SRI International

A FORMAL THEORY OF KNOWLEDGE AND ACTION

Technical Note 320

February 1984

By: Robert C. Moore, Staff Scientist
Artificial Intelligence Center
Computer Science and Technology Division

SRI Project 4488
SRI IR&D Project 500KZ

To appear in Formal Theories of the Commonsense World, J. R. Hobbs and R. C. Moore, eds., (Ablex Publishing Corp., Norwood, New Jersey, 1984).

The research reported herein was supported in part by the Air Force Office of Scientific Research under Contract No. F49620-82-K-0031. The views and conclusions expressed in this document are those of the author and should not be interpreted as necessarily representing the official policies or endorsements, either expressed of implied, of the Air Force Office of Scientific Research or the U.S. Government. This research was also made possible in part by a gift from the System Development Foundation as part of a coordinated research effort with the Center for the Study of Language and Information, Stanford University.

SRI International®
333 Ravenswood Ave. • Menlo Park, CA 94025
(415) 326-6200 • TWX: 910-373-2046 • Telex: 334-486

ABSTRACT

Most work on planning and problem solving within the field of artificial intelligence assumes that the agent has complete knowledge of all relevant aspects of the problem domain and problem situation. In the real world, however, planning and acting must frequently be performed without complete knowledge. This imposes two additional burdens on an intelligent agent trying to act effectively. First, when the agent entertains a plan for achieving some goal, he must consider not only whether the physical prerequisites of the plan have been satisfied, but also whether he has all the information necessary to carry out the plan. Second, he must be able to reason about what he can do to obtain necessary information that he lacks. In this paper, we present a theory of action in which these problems are taken into account, showing how to formalize both the knowledge prerequisites of action and the effects of action on knowledge.

CONTENTS

I THE INTERPLAY OF KNOWLEDGE AND ACTION

Planning sequences of actions and reasoning about their effects is one of the most thoroughly studied areas within artificial intelligence (AI). Relatively little attention has been paid, however, to the important role that an agent's knowledge plays in planning and acting to achieve a goal. Virtually all AI planning systems are designed to operate with complete knowledge of all relevant aspects of the problem domain and problem situation. Often any statement that cannot be inferred to be true is assumed to be false. In the real world, however, planning and acting must frequently be performed without complete knowledge of the situation.

This imposes two additional burdens on an intelligent agent trying to act effectively. First, when the agent entertains a plan for achieving some goal, he must consider not only whether the physical prerequisites of the plan have been satisfied, but also whether he has all the information necessary to carry out the plan. Second, he must be able to reason about what he can do to obtain necessary information that he lacks. AI planning systems are usually based on the assumption that, if there is an action an agent is physically able to perform, and carrying out that action would result in the achievement of a goal P, then the agent can achieve P. With goals such as opening a safe,

however, there are actions that any human agent of normal abilities is physically capable of performing that would result in achievement of the goal (in this case, dialing the combination of the safe), but it would be highly misleading to claim that an agent could open a safe simply by dialing the combination unless he actually knew that combination. On the other hand, if the agent had a piece of paper on which the combination of the safe was written, he could open the safe by reading what was on the piece of paper and then dialing the combination, even if he did not know it previously.

In this paper, we will describe a formal theory of knowledge and action that is based on a general understanding of the relationship between the two.[1] The question of generality is somewhat problematical, since different actions obviously have different prerequisites and results that involve knowledge. What we will try to do is to set up a formalism in which very general conclusions can be drawn, once a certain minimum of information has been provided concerning the relation between specific actions and the knowledge of agents.

To see what this amounts to, consider the notion of a test. The essence of a test is that it is an action with a directly observable result that depends conditionally on an unobservable precondition. In the use of litmus paper to test the pH of a solution, the observable result is whether the paper has turned red or blue, and the unobservable precondition is whether the solution is acid or alkaline. What makes

such a test useful for acquiring knowledge is that the agent can infer whether the solution is acid or alkaline on the basis of his knowledge of the behavior of litmus paper and the observed color of the paper. When one is performing a test, it is this inferred knowledge, rather than what is directly observed, that is of primary interest.

If we tried to formalize the results of such a test by making simple assertions about what the agent knows subsequent to the action, we would have to include the result that the agent knows whether the solution is acid or alkaline as a separate assertion from the result that he knows the color of the paper. If we did this, however, we would completely miss the point that knowledge of the pH of the solution is inferred from other knowledge, rather than being a direct observation. In effect, we would be stipulating what actions can be used as tests, rather than creating a formalism within which we can infer what actions can be used as tests.

If we want a formal theory of how an agent's state of knowledge is changed by his performing a test, we have to represent and be able to draw inferences from the agent's having several independent pieces of information. Obviously, we have to represent that, after the test is performed, the agent knows the observable result. Furthermore, we have to represent the fact that he knows that the test has been performed. If he just walks into the room and sees the litmus paper on the table, he will know what color it is, but, unless he knows its recent history, he will not have gained any knowledge about the acidity of the solution.

We also need to represent the fact that the agent understands how the test works; that is, he knows how the observable result of the action depends on the unobservable precondition. Even if he sees the litmus paper put into the solution and then sees the paper change color, he still will not know whether the solution is acid or alkaline unless he knows how the color of the paper is related to the acidity of the solution. Finally, we must be able to infer that, if the agent knows (i) that the test took place, (ii) the observable result of the test, and (iii) how the observable result depends on the unobservable precondition, then he will know the unobservable precondition. Thus we must know enough about knowledge to tell us when an agent's knowing a certain collection of facts implies that he knows other facts as well.

From the preceding discussion, we can conclude that any formalism that enables us to draw inferences about tests at this level of detail must be able to represent the following types of assertions:

(1) After A performs ACT, he knows whether Q is true.

(2) After A performs ACT, he knows that he has just performed ACT.

(3) A knows that Q will be true after he performs ACT if and only if P is true now.

Moreover, in order to infer what information an agent will gain as a result of performing a test, the formalism must embody, or be able to represent, general principles sufficient to conclude the following:

(4) If (1), (2), and (3) are true, then, after performing ACT, A will know whether P was true before he performed ACT.

It is important to emphasize that any work on these problems that is to be of real value must seek to elicit general principles. For instance, it would be possible to represent (1), (2), and (3) in an arbitrary, ad hoc manner and to add an axiom that explicitly states (4), thereby "capturing" the notion of a test. Such an approach, however, would simply restate the superficial observations put forth in this discussion. Our goal in this paper is to describe a formalism in which specific facts like (4) follow from the most basic principles of reasoning about knowledge and action.

II FORMAL THEORIES OF KNOWLEDGE

A. A Modal Logic of Knowledge

Since formalisms for reasoning about action have been studied extensively in AI, while formalisms for reasoning about knowledge have not, we will first address the problems of reasoning about knowledge. In Section III we will see that the formalism that we are led to as a solution to these problems turns out to be well suited to developing an integrated theory of knowledge and action.

The first step in devising a formalism for reasoning about knowledge is to decide what general properties of knowledge we want that formalism to capture. The properties of knowledge in which we will be most interested are those that are relevant to planning and acting. One such property is that anything that is known by someone must be true. If P is false, we would not want to say that anyone knows P. It might be that someone believes P or that someone believes he knows P, but it simply could not be the case that anyone knows P. This is, of course, a major difference between knowledge and belief. If we say that someone believes P, we are not committed to saying that P is either true or false, but if we say that someone knows P, we are committed to the truth of P. The reason that this distinction is important for planning and

acting is simply that, for an agent to achieve his goals, the beliefs on which he bases his actions must generally be true. After all, merely believing that performing a certain action will bring about a desired goal is not sufficient for being able to achieve the goal; the action must actually have the intended effect.

Another principle that turns out to be important for planning is that, if someone knows something, he knows that he knows it. This principle is often required for reasoning about plans consisting of several steps. Suppose an agent plans to use ACT_1 to achieve his goal, but, in order to perform ACT_1 he needs to know whether P is true and whether Q is true. Suppose, further, that he already knows that P is true and that he can find out whether Q is true by performing ACT_2. The agent needs to be able to reason that, after performing ACT_2, he will know whether P is true and whether Q is true. He knows that he will know whether Q is true because he understands the effects of ACT_2, but how does be know that he will know whether P is true? Presumably it works something like this: he knows that P is true, so he knows that he knows that P is true. If he knows how ACT_2 affects P, he knows that he will know whether P is true after he performs ACT_2. The key step in this argument is an instance of the principle that, if someone knows something, he knows that he knows it.

It might seem that we would also want to have the principle that, if someone does not know something, he knows that he does not know it--but this turns out to be false. Suppose that A believes that P, but P is not true. Since P is false, A certainly does not know that P, but it is highly unlikely that he knows that he does not know, since he thinks that P is true.

Probably the most important fact about knowledge that we will want to capture is that agents can reason on the basis of their knowledge. All our examples depend on the assumption that, if an agent trying to solve a problem has all the relevant information, he will apply his knowledge to produce a solution. This creates a difficulty for us, however, since agents (at least human ones) are not, in fact, aware of all the logical consequences of their knowledge. The trouble is that we can never be sure which of the inferences an agent could draw, he actually will. The principle people normally use in reasoning about what other people know seems to be something like this: if we can infer that something is a consequence of what someone knows, then, lacking information to the contrary, we will assume that the other person can draw the same inference.

This suggests the adoption some sort of "default rule" (Reiter, 1980) for reasoning about what inferences agents actually draw, but, for the purposes of this study, we will make the simplifying assumption that agents actually do draw all logically valid inferences from their knowledge. We can regard this as the epistemological version of the

"frictionless case" in classical physics. For a more general framework in which weaker assumptions about the deductive abilities of agents can be expressed, see the work of Konolige (1984).

Finally, we will need to include the fact that these basic properties of knowledge are themselves common knowledge. By this we mean that everyone knows them, and everyone knows that everyone knows them, and everyone knows that everyone knows that everyone knows them, ad infinitum. This type of principle is obviously needed when reasoning about what someone knows about what someone else knows, but it is also important in planning, because an agent must be able to reason about what he will know at various times in the future. In such a case, his "future self" is analogous to another agent.

In his pioneering work on the logic of knowledge and belief, Hintikka (1962) presents a formalism that captures all these properties. We will define a formal logic based on Hintikka's ideas, but modified somewhat to be more compatible with the additional ideas of this paper. So, what follows is similar to the logic developed by Hintikka in spirit, but not in detail.

The language we will use initially is that of propositional logic, augmented by an operator KNOW and terms denoting agents. The formula KNOW(A,P) is interpreted to mean that the agent denoted by the term A knows the proposition expressed by the formula P. So, if JOHN denotes John and LIKES(BILL,MARY) means that Bill likes Mary,

KNOW(JOHN,LIKES(BILL,MARY)) means that John knows that Bill likes Mary. The axioms of the logic are inductively defined as all instances of the following schemata:

M1. P, such that P is an axiom of ordinary propositional logic

M2. KNOW(A,P) ⊃ P

M3. KNOW(A,P) ⊃ KNOW(A,KNOW(A,P))

M4. KNOW(A,(P ⊃ Q)) ⊃ (KNOW(A,P) ⊃ KNOW(A,Q))

closed under the principle that

M5. If P is an axiom, then KNOW(A,P) is an axiom.

The closure of the axioms under the inference rule modus ponens (from (P ⊃ Q) and P, infer Q) defines the theorems of the system. This system is very similar to those studied in modal logic. In fact, if A is held fixed, the resulting system is isomorphic to the modal logic S4 (Hughes and Cresswell, 1968). We will refer to this system as the modal logic of knowledge.

These axioms formalize in a straightforward way the principles for reasoning about knowledge that we have discussed. M2 says that anything that is known is true. M3 says that, if someone knows something, he knows that he knows it. M4 says that, if someone knows a formula P and a formula of the form (P ⊃ Q), then he knows the corresponding formula Q. That is, everyone can (and does) apply modus ponens. M5 guarantees that the axioms are common knowledge. It first applies to M1-M4, which

says that everyone knows the basic facts about knowledge; however, since it also applies to its own output, we get axioms stating that everyone knows that everyone knows, etc. Since M5 applies to the axioms of propositional logic (M1), we can infer that everyone knows the facts they represent. Furthermore, because modus ponens is the only inference rule needed in propositional logic, the presence of M4 will enable us to infer that an agent knows any propositional consequence of his knowledge.

B. A Possible-World Analysis of Knowledge

We could try to use the modal logic of knowledge directly in a computational system for reasoning about knowledge and action, but, as we have argued elsewhere (Moore, 1980), all the obvious ways of doing this encounter difficulties. (Konolige's recent work (1984) suggests some new, more promising possibilities, but some important questions remain to be resolved.) There may well be solutions to these problems, but it turns out that they can be circumvented entirely by changing the language we use to describe what agents know. Instead of talking about the individual propositions that an agent knows, we will talk about what states of affairs are compatible with what he knows. In philosophy, these states of affairs are usually called "possible worlds," so we will adopt that term here as well.

This shift to describing knowledge in terms of possible worlds is based on a rich and elegant formal semantics for systems like our modal

logic of knowledge, which was developed by Hintikka (1962, 1971) in his work on knowledge and belief. The advantages of this approach are that it can be formalized within ordinary first-order classical logic in a way that permits the use of standard automatic-deduction techniques in a reasonably efficient manner[2] and that, moreover, it generalizes nicely to an integrated theory for describing the effects of actions on the agent's knowledge.

Possible-world semantics was first developed for the logic of necessity and possibility. From an intuitive standpoint, a possible world may be thought of as a set of circumstances that might have been true in the actual world. Kripke (1963) introduced the idea that a world should be regarded as possible, not absolutely, but only relative to other worlds. That is, the world W_1 might be a possible alternative to W_2, but not to W_3. The relation of one world's being a possible alternative to another is called the accessibility relation. Kripke then proved that the differences among some of the most important axiom systems for modal logic corresponded exactly to certain restrictions on the accessibility relation of the possible-world models of those systems. These results are reviewed in Kripke (1971). Concurrently with these developments, Hintikka (1962) published the first of his writings on the logic of knowledge and belief, which included a model theory that resembled Kripke's possible-world semantics. Hintikka's original semantics was done in terms of sets of sentences, which he

called model sets, rather than possible worlds. Later (Hintikka, 1971), however, he recast his semantics using Kripke's concepts, and it is that formulation we will use here.

Kripke's semantics for necessity and possibility can be converted into Hintikka's semantics for knowledge by changing the interpretation of the accessibility relation. To analyze statements of the form KNOW(A,P), we will introduce a relation K, such that $K(A,W_1,W_2)$ means that the possible world W_2 is compatible or consistent with what A knows in the possible world W_1. In other words, for all that A knows in W_1, he might just as well be in W_2. It is the set of worlds $\{w_2 \mid K(A,W_1,w_2)\}$ that we will use to characterize what A knows in W_1. We will discuss A's knowledge in W_1 in terms of this set, the set of states of affairs that are consistent with his knowledge in W_1, rather than in terms of the set of propositions he knows. For the present, let us assume that the first argument position of K admits the same set of terms as the first argument position of KNOW. When we consider quantifiers and equality, we will have to modify this assumption, but it will do for now.

Introducing K is the key move in our analysis of statements about knowedge, so understanding what K means is particularly important. To

illustrate, suppose that in the actual world--call it W_0--A knows that P, but does not know whether Q. If W_1 is a world where P is false, then W_1 is not compatible with what A knows in W_0; hence we would have $\neg K(A,W_0,W_1)$. Suppose that W_2 and W_3 are compatible with everything A knows, but that Q is true in W_2 and false in W_3. Since A does not know whether Q is true, for all he knows, he might be in either W_2 or W_3 instead of W_0. Hence, we would have both $K(A,W_0,W_2)$ and $K(A,W_0,W_3)$. This is depicted graphically in Figure 1.

Some of the properties of knowledge can be captured by putting constraints on the accessibility relation K. For instance, requiring that the actual world W_0 be compatible with what each knower knows in W_0, i.e., $\forall a\,(K(a,W_0,W_0))$, is equivalent to saying that anything that is known is true. That is, if the actual world is compatible with what everyone [actually] knows, then no one has any false knowledge. This corresponds to the modal axiom M2.

The definition of K implies that, if A knows that P in W_0, then P must be true in every world W_1 such that $K(A,W_0,W_1)$. To capture the fact that agents can reason with their knowledge, we will assume the

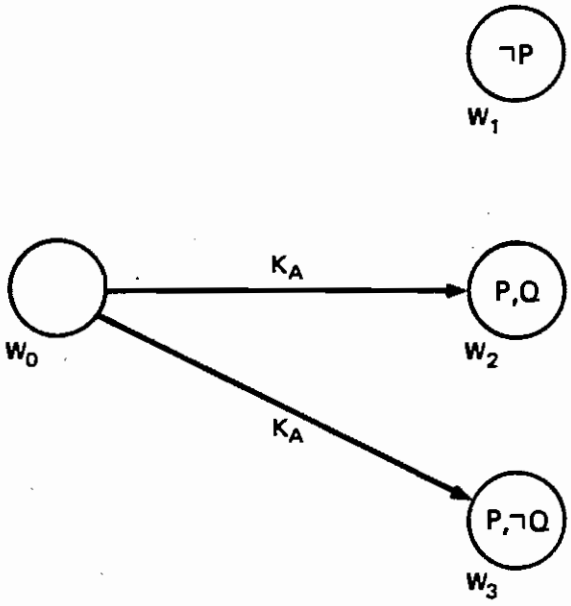

FIGURE 1 "A KNOWS THAT P"
"A DOESN'T KNOW WHETHER Q"

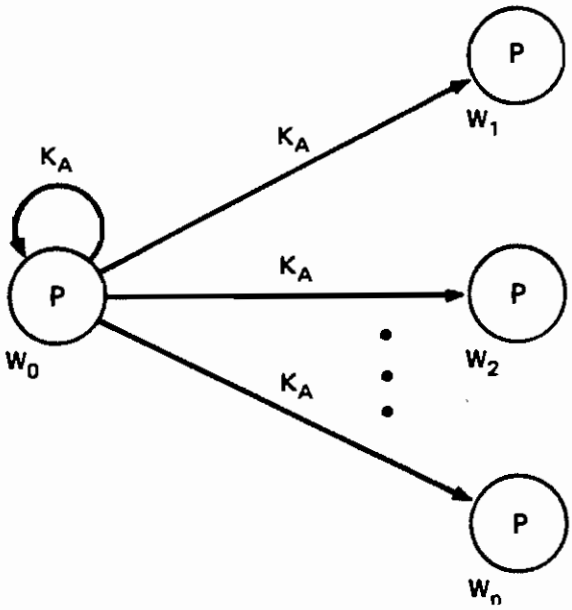

FIGURE 2 "P IS TRUE IN EVERY WORLD THAT IS COMPATIBLE WITH WHAT A KNOWS"

converse is also true. That is, we assume that, if P is true in every world W_1 such that $K(A,W_0,W_1)$, then A knows that P in W_0. (See Figure 2.) This principle is the model-theoretic analogue of axiom M4 in the modal logic of knowledge. To see that this is so, suppose that A knows that P and that $(P \supset Q)$. Therefore, P and $(P \supset Q)$ are both true in every world that is compatible with what A knows. If this is the case, though, then Q must be true in every world that is compatible with what A knows. By our assumption, therefore, we conclude that A knows that Q.

Since this assumption, like M4, is equivalent to saying that an agent knows all the logical consequences of his knowledge, it should be interpreted only as a default rule. In a particular instance, the fact that P follows from A's knowledge would be a justification for concluding that A knows P. However, we should be prepared to retract the conclusion that A knows P in the face of stronger evidence to the contrary.

With this assumption, we can get the effect of M3--the axiom stating that, if someone knows something, he knows that he knows it--by requiring that, for any W_1 and W_2, if W_1 is compatible with what A knows in W_0 and W_2 is compatible with what A knows in W_1, then W_2 is compatible with what A knows in W_0. Formally expressed, this is

$$\forall a_1, w_1, w_2 \, (K(a_1, W_1, w_2) \supset (K(a_1, w_1, w_2) \supset K(a_1, W_1, w_2)))$$

By our previous assumption, the facts that A knows are those that are true in every world that is compatible with what A knows in the actual world. Furthermore, the facts that A knows that he knows are those that are true in every world that is compatible with what he knows in every world that is compatible with what he knows in the actual world. By the constraint we have just proposed, however, all these worlds must also be compatible with what A knows in the actual world (see Figure 3), so, if A knows that P, he knows that he knows that P.

Finally, we can get the effect of M5, the principle that the basic facts about knowledge are themselves common knowledge, by generalizing these constraints so that they hold not only for the actual world, but for all possible worlds. This follows from the fact that, if these constraints hold for all worlds, they hold for all worlds that are compatible with what anyone knows in the actual world; they also hold for all worlds that are compatible with what anyone knows in all worlds that are compatible with what anyone knows in the actual world, etc. Therefore, everyone knows the facts about knowledge that are represented by the constraints, and everyone knows that everyone knows, etc. Note that this generalization has the effect that the constraint corresponding to M2 becomes the requirement that, for a given knower, K is reflexive, while the constraint corresponding to M3 becomes the requirement that, for a given knower, K is transitive.

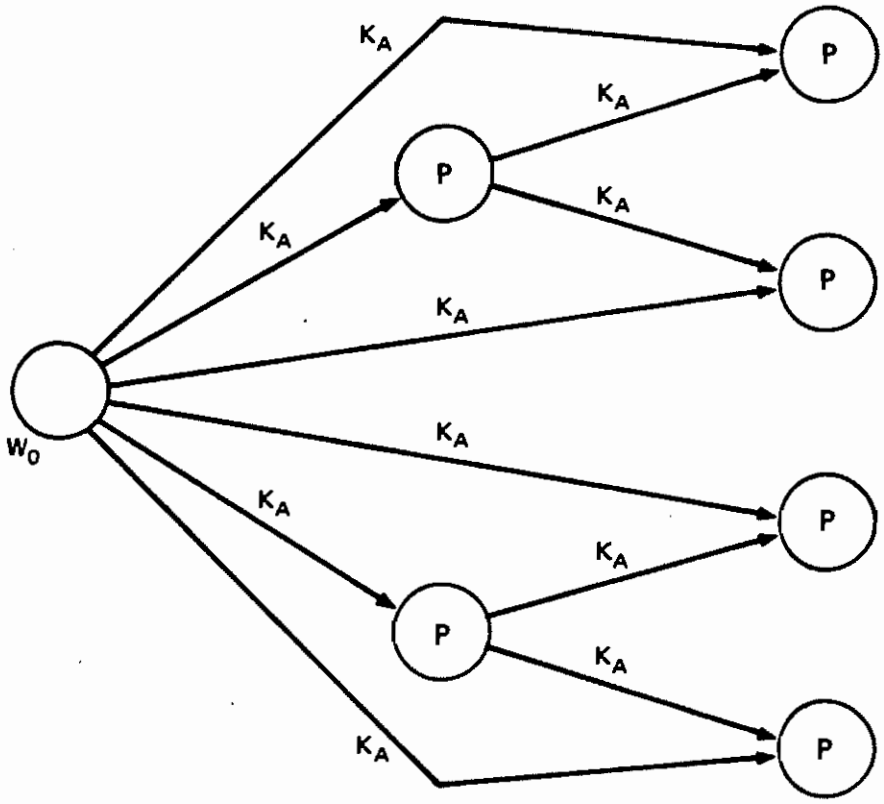

FIGURE 3 "IF A KNOWS THAT P, THEN HE KNOWS THAT HE KNOWS THAT P"

Analyzing knowledge in terms of possible worlds gives us a very nice treatment of knowledge about knowledge. Suppose A knows that B knows that P. Then, if the actual world is W_0, in any world W_1 such that $K(A,W_0,W_1)$, B knows that P. We now continue the analysis relative to W_1, so that, in any world W_2 such that $K(B,W_1,W_2)$, P is true. Putting both stages together, we obtain the analysis that, for any worlds W_1 and W_2 such that $K(A,W_0,W_1)$ and $K(B,W_1,W_2)$, P is true in W_2. (See Figure 4.)

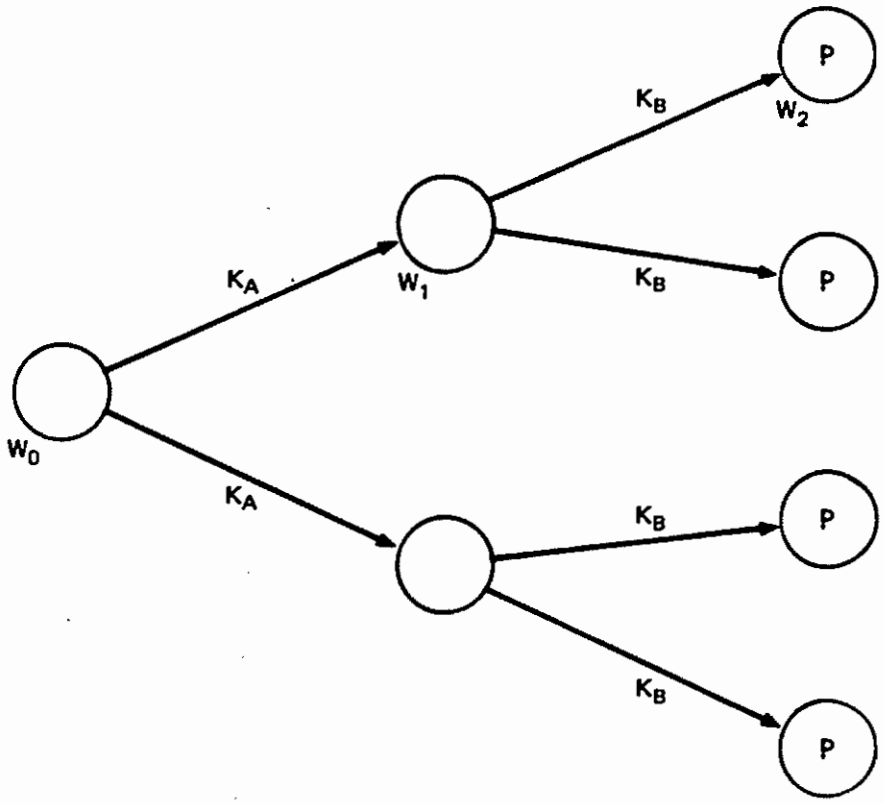

FIGURE 4 "A KNOWS THAT B KNOWS THAT P"

Given these constraints and assumptions, whenever we want to assert or deduce something that would be expressed in the modal logic of knowledge by KNOW(A,P), we can instead assert or deduce that P is true in every world that is compatible with what A knows. We can express this in ordinary first-order logic, by treating possible worlds as individuals (in the logical sense), so that K is just an ordinary relation. We will therefore introduce an operator T such that T(W,P) means that the formula P is true in the possible world W. If we let W_0 denote the actual world, we can convert the assertion KNOW(A,P) into

$$\forall w_1 (K(A, W_0, w_1) \supset T(w_1, P))$$

It may seem that we have not made any real progress, since, although we have gotten rid of one nonstandard operator, KNOW, we have introduced another one, T. However, T has an important property that KNOW does not. Namely, T "distributes" over ordinary logical operators. In other words, $\neg P$ is true in W just in case P is not true in W, $(P \vee Q)$ is true in W just in case P is true in W or Q is true in W, and so on. We might say that T is extensional, relative to a possible world. This means that we can transform any formula so that T is applied only to atomic formulas. We can then turn T into an ordinary first-order relation by treating all the nonintensional atomic formulas as names of atomic propositions, or we can get rid of T by replacing the atomic formulas with predicates on possible worlds. This is no loss to the expressive power of the language, since, where we would have previously asserted P, we now simply assert $T(W_0, P)$ or $P(W_0)$ instead.

C. Knowledge, Equality, and Quantification

The formalization of knowledge presented so far is purely propositional; a number of additional problems arise when we attempt to extend the theory to handle equality and quantification. For instance, as Frege (1949) pointed out, attibutions of knowledge and belief lead to violations of the principle of equality substitution. We are not entitled to infer KNOW(A,P(C)) from B = C and KNOW(A,P(B)) because A might not know that the identity holds.

The possible-world analysis of knowledge provides a very neat solution to this problem, once we realize that a term can denote different objects in different possible worlds. For instance, if B is the expression "the number of planets" and C is "nine," then, although B = C is true in the actual world, it would be false in a world in which there was a tenth planet. Thus, we will say that an equality statement such as B = C is true in a possible world W just in case the denotation of the term B in W is the same as the denotation of the term C in W. This is a special case of the more general rule that a formula of the form $P(A_1,\ldots,A_n)$ is true in W just in case the tuple consisting of the denotations in W of the terms $A_1,\ldots,A_n$ is in the extension in W of the relation expressed by P, provided that we fix the interpretation of = in all possible worlds to be the identity relation.

Given this interpretation, the inference of KNOW(A,P(C)) from B = C and KNOW(A,P(B)) will be blocked (as it should be). To infer KNOW(A,P(C)) from KNOW(A,P(B)) by identity substitution, we would have to know that B and C denote the same object in every world compatible with what A knows, but the truth of B = C guarantees only that they denote the same object in the actual world. On the other hand, if KNOW(A,P(B)) and KNOW(A,(B`= C)) are both true, then in all worlds that are compatible with what A knows, the denotation of B is in the extension of P and is the same as the denotation of C; hence, the denotation of C is in the extension of P. From this we can infer that KNOW(A,P(C)) is true.

The introduction of quantifiers also causes problems. To modify a famous example from Quine (1971), consider the sentence "Ralph knows that someone is a spy." This sentence has at least two interpretations. One is that Ralph knows that there is at least one person who is a spy, although he may have no idea who that person is. The other interpretation is that there is a particular person whom Ralph knows to be a spy. As Quine says (1971, p. 102), "The difference is vast; indeed, if Ralph is like most of us, [the first] is true and [the second] is false." This ambiguity was explained by Russell (1949) as a difference of scope. The idea is that indefinite noun phrases such as "someone" can be analyzed in context by paraphrasing sentences of the form P("someone") as "There exists a person x such that P(x)," or, more formally, $\exists x(\text{PERSON}(x) \wedge P(x))$. Russell goes on to point out that, in sentences of the form "A knows that someone is a P," the rule for eliminating "someone" can be applied to either the whole sentence or only the subordinate clause, "someone is a P." Applying this observation to "Ralph knows that someone is a spy," gives us the following two formal representations:

(1) $\text{KNOW}(\text{RALPH}, \exists x(\text{PERSON}(x) \wedge \text{SPY}(x)))$

(2) $\exists x(\text{PERSON}(x) \wedge \text{KNOW}(\text{RALPH}, \text{SPY}(x)))$

The most natural English paraphrases of these formulas are "Ralph knows that there is a person who is a spy," and "There is a person who

Ralph knows is a spy." These seem to correspond pretty well to the two interpretations of the original sentence. So, the ambiguity in the original sentence is mapped into an uncertainty as to the scope of the operator KNOW relative to the existential quantifier introduced by the indefinite description "someone."

Following a suggestion of Hintikka (1962), we can use a formula similar to (2) to express the fact that someone knows who or what something is. He points out that a sentence of the form "A knows who (or what) B is" intuitively seems to be equivalent to "there is someone (or something) that A knows to be B. But this can be represented formally as $\exists x(KNOW(A,(x = B)))$. To take a specific example, "John knows who the President is" can be paraphrased as "There is someone whom John knows to be the President," which can be represented by

(3) $\exists x(KNOW(JOHN,(x = PRESIDENT)))$

In (1), KNOW may still be regarded as a purely propositional operator, although the proposition to which it is applied now has a quantifier in it. Put another way, KNOW still is used simply to express a relation between a knower and the proposition he knows. But (2) and (3) are not so simple. In these formulas there is a quantified variable that, although bound outside the scope of the operator KNOW, has an occurrence inside; this is sometimes called "quantifying in." Quantifying into knowledge and belief contexts is frequently held to pose serious problems of interpretation. Quine (1971), for instance,

holds that it is unintelligible, because we have not specified what proposition is known unless we say what description is used to fix the value of the quantified variable.

The possible-world analysis, however, provides us with a very natural interpretation of quantifying in. We keep the standard interpretation that $\exists x(P(x))$ is true just in case there is some value for x that satisfies P. If P is KNOW(A,Q(x)), then a value for x satisfies P(x) just in case that value satisfies Q(x) in every world that is compatible with what A knows. So (2) is satisfied if there is a particular person who is a spy in every world that is compatible with what A knows. That is, in every such world the same person is a spy. On the other hand, (1) is satisfied if, in every world compatible with what A knows, there is some person who is a spy, but it does not have to be the same one in each case.

Note that the difference between (1) and (2) has been transformed from a difference in the relative scopes of an existential quantifier and the operator KNOW to a difference in the relative scopes of an existential and a universal quantifier (the "every" in "every possible world compatible with..."). Recall from ordinary first-order logic that $\exists x(\forall y(P(x,y)))$ entails $\forall y(\exists x(P(x,y)))$, but not vice versa. The possible-world analysis, then, implies that we should be able to infer "Ralph knows that there is a spy," from "There is someone Ralph knows to be a spy," as indeed we can.

When we look at how this analysis applies to our representation for "knowing who," we get a particularly satisfying picture. We said that A knows who B is means that there is someone whom A knows to be B. If we analyze this, we conclude that there is a particular individual who is B in every world that is compatible with what A knows. Suppose this were not the case, and that, in some of the worlds compatible with what A knows, one person is B, whereas in the other worlds, some other person is B. In other words, for all that A knows, either of these two people might be B. But this is exactly what we mean when we say that A does not know who B is! Basically, the possible-world view gives us the very natural picture that A knows who B is if A has narrowed the possibilities for B down to a single individual.

Another consequence of this analysis worth noting is that, if A knows who B is and A knows who C is, we can conclude that A knows whether B = C. If A knows who B is and who C is, then B has the the same denotation in all the worlds that are compatible with what A knows, and this is also true for C. Since, in all these worlds, B and C each have only one denotation, they either denote the same thing everywhere or denote different things everywhere. Thus, either B = C is true in every world compatible with what A knows or B $\neq$ C is. From this we can infer that either A knows that B and C are the same individual or that they are not.

We now have a coherent account of quantifying in that is not framed in terms of knowing particular propositions. Still, in some cases

knowing a certain proposition counts as knowing something that would be expressed by quantifying in. For instance, the proposition that John knows that 321-1234 is Bill's telephone number might be represented as

(4) KNOW(JOHN,(321-1234 = PHONE-NUM(BILL))),

which does not involve quantifying in. We would want to be able to infer from this, however, that John knows what Bill's telephone number is, which would be represented as

(5) $\exists$x(KNOW(JOHN,(x = PHONE-NUM(BILL)))).

It might seem that (5) can be derived from (4) simply by the logical principle of existential generalization, but that principle is not always valid in knowledge contexts. Suppose that (4) were not true, but that instead John simply knew that Mary and Bill had the same telephone number. We could represent this as

(6) KNOW(JOHN,(PHONE-NUM(MARY) = PHONE-NUM(BILL))).

It is clear that we would not want to infer from (6) that John knows what Bill's telephone number is--yet, if existential generalization were universally valid in knowledge contexts, this inference would go through.

It therefore seems that, in knowledge contexts, existential generalization can be applied to some referring expressions ("321-1234"), but not to others ("Mary's telephone number"). We will call the

expressions to which existential generalization can be applied standard identifiers, since they seem to be the ones an agent would use to identify an object for another agent. That is, "321-1234" is the kind of answer that would always be appropriate for telling someone what John's telephone number is, whereas "Mary's telephone number," as a general rule, would not.[3]

In terms of possible worlds, standard identifiers have a very straightforward interpretation. Standard identifiers are simply terms that have the same denotation in every possible world. Following Kripke (1972), we will call terms that have the same denotation in every possible world rigid designators. The conclusion that standard identifiers are rigid designators seems inescapable. If a particular expression can always be used by an agent to identify its referent for any other agent, then there must not be any possible circumstances under which it could refer to something else. Otherwise, the first agent could not be sure that the second was in a position to rule out those other possibilities.

The validity of existential generalization for standard identifiers follows immediately from their identification with rigid designators. The possible-world analysis of KNOW(A,P(B)) is that, in every world compatible with what A knows, the denotation of B in that world is in the extension of P in that world. Existential generalization fails in general because we are unable to conclude that there is any particular

individual that is in the extension of P in all the relevant worlds. If B is a rigid designator, however, the denotation of B is the same in every world. Consequently, it is the same in every world compatible with what A knows, and that denotation is an individual that is in the extension of P in all those worlds.

There are a few more observations to be made about standard identifiers and rigid designators. First, in describing standard identifiers we assumed that everyone knew what they referred to. Identifying them with rigid designators makes the stronger claim that what they refer to is common knowledge. That is, not only does everyone know what a particular standard identifier denotes, but everyone knows that everyone knows, etc. Second, although it is natural to think of any individual having a unique standard identifier, this is not required by our theory. What the theory does require is that, if there are two standard identifiers for the same individual, it should be common knowledge that they denote the same individual.

III FORMALIZING THE POSSIBLE-WORLD ANALYSIS OF KNOWLEDGE

A. Object Language and Metalanguage

As we indicated above, the analysis of knowledge in terms of possible worlds can be formalized completely within first-order logic by admitting possible worlds into the domain of quantification and making the extension of every expression depend on the possible world in which it is evaluated. For example, the possible-world analysis of "A knows who B is" would be as follows: There is some individual x such that, in every world w_1 that is compatible with what the agent who is A in the actual world knows in the actual world, x is B in w_1. This means that in our formal theory we translate the formula of the modal logic of knowledge,

$$\exists x(KNOW(A,(x = B))),$$

into the first-order formula,

$$\exists x(\forall w_1 (K(A(W_0),W_0,w_1) \supset (x = B(w_1)))).$$

One convenient way of stating the translation rules precisely is to axiomatize them in our first-order theory of knowledge. This can be

done by introducing terms to denote formulas of the modal logic of knowledge (which we will henceforth call the object language) and axiomatizing a truth definition for those formulas in a first-order language that talks about possible worlds (the metalanguage). This has the advantage of letting us use either the modal language or the possible-world language--whichever is more convenient for a particular purpose--while rigorously defining the connection between the two.

The typical method of representing expressions of one formal language in another is to use string operations like concatenation or list operations like CONS in LISP, so that the conjunction of P and Q might be represented by something like CONS(P,CONS('∧,CONS(Q,NIL))), which could be abbreviated LIST(P,'∧,Q). This would be interpreted as a list whose elements are P followed by the conjunction symbol followed by Q. Thus, the metalanguage expression CONS(P,CONS('∧,CONS(Q,NIL))) would denote the object language expression (P ∧ Q). McCarthy (1962) has devised a much more elegant way to do the encoding, however. For purposes of semantic interpretation of the object language, which is what we want to do, the details of the syntax of that language are largely irrelevant. In particular, the only thing we need to know about the syntax of conjunctions is that there is some way of taking P and Q and producing the conjunction of P and Q. We can represent this by having a function AND such that AND(P,Q) denotes the conjunction of P and Q. To use McCarthy's term, AND(P,Q) is an abstract syntax for representing the conjunction of P and Q.

We will represent object language variables and constants by metalanguage constants; we will use metalanguage functions in an abstract syntax to represent object language predicates, functions, and sentence operators. For example, we will represent the object lanaguage formula KNOW(JOHN,∃x(P(x))) by the metalanguage term KNOW(JOHN,EXIST(X,P(X))), where JOHN and X are metalanguage constants, and KNOW, EXIST, and P are metalanguage functions.

Since KNOW(JOHN,EXIST(X,P(X))) is a term, if we want to say that the object language formula it denotes is true, we have to do so explicitly by means of a metalanguage predicate TRUE:

TRUE(KNOW(JOHN,EXIST(X,P(X)))).

In the possible-world analysis of statements about knowledge, however, an object language formula is not absolutely true, but only relative to a possible world. Hence, TRUE expresses not absolute truth, but truth in the actual world, which we will denote by W_0. Thus, our first axiom is

L1. $\forall p_1 (TRUE(p_1) \equiv T(W_0, p_1))$,

where T(W,P) means that formula P is true in world W. To simplify the axioms, we will let the metalanguage be a many-sorted logic, with different sorts assigned to differents sets of variables. For instance, the variables $w_1, w_2, \ldots$ will range over possible worlds; $x_1, x_2, \ldots$

will range over individuals in the domain of the object language; and a_1, a_2,... will range over agents. Because we are axiomatizing the object language itself, we will need several sorts for different types of object language expressions. The variables p_1, p_2,... will range over object language formulas, and t_1,t_2,... will range over object language terms.

The recursive definition of T for the propositional part of the object language is as follows:

L2. $\forall w_1 \forall p_1 \forall p_2 (T(w_1, AND(p_1, p_2)) \equiv (T(w_1, p_1) \wedge T(w_1, p_2)))$

L3. $\forall w_1 \forall p_1 \forall p_2 (T(w_1, OR(p_1, p_2)) \equiv (T(w_1, p_1) \vee T(w_1, p_2)))$

L4. $\forall w_1 \forall p_1 \forall p_2 (T(w_1, IMP(p_1, p_2)) \equiv (T(w_1, p_1) \supset T(w_1, p_2)))$

L5. $\forall w_1 \forall p_1 \forall p_2 (T(w_1, IFF(p_1, p_2)) \equiv (T(w_1, p_1) \equiv T(w_1, p_2)))$

L6. $\forall w_1 \forall p_1 (T(w_1, NOT(p_1)) \equiv \neg T(w_1, p_1))$

Axioms L1-L6 merely translate the logical connectives from the object language to the metalanguage, using an ordinary Tarskian truth definition. For instance, according to L2, AND(P,Q) is true in a world if and only if P and Q are both true in the world. The other axioms state that all the truth-functional connectives are "transparent" to T in exactly the same way.

To represent quantified object language formulas in the metalanguage, we will introduce additional functions into the abstract syntax: EXIST and ALL. These functions will take two arguments--a term denoting an object language variable and a term denoting an object language formula. Axiomatizing the interpretation of quantified object language formulas presents some minor technical problems, however. We would like to say something like this: EXIST(X,P) is true in W if and only if there is some individual such that the open formula P is true of that individual in W. We do not have any way of saying that an open formula is true of an individual in a world, however; we just have the predicate T, which simply says that a formula is true in a world. One way of solving the problem would be to introduce a new predicate, or perhaps redefine T, to express the Tarskian notion of satisfaction rather than truth. This approach is semantically clean but syntactically clumsy, so we will instead follow the advice of Scott (1970, p. 151) and define the truth of a quantified statement in terms of substituting into the body of that statement a rigid designator for the value of the quantified variable.

In order to formalize this substitutional approach to the interpretation of object language quantification, we need a rigid designator in the object language for every individual. Since our representation of the object language is in the form of an abstract syntax, we can simply stipulate that there is a function @ that maps any individual in the object language's domain of discourse into an object

language rigid designator of that individual. The definition of T for quantified statements is then given by the following axiom schemata:

L7. $\forall w_1\ (T(w_1, \mathrm{EXIST}(X,P)) \equiv \exists x_1\ (T(w_1, P[@(x_1)/X])))$

L8. $\forall w_1\ (T(w_1, \mathrm{ALL}(X,P)) \equiv \forall x_1\ (T(w_1, P[@(x_1)/X])))$

In these schemata, P may be any object language formula, X may be any object language variable, and the notation $P[@(x_1)/X]$ designates the expression that results from substituting $@(x_1)$ for every free occurrence of X in P.

L7 says that an existentially quantified formula is true in a world W if and only if, for some individual, the result of substituting a rigid designator of that individual for the bound variable in the body of the formula is true in W. L8 says that a universally quantified formula is true in W if and only if, for every individual, the result of substituting a rigid designator of that individual for the bound variable in the body of the formula is true in W.

Except for the knowledge operator itself, the only part of the truth definition of the object language that remains to be given is the definition of T for atomic formulas. We remarked previously that a formula of the form $P(A_1,\ldots,A_n)$ is true in a world W just in case the tuple consisting of the denotations in W of the terms $A_1,\ldots,A_n$ is in

the extension in W of the relation P. To axiomatize this principle, we need two additions to the metalanguage. First, we need a function D that maps a possible world and an object language term into the denotation of that term in that world. Second, for each n-place object language predicate P, we need a corresponding n+1-place metalanguage predicate (which, by convention, we will write :P) that takes as its arguments the possible world in which the object language formula is to be evaluated and the denotations in that world of the arguments of the object language predicate. The interpretation of an object language atomic formula is then given by the axiom schema

$$\text{L9. } \forall w_1, t_1, \ldots, t_n \; (T(w_1, P(t_1, \ldots, t_n)) \equiv :P(w_1, D(w_1, t_1), \ldots, D(w_1, t_n)))$$

To eliminate the function D, we need to introduce a metalanguage expression corresponding to each object language constant or function. In the general case, the new expression will be a function with an extra argument position for the possible world of evaluation. The axiom schemata for D are then

$$\text{L10. } \forall w_1, x \; (D(w_1, @(x)) = x)$$

$$\text{L11. } \forall w_1 \; (D(w_1, C) = :C(w_1))$$

$$\text{L12. } \forall w_1, t_1, \ldots, t_n \; (D(w_1, F(t_1, \ldots, t_n)) = :F(w_1, D(w_1, t_1), \ldots, D(w_1, t_n))),$$

where C is an object language constant and F is an object langauage function, and we use the ":" convention already introduced for their metalanguage counterparts.

Since @(x_1) is a rigid designator of x_1, its value is x_1 in every possible world. In the general case, an object language constant will have a corresponding metalanguage function that picks out the denotation of the constant in a particular world. Similarly, an object language function will have a corresponding metalanguage function that maps a possible world and the denotations of the arguments of the object language function into the value of the object language function applied to those arguments in that world.

It will be convenient to treat specially those object language constants and functions that are (or can be used to construct) rigid designators. We could introduce additional axioms asserting that such expressions have the same value in every possible world, but we can accomplish the same end simply by making the corresponding metalanguage expressions independent of the possible world of evaluation. So, for object language constants that are rigid designators, we will have a variant of axiom L11:

L11a. $\forall w_1 (D(w_1, C) = :C)$ if C is a rigid designator.

We will similarly treat rigid functions--those that always map a particular tuple of arguments into the same value in all possible worlds:

L12a. $\forall w_1 t_1 \ldots t_n (D(w_1, F(t_1, \ldots, t_n)) = :F(D(w_1, t_1), \ldots, D(w_1, t_n)))$
if F is a rigid function.

Finally, we introduce a special axiom for the equality predicate of the object language, fixing its interpretation in all possible worlds to be the identity relation:

L13. $\forall w_1 t_1 t_2 (T(w_1, EQ(t_1, t_2)) \equiv (D(w_1, t_1) = D(w_1, t_2)))$

B. A First-Order Theory of Knowledge

The axioms given in the preceding section allow us to talk about a formula of first-order logic being true relative to a possible world rather than absolutely. This generalization would be pointless, however, if we never had occasion to mention any possible worlds other than the actual one. References to other possible worlds are introduced by our axioms for knowledge:

K1. $\forall w_1 t_1 p_1$
$(T(w_1, KNOW(t_1, p_1)) \equiv \forall w_2 (K(D(w_1, t_1), w_1, w_2) \supset T(w_2, p_1)))$

K2. $\forall a_1 w_1 (K(a_1, w_1, w_1))$

K3. $\forall a_1 w_1 w_2 (K(a_1, w_1, w_2) \supset \forall w_3 (K(a_1, w_2, w_3) \supset K(a_1, w_1, w_3)))$

K1 gives the possible-world analysis for object language formulas of the form KNOW(A,P). The interpretation is that KNOW(A,P) is true in

world W_1 just in case P is true in every world that is compatible with what the agent denoted by A in W_1 knows in W_1. Since an object language term may denote different individuals in different possible worlds, we use $D(W_1,A)$ to identify the denotation of A in W_1. K represents the accessibility relation associated with KNOW, so $K(D(W_1,A),W_1,W_2)$ is how we represent the fact W_2 is compatible with what the agent denoted by A in W_1 knows in W_1.

As we pointed out before, the principle embodied in K1 is that an agent knows everything entailed by his knowledge. Since this is too strong a generalization, in a more thorough analysis we would regard the inference from the right side of K1 to the left side as being a default inference. K2 and K3 state constraints on the accessibility relation K that we use to capture other properties of knowledge. They require that, for a fixed agent :A, $K(:A,w_1,w_2)$ be reflexive and transitive. We have already shown this entails that anything that anyone knows must be true, and that if someone knows something he knows that he knows it. Finally, the fact that K1-K3 are asserted to hold for all possible worlds implies that everyone knows the principles they embody, and everyone knows that everyone knows, etc. In other words, these principles are common knowledge.

To illustrate how our theory operates, we will show how to derive a simple result in the logic of knowledge, that from the premises that A knows that P(B) and A knows that B = C, we can conclude that A knows that P(C). Our proofs will be in natural-deduction form. The axioms and preceding lines that justify each step will be given to the right of the step. Subordinate proofs will be indicated by indented sections, and ASS will mark the assumptions on which these subordinate proofs are based. DIS(N,M) will indicate the discharge of the assumption on line N with respect to the conclusion on line M. The general pattern of proofs in this system will be to assert the object language premises of the problem, transform them into their metalanguage equivalents, using axioms L1-L13 and K1, then derive the metalanguage version of the conclusion using first-order logic and axioms such as K2 and K3, and finally transform the conclusion back into the object language, again using L1-L13 and K1.

Given: TRUE(KNOW(A,P(B)))
TRUE(KNOW(A,EQ(B,C)))

Prove: TRUE(KNOW(A,P(C)))

1. TRUE(KNOW(A,P(B)))	Given
2. $T(W_0, KNOW(A,P(B)))$	L1,1
3. $K(D(W_0,A),W_0,w_1) \supset T(w_1,P(B))$	K1,2
4. $K(:A(W_0),W_0,w_1) \supset T(w_1,P(B))$	L11,3
5. TRUE(KNOW(A,EQ(B,C)))	Given
6. $T(W_0, KNOW(A,EQ(B,C)))$	L1,5
7. $K(D(W_0,A),W_0,w_1) \supset T(w_1,EQ(B,C))$	K1,6

```
 8. K(:A(W ),W ,w ) ⊃ T(w ,EQ(B,C))                    L11,7
         0   0  1       1
 9.    K(:A(W ),W ,w )                                  ASS
          0   0  1
10.    T(w ,P(B))                                       4,9
         1
11.    :P(w ,D(w ,B))                                   L9,10
          1    1
12.    :P(w ,:B(w ))                                    L11,11
          1     1
13.    T(w ,EQ(B,C))                                    8,9
         1
14.    D(w ,B) = D(w ,C)                                L13,13
          1          1
15.    :B(w ) = :C(w )                                  L11,14
          1          1
16.    :P(w ,:C(w ))                                    12,15
          1     1
17.    :P(w ,D(w ,C))                                   L11,16
          1    1
18.    T(w ,P(C))                                       L9,17
         1
19. K(:A(W ),W ,w ) ⊃ T(w ,P(C))                       DIS(9,18)
         0   0  1       1
20. K(D(W ,A),W ,w ) ⊃ T(w ,P(C))                      L11,19
        0     0  1       1
21. T(W ,KNOW(A,P(C)))                                 K1,20
       0
22. TRUE(KNOW(A,P(C)))                                 L1,21
```

A knows that P(B) (Line 1), so P(B) is true in every world compatible with what A knows (Line 4). Similarly, since A knows that B = C (Line 5), B = C is true in every world compatible with what A knows (Line 8). Let w_1 be one of these worlds (Line 9). P(B) and B = C must be true in w_1 (Lines 12 and 15), hence P(C) must be true in w_1 (Line 16). Therefore, P(C) is true in every world compatible with what A knows (Line 19), so A knows that P(C) (Line 22). If TRUE(EQ(B,C)) had been given instead of TRUE(KNOW(A,EQ(B,C))), we would have had B = C

true in W_0 instead of w_1. In that case, the substitution of C for B in P(B) (Line 16) would not have been valid, and we could not have concluded that A knows that P(C). This proof seems long because we have made each routine step a separate line. This is worth doing once to illustrate all the formal details, but in subsequent examples we will combine some of the routine steps to shorten the derivation.

IV A POSSIBLE-WORLD ANALYSIS OF ACTION

In the preceding sections, we have presented a framework for describing what someone knows in terms of possible worlds. To characterize the relation of knowledge to action, we need a theory of action in these same terms. Fortunately, the standard way of looking at actions in AI gives us just that sort of theory. Most AI programs that reason about actions are based on a view of the world as a set of possible states of affairs, with each action determining a binary relation between states of affairs--one being the outcome of performing the action in the other. We can integrate our analysis of knowledge with this view of action by identifying the possible worlds used to describe knowledge with the possible states of affairs used to describe actions.

The identification of a possible world, as used in the analysis of knowledge, with the state of affairs at a particular time does not require any changes in the formalization already presented, but it does require a reinterpretation of what the axioms mean. If the variables $w_1, w_2, \ldots$ are reinterpreted as ranging over states of affairs, then "A knows that P" will be analyzed roughly as "P is true in every state of affairs that is compatible with what A knows in the actual state of

affairs." It might seem that taking possible worlds to be states of affairs, and therefore not extended in time, might make it difficult to talk about what someone knows regarding the past or future. That is not the case, however. Knowledge about the past and future can be handled by modal tense operators, with corresponding accessibility relations between possible states-of-affairs/worlds. We could have a tense operator FUTURE such that FUTURE(P) means that P will be true at some time to come. If we let F be an accessibility relation such that $F(W_1, W_2)$ means that the state-of-affairs/world W_2 lies in the future of the state-of-affairs/world W_1, then we can define FUTURE(P) to be true in W_1 just in case there is some W_2 such that $F(W_1, W_2)$ holds and P is true in W_2.

This much is standard tense logic (e.g., Rescher and Urquhart, 1971). The interesting point is that statements about someone's knowledge of the future work out correctly, even though such knowledge is analyzed in terms of alternatives to a state of affairs, rather than alternatives to a possible world containing an entire course of events. The proposition that John knows that P will be true is represented simply by KNOW(JOHN,FUTURE(P)). The analysis of this is that FUTURE(P) is true in every state of affairs that is compatible with what John knows, from which it follows that, for each state of affairs that is compatible with what John knows, P is true in some future alternative to

that state of affairs. An important point to note here is that two states of affairs can be "internally" similar (that is, they coincide in the truth-value assigned to any nonmodal statement), yet be distinct because they differ in the accessibility relations they bear to other possible states of affairs. Thus, although we treat a possible world as a state of affairs rather than a course of events, it is a state of affairs in the particular course of events defined by its relationships to other states of affairs.

For planning and reasoning about future actions, instead of a tense operator like FUTURE, which simply asserts what will be true, we need an operator that describes what would be true if a certain event occurred. Our approach will be to recast McCarthy's situation calculus (McCarthy, 1968) (McCarthy and Hayes, 1969) so that it meshes with our possible-world characterization of knowledge. The situation calculus is a first-order language in which predicates that can vary in truth-value over time are given an extra argument to indicate what situations (i.e., states of affairs) they hold in, with a function RESULT that maps an agent, an action, and a situation into the situation that results from the agent's performance of the action in the first situation. Statements about the effects of actions are then expressed by formulas like P(RESULT(A,ACT,S)), which means that P is true in the situation that results from A's performing ACT in situation S.

To integrate these ideas into our logic of knowledge, we will reconstruct the situation calculus as a modal logic. In parallel to the

operator KNOW for talking about knowledge, we introduce an object language operator RES for talking about the results of events. Situations will not be referred to explicitly in the object language, but they will reappear in the possible-world semantics for RES in the metalanguage. RES will be a two-place operator whose first arguments is a term denoting an event, and whose second argument is a formula. RES(E,P) will mean that it is possible for the event denoted by E to occur and that, if it did, the formula P would then be true. The possible-world semantics for RES will be specified in terms of an accessiblity relation R, parallel to K, such that $R(:E,W_1,W_2)$ means that W_2 is the situation/world that would result from the event :E happening in W_1.

We assume that, if it is impossible for :E to happen in W_1 (i.e., if the prerequisites of :E are not satisfied), then there is no W_2 such that $R(:E,W_1,W_2)$ holds. Otherwise we assume that there is exactly one W_2 such that $(:E,W_1,W_2)$ holds:

R1. $\forall w_1,w_2,w_3,e\ ((R(e_1,w_1,w_2) \wedge R(e_1,w_1,w_3)) \supset (w_2 = w_3))$

(Variables e_1, e_2,... range over events.) Given these assumptions,

RES(E,P) will be true in a situation/world W_1 just in case there is some W_2 that is the situation/world that results from the event described by E happening in W_1, and in which P is true:

R2. $\forall w_1, t_1, p_1 (T(w_1, RES(t_1, p_1)) \equiv \exists w_2 (R(D(w_1, t_1), w_1, w_2) \wedge T(w_2, p_1)))$

The type of event we will normally be concerned with is the performance of an action by an agent. We will let DO(A,ACT) be a description of the event consisting of the agent denoted by A performing the action denoted by ACT.[5] (We will assume that the set of possible agents is the same as the set of possible knowers.) We will want DO(A,ACT) to be the standard way of referring to the event of A's carrying out the action ACT, so DO will be a rigid function. Hence, DO(A,ACT) will be a rigid designator of an event if A is a rigid designator of an agent and ACT a rigid designator of an action.

Many actions can be thought of as general procedures applied to particular objects. Such a general procedure will be represented by a function that maps the objects to which the procedure is applied into the action of applying the procedure to those objects. For instance, if DIAL represents the general procedure of dialing combinations of safes, SF a safe, and COMB(SF) the combination of SF, then DIAL(COMB(SF),SF) represents the action of dialing the combination COMB(SF) on the safe SF, and DO(A,DIAL(COMB(SF),SF)) represents the event of A's dialing the combination COMB(SF) on the safe SF.

This formalism gives us the ability describe an agent's knowledge of the effects of carrying out an action. In the object language, we can express the claim that A_1 knows that P would result from A_2's doing ACT by saying that $KNOW(A_1, RES(DO(A_2, ACT), P))$ is true. The possible-world analysis of this statement is that, for every world compatible with what A_1 knows in the actual world, there is a world that is the result of A_2's doing ACT and in which P is true (see Figure 5). Formally, this is expressed by

$$\forall w_1 (K(:A_1, W_1, w_0, w_1) \supset \exists w_2 (R(:DO(:A_2, :ACT), w_1, w_2) \wedge T(w_2, P))),$$

if we assume that A_1, A_2, and ACT are rigid designators.

In addition to simple, one-step actions, we will want to talk about complex combinations of actions. We will therefore introduce expressions into the object language for action sequences, conditionals, and iteration. If P is a formula, and ACT_1 and ACT_2 are action descriptions, then $(ACT_1 ; ACT_2)$, $IF(P, ACT_1, ACT_2)$, and $WHILE(P, ACT_1)$ will also be action descriptions. Roughly speaking, $(ACT_1 ; ACT_2)$ describes the sequence of actions consisting of ACT_1 followed by ACT_2. $IF(P, ACT_1, ACT_2)$ describes the conditional action of doing ACT_1 if P is

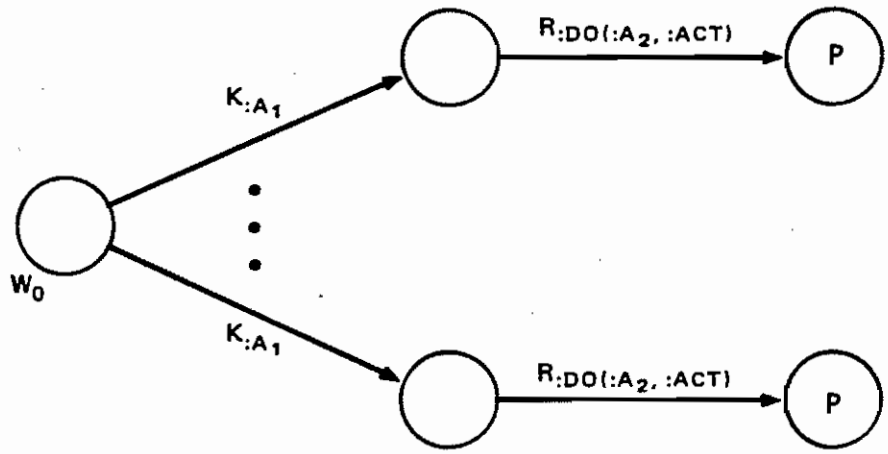

FIGURE 5 $TRUE(KNOW(A_1, RES(DO(A_2, ACT), P))) \equiv$
$\forall w_1(K(:A_1, W_0, w_1) \supset \exists w_2(R(:DO(:A_2, :ACT), w_1, w_2) \wedge T(w_2, P)))$

true, otherwise doing ACT_2. $WHILE(P, ACT_1)$ describes the iterative action of repeating ACT_1 as long as P is true.

Defining denotations for these complex action descriptions is somewhat problematical. The difficulty comes from the fact that, whenever we have an action described as a sequence of subactions, any expression used in specifying one of the subactions needs to be interpreted relative to the situation in which that subaction is carried out. For instance, if PUTON(X,Y) denotes the action of putting X on Y, STACK denotes a stack of blocks, TABLE denotes a table, and TOP picks out the top block of a stack, we would want the execution of

(PUTON(TOP(STACK),TABLE); PUTON(TOP(STACK),TABLE))

to result in what were initially the top two blocks of the stack being put on the table, rather than what was initially the top block being put on the table twice. The second occurrence of TOP(STACK) should be interpreted with respect to the situation in which the first block has already been removed. The problem is that, in general, what situation exists after one step of a sequence of actions has been excecuted depends on who the agent is. If John picks up a certain block, he will be holding the block; if, however, Mary performs the same action, she will be holding the block. If an action description refers to "the block Mary is holding," exactly which block it is may depend on which agent is carrying out the action, but this is not specified by the action description.

One way of getting around these difficulties conceptually would be to treat actions as functions from agents to events, but notational problems would remain nevertheless. We will therefore choose a different solution: treating complex actions as "virtual individuals" (Scott, 1970), or pseudoentities. That is, complex action descriptions will not be treated as referring expressions in themselves, but only as component parts of more complex referring expressions. In particular, if ACT is a complex action description (and A denotes an agent), we will treat the event description DO(A,ACT), but not ACT itself, as having a denotation. Complex action descriptions will be permitted to occur only as part of such event descriptions, and we will define the denotations

of the event descriptions in a way that eliminates reference to complex actions. We will, however, continue to treat actions as real entities that can be quantified over, and simple action descriptions such as DIAL(COMB(SF),SF) will still be considered to denote actions.

The denotations of event descriptions formed from conditional and iterative action descriptions can be defined as follows in terms of the denotations of event descriptions formed from action sequence descriptions:

R3. $\forall w_1, t_1, t_2, t_3, p_1$
$((T(w_1, p_1) \supset (D(w_1, DO(t_1, IF(p_1, t_2, t_3))) = D(w_1, DO(t_1, t_2)))) \wedge$
$(\neg T(w_1, p_1) \supset (D(w_1, DO(t_1, IF(p_1, t_2, t_3))) = D(w_1, DO(t_1, t_3))))$

R4. $\forall w_1, t_1, t_2, p_1$
$(D(w_1, DO(t_1, WHILE(p_1, t_2))) =$
$D(w_1, DO(t_1, IF(p_1, (t_2 ; WHILE(p_1, t_2)), NIL)))$

R3 says that performing the conditional action IF(P,ACT$_1$,ACT$_2$) results in the same event as carrying out ACT$_1$ in a situation where P is true or carrying out ACT$_2$ in a situation where P is false. R4 says that performing WHILE(P,ACT) always results in the same event as IF(P,(ACT; WHILE(P,ACT)),NIL), where NIL denotes the null action. In other words, doing WHILE(P,ACT) is equivalent to doing ACT followed by WHILE(P,ACT) if P is true, otherwise doing nothing--i.e., doing ACT as long as P remains true.

To define the denotation of events that consist of carrying out action sequences, we need some notation for talking about sequences of events. First, we will let ";" be a polymorphic operator in the object language, creating descriptions of event sequences in addition to action sequences. Speaking informally, if E_1 and E_2 are event descriptions, then $(E_1 ; E_2)$ names the event sequence consisting of E_1 followed by E_2, just as $(ACT_1 ; ACT_2)$ names the action sequence consisting of ACT_1 followed by ACT_2. In the metalanguage, event sequences will be indicated with angle brackets, so that $\langle :E_1 , :E_2 \rangle$ will mean $:E_1$ followed by $:E_2$. The denotations of expressions involving action and event sequences are then defined by the following axioms:

R5. $\forall w_1 , t_1 , t_2 , t_3$
$(D(w_1 , DO(t_1 , (t_2 ; t_3))) = D(w_1 , (DO(t_1 , t_2); DO(@(D(w_1 , t_1)), t_3))))$

R6. $\forall w_1 , w_2 , t_1 , t_2$
$(R(D(w_1 , t_1), w_2 , w_1) \supset (D(w_1 , (t_1 ; t_2)) = \langle D(w_1 , t_1), D(w_2 , t_2)\rangle))$

R5 says that the event consisting of an agent A's performance of the action sequence ACT_1 followed by ACT_2 is simply the event sequence that consists of A's carrying out ACT_1 followed by his carrying out

ACT$_2$. Note that, in the description of the second event, the agent is picked out by the expression @(D(w$_1$,A)), which guarantees that we get the same agent as in the first event, in case the original term picking out the agent changes its denotation after the first event has happened. R6 then defines the denotation of an event sequence description (E$_1$; E$_2$) as the sequence comprising the denotation of E$_1$ in the original situation followed by the denotation of E$_2$ in the situation resulting from the occurrence of E$_1$. If there is no situation that results from the occurence of E$_1$, we leave the denotation of (E$_1$; E$_2$) undefined.

Finally, we need to define the accessibility relation R for event sequences and for events in which the null action is carried out.

R7. $\forall w_1, w_2, e_1, e_2$
$(R(\langle e_1, e_2 \rangle, w_1, w_2) \equiv \exists w_3 (R(e_1, w_1, w_3) \wedge R(e_2, w_3, w_2)))$

R8. $\forall w_1, a_1 (R(:DO(a_1, :NIL), w_1, w_1))$

R7 says that a situation W$_2$ is the result of the event sequence $\langle E_1, E_2 \rangle$ occurring in W$_1$ if and only if there is a situation W$_3$ such that W$_3$ is the result of E$_1$ occurring in W$_1$, and W$_2$ is the result of E$_2$ occurring

in W_3^6. We will regard NIL as a rigid designator in the object language for the null action, so :NIL will be its metalanguage counterpart. R8, therefore, says that in any situation the result of doing nothing is the same situation.

V AN INTEGRATED THEORY OF KNOWLEDGE AND ACTION

A. The Dependence of Action on Knowledge

As we pointed out in the introduction, knowledge and action interact in two principal ways: (1) knowledge is often required prior to taking action; (2) actions can change what is known. In regard to the first, we need to consider knowledge prerequisites as well as physical prerequisites for actions. Our main thesis is that the knowledge prerequisites for an action can be analyzed as a matter of knowing what action to take. Recall the example of trying to open a locked safe. Why is it that, for an agent to achieve this goal by using the plan "Dial the combination of the safe," he must know the combination? The reason is that an agent could know that dialing the combination of the safe would result in the safe's being open, but still not know what to do because he does not know what the combination of the safe is. A similar analysis applies to knowing a telephone number in order to call someone on the telephone or knowing a password in order to gain access to a computer system.

It is important to realize that even mundane actions that are not usually thought of as requiring any special knowledge are no different from the examples just cited. For instance, none of the AI problem-

solving systems that have dealt with the blocks world have tried to take into account whether the robot possesses sufficient knowledge to be able to move block A to point B. Yet, if a command were phrased as "Move my favorite block back to its original position," the system could be just as much in the dark as with "Dial the combination of the safe." If the system does not know what actions satisfy the description, it will not be able to carry out the command. The only reason that the question of knowledge seems more pertinent in the case of dialing combinations and telephone numbers is that, in the contexts in which these actions naturally arise, there is usually no presumption that the agent knows what action fits the description. An important consequence of this view is that the specification of an action will normally not need to include anything about knowledge prerequisites. These will be supplied by a general theory of using actions to achieve goals. What we will need to specify are the conditions under which an agent knows what action is referred to by an action description.

In our possible-world semantics for knowledge, the usual way of knowing what entity is referred to by a description B is by having some description C that is a rigid designator, and by knowing that B = C. (Note, that if B itself is a rigid designator, it can be used for C.) In particular, knowing what action is referred to by an action description means having a rigid designator for the action described. But, if this is all the knowledge that is required for carrying out the action, then a rigid designator for an action must be an executable

description of the action--in the same sense that a computer program is an executable description of a computation to an interpreter for the language in which the program is written.

Often the actions we want to talk about are mundane general procedures that we would be willing to assume everyone knows how to perform. Dialing a telephone number or the combination of a safe is a typical example. In many of these cases, if an agent knows the general procedure and what objects the procedure is to be applied to, then he knows everything that is relevant to the task. In such cases, the function that represents the general procedure will be a rigid function, so that, if the arguments of the function are rigid designators, the term consisting of the function applied to the arguments will be a rigid designator. Hence, knowing what objects the arguments denote will amount to knowing what action the term refers to. We will treat dialing the combination of a safe, or dialing a telephone number as being this type of procedure. That is, we assume that anyone who knows what combination he is to dial and what safe he is to dial it on thereby knows what action he is to perform.

There are other procedures we might also wish to assume that anyone could perform, but that cannot be represented as rigid functions. Suppose that, in the blocks world, we let PUTON(B,C) denote the action of putting B on C. Even though we would not want to question anyone's ability to perform PUTON in general, knowing what objects B and C are will not be sufficient to perform PUTON(B,C); knowing where they are is

also necessary. We could have a special axiom stating that knowing what action PUTON(B,C) is requires knowing where B and C are, but this will be superfluous if we simply assume that everyone knows the definition of PUTON in terms of more primitive actions. If we define PUTON(X,Y) as something like

```
(MOVEHAND(LOCATION(X));
 GRASP;
 MOVEHAND(LOCATION(TOP(Y)));
 UNGRASP),
```

then we can treat MOVEHAND, GRASP, and UNGRASP as rigid functions, and we can see that executing PUTON requires knowing where the two objects are because their locations are mentioned in the definition. So, although PUTON itself is not a rigid function, we can avoid having a special axiom stating what the knowledge prerequisites of PUTON are by defining PUTON as a sequence of actions represented by rigid functions.

To formalize this theory, we will introduce a new object language operator CAN. CAN(A,ACT,P) will mean that A can achieve P by performing ACT, in the sense that A knows how to achieve P by performing ACT. We will not give a possible-world semantics for CAN directly; instead we will give a definition of CAN in terms of KNOW and RES, which we can use in reasoning about CAN to transform a problem into terms of possible worlds.

In the simplest case, an agent A can achieve P by performing ACT if he knows what action ACT is, and he knows that P would be true as a

result of his performing ACT. In the object language, we can express this fact by

$$\forall a(\exists x(KNOW(a,((x = ACT) \wedge RES(DO(a,ACT),P)))) \supset CAN(a,ACT,P)).$$

We cannot strengthen this assertion to a biconditional, however, because that would be too stringent a definition of CAN for complex actions. It would require the agent to know from the very beginning of his action exactly what he is going to do at every step. In carrying out a complex action, though, an agent may take some initial action that results in his acquiring knowledge about what to do later.

For an agent to be able to achieve a goal by performing a complex action, all that is really neccessary is that he know what to do first, and that he know that he will know what to do at each subsequent step. So, for any action descriptions ACT and ACT_1, the following formula also states a condition under which an agent can achieve P by performing ACT:

$$\forall a(\exists x(KNOW(a,((DO(a,(x; ACT_1)) = DO(a,ACT)) \wedge RES(DO(a,x),CAN(a,ACT_1,P))))) \supset CAN(a,ACT,P)).$$

This says that A can achieve P by doing ACT if there is an action X such that A knows that his execution of the sequence X followed by ACT_1 would be equivalent to his doing ACT, and that his doing X would result in his being able to achieve P by doing ACT_1.

Finally, with the following metalanguage axiom we can state that these are the only two conditions under which an agent can use a particular action to achieve a goal:

$$
\begin{aligned}
&\text{C1. } \forall w_1, t_1, t_2, t_3, p_1 \\
&\quad ((t_2 = @(D(w_1, t_1))) \supset \\
&\quad\ (T(w_1, CAN(t_1, t_3, p_1)) \equiv \\
&\quad\quad (T(w_1, EXIST(X, KNOW(t_1, AND(EQ(X, t_3), RES(DO(t_1, t_3), p_1))))) \vee \\
&\quad\quad\ \exists t_4\, (T(w_1, EXIST(X, KNOW(t_1, AND(EQ(DO(t_1, (X; t_4))), DO(t_1, t_4)), \\
&\quad\quad\quad RES(DO(t_1, X), \\
&\quad\quad\quad\quad CAN(t_1, t_4, p_1)))))))))
\end{aligned}
$$

Letting $t_1 = A_1$, $t_2 = A_1$, and $t_3 = ACT_1$, C1 says that, for any formula P, if A_1 is the standard identifier of the agent denoted by A, then A can achieve P by doing ACT if and only if one of the following conditions is met: (1) A knows what action ACT is and knows that P would be true as a result of A_1's (i.e., his) doing ACT, or (2) there is an action description $t_4 = ACT_1$ such that, for some action X, A knows that A_1's doing X followed by ACT is the same event as his doing ACT and knows that A_1's doing X would result his being able to achieve P by doing ACT_1.

As a simple illustration of these concepts, we will show how to derive the fact that an agent can open a safe, given the premise that he knows the combination. To do this, the only additional fact we need is that, if an agent does dial the correct combination of a safe, the safe will then be open:

D1. $\forall w_1, a_1, x_1$
$(:SAFE(x_1) \supset$
$\exists w_2 (R(:DO(a_2, :DIAL(:COMB(w_1, x_1), x_1)), w_1, w_2) \wedge$
$:OPEN(w_2, x_1)))$

D1 says that, for any possible world W_1, any agent :A, and any safe :SF, there is a world W_2 that is the result of :A's dialing the combination of :SF on :SF in W_1, and in which :SF is open. The important point about this axiom, is that the function :COMB (which picks out the combination to a safe) depends on what possible world it is evaluated in, while :DIAL (the function that maps a combination and a safe into the action of dialing the combination on the safe) does not. Thus we are implicitly assuming that, given a particular safe, there may be some doubt as to what its combination is, but, given a combination and a safe, there exists no possible doubt as to what action dialing the combination on the safe is. (We also simplify matters by omitting the possible world-argument to :SAFE, so as to avoid raising the question of knowing whether something is a safe.) Since this axiom is asserted to

hold for all possible worlds, we are in effect assuming that it is common knowledge.

Now we show that, for any safe, if the agent A knows its combination, he can open the safe by dialing that combination; or, more precisely, for all X, if X is a safe and there is some Y, such that A knows that Y is the combination of X, then A can open X by dialing the combination of X on X:

```
Prove: TRUE(ALL(X,IMP(AND(SAFE(X),EXIST(Y,KNOW(A,EQ(Y,COMB(X))))))
                      CAN(A,DIAL(COMB(X),X),OPEN(X))))
```

```
1.   T(W ,                                                    ASS
       0
      AND(SAFE(@(x ),
                1
          EXIST(Y,KNOW(A,EQ(Y,COMB(@(x ))))))))
                                                  1

2.   :SAFE(x )                                                1,L2,L9
        1

3.   ∀w (K(:A(W ),W ,w ) ⊃                                    1,L2,L7,K1,L11,
      1       0   0  1
        (:C = :COMB(w ,x )))                                  L13,L10,L12
                   1  1

4.     K(:A(W ),W ,w )                                        ASS
          0   0  1

5.     :C = :COMB(w ,x )                                      3,4
                1  1

6.     :DIAL(:C,x ) = :DIAL(:COMB(w ,x ),x )                  5
              1                    1  1   1

7.     T(w ,                                                  L10,L12,L12a,L13
         1
        EQ(@(:DIAL(:C,x )),
                    1
           DIAL(COMB(@(x )),@(x ))))
                      1      1
```

8. $\exists w_2$ (R(:DO(:A(W_0), :DIAL(:COMB(w_1, x_1), x_1)), w_1, w_2) ∧ :OPEN(w_2, x_1))) 2,D1

9. T(w_1, RES(DO(@(D(W_0, A)), DIAL(COMB(@(x_1)), @(x_1))), OPEN(@(x_1))) L11,L10,L12a,L9,R2

10. T(w_1, AND(EQ(@(:DIAL(:C, x_1)), DIAL(COMB(@(x_1)), @(x_1))), RES(DO(@(D(W_0, A)), DIAL(COMB(@(x_1)), @(x_1))), OPEN(@(x_1))))) 7,9,L2

11. K(:A(W_0), W_0, w_1) ⊃ T(w_1, AND(EQ(@(:DIAL(:C, x_1)), DIAL(COMB(@(x_1)), @(x_1))), RES(DO(@(D(W_0, A)), DIAL(COMB(@(x_1)), @(x_1))), OPEN(@(x_1))))) DIS(4,10)

```
12.  T(W ,                                                    11,L11,K1
       0
     KNOW(A,
          AND(EQ(@(:DIAL(:C,x )),
                              1
                 DIAL(COMB(@(x )),@(x ))),
                                1      1
              RES(DO(@(D(W ,A)),
                          0
                     DIAL(COMB(@(x )),@(x ))),
                                   1      1
                  OPEN(@(x ))))))
                         1

13.  T(W ,                                                    12,L7
       0
     EXIST(X,
           KNOW(A,
                AND(EQ(X,
                       DIAL(COMB(@(x )),
                                 1
                            @(x ))),
                              1
                    RES(DO(@(D(W ,A)),
                                0
                           DIAL(COMB(@(x )),
                                     1
                                @(x ))),
                                  1
                        OPEN(@(x )))))
                               1

14.  T(W ,                                                    13,C1
       0
     CAN(A,
         DIAL(COMB(@(x )),@(x )),
                          1      1
         OPEN(@(x ))))
                 1

15. T(W ,                                                     DIS(1,14)
       0
     AND(SAFE(@(x ),
                 1
         EXIST(Y,KNOW(A,EQ(Y,COMB(@(x )))))))) ⊃
                                          1
    T(W ,
       0
     CAN(A,DIAL(COMB(@(x )),@(x )),OPEN(@(x ))))
                          1      1           1
```

```
16. TRUE(ALL(X,                                          15,L4,L8,L1
             IMP(AND(SAFE(X),
                     EXIST(Y,
                           KNOW(A,
                                EQ(Y,COMB(X))))))
                 CAN(A,DIAL(COMB(X),X),OPEN(X))))
```

Suppose that x_1 is a safe and there is some C that A knows to be the combination of x_1 (Lines 1-3). Suppose w_1 is a world that is compatible with what A knows in the actual world, W_0 (Line 4). Then C is the combination of x_1 in w_1 (Line 5), so dialing C on x_1 is the same action as dialing the combination of x_1 on x_1 in w_1 (Lines 6 and 7). By axiom D1, A's dialing the combination of x_1 on x_1 in w_1 will result in x_1's being open (Lines 8 and 9). Since w_1 was an arbitrarily chosen world compatible with what A knows in W_0, it follows that in W_0 A knows dialing C on x_1 to be the act of dialing the combination of x_1 on x_1 and that his dialing the combination of x_1 on x_1 will result in x_1's being open (Lines 10-12). Hence, A knows what action dialing the combination of x_1 on x_1 is, and that his dialing the combination of x_1 on x_1 will result in x_1's being open (Line 13). Therefore A can open x_1 by dialing the combination of x_1 on x_1, provided that x_1 is a safe and he knows the

combination of x_1 (Lines 14 and 15). Finally, since x_1 was chosen arbitrarily, we conclude that A can open any safe by dialing the combination, provided he knows the combination (Line 16).

B. The Effects of Action on Knowledge

In describing the effects of an action on what an agent knows, we will distinguish actions that give the agent new information from those that do not. Actions that provide an agent with new information will be called informative actions. An action is informative if an agent would know more about the situation resulting from his performing the action after performing it than before performing it. In the blocks world, looking inside a box could be an informative action, but moving a block would probably not, because an agent would normally know no more after moving the block than he would before moving it. In the real world there are probably no actions that are never informative, because all physical processes are subject to variation and error. Nevertheless, it seems clear that we do and should treat many actions as noninformative from the standpoint of planning.

Even if an action is not informative in the sense we have just defined, performing the action will still alter the agent's state of knowledge. If the agent is aware of his action, he will know that it has been performed. As a result, the tense and modality of many of the things he knows will change. For example, if before performing the

action he knows that P is true, then after performing the action he will know that P was true before he performed the action. Similarly, if before performing the action he knows that P would be true after performing the action, then afterwards he will know that P is true.

We can represent this very elegantly in terms of possible worlds. Suppose :A is an agent and :E$_1$ an event that consists in :A's performing some noninformative action. For any possible worlds W$_1$ and W$_2$ such that W$_2$ is the result of :E$_1$'s happening in W$_1$, the worlds that are compatible with what :A knows in W$_2$ are exactly the worlds that are the result of :E$_1$'s happening in some world that is compatible with what :A knows in W$_1$. In formal terms, this is

$$\forall w_1 w_2 (R(:E_1, w_1, w_2) \supset \forall w_3 (K(:A, w_2, w_3) \equiv \exists w_4 (K(:A, w_1, w_4) \wedge R(:E_1, w_4, w_3))))),$$

which tells us exactly how what :A knows after :E$_1$ happens is related to what :A knows before :E$_1$ happens.

We can try to get some insight into this analysis by studying Figure 6. Sequences of possible situations connected by events can be thought of as possible courses of events. If W$_1$ is an actual situation

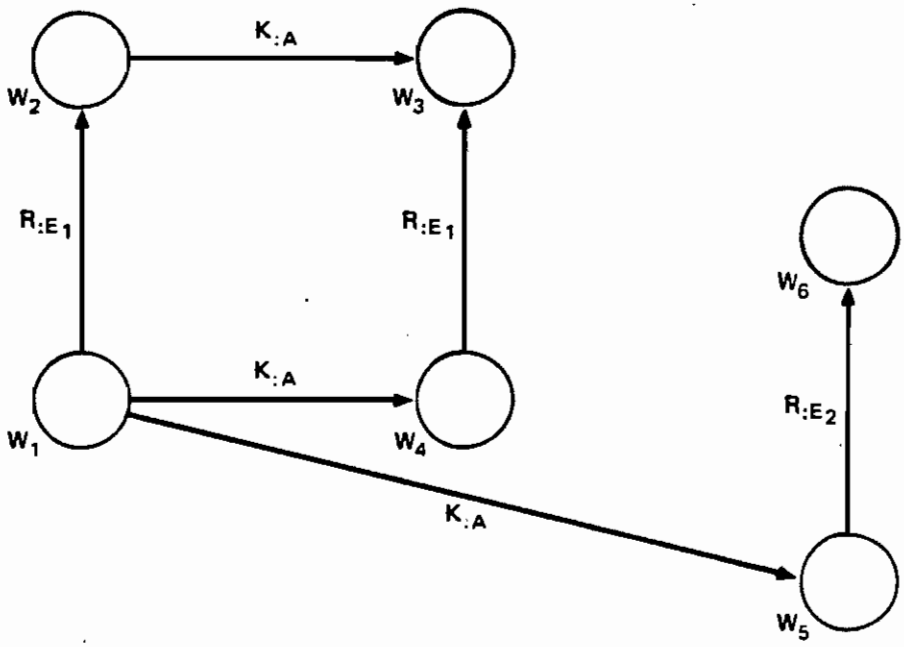

FIGURE 6 THE EFFECT OF A NONINFORMATIVE ACTION ON THE AGENT'S KNOWLEDGE

in which $:E_1$ occurs, thereby producing W_2, then W_1 and W_2 comprise a subsequence of the actual course of events. Now we can ask what other courses of events are compatible with what :A knows in W_1 and in W_2. Suppose that W_4 and W_3 are connected by $:E_1$ in a course of events that is compatible with what :A knows in W_1. Since $:E_1$ is not informative for :A, the only sense in which his knowledge is increased by $:E_1$ is that he knows that $:E_1$ has occurred. Since $:E_1$ occurs at the corresponding place in the course of events that includes W_4 and W_3,

this course of events will still be compatible with everything :A knows in W_2. However, the appropriate "tense shift" takes place. In W_1, W_4 is a possible alternative present for :A, and W_3 is a possible alternative future. In W_2, W_3 is a possible alternative present for :A, and W_4 is a possible alternative past.

Next consider a different course of events that includes W_5 and W_6 connected by a different event, $:E_2$. This course of events might be compatible with what :A knows in W_1 if he is not certain what he will do next, but, after $:E_1$ has happened and he knows that it has happened, this course of events is no longer compatible with what he knows. Thus, W_6 is not compatible with what :A knows in W_2. We can see, then, that even actions that provide the agent with no new information from the outside world still filter out for him those courses of events in which he would have performed actions other than those he actually did.

The idea of a filter on possible courses of events also provides a good picture of informative actions. With these actions, though, the filter is even stronger, since they not only filter out courses of events that differ from the actual course of events as to what event has just occurred, but they also filter out courses of events that are

incompatible with the information furnished by the action. Suppose :E is an event that consists in :A's performing an informative action, such that the information gained by the agent is whether the formula P is true. For any possible worlds W_1 and W_2 such that W_2 is the result of :E's happening in W_1, the worlds that are compatible with what :A knows in W_2 are exactly those worlds that are the result of :E's happening in some world that is compatible with what :A knows in W_1, and in which P has the same truth-value as in W_2:

$$\forall w_1, w_2 \, (R(:E, w_1, w_2) \supset \forall w_3 \, (K(:A, w_3, w_2) \equiv (\exists w_4 \, (K(:A, w_4, w_1) \wedge R(:E, w_4, w_3)) \wedge (T(w_2, P) \equiv T(w_3, P)))))$$

It is this final condition that distinguishes informative actions from those that are not.

Figure 7 illustrates this analysis. Suppose W_1 and W_2 are connected by :E and are part of the actual course of events. Suppose, further, that P is true in W_2. Let W_4 and W_3 also be connected by :E, and let them be part of a course of events that is compatible with what :A knows in W_1. If P is true in W_3 and the only thing :A learns about the world from :E (other than that it has occurred) is whether P is

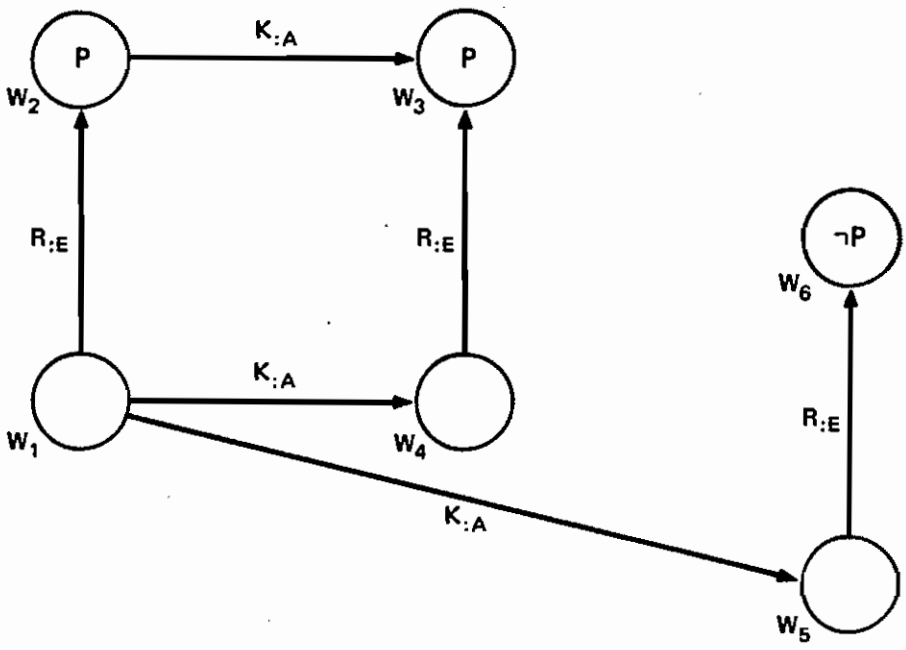

FIGURE 7 THE EFFECT OF AN INFORMATIVE ACTION ON THE AGENT'S KNOWLEDGE

true, this course of events will then still be compatible with what :A knows after :E has occurred. That is, W_3 will be compatible with what :A knows in W_2. Suppose, on the other hand, that W_5 and W_6 form part of a similar course of events, except that P is false in W_6. If :A does not know in W_1 whether P would be true after the occurrence of :E, this course of events will also be compatible with what he knows in W_1. After :E has occurred, however, he will know that P is true; consequently, this course of events will no longer be compatible with

what he knows. That is, W_6 will not be compatible with what :A knows in W_2.

It is an advantage of this approach to describing how an action affects what an agent knows that not only do we specify what he learns from the action, but also what he does not learn. Our analysis gives us necessary, as well as sufficient, conditions for :A's knowing that P is true after event :E. In the case of an action that is not informative, we can infer that, unless :A knows before performing the action whether P would be true, he will not know afterwards either. In the case of an informative action such that what is learned is whether Q is true, he will not know whether P is true unless he does already--or knows of some dependence of P on Q.

Within the context of this possible-world analysis of the effects of action on knowledge, we can formalize the requirements for a test that we presented in Section I. Suppose that TEST is the action of testing the acidity of a particular solution with blue litmus paper, RED is a propositional constant (a predicate of zero arguments) whose truth depends on the color of the litmus paper, and ACID is a propositional constant whose truth depends on whether the solution is acidic. The relevent fact about TEST is that the paper will be red after an agent A performs the test if and only if the solution is acidic at the time the test is performed:

(ACID ⊃ RES(DO(A,TEST),RED)) ∧
(¬ACID ⊃ RES(DO(A,TEST),¬RED))

In Section I we listed three conditions that ought to be sufficient for an agent to determine, by observing the outcome of a test, whether some unobservable precondition holds; in this case, for A to determine whether ACID is true by observing whether RED is true after TEST is performed:

(1) After A performs TEST, he knows whether RED is true.

(2) After A performs TEST, he knows that he has just performed TEST.

(3) A knows that RED will be true after TEST is performed just in case ACID was true before it was performed.

Conditions (1) and (2) will be satisfied if TEST is an informative action, such that the knowledge provided is whether RED is true in the resulting situation:

$$\begin{aligned}
&\text{T1. } \forall w_1, w_2, a_1 \\
&(R(:DO(a_1, :TEST), w_1, w_2) \supset \\
&\quad \forall w_3 (K(a_1, w_1, w_3) \equiv \\
&\quad\quad (\exists w_4 (K(a_1, w_1, w_4) \wedge R(:DO(a_1, :TEST), w_4, w_3)) \wedge \\
&\quad\quad\quad (:RED(w_2) \equiv :RED(w_3)))))
\end{aligned}$$

If :RED and :TEST are the metalanguage analogues of RED and TEST, T1 says that for any possible worlds W_1 and W_2 such that W_2 is the result

of an agent's performing TEST in W_1, the worlds that are compatible with what the agent knows in W_2 are exactly those that are the result of his performing TEST in some world that is compatible with what he knows in W_1, and in which RED has the same truth-value as in W_2. In other words, after performing TEST, the agent knows that he has done so and he knows whether RED is true in the resulting situation. As with our other axioms, the fact that it holds for all possible worlds makes it common knowledge.

Thus, A can use TEST to determine whether the solution is acid, provided that (1) is also satisfied. We can state this very succinctly if we make the further assumption that A knows that performing the test does not affect the acidity of the solution.[7] Given the axiom T1 for test, it is possible to show that

$$\text{ACID} \supset \text{RES}(\text{DO}(A,\text{TEST}),\text{KNOW}(A,\text{ACID})) \text{ and}$$
$$\neg\text{ACID} \supset \text{RES}(\text{DO}(A,\text{TEST}),\text{KNOW}(A,\neg\text{ACID}))$$

are true, provided that

$$\text{KNOW}(A,(\text{ACID} \supset \text{RES}(\text{DO}(A,\text{TEST}),(\text{ACID} \wedge \text{RED})))) \text{ and}$$
$$\text{KNOW}(A,(\neg\text{ACID} \supset \text{RES}(\text{DO}(A,\text{TEST}),(\neg\text{ACID} \wedge \neg\text{RED}))))$$

are both true and A is a rigid designator. We will carry out the proof in one direction, showing that, if the solution is acidic, after the test has been conducted the agent will know that it is acidic.

Given: TRUE(KNOW(A,IMP(ACID,RES(DO(A,TEST),AND(ACID,RED)))))

TRUE(KNOW(A,IMP(NOT(ACID),RES(DO(A,TEST),
AND(NOT(ACID),NOT(RED))))))

TRUE(ACID)

Prove: TRUE(RES(DO(A,TEST),KNOW(A,ACID)))

1. $\forall w_1$ (K(:A,W_0 ,w_1) ⊃ (:ACID(w_1) ⊃ $\exists w_2$ (R(:DO(:A,:TEST),w_1 ,w_2) ∧ :ACID(w_2) ∧ :RED(w_2)))). — Given,L1,L4,R2, L2,L9,L12,L11a

2. $\forall w_1$ (K(:A,W_0 ,w_1) ⊃ (¬:ACID(w_1) ⊃ $\exists w_2$ (R(:DO(:A,:TEST),w_1 ,w_2) ∧ ¬:ACID(w_2) ∧ ¬:RED(w_2)))) — Given,L1,L4,R2,L2, L6,L9,L12,L11a

3. :ACID(W_0) — L1,L9

4. :ACID(W_0) ⊃ $\exists w_2$ (R(:DO(:A,:TEST),W_0 ,w_2) ∧ :ACID(w_2) ∧ :RED(w_2)) — 1,K2

5. R(:DO(:A,:TEST),W_0 ,W_1) — 3,4

6. :RED(W_1) — 3,4

7.	$\forall w_2\ (K(:A, W_1, w_2) \equiv (\exists w_3\ (K(:A, W_0, w_3) \wedge R(:DO(:A, :TEST), w_3, w_2)) \wedge (:RED(W_1) \equiv :RED(w_2))))$	5,T1
8.	$K(:A, W_1, w_2)$	ASS
9.	$K(:A, W_0, W_3)$	7,8
10.	$R(:DO(:A, :TEST), W_3, w_2)$	7,8
11.	$:RED(W_1) \equiv :RED(w_2)$	7,8
12.	$:RED(w_2)$	6,11
13.	$\neg:ACID(W_3) \supset \exists w_4\ (R(:DO(:A, :TEST), W_3, w_4) \wedge \neg:ACID(w_4) \wedge \neg:RED(w_4))$	2,9
14.	$\neg:ACID(W_3)$	ASS
15.	$R(:DO(:A, :TEST), W_3, W_4)$	13,14
16.	$\neg:RED(W_4)$	13,14
17.	$w_2 = W_4$	15,R1
18.	$\neg:RED(w_2)$	16,17
19.	FALSE	12,18
20.	$:ACID(W_3)$	DIS(14,19)

21.	$:ACID(W_3) \supset \exists w_4 (R(:DO(:A,:TEST),W_3,w_4) \wedge :ACID(w_4) \wedge :RED(w_4))$	1,9
22.	$R(:DO(:A,:TEST),W_3,W_4)$	20,21
23.	$:ACID(W_4)$	20,21
24.	$w_2 = W_4$	15,22
25.	$:ACID(w_2)$	23,24
26.	$K(:A,W_1,w_2) \supset :ACID(w_2)$	DIS(8,25)
27.	$R(:DO(:A,:TEST),W_0,W_1) \wedge \forall w_2 (K(:A,W_1,w_2) \supset :ACID(w_2))$	5,26
28.	TRUE(RES(DO(A,TEST),KNOW(A,ACID)))	27,L9,L11a,L12, K2,R2,L1

The possible-world structure for this proof is depicted in Figure 8. Lines 1 and 2 translate the premises into the metalanguage. Since A knows that, if the solution is acidic, performing the test will result in the litmus paper's being red, it must be true in the actual world (W_0) that, if the solution is acidic, performing the test will result in the litmus paper's being red (Line 3). Suppose that, in fact, the solution is acidic (Line 4). Then, if W_1 is the result of performing the test in W_0 (Line 5), the paper will be red in W_1 (Line 6).

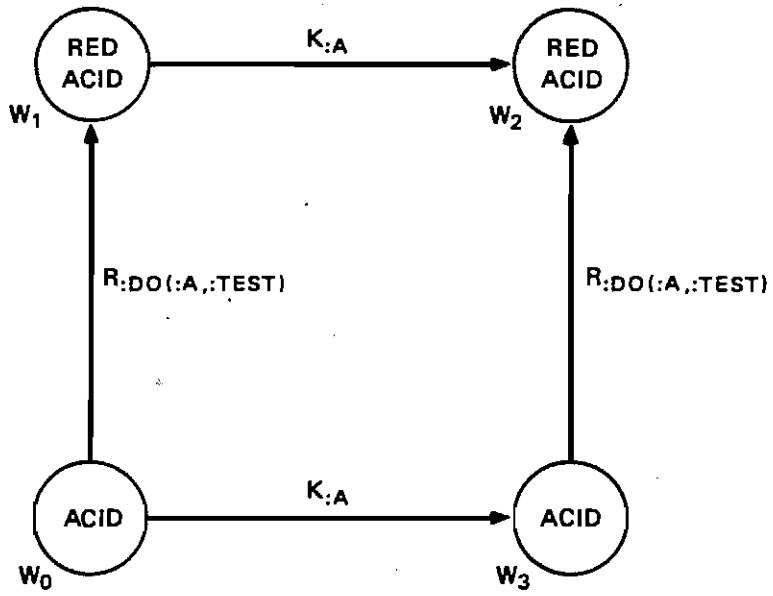

FIGURE 8 THE EFFECT OF A TEST ON THE AGENT'S KNOWLEDGE

Furthermore, the worlds that are compatible with what A knows in W_1 are those that are the result of his performing the test in some world that is compatible with what he knows in W_1, and in which the paper is red if and only if it is red in W_1 (Line 7). Suppose that w_2 is a world that is compatible with what A knows in W_1 (Line 8). Then there is a W_3 that is compatible with what A knows in W_0 (Line 9), such that w_2 is the result of A's performing the test in W_3 (Line 10). The paper is red in w_2, if and only if it is red in W_1 (Line 11); therefore, it is red in w_2

(Line 12). Since A knows how the test works, if the solution were not acidic in W_3, it would not be acidic, and the paper would not be red, in w_2 (Line 13).

Now, suppose the solution were not acid in W_3 (Line 14). If W_4 is the result of A's performing the test in W_3 (Line 15), the paper would not be red in W_4 (Line 16). But w_2 is the result of A's performing the test in W_3 (Line 17), so the paper would not be red in w_2 (Line 18). We know this is false (Line 19), however, so the solution must be acidic in W_3 (Line 20). If the solution is acidic in W_3, it must also be acidic in the situation resulting from A's performing the test in W_3 (Lines 21-23), but this is w_2 (Line 24). Therefore, the solution is acidic in w_2 (Line 25). Hence, in W_1, A knows that the solution is acidic (Line 26), so in the situation resulting from A's performing the test in W_0, he knows that the solution is acidic (Line 27). In other words (Line 28), A's performing the test would result in his knowing that the solution is acidic.

By an exactly parallel argument, we could show that, if the solution were not acidic, A could also find that out by carrying out the

test, so our analysis captures the sort of reasoning about tests that we described in Section I, based on general principles that govern the interaction of knowledge and action.

NOTES

[1] This paper presents the analysis of knowledge and action, and the representation of that analysis in first-order logic, that were developed in the author's doctoral thesis (Moore, 1980). The material in Sections III-A and III-B, however, has been substantially revised.

[2] Chapters 6 and 7 of (Moore, 1980) present a procedural interpretation of the axioms for knowledge and action given in this paper that seems to produce reasonably efficient behavior in an automatic deduction system.

[3] "Mary's telephone number" would be an appropriate way of telling someone what John's telephone number was if he already knew Mary's telephone number, but this knowledge would consist in knowing what expression of the type "321-1234" denoted Mary's telephone number. Therefore, even in this case, using "Mary's telephone number" to identify John's telephone number would just be an indirect way of getting to the standard indentifier.

[4] This amounts to an assumption that all events are deterministic, which might seem to be an unnecessary limitation. From a pragmatic

standpoint, however, it doesn't matter whether we say that a given event is nondeterministic, or we say that it is deterministic but no one knows precisely what the outcome will be. If we treated events as being nondeterministic, we could say that an agent knows exactly what situation he is in, but, because :E is nondeterministic, he doesn't know what situation would result if :E occurs. It would be completely equivalent, however, to say that :E is deterministic, and that the agent does not know exactly what situation he is in because he doesn't know what the result of :E would be in that situation.

[5] It would be more precise to say that DO(A,ACT) names a type of event rather than an individual event, since an agent can perform the same action on different occasions. We would then say that RES and R apply to event types. We will let the present usage stand, however, since we have no need to distinguish event types from individual events in this paper.

[6] R7 guarantees that the sequences $\langle\langle E_1, E_2\rangle, E_3\rangle$ and $\langle E_1, \langle E_2, E_3\rangle\rangle$ always define the same accessibility relation on situations; so, just as one would expect, we can regard sequence operators as being associative. Thus, when we have a sequence of more than two events or actions, we will not feel obliged to indicate a pairwise grouping.

[7] We have to add this extra condition to be able to infer that the

agent knows whether the solution is acidic, instead of merely that he knows whether it was acidic. The latter is a more general characteristic of tests, since it covers destructive as well as nondestructive tests. We have not, however, introduced any temporal operators into the object language that would allow us to make such a statement, although there would be no difficulty in stating the relevant conditions in the object language. Indeed, this is precisely what is done by axioms such as T1.

REFERENCES

Frege, G. (1949) "On Sense and Nominatum," in Readings in Philosophical Analysis, H. Feigl and W. Sellars, eds., pp. 85-102 (Appleton-Century-Crofts, Inc., New York, New York).

Hintikka, J. (1962) Knowledge and Belief (Cornell University Press, Ithica, New York).

Hintikka, J. (1971) "Semantics for Propositional Attitudes," in Reference and Modality, L. Linsky, ed., pp. 145-167 (Oxford University Press, London, England).

Hughes, G. E. and M. J. Cresswell (1968) An Introduction to Modal Logic (Methuen and Company, Ltd., London, England).

Konolige, K. (1984) "Belief and Incompleteness," to appear in Formal Theories of the Commonsense World, J. R. Hobbs and R. C. Moore, eds., (Ablex Publishing Corp., Norwood, New Jersey).

Kripke, S. A. (1963) "Semantical Analysis of Modal Logic," Zeitschrift fuer Mathematische Logik und Grundlagen der Mathematik, Vol. 9, pp. 67-96.

Kripke, S. A. (1971) "Semantical Considerations on Modal Logic," in Reference and Modality, L. Linsky, ed., pp. 63-72 (Oxford University Press, London, England).

Kripke, S. A. (1972) "Naming and Necessity," in Semantics of Natural Language, D. Davidson and G. Harmon, eds., pp. 253-355 (D. Reidel Publishing Company, Dordrecht, Holland).

McCarthy, J. (1962) "Towards a Mathematical Science of Computation," in Information Processing, Proceedings of IFIP Congress 62, C. Popplewell, ed., pp. 21-28 (North-Holland Publishing Company, Amsterdam, Holland).

McCarthy, J. (1968) "Programs with Common Sense," in Semantic Information Processing, M. Minsky, ed., pp. 403-418 (The MIT Press, Cambridge, Massachussets).

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

Moore, R. C. (1980) "Reasoning About Knowledge and Action," Artificial Intelligence Center Technical Note 191, SRI International, Menlo Park, California (October 1980).

Quine, W. V. O. (1971) "Quantifiers and Propositional Attitudes," in Reference and Modality, L. Linsky, ed., pp. 101-111 (Oxford University Press, London, England).

Reiter, R. (1980) "A Logic for Default Reasoning," Artificial Intelligence, Vol. 13, Nos. 1-2, pp. 81-113 (April 1980).

Rescher, N. and A. Urquhart (1971) Temporal Logic (Springer-Verlag, Vienna, Austria, 1971).

Russell, B. (1949) "On Denoting," in Readings in Philosophical Analysis, H. Feigl and W. Sellars, eds., pp. 103-115 (Appleton-Century-Crofts, Inc., New York, New York).

Scott, D. (1970) "Advice on Modal Logic," in Philosophical Problems in Logic: Some Recent Developments, K. Lambert, ed. (D. Reidel Publishing Company, Dordrecht, Holland).