EPISTEMOLOGICAL PROBLEMS OF ARTIFICIAL INTELLIGENCE

John McCarthy
Computer Science Department
Stanford University
Stanford, California 94305

Introduction

In (McCarthy and Hayes 1969), we proposed dividing the artificial intelligence problem into two parts - an epistemological part and a heuristic part. This lecture further explains this division, explains some of the epistemological problems, and presents some new results and approaches.

The epistemological part of AI studies what kinds of facts about the world are available to an observer with given opportunities to observe, how these facts can be represented in the memory of a computer, and what rules permit legitimate conclusions to be drawn from these facts. It leaves aside the heuristic problems of how to search spaces of possibilities and how to match patterns.

Considering epistemological problems separately has the following advantages:

1. The same problems of what information is available to an observer and what conclusions can be drawn from information arise in connection with a variety of problem solving tasks.

2. A single solution of the epistemological problems can support a wide variety of heuristic approaches to a problem.

3. AI is a very difficult scientific problem, so there are great advantages in finding parts of the problem that can be separated out and separately attacked.

4. As the reader will see from the examples in the next section, it is quite difficult to formalize the facts of common knowledge. Existing programs that manipulate facts in some of the domains are confined to special cases and don't face the difficulties that must be overcome to achieve very intelligent behavior.

We have found first order logic to provide suitable languages for expressing facts about the world for epistemological research. Recently we have found that introducing concepts as individuals makes possible a first order logic expression of facts usually expressed in modal logic but with important advantages over modal logic - and so far no disadvantages.

In AI literature, the term *predicate calculus* is usually extended to cover the whole of first order logic. While predicate calculus Includes just formulas built up from variables using predicate symbols, logical connectives, and quantifiers, first order logic also allows the use of function symbols to form terms and in its semantics interprets the equality symbol as standing for identity. Our first order systems further use conditional expressions (non-recursive) to form terms and -expressions with individuaal variables to form new function symbols. All these extensions are logically inessential, because every formula that includes them can be replaced by a formula of pure predicate calculus whose validity *is* equivalent to it. The extensions are heuristically non-trivial, because the equivalent predicate calculus may be much longer and is usually much more difficult to understand - for man or machine.

The use of first order logic in epistemological research is a separate issue from whether first order sentences are appropriate data structures for representing information within a program. As to the latter, sentences in logic are at one end of a spectrum of representations; they are easy to communicate, have logical consequences and can be logical consequences, and they can be meaningful in a wide context. Taking action on the basis of information stored as sentences, is slow and they are not the most compact representation of information. The opposite extreme is to build the information into hardware, next comes building it into machine language program, then a language like LISP, and then a language like MICROPLANNER, and then perhaps productions. Compiling or hardware building or "automatic programming" or just planning takes information from a more context independent form to a faster but more context dependent form. A clear expression of this is the transition from first order logic to MICROPLANNER, where much information is represented similarly but with a specification of how the information is to be used. A large AI system should represent some information as first order logic sentences and other information should be compiled. In fact, it will often be necessary to represent the same information in several ways. Thus a ball player habit of keeping his eye on the ball is built into his "program", but it is also explicitly represented as a sentence so that the advice can be communicated.

Whether first order logic makes a good programming language *is* yet another issue. So far it seems to have the qualities Samuel Johnson ascribed to a woman preaching or a dog walking on its hind legs - one is sufficiently impressed by seeing it done at all that one doesn't demand it be done well.

Suppose we have a theory of a certain class of phenomena axiomatized in (say) first order logic. We regard the theory as adequate for describing the epistemological aspects of a goal seeking process involving these phenomena provided the following criterion *is* satisfied:

Imagine a robot such that its inputs become sentences of the theory stored in the robot's data-base, and such that whenever a sentence of the form "*/ should emit output X now*" appears in its data base, the robot emits output *X*. Suppose that new sentences appear in its data base only as logical consequences of sentences already in the data base. The deduction of these sentences also use general sentences stored in the data base at the beginning constituting the theory being tested. Usually a data-base of sentences permits many different deductions to be made so that a deduction program would have to choose which deduction to make. If there was no program that could achieve the goal by making deductions allowed by the theory no matter how fast the program ran, we would have to say that the theory was epistemologically inadequate. A theory that was epistemologically adequate would be considered heuristically inadequate if no program running at a reasonable speed with any representation of the facts expressed by the data could do the Job. We believe that most present AI formalisms are epistemologically inadequate for general intelligence; i.e. they wouldn't achieve enough goals requiring general intelligence no matter how fast they were allowed to run. This is because the epistemological problems discussed in the following sections haven't even been attacked yet.

The word "epistemology" is used in this paper substantially as many philosophers use it, but the problems considered have a different emphasis. Philosophers emphasize what *is* potentially knowable with maximal opportunities to observe and compute, whereas AI must take into account what is knowable with available observational and computational facilities. Even so, many of the same formalizations have both philosophical and AI interest.

The subsequent sections of this paper list some epistemological problems, discuss some first order formalizations, introduce concepts as objects and use them to express facts about knowledge, describe a new mode of reasoning called circumscription, and place the AI problem in a philosphical setting.

Epistemological problems

We will discuss what facts a person or robot must take into account in order to achieve a goal by some strategy of action. We will ignore the question of how these facts are represented, e.g., whether they are represented by sentences from which deductions are made or whether they are built into the program. We start with great generality, so there many difficulties. We obtain successively easier problems by assuming that the difficulties we have recognized don't occur until we get to a class of problems we think we can solve.

I. We begin by asking whether solving the problem requires the co-operation of other people or overcoming their opposition. If either is true, there are two subcases. In the first subcase, the other people's desires and goals must be taken into account, and the actions they will take in given circumstances predicted on the hypothesis that they will try to achieve their goals, which may have to be discovered. The problem *is* even more difficult if bargaining is involved, because then the problems and indeterminacies of game theory are relevant. Even if bargaining is not involved, the robot still must "put himself in the place of the other people with whom he interacts". Facts like a person wanting a thing or a person disliking another must be described.

The second subcase makes the assumption that the other people can be regarded as machines with known input-output behavior. This is often a good assumption, e.g., one assumes that a clerk in a store will sell the goods in exchange for their price and that a professor will assign a grade in accordance with the quality of the work done. Neither the goals of the clerk or the professor need be taken into account; either might well regard an attempt to use them to optimize the interaction as an invasion of privacy. In such circumstances, man usually prefers to be regarded as a machine.

Let us now suppose that either other people are not involved in the problem or that the information available about their actions takes the form of input-output relations and does not involve understanding their goals.

2. The second question is whether the strategy involves the acquisition of knowledge. Even if we can treat other people as machines, we still may have to reason about what they know. Thus an airline clerk knows what airplanes fly from here to there and when, although he will tell you when asked without your having to motivate him. One must also consider information in books and in tables. The latter information is described by other information.

The second subcase of knowledge is according to whether the information obtained can be simply plugged into a program or whether it enters in a more complex way. Thus if the robot must telephone someone, its program can simply dial the number obtained, but it might have to ask a question, *"How can I get in touch with Mike?"* and reason about how to use the resulting information in conjunction with other information. The general distinction may be according to whether new sentences are generated or whether values are just assigned to variables.

An example worth considering is that a sophisticated air traveler rarely asks how he will get from the arriving flight to the departing flight at an airport where he must change planes. He is confident that the information will be available in a form he can understand at the time he will need it.

If the strategy *is* embodied in a program that branches on an environmental condition or reads a numerical parameter from the environment, we can regard it as obtaining knowledge, but this is obviously an easier case than those we have discussed.

3. A problem is more difficult if it involves concurrent events and actions. To me this seems to be the most difficult unsolved epistemological problem for AI - how to express rules that give the effects of actions and events when they occur concurrently. We may contrast this with the sequential case treated in (McCarthy and Hayes 1969). In the sequential case we can write

1) $s' - result(e,s)$

where s' is the situation that results when event e occurs in situation s. The effects of e can be described by sentences relating s', e and s. One can attempt a similar formalism giving a *partial situation* that results from an event in another partial situation, but it is difficult to see how to apply this to cases in which other events may affect with the occurrence.

When events are concurrent, it is usually necessary to regard time as continuous. We have events like *raining until the reservoir overflows* and questions like *Where was his train when we wanted to call him?.*

Computer science has recently begun to formalize parallel processes so that it *is* sometimes possible to prove that a system of parallel processes will meet its specifications. However, the knowledge available to a robot of the other processes going on in the world will rarely take the form of a Petri net or any of the other formalisms used in engineering or computer science. In fact, anyone who wishes to prove correct an airline reservation system or an air traffic control system must use information about the behavior of the external world that is less specific than a program. Nevertheless, the formalisms for expressing facts about parallel and indeterminate programs provide a start for axiomatizing concurrent action.

4. A robot must be able to express knowledge about space, and the locations, shapes and layouts of objects in space. Present

programs treat only very special cases. Usually locations are discrete - block A may be on block B but the formalisms do not allow anything to be said about where on block B it is, and what shape space is left on block B for placing other blocks or whether block A could be moved to project out a bit in order to place another block. A few are more sophisticated, but the objects must have simple geometric shapes. A formalism capable of representing the geometric information people get from seeing and handling objects has not, to my knowledge, been approached.

The difficulty in expressing such facts is indicated by the limitations of English in expressing human visual knowledge. We can describe regular geometric shapes precisely in English (fortified by mathematics), but the information we use for recognizing another person's face cannot ordinarily be transmitted in words. We can answer many more questions in the presence of a scene than we can from memory.

5. The relation between three dimensional objects and their two dimensional retinal or camera images is mostly untreated. Contrary to some philosophical positions, the three dimensional object is treated by our minds as distinct from its appearances. People blind from birth can still communicate in the same language as sighted people about three dimensional objects. We need a formalism that treats three dimensional objects as instances of patterns and their two dimensional appearances as projections of these patterns modified by lighting and occlusion.

6. Objects can be made by shaping materials and by combining other objects. They can also be taken apart, cut apart or destroyed in various ways. What people know about the relations between materials and objects remains to be described.

7. Modal concepts like *event e\ caused event e2* and *person e can do action a* are needed. (McCarthy and Hayes 1969) regards ability as a function of a person's position in a causal system and not at all as a function of his internal structure. This still seems correct, but that treatment is only metaphysically adequate, because it doesn't provide for expressing the information about ability that people actually have.

8. Suppose now that the problem can be formalized in terms of a single state that is changed by events. In interesting cases, the set of components of the state depends on the problem, but common general knowledge is usually expressed in terms of the effect of an action on one or a few components of the state. However, it cannot always be assumed that the other components are unchanged, especially because the state can be described in a variety of co-ordinate systems and the meaning of changing a single co-ordinate depends on the co-ordinate system. The problem of expressing information about what remains unchanged by an event was called *the frame problem* in (McCarthy and Hayes 1969). Minsky subsequently confused matters by using the word "frame" for patterns into which situations may fit. (His hypothesis seems to have been that almost all situations encountered in human problem solving fit into a small number of previously known patterns of situation and goal. I regard this as unlikely in difficult problems).

9. *The frame problem* may be a subcase of what we call the *quqlification problem,* and a good solution of the qualification problem may solve the frame problem also. In the *missionaries and cannibals* problem, a boat holding two people is stated to be available. In the statement of the problem, nothing is said about how boats are used to cross rivers, so obviously this information must come from common knowledge, and a computer program capable of solving the problem from an English description or from a translation of this description into logic must have the requisite common knowledge. The simplest statement about the use of boats says something like, "*// a boat is at one point on the shore of a body of water, and a set of things enter the boat, and the boat is propelled to the another point on the shore, and the things exit the boat, then they will be at the second point on the shore".* However, this statement is too rigid to be true, because anyone will admit that if the boat is a rowboat and has a leak or no oars, the action may not achieve its intended result. One might try amending the common knowledge statement about boats, but this encounters difficulties when a critic demands a qualification that the vertical exhaust stack of a diesel boat must not be struck square by a cow turd dropped by a passing hawk or some other event that no-one has previously thought of. We need to be able to say that the boat can be used as a vehicle for crossing a body of water unless something prevents it. However, since we are not willing to delimit in advance possible circumstances that may prevent the use of the boat, there is still a problem of proving or at least conjecturing that nothing prevents the use of the boat. A method of reasoning called *circumscription,* described in a subsequent section of this paper, is a candidate for solving the qualification problem. The reduction of the frame problem to the qualification problem has not been fully carried out, however.

Circumscription - a way of jumping to conclusions

There is an intuition that not all human reasoning can be translated into deduction in some formal system of mathematical logic, and therefore mathematical logic should be rejected as a formalism for expressing what a robot should know about the world The intuition in itself doesn't carry a convincing idea of what is lacking and how it might be supplied.

We can confirm part of the intuition by describing a previously unformalized mode of reasoning called *circumscription,* which we can show does not correspond to deduction in a mathematical system. The conclusions it yields are just conjectures and sometimes even introduce inconsistency. We will argue that humans often use circumscription, and robots must too. The second part of the intuition - the rejection of mathematical logic - is not confirmed; the new mode of reasoning is best understood and used within a mathematical logical framework and co-ordinates well with mathematical logical deduction. We think *circumscription* accounts for some of the successes and some of the errors of human reasoning.

The intuitive idea of *circumscription* is as follows: We know some objects in a given class and we have some ways of generating more. We jump to the conclusion that this gives all the objects in the class. Thus we *circumscribe* the class to the objects we know how to generate.

For example, suppose that objects a, *b* and *c* satisfy the predicate *P* and that the functions *fix)* and *g(x,y)* take arguments satisfying *P* into values also satisfying P. The first order logic expression of these facts is

$$2) \qquad P(a) \wedge P(b) \wedge P(c) \wedge (\forall x)(P(x) \supset P(f(x))) \wedge (\forall x\ y)(P(x) \wedge P(y) \supset P(g(x, y))).$$

The conjecture that everything satisfying *P* is generated from *a, b* and *c* by repeated application of the functions / and *g* is expressed by the sentence schema

3) $\Phi(a) \wedge \Phi(b) \wedge \Phi(c) \wedge (\forall x)(\Phi(x) \supset \Phi(f(x))) \wedge (\forall x\ y)(\Phi(x) \wedge \Phi(y) \supset \Phi(g(x,y))) \supset (\forall x)(\Phi(x) \supset P(x)),$

where ø *is* a *free* predicate variable for which any predicate may be substituted.

It is only a conjecture, because there might be an object *d* such that *P(d)* which is not generated in this way. (3) is one way of writing *the circumscription* of (2). The heuristics of circumscription - when one can plausibly conjecture that the objects generated in known ways are all there are - are completely unstudied.

Circumscription is not deduction in disguise, because every form of deduction has two properties that circumscription lacks - transitivity and what we may call *monotonicity.* Transitivity says that if *p + r* and *r + s,* then *p + s.* Monotonicity says that if *A + p* (where *A* is a set of sentences) and *A* c *B,* then *B + p* for deduction. Intuitively, circumscription should not be monotonic since it is the conjecture that the ways we know of generating P's are all there are. An enlarged set *B* of sentences may contain a new way of generating *P's.*

If we use second order logic or the language of set theory, then circumscription can be expressed as a sentence rather than as a schema. In set theory it becomes.

3') $(\forall\Phi)(a \in \Phi \wedge b \in \Phi \wedge c \in \Phi \wedge (\forall x)(x \in \Phi \supset f(x) \in \Phi) \wedge (\forall x\ y)(x \in \Phi \wedge y \in \Phi \supset g(x,y) \in \Phi)) \supset P \subset \Phi),$

but then we will still use the comprehension schema to form the set to be substituted for the set variable Φ.

The axiom schema of induction in arithmetic is the result of applying circumscription to the constant 0 and the successor operation.

There is a way of applying circumscription to an arbitrary sentence of predicate calculus. Let *p* be such a sentence and let ø be a predicate symbol. The *relativization* of *p* with respect to ø (written *pø)* is defined (as in some logic texts) as the sentence that results from replacing every quantification $(\forall x)E$ that occurs in *p* by $(\forall x)(\Phi(x) \supset E)$ and every quantification $(\exists x)E$ that occurs in *p* by $(\exists x)(\Phi(x) \wedge E)$. The circumscription of *p* is then the sentence

4) $p^{\Phi} \supset (\forall x)(P(x) \supset \Phi(x)).$

This form is correct only if neither constants nor function symbols occur in *p.* If they do, it is necessary to conjoin ø(c) for each constant *c* and $(\forall x)(\Phi(x) \supset \Phi(f(x)))$ for each single argument function symbol / to the premiss of (4). Corresponding sentences must be conjoined if there are function symbols of two or more arguments. The intuitive meaning of (4) is that the only objects satisfying *P* that exist are those that the sentence *p* forces to exist.

Applying the circumscription schema requires in venting a suitable predicate to substitute for the symbol ø (inventing a suitable set in the set-theoretic formulation). In this it resembles mathematical induction; in order to get the conclusion, we must invent a predicate for which the premiss Is true.

There is also a semantic way of looking at applying circumscription. Namely, a sentence that can be proved from a sentence *p* by circumscription is true in all minimal models of *p,* where a deduction from *p* is true in all models of *p.* Minimality is defined with respect to a containment relation s. We write that *M1 < M2* if every element of the domain of Ml is a member of the domain of *M2* and on the common members all predicates have the same truth value. It is not always true that a sentence true in all minimal models can be proved by circumscription. Indeed the minimal model of Peanos axioms *is* the standard model of arithmetic, and Codel's theorem is the assertion that not all true sentences are theorems. Minimal models don't always exist, and when they exist, they aren't always unique.

(McCarthy 1977a) treats circumscription in more detail.

Concepts as objects

We shall begin by discussing how to express such facts as *"Pat knows the combination of the safe",* although the idea of treating a concept as an object has application beyond the discussion of knowledge.

We shall use the symbol *safe* for the safe, and *combination(s)* is our notation for the combination of an arbitrary safe *s.* We aren't much interested in the domain of combinations, and we shall take them to be strings of digits with dashes in the right place, and, since a combination is a string, we will write it in quotes. Thus we can write

5) *combination(safe\)* = "45-25-17"

as a formalization of the English *"The combination of the safe is* 45-25-17". Let us suppose that the combination of *safe2* is, co-incidentally, also 45-25-17, so we can also write

6) *combination(safe2)* = "45-25-17".

Now we want to translate *"Pat knows the combination of the safe".* If we were to express it as

7) _*knows(pat, combination(safe_ I)),

the inference rule that allows replacing a term by an equal term in first order logic would let us conclude

8) _*knows{pat, combination(safe2)),_

which mightn't be true.

This problem was already recognized in 1879 by Frege, the founder of modern predicate logic, who distinguished between direct and indirect occurrences of expressions and would consider the occurrence of *combination(safe\)* in (7) to be indirect and not subject to replacement of equals by equals. The modern way of stating the problem *is* to call *Pat knows* a referentially opaque operator.

The way out of this difficulty currently most popular is to treat *Pat knows* as a *modal operator.* This involves changing the logic so that replacement of an expression by an equal expression is not allowed in opaque contexts Knowledge is not the only operator that admits modal treatment. There is also belief, wanting, and logical or physical necessity. For AI purposes, we would need all the above modal operators and many more in the same system. This would make the semantic discussion of the resulting modal logic extremely complex. For this reason, and because we want functions from material objects to concepts of them, we have followed a different path - introducing concepts as individual objects. This has not been popular in philosophy, although I suppose no-one would doubt that it could be done.

Our approach is to introduce the symbol *Safel* as a name for the concept of safel and the function *Combination* which takes a concept of a safe into a concept of its combination. The second operand of the function *knows* is now required to be a concept, and we can write

9) $knows(pat, Combination(Safe1))$

to assert that Pat knows the combination of safel. The previous trouble is avoided so long as we can assert

10) $Combination(Safe1) \neq Combination(Safe2),$

which is quite reasonable, since we do not consider the concept of the combination of *safe* to be the same as the concept of the combination of *safe2,* even if the combinations themselves are the same.

We write

11) $denotes(Safe1, safe1)$

and say that *safel* is the denotation of *Safel.* We can say that Pegasus doesn't exist by writing

12) $\neg(\exists x)(denotes(Pegasus, x))$

still admitting *Pegasus* as a perfectly good concept. If we only admit concepts with denotations (or admit partial functions into our system), we can regard denotation as a function from concepts to objects - including other concepts. We can then write

13) $safe1 = den(Safe1).$

The functions *combination* and *Combination* are related in a way that we may call extensional, namely

14) $(\forall S)(combination(den(S)) = den(Combination(S)),$

and we can also write this relation in terms of *Combination* alone as

15) $(\forall S1\ S2)(den(S1) = den(S2) \supset den(Combination(S1)) = den(Combination(S2))),$

or, in terms of the denotation predicate,

16) $(\forall S1\ S2\ s\ c)(denotes(S1, s) \wedge denotes(S2, s) \wedge denotes(Combination(S1), c) \supset denotes(Combination(S2), c)).$

It is precisely this property of extensionality that the above-mentioned *knows* predicate lacks in its second argument; it is extensional in its first argument.

Suppose we now want to say *"Pat knows that Mike knows the combination of safe\".* We cannot use *knows(mike, Combination(Safe\))* as an operand of another *knows* function for two reasons. First, the value of *knows(person, Concept)* is a truth value, and there are only two truth values, so we would either have Pat knowing all true statements or none. Second, English treats knowledge of propositions differently from the way it treats knowledge of the value of a term. To know a proposition is to know that it is true, whereas the analog of knowing a combination would be knowing whether the proposition is true.

We solve the first problem by introducing a new knowledge function Knows(Personconcept, *Concept). Knows(Mike, Combination(Safel))* is not a truth value but a *proposition,* and there can be distinct true propositions. We now need a predicate *true(proposition),* so we can assert

17) $true(Knows(Mike, Combination(Safe1))$

which is equivalent to our old-style assertion

18) $knows(mike, Combination(Safe1)).$

We now write

19) $true(Knows(Pat, Knows(Mike, Combination(Safe1))))$

to assert that Pat knows *whether* Mike knows the combination of safel. We define

20) $(\forall\ Person, Proposition)(K(Person, Proposition)\ true(Proposition)\ and\ Knows(Person, Proposition)),$

which forms the proposition *that* a person knows a proposition from the truth of the proposition and that he knows whether the proposition holds. Note that it is necessary to have new connectives to combine propositions and that an equality sign rather than an equivalence sign is used. As far as our first order logic is concerned, (20) is an assertion of the equality of two terms. These matters are discussed thoroughly in (McCarthy 1977b).

While a concept denotes at most one object, the same object can be denoted by many concepts. Nevertheless, there are often useful functions from objects to concepts that denote them. Numbers may conveniently be regarded has having *standard concepts,* and an object may have a distinguished concept relative to a particular person. (McCarthy 1977b) illustrates the use of functions from objects to concepts in formalizing such chestnuts as Russell's,"/ *thought your yacht was longer than it is".*

The most immediate AI problem that requires concepts for its successful formalism may be the relation between knowledge and ability. We would like to connect Mike's ability to open safel with his knowledge of the combination. The proper formalization of the notion of *can* that involves knowledge rather than just physical possibility hasn't been done yet. Moore (1977) discusses the relation between knowledge and action from a similar point of view, and the final version of (McCarthy 1977b) will contain some ideas about this.

There are obviously some esthetic disadvantages to a theory that has both *mike* and *Mike.* Moreover, natural language doesn't make such distinctions in its vocabulary, but in rather roundabout ways when necessary. Perhaps we could manage with just *Mike* (the concept), since the *denotation* function will be available for referring to *mike* (the person himself). It makes some sentences longer, and we have to use and equivalence relation which we may call *eqdenot* and say *"Mike eqdenot Brother{MaryY* rather than write *"mike = brother(mary)",* reserving the equality sign for equal concepts. Since many AI programs don't make much use of replacement of equals by equals, their notation may admit either interpretation, i.e., the formulas may stand for either objects or concepts. The biggest objection is that the semantics of reasoning about objects *is* more complicated if one *refers* to them only via concepts.

I believe that circumscription will turn out to be the key to inferring non-knowledge. Unfortunately, an adequate formalism has not yet been developed, so we can only give some ideas of why establishing non-knowledge is important for AI and how circumscription can contribute to it

If the robot can reason that it cannot open safe l, because it doesn't know the combination, it can decide that its next task is to find the combination. However, if it has merely failed to determine the combination by reasoning, more thinking might solve the problem. If it can safely conclude that the combination cannot be determined by reasoning, it can look for the information externally.

As another example, suppose someone asks you whether the President is standing, sitting or lying down at the moment you read the paper. Normally you will answer that you don't know and will not respond to a suggestion that you think harder. You conclude that no matter how hard you think, the information isn't to be found. If you really want to know, you must look for an external source of information. How do you know you can't solve the problem? The intuitive answer is that any answer is consistent with your other knowledge. However, you certainly don't construct a model of all your beliefs to establish this. Since you undoubtedly have some contradictory beliefs somewhere, you can't construct the required models anyway.

The process has two steps. The first is deciding what knowledge is relevant. This is a conjectural process, so its outcome is not guaranteed to be correct. It might be carried out by some kind of keyword retrieval from property lists, but there should be a less arbitrary method.

The second process uses the set of "relevant" sentences found by the first process and constructs models or circumscription predicates that allow for both outcomes if what is to be shown unknown is a proposition. If what is to be shown unknown has many possible values like a safe combination, then something more sophisticated *is* necessary. A parameter called the value of the combination is introduced, and a "model" or circumscription predicate is found in which this parameter occurs free. We used quotes, because a one parameter family of models is found rather than a single model.

We conclude with just one example of a circumscription schema dealing with knowledge. It is formalization of the assertion that all Mike knows is a consequence of propositions *P* and Q.

$$21)\qquad \Phi(P0) \wedge \Phi(Q0) \wedge (\forall P\ Q)(\Phi(P) \wedge \Phi(P\ \mathit{implies}\ Q) \supset \Phi(Q)) \supset (\forall P)(\mathit{knows}(\mathit{Mike}, P) \supset \Phi(P)).$$

Philosophical Notes

Philosophy has a more direct relation to artificial intelligence than it has to other sciences. Both subjects require the formalization of common sense knowledge and repair of its deficiencies. Since a robot with general intelligence requires some general view of the world, deficiencies in the programmers' introspection of their own world-views can result in operational weaknesses in the program. Thus many programs, including Winograd's SHRDLU, regard the history of their world as a sequence of situations each of which *is* produced by an event occuring in a previous situation of the sequence. To handle concurrent events, such programs must be rebuilt and not just provided with more facts.

This section is organized as a collection of disconnected remarks some of which have a direct technical character, while others concern the general structure of knowledge of the world. Some of them simply, give sophisticated justifications for some things that programmers are inclined to do anyway, so some people may regard them as superfluous.

1. Building a view of the world into the structure of a program does not in itself give the program the ability to state the view explicitly. Thus, none of the programs that presuppose history as a sequence of situations can make the assertion *"History is a sequence of situations".* Indeed, for a human to make his presuppositions explicit is often beyond his individual capabilities, and the sciences of psychology and philosophy still have unsolved problems in doing so.

2. Common sense requires scientific formulation. Both AI and philosophy require it, and philosophy might even be regarded as an attempt to make common sense into a science.

3. AI and philosophy both suffer from the following dilemma. Both need precise formalizations, but the fundamental structure of the world has not yet been discovered, so imprecise and even inconsistent formulations need to be used. If the imprecision merely concerned the values to be given to numerical constants, there wouldn't be great difficulty, but there is a need to use theories which are grossly wrong in general within domains where they are valid. The above-mentioned *history-as-a-sequcnce-of-situations* is such a theory. The sense in which this theory is an approximation to a more sophisticated theory hasn't been examined.

(McCarthy 1977c) discusses the need to use concepts that are meaningful only in an approximate theory. Relative to a Cartesian product co-ordinatization of situations, counterfactual sentences of the form "*// co-ordinate x had the value c and the other co-ordinates retained their values, then p would be true"* can be meaningful. Thus, within a suitable theory, the assertion *"The skier wouldn't have fallen if he had put his weight on his downhill ski"* is meaningful and perhaps true, but it is hard to give it meaning as a statement about the world of atoms and wave functions, because it is not clear what different wave functions are specified by *"if he had put his weight on his downhill ski".* We need an AI formalism that can use such statements but can go beyond them to the next level of approximation when possible and necessary. 1 now think that circumscription is a tool that will allow drawing conclusions from a given approximate theory for *use* in given circumstances without a total commitment to the theory.

5. One can imagine constructing programs either as empiricists or as realists. An empiricist program would build only theories connecting its sense data with its actions. A realist program would try to find facts about a world that existed independently of the program and would not suppose that the only reality is what might somehow interact with the program.

I favor building realist programs with the following example in mind. It has been shown that the Life two dimensional cellular automaton is universal as a computer and as a constructor. Therefore, there could be configurations of Life cells acting as self-reproducing computers with sensory and motor capabilities with respect to the rest of the Life plane. The program in such a computer could study the physics of its world by making theories and experiments to test them and might eventually come up with the theory that its fundamental physics is that of the Life cellular automaton.

We can test our theories of epistemology and common sense reasoning by asking if they would permit the Life-world computer to conclude, on the basis of experiments, that its physics was that of Life. If our epistemology isn't adequate for such a simple universe, it surely isn't good enough for our much more complicated universe This example is one of the reasons for preferring to build realist rather than empiricist programs. The empiricist program, if it was smart enough, would only end up with a statement that *"my experiences are best organized as if there were a Life cellular automaton and events isomorphic to my thoughts occurred in a certain subconfiguration of it".* Thus it would get a result equivalent to that of the realist program but more complicated and with less certainty.

More generally, we can imagine a *metaphilosophy* that has the same relation to philosophy that metamathematics has to mathematics. Metaphilosophy would study mathematical systems consisting of an "epistemologist" seeking knowledge in accordance with the epistemology to be tested and interacting with a "world". It would study what information about the world a given philosophy would obtain. This would depend also on the structure of the world and the "epistemologist"s opportunities to interact.

AI could benefit from building some very simple systems of this kind, and so might philosophy.

References

McCarthy, J. and Hayes, P.J. (1969) Some Philosophical Problems from the Standpoint of Artificial Intelligence. *Machine Intelligence* 4, pp. 463-502 (eds Meltzer, B. and Michie, D.). Edinburgh: Edinburgh University Press.

McCarthy, J. (1977a) Minimal Inference - A New Way of Jumping to Conclusions (to be published).

McCarthy, J. (1977b) First Order Theories of Individual Concepts (to be published).

McCarthy, J. (1977c) Ascribing Mental Qualities to Machines (to be published).

Moore, Robert C. (1977) Reasoning about Knowledge and Action, 1977 *IJCAI Proceedings.*