ARTIFICIAL INTELLIGENCE, LOGIC AND FORMALIZING COMMON SENSE

John McCarthy
Computer Science Department
Stanford University
Stanford, CA 94305
jmc@cs.stanford.edu
http://www-formal.stanford.edu/jmc/

1990

1 Introduction

This is a position paper about the relations among artificial intelligence (AI), mathematical logic and the formalization of common-sense knowledge and reasoning. It also treats other problems of concern to both AI and philosophy. I thank the editor for inviting it. The position advocated is that philosophy can contribute to AI if it treats some of its traditional subject matter in more detail and that this will advance the philosophical goals also. Actual formalisms (mostly first order languages) for expressing common-sense facts are described in the references.

Common-sense knowledge includes the basic facts about events (including actions) and their effects, facts about knowledge and how it is obtained, facts about beliefs and desires. It also includes the basic facts about material objects and their properties.

One path to human-level AI uses mathematical logic to formalize common-sense knowledge in such a way that common-sense problems can be solved by logical reasoning. This methodology requires understanding the common-sense world well enough to formalize facts about it and ways of achieving goals in it. Basing AI on understanding the common-sense world is different from basing it on understanding human psychology or neurophysiology. This approach to AI, based on logic and computer science, is complementary to approaches that start from the fact that humans exhibit intelligence, and that explore human psychology or human neurophysiology.

This article discusses the problems and difficulties, the results so far, and some improvements in logic and logical languages that may be required to formalize common sense. Fundamental conceptual advances are almost certainly required. The object of the paper is to get more help for AI from philosophical logicians. Some of the requested help will be mostly philosophical and some will be logical. Likewise the concrete AI approach may fertilize philosophical logic as physics has repeatedly fertilized mathematics.

There are three reasons for AI to emphasize common-sense knowledge rather than the knowledge contained in scientific theories.

(1) Scientific theories represent compartmentalized knowledge. In presenting a scientific theory, as well as in developing it, there is a common-sense pre-scientific stage. In this stage, it is decided or just taken for granted what phenomena are to be covered and what is the relation between certain formal terms of the theory and the common-sense world. Thus in classical mechanics it is decided what kinds of bodies and forces are to be used before the differential equations are written down. In probabilistic theories, the sample space is determined. In theories expressed in first order logic, the predicate and function symbols are decided upon. The axiomatic reasoning techniques used in mathematical and logical theories depend on this having been done. However, a robot or computer program with human-level intelligence will have to do this for itself. To use science, common sense is required.

Once developed, a scientific theory remains imbedded in common sense. To apply the theory to a specific problem, common-sense descriptions must be matched to the terms of the theory. For example, $d = \frac{1}{2}gt^2$ does not in itself identify d as the distance a body falls in time t and identify g as the acceleration due to gravity. (McCarthy and Hayes 1969) uses the *situation calculus* discussed in that paper to imbed the above formula in a formula describing the common-sense situation, for example

$$\begin{array}{l} dropped(x,s) \wedge height(x,s) = h \wedge d = \frac{1}{2}gt^2 \wedge d < h \\ \qquad \supset \\ \exists s'(F(s,s') \wedge time(s') = time(s) + t \ \wedge \ height(x,s') = h - d). \end{array} \tag{1}$$

Here x is the falling body, and we are presuming a language in which the functions *height*, *time*, etc. are formalized in a way that corresponds to what the English words suggest. s and s' denote *situations* as discussed in that paper, and $F(s, s')$ asserts that the situation s' is in the future of the situation s.

(2) Common-sense reasoning is required for solving problems in the common-sense world. From the problem solving or goal-achieving point of view, the common-sense world is characterized by a different *informatic situation* than that *within* any formal scientific theory. In the typical common-sense informatic situation, the reasoner doesn't know what facts are relevant to solving his problem. Unanticipated obstacles may arise that involve using parts of his knowledge not previously thought to be relevant.

(3) Finally, the informal metatheory of any scientific theory has a common-sense informatic character. By this I mean the thinking about the structure of the theory in general and the research problems it presents. Mathematicians invented the concept of a group in order to make previously vague parallels between different domains into a precise notion. The thinking about how to do this had a common-sense character.

It might be supposed that the common-sense world would admit a conventional scientific theory, e.g. a probabilistic theory. But no one has yet developed such a theory, and AI has taken a somewhat different course that involves nonmonotonic extensions to the kind of reasoning used in formal scientific theories. This seems likely to work better.

Aristotle, Leibniz, Boole and Frege all included common-sense knowledge when they discussed formal logic. However, formalizing much of common-sense knowledge and reasoning proved elusive, and the twentieth century emphasis has been on formalizing mathematics. Some important philosophers, e.g. Wittgenstein, have claimed that common-sense knowledge is unformalizable or mathematical logic is inappropriate for doing it. Though it is possible to give a kind of plausibility to views of this sort, it is much less easy to make a case for them that is well supported and carefully worked out. If a common-sense reasoning problem is well presented, one is well on the way to formalizing it. The examples that are presented for this negative view borrow much of their plausibility from the inadequacy of the specific collections of predicates and functions they take into consideration. Some of their force comes from not formalizing nonmonotonic reasoning, and some may be due to lack of logical tools still to be discovered. While I acknowledge this opinion, I haven't the time or the scholarship to deal with the full range of such arguments. Instead I will present the positive case, the problems that have arisen, what has been done and the problems that can be foreseen. These problems are often more interesting than the ones suggested by philosophers trying to show the futility of formalizing common sense, and they suggest productive research programs for both AI and philosophy.

In so far as the arguments against the formalizability of common-sense attempt to make precise intuitions of their authors, they can be helpful in identifying problems that have to be solved. For example, Hubert Dreyfus (1972) said that computers couldn't have "ambiguity tolerance" but didn't offer much explanation of the concept. With the development of nonmonotonic reasoning, it became possible to define some forms of *ambiguity tolerance* and show how they can and must be incorporated in computer systems. For example, it is possible to make a system that doesn't know about possible *de re/de dicto* ambiguities and has a default assumption that amounts to saying that a reference holds both *de re* and *de dicto*. When this assumption leads to inconsistency, the ambiguity can be discovered and treated, usually by splitting a concept into two or more.

If a computer is to store facts about the world and reason with them, it needs a precise language, and the program has to embody a precise idea of what reasoning is allowed, i.e. of how new formulas may be derived from old. Therefore, it was natural to try to use mathematical logical languages to express what an intelligent computer program knows that is relevant to the problems we want it to solve and to make the program use logical inference in order to decide what to do. (McCarthy 1959) contains the first proposals to use logic in AI for expressing what a program knows and how it should reason. (Proving logical formulas as a domain for AI had already been studied by several authors).

The 1959 paper said:

> The *advice taker* is a proposed program for solving problems by manipulating sentences in formal languages. The main difference between it and other programs or proposed programs for manipulating formal languages (the *Logic Theory Machine* of Newell, Simon and Shaw and the Geometry Program of Gelernter) is that in the previous programs the formal system was the subject matter but the heuristics were all embodied in the program. In this program the procedures will be described as much as possible in the language itself and, in particular, the heuristics are all so described.

> The main advantages we expect the *advice taker* to have is that its behavior will be improvable merely by making statements to it, telling it about its symbolic environment and what is wanted from it. To make these statements will require little if any knowledge of the program or the previous knowledge of the *advice taker*. One will be able to assume that the *advice taker* will have available to it a fairly wide class of immediate logical consequences of anything it is told and its previous knowledge. This property is expected to have much in common with what makes us describe certain humans as having *common sense*. We shall therefore say that *a program has common sense if it automatically deduces for itself a sufficiently wide class of immediate consequences of anything it is told and what it already knows.*

The main reasons for using logical sentences extensively in AI are better understood by researchers today than in 1959. Expressing information in declarative sentences is far more modular than expressing it in segments of computer program or in tables. Sentences can be true in much wider contexts than specific programs can be useful. The supplier of a fact does not have to understand much about how the receiver functions, or how or whether the receiver will use it. The same fact can be used for many purposes, because the logical consequences of collections of facts can be available.

The *advice taker* prospectus was ambitious in 1959, would be considered ambitious today and is still far from being immediately realizable. This is especially true of the goal of expressing the heuristics guiding the search for a way to achieve the goal in the language itself. The rest of this paper is largely concerned with describing what progress has been made, what the obstacles are, and how the prospectus has been modified in the light of what has been discovered.

The formalisms of logic have been used to differing extents in AI. Most of the uses are much less ambitious than the proposals of (McCarthy 1959). We can distinguish four levels of use of logic.

1. A machine may use no logical sentences—all its "beliefs" being implicit in its state. Nevertheless, it is often appropriate to ascribe beliefs and goals to the program, i.e. to remove the above sanitary quotes, and to use a principle of rationality—*It does what it thinks will achieve its goals.* Such ascription is discussed from somewhat different points of view in (Dennett 1971), (McCarthy 1979a) and (Newell 1981). The advantage is that the intent

of the machine's designers and the way it can be expected to behave may be more readily described *intentionally* than by a purely physical description.

The relation between the physical and the *intentional* descriptions is most readily understood in simple systems that admit readily understood descriptions of both kinds, e.g. thermostats. Some finicky philosophers object to this, contending that unless a system has a full human mind, it shouldn't be regarded as having any mental qualities at all. This is like omitting the numbers 0 and 1 from the number system on the grounds that numbers aren't required to count sets with no elements or one element. Indeed if your main interest is the null set or unit sets, numbers *are* irrelevant. However, if your interest is the number system you lose clarity and uniformity if you omit 0 and 1. Likewise, when one studies phenomena like belief, e.g. because one wants a machine with beliefs and which reasons about beliefs, it works better not to exclude simple cases from the formalism. One battle has been over whether it should be forbidden to ascribe to a simple thermostat the belief that the room is too cold. (McCarthy 1979a) says much more about ascribing mental qualities to machines, but that's not where the main action is in AI.

2. The next level of use of logic involves computer programs that use sentences in machine memory to represent their beliefs but use other rules than ordinary logical inference to reach conclusions. New sentences are often obtained from the old ones by ad hoc programs. Moreover, the sentences that appear in memory belong to a program-dependent subset of the logical language being used. Adding certain true sentences in the language may even spoil the functioning of the program. The languages used are often rather unexpressive compared to first order logic, for example they may not admit quantified sentences, or they may use a different notation from that used for ordinary facts to represent "rules", i.e. certain universally quantified implication sentences. Most often, conditional rules are used in just one direction, i.e. contrapositive reasoning is not used. Usually the program cannot infer new rules; rules must have all been put in by the "knowledge engineer". Sometimes programs have this form through mere ignorance, but the usual reason for the restriction is the practical desire to make the program run fast and deduce just the kinds of conclusions its designer anticipates. We believe the need for such specialized inference will turn out to be temporary and will be reduced or eliminated by improved ways of controlling general inference, e.g. by allowing the heuristic rules to be also expressed as sentences

as promised in the above extract from the 1959 paper.

3. The third level uses first order logic and also logical deduction. Typically the sentences are represented as clauses, and the deduction methods are based on J. Allen Robinson's (1965) method of resolution. It is common to use a theorem prover as a problem solver, i.e. to determine an x such that $P(x)$ as a byproduct of a proof of the formula $\exists x P(x)$. This level is less used for practical purposes than level two, because techniques for controlling the reasoning are still insufficiently developed, and it is common for the program to generate many useless conclusions before reaching the desired solution. Indeed, unsuccessful experience (Green 1969) with this method led to more restricted uses of logic, e.g. the STRIPS system of (Nilsson and Fikes 1971).

The commercial "expert system shells", e.g. ART, KEE and OPS-5, use logical representation of facts, usually ground facts only, and separate facts from rules. They provide elaborate but not always adequate ways of controlling inference.

In this connection it is important to mention logic programming, first introduced in Microplanner (Sussman et al., 1971) and from different points of view by Robert Kowalski (1979) and Alain Colmerauer in the early 1970s. A recent text is (Sterling and Shapiro 1986). Microplanner was a rather unsystematic collection of tools, whereas Prolog relies almost entirely on one kind of logic programming, but the main idea is the same. If one uses a restricted class of sentences, the so-called Horn clauses, then it is possible to use a restricted form of logical deduction. The control problem is then much eased, and it is possible for the programmer to anticipate the course the deduction will take. The price paid is that only certain kinds of facts are conveniently expressed as Horn clauses, and the depth first search built into Prolog is not always appropriate for the problem.

Even when the relevant facts can be expressed as Horn clauses supplemented by negation as failure, the reasoning carried out by a Prolog program may not be appropriate. For example, the fact that a sealed container is sterile if all the bacteria in it are dead and the fact that heating a can kills a bacterium in the can are both expressible as Prolog clauses. However, the resulting program for sterilizing a container will kill each bacterium individually, because it will have to index over the bacteria. It won't reason that heating the can kills all the bacteria at once, because it doesn't do universal generalization.

Here's a Prolog program for testing whether a container is sterile. The predicate symbols have obvious meanings.

```
not(P) :- P, !, fail.
not(P).
sterile(X) :- not(nonsterile(X)).
nonsterile(X) :-
    bacterium(Y), in(Y,X), not(dead(Y)).
hot(Y) :- in(Y,X), hot(X).
dead(Y) :- bacterium(Y), hot(Y).
bacterium(b1).
bacterium(b2).
bacterium(b3).
bacterium(b4).
in(b1,c1).
in(b2,c1).
in(b3,c2).
in(b4,c2).
hot(c1).
```

Giving Prolog the goal $sterile(c1)$ and $sterile(c2)$ gives the answers *yes* and *no* respectively. However, Prolog has indexed over the bacteria in the containers.

The following is a Prolog program that can verify whether a sequence of actions, actually just heating it, will sterilize a container. It involves introducing situations analogous to those discussed in (McCarthy and Hayes 1969).

```
not(P) :- P, !, fail.
not(P).
sterile(X,S) :- not(nonsterile(X,S)).
nonsterile(X,S) :-
    bacterium(Y), in(Y,X), not(dead(Y,S)).
hot(Y,S) :- in(Y,X), hot(X,S).
dead(Y,S) :- bacterium(Y), hot(Y,S).
bacterium(b1).
bacterium(b2).
bacterium(b3).
bacterium(b4).
in(b1,c1).
in(b2,c1).
in(b3,c2).
in(b4,c2).
```

```
hot(C,result(heat(C),S)).
```

When the program is given the goals $sterile(c1, heat(c1, s0))$ and $sterile(c2, heat(c1, s0))$ it answers *yes* and *no* respectively. However, if it is given the goal $sterile(c1, s)$, it will fail because Prolog lacks what logic programmers call "constructive negation".

The same facts as are used in the first Prolog program can be expressed in in a first order language as follows.

$$(\forall X)(sterile(X) \equiv (\forall Y)(bacterium(Y) \wedge in(Y, X) \supset dead(Y))),$$

$$(\forall XY)(hot(X) \wedge in(Y, X) \supset hot(Y)),$$

$$(\forall Y)(bacterium(Y) \wedge hot(Y) \supset dead(Y)),$$

and

$$hot(a).$$

However, from them we can prove $sterile(a)$ without having to index over the bacteria.

Expressibility in Horn clauses, whether supplemented by negation as failure or not, is an important property of a set of facts and logic programming has been successfully used for many applications. However, it seems unlikely to dominate AI programming as some of its advocates hope.

Although third level systems express both facts and rules as logical sentences, they are still rather specialized. The axioms with which the programs begin are not general truths about the world but are sentences whose meaning and truth is limited to the narrow domain in which the program has to act. For this reason, the "facts" of one program usually cannot be used in a database for other programs.

4. The fourth level is still a goal. It involves representing general facts about the world as logical sentences. Once put in a database, the facts can be used by any program. The facts would have the neutrality of purpose characteristic of much human information. The supplier of information would not have to understand the goals of the potential user or how his mind works. The present ways of "teaching" computer programs by modifying them or directly modifying their databases amount to "education by brain surgery".

A key problem for achieving the fourth level is to develop a language for a general common-sense database. This is difficult, because the *common-sense* *informatic situation* is complex. Here is a preliminary list of features and considerations.

1. Entities of interest are known only partially, and the information about entities and their relations that may be relevant to achieving goals cannot be permanently separated from irrelevant information. (Contrast this with the situation in gravitational astronomy in which it is stated in the informal introduction to a lecture or textbook that the chemical composition and shape of a body are irrelevant to the theory; all that counts is the body's mass, and its initial position and velocity.)

Even within gravitational astronomy, non-equational theories arise and relevant information may be difficult to determine. For example, it was recently proposed that periodic extinctions discovered in the paleontological record are caused by showers of comets induced by a companion star to the sun that encounters and disrupts the Oort cloud of comets every time it comes to perihelion. This theory is qualitative because neither the orbit of the hypothetical star nor those of the comets are available.

2. The formalism has to be *epistemologically adequate*, a notion introduced in (McCarthy and Hayes 1969). This means that the formalism must be capable of representing the information that is actually available, not merely capable of representing actual complete states of affairs.

For example, it is insufficient to have a formalism that can represent the positions and velocities of the particles in a gas. We can't obtain that information, our largest computers don't have the memory to store it even if it were available, and our fastest computers couldn't use the information to make predictions even if we could store it.

As a second example, suppose we need to be able to predict someone's behavior. The simplest example is a clerk in a store. The clerk is a complex individual about whom a customer may know little. However, the clerk can usually be counted on to accept money for articles brought to the counter, wrap them as appropriate and not protest when the customer then takes the articles from the store. The clerk can also be counted on to object if the customer attempts to take the articles without paying the appropriate price. Describing this requires a formalism capable of representing information about human social institutions. Moreover, the formalism must be capable of representing partial information about the institution, such as a three year old's knowledge of store clerks. For example, a three year old doesn't know the clerk is an employee or even what that means. He doesn't require detailed information about the clerk's psychology, and anyway this information is not ordinarily available.

The following sections deal mainly with the advances we see as required to achieve the fourth level of use of logic in AI.

2 Formalized Nonmonotonic Reasoning

It seems that fourth level systems require extensions to mathematical logic. One kind of extension is formalized *nonmonotonic reasoning*, first proposed in the late 1970s (McCarthy 1977, 1980, 1986), (Reiter 1980), (McDermott and Doyle 1980), (Lifschitz 1989a). Mathematical logic has been monotonic in the following sense. If we have $A \vdash p$ and $A \subset B$, then we also have $B \vdash p$.

If the inference is logical deduction, then exactly the same proof that proves p from A will serve as a proof from B. If the inference is model-theoretic, i.e. p is true in all models of A, then p will be true in all models of B, because the models of B will be a subset of the models of A. So we see that the monotonic character of traditional logic doesn't depend on the details of the logical system but is quite fundamental.

While much human reasoning is monotonic, some important human common-sense reasoning is not. We reach conclusions from certain premisses that we would not reach if certain other sentences were included in our premisses. For example, if I hire you to build me a bird cage, you conclude that it is appropriate to put a top on it, but when you learn the further fact that my bird is a penguin you no longer draw that conclusion. Some people think it is possible to try to save monotonicity by saying that what was in your mind was not a general rule about birds flying but a probabilistic rule. So far these people have not worked out any detailed epistemology for this approach, i.e. exactly what probabilistic sentences should be used. Instead AI has moved to directly formalizing nonmonotonic logical reasoning. Indeed it seems to me that when probabilistic reasoning (and not just the axiomatic basis of probability theory) has been fully formalized, it will be formally nonmonotonic.

Nonmonotonic reasoning is an active field of study. Progress is often driven by examples, e.g. the Yale shooting problem (Hanks and McDermott 1986), in which obvious axiomatizations used with the available reasoning formalisms don't seem to give the answers intuition suggests. One direction being explored (Moore 1985, Gelfond 1987, Lifschitz 1989a) involves putting facts about belief and knowledge explicitly in the axioms —even when the axioms concern nonmental domains. Moore's classical example (now 4 years old) is "If I had an elder brother I'd know it."

Kraus and Perlis (1988) have proposed to divide much nonmonotonic reasoning into two steps. The first step uses Perlis's (1988) autocircumscription to get a second order formula characterizing what is possible. The second step involves default reasoning to choose what is normally to be expected out of the previously established possibilities. This seems to be a promising approach.

(Ginsberg 1987) collects the main papers up to 1986. Lifschitz (1989c) summarizes some example research problems of nonmonotonic reasoning.

3 Some Formalizations and their Problems

(McCarthy 1986) discusses several formalizations, proposing those based on nonmonotonic reasoning as improvements of earlier ones. Here are some.

1. Inheritance with exceptions. Birds normally fly, but there are exceptions, e.g. ostriches and birds whose feet are encased in concrete. The first exception might be listed in advance, but the second has to be derived or verified when mentioned on the basis of information about the mechanism of flying and the properties of concrete.

There are many ways of nonmonotonically axiomatizing the facts about which birds can fly. The following axioms using a predicate *ab* standing for "abnormal" seem to me quite straightforward.

$$(\forall x)(\neg ab(aspect1(x)) \supset \neg flies(x)) \tag{1}$$

Unless an object is abnormal in *aspect*1, it can't fly.

It wouldn't work to write $ab(x)$ instead of $ab(aspect1(x))$, because we don't want a bird that is abnormal with respect to its ability to fly to be automatically abnormal in other respects. Using aspects limits the effects of proofs of abnormality.

$$(\forall x)(bird(x) \supset ab(aspect1(x))). \tag{2}$$

$$(\forall x)(bird(x) \wedge \neg ab(aspect2(x)) \supset flies(x)). \tag{3}$$

Unless a bird is abnormal in *aspect*2, it can fly.

When these axioms are combined with other facts about the problem, the predicate *ab* is then to be *circumscribed*, i.e. given its minimal extent compatible with the facts being taken into account. This has the effect that a bird will be considered to fly unless other axioms imply that it is abnormal in *aspect*2. (2) is called a cancellation of inheritance axiom, because it explicitly cancels the general presumption that objects don't fly. This approach works fine when the inheritance hierarchy is given explicitly. More elaborate approaches, some of which are introduced in (McCarthy 1986) and improved in (Haugh 1988), are required when hierarchies with indefinite numbers of sorts are considered.

2. (McCarthy 1986) contains a similar treatment of the effects of actions like moving and painting blocks using the situation calculus. Moving and painting are axiomatized entirely separately, and there are no axioms saying that moving a block doesn't affect the positions of other blocks or the colors of blocks. A general "common-sense law of inertia"

$$\begin{aligned}(\forall pes)(holds(p,s) \wedge \neg ab(aspect1(p,e,s)) \\ \supset holds(p,result(e,s))),\end{aligned} \tag{2}$$

asserts that a fact p that holds in a situation s is presumed to hold in the situation $result(e,s)$ that results from an event e unless there is evidence to the contrary. Unfortunately, Lifschitz (1985 personal communication) and Hanks and McDermott (1986) showed that simple treatments of the common-sense law of inertia admit unintended models. Several authors have given more elaborate treatments, but in my opinion, the results are not yet entirely satisfactory. The best treatment so far seems to be that of (Lifschitz 1987).

4 Ability, Practical Reason and Free Will

An AI system capable of achieving goals in the common-sense world will have to reason about what it and other actors can and cannot do. For concreteness, consider a robot that must act in the same world as people and perform tasks that people give it. Its need to reason about its abilities puts the traditional philosophical problem of free will in the following form. What view shall we build into the robot about its own abilities, i.e. how shall we make it reason about what it can and cannot do? (Wishing to avoid begging any questions, by *reason* we mean *compute* using axioms, observation sentences, rules of inference and nonmonotonic rules of conjecture.)

Let A be a task we want the robot to perform, and let B and C be alternate intermediate goals either of which would allow the accomplishment of A. We want the robot to be able to choose between attempting B and attempting C. It would be silly to program it to reason: "I'm a robot and a deterministic device. Therefore, I have no choice between B and C. What I will do is determined by my construction." Instead it must decide in some way which of B and C it can accomplish. It should be able to conclude in some cases that it can accomplish B and not C, and therefore it should take B as a subgoal on the way to achieving A. In other cases it should conclude that it *can* accomplish either B or C and should choose whichever is evaluated as better according to the criteria we provide it.

(McCarthy and Hayes 1969) proposes conditions on the semantics of any formalism within which the robot should reason. The essential idea is that what the robot can do is determined by the place the robot occupies in the world—not by its internal structure. For example, if a certain sequence of outputs from the robot will achieve B, then we conclude or it concludes that the robot can achieve B without reasoning about whether the robot will actually produce that sequence of outputs.

Our contention is that this is approximately how any system, whether human or robot, must reason about its ability to achieve goals. The basic formalism will be the same, regardless of whether the system is reasoning about its own abilities or about those of other systems including people.

The above-mentioned paper also discusses the complexities that come up when a strategy is required to achieve the goal and when internal inhibitions or lack of knowledge have to be taken into account.

5 Three Approaches to Knowledge and Belief

Our robot will also have to reason about its own knowledge and that of other robots and people.

This section contrasts the approaches to knowledge and belief characteristic of philosophy, philosophical logic and artificial intelligence. Knowledge and belief have long been studied in epistemology, philosophy of mind and in philosophical logic. Since about 1960, knowledge and belief have also been studied in AI. (Halpern 1986) and (Vardi 1988) contain recent work, mostly oriented to computer science including AI.

It seems to me that philosophers have generally treated knowledge and belief as *complete natural kinds*. According to this view there is a fact to be discovered about what beliefs are. Moreover, once it is decided what the objects of belief are (e.g. sentences or propositions), the definitions of belief ought to determine for each such object p whether the person believes it or not. This last is the completeness mentioned above. Of course, only human and sometimes animal beliefs have mainly been considered. Philosophers have differed about whether machines can ever be said to have beliefs, but even those who admit the possibility of machine belief consider that what beliefs are is to be determined by examining human belief.

The formalization of knowledge and belief has been studied as part of philosophical logic, certainly since Hintikka's book (1964), but much of the earlier work in modal logic can be seen as applicable. Different logics and axioms systems sometimes correspond to the distinctions that less formal philosophers make, but sometimes the mathematics dictates different distinctions.

AI takes a different course because of its different objectives, but I'm inclined to recommend this course to philosophers also, partly because we want their help but also because I think it has philosophical advantages.

The first question AI asks is: Why study knowledge and belief at all? Does a computer program solving problems and achieving goals in the common-sense world require beliefs, and must it use sentences about beliefs? The answer to both questions is approximately yes. At least there have to be data structures whose usage corresponds closely to human usage in some cases. For example, a robot that could use the American air transportation system has to know that travel agents know airline schedules, that there is a book (and now a computer accessible database) called the OAG that contains this information. If it is to be able to plan a trip with intermediate stops it has to have the general information that the departure gate from an intermediate stop is not to be discovered when the trip is first planned but will be available on arrival at the intermediate stop. If the robot has to keep secrets, it has to know about how information can be obtained by inference from other information, i.e. it has to have some kind of information model of the people from whom it is to keep the secrets.

However, none of this tells us that the notions of knowledge and belief to be built into our computer programs must correspond to the goals philosophers have been trying to achieve. For example, the difficulties involved in building a system that knows what travel agents know about airline schedules are not substantially connected with questions about how the travel agents can be absolutely certain. Its notion of knowledge doesn't have to be complete; i.e. it doesn't have to determine in all cases whether a person is to be regarded as knowing a given proposition. For many tasks it doesn't have to have opinions about when true belief doesn't constitute knowledge. The designers of AI systems can try to evade philosophical puzzles rather than solve them.

Maybe some people would suppose that if the question of certainty is avoided, the problems formalizing knowledge and belief become straightforward. That has not been our experience.

As soon as we try to formalize the simplest puzzles involving knowledge, we encounter difficulties that philosophers have rarely if ever attacked.

Consider the following puzzle of Mr. S and Mr. P.

Two numbers m and n are chosen such that $2 \leq m \leq n \leq 99$. Mr. S is told their sum and Mr. P is told their product. The following dialogue ensues:

Mr. P: I don't know the numbers.
Mr. S: I knew you didn't know them. I don't know them either.
Mr. P: Now I know the numbers.
Mr. S: Now I know them too.

In view of the above dialogue, what are the numbers?

Formalizing the puzzle is discussed in (McCarthy 1989). For the present we mention only the following aspects.

1. We need to formalize *knowing what*, i.e. knowing what the numbers are, and not just *knowing that*.

2. We need to be able to express and prove non-knowledge as well as knowledge. Specifically we need to be able to express the fact that as far as Mr. P knows, the numbers might be any pair of factors of the known product.

3. We need to express the joint knowledge of Mr. S and Mr. P of the conditions of the problem.

4. We need to express the change of knowledge with time, e.g. how Mr. P's knowledge changes when he hears Mr. S say that he knew that Mr. P didn't know the numbers and doesn't know them himself. This includes inferring what Mr. S and Mr. P still won't know.

The first order language used to express the facts of this problem involves an accessibility relation $A(w1, w2, p, t)$, modeled on Kripke's semantics for modal logic. However, the accessibility relation here is in the language itself rather than in a metalanguage. Here $w1$ and $w2$ are possible worlds, p is a person and t is an integer time. The use of possible worlds makes it convenient to express non-knowledge. Assertions of non-knowledge are expressed as the existence of accessible worlds satisfying appropriate conditions.

The problem was successfully expressed in the language in the sense that an arithmetic condition determining the values of the two numbers can be deduced from the statement. However, this is not good enough for AI. Namely, we would like to include facts about knowledge in a general purpose common-sense database. Instead of an *ad hoc* formalization of Mr. S and Mr. P, the problem should be solvable from the same general facts about knowledge that might be used to reason about the knowledge possessed by travel agents supplemented only by the facts about the dialogue. Moreover, the language of the general purpose database should accommodate all the modalities that might be wanted and not just knowledge. This suggests using ordinary logic, e.g. first order logic, rather than modal logic, so that the modalities can be ordinary functions or predicates rather than modal operators.

Suppose we are successful in developing a "knowledge formalism" for our common-sense database that enables the program controlling a robot to solve puzzles and plan trips and do the other tasks that arise in the common-sense environment requiring reasoning about knowledge. It will surely be asked whether it is really *knowledge* that has been formalized. I doubt that the question has an answer. This is perhaps the question of whether knowledge is a natural kind.

I suppose some philosophers would say that such problems are not of philosophical interest. It would be unfortunate, however, if philosophers were to abandon such a substantial part of epistemology to computer science. This is because the analytic skills that philosophers have acquired are relevant to the problems.

6 Reifying Context

We propose the formula $holds(p, c)$ to assert that the proposition p holds in context c. It expresses explicitly how the truth of an assertion depends on context. The relation $c1 \leq c2$ asserts that the context $c2$ is more general than the context $c1$.[1]

Formalizing common-sense reasoning needs contexts as objects, in order to match human ability to consider context explicitly. The proposed database of general common-sense knowledge will make assertions in a general context called $C0$. However, $C0$ cannot be maximally general, because it will surely involve unstated presuppositions. Indeed we claim that there can be no maximally general context. Every context involves unstated presuppositions, both linguistic and factual.

Sometimes the reasoning system will have to transcend $C0$, and tools will have to be provided to do this. For example, if Boyle's law of the dependence of the volume of a sample of gas on pressure were built into $C0$, discovery of its dependence on temperature would have to trigger a process of generalization that might lead to the perfect gas law.

The following ideas about how the formalization might proceed are tentative. Moreover, they appeal to recent logical innovations in the formalization of nonmonotonic reasoning. In particular, there will be nonmonotonic "inheritance rules" that allow default inference from $holds(p, c)$ to $holds(p, c')$, where c' is either more general or less general than c.

Almost all previous discussion of context has been in connection with natural language, and the present paper relies heavily on examples from natural language. However, I believe the main AI uses of formalized context will not be in connection with communication but in connection with reasoning about the effects of actions directed to achieving goals. It's just that natural language examples come to mind more readily.

As an example of intended usage, consider

$$holds(at(he, inside(car)), c17).$$

Suppose that this sentence is intended to assert that a particular person is in a particular car on a particular occasion, i.e. the sentence is not just being used as a linguistic example but is meant seriously. A corresponding English sentence is "He's in the car" where who he is and which car and when is determined by the context in which the sentence is uttered. Suppose, for simplicity, that the sentence is said by one person to another in a situation in which the car is visible to the speaker but not to the hearer and the time at which the the subject is asserted to be in the car is the same time at which the sentence is uttered.

[1] 1996: In subsequent papers the notation $ist(c, p)$ was used.

In our formal language $c17$ has to carry the information about who he is, which car and when.

Now suppose that the same fact is to be conveyed as in example 1, but the context is a certain Stanford Computer Science Department 1980s context. Thus familiarity with cars is presupposed, but no particular person, car or occasion is presupposed. The meanings of certain names is presupposed, however. We can call that context (say) $c5$. This more general context requires a more explicit proposition; thus, we would have

$$\begin{aligned} &holds(at(\text{“}Timothy\ McCarthy\text{”}, inside((\iota x)(iscar(x) \wedge \\ &\quad \wedge belongs(x, \text{“}John\ McCarthy\text{”})))), c5). \end{aligned} \tag{3}$$

A yet more general context might not identify a specific John McCarthy, so that even this more explicit sentence would need more information. What would constitute an adequate identification might also be context dependent.

Here are some of the properties formalized contexts might have.

1. In the above example, we will have $c17 \leq c5$, i.e. $c5$ is more general than $c17$. There will be nonmonotonic rules like

$$(\forall c1\ c2\ p)(c1 \leq c2) \wedge holds(p, c1) \wedge \neg ab1(p, c1, c2) \supset holds(p, c2) \tag{4}$$

and

$$(\forall c1\ c2\ p)(c1 \leq c2) \wedge holds(p, c2) \wedge \neg ab2(p, c1, c2) \supset holds(p, c1). \tag{5}$$

Thus there is nonmonotonic inheritance both up and down in the generality hierarchy.

2. There are functions forming new contexts by specialization. We could have something like

$$c19 = specialize(he = Timothy\ McCarthy, belongs(car, John\ McCarthy), c5). \tag{6}$$

We will have $c19 \leq c5$.

3. Besides $holds(p, c)$, we may have $value(term, c)$, where $term$ is a term. The domain in which $term$ takes values is defined in some outer context.

4. Some presuppositions of a context are linguistic and some are factual. In the above example, it is a linguistic matter who the names refer to. The

properties of people and cars are factual, e.g. it is presumed that people fit into cars.

5. We may want meanings as abstract objects. Thus we might have

$$\mathit{meaning}(\mathit{he}, c17) = \mathit{meaning}(\text{“\textit{Timothy McCarthy}”}, c5).$$

6. Contexts are “rich” entities not to be fully described. Thus the “normal English language context” contains factual assumptions and linguistic conventions that a particular English speaker may not know. Moreover, even assumptions and conventions in a context that may be individually accessible cannot be exhaustively listed. A person or machine may know facts about a context without “knowing the context”.

7. Contexts should not be confused with the situations of the situation calculus of (McCarthy and Hayes 1969). Propositions about situations can hold in a context. For example, we may have

$$\begin{aligned}&\mathit{holds}(\mathit{Holds}1(\mathit{at}(I, \mathit{airport}), \mathit{result}(\mathit{drive\text{-}to}(\mathit{airport},\\ &\mathit{result}(\mathit{walk\text{-}to}(\mathit{car}), S0))), c1).\end{aligned} \tag{7}$$

This can be interpreted as asserting that under the assumptions embodied in context $c1$, a plan of walking to the car and then driving to the airport would get the robot to the airport starting in situation $S0$.

8. The context language can be made more like natural language and more extensible if we introduce notions of entering and leaving a context. These will be analogous to the notions of making and discharging assumptions in natural deduction systems, but the notion seems to be more general. Suppose we have $\mathit{holds}(p, c)$. We then write

enter c.

This enables us to write p instead of $\mathit{holds}(p, c)$. If we subsequently infer q, we can replace it by $\mathit{holds}(q, c)$ and leave the context c. Then $\mathit{holds}(q, c)$ will itself hold in the outer context in which $\mathit{holds}(p, c)$ holds. When a context is entered, there need to be restrictions analogous to those that apply in natural deduction when an assumption is made.

One way in which this notion of entering and leaving contexts is more general than natural deduction is that formulas like $\mathit{holds}(p, c1)$ and (say) $\mathit{holds}(\mathit{notp}, c2)$ behave differently from $c1 \supset p$ and $c2 \supset \neg p$ which are their natural deduction analogs. For example, if $c1$ is associated with the time 5pm and $c2$ is associated with the time 6pm and p is $at(I, office)$, then $holds(p, c1) \wedge holds(not\ p, c2)$ might be used to infer that I left the office between 5pm and 6pm. $(c1 \supset p) \wedge (c2 \supset \neg p)$ cannot be used in this way; in fact it is equivalent to $\neg c1 \vee \neg c2$.

9. The expression $Holds(p, c)$ (note the caps) represents the proposition that p holds in c. Since it is a proposition, we can assert $holds(Holds(p, c), c')$.

10. Propositions will be combined by functional analogs of the Boolean operators as discussed in (McCarthy 1979b). Treating propositions involving quantification is necessary, but it is difficult to determine the right formalization.

11. The major goals of research into formalizing context should be to determine the rules that relate contexts to their generalizations and specializations. Many of these rules will involve nonmonotonic reasoning.

7 Remarks

The project of formalizing common-sense knowledge and reasoning raises many new considerations in epistemology and also in extending logic. The role that the following ideas might play is not clear yet.

7.1 Epistemological Adequacy often Requires Approximate Partial Theories

(McCarthy and Hayes 1969) introduces the notion of epistemological adequacy of a formalism. The idea is that the formalism used by an AI system must be adequate to represent the information that a person or program with given opportunities to observe can actually obtain. Often an epistemologically adequate formalism for some phenomenon cannot take the form of a classical scientific theory. I suspect that some people's demand for a classical scientific theory of certain phenomena leads them to despair about formalization. Consider a theory of a dynamic phenomenon, i.e. one that changes in time. A classical scientific theory represents the state of the phenomenon in some way and describes how it evolves with time, most classically by differential equations.

What can be known about common-sense phenomena usually doesn't permit such complete theories. Only certain states permit prediction of the future. The phenomenon arises in science and engineering theories also, but I suspect that philosophy of science sweeps these cases under the rug. Here are some examples.

(1) The theory of linear electrical circuits is complete within its model of the phenomena. The theory gives the response of the circuit to any time varying voltage. Of course, the theory may not describe the actual physics, e.g. the current may overheat the resistors. However, the theory of sequential digital circuits is incomplete from the beginning. Consider a circuit built from NAND-gates and D flipflops and timed synchronously by an appropriate clock. The behavior of a D flipflop is defined by the theory when one of its inputs is 0 and the other is 1 when the inputs are appropriately clocked. However, the behavior is not defined by the theory when both inputs are 0 or both are 1. Moreover, one can easily make circuits in such a way that both inputs of some flipflop get 0 at some time.

This lack of definition is not an oversight. The actual signals in a digital circuit are not ideal square waves but have finite rise times and often overshoot their nominal values. However, the circuit will behave as though the signals were ideal provided the design rules are obeyed. Making both inputs to a flipflop nominally 0 creates a situation in which no digital theory can describe what happens, because the behavior then depends on the actual time-varying signals and on manufacturing variations in the flipflops.

(2) Thermodynamics is also a partial theory. It tells about equilibria and it tells which directions reactions go, but it says nothing about how fast they go.

(3) The common-sense database needs a theory of the behavior of clerks in stores. This theory should cover what a clerk will do in response to bringing items to the counter and in response to a certain class of inquiries. How he will respond to other behaviors is not defined by the theory.

(4) (McCarthy 1979a) refers to a theory of skiing that might be used by ski instructors. This theory regards the skier as a stick figure with movable joints. It gives the consequences of moving the joints as it interacts with the shape of the ski slope, but it says nothing about what causes the joints to be moved in a particular way. Its partial character corresponds to what experience teaches ski instructors. It often assigns truth values to counterfactual conditional assertions like, “If he had bent his knees more, he wouldn’t have fallen”.

7.2 Meta-epistemology

If we are to program a computer to think about its own methods for gathering information about the world, then it needs a language for expressing assertions about the relation between the world, the information gathering methods available to an information seeker and what it can learn. This leads to a subject I like to call meta-epistemology. Besides its potential applications to AI, I believe it has applications to philosophy considered in the traditional sense.

Meta-epistemology is proposed as a mathematical theory in analogy to metamathematics. Metamathematics considers the mathematical properties of mathematical theories as objects. In particular model theory as a branch of metamathematics deals with the relation between theories in a language and interpretations of the non-logical symbols of the language. These interpretations are considered as mathematical objects, and we are only sometimes interested in a preferred or true interpretation.

Meta-epistemology considers the relation between the world, languages for making assertions about the world, notions of what assertions are considered meaningful, what are accepted as rules of evidence and what a knowledge seeker can discover about the world. All these entities are considered as mathematical objects. In particular the world is considered as a parameter. Thus meta-epistemology has the following characteristics.

1. It is a purely mathematical theory. Therefore, its controversies, assuming there are any, will be mathematical controversies rather than controversies about what the real world is like. Indeed metamathematics gave many philosophical issues in the foundations of mathematics a technical content. For example, the theorem that intuitionist arithmetic and Peano arithmetic are equi-consistent removed at least one area of controversy between those whose mathematical intuitions support one view of arithmetic or the other.

2. While many modern philosophies of science assume some relation between what is meaningful and what can be verified or refuted, only special meta-epistemological systems will have the corresponding mathematical property that all aspects of the world relate to the experience of the knowledge seeker.

This has several important consequences for the task of programming a knowledge seeker.

A knowledge seeker should not have a priori prejudices (principles) about what concepts might be meaningful. Whether and how a proposed concept about the world might ever connect with observation may remain in suspense for a very long time while the concept is investigated and related to other concepts.

We illustrate this by a literary example. Moliére's play *La Malade Imaginaire* includes a doctor who explains sleeping powders by saying that they contain a "dormitive virtue". In the play, the doctor is considered a pompous fool for offering a concept that explains nothing. However, suppose the doctor had some intuition that the dormitive virtue might be extracted and concentrated, say by shaking the powder in a mixture of ether and water. Suppose he thought that he would get the same concentrate from all substances with soporific effect. He would certainly have a fragment of scientific theory subject to later verification. Now suppose less—namely, he only believes that a common component is behind all substances whose consumption makes one sleepy but has no idea that he should try to invent a way of verifying the conjecture. He still has something that, if communicated to someone more scientifically minded, might be useful. In the play, the doctor obviously sins intellectually by claiming a hypothesis as certain. Thus a knowledge seeker must be able to form new concepts that have only extremely tenuous relations with their previous linguistic structure.

7.3 Rich and poor entities

Consider my next trip to Japan. Considered as a plan it is a discrete object with limited detail. I do not yet even plan to take a specific flight or to fly on a specific day. Considered as a future event, lots of questions may be asked about it. For example, it may be asked whether the flight will depart on time and what precisely I will eat on the airplane. We propose characterizing the actual trip as a rich entity and the plan as a poor entity. Originally, I thought that rich events referred to the past and poor ones to the future, but this seems to be wrong. It's only that when one refers to the past one is usually referring to a rich entity, while the future entities one refers to are more often poor. However, there is no intrinsic association of this kind. It seems that planning requires reasoning about the plan (poor entity) and the event of its execution (rich entity) and their relations.

(McCarthy and Hayes 1969) defines situations as rich entities. However, the actual programs that have been written to reason in situation calculus might as well regard them as taken from a finite or countable set of discrete states.

Possible worlds are also examples of rich entities as ordinarily used in philosophy. One never prescribes a possible world but only describes classes of possible worlds.

Rich entities are open ended in that we can always introduce more properties of them into our discussion. Poor entities can often be enumerated, e.g. we can often enumerate all the events that we consider reasonably likely in a situation. The passage from considering rich entities in a given discussion to considering poor entities is a step of nonmonotonic reasoning.

It seems to me that it is important to get a good formalization of the relations between corresponding rich and poor entities. This can be regarded as formalizing the relation between the world and a formal model of some aspect of the world, e.g. between the world and a scientific theory.

8 Acknowledgements

I am indebted to Vladimir Lifschitz and Richmond Thomason for useful suggestions. Some of the prose is taken from (McCarthy 1987), but the examples are given more precisely in the present paper, since *Daedalus* allows no formulas.

The research reported here was partially supported by the Defense Advanced Research Projects Agency, Contract No. N00039-84-C-0211.

9 References

Dennett, D.C. (1971): "Intentional Systems", *Journal of Philosophy,* vol. 68, No. 4, Feb. 25.

Dreyfus, Hubert L. (1972): *What Computers Can't Do: the Limits of Artificial Intelligence*, revised edition 1979, New York : Harper & Row.

Fikes, R, and Nils Nilsson, (1971): "STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving", *Artificial Intelligence*, Volume 2, Numbers 3,4, January, pp. 189-208.

Gelfond, M. (1987): "On Stratified Autoepistemic Theories", *AAAI-87* **1**, 207-211.

Ginsberg, M. (ed.) (1987): *Readings in Nonmonotonic Reasoning*, Morgan Kaufmann, 481 pp.

Green, C., (1969): "Application of Theorem Proving to Problem Solving," *First International Joint Conference on Artificial Intelligence,* pp. 219-239.

Halpern, J. (ed.) (1986): *Reasoning about Knowledge*, Morgan Kaufmann, Los Altos, CA.

Hanks, S. and D. McDermott (1986): "Default Reasoning, Nonmonotonic Logics, and the Frame Problem", AAAI-86, pp. 328-333.

Haugh, Brian A. (1988): "Tractable Theories of Multiple Defeasible Inheritance in Ordinary Nonmonotonic Logics", *Proceedings of the Seventh National Conference on Artificial Intelligence (AAAI-88)*, Morgan Kaufmann.

Hintikka, Jaakko (1964): *Knowledge and Belief; an Introduction to the Logic of the Two Notions*, Cornell Univ. Press, 179 pp.

Kowalski, Robert (1979): *Logic for Problem Solving*, North-Holland, Amsterdam.

Kraus, Sarit and Donald Perlis (1988): "Names and Non-Monotonicity", UMIACS-TR-88-84, CS-TR-2140, Computer Science Technical Report Series, University of Maryland, College Park, Maryland 20742.

Lifschitz, Vladimir (1987): "Formal theories of action", *The Frame Problem in Artificial Intelligence, Proceedings of the 1987 Workshop*, reprinted in (Ginsberg 1987).

Lifschitz, Vladimir (1989a): *Between Circumscription and Autoepistemic Logic,* to appear in the Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning, Morgan Kaufmann.

Lifschitz, Vladimir (1989b): "Circumscriptive Theories: A Logic-based Framework for Knowledge Representation," this collection.

Lifschitz, Vladimir (1989c): "Benchmark Problems for Formal Nonmonotonic Reasoning", *Non-Monotonic Reasoning*, 2nd International Workshop, Grassau, FRG, Springer-Verlag.

McCarthy, John (1959): "Programs with Common Sense", *Proceedings of the Teddington Conference on the Mechanization of Thought Processes*, Her Majesty's Stationery Office, London.

McCarthy, John and P.J. Hayes (1969): "Some Philosophical Problems from the Standpoint of Artificial Intelligence", D. Michie (ed.), *Machine Intelligence 4*, American Elsevier, New York, NY.

McCarthy, John (1977): "On The Model Theory of Knowledge" (with M. Sato, S. Igarashi, and T. Hayashi), *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, M.I.T., Cambridge, Mass.

McCarthy, John (1977): "Epistemological Problems of Artificial Intelligence", *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, M.I.T., Cambridge, Mass.

McCarthy, John (1979a): "Ascribing Mental Qualities to Machines", *Philosophical Perspectives in Artificial Intelligence*, Ringle, Martin (ed.), Harvester Press, July 1979.

McCarthy, John (1979b): "First Order Theories of Individual Concepts and Propositions", Michie, Donald (ed.), *Machine Intelligence 9*, (University of Edinburgh Press, Edinburgh).

McCarthy, John (1980): "Circumscription—A Form of Non-Monotonic Reasoning", *Artificial Intelligence*, Volume 13, Numbers 1,2, April.

McCarthy, John (1983): "Some Expert Systems Need Common Sense", *Computer Culture: The Scientific, Intellectual and Social Impact of the Computer*, Heinz Pagels (ed.), vol. 426, Annals of the New York Academy of Sciences.

McCarthy, John (1986): "Applications of Circumscription to Formalizing Common Sense Knowledge", *Artificial Intelligence*, April 1986.

McCarthy, John (1987): "Mathematical Logic in Artificial Intelligence", *Daedalus*, vol. 117, No. 1, American Academy of Arts and Sciences, Winter 1988.

McCarthy, John (1989): "Two Puzzles Involving Knowledge", *Formalizing Common Sense,* Ablex 1989.

McDermott, D. and J. Doyle, (1980): "Non-Monotonic Logic I", *Artificial Intelligence*, Vol. 13, N. 1

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

Newell, Allen (1981): "The Knowledge Level". *AI Magazine*, Vol. 2, No. 2.

Perlis, D. (1988): "Autocircumscription", *Artificial Intelligence*, **36** pp. 223-236.

Reiter, Raymond (1980): "A Logic for Default Reasoning", *Artificial Intelligence*, Volume 13, Numbers 1,2, April.

Russell, Bertrand (1913): "On the Notion of Cause", *Proceedings of the Aristotelian Society*, 13, pp. 1-26.

Robinson, J. Allen (1965): "A Machine-oriented Logic Based on the Resolution Principle", *JACM*, 12(1), pp. 23-41.

Sterling, Leon and Ehud Shapiro (1986): *The Art of Prolog*, MIT Press.

Sussman, Gerald J., Terry Winograd, and Eugene Charniak (1971): "Micro-planner Reference Manual", Report AIM-203A, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge.

Vardi, Moshe (1988): *Conference on Theoretical Aspects of Reasoning about Knowledge*, Morgan Kaufmann, Los Altos, CA.

Department of Computer Science
Stanford University
Stanford, CA 94305