SOME PHILOSOPHICAL PROBLEMS FROM THE STANDPOINT OF ARTIFICIAL INTELLIGENCE

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

1969

1 Introduction

A computer program capable of acting intelligently in the world must have a general representation of the world in terms of which its inputs are interpreted. Designing such a program requires commitments about what knowledge is and how it is obtained. Thus, some of the major traditional problems of philosophy arise in artificial intelligence.

More specifically, we want a computer program that decides what to do by inferring in a formal language that a certain strategy will achieve its assigned goal. This requires formalizing concepts of causality, ability, and knowledge. Such formalisms are also considered in philosophical logic.

The first part of the paper begins with a philosophical point of view that seems to arise naturally once we take seriously the idea of actually making an intelligent machine. We go on to the notions of metaphysically and epistemologically adequate representations of the world and then to an explanation of *can, causes,* and *knows* in terms of a representation of the world by a system of interacting automata. A proposed resolution of the problem of freewill in a deterministic universe and of counterfactual conditional sentences is presented.

The second part is mainly concerned with formalisms within which it can be proved that a strategy will achieve a goal. Concepts of situation, fluent, future operator, action, strategy, result of a strategy and knowledge are formalized. A method is given of constructing a sentence of first order logic which will be true in all models of certain axioms if and only if a certain strategy will achieve a certain goal.

The formalism of this paper represents an advance over McCarthy (1963) and Green (1969) in that it permits proof of the correctness of strategies that contain loops and strategies that involve the acquisition of knowledge, and it is also somewhat more concise.

The third part discusses open problems in extending the formalism of Part two (section 3).

The fourth part is a review of work in philosophical logic in relation to problems of artificial intelligence and a discussion of previous efforts to program 'general intelligence' from the point of view of this paper.

2 PHILOSOPHICAL QUESTIONS

2.1 Why Artificial Intelligence Needs Philosophy

The idea of an intelligent machine is old, but serious work on the artificial intelligence problem or even serious understanding of what the problem is awaited the stored program computer. We may regard the subject of artificial intelligence as beginning with Turing's article *Computing Machinery and Intelligence* (Turing 1950) and with Shannon's (1950) discussion of how a machine might be programmed to play chess.

Since that time, progress in artificial intelligence has been mainly along the following lines. Programs have been written to solve a class of problems that give humans intellectual difficulty: examples are playing chess or checkers, proving mathematical theorems, transforming one symbolic expression into another by given rules, integrating expressions composed of elementary functions, determining chemical compounds consistent with mass-spectrographic and other data. In the course of designing these programs intellectual mechanisms of greater or lesser generality are identified sometimes by introspection, sometimes by mathematical analysis, and sometimes by experiments with human subjects. Testing the programs sometimes leads to better understanding of the intellectual mechanisms and the identification of new ones.

An alternative approach is to start with the intellectual mechanisms (for example, memory, decision-making by comparisons of scores made up of weighted sums of sub-criteria, learning, tree-search, extrapolation) and make up problems that exercise these mechanisms.

In our opinion the best of this work has led to increased understanding of intellectual mechanisms and this is essential for the development of artificial intelligence even though few investigators have tried to place their particular mechanism in the general context of artificial intelligence. Sometimes this is because the investigator identifies his particular problem with the field as a whole; he thinks he sees the woods when in fact he is looking at a tree. An old but not yet superseded discussion on intellectual mechanisms is in Minsky (1961); see also Newell's (1965) review of the state of artificial intelligence.

There have been several attempts to design intelligence with the same kind of flexibility as that of a human. This has meant different things to different investigators, but none has met with much success even in the sense of general intelligence used by the investigator in question. Since our criticism of this work will be that it does not face the philosophical problems discussed in this paper, we shall postpone discussing it until a concluding section. However, we are obliged at this point to present our notion of general intelligence.

It is not difficult to give sufficient conditions for general intelligence. Turing's idea that the machine should successfully pretend to a sophisticated observer to be a human being for half an hour will do. However, if we direct our efforts towards such a goal our attention is distracted by certain superficial aspects of human behaviour that have to be imitated. Turing excluded some of these by specifying that the human to be imitated is at the end of a teletype line, so that voice, appearance, smell, etc., do not have to be considered. Turing did allow himself to be distracted into discussing the imitation of human fallibility in arithmetic, laziness, and the ability to use the English language.

However, work on artificial intelligence, especially general intelligence, will be improved by a clearer idea of what intelligence is. One way is to give a purely behavioural or black-box definition. In this case we have to say that a machine is intelligent if it solves certain classes of problems requiring intelligence in humans, or survives in an intellectually demanding environment. This definition seems vague; perhaps it can be made somewhat more precise without departing from behavioural terms, but we shall not try to do so.

Instead, we shall use in our definition certain structures apparent to introspection, such as knowledge of facts. The risk is twofold: in the first place we might be mistaken in our introspective views of our own mental structure; we may only think we use facts. In the second place there might be entities which satisfy behaviourist criteria of intelligence but are not organized in this way. However, we regard the construction of intelligent machines as fact manipulators as being the best bet both for constructing artificial intelligence and understanding natural intelligence.

We shall, therefore, be interested in an intelligent entity that is equipped with a representation or model of the world. On the basis of this representation a certain class of internally posed questions can be answered, not always correctly. Such questions are

1. What will happen next in a certain aspect of the situation?
2. What will happen if I do a certain action?
3. What is $3 + 3$?
4. What does he want?
5. Can I figure out how to do this or must I get information from someone else or something else?

The above are not a fully representative set of questions and we do not have such a set yet.

On this basis we shall say that an entity is intelligent if it has an adequate model of the world (including the intellectual world of mathematics, understanding of its own goals and other mental processes), if it is clever enough to answer a wide variety of questions on the basis of this model, if it can get additional information from the external world when required, and can perform such tasks in the external world as its goals demand and its physical abilities permit.

According to this definition intelligence has two parts, which we shall call the epistemological and the heuristic. The epistemological part is the representation of the world in such a form that the solution of problems follows from the facts expressed in the representation. The heuristic part is the mechanism that on the basis of the information solves the problem and decides what to do. Most of the work in artificial intelligence so far can be regarded as devoted to the heuristic part of the problem. This paper, however, is entirely devoted to the epistemological part.

Given this notion of intelligence the following kinds of problems arise in constructing the epistemological part of an artificial intelligence:

1. What kind of general representation of the world will allow the incorporation of specific observations and new scientific laws as they are discovered?
2. Besides the representation of the physical world what other kinds of entities have to be provided for? For example, mathematical systems, goals, states of knowledge.
3. How are observations to be used to get knowledge about the world, and how are the other kinds of knowledge to be obtained? In particular what kinds of knowledge about the system's own state of mind are to be provided for?
4. In what kind of internal notation is the system's knowledge to be expressed?

These questions are identical with or at least correspond to some traditional questions of philosophy, especially in metaphysics, epistemology and philosophic logic. Therefore, it is important for the research worker in artificial intelligence to consider what the philosophers have had to say.

Since the philosophers have not really come to an agreement in 2500 years it might seem that artificial intelligence is in a rather hopeless state if it is to depend on getting concrete enough information out of philosophy to write computer programs. Fortunately, merely undertaking to embody the philosophy in a computer program involves making enough philosophical presuppositions to exclude most philosophy as irrelevant. Undertaking to construct a general intelligent computer program seems to entail the following presuppositions:

1. The physical world exists and already contains some intelligent machines called people.

2. Information about this world is obtainable through the senses and is expressible internally.

3. Our common-sense view of the world is approximately correct and so is our scientific view.

4. The right way to think about the general problems of metaphysics and epistemology is not to attempt to clear one's own mind of all knowledge and start with 'Cogito ergo sum' and build up from there. Instead, we propose to use all of our knowledge to construct a computer program that knows. The correctness of our philosophical system will be tested by numerous comparisons between the beliefs of the program and our own observations and knowledge. (This point of view corresponds to the presently dominant attitude towards the foundations of mathematics. We study the structure of mathematical systems—from the outside as it were—using whatever metamathematical tools seem useful instead of assuming as little as possible and building up axiom by axiom and rule by rule within a system.)

5. We must undertake to construct a rather comprehensive philosophical system, contrary to the present tendency to study problems separately and not try to put the results together.

6. The criterion for definiteness of the system becomes much stronger. Unless, for example, a system of epistemology allows us, at least in principle, to construct a computer program to seek knowledge in accordance with it, it must be rejected as too vague.

7. The problem of 'free will' assumes an acute but concrete form. Namely, in common-sense reasoning, a person often decides what to do by evaluating the results of the different actions he can do. An intelligent program must use this same process, but using an exact formal sense of *can*, must be able to show that it has these alternatives without denying that it is a deterministic machine.

8. The first task is to define even a naive, common-sense view of the world precisely enough to program a computer to act accordingly. This is a very difficult task in itself.

We must mention that there is one possible way of getting an artificial intelligence without having to understand it or solve the related philosophical problems. This is to make a computer simulation of natural selection in which intelligence evolves by mutating computer programs in a suitably demanding environment. This method has had no substantial success so far, perhaps due to inadequate models of the world and of the evolutionary process, but it might succeed. It would seem to be a dangerous procedure, for a program that was intelligent in a way its designer did not understand might get out of control. In any case, the approach of trying to make an artificial intelligence through understanding what intelligence is, is more congenial to the present authors and seems likely to succeed sooner.

2.2 Reasoning programs and the Missouri program

The philosophical problems that have to be solved will be clearer in connection with a particular kind of proposed intelligent program, called a reasoning program or RP for short. RP interacts with the world through input and output devices some of which may be general sensory and motor organs (for example, television cameras, microphones, artificial arms) and others of which are communication devices (for example, teletypes or keyboard-display consoles). Internally, RP may represent information in a variety of ways. For example, pictures may be represented as dot arrays or a list of regions and edges with classifications and adjacency relations. Scenes may be represented as lists of bodies with positions, shapes, and rates of motion. Situations may be represented by symbolic expressions with allowed rules of transformation. Utterances may be represented by digitized functions of time, by sequences of phonemes, and parsings of sentences.

However, one representation plays a dominant role and in simpler systems may be the only representation present. This is a representation by sets of sentences in a suitable formal logical language, for example ω-order logic with function symbols, description operator, conditional expressions, sets, etc. Whether we must include modal operators with their referential opacity is undecided. This representation dominates in the following sense:

1. All other data structures have linguistic descriptions that give the relations between the structures and what they tell about the world.

2. The subroutines have linguistic descriptions that tell what they do, either internally manipulating data, or externally manipulating the world.

3. The rules that express RP's beliefs about how the world behaves and that give the consequences of strategies are expressed linguistically.

4. RP's goals, as given by the experimenter, its devised subgoals, its opinion on its state of progress are all linguistically expressed.

5. We shall say that RP's information is adequate to solve a problem if it is a logical consequence of all these sentences that a certain strategy of

action will solve it.

6. RP is a deduction program that tries to find strategies of action that it can prove will solve a problem; on finding one, it executes it.

7. Strategies may involve subgoals which are to be solved by RP, and part or all of a strategy may be purely intellectual, that is, may involve the search for a strategy, a proof, or some other intellectual object that satisfies some criteria.

Such a program was first discussed in McCarthy (1959) and was called the Advice Taker. In McCarthy (1963) a preliminary approach to the required formalism, now superseded by this paper, was presented. This paper is in part an answer to Y. Bar-Hillel's comment, when the original paper was presented at the 1958 Symposium on the Mechanization of Thought Processes, that the paper involved some philosophical presuppositions.

Constructing RP involves both the epistemological and the heuristic parts of the artificial intelligence problem: that is, the information in memory must be adequate to determine a strategy for achieving the goal (this strategy may involve the acquisition of further information) and RP must be clever enough to find the strategy and the proof of its correctness. Of course, these problems interact, but since this paper is focused on the epistemological part, we mention the Missouri program (MP) that involves only this part.

The Missouri program (its motto is, 'Show me') does not try to find strategies or proofs that the strategies achieve a goal. Instead, it allows the experimenter to present it proof steps and checks their correctness. Moreover, when it is 'convinced' that it ought to perform an action or execute a strategy it does so. We may regard this paper as being concerned with the construction of a Missouri program that can be persuaded to achieve goals.

2.3 Representations of the world

The first step in the design of RP or MP is to decide what structure the world is to be regarded as having, and how information about the world and its laws of change are to be represented in the machine. This decision turns out to depend on whether one is talking about the expression of general laws or specific facts. Thus, our understanding of gas dynamics depends on the representation of a gas as a very large number of particles moving in space; this representation plays an essential role in deriving the mechanical, thermal electrical and optical properties of gases. The state of the gas at a given instant is regarded as determined by the position, velocity and excitation states of each particle. However, we never actually determine the position, velocity or excitation of even a single molecule. Our practical knowledge of a particular sample of gas is expressed by parameters like the pressure, temperature and velocity fields or even more grossly by average pressures and temperatures. From our philosophical point of view this is entirely normal, and we are not inclined to deny existence to entities we cannot see, or to be so anthropocentric as to imagine that the world must be so constructed that we have direct or even indirect access to all of it.

From the artificial intelligence point of view we can then define three kinds of adequacy for representations of the world.

A representation is called metaphysically adequate if the world could have that form without contradicting the facts of the aspect of reality that interests us. Examples of metaphysically adequate representations for different aspects of reality are:

1. The representation of the world as a collection of particles interacting through forces between each pair of particles.
2. Representation of the world as a giant quantum-mechanical wave function.
3. Representation as a system of interacting discrete automata. We shall make use of this representation.

Metaphysically adequate representations are mainly useful for constructing general theories. Deriving observable consequences from the theory is a further step.

A representation is called epistemologically adequate for a person or machine if it can be used practically to express the facts that one actually has about the aspect of the world. Thus none of the above-mentioned representations are adequate to express facts like 'John is at home' or 'dogs chase cats' or 'John's telephone number is 321-7580'. Ordinary language is obviously adequate to express the facts that people communicate to each other in ordinary language. It is not, for instance, adequate to express what people know about how to recognize a particular face. The second part of this paper is concerned with an epistemologically adequate formal representation of common-sense facts of causality, ability and knowledge.

A representation is called heuristically adequate if the reasoning processes actually gone through in solving a problem are expressible in the language.

We shall not treat this somewhat tentatively proposed concept further in this paper except to point out later that one particular representation seems epistemologically but not heuristically adequate.

In the remaining sections of the first part of the paper we shall use the representations of the world as a system of interacting automata to explicate notions of causality, ability and knowledge (including self-knowledge).

2.4 The automaton representation and the notion of 'can'

Let S be a system of interacting discrete finite automata such as that shown in Figure 1.

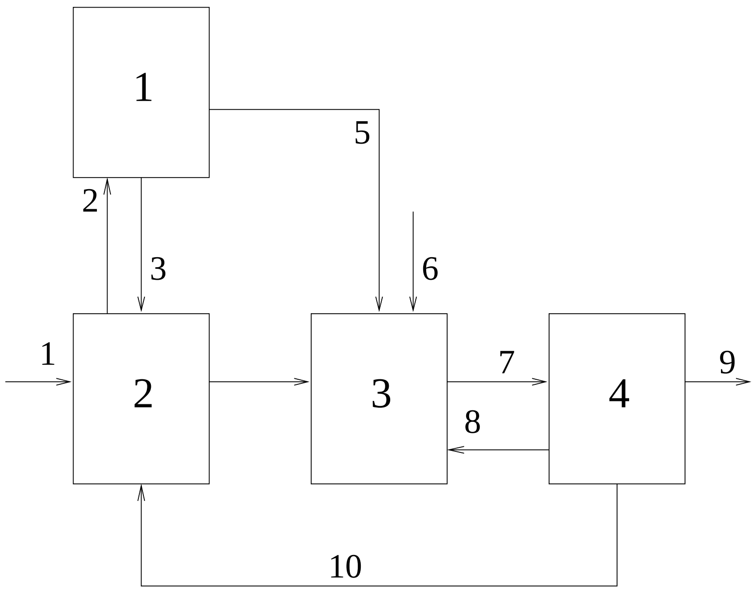

Figure 1

Each box represents a subautomaton and each line represents a signal. Time takes on integer values and the dynamic behaviour of the whole automaton is given by the equations:

$$
\begin{array}{lrcl}
(1) & a_1(t+1) & = & A_1(a_1(t),\, s_2(t)) \\
& a_2(t+1) & = & A_2(a_2(t),\, s_1(t),\, s_3(t),\, s_{10}(t)) \\
& a_3(t+1) & = & A_3(a_3(t),\, s_4(t),\, s_5(t),\, s_6(t),\, s_8(t)) \\
& a_4(t+1) & = & A_4(a_4(t),\, s_7(t)) \\
\\
(2) & s_2(t) & = & S_2(a_2(t)) \\
& s_3(t) & = & S_3(a_1(t)) \\
& s_4(t) & = & S_4(a_2(t)) \\
& s_5(t) & = & S_5(a_1(t)) \\
& s_7(t) & = & S_7(a_3(t)) \\
& s_8(t) & = & S_8(a_4(t)) \\
& s_9(t) & = & S_9(a_4(t)) \\
& s_{10}(t) & = & S_{10}(a_4(t))
\end{array} \qquad (1)
$$

The interpretation of these equations is that the state of any automaton at time $t+1$ is determined by its state at time t and by the signals received at time t. The value of a particular signal at time t is determined by the state at time t of the automaton from which it comes. Signals without a source automaton represent inputs from the outside and signals without a destination represent outputs.

Finite automata are the simplest examples of systems that interact over time. They are completely deterministic; if we know the initial states of all the automata and if we know the inputs as a function of time, the behaviour of the system is completely determined by equations (1) and (2) for all future time.

The automaton representation consists in regarding the world as a system of interacting subautomata. For example, we might regard each person in the room as a subautomaton and the environment as consisting of one or more additional subautomata. As we shall see, this representation has many of the qualitative properties of interactions among things and persons. However, if we take the representation too seriously and attempt to represent particular situations by systems of interacting automata, we encounter the following difficulties:

1. The number of states required in the subautomata is very large, for example $2^{10^{10}}$, if we try to represent someone's knowledge. Automata this large have to be represented by computer programs, or in some other way that does not involve mentioning states individually.

2. Geometric information is hard to represent. Consider, for example, the location of a multi-jointed object such as a person or a matter of even more difficulty—the shape of a lump of clay.

3. The system of fixed interconnections is inadequate. Since a person may handle any object in the room, an adequate automaton representation would require signal lines connecting him with every object.

4. The most serious objection, however, is that (in our terminology) the automaton representation is epistemologically inadequate. Namely, we do not ever know a person well enough to list his internal states. The kind of information we do have about him needs to be expressed in some other way.

Nevertheless, we may use the automaton representation for concepts of *can, causes,* some kinds of counterfactual statements ('If I had struck this match yesterday it would have lit') and, with some elaboration of the representation, for a concept of *believes.*

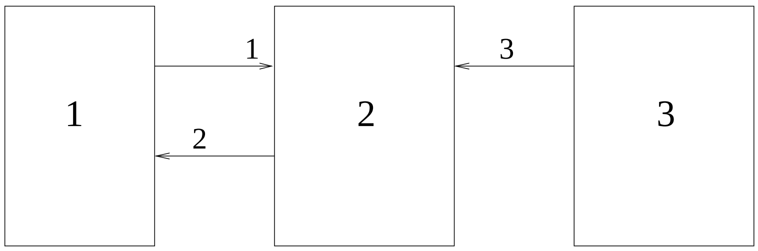

Figure 2. System S.

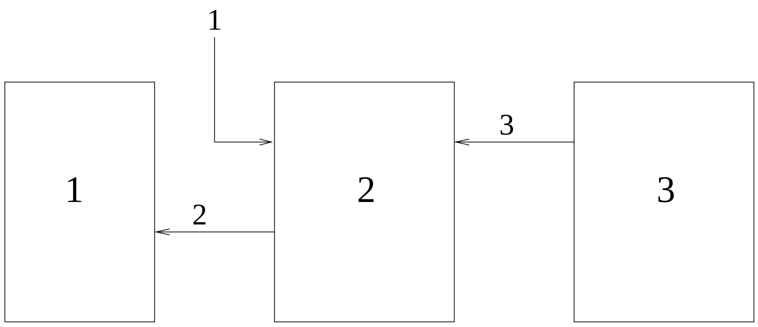

Figure 3. System S.

Let us consider the notion of *can*. Let S be a system of subautomata without external inputs such as that of Figure 2. Let p be one of the subautomata, and suppose that there are m signal lines coming out of p. What p

can do is defined in terms of a new system S_p, which is obtained from the system S by disconnecting the m signal lines coming from p and replacing them by m external input lines to the system. In Figure 2, subautomaton 1 has one output, and in the system S_1 (Figure 3) this is replaced by an external input. The new system S_p always has the same set of states as the system S. Now let π be a condition on the state such as, 'a_2 is even' or '$a_2 = a_3$'. (In the applications π may be a condition like 'The box is under the bananas'.)

We shall write

$$can(p, \pi, s)$$

which is read, 'The subautomaton p *can* bring about the condition π in the situation s' if there is a sequence of outputs from the automaton S_p that will eventually put S into a state a' that satisfies $\pi(a')$. In other words, in determining what p can achieve, we consider the effects of sequences of its actions, quite apart from the conditions that determine what it actually will do.

In Figure 2, let us consider the initial state a to be one in which all subautomata are initially in state 0. Then the reader will easily verify the following propositions:

1. Subautomaton 2 *will* never be in state 1.
2. Subautomaton 1 *can* put subautomaton 2 in state 1.
3. Subautomaton 3 *cannot* put subautomaton 2 in state 1.

Figure 2. System S

$a_1(t+1) = a_1(t) + s_2(t)$
$a_2(t+1) = a_2(t) + s_1(t) + 2s_3(t)$
$a_3(t+1) =$ **if** $a_3(t) = 0$ **then** 0 **else** $a_3(t) + 1$
$s_1(t) =$ **if** $a_1(t) = 0$ **then** 2 **else** 1
$s_2(t) = 1$
$s_3(t) =$ **if** $a_3(t) = 0$ **then** 0 **else** 1

We claim that this notion of *can* is, to a first approximation, the appropriate one for an automaton to use internally in deciding what to do by reasoning. We also claim that it corresponds in many cases to the common sense notion of *can* used in everyday speech.

In the first place, suppose we have an automaton that decides what to do by reasoning; for example, suppose it is a computer using an RP. Then its output is determined by the decisions it makes in the reasoning process. It does not know (has not computed) in advance what it will do, and, therefore, it is appropriate that it considers that it can do anything that can be achieved by some sequence of its outputs. Common-sense reasoning seems to operate in the same way.

The above rather simple notion of *can* requires some elaboration, both to represent adequately the commonsense notion and for practical purposes in the reasoning program. First, suppose that the system of automata admits external inputs. There are two ways of defining *can* in this case. One way is to assert $can(p, \pi, s)$ if p can achieve π regardless of what signals appear on the external inputs. Thus, we require the existence of a sequence of outputs of p that achieves the goal regardless of the sequence of external inputs to the system. Note that, in this definition of *can*, we are not requiring that p have any way of knowing what the external inputs were. An alternative definition requires the outputs to depend on the inputs of p. This is equivalent to saying that p can achieve a goal, provided the goal would be achieved for arbitrary inputs by some automaton put in place of p. With either of these definitions *can* becomes a function of the place of the subautomaton in the system rather than of the subautomaton itself. We do not know which of these treatments is preferable, and so we shall call the first concept *cana* and the second *canb*.

The idea that what a person can do depends on his position rather than on his characteristics is somewhat counter-intuitive. This impression can be mitigated as follows: Imagine the person to be made up of several subautomata; the output of the outer subautomaton is the motion of the joints. If we break the connection to the world at that point we can answer questions like, 'Can he fit through a given hole?' We shall get some counter-intuitive answers, however, such as that he can run at top speed for an hour or can jump over a building, since these are sequences of motions of his joints that would achieve these results.

The next step, however, is to consider a subautomaton that receives the nerve impulses from the spinal cord and transmits them to the muscles. If we break at the input to this automaton, we shall no longer say that he can jump over a building or run long at top speed since the limitations of the muscles will be taken into account. We shall, however, say that he can ride a unicycle since appropriate nerve signals would achieve this result.

The notion of *can* corresponding to the intuitive notion in the largest number of cases might be obtained by hypothesizing an *organ of will,* which makes decisions to do things and transmits these decisions to the main part of the brain that tries to carry them out and contains all the knowledge of particular facts. If we make the break at this point we shall be able to say that so-and-so cannot dial the President's secret and private telephone number because he does not know it, even though if the question were asked could he dial that particular number, the answer would be yes. However, even this break would not give the statement, 'I cannot go without saying goodbye, because this would hurt the child's feelings'.

On the basis of these examples, one might try to postulate a sequence of narrower and narrower notions of *can* terminating in a notion according to which a person can do only what he actually does. This notion would then be superfluous. Actually, one should not look for a single best notion of *can*; each of the above-mentioned notions is useful and is actually used in some circumstances. Sometimes, more than one notion is used in a single sentence, when two different levels of constraint are mentioned.

Besides its use in explicating the notion of *can*, the automaton representation of the world is very suited for defining notions of causality. For, we may say that subautomaton p caused the condition π in state s, if changing the output of p would prevent π. In fact the whole idea of a system of interacting automata is just a formalization of the commonsense notion of causality.

Moreover, the automaton representation can be used to explicate certain counterfactual conditional sentences. For example, we have the sentence, 'If I had struck this match yesterday at this time it would have lit.' In a suitable automaton representation, we have a certain state of the system yesterday at that time, and we imagine a break made where the nerves lead from my head or perhaps at the output of my 'decision box', and the appropriate signals to strike the match having been made. Then it is a definite and decidable question about the system S_p, whether the match lights or not, depending on whether it is wet, etc. This interpretation of this kind of counterfactual sentence seems to be what is needed for RP to learn from its mistakes, by accepting or generating sentences of the form, 'had I done thus-and-so I would have been successful, so I should alter my procedures in some way that would have produced the correct action in that case'.

In the foregoing we have taken the representation of the situation as a system of interacting subautomata for granted. However, a given overall situation might be represented as a system of interacting subautomata in a number of ways, and different representations might yield different results about what a given subautomaton can achieve, what would have happened if some subautomaton had acted differently, or what caused what. Indeed, in a different representation, the same or corresponding subautomata might not be identifiable. Therefore, these notions depend on the representation chosen.

For example, suppose a pair of Martians observe the situation in a room. One Martian analyzes it as a collection of interacting people as we do, but the second Martian groups all the heads together into one subautomaton and all the bodies into another. (A creature from momentum space would regard the Fourier components of the distribution of matter as the separate interacting subautomata.) How is the first Martian to convince the second that his representation is to be preferred? Roughly speaking, he would argue that the interaction between the heads and bodies of the same person is closer than the interaction between the different heads, and so more of an analysis has been achieved from 'the primordial muddle' with the conventional representation. He will be especially convincing when he points out that when the meeting is over the heads will stop interacting with each other, but will continue to interact with their respective bodies.

We can express this kind of argument formally in terms of automata as follows: Suppose we have an autonomous automaton A, that is an automaton without inputs, and let it have k states. Further, let m and n be two integers such that $m, n \geq k$. Now label k points of an m-by-n array with the states of A. This can be done in $\binom{mn}{k}!$ ways. For each of these ways we have a representation of the automaton A as a system of an m-state automaton B interacting with an n-state automaton C. Namely, corresponding to each row of the array we have a state of B and to each column a state of C. The signals are in 1–1 correspondence with the states themselves; thus each subautomaton has just as many values of its output as it has states. Now it may happen that two of these signals are equivalent in their effect on the other subautomaton, and we use this equivalence relation to form equivalence classes of signals. We may then regard the equivalence classes as the signals themselves. Suppose then that there are now r signals from B to C and s signals from C to B. We ask how small r and s can be taken in general compared to m and n. The answer may be obtained by counting the number of inequivalent automata with k states and comparing it with the number of systems of two automata with m and n states respectively and r and s signals going in the respective directions. The result is not worth working out in detail, but tells us that only a few of the k state automata admit such a decomposition with r and s small compared to m and n. Therefore, if an automaton happens to admit such a decomposition it is very unusual for it to admit a second such decomposition that is not equivalent to the first with respect to some renaming of states. Applying this argument to the real world, we may say that it is overwhelmingly probable that our customary decomposition of the world automaton into separate people and things has a unique, objective and usually preferred status. Therefore, the notions of *can*, of causality, and of counterfactual associated with this decomposition also have a preferred status.

In our opinion, this explains some of the difficulty philosophers have had in analyzing counterfactuals and causality. For example, the sentence, 'If I had struck this match yesterday, it would have lit' is meaningful only in terms of a rather complicated model of the world, which, however, has an objective preferred status. However, the preferred status of this model depends on its correspondence with a large number of facts. For this reason, it is probably not fruitful to treat an individual counterfactual conditional sentence in isolation.

It is also possible to treat notions of belief and knowledge in terms of the automaton representation. We have not worked this out very far, and the ideas presented here should be regarded as tentative. We would like to be able to give conditions under which we may say that a subautomaton p believes a certain proposition. We shall not try to do this directly but only relative to a predicate $B_p(s, w)$. Here s is the state of the automaton p and w is a proposition; $B_p(s, w)$ is true if p is to be regarded as believing w when in state s and is false otherwise. With respect to such a predicate B we may ask the following questions:

1. Are p's beliefs consistent? Are they correct?

2. Does p reason? That is, do new beliefs arise that are logical consequences of previous beliefs?

3. Does p observe? That is, do true propositions about automata connected to p cause p to believe them?

4. Does p behave rationally? That is, when p believes a sentence asserting that it should do something does p do it?

5. Does p communicate in language L? That is, regarding the content of a certain input or output signal line as in a text in language L, does this line transmit beliefs to or from p?

6. Is p self-conscious? That is, does it have a fair variety of correct beliefs about its own beliefs and the processes that change them?

It is only with respect to the predicate B_p that all these questions can be asked. However, if questions 1 thru 4 are answered affirmatively for some predicate B_p, this is certainly remarkable, and we would feel fully entitled to consider B_p a reasonable notion of belief.

In one important respect the situation with regard to belief or knowledge is the same as it was for counterfactual conditional statements: no way is provided to assign a meaning to a single statement of belief or knowledge, since for any single statement a suitable B_p can easily be constructed. Individual statements about belief or knowledge are made on the basis of a larger system which must be validated as a whole.

3 FORMALISM

In part 2 we showed how the concepts of ability and belief could be given formal definition in the metaphysically adequate automaton model and indicated the correspondence between these formal concepts and the corresponding commonsense concepts. We emphasized, however, that practical systems require epistemologically adequate systems in which those facts which are actually ascertainable can be expressed.

In this part we begin the construction of an epistemologically adequate system. Instead of giving formal definitions, however, we shall introduce the formal notions by informal natural-language descriptions and give examples of their use to describe situations and the possibilities for action they present. The formalism presented is intended to supersede that of McCarthy (1963).

3.1 Situations

A situation s is the complete state of the universe at an instant of time. We denote by Sit the set of all situations. Since the universe is too large for complete description, we shall never completely describe a situation; we shall only give facts about situations. These facts will be used to deduce further facts about that situation, about future situations and about situations that persons can bring about from that situation.

This requires that we consider not only situations that actually occur, but also hypothetical situations such as the situation that would arise if Mr. Smith sold his car to a certain person who has offered $250 for it. Since he is not going to sell the car for that price, the hypothetical situation is not completely defined; for example, it is not determined what Smith's mental state would be and therefore it is also undetermined how quickly he would return to his office, etc. Nevertheless, the representation of reality is adequate to determine some facts about this situation, enough at least to make him decide not to sell the car.

We shall further assume that the laws of motion determine, given a situation, all future situations.[1]

In order to give partial information about situations we introduce the notion of fluent.

3.2 Fluents

A *fluent* is a function whose domain is the space Sit of situations. If the range of the function is $(true, false)$, then it is called a *propositional fluent*. If its range is Sit, then it is called a *situational fluent.*

Fluents are often the values of functions. Thus $raining(x)$ is a fluent such that $raining(x)(s)$ is true if and only if it is raining at the place x in the situation s. We can also write this assertion as $raining(x, s)$ making use of the well-known equivalence between a function of two variables and a function of the first variable whose value is a function of the second variable.

Suppose we wish to assert about a situation s that person p is in place x and that it is raining in place x. We may write this in several ways each of which has its uses:

1. $at(p, x)(s) \wedge raining(x)(s)$. This corresponds to the definition given.

2. $at(p, x, s) \wedge raining(x, s)$. This is more conventional mathematically and a bit shorter.

3. $[at(p, x) \wedge raining(x)](s)$. Here we are introducing a convention that operators applied to fluents give fluents whose values are computed by applying the logical operators to the values of the operand fluents, that is, if f and g are fluents then

$$(f \textbf{ op } g)(s) = f(s) \textbf{ op } g(s).$$

4. $[\lambda s'.at(p, x, s') \wedge raining(x, s')](s)$. Here we have formed the composite fluent by λ-abstraction.

[1] This assumption is difficult to reconcile with quantum mechanics, and relativity tells us that any assignment of simultaneity to events in different places is arbitrary. However, we are proceeding on the basis that modern physics is irrelevant to common sense in deciding what to do, and in particular is irrelevant to solving the 'free will problem'.

Here are some examples of fluents and expressions involving them:

1. $time(s)$. This is the time associated with the situation s. It is essential to consider time as dependent on the situation as we shall sometimes wish to consider several different situations having the same time value, for example, the results of alternative courses of actions.

2. $in(x, y, s)$. This asserts that x is in the location y in situation s. The fluent in may be taken as satisfying a kind of transitive law, namely:

$$\forall x. \forall y. \forall z. \forall s. in(x, y, s) \wedge in(y, z, s) \rightarrow in(x, z, s).$$

We can also write this law

$$\forall x. \forall y. \forall z. \forall . in(x, y) \wedge in(y, z) \rightarrow in(x, z)$$

where we have adopted the convention that a quantifier without a variable is applied to an implicit situation variable which is the (suppressed) argument of a propositional fluent that follows. Suppressing situation arguments in this way corresponds to the natural language convention of writing sentences like, 'John was at home' or 'John is at home' leaving understood the situations to which these assertions apply.

3. $has(Monkey, Bananas, s)$. Here we introduce the convention that capitalized words denote proper names, for example, 'Monkey' is the name of a particular individual. That the individual is a monkey is not asserted, so that the expression $monkey(Monkey)$ may have to appear among the premisses of an argument. Needless to say, the reader has a right to feel that he has been given a hint that the individual Monkey will turn out to be a monkey. The above expression is to be taken as asserting that in the situation s the individual $Monkey$ has the object $Bananas$. We shall, in the examples below, sometimes omit premisses such as $monkey(Monkey)$, but in a complete system they would have to appear.

3.3 Causality

We shall make assertions of causality by means of a fluent $F(\pi)$ where π is itself a propositional fluent. $F(\pi, s)$ asserts that the situation s will be followed (after an unspecified time) by a situation that satisfies the fluent π. We may use F to assert that if a person is out in the rain he will get wet, by writing:

$$\forall x. \forall p. \forall s. raining(x, s) \wedge at(p, x, s) \wedge outside(p, s) \rightarrow F(\lambda s'. wet(p, s'), s).$$

Suppressing explicit mention of situations gives:

$$\forall x.\forall p.\forall raining(x) \wedge at(p,x) \wedge outside(p) \rightarrow F(wet(p)).$$

In this case suppressing situations simplifies the statement.

F can also be used to express physical laws. Consider the law of falling bodies which is often written

$$h = h_0 + v_0 \times (t - t_0) - \frac{1}{2}g \times (t - t_0)^2$$

together with some prose identifying the variables. Since we need a formal system for machine reasoning we cannot have any prose. Therefore, we write:

$$\begin{aligned} &\forall b.\forall t.\forall s.falling(b,s) \wedge t \geq 0 \wedge height(b,s) + velocity(b,s) \times t - \tfrac{1}{2}gt^2 > 0 \\ &\rightarrow F(\lambda s'.time(s') = time(s) + t \wedge falling(b,s') \\ &\qquad \wedge\ height(b,s') = height(b,s) + velocity(b,s) \times t - \tfrac{1}{2}gt^2, s). \end{aligned} \tag{2}$$

Suppressing explicit mention of situations in this case requires the introduction of real auxiliary quantities v, h and τ so that the sentence takes the following form:

$$\begin{aligned} &\forall b.\forall t.\forall \tau.\forall v.\forall h.\forall[\\ &falling(b) \wedge t \geq 0 \wedge h = height(b) \wedge v = velocity(b) \wedge h + vt - \tfrac{1}{2}gt^2 > 0 \\ &\wedge time = \tau \\ &\qquad \rightarrow F(time = t + \tau \wedge falling(b) \wedge height(b) = h + vt - \tfrac{1}{2}gt^2)]. \end{aligned} \tag{3}$$

There has to be a convention (or declarations) so that it is determined that $height(b)$, $velocity(b)$ and $time$ are fluents, whereas t, v, τ and h denote ordinary real numbers.

$F(\pi, s)$ as introduced here corresponds to A. N. Prior's (1957, 1968) expression $F\pi$.

The use of situation variables is analogous to the use of time-instants in the calculi of world-states which Prior (1968) calls *U-T* calculi. Prior provides many interesting correspondences between his *U-T* calculi and various axiomatizations of the modal tense-logics (that is, using this F-operator: see part 5). However, the situation calculus is richer than any of the tense-logics Prior considers.

Besides F he introduces three other operators which we also find useful; we thus have:

1. $F(\pi, s)$. For some situation s' in the future of $s, \pi(s')$ holds.
2. $G(\pi, s)$. For all situations s' in the future of $s, \pi(s')$ holds.
3. $P(\pi, s)$. For some situations s' in the past of $s, \pi(s')$ holds.
4. $H(\pi, s)$. For all situations s' in the past of $s, \pi(s')$ holds.

It seems also useful to define a situational fluent $next(\pi)$ as the next situation s' in the future of s for which $\pi(s')$ holds. If there is no such situation, that is, if $\neg F(\pi, s)$, then $next(\pi, s)$ is considered undefined. For example, we may translate the sentence 'By the time John gets home, Henry will be home too' as

$$at(Henry, home(Henry), next(at(John, home(John)), s)).$$

Also the phrase 'when John gets home' translates into

$$time(next(at(John, home(John)), s)).$$

Though $next(\pi, s)$ will never actually be computed since situations are too rich to be specified completely, the values of fluents applied to $next(\pi, s)$ will be computed.

3.4 Actions

A fundamental role in our study of actions is played by the situational fluent

$$result(p, \sigma, s)$$

Here, p is a person, σ is an action or more generally a strategy, and s is a situation. The value of $result(p, \sigma, s)$ is the situation that results when p carries out σ, starting in the situation s. If the action or strategy does not terminate, $result(p, \sigma, s)$ is considered undefined.

With the aid of $result$ we can express certain laws of ability. For example:

$$has(p, k, s) \wedge fits(k, sf) \wedge at(p, sf, s) \rightarrow open(sf, result(p, opens(sf, k), s)).$$

This formula is to be regarded as an axiom schema asserting that if in a situation s a person p has a key k that fits the safe sf, then in the situation resulting from his performing the action $opens(sf, k)$, that is, opening the safe sf with the key k, the safe is open. The assertion $fits(k, sf)$ carries the information that k is a key and sf a safe. Later we shall be concerned with combination safes that require p to *know* the combination.

3.5 Strategies

Actions can be combined into strategies. The simplest combination is a finite sequence of actions. We shall combine actions as though they were ALGOL statements, that is, procedure calls. Thus, the sequence of actions, ('move the box under the bananas', 'climb onto the box', and 'reach for the bananas') may be written:

$$\mathbf{begin}\ move(Box, Under\text{-}Bananas); climb(Box); reach\text{-}for(Bananas)\ \mathbf{end};$$

A strategy in general will be an ALGOL-like compound statement containing actions written in the form of procedure calling assignment statements, and conditional **go to**'s. We shall not include any declarations in the program since they can be included in the much larger collection of declarative sentences that determine the effect of the strategy.

Consider for example the strategy that consists of walking 17 blocks south, turning right and then walking till you come to Chestnut Street. This strategy may be written as follows:

$$\begin{array}{ll}
\mathbf{begin} & \\
 & face(South); \\
 & n := 0; \\
b: & \mathbf{if}\ n = 17\ \mathbf{then\ go\ to}\ a; \\
 & walk\text{-}a\text{-}block; n := n + 1; \\
 & \mathbf{go\ to}\ b; \\
a: & turn\text{-}right; \\
c: & walk\text{-}a\text{-}block; \\
 & \mathbf{if}\ \ name\text{-}on\text{-}street\text{-}sign \neq \text{'}Chestnut\ Street\text{'}\ \mathbf{then\ go\ to}\ \ c \\
\mathbf{end}; &
\end{array} \tag{4}$$

In the above program the external actions are represented by procedure calls. Variables to which values are assigned have a purely internal significance (we may even call it mental significance) and so do the statement labels and the go to statements.

For the purpose of applying the mathematical theory of computation we shall write the program differently: namely, each occurrence of an action α is to be replaced by an assignment statement $s := result(p, \alpha, s)$. Thus the above program becomes

begin

$$
\begin{array}{ll}
 & s := result(p, face(South), s); \\
 & n := 0; \\
b: & \textbf{if } n = 17 \textbf{ then go to } a; \\
 & s := result(p, walk\text{-}a\text{-}block, s); \\
 & n := n + 1; \\
 & \textbf{go to } b; \\
a: & s := result(p, turn\text{-}right, s); \\
c: & s := result(p, walk\text{-}a\text{-}block, s); \\
 & \textbf{if } \ name\text{-}on\text{-}street\text{-}sign \neq 'Chestnut\ Street' \textbf{ then go to } \ c
\end{array}
$$

end;

(5)

Suppose we wish to show that by carrying out this strategy John can go home provided he is initially at his office. Then according to the methods of Zohar Manna (1968a, 1968b), we may derive from this program together with the initial condition $at(John, office(John), s_0)$ and the final condition $at(John, home(John), s)$, a sentence W of first-order logic. Proving W will show that the procedure terminates in a finite number of steps and that when it terminates s will satisfy $at(John, home(John), s)$.

According to Manna's theory we must prove the following collection of sentences inconsistent for arbitrary interpretations of the predicates $q1$ and $q2$ and the particular interpretations of the other functions and predicates in the program:

$$
\begin{array}{l}
at(John, office(John), s_0), \\
q1(O, result(John, face(South), s_0)), \\
\forall n.\forall s.q1(n, s) \rightarrow \ \textbf{if } \ n = 17 \\
\qquad\qquad \textbf{then } q2(result(John, walk\text{-}a\text{-}block, result(John, turn\text{-}right, s))) \\
\qquad\qquad \textbf{else } q1(n + 1, result(John, walk\text{-}a\text{-}block, s)), \\
\forall s.q2(s) \rightarrow \ \textbf{if } name\text{-}on\text{-}street\text{-}sign(s) \neq 'Chestnut\ Street' \\
\qquad\qquad \textbf{then } q2(result(John, walk\text{-}a\text{-}block, s)) \\
\qquad\qquad \textbf{else } \neg at(John, home(John), s).
\end{array}
$$

(6)

Therefore the formula that has to be proved may be written

$$
\begin{array}{l}
\exists s_0\{at(John, office(John), s_0) \wedge q1(O, result(John, face(South), s_0))\} \\
\qquad\qquad\qquad\qquad \rightarrow \\
\exists n.\exists s.\{q1(n, s) \wedge \ \mathbf{if}\ n = 17 \\
\qquad\qquad \mathbf{then}\ q2(result(John, \mathit{walk\text{-}a\text{-}block}, result(John, \mathit{turn\text{-}right}, s))) \\
\qquad\qquad \mathbf{else}\ \neg q1(n+1, result(John, \mathit{walk\text{-}a\text{-}block}, s))\} \\
\qquad\qquad\qquad\qquad \vee \\
\exists s.\{q2(s) \wedge \ \mathbf{if}\ \mathit{name\text{-}on\text{-}street\text{-}sign}(s) \neq \mathit{'Chestnut\ Street'} \\
\qquad\qquad \mathbf{then}\ \neg q2(result(John, \mathit{walk\text{-}a\text{-}block}, s)) \\
\qquad\qquad \mathbf{else}\ at(John, home(John), s)\}.
\end{array}
\tag{7}
$$

In order to prove this sentence we would have to use the following kinds of facts expressed as sentences or sentence schemas of first-order logic:

1. Facts of geography. The initial street stretches at least 17 blocks to the south, and intersects a street which in turn intersects Chestnut Street a number of blocks to the right; the location of John's home and office.

2. The fact that the fluent *name-on-street-sign* will have the value 'Chestnut Street' at that point.

3. Facts giving the effects of action α expressed as predicates about $result(p, \alpha, s)$ deducible from sentences about s.

4. An axiom schema of induction that allows us to deduce that the loop of walking 17 blocks will terminate.

5. A fact that says that Chestnut Street is a finite number of blocks to the right after going 17 blocks south. This fact has nothing to do with the possibility of walking. It may also have to be expressed as a sentence schema or even as a sentence of second-order logic.

When we consider making a computer carry out the strategy, we must distinguish the variable s from the other variables in the second form of the program. The other variables are stored in the memory of the computer and the assignments may be executed in the normal way. The variable s represents the state of the world and the computer makes an assignment to it by performing an action. Likewise the fluent *name-on-street-sign* requires an action, of observation.

3.6 Knowledge and Ability

In order to discuss the role of knowledge in one's ability to achieve goals let us return to the example of the safe. There we had

1. $has(p, k, s) \wedge fits(k, sf) \wedge at(p, sf, s) \rightarrow open(sf, result(p, opens(sf, k), s))$,

which expressed sufficient conditions for the ability of a person to open a safe with a key. Now suppose we have a combination safe with a combination c. Then we may write:

2. $fits2(c, sf) \wedge at(p, sf, s) \rightarrow open(sf, result(p, opens2(sf, c), s))$,

where we have used the predicate $fits2$ and the action $opens2$ to express the distinction between a key fitting a safe and a combination fitting it, and also the distinction between the acts of opening a safe with a key and a combination. In particular, $opens2(sf, c)$ is the act of manipulating the safe in accordance with the combination c. We have left out a sentence of the form $has2(p, c, s)$ for two reasons. In the first place, it is unnecessary: if you manipulate a safe in accordance with its combination it will open; there is no need to have anything. In the second place it is not clear what $has2(p, c, s)$ means. Suppose, for example, that the combination of a particular safe sf is the number 34125, then $fits(34125, sf)$ makes sense and so does the act $opens2(sf, 34125)$. (We assume that $open(sf, result(p, opens2(sf, 34111), s))$ would not be true.) But what could $has(p, 34125, s)$ mean? Thus, a direct parallel between the rules for opening a safe with a key and opening it with a combination seems impossible.

Nevertheless, we need some way of expressing the fact that one has to know the combination of a safe in order to open it. First we introduce the function $combination(sf)$ and rewrite 2 as

3. $at(p, sf, s) \wedge csafe(sf)$
$\rightarrow open(sf, result(p, opens2(sf, combination(sf)), s))$,

where $csafe(sf)$ asserts that sf is a combination safe and $combination(sf)$ denotes the combination of sf. (We could not write $key(sf)$ in the other case unless we wished to restrict ourselves to the case of safes with only one key.)

Next we introduce the notion of a feasible strategy for a person. The idea is that a strategy that would achieve a certain goal might not be feasible for a person because he lacks certain knowledge or abilities.

Our first approach is to regard the action $opens2(sf, combination(sf))$ as infeasible because p might not know the combination. Therefore, we introduce a new function $idea\text{-}of\text{-}combination(p, sf, s)$ which stands for person p's idea of the combination of sf in situation s.

The action $opens2(sf, idea\text{-}of\text{-}combination(p, sf, s))$ is regarded as feasible for p, since p is assumed to know his idea of the combination if this is defined. However, we leave sentence 3 as it is so we cannot yet prove $open(sf, result(p, opens2(sf, idea\text{-}of\text{-}combination(p, sf, s)), s))$. The assertion that p knows the combination of sf can now be expressed as[2]

5. $idea\text{-}of\text{-}combination(p, sf, s) = combination(sf)$

and with this, the possibility of opening the safe can be proved.

Another example of this approach is given by the following formalization of getting into conversation with someone by looking up his number in the telephone book and then dialing it.

The strategy for p in the first form is

$$\begin{array}{l} \textbf{begin} \\ \quad lookup(q, Phone\text{-}book); \\ \quad dial(idea\text{-}of\text{-}phone\text{-}number(sq, p)) \\ \textbf{end}; \end{array} \tag{8}$$

or in the second form

$$\begin{array}{l} \textbf{begin} \\ \quad s := result(p, lookup(q, Phone\text{-}book), s_0); \\ \quad s := result(p, dial(idea\text{-}of\text{-}phone\text{-}number(q, p, s)), s) \\ \textbf{end}; \end{array} \tag{9}$$

The premisses to write down appear to be

1. $has(p, Phone\text{-}book, s_0)$
2. $listed(q, Phone\text{-}book, s_0)$
3. $\forall s.\forall p.\forall q.has(p, Phone\text{-}book, s) \wedge listed(q, Phone\text{-}book, s)$
$\rightarrow phone\text{-}number(q) = idea\text{-}of\text{-}phone\text{-}number(p, q,$
$result(p, lookup(q, Phone\text{-}book), s))$
4. $\forall s.\forall p.\forall q.\forall x.at(q, home(q), s) \wedge has(p, x, s) \wedge telephone(x)$

[2]1996: Apparently there was never an equation 4.

$\rightarrow in\text{-}conversation(p, q, result(p, dial(phone\text{-}number(q)), s))$

5. $at(q, home(q), s_0)$
6. $telephone(Telephone)$
7. $has(p, Telephone, s_0)$

Unfortunately, these premisses are not sufficient to allow one to conclude that

$$\begin{aligned}&in\text{-}conversation(p, q, result(p,\ \mathbf{begin}\ lookup(q, Phone\text{-}book);\\ &dial(idea\text{-}of\text{-}phone\text{-}number(q, p))\ \mathbf{end}; s_0)).\end{aligned} \tag{10}$$

The trouble is that one cannot show that the fluents $at(q, home(q))$ and $has(p, Telephone)$ still apply to the situation $result(p, lookup(q, Phone\text{-}book), s_0)$. To make it come out right we shall revise the third hypothesis to read:

$$\begin{aligned}&\forall s.\forall p.\forall q.\forall x.\forall y.\\ &\quad at(q, y, s) \wedge has(p, x, s) \wedge has(p, Phone\text{-}book, s) \wedge listed(q, Phone\text{-}book)\\ &\qquad \rightarrow [\lambda r.at(q, y, r) \wedge has(p, x, r) \wedge phone\text{-}number(q)\\ &\qquad\qquad = idea\text{-}of\text{-}phone\text{-}number(p, q, r)]\\ &\qquad\qquad\qquad (result(p, lookup(q, Phone\text{-}book), s)).\end{aligned} \tag{11}$$

This works, but the additional hypotheses about what remains unchanged when p looks up a telephone number are quite *ad hoc*. We shall treat this problem in a later section.

The present approach has a major technical advantage for which, however, we pay a high price. The advantage is that we preserve the ability to replace any expression by an equal one in any expression of our language. Thus if $phone\text{-}number(John) = 3217580$, any true statement of our language that contains 3217580 or $phonenumber(John)$ will remain true if we replace one by the other. This desirable property is termed referential transparency.

The price we pay for referential transparency is that we have to introduce $idea\text{-}of\text{-}phone\text{-}number(p, q, s)$ as a separate *ad hoc* entity and cannot use the more natural $idea\text{-}of(p, phone\text{-}number(q), s)$ where $idea\text{-}of(p, con, s)$ is some kind of operator applicable to the concept con. Namely, the sentence

$$idea\text{-}of(p, phone\text{-}number(q), s) = phone\text{-}number(q)$$

would be supposed to express that p knows q's phone-number, but $idea\text{-}of(p, 321\text{-}7580, s) = 3217580$ expresses only that p understands that number. Yet with transparency and the fact that $phone\text{-}number(q) = 3217580$ we could derive the former statement from the latter.

A further consequence of our approach is that feasibility of a strategy is a referentially opaque concept since a strategy containing $idea\text{-}of\text{-}phone\text{-}number(p, q, s)$ is regarded as feasible while one containing $phone\text{-}number(q)$ is not, even though these quantities may be equal in a particular case. Even so, our language is still referentially transparent since feasibility is a concept of the metalanguage.

A classical poser for the reader who wants to solve these difficulties to ponder is, 'George IV wondered whether the author of the Waverly novels was Walter Scott' and 'Walter Scott is the author of the Waverly novels', from which we do not wish to deduce, 'George IV wondered whether Walter Scott was Walter Scott'. This example and others are discussed in the first chapter of Church's *Introduction to Mathematical Logic* (1956).

In the long run it seems that we shall have to use a formalism with referential opacity and formulate precisely the necessary restrictions on replacement of equals by equals; the program must be able to reason about the feasibility of its strategies, and users of natural language handle referential opacity without disaster. In part 5 we give a brief account of the partly successful approach to problems of referential opacity in modal logic.

4 REMARKS AND OPEN PROBLEMS

The formalism presented in part 3 is, we think, an advance on previous attempts, but it is far from epistemological adequacy. In the following sections we discuss a number of problems that it raises. For some of them we have proposals that might lead to solutions.

4.1 The approximate character of $result(p, \sigma, s)$

Using the situational fluent $result(p, \sigma, s)$ in formulating the conditions under which strategies have given effects has two advantages over the $can(p, \pi, s)$ of part 2. It permits more compact and transparent sentences, and it lends itself to the application of the mathematical theory of computation to prove that certain strategies achieve certain goals.

However, we must recognize that it is only an approximation to say that an action, other than that which will actually occur, leads to a definite situation. Thus if someone is asked, 'How would you feel tonight if you challenged him to a duel tomorrow morning and he accepted?' he might well reply, 'I can't imagine the mental state in which I would do it; if the words inexplicably popped out of my mouth as though my voice were under someone else's control that would be one thing; if you gave me a long-lasting belligerence drug that would be another.'

From this we see that $result(p, \sigma, s)$ should not be regarded as being defined in the world itself, but only in certain representations of the world; albeit in representations that may have a preferred character as discussed in part 2.

We regard this as a blemish on the smoothness of interpretation of the formalism, which may also lead to difficulties in the formal development. Perhaps another device can be found which has the advantages of *result* without the disadvantages.

4.2 Possible Meanings of 'can' for a Computer Program

A computer program can readily be given much more powerful means of introspection than a person has, for we may make it inspect the whole of its memory including program and data to answer certain introspective questions, and it can even simulate (slowly) what it would do with given initial data. It is interesting to list various notions of $can(Program, \pi)$ for a program.

1. There is a sub-program σ and room for it in memory which would achieve π if it were in memory, and control were transferred to σ. No assertion is made that *Program* knows σ or even knows that σ exists.

2. σ exists as above and that σ will achieve π follows from information in memory according to a proof that *Program* is capable of checking.

3. *Program*'s standard problem-solving procedure will find σ if achieving π is ever accepted as a subgoal.

4.3 The Frame Problem

In the last section of part 3, in proving that one person could get into conversation with another, we were obliged to add the hypothesis that if a person has a telephone he still has it after looking up a number in the telephone book. If we had a number of actions to be performed in sequence we would have quite a number of conditions to write down that certain actions do not change the values of certain fluents. In fact with n actions and m fluents we might have to write down mn such conditions.

We see two ways out of this difficulty. The first is to introduce the notion of frame, like the state vector in McCarthy (1962). A number of fluents are declared as attached to the frame and the effect of an action is described by telling which fluents are changed, all others being presumed unchanged.

This can be formalized by making use of yet more ALGOL notation, perhaps in a somewhat generalized form. Consider a strategy in which p performs the action of going from x to y. In the first form of writing strategies we have $go(x, y)$ as a program step. In the second form we have $s := result(p, go(x, y), s)$. Now we may write

$$location(p) := tryfor(y, x)$$

and the fact that other variables are unchanged by this action follows from the general properties of assignment statements. Among the conditions for successful execution of the program will be sentences that enable us to show that when this statement is executed, $tryfor(y, x) = y$. If we were willing to consider that p could go anywhere we could write the assignment statement simply as

$$location(p) := y$$

The point of using $tryfor$ here is that a program using this simpler assignment is, on the face of it, not possible to execute, since p may be unable to go to y. We may cover this case in the more complex assignment by agreeing that when p is barred from y, $tryfor(y, x) = x$.

In general, restrictions on what could appear on the right side of an assignment to a component of the situation would be included in the conditions for the feasibility of the strategy. Since components of the situation that change independently in some circumstances are dependent in others, it may be worthwhile to make use of the block structure of ALGOL. We shall not explore this approach further in this paper.

Another approach to the frame problem may follow from the methods of the next section; and in part 5 we mention a third approach which may be useful, although we have not investigated it at all fully.

4.4 Formal Literatures

In this section we introduce the notion of formal literature which is to be contrasted with the well-known notion of formal language. We shall mention some possible applications of this concept in constructing an epistemologically adequate system.

A formal literature is like a formal language with a history: we imagine that up to a certain time a certain sequence of sentences have been said. The literature then determines what sentences may be said next. The formal definition is as follows.

Let A be a set of potential sentences, for example, the set of all finite strings in some alphabet. Let $Seq(A)$ be the set of finite sequences of elements of A and let $L : Seq(A) \rightarrow \{\mathbf{true}, \mathbf{false}\}$ be such that if $s \in Seq(A)$ and $L(s)$, that is $L(s) = true$, and σ_1 is an initial segment of σ then $L(\sigma_1)$. The pair (A, L) is termed a *literature*. The interpretation is that a_n may be said after $a_1, \ldots, a_{n-1})$, provided $L((a_1, \ldots, a_n))$. We shall also write $\sigma \in L$ and refer to σ as a string of the literature L.

From a literature L and a string $\sigma \in L$ we introduce the derived literature L_σ. Namely, $\tau \in L_\sigma$ if and only if $\sigma * \tau \in L$, where $\sigma * \tau$ denotes the concatenation of σ and τ.

We shall say that the language L is universal for the class Φ of literatures if for every literature $M \in \Phi$ there is a string $\sigma(M) \in L$ such that $M = L_{\sigma(M)}$; that is, $\tau \in M$ if and only if $\sigma(M) * \tau \in L$.

We shall call a literature computable if its strings form a recursively enumerable set. It is easy to see that there is a computable literature $U(C)$ that is universal with respect to the set C of computable literatures. Namely, let e be a computable literature and let c be the representation of the Gödel number of the recursively enumerable set of e as a string of elements of A. Then, we say $c * \tau \in U_C$ if and only if $\tau \in e$.

It may be more convenient to describe natural languages as formal literatures than as formal languages: if we allow the definition of new terms and require that new terms be used in accordance with their definitions, then we have restrictions on sentences that depend on what sentences have previously been uttered. In a programming language, the restriction that an identifier not be used until it has been declared, and then only consistently with the declaration, is of this form.

Any natural language may be regarded as universal with respect to the set of natural languages in the approximate sense that we might define French in terms of English and then say 'From now on we shall speak only French'.

All the above is purely syntactic. The applications we envisage to artificial intelligence come from a certain kind of interpreted literature. We are not able to describe precisely the class of literatures that may prove useful, only to sketch a class of examples.

Suppose we have an interpreted language such as first-order logic perhaps including some modal operators. We introduce three additional operators: $consistent(\Phi)$, $normally(\Phi)$, and $probably(\Phi)$. We start with a list of sentences as hypotheses. A new sentence may be added to a string σ of sentences according to the following rules:

1. Any consequence of sentences of σ may be added.
2. If a sentence Φ is consistent with σ, then $consistent(\Phi)$ may be added. Of course, this is a non-computable rule. It may be weakened to say that $consistent(\Phi)$ may be added provided Φ can be shown to be consistent with σ by some particular proof procedure.
3. $normally(\Phi), consistent(\Phi) \vdash probably(\Phi)$.
4. $\Phi \vdash probably(\Phi)$ is a possible deduction.
5. If $\Phi_1, \Phi_2, \ldots, \Phi_n \vdash \Phi$ is a possible deduction then

$$probably(\Phi_1), \ldots, probably(\Phi_n) \vdash probably(\Phi)$$

is also a possible deduction.

The intended application to our formalism is as follows:

In part 3 we considered the example of one person telephoning another, and in this example we assumed that if p looks up q's phone-number in the book, he will know it, and if he dials the number he will come into conversation with q. It is not hard to think of possible exceptions to these statements such as:

1. The page with q's number may be torn out.
2. p may be blind.
3. Someone may have deliberately inked out q's number.
4. The telephone company may have made the entry incorrectly.
5. q may have got the telephone only recently.
6. The phone system may be out of order.
7. q may be incapacitated suddenly.

For each of these possibilities it is possible to add a term excluding the difficulty in question to the condition on the result of performing the action. But we can think of as many additional difficulties as we wish, so it is impractical to exclude each difficulty separately.

We hope to get out of this difficulty by writing such sentences as

$$\begin{aligned}&\forall p.\forall q.\forall s.at(q, home(q), s)\\ &\rightarrow normally(in\text{-}conversation(p, q, result(p, dials(phone\text{-}number(q)), s))).\end{aligned} \tag{12}$$

We would then be able to deduce

$$probably(in\text{-}conversation(p, q, result(p, dials(phone\text{-}number(q)), s_0)))$$

provided there were no statements like

$$kaput(Phone\text{-}system, s_0)$$

and

$$\begin{aligned}&\forall s.kaput(Phone\text{-}system, s)\\ &\rightarrow \neg in\text{-}conversation(p, q, result(p, dials(phone\text{-}number(q)), s))\end{aligned} \tag{13}$$

present in the system.

Many of the problems that give rise to the introduction of frames might be handled in a similar way.

The operators *normally*, *consistent* and *probably* are all modal and referentially opaque. We envisage systems in which $probably(\pi)$ and $probably(\neg\pi)$ and therefore $probably(\mathbf{false})$ will arise. Such an event should give rise to a search for a contradiction.

We hereby warn the reader, if it is not already clear to him, that these ideas are very tentative and may prove useless, especially in their present form. However, the problem they are intended to deal with, namely the impossibility of naming every conceivable thing that may go wrong, is an important one for artificial intelligence, and some formalism has to be developed to deal with it.

4.5 Probabilities

On numerous occasions it has been suggested that the formalism take uncertainty into account by attaching probabilities to its sentences. We agree that the formalism will eventually have to allow statements about the probabilities of events, but attaching probabilities to all statements has the following objections:

1. It is not clear how to attach probabilities to statements containing quantifiers in a way that corresponds to the amount of conviction people have.
2. The information necessary to assign numerical probabilities is not ordinarily available. Therefore, a formalism that required numerical probabilities would be epistemologically inadequate.

4.6 Parallel Processing

Besides describing strategies by ALGOL-like programs we may also want to describe the laws of change of the situation by such programs. In doing so we must take into account the fact that many processes are going on simultaneously and that the single-activity-at-a-time ALGOL-like programs will have to be replaced by programs in which processes take place in parallel, in order to get an epistemologically adequate description. This suggests examining the so-called simulation languages; but a quick survey indicates that they are rather restricted in the kinds of processes they allow to take place in parallel and in the types of interaction allowed. Moreover, at present there is no developed formalism that allows proofs of the correctness of parallel programs.

5 DISCUSSION OF LITERATURE

The plan for achieving a generally intelligent program outlined in this paper will clearly be difficult to carry out. Therefore, it is natural to ask if some simpler scheme will work, and we shall devote this section to criticising some simpler schemes that have been proposed.

1. L. Fogel (1966) proposes to evolve intelligent automata by altering their state transition diagrams so that they perform better on tasks of greater and greater complexity. The experiments described by Fogel involve machines

with less than 10 states being evolved to predict the next symbol of a quite simple sequence. We do not think this approach has much chance of achieving interesting results because it seems limited to automata with small numbers of states, say less than 100, whereas computer programs regarded as automata have 2^{10^5} to 2^{10^7} states. This is a reflection of the fact that, while the representation of behaviours by finite automata is metaphysically adequate—in principle every behaviour of which a human or machine is capable can be so represented—this representation is not epistemologically adequate; that is, conditions we might wish to impose on a behaviour, or what is learned from an experience, are not readily expresible as changes in the state diagram of an automaton.

2. A number of investigators (Galanter 1956, Pivar and Finkelstein 1964) have taken the view that intelligence may be regarded as the ability to predict the future of a sequence from observation of its past. Presumably, the idea is that the experience of a person can be regarded as a sequence of discrete events and that intelligent people can predict the future. Artificial intelligence is then studied by writing programs to predict sequences formed according to some simple class of laws (sometimes probabilistic laws). Again the model is metaphysically adequate but epistemologically inadequate.

In other words, what we know about the world is divided into knowledge about many aspects of it, taken separately and with rather weak interaction. A machine that worked with the undifferentiated encoding of experience into a sequence would first have to solve the encoding, a task more difficult than any the sequence extrapolators are prepared to undertake. Moreover, our knowledge is not usable to predict exact sequences of experience. Imagine a person who is correctly predicting the course of a football game he is watching; he is not predicting each visual sensation (the play of light and shadow, the exact movements of the players and the crowd). Instead his prediction is on the level of: team A is getting tired; they should start to fumble or have their passes intercepted.

3. Friedberg (1958,1959) has experimented with representing behaviour by a computer program and evolving a program by random mutations to perform a task. The epistemological inadequacy of the representation is expressed by the fact that desired changes in behaviour are often not representable by small changes in the machine language form of the program. In particular, the effect on a reasoning program of learning a new fact is not so representable.

4. Newell and Simon worked for a number of years with a program called

the General Problem Solver (Newell *et. al.* 1959, Newell and Simon 1961). This program represents problems as the task of transforming one symbolic expression into another using a fixed set of transformation rules. They succeeded in putting a fair variety of problems into this form, but for a number of problems the representation was awkward enough so that GPS could only do small examples. The task of improving GPS was studied as a GPS task, but we believe it was finally abandoned. The name, General Problem Solver, suggests that its authors at one time believed that most problems could be put in its terms, but their more recent publications have indicated other points of view.

It is interesting to compare the point of view of the present paper with that expressed in Newell and Ernst (1965) from which we quote the second paragraph:

> We may consider a problem solver to be a process that takes a problem as input and provides (when successful) the solution as output. The problem consists of the problem statement, or what is immediately given, and auxiliary information, which is potentially relevant to the problem but available only as the result of processing. The problem solver has available certain methods for attempting to solve the problem. For the problem solver to be able to work on a problem it must first transform the problem statement from its external form into the internal representation. Thus (roughly), the class of problems the problem solver can convert into its internal representation determines how broad or general it is, and its success in obtaining solutions to problems in internal form determines its power. Whether or not universal, such a decomposition fits well the structure of present problem solving programs.

In a very approximate way their division of the problem solver into the input program that converts problems into internal representation and the problem solver proper corresponds to our division into the epistemological and heuristic pats of the artificial intelligence problem. The difference is that we are more concerned with the suitability of the internal representation itself.

Newell (1965) poses the problem of how to get what we call heuristically adequate representations of problems, and Simon (1966) discusses the concept of 'can' in a way that should be compared with the present approach.

5.1 Modal Logic

It is difficult to give a concise definition of modal logic. It was originally invented by Lewis (1918) in an attempt to avoid the 'paradoxes' of implication (a false proposition implies any proposition). The idea was to distinguish two sorts of truth: *necessary* truth and mere *contingent* truth. A contingently true proposition is one which, though true, could be false. This is formalized by introducing the modal operator $\Box$ (read 'necessarily') which forms propositions from propositions. Then p's being a necessary truth is expressed by $\Box p$'s being true. More recently, modal logic has become a much-used tool for analyzing the logic of such various propositional operators as belief, knowledge and tense.

There are very many possible axiomatizations of the logic of $\Box$ none of which seem more intuitively plausible than many others. A full account of the main classical systems is given by Feys (1965), who also includes an excellent bibliography. We shall give here an axiomatization of a fairly simple modal logic, the system M of Feys – von Wright. One adds to any full axiomatization of propositional calculus the following:

$Ax.1 : \Box p \to p.$
$Ax.2 : \Box(p \to q) \to (\Box p \to \Box q).$
Rule 1: from p and $p \to q$, infer q.
Rule 2: from p, infer $\Box p$.

(This axiomatization is due to Gödel.) There is also a dual modal operator $\diamond$, defined as $\neg\Box\neg$. Its intuitive meaning is 'possibly': $\diamond p$ is true when p is at least possible, although p may be in fact false (or true). The reader will be able to see the intuitive correspondence between $\neg\diamond p$—p is impossible, and $\Box\neg p$—that is, p is necessarily false.

M is a fairly weak modal logic. One can strengthen it by adding axioms, for example, adding $Ax.3 : \Box p \to \Box\Box p$ yields the system called $S4$; adding $Ax.4 : \diamond p \to \Box\diamond p$ yields $S5$; and other additions are possible. However, one can also weaken all the systems in various ways, for instance by changing $Ax.1$ to $Ax.1' : \Box p \to \diamond p$. One easily sees that $Ax.1$ implies $Ax.1'$, but the converse is not true. The systems obtained in this way are known as the *deontic* versions of the systems. These modifications will be useful later when we come to consider tense-logics as modal logics.

One should note that the truth or falsity of $\Box p$ is not decided by p's being true. Thus $\Box$ is not a truth-functional operator (unlike the usual logical connectives, for instance) and so there is no direct way of using truth-tables to analyze propositions containing modal operators. In fact the decision problem for modal propositional calculi has been quite nontrivial. It is just this property which makes modal calculi so useful, as belief, tense, etc., when interpreted as propositional operators, are all nontruthfunctional.

The proliferation of modal propositional calculi, with no clear means of comparison, we shall call the *first problem* of modal logic. Other difficulties arise when we consider modal predicate calculi, that is, when we attempt to introduce quantifiers. This was first done by Barcan-Marcus (1946).

Unfortunately, all the early attempts at modal predicate calculi had unintuitive theorems (see for instance Kripke 1963a), and, moreover, all of them met with difficulties connected with the failure of Leibniz' law of identity, which we shall try to outline. Leibniz' law is

$$L : \forall x.\forall y.x = y \rightarrow (F(x) \equiv F(y))$$

where F is any open sentence. Now this law fails in modal contexts. For instance, consider this instance of L:

$$L_1 : \forall x.\forall y.x = y \rightarrow (\Box(x = x) \equiv \Box(x = y)).$$

By rule 2 of M (which is present in almost all modal logics), since $x = x$ is a theorem, so is $\Box(x = x)$. Thus L_1 yields

$$L_2 : \forall x.\forall y.x = y \rightarrow \Box(x = y).$$

But, the argument goes, this is counterintuitive. For instance the morning star is in fact the same individual as the evening star (the planet Venus). However, they are not *necessarily* equal: one can easily imagine that they might be distinct. This famous example is known as the 'morning star paradox'.

This and related difficulties compel one to abandon Leibniz' law in modal predicate calculi, or else to modify the laws of quantification (so that it is impossible to obtain the undesirable instances of universal sentences such as L_2). This solves the purely formal problem, but leads to severe difficulties in interpreting these calculi, as Quine has urged in several papers (cf. Quine 1964).

The difficulty is this. A sentence $\Phi(a)$ is usually thought of as ascribing some property to a certain individual a. Now consider the morning star;

clearly, the morning star is necessarily equal to the morning star. However, the evening star is not necessarily equal to the morning star. Thus, this one individual—the planet Venus—both has and does not have the property of being necessarily equal to the morning star. Even if we abandon proper names the difficulty does not disappear: for how are we to interpret a statement like $\exists x.\exists y(x = y \wedge \Phi(x) \wedge \neg\Phi(y))$?

Barcan-Marcus has urged an unconventional reading of the quantifiers to avoid this problem. The discussion between her and Quine in Barcan-Marcus (1963) is very illuminating. However, this raises some difficulties—see Belnap and Dunn (1968)—and the recent semantic theory of modal logic provides a more satisfactory method of interpreting modal sentences.

This theory was developed by several authors (Hintikka 1963, 1967a; Kanger 1957; Kripke 1963a, 1963b, 1965), but chiefly by Kripke. We shall try to give an outline of this theory, but if the reader finds it inadequate he should consult Kripke (1963a).

The idea is that modal calculi describe several *possible worlds* at once, instead of just one. Statements are not assigned a single truth-value, but rather a spectum of truth-values, one in each possible world. Now, a statement is necessary when it is true in *all* possible worlds—more or less. Actually, in order to get different modal logics (and even then not all of them) one has to be a bit more subtle, and have a binary relation on the set of possible worlds—the alternativeness relation. Then a statement is necessary in a world when it is true in all alternatives to that world. Now it turns out that many common axioms of modal propositional logics correspond directly to conditions of alternativeness. Thus for instance in the system M above, $Ax.1$ corresponds to the reflexiveness of the alternativeness relation; $Ax.3(\Box p \rightarrow \Box\Box p)$ corresponds to its transitivity. If we make the alternativeness relation into an equivalence relation, then this is just like not having one at all; and it corresponds to the axiom: $\diamond p \rightarrow \Box \diamond p$.

This semantic theory already provides an answer to the first problem of modal logic: a rational method is available for classifying the multitude of propositional modal logics. More importantly, it also provides an intelligible interpretation for modal predicate calculi. One has to imagine each possible world as having a set of individuals and an assignment of individuals to names of the language. Then each statement takes on its truth value in a world s according to the particular set of individuals and assignment associated with s. Thus, a possible world is an interpretation of the calculus, in the usual sense.

Now, the failure of Leibniz' law is no longer puzzling, for in one world the morning star—for instance—may be equal to (the same individual as) the evening star, but in another the two may be distinct.

There are still difficulties, both formal—the quantification rules have to be modified to avoid unintuitive theorems (see Kripke, 1963a, for the details)—and interpretative: it is not obvious what it means to have the *same* individual existing in *different* worlds.

It is possible to gain the expressive power of modal logic without using modal operators by constructing an ordinary truth-functional logic which describes the multiple-world semantics of modal logic directly. To do this we give every predicate an extra argument (the world-variable; or in our terminology the situation-variable) and instead of writing '$\Box\Phi$', we write

$$\forall t.A(s,t) \rightarrow \Phi(t),$$

where A is the alternativeness relation between situations. Of course we must provide appropriate axioms for A.

The resulting theory will be expressed in the notation of the situation calculus; the proposition Φ has become a propositional fluent $\lambda s.\Phi(s)$, and the 'possible worlds' of the modal semantics are precisely the situations. Notice, however, that the theory we get is weaker than what would have been obtained by adding modal operators directly to the situation calculus, for we can give no translation of assertions such as $\Box\pi(s)$, where s is a situation, which this enriched situation calculus would contain.

It is possible, in this way, to reconstruct within the situation calculus subtheories corresponding to the tense-logics of Prior and to the knowledge logics of Hintikka, as we shall explain below. However, there is a qualification here: so far we have only explained how to translate the propositional modal logics into the situation calculus. In order to translate quantified modal logic, with its difficulties of referential opacity, we must complicate the situation calculus to a degree which makes it rather clumsy. There is a special predicate on individuals and situations—*exists(i,s)*—which is regarded as true when i names an individual existing in the situation s. This is necessary because situations may contain different individuals. Then quantified assertions of the modal logic are translated according to the following scheme:

$$\forall x.\Phi(x) \rightarrow \forall x.exists(x,s) \rightarrow \Phi(x,s)$$

where s is the introduced situation variable. We shall not go into the details of this extra translation in the examples below, but shall be content to define the translations of the propositional tense and knowledge logics into the situation calculus.

5.2 Logic of Knowledge

The logic of knowledge was first investigated as a modal logic by Hintikka in his book *Knowledge and belief* (1962). We shall only describe the knowledge calculus. He introduces the modal operator K_a (read 'a knows that'), and its dual P_a, defined as $\neg K_a \neg$. The semantics is obtained by the analogous reading of K_a as: 'it is true in all possible worlds compatible with a's knowledge that'. The propositional logic of K_a (similar to $\Box$) turns out to be $S4$, that is $M + Ax.3$; but there are some complexities over quantification. (The last chapter of the book contains another excellent account of the overall problem of quantification in modal contexts.) This analysis of knowledge has been criticized in various ways (Chisholm 1963, Follesdal 1967) and Hintikka has replied in several important papers (1967b, 1967c, 1972). The last paper contains a review of the different senses of 'know' and the extent to which they have been adequately formalized. It appears that two senses have resisted capture. First, the idea of 'knowing how', which appears related to our 'can'; and secondly, the concept of knowing a person (place, etc.) when this means 'being acquainted with' as opposed to simply knowing *who* a person *is*.

In order to translate the (propositional) knowledge calculus into 'situation' language, we introduce a three-place predicate into the situation calculus termed 'shrug'. $Shrug(p, s_1, s_2)$, where p is a person and s_1 and s_2 are situations, is true when, if p is in fact in situation s_2, then for all he knows he might be in situation s_1. That is to say, s_1 is an *epistemic alternative* to s_2, as far as the individual p is concerned—this is Hintikka's term for his alternative worlds (he calls them model-sets).

Then we translate $K_p q$, where q is a proposition of Hintikka's calculus, as $\forall t.shrug(p, t, s) \rightarrow q(t)$, where $\lambda s.q(s)$ is the fluent which translates q. Of course we have to supply axioms for $shrug$, and in fact so far as the pure knowledge-calculus is concerned, the only two necessary are

$$K1 : \forall s.\forall p.shrug(p, s, s)$$

and

$$K2 : \forall p.\forall s.\forall t.\forall r.(shrug(p, t, s) \wedge shrug(p, r, t)) \rightarrow shrug(p, r, s)$$

that is, reflexivity and transitivity.

Others of course may be needed when we add tenses and other machinery to the situation calculus, in order to relate knowledge to them.

5.3 Tense Logics

This is one of the largest and most active areas of philosophic logic. Prior's book *Past, Present and Future* (1968) is an extremely thorough and lucid account of what has been done in the field. We have already mentioned the four propositional operators F, G, P, H which Prior discusses. He regards these as modal operators; then the alternativeness relation of the semantic theory is simply the time-ordering relation. Various axiomatizations are given, corresponding to deterministic and nondeterministic tenses, ending and nonending times, etc; and the problems of quantification turn up again here with renewed intensity. To attempt a summary of Prior's book is a hopeless task, and we simply urge the reader to consult it. More recently several papers have appeared (see, for instance, Bull 1968) which illustrate the technical sophistication tense-logic has reached, in that full completeness proofs for various axiom systems are now available.

As indicated above, the situation calculus contains a tense-logic (or rather several tense-logics), in that we can define Prior's four operators in our system and by suitable axioms reconstruct various axiomatizations of these four operators (in particular, all the axioms in Bull (1968) can be translated into the situation calculus).

Only one extra nonlogical predicate is necessary to do this: it is a binary predicate of situations called *cohistorical*, and is intuitively meant to assert of its arguments that one is in the other's future. This is necessary because we want to consider some pairs of situations as being not temporally related at all. We now define F (for instance) thus:

$$F(\pi, s) \equiv \exists t. cohistorical(t, s) \wedge time(t) > time(s) \wedge \pi(t).$$

The other operators are defined analogously.

Of course we have to supply axioms for 'cohistorical' and time: this is not difficult. For instance, consider one of Bull's axioms, say $Gp \rightarrow GGp$, which is better (for us) expressed in the form $FFp \rightarrow Fp$. Using the definition, this translates into:

$$
\begin{aligned}
&(\exists t.cohistorical(t,s) \wedge time(t) > time(s)\\
&\wedge \exists r.cohistorical(r,t) \wedge time(r) > time(t) \wedge \pi(r))\\
&\qquad \rightarrow (\exists r.cohistorical(r,s) \wedge time(r) > time(s) \wedge \pi(r))
\end{aligned} \tag{14}
$$

which simplifies (using the transitivity of '>') to

$$\forall t.\forall r.(cohistorical(r,t) \wedge cohistorical(t,s)) \rightarrow cohistorical(r,s)$$

that is, the transitivity of 'cohistorical'. This axiom is precisely analogous to the $S4$ axiom $\Box p \rightarrow \Box\Box p$, which corresponded to transitivity of the alternativeness relation in the modal semantics. Bull's other axioms translate into conditions on 'cohistorical' and time in a similar way; we shall not bother here with the rather tedious details.

Rather more interesting would be axioms relating 'shrug' to 'cohistorical' and time; unfortunately we have been unable to think of any intuitively plausible ones. Thus, if two situations are epistemic alternatives (that is, $shrug(p, s_1, s_2)$) then they may or may not have the same time value (since we want to allow that p may not know what the time is), and they may or may not be cohistorical.

5.4 Logics and Theories of Action

The most fully developed theory in this area is von Wright's action logic described in his book *Norm and Action* (1963). Von Wright builds his logic on a rather unusual tense-logic of his own. The basis is a binary modal connective T, so that pTq, where p and q are propositions, means '$p, then q$'. Thus the action, for instance, of opening the window is: *(the window is closed) T (the window is open)*. The formal development of the calculus was taken a long way in the book cited above, but some problems of interpretation remained as Castaneda points out in his review (1965). In a more recent paper von Wright (1967) has altered and extended his formalism so as to answer these and other criticisms, and also has provided a sort of semantic theory based on the notion of a life-tree.

We know of no other attempts at constructing a single theory of actions which have reached such a degree of development, but there are several discussions of difficulties and surveys which seem important. Rescher (1967) discusses several topics very neatly, and Davidson (1967) also makes some cogent points. Davidson's main thesis is that, in order to translate statements involving actions into the predicate calculus, it appears necessary to allow actions as values of bound variables, that is (by Quine's test) as real individuals. The situation calculus of course follows this advice in that we allow quantification over strategies, which have actions as a special case. Also important are Simon's papers (1965, 1967) on command-logics. Simon's main purpose is to show that a special logic of commands is unnecessary, ordinary logic serving as the only deductive machinery; but this need not detain us here. He makes several points, most notably perhaps that agents are most of the time not performing actions, and that in fact they only stir to action when forced to by some outside interference. He has the particularly interesting example of a serial processor operating in a parallel-demand environment, and the resulting need for interrupts. Action logics such as von Wright's and ours do not distinguish between action and inaction, and we are not aware of any action-logic which has reached a stage of sophistication adequate to meet Simon's implied criticism.

There is a large body of purely philosophical writings on action, time, determinism, etc., most of which is irrelevant for present purposes. However, we mention two which have recently appeared and which seem interesting: a paper by Chisholm (1967) and another paper by Evans (1967), summarizing the recent discussion on the distinctions between states, performances and activities.

5.5 Other Topics

There are two other areas where some analysis of actions has been necessary: command-logics and logics and theories of obligation. For the former the best reference is Rescher's book (1966) which has an excellent bibliography. Note also Simon's counterarguments to some of Rescher's theses (Simon 1965, 1967). Simon proposes that no special logic of commands is necessary, commands being analyzed in the form 'bring it about that p!' for some proposition p, or, more generally, in the form 'bring it about that $P(x)$ by changing x!', where x is a *command* variable, that is, under the agent's control. The translations between commands and statements take place only in the context of a 'complete model', which specifies environmental constraints and defines the command variables. Rescher argues that these schemas for commands are inadequate to handle the *conditional command* 'when p, do q', which becomes 'bring it about that $(p \rightarrow q)$!': this, unlike the former, is satisfied by making p false.

There are many papers on the logic of obligation and permission. Von Wright's work is oriented in this direction; Castañeda has many papers on the subject and Anderson also has written extensively (his early influential report (1956) is especially worth reading). The review pages of the *Journal of Symbolic Logic* provide many other references. Until fairly recently these theories did not seem of very much relevance to logics of action, but in their new maturity they are beginning to be so.

5.6 Counterfactuals

There is, of course, a large literature on this ancient philosophical problem, almost none of which seems directly relevant to us. However, there is one recent theory, developed by Rescher (1964), which may be of use. Rescher's book is so clearly written that we shall not attempt a description of his theory here. The reader should be aware of Sosa's critical review (1967) which suggests some minor alterations.

The importance of this theory for us is that it suggests an alternative approach to the difficulty which we have referred to as the frame problem. In outline, this is as follows. One assumes, as a rule of procedure (or perhaps as a rule of inference), that when actions are performed, *all* propositional fluents which applied to the previous situation also apply to the new situation. This will often yield an inconsistent set of statements about the new situation; Rescher's theory provides a mechanism for restoring consistency in a rational way, and giving as a by-product those fluents which change in value as a result of performing the action. However, we have not investigated this in detail.

5.7 The Communication Process

We have not considered the problems of formally describing the process of communication in this paper, but it seems clear that they will have to be tackled eventually. Philosophical logicians have been spontaneously active here. The major work is Harrah's book (1963); Cresswell has written several papers on 'the logic of interrogatives', see for instance Cresswell (1965). Among other authors we may mention Åqvist (1965) and Belnap (1963); again the review pages of the *Journal of Symbolic Logic* will provide other references.

Acknowledgements

The research reported here was supported in part by the Advanced Research Projects Agency of the Office of the Secretary of Defense (SD-183), and in part by the Science Research Council (B/SR/2299).

6 References

Anderson, A.R. (1956). The formal analysis of normative systems. Reprinted in *The Logic of decision and action* (ed. Rescher, N.). Pittsburgh: University of Pittsburgh Press.

Åqvist, L. (1965). *A new approach to the logical theory of interrogatives, part I.* Uppsala: Uppsala Philosophical Association.

Barcan-Marcus, R.C. (1946). A functional calculus of the first order based on strict implication. *Journal of Symbolic Logic*, 11, 1-16.

Barcan-Marcus, R.C. (1963). Modalities and intensional languages. *Boston studies in the Philosophy of Science.* (ed. Wartofsky, W.). Dordrecht, Holland.

Belnap, N.D. (1963). *An analysis of questions.* Santa Monica.

Belnap, N.D. and Dunn, J.M. (1968). The substitution interpretation of the quantifiers. *Nous*, 2, 177-85.

Bull, R.A. (1968). An algebraic study of tense logics with linear time. *Journal of Symbolic Logic*, 33, 27-39.

Castañeda, H.N. (1965). The logic of change, action and norms. *Journal of Philosophy*, 62, 333-4.

Chisholm, R.M. (1963). The logic of knowing. *Journal of Philosophy*, 60, 773-95.

Chisholm, R.M. (1967). He could have done otherwise. *Journal of Philosophy*, 64, 409-17.

Church, A. (1956). *Introduction to Mathematical Logic.* Princeton: Princeton University Press.

Cresswell, M.J. (1965). The logic of interrogatives. *Formal systems and re-*

cursive functions. (ed. Crossley, J.M. and Dummett, M.A.E.). Amsterdam: North-Holland.

Davidson, D. (1967). The logical form of action sentences. *The logic of decision and action.* (ed. Rescher, N.). Pittsburgh: University of Pittsburgh Press.

Evans, C.O. (1967). States, activities and performances. *Australian Journal of Philosophy*, 45, 293-308.

Feys, R. (1965). *Modal Logics.* (ed. Dopp, J.). Louvain: Coll. de Logique Math. serie B.

Fogel, L.J., Owens, A.J. and Walsh, M.J. (1966). *Artificial Intelligence through simulated evolution.* New York: John Wiley.

Follesdal, D. (1967). Knowledge, identity and existence. *Theoria*, 33, 1-27.

Friedberg, R.M. (1958). A learning machine, part I. *IBM J. Res. Dev.*, 2, 2-13.

Friedberg, R.M., Dunham, B., and North, J.H. (1959). A learning machine, part II. *IBM J. Res. Dev.*, 3, 282-7.

Galanter, E. and Gerstenhaber, M. (1956). On thought: the extrinsic theory. *Psychological Review*, 63, 218-27.

Green, C. (1969). Theorem-proving by resolution as a basis for question-answering systems. *Machine Intelligence 4,* pp. 183-205 (eds. Meltzer, B. and Michie, D.). Edinburgh: Edinburgh University Press.

Harrah, D. (1963). *Communication: a logical model.* Cambridge, Massachusetts: MIT Press.

Hintikka, J. (1962). *Knowledge and belief: an introduction to the logic of two notions.* New York: Cornell University Press.

Hintikka, J. (1963). The modes of modality. *Acta Philosophica Fennica*, 16, 65-82.

Hintikka, J. (1967a). A program and a set of concepts for philosophical logic. *The Monist,* 51, 69-72.

Hintikka, J. (1967b). Existence and identity in epistemic contexts. *Theoria,* 32, 138-47.

Hintikka, J. (1967c). Individuals, possible worlds and epistemic logic. *Nous,* 1, 33-62.

Hintikka, J. (1977). Different constructions in terms of the basic epistemological verbs. *Contemporary Philosophy in Scandinavia* (eds. Olsen, R.E. and Paul, A. M.), Baltimore: The John Hopkins Press, 105-122.

Kanger, S. (1957). A note on quantification and modalities. *Theoria,* 23, 133-4.

Kripke, S. (1963a). Semantical considerations on modal logic. *Acta Philosophica Fennica,* 16, 83-94.

Kripke, S. (1963b). Semantical analysis of modal logic I. *Zeitschrift fur math. Logik und Grundlagen der Mathematik,* 9, 67-96.

Kripke, S. (1965). Semantical analysis of modal logic II. *The theory of models* (eds. Addison, Henkin and Tarski). Amsterdam: North-Holland.

Lewis, C.I. (1918). *A survey of symbolic logic.* Berkeley: University of California Press.

Manna, Z. (1968a). *Termination of algorithms.* Ph.D. Thesis, Carnegie-Mellon University.

Manna, Z. (1968b). *Formalization of properties of programs* Stanford Artificial Intelligence Report: Project Memo AI-64.

McCarthy, J. (1959). Programs with common sense. *Mechanization of thought processes,* Vol. I. London: Her Majesty's Stationery Office. (Reprinted in this volume, pp. 000–000).

McCarthy, J. (1962). Towards a mathematical science of computation. *Proc. IFIP Congress* 62. Amsterdam: North-Holland Press.

McCarthy, J. (1963). *Situations, actions and causal laws.* Stanford Artificial Intelligence Project: Memo 2.

Minsky, M. (1961). Steps towards artificial intelligence. *Proceedings of the I.R.E.,* 49, 8-30.

Newell, A., Shaw, V.C. and Simon, H.A. (1959). Report on a general problem-solving program. *Proceedings ICIP.* Paris:UNESCO House.

Newell, A. and Simon, H.A. (1961). GPS - a program that simulates human problem-solving. *Proceedings of a conference in learning automata.* Munich:

Oldenbourgh.

Newell, A. (1965). Limitations of the current stock of ideas about problem-solving. *Proceedings of a conference on Electronic Information Handling,* pp. 195-208 (eds. Kent, A. and Taulbee, O.). New York: Spartan.

Newell, A. and Ernst, C. (1965). The search for generality. *Proc. IFIP Congress 65.*

Pivar, M. and Finkelstein, M. (1964). *The Programming Language LISP: its operation and applications* (eds. Berkely, E.C. and Bobrow, D.G.). Cambridge, Massachusetts: MIT Press.

Prior, A.N. (1957). *Time and modality.* Oxford: Clarendon Press.

Prior, A.N. (1968). *Past, present and future.* Oxford: Clarendon Press.

Quine, W.V.O. (1964). Reference and modality. *From a logical point of view.* Cambridge, Massachusetts: Harvard University Press.

Rescher, N. (1964). *Hypothetical reasoning.* Amsterdam: North-Holland.

Rescher, N. (1966). *The logic of commands.* London: Routledge.

Rescher, N. (1967). Aspects of action. *The logic of decision and action* (ed. Rescher, N.). Pittsburgh: University of Pittsburgh Press.

Shannon, C. (1950). Programming a computer for playing chess. *Philosophical Magazine,* 41.

Simon, H.A. (1965). The logic of rational decision. *British Journal for the Philosophy of Science,* 16, 169-86.

Simon, H.A (1966). *On Reasoning about actions.* Carnegie Institute of Technology: Complex Information Processing Paper 87.

Simon, H.A. (1967). The logic of heuristic decision making. *The logic of decision and action* (ed. Rescher, N.). Pittsburgh: University of Pittsburgh Press.

Sosa, E. (1967). Hypothetical reasoning. *Journal of Philosophy,* 64, 293-305.

Turing, A.M. (1950). Computing machinery and intelligence. *Mind,* 59, 433-60.

von Wright, C.H. (1963). *Norm and action: a logical enquiry.* London: Routledge.

von Wright, C.H. (1967). The Logic of Action - a sketch. *The logic of decision and action* (ed. Rescher, N.). Pittsburgh:University of Pittsburgh Press.