THE LOGIC AND PHILOSOPHY OF ARTIFICIAL INTELLIGENCE

John McCarthy

The goal of artificial intelligence (A.I.) is machines more capable than humans at solving problems and achieving goals requiring intelligence. There has been some useful success, but the ultimate goal still requires major conceptual advances and is probably far off.

There are three ways of attacking the goal. The first is to imitate the human nervous system. The second is to study the psychology of human intelligence. The third is to understand the common sense world in which people achieve their goals and develop intelligent computer programs. This last one is the computer science approach. It is the one that has had the most success so far, and it is the one I will discuss in this lecture.

Among A.I. researchers, opinions differ about how much to emphasize mathematical logical languages for expressing what the machine knows about the world. Briefly, the advantage is that facts can be learned independently of one another and of the use to which they will be put. The argument against it is that much human reasoning doesn't seem very logical. The current expert system technology uses logical languages for expressing much of their knowledge, but a lot is built into computer programs and also into the arrangement of the logical sentences. I will emphasize the logical approach in this lecture. As perhaps befits the occasion, I will emphasize my own work.

I became interested in artificial intelligence in September, 1948 when I attended some sessions of the Hixon Symposium on Cerebral Mechanisms in Behavior (Jeffress 1951) held at the California Institute of Technology where I was starting graduate work in mathematics. The symposium included the mathematician John Von Neumann (one of the inventors of the stored program computer), Warren McCulloch (who had shown how networks of hypothetical neurons could be used for computing) and many famous psychologists and neurophysiologists. Von Neumann's paper was entitled "The General and Logical Theory of Automata," and there was much interest in comparing what was known about the human brain with the new idea of an electronic computer. At that time, stored program computers were under construction, but the first wasn't finished until the following year.

When I recently re-examined the proceedings of the symposium, I discovered that my memory was incorrect and nothing had been said about trying to make intelligent computers. All the famous participants were fully committed to their existing research programs in biology, psychology and mathematics. Von Neumann discussed two ideas that he later developed more fully—how to make reliable machines out of unreliable components, and how to make self-reproducing machines. Although he had done important work in mathematical logic, he didn't discuss, then or later, representing facts about the world in logic. Although he was actively involved in developing computers, he didn't discuss programming them to behave intelligently. Nevertheless, the conference oriented my own scientific ambitions toward artificial intelligence.

While I started thinking about A.I. in 1948, I didn't think about programming computers or about representing facts in logic. I didn't know much about either logic or computers. Instead I successively pursued two ideas that I later came to regard as blind alleys. The first combined automata theory and information theory, both newly fashionable. It considered a brain as a finite automaton connected to an environment also considered as an automaton. To represent the fact that the brain is uncertain about what the environment is like, I considered an ensemble (i.e. a set with probabilities) of environment automata. Information theory applied to this ensemble permitted defining an entropy at time 0 when the brain was first attached to the environment and later when the system had run for a while and the state of the brain was partially dependent on which environment from the ensemble had been chosen. The difference of these entropies measured how much the brain had learned about the environment.

I came to believe that this was a bad idea and never tried to publish it. The trouble was that I couldn't see how to represent specific facts about our own world in terms of an ensemble of environments. To anticipate later terminology, the representation was epistemologically inadequate. It couldn't represent the information actually available.

My next bad idea was more mathematical. What is a problem, i.e. one that we might want an intelligent machine to solve? I proposed that a "well-defined problem" is given by a method for testing proposed solutions. That's a good idea and has survived in one form or another, but maybe it's obvious. The method of testing proposed that solutions have to be well defined, so let it be given by a Turing machine—the abstract form of computer introduced by Alan Turing in 1936. Therefore, problem solving may be regarded as the problem of inverting functions defined by Turing machines. I wrote a paper (McCarthy 1956) on this while working on a summer job with Claude Shannon in 1952, and it was included in our Automata Studies volume published as (Shannon and McCarthy 1956).The trouble is that this is also epistemologically inadequate.

That stored program computers were the right vehicle for developing artificial intelligence was still not apparent to me or to any of the others who contributed to *Automata Studies*.

The person to whom it was first apparent was Alan Turing. His (1950) paper "Computing Machinery and Intelligence" was in a philosophy journal and didn't become well known until it was reprinted by Edward R. Newman in his *World of Mathematics* in 1956. I had to come to the idea independently and so did Allen Newell and Herbert Simon. Newell was probably the first person to make programming computers to behave intelligently a career. Turing died in 1954, before computers were good enough to do much in the way of A.I..

My introduction to computers came in 1955, when Nathaniel Rochester of IBM hired me for the summer in his pioneering Information Research Department in Poughkeepsie. I learned to program the IBM 702, and re-oriented my thinking about A.I. to making intelligent computer programs. I was an assistant professor of mathematics at Dartmouth College then, and it occurred to me that summer that there was enough interest in A.I. to warrant a summer research project at Dartmouth in 1956 that would invite everyone we knew who might be interested. That wasn't too many, and we were able to invite a number of people that we only hoped would be interested. Claude Shannon, Marvin Minsky, Oliver Selfridge and Nathaniel Rochester joined me in making a proposal to the Rockefeller Foundation, and they awarded us $7,500 to pay for some travel and living expenses.

We had to decide what to call the subject, and I chose "artificial intelligence." The reason for choosing such a bold title was the desired effect on participants. When Shannon and I collected papers for *Automata Studies*, it seemed to me that we got too many papers treating the mathematical properties of automata that were only peripherally relevant to making intelligent automata. The name “artificial intelligence” nailed the flag to the mast. Many people have criticized the name, but I think it was and still is important to have a name that makes people compare their present projects with the ambitious long term goal.

To me the most interesting work presented at Dartmouth was the Newell-Simon “Logic Theory Machine.” Their program simulated the processes that naive student subjects went through in solving problems in elementary logic. In the course of this, they also developed the IPL list processing language and introduced recursive subroutines. Other interesting projects were Arthur Samuel's program for the game of checkers and Alex Bernstein's projected program for chess, both IBM projects. Marvin Minsky proposed a theorem prover for plane geometry that used a “diagram” in the memory of the computer to decide which sentences were plausible enough to try to prove. I had started thinking about the use of logic to represent facts, but I didn't have much to present at the Dartmouth meeting.

After the meeting, Rochester decided to implement Minsky's ideas of the geometry theorem prover and hired Herbert Gelernter to do it, with me as consultant. I had been impressed by the Newell-Simon idea of a list processing language, but their particular formalism didn't appeal to me at all; IBM's projected Fortran had a much more attractive style. Therefore, I proposed to Gelernter that he do his list processing for geometry in Fortran and proposed some basic functions—the *car* and *cdr* of LISP. Gelernter and Carl Gerberich did it, and added something more—making *cons* a function so it could be composed. (I don't have the space to describe the intellectual situation well enough to convince you that this wasn't the one obvious thing to do.)

I didn't become a full-time computer scientist until I went to MIT in Fall 1957. By then I was convinced that logic was the key formalism for artificial intelligence. While Newell and Simon (and several others soon after) had already written programs to do logic, none of them were interested in using logic except as a subject domain. My conversion to logic was expressed in my 1958 paper “Programs with Common Sense” (McCarthy 1959).

In 1958 I spent another summer with Rochester's IBM group, where Gelernter and Gerberich were finishing their plane geometry program. I became convinced that

Fortran didn't permit the expression of recursive list processing functions and started work on formulating LISP. When I returned to MIT in fall 1958, Minsky and I were given a big work room, a key punch, a secretary, two programmers and six mathematics graduate students. This was a very decisive and prompt action by Jerry Wiesner, considering that we had asked for these things the preceding spring in order to form an artificial intelligence project and hadn't even prepared a written proposal. Fortunately, the Research Laboratory of Electronics at MIT had just been given an open-ended Joint Services Contract by the U.S. Armed Forces and hadn't yet committed all the resources. I think that such flexibility is one of the reasons the U.S. started in A.I. ahead of other countries. The Newell-Simon work was also possible because of the flexible support the U.S. Air Force provided the Rand Corporation.

In September 1958 we started on LISP, and the first interpreter was working in early 1959. The first idea was just a Fortran-like language with list structures as data. However, a look at the logical structure of the process of differentiating algebraic expressions, as normally described to college freshmen, showed that that function could be described by a simple formula provided it could be used recursively, provided conditional expressions were allowed, and provided explicit erasure could be avoided by using garbage collection. I had invented conditional expressions the previous year in connection with a chess program written in Fortran. Lambda expressions were another tool whose use was suggested by the differentiation example. In order to show mathematicians that a universal computer could be described much more compactly in LISP than with Turing machines, I wrote the function EVAL. Steve Russell, one of our programmers, who promptly turned it into an interpreter for LISP (McCarthy 1977a), describes this history.

Early in 1979 James Slagle started his work on symbolic integration under Minsky's supervision. About this time, the promising IBM A.I. work was shut down, and IBM didn't really resume A.I. work until 1983 and still plays a minor role. The reason for shutting it down was apparently a combination of uninformed scientific criticism and public relations. IBM wanted computers to have the image of mere data processors—nothing revolutionary was wanted.

The work in A.I. also led to my first proposals for time-sharing computer systems. Early ideas about computers emphasized running a program for a long time.

The programs were imagined to be numerical. The process of developing the program was considered auxiliary and so was any interaction with it. It seemed to me that artificial intelligence required a quite different approach. Someone doing A.I. research might spend almost all of his time developing the program and interacting with it. He needed to sit at a terminal in his own office and interact at his own pace and convenience, rather than sign up for half an hour at 2 a.m. or submit decks of cards. Again it would require a lot of explanation today to convince you that the idea wasn't obvious. Not only wasn't it obvious, but it encountered considerable resistance. For example, the IBM developers of the 360 knew about the idea but didn't believe it. Fortunately, the Digital Equipment Corporation developers of the PDP-6 computer (somewhat later) had been MIT students and did time-sharing from the start.

Actually there were two approaches to providing on-line computation. One was time-sharing, in which a large computer switched its time among users, and the other was the personal computer, pioneered at MIT Lincoln Laboratories by Wes Clark, who created the TX-0 and TX-2 computers, both extremely expensive machines intended to be used on-line by a single user. Personal computers didn't become economical until the 1970s, and there is still a conflict between providing on-line service to many users and providing expensive personal computers to a few users.

Logic in Artificial Intelligence

The 1959 “Programs with Common Sense” 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 A.I. are better understood by researchers today than in 1958. Expressing information in declarative sentences is far more flexible than expressing it in segments of a 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 1958, 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 formalism given in “McCarthy 1959” was just a sketch of a theory of the achievement of goals by sequences of actions. It took a more definite form in the *situation calculus* of a 1964 Stanford report and was published in “McCarthy and Hayes 1969.” The basic idea is to use the formula

$$s' = result(e,s)$$

to represent the new situation s' that results when the event e occurs in situation s. The events most studied are actions, and there are usually conditions that s has to satisfy before we can infer much about s. The situation calculus embodies a special case of reasoning about actions and other events. First, the events can be regarded as discrete; they occur in one situation and result in another, and we don't need to reason about what happens during the event. Second, we consider only one event occurring at a time; concurrent events are not analyzed.

One feature of situations was emphasized conceptually but didn't play much role in the actual axiomatizations. Situations were regarded as infinitely detailed, e.g. a block on the table was in a particular location and had a detailed shape and distribution of material, perhaps down to an atomic level. Therefore, the formalism did not provide for knowing a situation exactly but only for knowing facts about a situation that partially characterized it. In this respect situation calculus differed essentially from the mathematical model used in physics and discussed in most philosophy of science, e.g. in gravitational astronomy. In physics models, it is customary to decide what planets are to be taken into account and whether to represent them as mass points or whether to consider (say) some moments of their mass distributions. In contrast to this, situation calculus was intended to provide a model to which new detail could be added at any time. I contend that this open-endedness is an essential characteristic of the common sense information situation, whether the reasoning is done by people or by machines. I now refer to entities like situations that have infinite detail as rich entities and contrast them with discrete entities that can be completely described. Reasoning involves using discrete entities to approximate rich entities, but the level of approximation can change during a reasoning process.

The situation calculus was used in Cordell Green's (1969) Ph.D. thesis along with resolution theorem proving and a method of “answer extraction” he devised. However, he and his colleagues at SRI found their theorem prover did too much search to be practical. My opinion was this was because they didn't have any way of using heuristic facts to control the search, but I didn't have a proposal of how to do it. I have more ideas about that now, but they still don't amount to a definite proposal for controlling reasoning with facts.

In this event, Fikes and Nilsson (1971) went to a restricted formalism called

STRIPS in which resolution theorem proving was used for reasoning about properties of single situations, whereas going from one situation to the next was done by a program that interpreted the action descriptions directly. The consequence was a faster program, but it didn't allow sentences that involved more than one situation. Because the control problem still isn't solved, the practical A. I. systems that reason about actions all use restricted languages.

In the late 1970s' the introduction of formalized nonmonotonic reasoning revolutionized the use of logic in A.I. (McCarthy 1977b, 1980, 1986), (Reiter 1980), (McDermott and Doyle 1980).Traditional logic is monotonic in the following sense. If a sentence *p* is inferred from a collection *A* of sentences, and *B* is a more inclusive set of sentences (symbolically $A \subset B$), then *p* can be inferred from *B*.

If the inference is a 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 corresponds to that of traditional logic, some important human common sense reasoning is not monotonic. We reach conclusions from certain premisses that we would not reach if certain other sentences were included in our premisses. For example, learning that I own a car, you conclude that it is appropriate on a certain occasion to ask me for a ride, but when you learn the further fact that the car is in the garage being fixed 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 asking for a ride from car owners but a probabilistic rule. So far it has not proved possible to try to work out the detailed epistemology of this approach, i.e. exactly what probabilistic sentences should be used. In fact, it seems that the probabilistic reason would end up using nonmonotonic techniques. Anyway, A.I. has moved to directly formalizing nonmonotonic logical reasoning.

Formalized nonmonotonic reasoning is under rapid development and many kinds of systems have been proposed. I shall concentrate on an approach called circumscription. It has met with wide acceptance and is currently the most actively pursued. The idea is to single out among the models of the collection of sentences being assumed some “preferred” or “standard” models. The preferred models are those that satisfy a certain minimum principle. What should be minimized is not yet decided in complete generality and may be problem dependent. However, many domains that have been studied yield quite general theories using minimizations of abnormality or of the set of some kind of entity. The idea is not completely unfamiliar. For example, Ockham's razor, “Do not multiply entities beyond necessity,” leads to such minimum principles if one tries to formalize it in logic.

Minimization in logic is another example of an area of mathematics being discovered in connection with applications rather than via the normal internal development of mathematics. Of course, the reverse is happening on an even larger scale; many logical concepts developed for purely mathematical reasons turn out to have A.I. importance.

As a more concrete example of nonmonotonic reasoning, consider the conditions under which a boat may be used to cross a river. We all know of certain things that might be wrong with a boat, e.g. a leak, no oars or motor or sails, depending on what kind of a boat it is. It would be reasonably convenient to list some of them in a set of axioms. However, besides those we can expect to list in advance, human reasoning will admit still others, should they arise, but we cannot be expected to think of them in advance, e.g. a fence down in the middle of the river. This is handled using circumscription by minimizing the set of “things preventing the boat from crossing the river,” i.e. the set of obstacles to be overcome. If the reasoner knows of none in a particular case, he or it will conjecture that the boat can be used, but if he learns of one, he will get a different result when he minimizes.

This illustrates the fact that non-monotonic reasoning is conjectural rather than rigorous. Indeed it has been shown that certain mathematical logical systems cannot be rigorously extended, i.e. that they have a certain kind of completeness.

It is as misleading to conduct a discussion of this kind entirely without formulas as it would be to discuss the foundations of physics without formulas. Unfortunately, many people are unaware of this fact. Therefore, we present a formalization by Vladimir Lifschitz (1987) of a simple example called “The Yale shooting problem.” Drew McDermott (1987), who has become discouraged about the use of logic in A.I. and especially about the non-monotonic formalisms, invented it as a challenge. (The formal part is only one page, however). Some earlier styles of axiomatizing facts about change didn't work right on this problem. Lifschitz's method works well here, but I think it will require further modification.

In an initial situation there is an unloaded gun and a person named Fred. The gun is loaded, then there is a wait, and then the gun is pointed at Fred and fired. The desired conclusion is the death of Fred. Informally, the rules are (1) that a living person remains alive until something happens to him, (2) that loading causes a gun to become loaded, (3) that a loaded gun remains loaded until something unloads it, (4) that shooting unloads a gun and (5) that shooting a loaded gun at a person kills him. We are intended to reason as follows. Fred will remain alive until the gun is fired, because nothing can be inferred to happen to him. The gun will remain loaded until it is fired, because nothing can be inferred to happen to it. Fred will then die when the gun is fired. The non-monotonic part of the reasoning is minimizing “the things that happen” or assuming that “nothing happens without a reason.”

The logical sentences are intended to express the above 5 premisses, but they don't explicitly say that no other phenomenon occurs. For example, it isn't asserted that Fred isn't wearing a bulletproof vest, nor are any properties of bulletproof vests mentioned. Nevertheless, a human will conclude that unless some unmentioned aspect of the situation is present, Fred will die. The difficulty is that the sentences admit an *unintended minimal model*, to use the terminology of mathematical logic. Namely, it might happen for some unspecified reason the gun becomes unloaded during the wait, so that Fred remains alive. The way nonmonotonic formalisms, e.g. circumscription and Reiter's logic of defaults, were previously used to formulate the problem, minimizing “abnormality” results in two possiblities, not one. The unintended possibility is that the gun mysteriously becomes unloaded.

Lifschitz's Cauality Axioms for the Yale Shooting Problem

Lifschitz's axioms use the situation calculus but introduce a predicate *causes* as an undefined notion.

We quote from (Lifschitz 1987).

"Our axioms for the shooting problem can be classified into three groups. The first group describes the initial situation:

$$holds(alive, SO), \qquad (Y\,1.\,1)$$

$$\neg\, holds,(loaded, SO) \qquad (Y\,1.\,2)$$

The second group tells us how the fluents are affected by actions:

$$causes(load,\ loaded,\ true), \qquad (Y\,2.\,1)$$

$$causes(shoot,\ loaded,\ false), \qquad (Y\,2.\,2)$$

$$causes(shoot,\ alive,\ false). \qquad (Y\,2.\,3)$$

These axioms describe the effects of *successfully performed* actions; they do not say *when* an action can be successful. This information is supplied separately:

$$precond(loaded,shoot) \qquad (Y\,2.\,4)$$

The last group consists of two axioms of a more general nature. We use the abbreviations:

$$success(a,s) \equiv A\, f(precond(f,a) \supset holds(f,s)),$$
$$affects(a,f,s) \equiv success(a,s) \wedge \exists v\ causes(a,f,v)$$

One axiom describes how the value of a fluent changes after an action affecting this fluent is carried out:

$$success(a,s) \wedge causes(a,f,v) \supset (holds(f,\ result(a,s)) \equiv v = (true). \qquad (Y\,3.\,1)$$

(Recall that *v* can take on two values here, *true* and *false*; the equivalence in *Y3. 1* reduces to *holds (f, result(a,s))* in the first case and to the negation of this formula in the second.) If the fluent is not affected then its value remains the same:

$$\neg affects(a,f,s) \supset (holds(f,result(a,s)) \equiv holds(f,s)) \qquad (Y\,3.\,2)"$$

Minimizing *causes* and *precond* makes the right thing happen. While these axioms and *circumscription policy* solve this problem, it remains to be seen whether we can write a large body of common sense knowledge in the formalism without getting other unpleasant surprises. Another current question is whether we can get by with axioms about the external world only or whether the axioms must contain information about the purposes of the reasoning in order to determine the preferred models. Moreover, there are many more possibilities to explore for the formal minimum principle required for common sense reasoning.

Conclusions and Remarks

1. Progress in using logic to express facts about the world has always been slow. Aristotle didn't invent any formalisms. Leibniz, who wanted to replace argument by calculation in human affairs, didn't invent propositional calculus, although it is technically far easier than the infinitesimal calculus of which he was a co-inventor with Newton. Boole, who invented propositional calculus, and who called his book, "*The Laws of Thought*" didn't invent predicate calculus. Frege and his successors saw no need or possibility of formalizing non-monotonic reasoning. It seems to me that almost any of these ideas would have been accepted by preceding innovators, e.g. Leibniz would have accepted propositional calculus, had it been suggested. Therefore, we must conclude that we humans find it difficult to formulate many facts about our thought processes that are apparent when suggested.

Even with formalized non-monotonic reasoning there are obstacles to expressing the general facts about the common sense world in an epistemologically adequate and elaboration tolerant way. I take this as a sign that major innovations are yet to come, and I will have some suggestions in my more technical lecture.

I have always wondered why science doesn't progress more rapidly than it does. Progress in some sciences is limited by instrumentation and experimental technique. For example, this is true of molecular biology. In other sciences, progress is limited by mistaken ideas that direct attention away from the ideas that lead to progress. When this occurs, it is often hard to say why a discovery wasn't made 20 or more years earlier than it was. For example, there is no technical reason why non-monotonic reasoning couldn't have been formalized in the 1920s.

2. Artificial intelligence research took form in the late 1950s in ways that were quite different from the ideas of senior scientists who considered the problem. I, and I think most other A.I. researchers avoided our seniors, rather than either following them or opposing them. This was possible, because it turned out that we were not dependent on them either for research support or academic position. I suspect this would not be good advice in general. However, it seems to have turned out well in artificial intelligence in that the ideas related to A.I. proposed by most of them—Von Neumann, Wiener and McCulloch have not so far been fruitful. In my opinion, only Turing proposed the line of research that has given us the main results in A.I. achieved so far. Of course, the older proposals have not been refuted, and maybe new people will find them fruitful. Connectionism might be regarded as one such gamble.

3. My work in A.I. has been in accordance with a certain philosophical point of view. To what extent the philosophy has influenced the research is harder to say. My subjective impression is philosophy has been important.

a. Our human knowledge of the world does not consist of knowledge of sense data or even summaries of it. Material objects and human purposes, for example, are far more stable than our sense impressions of them. This isn't peculiar to the human situation; it will also be true for any robots we might build. It is possible that babies learn from their experience that material objects exist, but it's also possible that this organization of the world is in our genes. Animals that presume material objects and some other things are likely to have an evolutionary advantage. From the A.I. point of view, this suggests that we build into our programs presumptions of structures that exist in the real world. If we are going to use logical sentences to represent knowledge, then there should be variables ranging over material objects, and also variables ranging over other entities like actions and beliefs.

This may seem obvious, but it is in contrast with some views. For example, some people have proposed that since the experience of a human or robot is representable by a sequence of sense impressions, then intelligence might consist of the ability to predict the future of a sequence from its value up to a given point. This presumes nothing about the world and therefore might be preferred by an extremely cautious philosopher. Efforts to program computers to predict sequences have not been informative, nothing that wasn't obvious resulted from the experiments, and the experimenters were at a loss for how to proceed further.

I suppose the positivistic emphasis on sense data is a residue of the early 20th century reaction against 19th century idealistic philosophy. This philosophy involved many vague postulated entities, and the reaction against it took the form of admitting only the most directly observable entities as basic.

b. Our human ascription of mental qualities, e.g. beliefs, to each other is warranted by the usefulness of the ascriptions in understanding, predicting and affecting other people's behavior. We don't know whether the tendency to do this is genetic, but one explanation of autistic children might be a failure of such a mechanism to develop normally. It is legitimate to ascribe mental qualities to any system where it helps explain its behavior. This idea is developed as philosophy in “Dennett 1971 and 1987”and for A.I. in “Newell 1980”and“McCarthy 1979a.” It still isn't generally accepted in philosophy.

c. Many qualities, including especially mental qualities, that we ascribe to various systems are meaningful only in terms of approximate theories that enable us to understand aspects of phenomena but which cannot be given precise definitions in terms of the state of the world. See (McCarthy 1979a).

d. Today artificial intelligence is far from being able to produce systems of human capability in general reasoning. This has led to a modesty that one might also recommend to philosophers. An A.I. system involving some concept like *belief* can't purport to use “the true” completely general concept of belief, because there is no agreement about that. Indeed at the level of precision required for computational implementation, there aren't even any candidates. Therefore, A.I. has to get by with limited, approximate concepts. But maybe that's all humans have either. Maybe the philosophers' attempts to understand belief in general merely force them to construct a concept rather than discover it. Maybe there is no completely general concept of belief that corresponds to what humans actually do or what computers can be programmed to do.

References:

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

Dennett, D.C. (1987): *The Intentional Stance*, MIT Press.

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.

Green. C, (1969): "Application of Theorem Proving to Problem Solving," in IJCAI-1, pp. 219-239.

Jeffress, Lloyd A. (ed.) (1951): *Cerebral Mechanisms in Behavior : The Hixon Symposium,* Wiley.

Lifschitz, Vladimir (1987): "Formal Theories of Action", in F.M Brown (ed.), *The Frame Problem in Artificial Intelligence,* Morgan Kaufmann, pp. 35-58. Reprinted in M.L Ginsberg(ed.), *Readings in Nonmonotonic Reasoning,* Morgan-Kaufmann, pp. 410-432.

McCarthy, John (1956): "The Inversion of Functions Defined by Turing Machines," in *Automata Studies, Annals of Mathematical Study, No. 34,* Princeton, pp. 177-181.

McCarthy, John (1959):"Programs with Common Sense", in *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,” in D. Michie (ed), *Machine Intelligence 4,* American Elsevier, New York, NY.
McCarthy, John (1977a): “History of LISP,” in Proceedings of the ACM Conference on the History of Programming Languages, Los Angeles.

McCarthy, John (1977b): “Epistemological Problems of Artificial Intelligence,” *Proceedings of the Fifth International Joint Conference on Artificial Intelligence,* MIT, Cambridge, Mass.

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

McCarthy, John (1979b): “First Order Theories of Individual Concepts and Propositions,” in 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 (1982): “Common Business Communication Language,” in *Textverarbeitung und Bürosysteme,* Albert Endres and Jürgen Reetz (eds), R. Oldenbourg Verlag, Munich and Vienna, 1982.

McCarthy, John (1983): “Some Expert Systems Need Common Sense,” in *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): "Generality in Artificial Intelligence," *Communications of the ACM.* Vol.30, No.12, pp.1030-1035.

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

McDermott, D. and J. Doyle (1980): "Non-Monotonic Logic I," *Artificial Intelligence,* Vol.13, No.1.

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

Reiter, R.A. (1980): "A Logic for default reasoning," *Artificial Intelligence,* 13 (1, 2), pp. 81-132.

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

Shannon, Claude and John McCarthy (eds.) (1956): *Automata Studies,* Annals of Mathematics Study 34, Princeton University Press.

Turing, A.M. (1950): "Computing Machinery and Intelligence," *Mind* **59**, pp. 433-460.