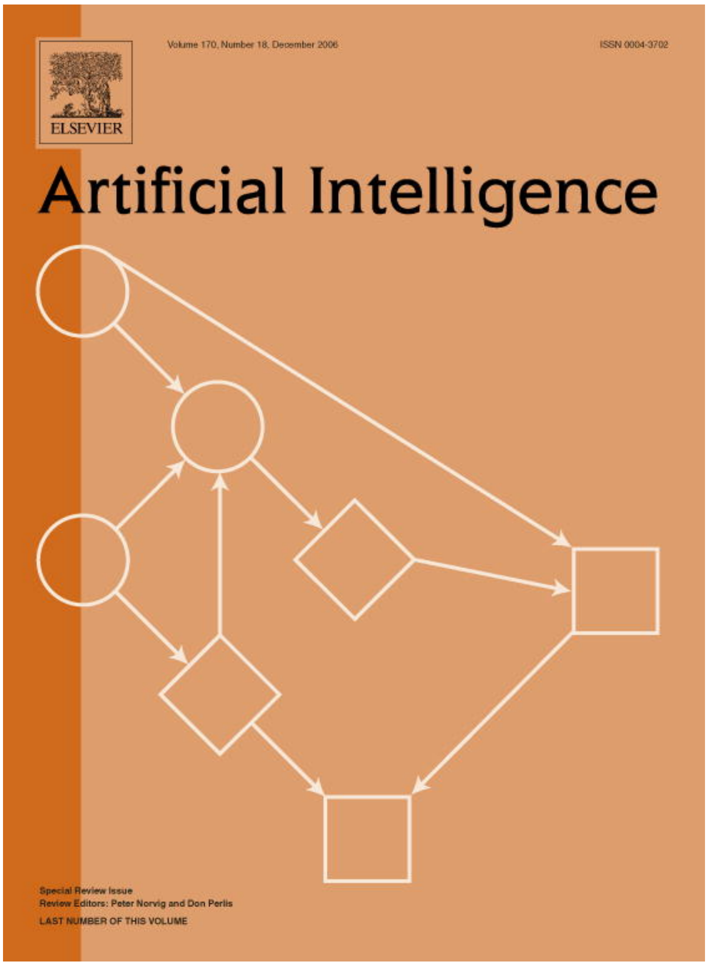
Volume 170, Number 18, December 2006
ISSN 0004-3702
ELSEVIER
Artificial Intelligence
Special Review Issue
Review Editors: Peter Norvig and Don Perlis
LAST NUMBER OF THIS VOLUME

Available online at www.sciencedirect.com
ScienceDirect
ELSEVIER
Artificial Intelligence 170 (2006) 1239–1250
Artificial Intelligence
www.elsevier.com/locate/artint

Book reviews

Ronald J. Brachman, Hector J. Levesque, Knowledge Representation and Reasoning, Morgan Kaufmann, ISBN 1558609326, 2004, 381 pages.
Raymond Reiter, Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems, MIT Press, ISBN 0262182181, 2001, 448 pages.
Erik T. Mueller, Commonsense Reasoning, Morgan Kaufmann, ISBN 0123693888, 2006, 432 pages.
Chitta Baral, Knowledge Representation, Reasoning and Declarative Problem Solving, Cambridge University Press, ISBN 0521818028, 2003, 544 pages.

Knowledge representation and commonsense reasoning: Reviews of four books

Leora Morgenstern

IBM T.J. Watson Research Center, Hawthorne, NY 10532, USA

1. Characterizing knowledge representation and commonsense reasoning

Knowledge representation and the automation of commonsense reasoning have been bound together since the publication of John McCarthy's "Programs With Common Sense" [15]. McCarthy argued that common sense in a computer could best be achieved using a declarative representation of commonsense knowledge and some means for deducing the consequences of that knowledge. That is, in order to get a computer to behave in a way that exhibits common sense, one must first provide it with a fair amount of basic facts about the world. Moreover, these facts should be represented in a declarative language like first-order logic.

This description of the formal commonsense reasoning (CSR) enterprise seems quite similar to our understanding of the knowledge representation (KR) enterprise: the development of formal languages and structures for representing knowledge, along with methods for reasoning with that knowledge. But the two fields, though closely related, are not identical. Rather, they are connected through their shared communities, as well as through a symbiosis of problems and results; and they are distinguished by their different focuses of interest.

Shared communities: There is a great deal of overlap between KR and CSR. Someone working on theories of causation, for example, or on the interaction between knowledge and ability, can just as well say that he is doing KR as he can say that he is doing CSR. As a result, there is a large intersection between the KR and CSR communities.

Partial symbiosis: CSR and KR both feed into one another. CSR can be seen as a source of—and even a breeding ground for—research problems for the KR community. The development of theories of action, causation, planning, knowledge, and default reasoning began as attempts to formalize commonsense reasoning problems before they became subareas of KR.

At the same time, the KR community has other sources for its research problems, most notably the study of knowledge bases and the development of expert systems. The questions of how one can represent and store large

E-mail addresses: leora@steam.stanford.edu, leora@us.ibm.com (L. Morgenstern).

doi:10.1016/j.artint.2006.10.012

amounts of information in a knowledge base system, and query the knowledge bases about simple consequences of the stored information, have led to work on frame-based systems, semantic networks, description logics, and various proof methods; the question of how to develop expert systems has led to the development of rule-based systems and accompanying proof strategies. These research areas are generally not considered to be CSR.

Note that in the same way that CSR feeds into KR, KR feeds into CSR. The CSR community uses the results that have been developed in the KR community, such as, for example, methods for translating from one temporal language into another.

Different focuses: Nevertheless, the KR and CSR communities have different focuses of interest. CSR tends to be more concerned with the details of axiomatizing a particular domain, while KR tends to be more concerned with meta-domain research, including results demonstrating equivalence between various formalisms and complexity issues. This is not to say that those in the CSR community are not interested in meta-domain research. Rather, it is the case that if one is doing serious object-level formalization, one is almost certain to be a member of the CSR community and not only a member of the KR community.

There are several commonsense domains that are of particular interest to the CSR and the KR communities because they are so central to any commonsense problems that one can think of. These are theories of time, action, and causation, and theories of knowledge or belief. It is hard to find any commonsense reasoning problem that does not somehow refer to events that happen in the world, and how things change over time; thus, there is a need for theories of action and causation. Likewise, commonsense problems that involve sensing or communication are ubiquitous; these nearly always involve reasoning about an agent's beliefs or knowledge. Indeed, much of the work in the CSR community—and, as a result, in the KR community—has involved research into these domains.

Let us call a commonsense domain that is present in a large segment of commonsense reasoning problems a *basic commonsense domain* or *basic domain*. Besides action and belief, discussed above, I would argue that basic domains include spatial reasoning, physical reasoning, reasoning about multiple agents, reasoning about social interaction, and reasoning about emotions. The amount of existing research varies widely among these different basic domains. Relative to the amount of research on action, belief, and spatial reasoning [3], and sometimes (say, for reasoning about emotions) even in absolute terms, little research has been done in the CSR community on most of these basic domains. This is one of the reasons that it has been difficult to solve the challenge problems on the commonsense problem page.[1]

Consider, for example, one of the problems on that web site, the Surprise Birthday Present Problem [6]. The problem concerns two siblings who wish to surprise their sister with a joint present for her birthday, two weeks from now. They go into a closed room to decide on the present and plan how they will buy it. The problem is to reason that the plan will work under certain circumstances (they are not overheard by their sister, they decide on a present, they have enough money to buy it, etc.) and will not work under other circumstances (their sister is in the closed room, they cannot decide on a present, or they wait until after their sister's birthday).

A variety of basic commonsense domain theories are needed to properly represent this problem: theories of space and physics to define closed and open rooms, and characterize when conversations can be overheard; theories of multi-agent planning, to reason about how agents come to joint decisions, and how they perform actions together or delegate actions to one another; theories of mind to characterize under what circumstances agents are surprised. A robust basic domain theory must be *elaboration tolerant* [16], able to handle the variants posed in the problem statement as well as similar ones. In addition, non-basic domain theories are needed—e.g., facts about birthdays and how one purchases things. Moreover, to properly formalize this problem, these different basic and non-basic domain theories must be integrated. That itself is a huge challenge, as Davis [5] has pointed out.

The CSR community cannot yet solve problems such as these (see [22] for an analysis of some of the difficulties involved in solving the Surprise Birthday Present Problem). The KR community cannot solve these problems either, but they are currently not even focussing on them; indeed, they are not yet on their radar.

[1] This web site contains a list of challenge problems for the formal commonsense reasoning community. It is currently located at http://www-formal.stanford.edu/leora/commonsense/ and scheduled to migrate soon to http://www.commonsensereasoning.org.

2. Teaching KR and CSR

Whatever the exact nature of the connections between these two communities, it is indisputable that the past several decades have seen a great deal of activity in both KR and CSR. Since the late 1980s and early 1990s, especially, there has been a growing emphasis on synthesizing work among different groups of researchers working on a particular topic—so that, for example, there are many results comparing the expressivity of various languages and theories of action [29], or comparing various description logics to one another [1]—and on developing standards and criteria by which to evaluate such research. (These are still being developed for that portion of CSR research which focusses on object-level formalization of less popular domains; it is especially difficult to develop evaluation criteria before there is a critical mass of work in a particular area.) The activity is evidenced by the growth and popularity of the KR biannual conferences, and all the offshoots of this enterprise, including the Description Logic workshop series, the semantic web conferences, and the biannual commonsense symposia.

Part of promulgating an area of research is, of course, teaching courses and seminars in the area. Despite the large amount of activity, there have been, until recently, few texts available for courses in KR and CSR. I taught several courses on KR and CSR from the late 1980s through the 1990s at several universities. I usually chose as the primary text Ernie Davis's excellent (and now out-of-print) *Representations of Commonsense Knowledge* [4], even in the late 1980s when it was only in draft form. (Disclaimer: Ernie Davis was my thesis advisor, and we have collaborated on a number of things over the years, so I cannot claim objectivity.) However, I always had to resort to supplementing with chapters of other books or journal papers: material on semantic networks for KR courses, additional material on nonmonotonic reasoning for CSR courses. This has been an issue that I have faced not only in my side life teaching courses at universities. As a research scientist at IBM T.J. Watson Research Center, I am often responsible for teaching KR—to customers, for example, who will have to maintain some reasoning tool, or to team members who need to be brought up to speed on basic concepts of knowledge representation and reasoning. It is especially important in these situations to have a comprehensive text that is easy for novices to use and understand.

In a joint paper with Rich Thomason [23] in 2000, we pointed to the dearth of textbooks in the area and the fact that it was impossible to get a single text that covered a KR syllabus. We proposed as a sample syllabus for a KR course—and correspondingly, a suggested outline for a textbook—the following:

1. a review and discussion of first-order logic
2. an analysis of why first-order logic is not sufficient
3. an example of at least one extension of first-order logic (modal logics, logics augmented by quotation, or a nonmonotonic logic)
4. semantic networks and their applications
5. inheritance with and without exceptions
6. temporal logic and planning
7. a discussion of how KR systems have been used in real-world applications
8. time permitting, more detailed discussion of modal logic; some study of domains of CSR such as physical or spatial reasoning; logic programming; specific nonmonotonic formalisms; or Bayesian networks.

Six years later, I would suggest only minor changes to this list: the incorporation of OWL [2] into the section on semantic networks; a section on answer-set programming [7]; and some material tying KR into the work on the semantic web. It would also be nice to have a discussion of large-scale practical KR systems, and in particular, how to acquire and reason with large amounts of knowledge.

What about a textbook for formal commonsense reasoning? An ideal text for CSR would have a good deal of overlap with a KR text, but would differ because the two fields, as I have argued above, have different concerns. A CSR text would be less concerned with reasoning tools, but considerably more focussed on specific domains. The requirements would include:

1. the basics of first-order logic
2. a significant discussion of the extensions of first-order logic that are commonly used in CSR, including modal logics, nonmonotonic logics, and probabilistic logics

3. a detailed discussion of the basic domain theory of action and causation, along with an extended discussion of planning
4. a detailed discussion of the basic domain theory of knowledge/belief
5. some discussion of other basic domain theories, such as
   - spatial reasoning
   - physical reasoning
   - multi-agent reasoning
   - social interaction
   - emotions
6. tools that can be used in CSR systems
7. some discussion of the efforts to build a large database for commonsense reasoning (such as CYC [11]), including the recent attempts to construct a shallow commonsense knowledge base using mass human input [32] or by the analysis of existing web pages [13,26]; and the connections between this work and formal commonsense reasoning.

A note about requirements 3, 4, and 5: A primary purpose of these sections would be to teach the reader *how to* do formal commonsense reasoning. These sections should follow certain principles of doing CSR: clearly stating the problem, choosing a portion of the problem to solve, trying to develop a model, constructing axioms that characterize the model, checking for elaboration tolerance, and so on.

This is important because formalizing deep commonsense is still not well understood. The reader needs to understand the difficulties and the struggle. It is certainly not enough to present axiomatizations. A CSR text needs to get the reader to understand why the enterprise is important, why it is challenging, and how to manage even when it seems too difficult.

Any good textbook, of course, needs to do more than cover a set of issues. It needs to be accessible to students, researchers, and other teachers; it should demonstrate a decent writing style—if only to make sure that it will not confuse readers, annoy them, or put them to sleep; and it will hopefully have something that makes the book unforgettable in some way, something that makes it flow and sing, inspires students and researchers, or covers the field so solidly that it can serve as a reference volume for years to come.

After the comparative drought of the 1980s and 1990s, we now have several books published during the last five years which might be considered by a KR or CSR teacher looking for a textbook. How do they measure up to our criteria?

3. The reviews

3.1. Ronald J. Brachman and Hector J. Levesque: Knowledge Representation and Reasoning, Morgan Kaufmann, 2004

This excellent text covers virtually every topic in knowledge representation that Rich Thomason and I argued for in [23], plus a good deal more. This is no coincidence. Brachman and Levesque, by their own account, spent ten years planning the book, and almost another ten years writing it; virtually everyone in the KR community knew that they were writing it; and large parts of the structure of the book had no doubt entered our collective consciousness. [2]

It will come as no surprise to anyone who has read papers by Brachman and Levesque, or who has heard either speak, that the writing style is very, very good: clear, concise, and to the point. A text is often especially good when its authors have originated the material, and that is the case for much of this work. For example, Brachman and Levesque did large parts of the seminal work on frame-based languages and description logics, as well as on the trade-offs between expressiveness and tractability; Levesque has pioneered the use of the situation calculus for practical planning problems.

[2] The question is why Rich Thomason and I did not mention, in our paper, that Brachman and Levesque's book would be coming out shortly. I think that we were at the time like true but weary believers: while we had no doubt that the messiah would eventually come, we had given up predicting the date of his arrival.

This book is written from a decidedly pragmatic point of view. It is geared to the typical student in an introductory KR course—the student who is not planning an academic career, but who may very well wind up in a job requiring knowledge of practical KR techniques. The lion's share of this book focusses on such practical techniques—proof methods, procedural control of reasoning, production rules, frame-based systems, description logics, inheritance, Bayesian reasoning. Nevertheless, there is a reasonable amount of material on more theoretical areas of KR such as default reasoning. The chapter on this area includes lucid, though brief, discussions of circumscription, Reiter's default logic, and autoepistemic logic. Likewise, the chapters on temporal reasoning and planning set down the motivation for having specialized action languages, the major problems in reasoning about actions (such as the frame problem), along with solutions to these problems, and the basic paradigms of planning, with exceptional clarity in remarkably little space. These chapters suffice for a fine introduction to these areas, though there will certainly be teachers who will want a more comprehensive treatment for their KR courses.

On the whole, the decision to balance the book toward the introductory reader whose primary interest is in learning techniques that he can apply to practical problems in the real world is a good one, for several reasons. First, there are many more students who are taking KR to satisfy a university requirement, or because they want to learn some basic techniques, than there are students who are planning to embark on hard-core KR research. Second, the more we teach basic, practical KR techniques to as many students as possible, the greater the chance that these techniques will actually be used in the real world; that when these students get jobs, they will be able to help design intelligent databases; that they will understand how to write rules in a production system; that they will adhere to proper principles of knowledge engineering. Third, such students need a KR textbook much more than students who are starting their academic careers. The academically-oriented student will have no problem consulting the instructor's suggestions for further readings on circumscription or epistemic logic; he may find it less convenient, but he will not lose out.

Given its practical bent, however, the book might have been even stronger if two modifications had been made.

First, a chapter on logic programming, with an emphasis on answer-set programming, would have been a useful addition to this text. Answer-set programming has become increasingly popular in recent years, with tools being developed in many universities around the world, and the potential for use in many practical applications. (Chitta Baral's book on answer-set programming, reviewed below, provides an excellent account, but there is still a need for a chapter-length introduction to some of the major concepts for all those readers who are not willing to plow through an entire book on the subject.)

Second, the introductory chapter on first-order logic could be clearer, longer, and more accessible. The chapter is lucid and comprehensive, but it is more of a review chapter than a chapter meant to teach logic to the uninitiated. It begins with formal definitions of syntax, goes on to a discussion of semantics, interpretation, and denotation, and continues with an abstract discussion of logical consequence. There are comparatively few examples, and relatively little discussion of the examples. (There are plenty of examples in the next chapter, Expressing Knowledge, which sets down the principles of knowledge engineering and shows how one might start writing down facts about the world, and in subsequent chapters, as well.)

My concern is that this chapter will be difficult for many students who have not already had a logic course. Now, one could argue that such students should not be in a KR course, and that, indeed, a course in first-order logic should be a prerequisite to taking knowledge representation. My experience, however—and that of other instructors with whom I have spoken—is that there are always a few (and sometimes many) students who prove to have only the sketchiest memories of anything outside of Venn diagrams and truth tables. My approach to this situation has been to give students photocopied chapters of accessible logic texts, such as Benson Mates's *Elementary Logic* [12] and to schedule, during the first few weeks of class, crash courses in first-order logic, with a heavy emphasis on translating from English to logic, and on proof techniques. It would be very helpful if these students could turn to their KR textbook for an accessible and comprehensive account of first-order logic.

Brachman and Levesque certainly are not unique in their decision to offer a perhaps overly concise introduction to first-order logic. Davis's *Representations of Commonsense Knowledge* makes a similar choice, and the other books reviewed in this article give summaries of first-order logic that are no more accessible than this. Interestingly, Russell and Norvig's comprehensive AI textbook *Artificial Intelligence: A Modern Approach* [28] gives an extremely thorough and easy-to-follow introduction to both propositional and first-order logic. They spend time motivating the need for first-order logic, and they use many natural language examples. That detailed a treatment would have been welcome in this text.

These, however, are minor points; they do not detract from the overall excellence of the text. An added bonus is that Levesque has made available at his web site a comprehensive set of slides (almost 300, covering nearly all of the book) to accompany this book.

This would make an excellent primary text for an undergraduate or first-year graduate KR course—augmented, if necessary, with remedial material on first-order logic. It is also suitable for a more senior graduate course, but should probably be supplemented with various papers, including a paper or two on a nonmonotonic logic, some basic papers on answer-set programming, and time permitting, papers on more advanced areas, such as epistemic logics and multi-agent reasoning.

3.2. Raymond Reiter: Knowledge in Action, MIT Press, 2001

This book is a synthesis and retelling of much of the work done by this great master during the last decade of his life. Reiter's work during that time touched on many of the central areas of KR and CSR.

Starting in the early 1990s, Reiter, who had previously been known for his seminal work in default reasoning, diagnosis, and relational databases, developed a strong interest in reasoning about action. His first major contribution in this area was a solution to the frame problem, at a time when the recalcitrance of that problem to seemingly straightforward solutions was causing considerable discomfort to the logicist AI community. The frame problem—the problem of efficiently determining what things remain the same in a changing world—had first been recognized by McCarthy and Hayes in the late 1960s [14]. With the development of nonmonotonic logics in the late 1970s and early 1980s, it was hoped that integrating nonmonotonicity into temporal reasoning languages would effectively solve the problem. The discovery of the Yale shooting problem [8] showed that a straightforward integration led to unexpected—or less tactfully put, wrong—solutions.

A few years later, Reiter, inspired by the work of Pednault [25], decided to explore whether it was possible to solve the frame problem within a *monotonic logic* [27]. Specifically, he integrated Pednault's work on the automated generation of frame axioms and Schubert's use of *explanation closure* axioms [30] (independently developed by [4]) into an elegant solution to the frame problem within the situation calculus. The solution worked for problems that could be expressed in a particular syntactic form, but this form was general enough to be broadly useful. It was in this paper that Reiter also extended Pednault's notion of *regression* in the situation calculus, showing how temporal projection queries—queries about what would be true in a particular situation, given what was true in the initial situation, and a description of the actions that had occurred—could be rephrased as queries about the initial situation.

This paper was the first of an explosion of groundbreaking papers on the situation calculus that Reiter produced in the 1990s. His research in the area included results on foundational axioms for the situation calculus, extensions to the situation calculus for concurrent actions, the development of a declarative logic programming language (Golog) which could be translated into a set of sentences in the situation calculus, the development of methods for expressing and reasoning with sensing actions and knowledge within the situation calculus, and the design and implementation of situation calculus planning systems. (Much of this work was done jointly with various researchers, including Hector Levesque, Fangzhen Lin, and Fiora Pirri.) Motivation for the development of his work came from a variety of sources, including Reiter's continued interest in database theory, but was mostly inspired by the ongoing work of the Cognitive Robotics Lab at Reiter's base at the University of Toronto. Getting robots to work in a real-world (even if controlled) environment prevents one from making the simplifying assumptions that pure theoreticians can make. In the real world, several actions may happen at the same time; thus there must be a way of dealing with concurrency. In the real world, robots must use sensors, and must use the results of their sensing actions to update their knowledge; thus, there must be some way of expressing and reasoning with sensing actions and knowledge.

Knowledge in Action includes much of the research that Reiter did in the 1990s. But it is far more than a compilation of his papers on the situation calculus. First, Reiter has simplified, unified, and made consistent his exposition throughout. Second, there are useful introductory sections on first-order logic, second-order logic, and the situation calculus, and there continues to be introductory and explanatory material throughout the book. Third, there is material in this book, such as the material on stochastic Golog in Chapter 12, that does not seem to be present in Reiter's papers. Fourth, and most important, the theme of developing the situation calculus for use in dynamical systems runs clearly through the book, and yields a superb example of how to pursue a research direction in a way that can profoundly influence and enhance an area of research. What the reader gets from this book is an illustration of how a scientist ought to do research. A scientist begins with some set of questions. However he answers them, whatever his initial results, it is nearly certain that in the course of doing his research, he has made simplifying assumptions, or restricted the set of problems that he examined, or come up with a set of results that were not as general as he would have wished. Ideally, the questions of how to minimize the simplifications or ease the restrictions become the motivation for future research. None of this is news, but it is not all that common that one sees it clearly. (Another example of this sort is Shanahan's [31] excellent book on solving the frame problem in the event calculus.) In Reiter's book, for example, the chapter on the "simple solution" to the frame problem explains that the simplified form in which effect axioms must be written precludes reasoning about indeterminate actions (actions with uncertain effects), nondeterministic actions, and complex actions, and points the reader to the chapters in which such actions are treated. In general, instead of starting out the book with an a priori list of all the results that he will show, Reiter lets us into the researcher's mind, showing the way the problems unfold, and how the solutions are developed. The book doesn't just teach; it inspires.

This is not to say that one of Reiter's central arguments—that the situation calculus can be extended to make it highly expressive and useful—is always wholly convincing. For example, Reiter demonstrates how one can express concurrency in the situation calculus. The idea is to talk about processes as well as actions. One adds an action that starts a process, an action that ends a process, and a fluent (some property that is true in some subset of situations) that describes the process. For example, one can have an action $startWalk(A, B)$, an action $endWalk(A, B)$, and a fluent $walking(s)$. In this way, for example, one can express walking and chewing gum at the same time (as long as the start and end actions for walking and chewing gum do not happen at exactly the same time). This seems to work for most applications, but it is hard to refrain from contrasting this labored approach to the natural way in which one can express concurrency in the event calculus. It would have been helpful to contrast different temporal formalisms, and to point out that while other formalisms may be more natural for expressing concurrency, the situation calculus is superior for expressing hypothetical reasoning, and is thus especially suitable for planning.

A minor quibble regarding the treatment of knowledge in Chapter 11 is that the presentation is not as clear as it might have been. Indeed, it may be quite confusing for those who have not already studied epistemic logics. The most common way to treat the knowledge operator *Know* is as a modal operator on sentences in a modal logic, where the semantics for *Know* are given in terms of a knowledge accessibility relation on possible worlds, as formalized by Kripke [10]. (Another popular method is to treat *Know* as a standard predicate on strings, where strings represent sentences in a first-order logic augmented with a quotation mechanism.) Reiter instead uses the approach of Moore [20], in which one identifies the situations of situation calculus with possible worlds, and one then directly expresses facts in terms of these situations and the knowledge accessibility relation. For example, to say that an (implicit) agent knows that BlockA is on BlockB in situation S0, one could say $\forall s\ K(s, S0) \supset On(BlockA, BlockB, s)$. Reiter then introduces *Know* as a macro.

There are certain advantages to this approach—one can stay in a first-order logic, and proofs are computationally simpler—but it is likely that a student who has not seen possible worlds semantics or modal logic before will have difficulty understanding this.

The diagrams in this section may also be a bit confusing. For example, Fig. 11.1 (page 284) shows an agent reasoning about two alternatives to the actual world. In the diagram, we see the real world, where Block A is on Block B, Block C is on Block D, and both Block B and Block D are on the table, as well as two epistemic alternatives to this world: one in which all blocks are on the table; and another in which Block A is on Block B and blocks B, C, and D are on the table. The text accompanying this figure makes it clear that these are the only epistemic alternatives, and that the agent is quite sure that Block C is on the table. In that case, however, the real world is not knowledge-accessible to the agent: What the agent is quite sure about is in fact false, and thus what is being discussed is belief rather than knowledge. All of this may serve to confuse the reader who is not already thoroughly familiar with this domain.

Another minor quibble: It would have been nice to see some mention of the need to reason about multiple agents, and some preliminary discussion about the challenges that this would pose for situation calculus theories of action.

But all these are minor cavils in what is otherwise a jewel of a book.

This is a superb book, essential for anyone who wants to seriously study theories of action. It could serve as an excellent secondary text in a knowledge representation or commonsense reasoning course, or as a primary text for a graduate level seminar on theories of action. It is probably too advanced, and does not have enough coverage of CSR and KR topics, to serve as a primary text for an undergraduate or beginning graduate course in KR or CSR.

3.3. Erik T. Mueller: Commonsense Reasoning, Morgan Kaufmann, 2006

Let me begin this section with a disclaimer: I am a colleague of Erik Mueller at IBM, and in fact, first introduced him to the event calculus, a temporal language of which I am very fond.

Commonsense Reasoning is a long and detailed exploration of the event calculus. The event calculus, originated by Kowalski and Sergot [9], is a language that has proved to be very versatile for knowledge representation. The ontology is a simple time line, and one can express the occurrence of an event by simply writing $\mathit{Happens}(e,t)$, where e is an event and t is a point of time. This makes it easy to express concurrency. The event calculus is particularly good for representing narratives in which some actions may be incompletely specified or missing. The event calculus was extended in several papers by Shanahan and Miller [17–19] to nonmonotonic logics. A book by Shanahan, *Solving the Frame Problem* [31], demonstrated how one could use circumscription and a particular formalization of the event calculus to solve the frame and ramification problems within a nonmonotonic logic.

Mueller goes through much of this material in detail. The material is often presented, in fact, in greater detail and with more examples than in Shanahan's *Solving the Frame Problem.* Indeed, Mueller's book can be viewed as a commentary on a portion of Shanahan's book and on several of Miller's and Shanahan's papers. Unlike Shanahan, who in his book explores representations of action within the situation calculus and explains how logic programming's negation-as-failure mechanism can be represented and used to solve temporal reasoning problems within both the situation calculus and the event calculus, Mueller devotes almost of all his book to the event calculus, compressing a discussion of other temporal formalisms (including the situation calculus, the fluent calculus, $\mathcal{C}+$, and Temporal Action Logics) into a single chapter at the end of the book.

The book starts with a brief introduction, in Chapter 2, to first-order logic, the basics of the event calculus, and circumscription. In addition to a lengthy discussion, in Chapters 3 through 9, of the event calculus—including event effects, event preconditions, and state constraints; event triggers; the principle of inertia; indirect effects; trajectory and anti-trajectory axioms; concurrent events; and nondeterministic effects—Mueller also presents, in Chapter 13, his own work on an implementation of a program for solving problems in the event calculus. The program, the Discrete Event Calculus Reasoner, works by encoding a problem in the event calculus into a satisfiability problem, using satisfiability solvers to solve the encoded problem, and then decoding the problem back into the language of the event calculus. The encoding is made feasible by the use of an event calculus restricted to a finite universe. (The program is available for download on the web.) Mueller provides in Chapter 14 a discussion of practical applications by several researchers that have used the event calculus, including the implementation of payment protocols, workflow modeling, and the higher-level vision component of a robot. The chapter also includes a description of Mueller's own research in using the event calculus for story understanding.

It is not clear why this book was named *Commonsense Reasoning*. The bulk of the book is about the event calculus, an important action formalism, but hardly something that comprises all or even most of commonsense reasoning. Sandwiched in between the exposition of previous results on the event calculus, and the presentation of Mueller's Discrete Event Calculus Reasoner, there are two chapters on domain theories of commonsense reasoning, one on spatial reasoning, and one on mental states. But these chapters do not seem to match, in spirit, the rest of the book.

Consider Chapter 11, which discusses mental states. Mueller introduces the notion of belief, which he says is a property considered to be true or false, and the notion of a goal, a property that the agent desires to be true. He introduces a belief sort and a goal sort, both of which are fluents. (It is not quite clear if these sorts are disjoint. The implicit assumption seems to be that they are disjoint; however, in the text, Mueller uses both the goal sort and the belief sort as arguments to belief.) He then proceeds to introduce axioms in the event calculus for adding and dropping beliefs and goals: for example, using the event calculus *Initiates* predicate to say $\mathit{Initiates}(\mathit{AddBelief}(a,b),\mathit{Believe}(a,b),t)$ which says that if the $\mathit{AddBelief}(a,b)$ action is initiated for any agent a and belief b at any time t, then a will start believing b.

Throughout the exposition and discussion of beliefs and goals, and throughout the presentation of the detailed example, there is no discussion of any semantics—possible worlds or otherwise—for belief. There is no discussion of any axioms on belief, no discussion of the properties that hold for belief. If an agent believes something, does he believe that he believes it? Does he believe the consequences of his knowledge? (How would that be expressed in this framework?) None of these issues are addressed.

Of course, there is no requirement to provide a semantics for one's belief operator, or to characterize one's belief operator with axioms. It is, however, accepted practice in much of the KR community to provide semantics and/or axiomatizations for one's operators. Moreover, Mueller abides by this practice in the first part of the book, which makes this chapter feel somewhat out of place.

The representation for belief and goals would seem to predate the work of Kripke and Moore; this contrasts with the up-to-date results given for the event calculus. It is interesting to see how the example proof in this section goes through. The example, long and complex, is of a cat who plans to get some food, whose plan is thwarted, who replans to get some other food in another way, and who serendipitously realizes, midway through executing his second plan, that he can get food in a quicker way. Mueller approaches the problem by taking as primitive a relatively large number of complex functions—such as an agent believing that a particular plan is a sound plan for achieving some goal—and then linking up these functions with correspondingly complex axioms.

Again, this approach contrasts oddly with the earlier part of the book, which presents the elegant and small set of carefully chosen primitives used by Shanahan and Miller. It also stands in contrast to the custom of so many KR and CSR practitioners—that of developing a theory using a small core of primitives (functions or predicates) and carefully defining other operators using this small set.

It would be helpful if there were some explanation of why this approach is taken, especially because this approach might potentially lead to difficulties. To illustrate this, consider again the example of the cat. Mueller expresses the fact (page 191) that if an agent is at a particular location and can jump to another location, then his initiating the jump from one location to the second will result in his being at the second location:

$$\mathit{HoldsAt}(\mathit{On}(a, s1), t) \wedge \mathit{HoldsAt}(\mathit{CanJump}(s1, s2), t) \Rightarrow \mathit{Initiates}(\mathit{Jump}(a, s1, s2), \mathit{On}(a, s2), t).$$

Consider the *CanJump* operator. Oddly, there is no agent argument for this operator. Note also that there is no logical connection between the *CanJump* and *Jump* operators, despite their similar appearance to the human eye. Another approach—one that was used by [20,21,27]—is to use some sort of general feasibility operator that would apply to actions. (Note that [20] and [21] discuss general feasibility operators in addititionto knowledge preconditions operators.) The use of such a general feasibility operator would have two related advantages. First, it would facilitate making explicit the general principles governing feasibility of action classes. Second, as a result, it could lead to a theory that is more elaboration tolerant. There is a sizable set of axioms concerning how *CanJump* links to *Jump*; if the theory considers cats who can walk, run, and climb, parallel sets of axioms will have to be added for these actions as well. Using the general feasibility operator could eliminate the need for such duplication, because one could reason about general feasibility principles.

The second section of Chapter 11 deals with emotions. Mueller uses in this section parts of the classification of emotions provided by Ortony, Clore, and Collins [24]. He introduces four fluents to represent factors used to specify the "eliciting conditions" of emotions: belief, desirability, praiseworthiness, and anticipation. He then introduces fluents for 22 emotions, 18 of which are non-compound emotions, including joy, hope, happyfor, relief, fear, disappointment, appreciation, reproach, and self-reproach, and 4 of which are compound emotions, namely, gratitude, anger, gratification, and remorse.

No semantics is given for these emotions. For each of these emotions, there are 3 axioms provided: two to describe how the fluent is initiated and terminated, and one to describe how the emotion is elicited. For example, the 3 axioms given for joy (pages 210, 213) are:

$$\mathit{Initiates}(\mathit{AddJoy}(a, e), \mathit{Joy}(a, e), t)$$

$$\mathit{Terminates}(\mathit{DropJoy}(a, e), \mathit{Joy}(a, e), t)$$

$$\neg \mathit{HoldsAt}(\mathit{Joy}(a, e), t) \wedge \mathit{HoldsAt}(\mathit{Desirability}(a, a, e, x), t) \wedge x > 0 \wedge \mathit{HoldsAt}(\mathit{Believe}(a, e), t) \Rightarrow$$
$$\mathit{Happens}(\mathit{AddJoy}(a, e), t)$$

The representation seems, first, to be a bit muddled. Events are listed (in Chapter 11, page 208) as both subsorts of belief and as not. In the formulas above, e seems to be an event type rather than an event, though this is not entirely clear. But what does it mean to believe in an event type? Or to be joyful about an event type? (One can be joyful about one occurrence of an event type, but sorrowful about another occurrence of the same event type.)

Moreover, and more fundamentally, it is unclear what these axioms mean. What is the connection between *Joy* and *AddJoy* beside their similar appearance in English? What, in short, is the semantics?

Except for the 4 compound emotions, there is no attempt to relate any of the emotions to one another. Thus, for example, there are no axioms in the theory to relate joy and relief, or joy and hope, or joy and happyfor.

Axioms to relate different emotions to one another are precisely what one would expect in a commonsense domain theory, because such axioms are what would allow one to prove interesting results. Their absence is therefore puzzling.

Perhaps Mueller was attempting to create the sort of shallow axiomatization that he and other proponents of the OpenMind project [32] have proposed to collect using mass human input. However, one would expect that even a shallow axiomatization would attempt to say something about how different emotions relate to one another.

Overall, this book appears to be best in its updated account of the event calculus.

3.4. Chitta Baral: Knowledge Representation, Reasoning and Declarative Problem Solving, Cambridge University Press, 2003

Answer-set programming, a form of logic programming based on *stable-set semantics*, is an area of knowledge representation that has received a great deal of attention during the last decade. Stable-set semantics is intuitively familiar to anyone who has studied model theory in a monotonic or nonmonotonic logic. A set of axioms will often—in fact, usually does—characterize a set of models, rather than a single model. In logic programming, an axiom in a certain form is called a *rule*, and a set of rules constitutes a program. Thus, a logic program—that is, a set of rules or axioms—can have multiple stable models. This is in contrast to what is referred to as *well-founded semantics* in which there is a single model for a logic program. This single model, however, is three-valued: sentences are true, false, or unknown.

There has been, in the past 15 years, a flurry of activity in both the theoretical and practical sides of answer-set programming (ASP). Chitta Baral is well versed in both. He has written dozens of papers about the foundations of ASP; at the same time, he has been involved in the development of large-scale systems using these techniques, including computational biology systems that reason about activities within a cell. In his book *Knowledge Representation, Reasoning and Declarative Problem Solving*, he makes both sides accessible to anyone with a formal AI background.

This is a very dense, information-rich book. There is no wasted space. The book needs to be read slowly; a single chapter can pack in a great deal of material. For all that, it is clearly written and well presented, so that slow, steady progress through the book is achievable with some work. It is also possible to dip into the book, omitting many of the foundational results, but still learning basic answer-set programming techniques.

The first chapter provides an excellent overview. The reader is introduced to the syntax for the basic classes of answer-set programs, AnsProlog, in Baral's terminology. Baral contrasts the declarative semantics of answer-set programming with the semi-procedural semantics of Prolog, arguing convincingly for the superiority of the declarative semantics and for the suitability of AnsProlog for knowledge representation. He also dives right into some important foundational results, including the relations between programs in different classes of AnsProlog programs.

Baral is clearly intent on making sure that the reader who gets through his book can join the troops of answer-set programmers. Chapter 2 focusses on simple modules, fragments of code that will frequently be re-used by an answer-set programmer. Essentially, Baral is sharing his bag of programming tricks with his readers. The bag of tricks includes ordinary declarative problem-solving modules (e.g., reasoning with integrity constraints, constrained enumeration, propositional satisfiability, and cardinality constraints) as well as knowledge representation and reasoning modules. These knowledge representation and reasoning modules include, for example, fragments of logic programs for dealing with inheritance with exceptions (the Tweety-penguin problem) and the Yale shooting problem. This is continued in Chapter 4, which presents the development of and solution to several problem domains, such as the n-queens problem. Likewise, Chapter 5, which shows how answer-set programs can be used to reason with the causal and temporal-reasoning language $\mathcal{A}$, contains numerous examples of specific code fragments detailing how one can handle the frame, qualification, and other related problems in theories of action. There is also an extensive development (in Chapter 7) of algorithms to compute answer sets and (in Chapter 8) discussion of specific systems such as Smodels and dlv.

But there is an equal emphasis on theory. (The theoretical material is interleaved with the more practical material in a sometimes unpredictable manner. Perhaps Baral felt that it was best not to overwhelm the reader with either too many code fragments or too many theoretical results at one time. To some extent, however, the frequent shift in focus detracts from the flow of the text.) There is much material (in Chapter 3) on foundational results on principles and properties of different classes of answer-set programs; this is extended in Chapter 5 to theorems guaranteeing the existence of models for certain classes of theories of action; and in Chapter 6 to complexity results, as well as to relations between various classes of AnsProlog and several KR formalisms.

Overall, this is an excellent introduction to and manual for answer-set programming, and for demonstrating how KR problems can be solved within an ASP system. However, there are a few ways in which the book might have been improved.

First, while it is a pleasure to work through a book that is so strong in theory and at the same time so rich with examples, the vast amount of examples and code fragments does come at a price. There are places in the text where these examples are not sufficiently motivated. This happens all too frequently in Chapter 5. Problems like the frame and ramification problems are subtle. A brief presentation of a list of axioms along with a concise English description of the problem will not necessarily hit home with readers who are unfamiliar with this problem. For example, in the initial discussion of the frame problem in Chapter 2, the reader is not explicitly told that the frame problem is a particular example of the multiple extension problem. From the pedagogical point of view, moreover, it is not necessarily best to present the correct solutions too soon; it is more fruitful, though it takes longer, to discuss several potential formalizations in depth, and point out their problems. In addition, the many, many variations of the frame problem and related reasoning problems discussed in Chapter 5 are almost sure to be overkill for non-experts. A slower, gentler approach is needed. Baral has missed an opportunity to turn Chapter 5 into a comprehensive introduction to the basic problems of temporal reasoning.

Second, there is at least one example, early in the book, where the choice of representation seems suspect. Baral uses (page 81) the predicate $m\_par(Y, X)$ to represent the fact that Y may be the parent of X. He then uses rules like $m\_par(Y, X) \Leftarrow not \neg par(Y, X)$, where *not* denotes negative by failure. *Maybe* seems suspiciously close to the modal operator *possibly*. The idea of subsuming what we know about *maybe* and *possibly* and trying to encode at least part of the meaning using negation as failure seems like a risky and error-prone enterprise, at least for the novice answer-set programmer.

On a related note, the methods that Baral demonstrates for turning off and on the closed-world assumption for various predicates, while powerful, may be confusing and error-prone, especially when trying to predict how all the closed-world assumption switches will interact. This is probably a trivial exercise for an expert like Baral, but for the novices that many of his readers are sure to be, it is likely to be confusing.

Third (a relatively minor point), references are given using a concatenation of an abbreviation of the author's name or initial letters of multiple authors' names, and the year of publication. This style is unfortunate. It is difficult to tell who the authors of a reference are without going to the bibliography. Moreover, the bibliography is alphabetized by abbreviated reference label and is thus not in alphabetical order.

Fourth, there are numerous small grammatical errors, such as instances of missing commas and misused determiners, throughout the book. It is a sign of the strength of the underlying writing that the text is nonetheless very clear.

Baral provides a set of slides for his book. These are not comprehensive, and do not cover all chapters of the book, but are still a useful accessory. He also provides software with which one can run the myriad examples in the book.

This is an essential text for a course on logic programming and answer-set programming. It would likewise serve well as a good secondary text for a knowledge representation course.

4. Conclusion

Both knowledge representation and formal commonsense reasoning have progressed markedly over the past several decades. A sign of the maturity of these areas of research is the publication of several textbook-length treatments of various aspects of these fields. A KR instructor can now use a single high-quality text that covers every essential topic in the KR syllabus.

There does not yet exist a single text that covers all desired topics in the CSR syllabus. Some KR textbooks, however, contain excellent treatments of certain aspects of CSR.

References

[1] F. Baader, W. Nutt, Basic description logics, in: The Description Logic Handbook, Cambridge University Press, Cambridge, 2003, pp. 43–95.
[2] S. Bechhofer, F. van Harmelen, J. Hendler, I. Horrocks, D. McGuinness, P. Patel-Schneider, L.A. Stein, OWL web ontology language reference, http://www.w3.org/TR/owl-ref/, 2004.
[3] A. Cohen, S. Hazarika, Qualitative reasoning: An overview, Fundamenta Informaticae 46 (1–2) (2001) 1–29.
[4] E. Davis, Representations of Commonsense Knowledge, Morgan Kaufmann, San Francisco, CA, 1990.

[5] E. Davis, The naive physics perplex, AI Magazine 19 (3) (1998) 51–79.
[6] E. Davis, The surprise birthday present problem, http://www-formal.stanford.edu/leora/commonsense/birthday, 2001.
[7] P. Ferraris, V. Lifschitz, Mathematical foundations of answer set programming, in: S.N. Artëmov, H. Barringer, A.S. d'Avila Garcez, L.C. Lamb, J. Woods (Eds.), We Will Show Them! (1), College Publications, 2005, pp. 615–664.
[8] S. Hanks, D.V. McDermott, Nonmonotonic logic and temporal projection, Artificial Intelligence 33 (3) (1987) 379–412.
[9] R.A. Kowalski, M.J. Sergot, A logic-based calculus of events, New Generation Computing 4 (1) (1986) 67–95.
[10] S. Kripke, A semantical analysis of modal logic i: Normal modal propositional calculi, Zeitschrift für Mathematische Logic und Grundlagen der Mathematik 9 (1963) 67–97.
[11] D. Lenat, R. Guha, Building Large Knowledge Based Systems: Representation and Inference in the Cyc Project, Addison–Wesley, Reading, MA, 1990.
[12] B. Mates, Elementary Logic, second ed., Oxford University Press, Oxford, 1972.
[13] C. Matuszek, M.J. Witbrock, R.C. Kahlert, J. Cabral, D. Schneider, P. Shah, D.B. Lenat, Searching for common sense: Populating cyc from the web, in: M.M. Veloso, S. Kambhampati (Eds.), AAAI, AAAI Press/The MIT Press, 2005, pp. 1430–1435.
[14] J. McCarthy, P.J. Hayes, Some philosophical problems from the standpoint of artificial intelligence, in: B. Meltzer, D. Michie (Eds.), Machine Intelligence 4, Edinburgh University Press, Edinburgh, 1969, pp. 463–502.
[15] J.L. McCarthy, Programs with common sense, in: Proceedings of the Teddington Conference on the Mechanization of Thought Processes, Her Majesty's Stationary Office, London, 1959, pp. 75–91.
[16] J.L. McCarthy, Elaboration tolerance, in: Working Papers of the Fourth International Symposium on Logical Formalizations of Commonsense Reasoning, Common Sense 98, 1998.
[17] R. Miller, M. Shanahan, Narratives in the situation calculus, J. Log. Comput. 4 (5) (1994) 513–530.
[18] R. Miller, M. Shanahan, Reasoning about discontinuities in the event calculus, in: KR, 1996, pp. 63–74.
[19] R. Miller, M. Shanahan, Some alternative formulations of the event calculus, in: A.C. Kakas, F. Sadri (Eds.), Computational Logic: Logic Programming and Beyond, in: Lecture Notes in Computer Science, vol. 2408, Springer, Berlin, 2002, pp. 452–490.
[20] R.C. Moore, Reasoning about knowledge and action, SRI Technical Report 191, 1980.
[21] L. Morgenstern, Foundations of a logic of knowledge, action, and communication, PhD thesis, 1988.
[22] L. Morgenstern, A first-order axiomatization of the surprise birthday present problem: Preliminary report, in: Proceedings of the Seventh International Symposium on Logical Formalizations of Commonsense Reasoning, 2005. Also published as Dresden Technical Report ISSN 1430-211X.
[23] L. Morgenstern, R.H. Thomason, Teaching knowledge representation: Challenges and proposals, in: A.G. Cohn, F. Giunchiglia, B. Selman (Eds.), KR 2000: Principles of Knowledge Representation and Reasoning: Proceedings of the Seventh International Conference, Morgan Kaufmann, 2000, pp. 725–733.
[24] A. Ortony, G.L. Clore, A. Collins, The Cognitive Structure of Emotions, Cambridge University Press, Cambridge, 1988.
[25] E.P.D. Pednault, Adl: Exploring the middle ground between strips and the situation calculus, in: KR, 1989, pp. 324–332.
[26] W. Pentney, A.-M. Popescu, S. Wang, H.A. Kautz, M. Philipose, Sensor-based understanding of daily life via large-scale use of common sense, in: AAAI, AAAI Press, 2006.
[27] R. Reiter, The frame problem in the situation calculus: A simple solution (sometimes) and a completeness result for goal regression, in: V. Lifschitz (Ed.), Artificial Intelligence and Mathematical Theory of Computation: Papers in Honor of John McCarthy, Academic Press, 1991, pp. 359–380.
[28] S. Russell, P. Norvig, Artificial Intelligence: A Modern Approach, second ed., Prentice Hall, Englewood Cliffs, NJ, 2003.
[29] E. Sandewall, Features and Fluents: The Representation of Knowledge about Dynamical Systems, vol. I, Oxford University Press, Oxford, 1994.
[30] L.K. Schubert, Monotonic solution of the frame problem in the situation calculus: An efficient method for worlds with fully specified actions, in: H. Kyburg, R. Loui, G. Carlson (Eds.), Knowledge Representation and Defeasible Reasoning, Kluwer Academic Publishers, Dordrecht, 1990.
[31] M. Shanahan, Solving the Frame Problem, MIT Press, Cambridge, MA, 1997.
[32] P. Singh, T. Lin, E.T. Mueller, G. Lim, T. Perkins, W.L. Zhu, Open mind common sense: Knowledge acquisition from the general public, in: CoopIS/DOA/ODBASE, 2002, pp. 1223–1237.