Ann. Rev. Comput. Sci. 1986. 1:255–87
Copyright © 1986 by Annual Reviews Inc. All rights reserved

KNOWLEDGE REPRESENTATION AND REASONING[1]

Hector J. Levesque[2]

Department of Computer Science, University of Toronto, Toronto, Ontario M5S 1A4 Canada

The notion of a *representation of knowledge* is at heart easy to understand. It simply has to do with writing down, in some language or communicative medium, descriptions or pictures that correspond in some salient way to the world or a state of the world. In artificial intelligence (AI) we are concerned with writing down descriptions of the world in which an intelligent machine might be embedded in such a way that the machine can come to new conclusions about its world by manipulating these symbolic representations. David Israel characterizes this "representation problem" as follows:

> All parties to the debate agree that a central goal of research is that computers must somehow come to "know" a good deal of what every human being knows about the world and about the organisms, natural or artificial, that inhabit it. This body of knowledge—indefinite, no doubt, in its boundaries—goes by the name "common sense." The problem we face is how to impart such knowledge to a robot. That is, how do we design a robot with a reasoning capacity sufficiently powerful and fruitful that when provided with some subbody of this knowledge, the robot will be able to generate enough of the rest to intelligently adapt to and exploit its environment? (68, p. 37)

Despite the apparent simplicity of this goal, the research area of knowledge representation (KR) has a long, complex, and as yet nonconvergent history.[3]

[1]This review draws heavily from work done with Ron Brachman. I also thank Jim des Rivières, Bart Selman, and John Tsotsos, for helping with an earlier version of the document. Financial support was received from the Natural Sciences and Engineering Research Council of Canada.

[2]Fellow of the Canadian Institute for Advanced Research.

[3]Research in KR originated with a single paper written by John McCarthy in 1958, and republished as (89).

Even though many current AI programs have a knowledge base (KB) containing symbolic descriptions in some "representation scheme" along the lines described above, there is still significant disagreement among researchers about many of the most fundamental issues. Part of the problem is that KR has evolved (and absorbed material) from a number of research areas with quite different goals and methods, such as psychology (in terms of realistic human memory models), linguistics (representations of word senses), philosophy (the nature of concepts and propositions), logic (varieties of formal reasoning), and computer science (information systems). For a closer look at this amazing diversity, see (131a), a very revealing questionnaire about KR answered by the researchers themselves.

But a more coherent interpretation based primarily on logic and computer science appears to be emerging. So rather than recapitulate the history of KR, this article takes a definite stand on the nature of the research. KR is intimately connected with reasoning, because an AI system will almost always need to generate explicitly at least some of what has been implicitly represented. Moreover, for sufficiently expressive representation languages, calculating these implications may be too demanding computationally, and so compromises are necessary. Knowledge representation, then, can be thought of as the study of what options are available in the use of a representation scheme to ensure the computational tractability of reasoning.

The idea of constructing systems that perform their tasks by reasoning with explicitly represented knowledge is just a working hypothesis about how to achieve generality, flexibility, modularity, and the like in complex systems. In the next section, we examine this hypothesis in more detail, and we show how it leads to a view of KR that is grounded in logic, on the one hand, and computational complexity, on the other. In fact, the main theme of this article is that we can understand much of KR research as attempts to reconcile simultaneously the demands of logical coherence and computational tractability.

Perhaps the most interesting type of KR research that is not well covered by this emphasis on the demands of tractable reasoning has been the attempt to formalize commonsense knowledge of the world. Indeed, Hayes argues (61) that the most pressing task for KR research is to analyze what kinds of knowledge need to be represented in specific subject areas, without too much concern (initially, at least) for how the knowledge will be represented or whether it can be effectively used by any system. This leads to a very different view of KR—a view that concentrates on formal representational theories of domains of general interest such as the properties of time (3, 95, 134), space (75), and liquids (60). Two collections of research papers in this style of "applied" KR research have been published (10, 66). For more comprehensive surveys of KR, see (6) and (105). An excellent general introduction to KR (and much of AI) is found in (107). Finally, an annotated collection of thirty of the most influential papers in the area has been published (22).

1. THE KNOWLEDGE REPRESENTATION HYPOTHESIS

The basic assumption underlying KR (and much of AI) is that thinking can be usefully understood as mechanical operations over symbolic representations. This hypothesis is, in fact, quite old, much older than computers, and seems to have originated with the philosopher Leibniz (1646–1716):

> There is little doubt, however, that Leibniz' ideas, which far outstripped in detail and understanding any earlier hints, were his own spontaneous creation. "While I was yet a boy with a knowledge only of common logic, and without instruction in mathematics, the thought came to me, I know not by what instinct, that an analysis of ideas could be devised, whence in some combinatory way, truths could arise and be estimated as though by numbers" *(Elementa Rationis).* He was thereafter constantly occupied with such notions and attempted to contrive an alphabet of thought, or *characteristica universalis,* which would represent ideas in a logical way, not things in a pictorial way, and would be mechanical in operation, unambiguous, and nonquantitative. (*The Encyclopedia of Philosophy,* Vol. 4, p. 538)

Just as there is a calculus of arithmetic, where numerical expressions are formally manipulated in a value-preserving way, so might there be a calculus of thought, where propositional expressions could be formally manipulated in a truth-preserving way.

This agrees remarkably well with most current views of KR within AI. Many AI systems represent knowledge as explicitly as possible, and (relatively) declaratively, in some formal language. The term *knowledge-based system* (or KBS) is currently popular in AI. An AI system can be described as knowledge-based not because knowledge is required to build it, nor just that it behaves as if it had knowledge, but rather that its architecture includes explicit KBs that are more or less direct symbolic encodings of the knowledge of the system. This is certainly true of Expert Systems [see (135), for example], currently the most visible and plentiful type of AI system.

This is not to say that all AI systems exhibiting knowledge are knowledge based in this sense. A typical game-playing program, for instance, might act as if it believed that it should bring its knight out early in the game. Yet this could be the result of applying a complex evaluation function to the leaves of a generated game tree. Specifically, there need not be any symbolic structure anywhere in the program corresponding to that belief. But according to the *Knowledge Representation hypothesis* (131), intelligence (or at least a certain kind of generality, versatility, etc.) is best served by explicitly representing in the data structures of a program as much as possible of what a system needs to know.

Implicit in this hypothesis[4] are two major properties that the structures forming a KB must satisfy:

- For the structures to represent knowledge, it must be possible to interpret them *propositionally*, that is, as expressions in a language with a *truth theory*. We should be able to point to one of them and say what the world would have to be like for it to be true.
- The system should act the way it does because of the presence of these structures. Clearly, the hypothesis would not be satisfied in a system where the KB was completely ignored (like comments in a program, for example).

The key point is that an account of cognitive activity in terms of computational operations over propositionally interpreted structures imposes constraints on how a KBS can be realized. First of all, it rules out data structure operations that do not respect (in some sense) the propositional interpretation of a KB (e.g. reversing the words of a sentence). Secondly, because of the causal role of a KB, it rules out operations that are not computationally manageable. In other words, the operations on a KB need to be semantically coherent without demanding more than what any computer can be expected to do. To better understand these constraints, we need to examine what it means to operate on structures in a way that respects their semantic interpretation.

2. KNOWLEDGE REPRESENTATION AND LOGIC

Wanting to deal with truth-preserving operations over symbolic structures puts us into the domain of *logic*. Indeed, standard formal logic is a perfect example of a system that deals with symbolic structures (its well-formed formulas) and operations over them that, on the one hand, can be defined purely formally (as specified by a proof theory) and, on the other hand, can be shown to preserve semantic properties (as specified by a truth theory).

For a detailed explanation of formal logic see, for example, (97). Recall, however, that there are two major relationships of interest defined over the sentences of a logical language.

- *logical consequence* A sentence α is a logical consequence of a set of sentences S (written $S \models \alpha$) iff any interpretation of the logic that makes every sentence in S true also makes α true. A sentence is *valid* if it is a logical consequence of the empty set.

[4]It is worth keeping in mind that this is only a working hypothesis. Perhaps its strongest defense is that it seems to be "the only game in town"; but even there, alternative trends appear to be emerging (123). For a criticism of this hypothesis on more philosophical grounds, see (39).

- *derivability* A sentence α is derivable from a set of sentences S (written $S \vdash \alpha$) iff there is a sequence of sentences $\alpha_1, \ldots, \alpha_n$ where α_n is α, such that each α_i either is an element of S, or a logical axiom, or follows from earlier α_j's by a rule of inference. A sentence is a *theorem* if it is derivable from the empty set.

Once a language has been fixed, the logic must specify the allowable interpretations, on the one hand, and the axioms and rules of inference, on the other.

The main point is that derivability in formal logic is defined in a way that does not mention what the formulas are intended to mean. Yet soundness and completeness properties can establish links between the two: A sentence will be derivable from a set of sentences precisely when it is a logical consequence of that set. For logics that are sound and complete, this is the sense in which purely formal operations can be shown to preserve semantic properties, as required by the KR hypothesis.

Logic however, does not guarantee the computational tractability of the operations of its proof theory. The derivability of a formula depends on the existence of a certain sequence of formulas, but not on whether or not that sequence can be found purely mechanically, or in a reasonable amount of time. So it is not immediately obvious how logic can be made to play the causal role required by the KR hypothesis.

Indeed, the role of formal logic in KR has been hotly contested from the very beginning.[5] Part of the problem concerns the history and goals of KR and their differences from those of symbolic logic. After Leibniz, the next big advance in formal logic was the work of Frege at the turn of the century who, along with Russell, Peano, and others, gave logic much of the flavor it has today. The goal of this early work was to put mathematics and mathematical reasoning on a sound theoretical footing. Indeed the major application of symbolic logic until recently, and one of the success stories of twentieth century mathematics, was the analysis of formal theories of sets and numbers.

The goals of KR, however, were always quite different, and perhaps much more in line with Leibniz' original dream. On the one hand, KR schemes were being used to represent the semantic content of natural language concepts (e.g. 125, 128), and on the other, to represent psychologically plausible memory models (e.g. 109, 112). In neither case was there a clear relationship to formal languages of any kind [as Woods, among others, has pointed out (141)]. Gradually, however, KR schemes began to be used as a very flexible and modular way to represent the facts that a system needed to know to

[5]To get an idea of the range of opinions involved at various points, see (58, 99, 102, 111).

behave intelligently in a complex environment [an early example is (137)].

This view of KR, involving propositional representations of the beliefs of a system, along the lines of the KR hypothesis, has slowly come to dominate the field. With it has come an emerging consensus about at least some of the issues relating KR and logic. For example, the need for a truth theory for representation languages is generally acknowledged. Without some concrete specification of the meaning of a notational convention, what is implied by an expression in that language is unclear, and comparisons to other notational systems are impossible (57). Also, it is generally accepted that some form of deductive reasoning will be necessary to extract what is implicit in the explicit beliefs to which a system has access. When a KBS needs to know something about its domain, it must use the symbolic structures representing what it knows in order to decide what is true or false about the world. In general, there is no guarantee that what it needs to know will be represented directly in these data structures, but it may very well be logically implicit in these structures.

Suppose, for example, that a medical diagnosis system needs to know whether or not a patient P is allergic to medication M before prescribing a treatment. It may be the case that the KB in question has a structure representing the fact that

Patient P is allergic to medication M.

For instance, there may be a table in a database relating patients to the medicines to which they are allergic. In this case, the system can determine by retrieval alone whether or not it can use M. On the other hand, suppose the KB has explicitly represented only the following facts:

Patient P is allergic to medication M*.

and

Anybody allergic to M* is also allergic to M.

In this case, it will be necessary to determine facts that are only implicitly represented in order to be able to decide what medication can be prescribed. In general, a KBS will be concerned not with what is explicitly represented in a KB but, rather, with what these structures taken together tell about its application domain. In other words, the main concern of a KBS is what information is contained either implicitly or explicitly in a collection of propositionally interpreted data structures.

To a first approximation then, if we restrict our attention to the yes-no questions about the world that a system might be interested in, a KR system has to be able to determine the logical consequences of what is present in the KB. In other words, given a yes-no question α, and a KB understood as a collection of sentences, a KR system must be able to determine whether or not KB $\models \alpha$.[6] So if, as a starting point, we now fix the representation language to be the first-order predicate calculus (or FOL)[7], we can give a very precise definition of KB $\models \alpha$ (and thus, the reasoning service to be provided by the KR system). Moreover, if we assume that the KB can be understood as a finite set of sentences and let KB stand for their conjunction, then because of soundness, completeness, and other properties of FOL, KB $\models \alpha$ iff the sentence (KB $\supset \alpha$) is derivable from no premises, that is, it is a theorem of FOL. Thus, the question-answering operation becomes one of *theorem proving* in FOL.

2.1 The Problem

The good news in reducing the KR service to theorem proving is that we now have a very clear, very specific notion of what the KR system should do; the bad news is that it is also clear that this service cannot be provided. The fact is that deciding whether or not a sentence of FOL is a theorem is unsolvable [again, see (97)]. Moreover, even if we restrict the language practically to the point of triviality by eliminating the quantifiers, the decision problem, though now solvable, does not appear to be solvable in anywhere near reasonable time.[8] It is important to realize that this is a property not of particular algorithms that have been discovered so far, but of the problem itself: There cannot be an algorithm that does the test for theoremhood correctly in a reasonable amount of time. This bodes poorly, to say the least, for a service that is supposed to be only a part of a larger KBS.

One aspect of these intractability results is that they deal with the worst-case behavior of algorithms. In practice, a given theorem-proving algorithm may work quite well. In other words, a given program might behave properly for a very wide range of questions, even though there will always be questions whose answers will not be returned for a very long time, if at all. How serious is the problem, then? To a large extent this depends on the kind of question asked of a KR subsystem. The worst-case prospect might be perfectly tolerable in a mathematical application where a question is an open problem in mathematics. Provided that progress is being made, a user might be quite

[6]In this view of a KR system, the service it provides to a knowledge-based system depends only on the truth theory of the language of representation. No commitment is necessary about how such a service can be realized, if indeed it can. This is in keeping with what we have called elsewhere a *functional* view of KR (see 18, 79), where the service performed by a KR system is defined separately from the techniques a system might use to realize that service.

[7]See (97) for details on this language. In what follows, we will have reason to consider other languages as well, but mainly as departures from FOL.

[8]The problem is now co-NP-hard, meaning that it is strongly believed to be computationally intractable (see 52).

willing to stop and redirect a theorem prover after a few months, if it seems to be thrashing. Worst-case behavior is irrelevant; this might be the only case of interest.

But imagine, on the other hand, a robot that needs to know about its external world (such as whether or not it is raining outside, where its umbrella is, and so on) before it can act. If this robot has to repeatedly invoke a KR utility as a subroutine on a wide range of questions, the worst case prospect is much more serious. To be bogged down on a logically difficult but low-level subgoal and be unable to continue without human intervention is clearly an unreasonable form of behavior for something aspiring to intelligence.

But "on the average" the robot might do alright. The trouble is that nobody seems to be able to characterize what an "average" case might be like. As responsible computer scientists, we should not be providing a general inferential service if all that can be said about it is that by and large it will probably work satisfactorily. If the KR service is to be a module in a larger system and if it is not available for introspection or control, then it had better be dependable both in terms of its correctness and the resources it consumes. Unfortunately, this rules out a service based on theorem proving (in full first-order logic).

2.2 *Some Pseudo-Solutions*

There are at least two fairly obvious ways to minimize the intractability problem. The first is to push the computational barrier as far back as possible. An entire subarea of AI called *Automatic Theorem Proving* (ATP) studies techniques for avoiding redundancies and speeding up certain operations in theorem provers.[9] Significant progress has been achieved here, which allows open questions in mathematics to be answered (136, 143). Along similar lines, VLSI and parallel architectural support no doubt will improve the performance of theorem provers (85).

The second way to make theorem provers more usable is to relax our notion of correctness. A very simple way of doing this is to make a theorem-proving program always return an answer after a certain amount of time. If it has been unable to prove either that a sentence or its negation is implicit in the KB, it could assume that it was independent of the KB and answer "unknown" (or maybe reassess the importance of the question and try again). This form of error (i.e. one introduced by a resource-limited theorem prover), is not nearly as serious as returning a "yes" for a "no", and is obviously preferrable to an answer that never arrives.[10]

[9]A useful introduction to ATP is (86). An overview of the research in the area can be found in (142).

[10]See (108, 139) for a fuller discussion of resource-limited processing. Many of these ideas were eventually incorporated into the KRL (11) representation language.

However, from the point of view of KR, both of these are only pseudo-solutions. Clearly, the first one alone does not help us guarantee anything about an inferential service. The second one, on the other hand, might allow us to guarantee an answer within certain time bounds but would make it very hard for us to specify what that answer would be. If we think of the KR service as reasoning according to a certain logic, then the logic being followed is immensely complicated (compared to that of FOL) when resource limitations are present. Indeed, the whole notion of the KR system calculating what is implicit in the KB (which was our original goal) would have to be replaced by some other notion that went beyond the truth theory of the representation language to include the inferential power of a particular theorem-proving program. In a nutshell, we can guarantee getting an answer, but not the one we wanted.

2.3 Knowledge Representation as Compromise

So there are serious problems with a KR service that attempts to calculate in a reasonable amount of time what is logically implicit in a KB. In this review we demonstrate that much of the research in KR can be construed as trading off a certain generality offered by FOL for a more tractable form of inference. To see where this trade-off between expressiveness and tractability originates, we should first look at the use of the expressive power of FOL in KR and how it differs from its use in mathematics.

In the study of mathematical foundations, the main use of FOL is in the formalization of infinite collections of entities. So, for example, we have first-order theories of numbers and sets that use quantifiers to range over these classes, and conditionals to state the properties that these entities have. This is exactly how Frege intended his formalism to be used.

In KR, on the other hand, the domains being characterized are usually finite. The power of FOL is used not so much to deal with infinities but to deal with *incomplete knowledge* (77, 78, 102).[11] Consider the kind of facts[12] that might be represented using FOL:

1. $\neg$Student(john).

Sentence 1 says that John is not a student, without saying what he is.

2. Parent(sue,bill) $\vee$ Parent(sue,george).

Sentence 2 says that either Bill or George is a parent of Sue, but it does not specify which one is the parent.

[11]A KB is said to be incomplete if it tells us that one of a (possibly infinite) set of sentences is true without telling us which.

[12]The use of FOL to capture *terminology* or laws is somewhat different [see (20) for details].

3. $\exists x$ Cousin(bill,x) $\wedge$ Male(x).

Sentence 3 says that Bill has at least one male cousin, but it does not say who that cousin is.

4. $\forall x$ Friend(george,x) $\supset$ $\exists y$ Child(x,y).

Sentence 4 says that all of George's friends have children, without saying who those friends or their children are, or even if there are any.
The main feature of these examples is that FOL is used not to capture complex details about the domain but to avoid having to represent details that may not be known. The expressive power of FOL determines not so much what can be said but what can be left unsaid.

For a system that has to be able to acquire arbitrary knowledge in a piecemeal fashion, there may be no alternative to full logical reasoning with a language as expressive as FOL. But we may be able to get by with much less. In what follows, we examine KR research by considering three distinct ways of reducing the computational demands on a general KR reasoning service.

1. *special-purpose languages* By limiting what can be expressed in a KR language, we can rule out certain forms of weak statements allowed by full FOL. Certain types of incomplete knowledge may not be representable, but full logical reasoning may now be within reach. The result, then, is a KB that is forced to be more complete because of the inexpressiveness of its representation language.
2. *models of limited reasoning* Another possibility is to leave the expressiveness unchanged, but to give up trying to ferret out all the implications of an incomplete KB by considering a form of implication that is weaker than full logical consequence. Because only some of the possibilities admitted by an incomplete KB will be considered during reasoning, an incomplete KB ends up being treated as if it were more complete.
3. *defaults and assumptions* Finally, we can leave the language and its logic unchanged but can transform the incomplete KB itself into one that is more complete. By making assumptions based on the absence of contradicting evidence, a KR system may come to conclusions not necessarily implied logically by the original KB. The result of this type of reasoning is that an incomplete KB will be made more complete by filling in the holes in what it knows.

So the view of KR we are considering here is certainly based on logic, but not necessarily classical logic. Rather than dealing with a very expressive language (i.e. FOL), and inferences that are logically complete and sound, we are concerned with inexpressive languages and with logics that are classically incomplete and unsound. First, we will look at some special-purpose KR languages, focussing on their expressive limitations. Next, we will look at logics of knowledge and belief and how these can be made to provide a weaker notion of implication than that of standard logic. Finally, we will examine some of the research in nonmonotonic reasoning in an attempt to characterize the use of defaults and assumptions. In all cases, the main point is that there is much to be gained by looking at the logical form of what is being represented and how the necessary reasoning can be kept (or at least has a hope of being kept) computationally tractable.

As for other sources on this topic outside of AI, there is unfortunately almost nothing in philosophy on managing the dual concerns of logic and computational complexity [a notable exception is (27)]. There have been results in theoretical computer science and mathematical logic in special cases of logics that are computationally manageable (e.g. see 83), but none of these results focus on whether these special cases are representationally significant. Although the issues addressed here provide a very useful perspective on KR research, they are only beginning to be examined in depth.

3. SPECIAL-PURPOSE LANGUAGES

A KR language with restricted expressive power is simply a language whose expressions can be understood propositionally, but without being closed under the normal logical operations of disjunction, existential quantification, and so on. Given that various types of incomplete knowledge will no longer be representable in such a language, the logical consequences of a KB can often be calculated using efficient special-purpose techniques, for example, avoiding, time-consuming (or infinite) case analyses and their equivalents.

Consider, for instance, information that we may have about two positive integral quantities m and n expressed in a first-order logical language that has symbols for the arithmetic operations and relations. So, for example, we might know that $2m - n = 6$ and that either $mn = 80$ or $m^2 < 5n$. If we call these two facts E and add to them Q, the single axiom of Robinson's system of arithmetic (97), then what follows logically is a number of properties of m and n, such as that $2m + n > 20$. This means that this property (call it P) is implicit in a KB that contains E and Q. Moreover, standard theorem proving (among other methods) can be used to discover this, since $((E \wedge Q) \supset P)$ must be a theorem of first-order logic.

But suppose that instead of E, we know E', which says that $2m - n = 6$ and $m + n = 15$. Again, P follows from E' and Q, and again, we could determine this using first-order theorem proving. But since our knowledge about m and n is expressible as a pair of linear equations, there is a much better way to do this: Solve the set of equations (using Gaussian elimination with back substitution) and calculate directly whether or not P is true of these values. In general, k equations can be solved in roughly k^3 operations, and assuming a unique solution, P can be checked in a linear number of operations.[13] Specialized techniques like this are not possible in general, and sets of linear equations are only a small subset of what can be expressed in the full language of first-order arithmetic. But it is nonetheless a very rich and useful subset. So it makes sense to build systems whose input language is restricted to lie in this subset.

Much of the work in KR has involved inventing new KR formalisms and embedding these within KR systems. In retrospect, a good proportion of this research can be seen as the search for useful compromises between expressive power, on the one hand, and tractability of reasoning, on the other (80). In fact, with the possible exception of nonmonotonic facilities (see Section 5 below), these formalisms can almost inevitably be understood as subsets of classical first-order logic.

3.1 Databases

The most obvious restriction to the form of a KB is what might be called *database form*. The idea is to restrict a KB to contain only the kind of information that can be represented in a simple standard database. Consider, for example, a very trivial database that deals with university courses. If we had to characterize in FOL the information contained in the database, we might use a collection of function-free atomic sentences like

```
Course(csc248)   Dept(csc373,ComputerScience)    Enrollment(psy400,42) . . .
Course(mat100)   Dept(his100,History)            . . .
   . . .
```

In other words, a standard tabular database characterizes exactly the positive instances of the various predicates. But more to the point, since we never end up with sentences like

Dept(mat100,Mathematics) $\lor$ Dept(mat100,History),

certain kinds of incomplete knowledge cannot be represented.

There is, moreover, additional hidden information in a standard database that makes it even more complete. Consider the question

How many courses are offered by the Computer Science Department?

The knowledge expressed by the above FOL sentences is insufficient to answer the question: Nothing says that Computer Science has at least two courses (since "csc373" and "csc248" could be names of the same individual), and nothing says that it has at most two courses (since there could be courses

[13]This will be true of any P that can be evaluated directly once the values of its variables are known. So, for example, any P that does not use quantifiers has this property.

other than those mentioned in the list of sentences). But a database system could answer the question successfully by interpreting it as (something like)

> How many tuples in the COURSE relation have ComputerScience in their Dept field?

This is a question not about the world being modelled but about the data itself. To be able to reinterpret it as the intuitive question about courses and departments (rather than as one about tuples and fields), we need to account for additional information taking us beyond the stored data itself. In particular, we need FOL sentences of the form

$c_i \neq c_j$, for distinct constants c_i and c_j,

stating that each constant represents a unique individual. In addition, for each predicate, we need a sentence similar in form to

$\forall x[\text{Course}(x) \supset x = \text{csc248} \vee \ldots \vee x = \text{mat100}]$,

saying that the only instances of the predicate are the ones named explicitly.[14] If we now consider a KB consisting of all of these sentences, a KR system would indeed conclude (just like its database management counterpart) that there were exactly two Computer Science courses.

The main observation here is that answering questions with a KB in database form is much easier than in the general case. Because there is no incompleteness in our knowledge, inference reduces to simple calculation. We do not, for instance, have to reason by cases or by contradiction, as we might in a more general setting, and we can represent symbolically what is known about the world using sets of tuples, exactly like a standard database system. Then, to infer the number of courses of a certain type, all we have to do is count how many appropriate tuples appear in the COURSE relation. From this perspective, a database is a knowledge base whose limited form permits a very special form of inference.

Although database languages are almost always too weak for AI purposes, the limitations they impose on the logical form of a KB are found in many KR languages. There is very close structural correspondence between some part of the KB and the domain of interest: For each entity in the domain, there is a unique representational object that stands for it; for each relationship that it participates in, there is some sort of connection in the KB that corresponds to it. In a very real sense, that part of the KB is an *analogue* of the domain of interest, not so different from other analogues such as maps or physical models. The main advantage of having such an analogue is that it can be used directly to answer questions about the domain. Calculations on the model

[14]This is one form of what has been called the *closed world assumption* (118).

itself play the role of more general reasoning procedures (much the way arithmetic can replace reasoning with Peano's axioms). The disadvantage of an analogue, however, should also be clear: There are certain kinds of facts about the domain that it cannot leave unsaid.[15] In this sense, an analogue representation can be viewed as a special case of a propositional one, where the information it contains is relatively complete.

3.2 Logic Programs

The second restriction on the form of a KB that we will consider is a generalization of the previous one found in programs written in PROLOG (73), PLANNER (62), and related languages.[16] A KB in logic program form contains a collection of first-order sentences (called Horn sentences) of the form

$$\forall x_1 \ldots x_n[P_1 \wedge \ldots \wedge P_m \supset P_{m+1}] \quad \text{where } m \geq 0 \text{ and each } P_i \text{ is atomic.}$$

In the case where $m = 0$ and the arguments to the predicates are all constants, the logic program form coincides with the database form. As in the database case, since we are interested in more than just the universe of terms, we again have to include additional facts in the KB. In general, the resulting KB will be infinite, since it must contain all sentences of the form $(s \neq t)$, for any two distinct terms formed using function and constant symbols appearing in the KB. As before, the KB must also contain a version of the closed world assumption, which is now the negation of every ground atomic sentence not implied by the Horn sentences in the original KB.

The net result is a KB that once again has complete knowledge of the world (within a given language),[17] but this time it requires nontrivial inference to answer questions. The necessary reasoning is the *execution* of the logic program. For example, given a KB in logic program form consisting of

```
parent(bill,mary).
parent(bill,sam).
parent(X,Y), female(Y) ⊃ mother(X,Y).
female(mary).
```

[15]The same is true for the standard analogues. A map does not allow you to say, for example, that a river passes through one of two widely separated towns, without specifying which town. Similarly, a plastic model of a ship cannot tell us that the ship it represents does not have two smokestacks, without also telling us how many it does have. This is not to say that there is no uncertainty associated with an analogue, but that this uncertainty is limited in various ways. See (82, 129, 130) for more information and see (51) for the use of analogues in problem solving.

[16]See Section 5.2 for another view of the procedural school of KR.

[17]Notice that it is impossible to state in a KB of this form that $(P \vee Q)$ is true without stating which one is true, or that $\exists x P(x)$ is true without saying what x is.

we know exactly who the mother of Bill is, but only after having executed the program.

In some sense, the logic program form does not provide any computational advantage to an FOL reasoning system, since determining if a ground atomic sentence is implied by a collection of Horn sentences (containing function symbols) is undecidable.[18] On the other hand, the form is much more manageable than in the general case, since the necessary inference can be split very nicely into two components: a *retrieval* component that extracts (atomic) facts from a database by pattern-matching, and a *search* component that tries to use the nonatomic Horn sentences to complete the inference. In actual systems like PROLOG and PLANNER, moreover, the search component is (partially) under user control, giving the user the ability to incorporate domain-specific control knowledge. The only purely automatic inference is the retrieval component.

This suggests a different way of looking at the inferential service provided by a KR system (without even taking into account the logical form of the KB). Instead of automatically performing the full deduction necessary to answer questions, a KR system could manage a limited form of inference and leave to the rest of the knowledge-based system (or to the user) the responsibility of intelligently completing it. As suggested in (50), the idea is to take the "muscle" out of the automatic component and leave the difficult part of reasoning as a problem that the overall system can (meta-)reason about and plan to solve (54).[19] It is clear that one of the major attractions of PROLOG is its programmability, and its potential for integrating procedural and declarative concerns, rather than its power as a KR language.[20]

3.3 Semantic Networks

Whereas databases and logic programs have been studied quite independently of KR, semantic networks are much more of a pure KR phenomenon.[21] The first observation about a KB in this form is that it only contains unary and binary predicates. For example, instead of representing the fact that John's grade in cs100 was 85 by

[18]In many simple Expert System applications, however, the problem is solvable because no function symbols (except for constants) are used.

[19]This topic is discussed further in Section 4.

[20]For more information on PROLOG as a programming language, see (29, 74).

[21]Originally, semantic networks were proposed as models of associative memory (e.g. 112). Gradually they came to be used for general KR purposes, but not without some confusion regarding how they were to be interpreted (15, 16, 141). For a collection of research papers demonstrating some of the variety in semantic network representations, see (47).

Grade(john, cs100, 85),

we might postulate the existence of objects called "grade-assignments" and represent the fact about John in terms of a particular grade-assignment g_1 as

Grade-assignment(g_1) $\wedge$ Student(g_1,john) $\wedge$ Course(g_1,cs100) $\wedge$ Mark(g_1,85).

This part of a KB in semantic net form is also in database form: a collection of function-free ground atoms, along with sentences stating the uniqueness of constants and closed world assumptions.

The main feature of a semantic net (and of frames, below), however, is not how individuals are handled, but the treatment of the unary predicates (which we will call *types*) and the binary ones (which we will call *attributes*). First, types are organized into a taxonomy, which, for our purposes, can be treated as a set of sentences of the form[22]

$\forall x[B(x) \supset A(x)]$.

Finally, there are sentences that place constraints on an attribute as it applies to instances of a type:

$\forall x[B(x) \supset \exists y(R(x,y) \wedge V(y))]$ or $\forall x[B(x) \supset R(x,c)]$.

One property of a KB in this form is that it can be represented by a labelled directed graph (and displayed in the usual way). The nodes are either constants or types, and the edges are labelled either with an attribute or with the special label *is-a*. The significance of this graphical representation is that it allows certain kinds of inference to be performed by simple graph-searching techniques. For example, to find out if a particular individual has a certain attribute, it is sufficient to search from the constant representing that individual, up *is-a* links, for a node having an edge labelled with the attribute. By placing the attribute as high as possible in the taxonomy, all individuals below it can *inherit* the property. Computationally, any improvement in the efficiency of graph-searching will improve the performance of inference in a KB of this form (although in general, of course, graph-theoretic operations can be as intractable as any other operations).

In addition, for better or for worse, the graph representation suggests different kinds of inference based more directly on the structure of the KB than on its logical content. For example, we can ask for the connection between two nodes and answer by finding a path in the graph between them.[23] Another example is to use the *is-a* taxonomy for default reasoning.[24] For

[22]See (16, 59) for a discussion of some of the subtleties involved here.

[23]Quillian (113) proposed a "semantic intersection" approach to answering questions in his original work on semantic nets. See also (30) for later work on the same topic.

[24]Default reasoning is discussed in detail in Section 5.

instance, the *elephant* node can have a *color* link to the value *gray,* but anything below *elephant* (such as *albino-elephant*) can be linked to a different color value. To infer the color of an individual only requires searching up the taxonomy for a value, and stopping when the first one is found, preempting any higher values.[25] The trouble with this type of reasoning is that, with only a procedural account like the one above, it is very easy to lose the import of exactly what is being represented (17).

3.4 Frames

The final form we consider—frame descriptions—is mainly an elaboration of the semantic network form.[26] The emphasis, in this case, is on the structure of the types themselves (usually called *frames*) in terms of their attributes (now called *slots*). Typically, the kind of detail associated with the slots of a frame includes

1. *values,* stating exactly what the attribute of an instance should be. Alternatively, the value may be just a *default,* in which case an individual inherits the value, provided it does not override it.
2. *restrictions,* stating what constraints must be satisfied by attribute values. These can be *value* restrictions, specified by a type that attribute values should be instances of, or *number* restrictions, specified in terms of a minimum and a maximum number of attribute values.
3. *attached procedures,* providing procedural advice on how the attribute should be used. An *if-needed* procedure explains how to calculate attribute values if none have been specified; an *if-added* procedure explains what should be done when a new value is discovered.

Like semantic networks, frame languages tend to take liberties with logical form, and the developers of these languages have been notoriously lax in characterizing their truth theories (17, 59). Restricting ourselves to a noncontroversial subset of a frame language, we might have descriptions like

```
(Student
   with a dept is computer-science and
   with ≥ 3 enrolled-course is a
      (Graduate-Course with a dept is a Engineering-Department)).
```

This is intended to be a structured type that describes Computer Science students taking at least three graduate courses in departments within Engineer-

[25]For more complex examples of reasoning based directly on the form of semantic networks, see (121) and (140).

[26]The theory of frames in KR was first presented in (99), although many of the ideas were already "in the air". KRL (11–13) and KL-ONE (14, 23) are perhaps the most representative KR languages based on these ideas.

ing. If this type had a name (say A), we could express the type in FOL by a "meaning postulate" of the form

$$\begin{aligned}\forall x A(x) \equiv [&\text{Student}(x) \wedge \text{dept}(x, \text{computer-science}) \wedge \\ &\exists y_1 y_2 y_3\, (y_1 \neq y_2 \wedge y_1 \neq y_3 \wedge y_2 \neq y_3 \wedge \\ &\quad \text{enrolled-course}(x,y_1) \wedge \text{Graduate-Course}(y_1) \wedge \\ &\quad\quad \exists z(\text{dept}(y_1,z) \wedge \text{Engineering-Department}(z)) \wedge \\ &\quad \text{enrolled-course}(x,y_2) \wedge \text{Graduate-Course}(y_2) \wedge \\ &\quad\quad \exists z(\text{dept}(y_2,z) \wedge \text{Engineering-Department}(z)) \wedge \\ &\quad \text{enrolled-course}(x,y_3) \wedge \text{Graduate-Course}(y_3) \wedge \\ &\quad\quad \exists z(\text{dept}(y_3,z) \wedge \text{Engineering-Department}(z)))].\end{aligned}$$

Similarly, it should be clear how to state equally clumsily[27] in FOL that an individual is an instance of this type.

One interesting property of these structured types is that we do not have to explicitly assert that one of them is below another in the taxonomy. The descriptions themselves implicitly define a taxonomy of *subsumption*, where type A is subsumed by type B if, by virtue of the form of A and B, every instance of A must be an instance of B. For example, without any world knowledge at all, we can determine that the type

(Person with every male friend is a Doctor)

subsumes

(Person with every friend is a (Doctor with a specialty is surgery)).

Analytic relationships like subsumption are useful properties of structured types that are not available in a semantic network where all of the types are essentially atomic (20). In the KRYPTON KR language (18, 19), a full first-order KB is used to represent facts about the world, but subsumption information is also available. The reason for this is that while subsumption can be defined in terms of logical implication,[28] there can be very good special-purpose "description matching" algorithms for calculating these relationships (21). Again, because the logical form is sufficiently constrained, the required inference can be much more tractable.

3.5 Other Possibilities

One final class of special-purpose language worth mentioning are those that are geared to specific subject matters. For example, the representation of

[27]These sentences are especially awkward in FOL because of the number restrictions. For example, the sentence "There are a hundred billion stars in the Milky Way Galaxy" would be translated into an FOL sentence with on the order of 10^{22} conjuncts.

[28]Specifically, type A is subsumed by type B iff the meaning postulates for A and B logically imply $\forall x[A(x) \supset B(x)]$.

knowledge about time proposed by Allen (3) is clearly limited in what kinds of knowledge it can represent. The payoff is that there is a special-purpose algorithm for deciding if knowledge about a collection of temporal intervals is consistent. In a similar vein, Schubert (126, 127) discusses a limited representation for knowledge about parts, where questions about whether or not one object is an ultimate subpart of another can be answered in constant time. Of course, in some sense, the input languages to linear equation solvers, circuit simulators, and program compilers can all be thought of as restricted representation languages that admit very specialized forms of reasoning. Additional domain independent but special-purpose reasoning situations are described in (25).

One final aspect of this approach involves combining more than one special-purpose language in what is called a *hybrid* system. Examples of such systems are the above-mentioned KRYPTON, CAKE (120), and Oppen's co-operating decision procedures (106). The issue in these systems is how to make multiple reasoners co-operate, that is, solve complementary parts of a reasoning problem by exchanging the right sort of information, with a minimum of redundancy.[29] The promise of such systems (which is only starting to be realized) is the possibility of efficiently handling representation languages that are more expressive than those of the subreasoners.

4. MODELS OF LIMITED REASONING

So far, we have considered keeping the reasoning task tractable by limiting the KR language and thus restricting the set of implications to be computed by a KR service. But there are other possibilities. In this section, we consider another approach that leaves the language intact but uses a weaker form of implication.

Technically, there are at least two ways to achieve this. The obvious one is to use (or develop) a nonstandard logic with a different semantics leading to a weaker notion of logical consequence. However, this would force us to give up all of the usual properties of classical logic. Another approach, and one that has received more attention in KR, is to preserve the standard notion of logical consequence, but to augment the language with a knowledge or belief operator, so that the sentence $B\alpha$ is to be read as "the KB believes that α".[30] The idea is to take seriously the concept of belief built into the term "know-

[29]Hybrid systems should be distinguished from those with multiple representation languages that do not attempt to avoid redundancy or to pass information among multiple reasoners, such as the systems described in (1, 5, 26).

[30]Strictly speaking, the term "knowledge" is inappropriate, since what is represented in a KB is typically not required to be true; at the very least, we should be talking about belief. Because of this, and to be consistent with most philosophers, we use the term "belief" throughout.

ledge representation" and to formalize explicitly a weaker sense of belief in terms of the semantics of $B\alpha$. So, instead of basing a KR service on whether or not α is a logical consequence of a set of sentences S, we consider whether or not believing α is a logical consequence of believing the elements of S according to the augmented logic.[31]

4.1 Logical Omniscience

The first formal logic of belief along these lines originated with Hintikka (64) and was based on a *possible-world semantics*.[32] The idea, roughly, is to acknowledge in the semantics that things might have happened differently from the way they did in fact occur, and possible worlds are theoretical entities that index into this space of possibilities. The important point is that within a given interpretation, certain sentences can be true in some possible worlds and false in others. A belief state, then, is characterized by a set of possible worlds, and what is considered to be believed in a belief state is precisely what is true in all of those worlds. So having no opinion on some sentence simply means that it will come out true in some of these possible worlds and false in others.

As is, however, the possible-world model does not formalize a type of belief that is more restricted than the KR service described so far. It suffers from *logical omniscience* (65): At any given point, the set of sentences considered to be believed in this model is closed under logical consequence. It is built into possible-world semantics that if α is believed and α logically implies β, then β is believed as well. A number of attempts have recently been made to formalize a more limited notion of belief. Before looking into them, it should be noted that the main emphasis in this research has been to accurately model the beliefs of an agent, and none of these models have yet been applied in a direct way to the design of new KR architectures.

4.2 The Syntactic Approach

When talking about what an agent actually believes, we want to be able to distinguish between believing only α and $(\alpha \supset \beta)$ on the one hand, and believing α, $(\alpha \supset \beta)$ and β, on the other. While the picture of the world is the same in both cases, only the second involves realizing that β is true. This is somewhat of a problem semantically, since the two sets of beliefs are true in precisely the same possible worlds and so are, in some sense, semantically indistinguishable. Thus, in the possible-worlds approach to belief (as in 77,

[31]Another reason for preferring this approach is that it quite naturally generalizes to deal with beliefs about other beliefs (by using a B operator within the scope of another). This allows facts about self-knowledge to be expressed, which has applications discussed in Section 5.

[32]See (67) for an introduction to possible worlds and the modal logics based on them.

100, 101), it is impossible to distinguish between the two cases. But if the two belief sets are semantically indistinguishable, they are certainly syntactically distinguishable. This might suggest that any realistic model of belief will have to include (something isomorphic to) an actual set of sentences to distinguish between the two belief states above. In other words, part of what will constitute a belief state in this *syntactic* approach to belief are the actual sentences that are believed.

A number of variants of the syntactic model of belief have been proposed (32, 40, 43, 91, 104). Perhaps the most sophisticated one is by Konolige (70–72). In his case, a belief state is characterized by an initial set of sentences (representing a base set of beliefs) and a set of logically sound deduction rules for obtaining new derived beliefs. Logical omniscience is avoided by allowing the deduction rules to be logically incomplete. Among other things, Konolige is able to demonstrate how many variants of the possible-world model can be seen as special cases of his model, where the deduction rules are required to satisfy certain properties. Moreover, Konolige's model allows introspection to be characterized very naturally and concretely using a submodel (with different base beliefs and deduction rules), which is the system's model of itself.

With or without deduction rules, the syntactic approach suffers from a serious defect that is, in some sense, the opposite of the problem with possible worlds. A possible-world semantics is too *coarse-grained* to model belief in that it cannot distinguish belief sets that logically imply the same set of sentences. The syntactic approach, on the other hand, is too *fine-grained* in that it considers any two sets of sentences as distinct semantic entities and, consequently, different belief sets. In general, $B\alpha$ will logically imply $B\beta$ only if α and β are identical. So, for example, a system that believes that $(\alpha \vee \beta)$ is true is not even required to believe that $(\beta \vee \alpha)$ must also be true.

Of course, the syntactic approach can be embellished by requiring that belief sets satisfy certain additional properties, for example, respecting the commutativity of disjunction. The trouble with this kind of stipulation is that it does not show the source of this property, namely that a disjunction is true exactly when one of the disjuncts is. Clearly, it would be preferable to have a model of belief where restrictions on belief sets followed from the definition of belief, that is, a model based in some way on a concept of truth rather than on a collection of ad hoc restrictions to sets of sentences.

4.3 Impossible and Partial Worlds

Recent research in KR has attempted to deal with the problem of logical omniscience by using a generalized notion of possible world. Logical omniscience (and the attendant computational difficulty) is a direct result of the characterization of belief states as sets of possible worlds. If the agent only believes that α is true, the set of worlds will be all those where α is true: some, for example, where β is true, others, where β is false. However, because valid sentences will also be true in all of these possible worlds, the agent is thought of as believing them just as if they were among his actual beliefs. In terms of the possible worlds, there is no way to distinguish α from a valid sentence.

One solution is to make this notion of what an agent thinks the world is like be more relevant to what he actually believes. This can be done by replacing the possible worlds by a different kind of semantic entity that does not necessarily deal with the truth of all sentences. In particular, sentences not relevant to what an agent actually believes (including some valid ones) need not get a truth value in one of these partial possible worlds [which we call *situations,* following (7)]. In fact, we can think of possible worlds as those limiting cases where every sentence has a truth value. Indeed, the concept of a possible world being *compatible* with a situation is intuitively clear: Every sentence whose truth is supported by the situation should be true in that possible world, and every sentence whose falsity is supported should be false. Also, we can allow for situations that have no compatible possible worlds. These are situations that support both the truth and falsity of some sentence. Although they can never be real, such impossible situations can be imagined and are very useful, since they allow an agent to have an incoherent picture of the world.[33]

The "trick," then, that underlies the models of belief is to identify belief with a *set of situations* rather than with a set of possible worlds. This has the following effect (roughly): Not all valid sentences have to be believed, since situations can fail to support them by being partial. Also, because of impossible situations, beliefs need not be closed under logical consequence. For example, a situation can support both α and $(\neg\alpha \vee \beta)$, without supporting β, by supporting both α and its negation.

Although there were certainly antecedents in the philosophical literature (e.g. 114), the first proposal for using this kind of logic in a computer system was by Belnap (8). He sought to use *relevance logic* (4), which embodies these situations (in terms of four truth values), as a means of dealing with the partiality and potential inconsistency of information presented to a machine. The relation of logical consequence in this type of relevance logic (called *tautological entailment*) is a strict subset of the relation of logical consequence in classical logic. Levesque (81) showed how to use these ideas to establish a formal model of belief parallel to the one based on possible worlds. More importantly, Levesque described a KR service based on this weaker notion of implication and proved that the required inference was indeed more

[33]Although they are used here, impossible situations are not strictly necessary to model inconsistent beliefs, as is discussed later.

computationally tractable (for a language without quantifiers) than the one based on classical logic.

As the foundation for a KR service, relevance logic does have serious drawbacks, however. As mentioned above, it is the impossible situations that allow beliefs to be open under logical consequence: Any agent whose belief state is characterized by a set of possible situations always believes β if α and $(\neg\alpha \vee \beta)$ are believed. Intuitively, one would like to say that failing to believe β results from failing to put the other two beliefs together, rather than allowing for the possibility that both α and its negation might be true. The idea is to have a model where believing a conjunction is not the same thing as believing both of the conjuncts; it should require putting the two conjuncts together. This type of model is presented in (43) and illustrates how logical omniscience can be avoided without requiring impossible situations.

Another problem with the taulogical entailment relation of relevance logic is that the obvious first-order version of it is undecidable (110). This means that a computationally manageable model of belief dealing with full first-order logic cannot be derived simply from propositional versions. Three (related) proposals for dealing with full first-order logic are found in (49, 76, 110). Finally, the relevance logic approach by itself gives no indication of how to deal with beliefs about beliefs, for suggestions in this regard see (43).

5. DEFAULTS AND ASSUMPTIONS

As discussed earlier, avoiding incomplete knowledge is an important way to keep the reasoning of a KR service tractable. In fact, in many cases where knowledge is incomplete, a useful strategy is to make assumptions to complete this knowledge, to reason with the enlarged (tractable) belief set, noting what assumptions are in effect, and perhaps later to retract those conclusions derived from assumptions invalidated by new information. For example, given only that Clyde is an elephant, one may assume (until there are reasons to believe otherwise) that Clyde is gray. Any reasoning that then depends on this color would not need to be duplicated for the gray and nongray cases. The hypothesis is that it will be sufficiently advantageous to deal with relatively complete KBs even if it is occasionally necessary to retract assumptions (and conclusions drawn from them) in the face of conflicting evidence. In other words, provided that assumptions are chosen judiciously, there will be real computational advantage to building systems that do not allow for all of the possibilities admitted by the information they possess.

5.1 The Source of Assumptions

This way of completing a KB presumes that there will be principles for selecting assumptions that are most likely to be correct and least likely to require (costly) retraction. Fortunately, this seems to be the case. The most obvious source of assumptions are *defaults,* which come in a number of forms (117):

the closed world assumption (see Section 3.1) Given a collection of facts about some situation, a common assumption is that any relationship that cannot be inferred to hold among the entities being discussed in fact does not hold (116). Among other things, this assumption allows one to avoid having to specify relationships that do not hold.[34] A similar assumption appears in some *is-a* hierarchies in semantic networks (16), where concepts are assumed to be disjoint unless they share descendants. Other related assumptions are *domain closure* (90), where entities whose existence is not implied are assumed not to exist, and the *unique name assumption* (116), where distinct names are taken to represent distinct entities.

the frame assumption Any system that has to reason about the effects of actions must deal with what exactly is not changed by the occurrence of an action. Since any action will tend to have a very localized effect, a common assumption is that nothing is changed by an action except for what is known to be changed. So, for example, moving an object normally does not change its color and, even less likely, the color of other distant objects. This type of assumption was built into some AI planning systems such as STRIPS (46). However, while the problem of circumscribing the effects of actions is one that was recognized very early in AI [see (94) or (115)], it remains to be solved satisfactorily [see (35) and (48) for suggestions as to why].

prototypes Much of what we know about various categories of objects involves properties that almost always apply. So, for example, polar bears are normally white, birds normally fly, and so on. Also, there seems to be fairly clear evidence that people understand the world based very much on notions of normality and abnormality (122). While there have been proposals in the literature for dealing with these probabilistically (53, 55, 144), another possibility is to understand these probabilities as (meta-theoretic) evidence for why these defaults are reasonable assumptions to make. That is, if there is sufficiently high probability that a polar bear will be white, then there is an equally good chance that the default assumption will not have to be revised.

locality assumptions Another powerful source of default assumptions is based on aspects of a "current" context. In communication, one can often fill in details in an elliptical utterance based on using "here," "now," and "me" when the place, time, or person involved is not mentioned. More generally, the context of conversation will introduce entities that can be used as defaults in later discourse (e.g. see 125). Similarly, when planning to achieve an effect, appropriate instruments might default to those readily at hand.

[34]This assumption is certainly used in standard databases where only positive information is stored. This is also related to the "negation as failure" of PROLOG [28, 119].

Overall, because there are so many sources of default assumptions that do indeed seem to work in a majority of cases, reasoning in this manner seems perfectly justifiable. In fact, Reiter claims (117, p. 218) that "Default reasoning may very well be the rule rather than the exception since normally we act in the presence of incomplete knowledge."

Finally, as pointed out in (82), there seem to be cases where assumptions can be made without the benefit of the statistical justification of defaults. If it turns out that the subject of a potential assumption is not likely to be the subject of later information, it may still be worth making some assumption for computational reasons, as it will be unlikely later to give rise to a conflict. So, for example, given information that a dog chased a cat around a tree, one might assume that the direction of the chase was clockwise, even though statistically, it could go either way.[35] Of course, the real trick here will be to find principles for making assumptions that take into account not only possible computational gains but also the likelihood of errors and their severity.

5.2 Truth Maintenance

One branch of KR research has focussed on appropriate mechanisms for managing families of assumptions and keeping track of justifications to allow a graceful backing out. These have been called *truth maintenance systems* (or sometimes *reason* maintenance systems) (36, 37, 88). Of particular interest in this area are those systems that are explicit about the assumptions being carried or being excluded, such as (33) and (87).[36] Most of this work, however, deals with efficient mechanisms for actually implementing truth maintenance. This in turn derives from earlier work on *procedural representations* of knowledge, where a major focus of attention was the control of reasoning, especially during the kind of backtracking that arises as a result of defeated assumptions. This procedural school began more or less with the PLANNER representation language (62, 63) and received perhaps a last push in the AMORD system (34). A general discussion of the issues involved in this approach to representation can be found in (57, 102, 138). Ultimately, the problem of truth maintenance is that of *belief revision* and theory evolution: what beliefs should be discarded in the face of conflicting evidence [e.g. see (2) and also (38) for a survey of work in this area].

[35]Indeed, in "visualizing" a described situation, we are filling in a large number of visually significant details that need not be implied by anything in the original description nor by statistically relevant defaults. So, for example, it is hard to think about Ronald Reagan standing beside Margaret Thatcher without thinking of them from a specific point of view with one of them on the left and the other on the right.

[36]Interestingly enough, Shapiro and Martins argue that relevance logic is the appropriate framework for maintaining a network of assumptions, quite independently of its role in limiting reasoning (as discussed in Section 4.3).

5.3 Nonmonotonic Reasoning

The major conceptual issue faced by KR systems that must reason with assumptions is what kinds of assumptions there are and what is the appropriate logic for reasoning with them. This has led to the development of a branch of AI called *nonmonotonic reasoning*. The idea, roughly, is this. Ordinary reasoning in a KBS is monotonic: The set of conclusions that can be drawn from a body of information grows as the body of information grows (since anything logically implied by a set of sentences is also implied by a superset of that set). But when the body of information in question also contains potential assumptions to be made (barring information to the contrary), the set of conclusions need not grow monotonically, since new information may very well invalidate previous assumptions.

The area of formal nonmonotonic reasoning has recently become a very active and fruitful one. Although no KR systems have been designed to incorporate the general principles, and very little is currently known about the impact of nonmonotonic reasoning on the tractability of the reasoning problem, a number of formalisms capturing different aspects of the problem have been proposed and analyzed.

An early collection of work in this area appears in (9). In this volume, three formalisms for nonmonotonic reasoning are proposed. All of them are based on modifications to classical first-order logics, but unfortunately none of them enjoy the benefit of a semantic account. The first proposal, by Reiter (118), is called *default logic*. Theories in this logic consist of two parts: a base set of standard first-order sentences, and a set of default rules specified by triples of formulas. The proof-theoretic understanding of a default rule is that given (an instance of) the first formula, the third formula can be inferred provided that the second one has not been. An *extension* of a default theory is any consistent minimal set of sentences containing the base set, which is both deductively closed and closed under the default rules. Reiter's interpretation is that any such extension constitutes a reasonable set of beliefs given the base set and defaults, and he shows how certain default theories may have none or more than one such extension. The nonmonotonicity of default logic arises from the fact that an extended base set may prohibit the application of certain default rules.

The second nonmonotonic formalism, called *nonmonotonic logic*, was proposed by McDermott & Doyle (96). In this case, a standard first-order language is augmented with a unary sentential operator M, where Mα is to be read as "α is consistent." Their version of an extension of a base set of sentences is any minimal superset that is deductively closed and has the property that Mα is an element of the extension whenever $\neg\alpha$ is not an element. Unlike Reiter, the authors identify belief with the set of sentences that are members of all such extensions, even though such a belief set need not itself be an extension. The nonmonotonicity arises here from the fact that as the base set grows, fewer Mα sentences will be added to an extension.

The third formalism is called *circumscription* and is due to John McCarthy (92). The idea is to take any finite set of first-order sentences as the base and "circumscribe" a predicate in it by adding to the base an infinite collection of sentences (as specified by a circumscription schema). The intent of these sentences is to make the extended theory state that the predicate is as small as possible, given the base set. The way this is done for a predicate P is to include in the collection a sentence for each open first-order formula ϕ with the same number of arguments as P. What this sentence says is that if the base sentences with P replaced by ϕ are true and ϕ is a subset of P, then ϕ and P are equal. The net effect is to prohibit any property (specifiable by an open formula) from simultaneously satisfying the base sentences and being a proper subset of the circumscribed predicate. In other words, the extended theory makes the circumscribed predicate minimal. This is nonmonotonic, because any change to the base set (additive or not) will lead to a different extended theory, since every sentence in that theory uses the base set directly.

Since the publication of these three formalisms, there has been extensive research on their properties. It is generally agreed that the first two have serious drawbacks. In Reiter's model, domain-dependent knowledge has to be encoded in inference rules, more or less ruling out a semantic account for the logic (103); McDermott & Doyle's proposal appears to be based on a notion of consistency that is inappropriate for the job but is best understood in terms of a logic of knowledge or belief (103). Circumscription remains the most successful of the three and certainly the one that has received the most attention (e.g. 41, 84, 119). It does have expressive drawbacks, however, and new versions of it are under active investigation (e.g. 93, 98).

Gaining in popularity is an attempt to understand nonmonotonicity in terms of explicit knowledge and belief. Thus, in (56, 69, 77, 103) a general form of nonmonotonicity is explained in terms of the inherent nonmonotonicity of self-knowledge. The idea is that certain kinds of assumptions are based on a lack of other beliefs (31). For example, one might be willing to believe that there is no city in New Zealand larger than Los Angeles based on a belief that one would know if there were such a city. So belief in the lack of a certain belief is sufficient to make the assumption. But this is a nonmonotonic process, since beliefs about what one does not believe must surely be revised as new beliefs are acquired. On the other hand, it is not clear that all nonmonotonic reasoning behaves this way. For example, the fact that birds generally fly seems to have nothing to do with what is or is not believed at any given point.

At this stage of research, the computational relevance of this work remains somewhat questionable. For the three formalisms described above, it appears (counter-intuitively) that reasoning with assumptions and defaults is even more difficult than reasoning without them.[37] From a computational standpoint, the formalizations seem to be moving in the wrong direction. Not that systems have been unable to take computational advantage of defaults. As discussed in Section 3.3, the use of defaults goes back to Quillian and the very early semantic networks (112), and it remains a major feature of inheritance hierarchies (16, 44, 132, 133). Defaults were also a major part of the frame concept (99) as can be seen in representation languages like KRL (11). But it has often been much easier to construct systems that reason in certain (nonmonotonic) ways than to justify the correctness of that reasoning. Indeed, systems whose behavior seemed appropriate at first glance were later shown to exhibit reasoning anomalies (42, 45). So it remains to be seen how the computational promise of nonmonotonic reasoning can be correctly realized.

6. CONCLUSION

According to the Knowledge Representation hypothesis, intelligent behavior ultimately depends on explicitly represented knowledge. No other design strategy as yet seems plausible to explain behavior, or have it depend in a flexible way on what one is told, or to isolate the assumptions that govern it, or to achieve any number of other desirable traits. But KR by itself does not solve anything unless a system is able to reason effectively with what it has explicitly represented. Specifically, if reasoning needs to be performed automatically as part of a larger task, it must be dependable both in terms of what it calculates and how long it takes. Thus, KR, as it has been presented here, is the study of what information can be extracted, in a computationally dependable way, from what forms of represented knowledge. In other words, it investigates the area within the confines of the KR hypothesis. In some sense, theorem proving with classical first-order logic is the base camp for this investigation, but as this review attempts to show, there is much to be learned from excursions into nearby regions.

[37]In fact, in the first two cases, applying an assumption depends on that assumption being consistent with some theory of the world. But for sufficiently expressive languages like that of FOL, the consistent sentences are not even recursively enumerable.

Literature Cited

1. Aikins, J. 1983. Prototypical knowledge for expert systems. *Artif. Intell.* 20(2): 163–210
2. Alchouron, C., Gardenfors, P., Makinson, D. 1985. On the logic of theory change: partial meet contraction and revision functions. *J. Symbol. Logic* 50(2):510–30
3. Allen, J. 1983. Maintaining knowledge about temporal intervals. *Commun. ACM* 26(11):832–43
4. Anderson, A., Belnap, N. 1975. *Entailment: The Logic of Relevance and Necessity*. Princeton, NJ: Princeton University Press
5. Attardi, G., Simi, M. 1981. Consistency and completeness of OMEGA, a logic for knowledge representation. In *Proc. Int. Jt. Conf. Artif. Intell., Vancouver, BC,* pp. 504–10
6. Barr, A., Davidson, J. 1981. Representation of knowledge. In *The Handbook of Artificial Intelligence*, ed. A. Barr, E. Feigenbaum, pp. 141–222. Los Altos, Calif: W. Kaufmann
7. Barwise, J., Perry, J. 1983. *Situations and Attitudes*. Cambridge, Mass: Bradford Books, MIT Press
8. Belnap, N. 1977. A useful four-valued logic. In *Modern Uses of Multiple-Valued Logic*, ed. J. Dunn, G. Epstein, pp. 8–37. The Netherlands: Reidel
9. Bobrow, D. 1980. *Special Volume on Non-Monotonic Reasoning*. *Artif. Intell.* 13(1, 2)
10. Bobrow, D. 1984. *Special Volume on Qualitative Reasoning about Physical Systems*. *Artif. Intell.* 24(1–3)

10a. Bobrow, D., Collins, A., eds. 1975. *Representation and Understanding: Studies in Cognitive Science*. New York: Academic

11. Bobrow, D., Winograd, T. 1977. An overview of KRL, a knowledge representation language. *Cognitive Sci.* 1(1):3–46
12. Bobrow, D., Winograd, T. 1979 KRL: another perspective. *Cognitive Sci.* 3(1):29–42
13. Bobrow, D., Winograd, T., the KRL Research Group. 1977. Experience with KRL-0: one cycle of a knowledge representation language. In *Proc. Int. Jt. Conf. Artif. Intell. Cambridge, Mass.*, pp. 213–22
14. Brachman, R. 1978. A structural paradigm for representing knowledge. *BBN Rep. 3605, Bolt Beranek & Newman, Cambridge, Mass.*
15. Brachman, R. 1979. On the epistemological status of semantic networks. See Ref. 47, pp. 3–50
16. Brachman, R. 1983. What is-a is and isn't: an analysis of taxonomic links in semantic networks. *IEEE Comput.* 16(10):30–36
17. Brachman, R. 1985. I lied about the trees (or, defaults and definitions in knowledge representation). *AI Mag.* 6(3):80–93
18. Brachman, R., Fikes, R., Levesque, H. 1983. KRYPTON: a functional approach to knowledge representation. *IEEE Comput.* 16(10):67–73
19. Brachman, R., Gilbert, V., Levesque, H. 1985. An essential hybrid reasoning system: knowledge and symbol level accounts of KRYPTON. In *Proc. Int. Jt. Conf. Artif. Intell., Los Angeles, Calif.*
20. Brachman, R., Levesque, H. 1982. Competence in knowledge representation. In *Proc. Natl. Conf. Am. Assoc. Artif. Intell., Pittsburgh, Pa.*, pp. 189–92
21. Brachman, R., Levesque, H. 1984. The tractability of subsumption in frame-based description languages. In *Proc. Natl. Conf. Am. Assoc. Artif. Intell., Austin, Texas,* pp. 34–37
22. Brachman, R., Levesque, H., eds. 1985. *Readings in Knowledge Representation*. Los Altos, Calif: Morgan Kaufmann
23. Brachman, R., Schmolze, J. 1985. An overview of the KL-ONE knowledge representation system. *Cognitive Sci.* 9(2):171–216
24. Deleted in proof
25. Bundy, A., Byrd, L., Mellish, C. 1985. Special-purpose, but domain-independent, inference mechanisms. In *Progress in Artificial Intelligence,* ed. L. Steels, J. Campbell, pp. 93–111. London: Ellis Horwood
26. Charniak, E. 1981. A common representation for problem-solving and language-comprehension information. *Artif. Intell.* 16(3):225–55
27. Cherniak, C. 1984. Computational complexity and the universal acceptance of logic. *J. Philos.*, 81(12):739–58
28. Clark, K. 1978. Negation as failure. See Ref. 51a, pp. 293–322
29. Clocksin, W., Mellish, C. 1981. *Programming in* PROLOG. New York: Springer-Verlag
30. Collins, A., Loftus, E. 1975. A spreading-activation theory of semantic

processing. *Psychol. Rev.* 82(6):407–28
31. Collins, A., Warnock, E., Nelleke, A., Miller, M. 1975. Reasoning from incomplete knowledge. See Ref. 10a, pp. 383–416
32. Creary, L. 1979. Propositional attitudes: Fregean representation and simulative reasoning. In *Proc. Int. Jt. Conf. Artif. Intell., Tokyo,* pp. 176–81
33. de Kleer, J. 1986. An assumption-based TMS. *Artif. Intell.* 28(2):127–62
34. de Kleer, J., Doyle, J., Steele, G., Sussman, G., 1977. AMORD: explicit control of reasoning. In *Symp. Artif. Intell. Program. Lang., Rochester, NY,* pp. 116–25
35. Dennett, D. 1986. Cognitive wheels: the frame problem of artificial intelligence. In *Minds, Machines, and Evolution,* ed. C. Hookway. Cambridge, England: Cambridge Univ. Press
36. Doyle, J. 1982. A glimpse of truth-maintenance. In *Artificial Intelligence: An MIT Perspective,* ed. P. Winston, R. Brown, pp. 119–35. Cambridge, Mass: MIT Press
37. Doyle, J. 1983. The ins and outs of reason maintenance. In *Proc. Int. Jt. Conf. Artif. Intell. Karlsruhe, FRG,* pp. 349–51
38. Doyle, J., London, P. 1980. A selected descriptor-based bibliography to the literature on belief revision. *SIGART Newsl.* 71:7–23
39. Dreyfus, H. 1981. From micro-worlds to knowledge representation: AI at an impasse. In *Mind Design,* ed. J. Haugeland, pp. 161–204. Cambridge, Mass: MIT Press
40. Eberle, R. 1974. A logic of believing, knowing and inferring. *Synthese* 26: 356–82
41. Etherington, D., Mercer, R., Reiter, R. 1984. On the adequacy of predicate circumscription for closed-world reasoning. In *The Non-Monotonic Reasoning Workshop, New Paltz, NY,* pp. 70–81
42. Etherington, D., Reiter, R. 1983. On inheritance hierarchies with exceptions. In *Proc. Natl. Conf. Am. Assoc. Artif. Intell., Washington DC,* pp. 104–8
43. Fagin, R., Halpern, J. 1985. Belief, awareness and limited reasoning. In *Proc. Int. Jt. Conf. Artif. Intell., Los Angeles, Calif.,* pp. 491–501
44. Fahlman, S. 1979. *NETL: A System for Representing and Using Real-World Knowledge.* Cambridge, Mass: MIT Press
45. Fahlman, S., Touretzky, D., van Roggen, W. 1981. Cancellation in a parallel semantic network. In *Proc. Int. Jt. Conf. Artif. Intell., Vancouver, BC,* pp. 257–63
46. Fikes, R., Nilsson, N. 1971. STRIPS: a new approach to the application of theorem proving to problem solving. *Artif. Intell.* 2(3–4):189–208
47. Findler, N., ed. 1979. *Associative Networks: Representation and Use of Knowledge by Computers.* New York: Academic
48. Fodor, J. 1983. *The Modularity of Mind.* Cambridge, Mass: Bradford Books, MIT Press
49. Frisch, A. 1985. Using model theory to specify AI programs. In *Proc. Int. Jt. Conf. Artif. Intell., Los Angeles,* pp. 148–54
50. Frisch, A., Allen, J. 1982. Knowledge representation and retrieval for natural language processing. *Tech. Rep. TR 104, Dep. Comput. Sci., Univ. Rochester, NY*
51. Funt, B. 1980. Problem-solving with diagrammatic representations. *Artif. Intell.* 13(3):201–30
51a. Gallaire, H., Minker, J., eds. 1978. *Logic and Databases.* New York: Plenum
52. Garey, M., Johnson, D. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* San Francisco: W. H. Freeman
53. Garvey, T., Lowrance, J., Fischler, M. 1981. An inference technique for integrating knowledge from disparate sources. In *Proc. Int. Jt. Conf. Artif. Intell., Vancouver, BC,* pp. 319–25
54. Genesereth, M. 1983. An overview of meta-level architecture. In *Proc. Natl. Conf. Am. Assoc. Artif. Intell., Washington, DC,* pp. 119–24
55. Ginsberg, M. 1984. Non-monotonic reasoning using Dempster's rule. In *Proc. Natl. Conf. Am. Assoc. Artif. Intell., Austin, Texas,* pp. 126–29
56. Halpern, J., Moses, Y. 1984. Towards a theory of knowledge and ignorance: preliminary report. In *The Non-Monotonic Reasoning Workshop, New Paltz, NY,* pp. 125–43
57. Hayes, P. 1974. Some problems and non-problems in representation theory. In *AISB Summer Conf. Univ. Sussex,* pp. 63–79
58. Hayes, P. 1977. In defence of logic. In *Proc. Int. Jt. Conf. Artif. Intell., Cambridge, Mass.,* pp. 559–65
59. Hayes, P. 1979. The logic of frames. In *Frame Conceptions and Text Understanding,* ed. D. Metzing, pp. 46–61. Berlin: de Gruyter

60. Hayes, P. 1985. Naive physics I: ontology for liquids. See Hobbs & Moore 1985, pp. 71–107
61. Hayes, P. 1985. The second naive physics manifesto. See Hobbs & Moore 1985, pp. 1–36
62. Hewitt, C. 1969. PLANNER: a language for proving theorems in robots. In *Proc. Int. Jt. Conf. Artif. Intell., Washington, DC*, pp. 295–301
63. Hewitt, C. 1972. Description and theoretical analysis (using schemata) of PLANNER, a language for proving theorems and manipulating models in a robot. *Tech. Rep. TR-258, AI Lab., MIT, Cambridge, Mass.*
64. Hintikka, J. 1962. *Knowledge and Belief: An Introduction to the Logic of the Two Notions*. Ithaca, NY: Cornell Univ. Press
65. Hintikka, J. 1975. Impossible possible worlds vindicated. *J. Philos.* 4:475–84
66. Hobbs, J., Moore, R., eds. 1985. *Formal Theories of the Commonsense World*. Norwood, NJ: Ablex
67. Hughes, G., Cresswell, M. 1968. *An Introduction to Modal Logic*. London: Methuen
68. Israel, D. 1983. The role of logic in knowledge representation. *IEEE Comput.* 16(10):37–42
69. Konolige, K. 1982. Circumscriptive ignorance. In *Proc. Natl. Conf. Am. Assoc. Artif. Intell., Pittsburgh, Pa.*, pp. 202–4
70. Konolige, K. 1983. A deductive model of belief. In *Proc. Int. Jt. Conf. Artif. Intell., Karlsruhe, FRG*, pp. 377–81
71. Konolige, K. 1984. *A deduction model of belief and its logics*. PhD thesis. Dep. Comput. Sci., Stanford Univ., Palo Alto, Calif.
72. Konolige, K. 1985. A computational theory of belief introspection. In *Proc. Int. Jt. Conf. Artif. Intell., Los Angeles*, pp. 502–8
73. Kowalski, R. 1974. Predicate logic as a programming language. In *IFIP Congress, Stockholm*, pp. 569–74
74. Kowalski, R. 1979. *Logic for Problem Solving*. Amsterdam: Elsevier North-Holland
75. Kuipers, B. 1979. On representing commonsense knowledge. See Ref. 47, pp. 393–408
76. Lakemeyer, G. 1986. Steps towards a first order logic of implicit and explicit belief. In *Theoretical Aspects of Reasoning about Knowledge: Proc. 1986 Conf.*, ed. J. Halpern, pp. 325–40. Los Altos, Calif: Morgan Kaufmann
77. Levesque, H. 1981. *A formal treatment of incomplete knowledge bases*. PhD thesis, Dep. Comput. Sci., Univ. Toronto, Ontario
78. Levesque, H. 1983. The logic of incomplete knowledge bases. See Ref. 105, pp. 165–86
79. Levesque, H. 1984. Foundations of a functional approach to knowledge representation. *Artif. Intell.* 23(2):155–212
80. Levesque, H. 1984. A fundamental tradeoff in knowledge representation and reasoning. In *Proc. Bienn. Conf. Can. Soc. Comput. Stud. Intell., London, Ontario*, pp. 141–52
81. Levesque, H. 1984. A logic of implicit and explicit belief. In *Proc. Natl. Conf. Am. Assoc. Artif. Intell., Austin, Texas*, pp. 198–202
82. Levesque, H. 1986. Making believers out of computers. *Artif. Intell.* In press
83. Lewis, H. 1978. Complexity of solvable cases of the decision problem for the predicate calculus. In *Proc. 19th IEEE Symp. Found. Comput. Sci.*, pp. 35–47
84. Lipschitz, V. 1985. Closed-world databases and circumscription. *Artif. Intell.*, 27(2):229–236
85. Loganantharaj, R. 1985. *Theoretical and implementational aspects of parallel link resolution in connection graphs*. PhD thesis. Dep. Comput. Sci., Colo. State Univ., Fort Collins
86. Loveland, D. 1978. *Automated Theorem Proving: A Logical Basis*. New York: North-Holland
87. Martins, J., Shapiro, S. 1983. Reasoning in multiple belief spaces. In *Proc. Int. Jt. Conf. Artif. Intell., Karlsruhe, FRG*, pp. 370–73
88. McAllester, D. 1980. *The Use of Equality in Deduction and Knowledge Representation*. MS thesis. AI Lab., Mass. Inst. Technol., Cambridge
89. McCarthy, J. 1968. Programs with common sense. See Ref. 98a, pp. 403–18
90. McCarthy, J. 1977. Epistemological problems in artificial intelligence. In *Proc. Int. Jt. Conf. Artif. Intell., Cambridge, Mass.*, pp. 1038–44
91. McCarthy, J. 1979. First order theories of individual concepts and propositions. In *Machine Intelligence*, ed. J. Hayes, D. Michie, L. Mikulich, 9:129–47. Chichester, England: Ellis Horwood
92. McCarthy, J. 1980. Circumscription—a form of non-monotonic reasoning. *Artif. Intell.* 13(1,2):27–39
93. McCarthy, J. 1984. Applications of circumscription to formalizing commonsense knowledge. In *The Non-Monotonic Reasoning Workshop, New Paltz, NY*, pp. 295–324

94. McCarthy, J., Hayes, P. 1969. Some philosophical problems from the standpoint of artificial intelligence. In *Machine Intelligence,* ed. B. Meltzer, D. Michie, 4:463–502. Edinburgh: Edinburgh Univ. Press
95. McDermott, D. 1982. A temporal logic for reasoning about processes and plans. *Cognitive Sci.* 6(2):101–55
96. McDermott, D., Doyle, J. 1980. Nonmonotonic logic I. *Artif. Intell.* 13(1,2): 41–72
97. Mendelson, E. 1964. *Introduction to Mathematical Logic.* New York: Van Nostrand Reinhold
98. Minker, J., Perlis, D. 1984. Protected circumscription. In *Workshop on Non-Monotonic Reasoning, New Paltz, NY,* pp. 337–43
98a. Minsky, M., ed. 1968. *Semantic Information Processing.* Cambridge, Mass: MIT Press
99. Minsky, M. 1981. A framework for representing knowledge. In *Mind Design,* ed. J. Haugeland, pp. 95–128. Cambridge, Mass: MIT Press
100. Moore, R. 1977. Reasoning about knowledge and action. In *Proc. Int. Jt. Conf. Artif. Intell., Cambridge, Mass.,* pp. 223–27
101. Moore, R. 1980. Reasoning about knowledge and action. *Tech. Note 191, Artifi. Intell. Cent., SRI Int., Menlo Park, Calif.*
102. Moore, R. 1982. The role of logic in knowledge representation and commonsense reasoning. In *Proc. Natl. Conf. Am. Assoc. Artif. Intell., Pittsburgh, Pa.,* pp. 428–33
103. Moore, R. 1983. Semantical considerations on nonmonotonic logic. In *Proc. Int. Jt. Conf. Artif. Intell., Karlsruhe, FRG,* pp. 272–79
104. Moore, R., Hendrix, G. 1979. Computational models of belief and the semantics of belief sentences. *Tech. Note 187, Artifi. Intell. Cent., SRI Int., Menlo Park, Calif.*
105. Mylopoulos, J., Levesque, H. 1983. An overview of knowledge representation. In *On Conceptual Modelling: Perspectives from Artificial Intelligence, Databases, and Programming Languages,* ed. M. Brodie, J. Mylopoulos, J. Schmidt, pp. 3–17. New York: Springer-Verlag
106. Nelson, G., Oppen, D. 1979. Simplification by cooperating decision procedures. *ACM Trans. Program. Lang. Sys.* 1(2):245–57
107. Nilsson, N. 1980. *Principles of Artificial Intelligence.* Palo Alto, Calif: Tioga
108. Norman, D., Bobrow, D. 1975. On data limited and resource limited processing. *Cognitive Psychol.* 7:44–64
109. Norman, D., Rumelhart, D., eds. 1975. *Explorations in Cognition.* San Francisco: W. H. Freeman
110. Patel-Schneider, P. 1985. A decidable first-order logic for knowledge representation. In *Proc. Int. Jt. Conf. Artif. Intell., Los Angeles,* pp. 455–58
111. Pentland, A., Fischler, M. 1983. A more rational view of logic. *AI Mag.* 4(4):15–18
112. Quillian, M. 1967. Word concepts: a theory and simulation of some basic semantic capabilities. *Behav. Sci.* 12: 410–30
113. Quillian, M. 1968. Semantic memory. See Ref. 98a, pp. 227–70
114. Rantala, V. 1982. Impossible world semantics and logical omniscience. *Acta Philos. Fenni.* 35:106–15
115. Raphael, B. 1971. The frame problem in problem solving systems. In *Artificial Intelligence and Heuristic Programming.* New York: American Elsevier
116. Reiter, R. 1978. On closed world databases. See Ref. 51a, pp. 55–76
117. Reiter, R. 1978. On reasoning by default. In *Proc. Conf. Theor. Issues Nat. Lang. Process, Univ. Ill. Urbana-Champaign*
118. Reiter, R. 1980. A logic for default reasoning. *Artif. Intell.* 13(1,2):81–132
119. Reiter, R. 1982. Circumscription implies predicate completion (sometimes). In *Proc. Natl. Conf. Am. Assoc. Artif. Intell., Pittsburgh, Pa.,* pp. 418–20
120. Rich, C. 1980. Knowledge representation languages and predicate calculus: how to have your cake and eat it too. In *Proc. Natl. Conf. Am. Assoc. Artif. Intell., Pittsburgh, Pa.,* pp. 193–96
121. Rieger, C. 1976. An organization of knowledge for problem solving and language comprehension. *Artif. Intell.* 7(2):89–127
122. Rosch, E., Mervis, C. 1975. Family resemblances: studies in the internal structure of categories. *Cognitive Psychol.* 7:573–605
123. Rosenschein, S. 1986. Formal theories of knowledge in AI and robotics. *New Generation Comput.* 3(4): In press
124. Schank, R. 1973. Identification of conceptualizations underlying natural language. In *Computer Models of Thought and Language,* ed. R. Schank, K. Colby, pp. 187–247. San Francisco: W. H. Freeman
125. Schank, R. 1975. *Conceptual Informa-*

tion Processing. Amsterdam: North-Holland

126. Schubert, L. 1979. Problems with parts. In *Proc. Int. Jt. Conf. Artif. Intell., Tokyo*, pp. 778–84
127. Schubert, L., Papalaskaris, M., Taugher, J. 1983. Determining type, part, color, and time relationships. *IEEE Comput*. 16(10):53–60
128. Simmons, R. 1973. Semantic networks: their computation and use for understanding English sentences. See Ref. 124, pp. 63–113
129. Sloman, A. 1971. Interactions between philosophy and artificial intelligence: the role of intuition and non-logical reasoning in intelligence. *Artif. Intell*. 2:209–25
130. Sloman, A. 1975. Afterthoughts on analogical representation. In *Proc. Conf. Theoretical Issues in Natural Language Processing, Cambridge, Mass*, pp. 164–68
131. Smith, B. 1982. Reflection and semantics in a procedural language. *Tech. Rep. MIT/LCS/TR-272, Mass. Inst. Technol., Cambridge*

131a. 1980. *Special Issue on Knowledge Representation. SIGART Newsl*., Vol. 70

132. Touretzky, D. 1984. Implicit ordering of defaults in inheritance systems. In *Proc. Natl. Conf. Am. Assoc. Artif. Intell., Austin, Texas*, pp. 322–25
133. Touretzky, D. 1984. *The mathematics of inheritance systems*. PhD thesis. Dep. Comput. Sci., Carnegie-Mellon Univ., Pittsburgh, Pa.
134. Vilain, M. 1982. A system for reasoning about time. In *Proc. Natl. Conf. Am. Assoc. Artif. Intell., Pittsburgh, Pa.*, pp. 197–201
135. Waterman, D., Hayes-Roth, F., ed. 1978. *Pattern-Directed Inference Systems*. New York: Academic
136. Winker, S. 1982. Generation and verification of finite models and counterexamples using an automated theorem prover answering two open questions. *J. ACM* 29(2):273–84
137. Winograd, T. 1972. *Understanding Natural Language*. New York: Academic
138. Winograd, T. 1975. Frame representations and the declarative/procedural controversy. See Ref. 10a, pp. 185–210
139. Winograd, T. 1980. Extended inference modes in reasoning by computer systems. *Artif. Intell*. 13(1–2):5–26
140. Winston, P. 1975. Learning structural descriptions from examples. In *The Psychology of Computer Vision*, ed. P. Winston, pp. 157–209. New York: McGraw-Hill
141. Woods, W. 1975. What's in a link: foundations for semantic networks. See Ref. 10a, pp. 35–82
142. Wos, L., Pereira, F., Hong, R., Boyer, R., Moore, J., et al. 1985. An overview of automated reasoning and related fields. *J. Autom. Reason*. 1(1):5–48
143. Wos, L., Winker, S., Smith, B., Veroff, R., Henschen, L. 1984. A new use of an automated reasoning assistant: open questions in equivalential calculus and the study of infinite domains. *Artif. Intell*. 22(3):303–56
144. Zadeh, L. 1983. Commonsense knowledge representation based on fuzzy logic. *IEEE Comput*. 16(10):61–66