IJCAI-77

5TH INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE -1977

IJCAI-77 • PROCEEDINGS OF THE CONFERENCE

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
CAMBRIDGE, MASSACHUSETTS, USA
AUGUST 22 - 25, 1977

VOLUME ONE

Sponsored by the
International Joint Conferences on Artificial Intelligence

Inquiries for purchase of copies of IJCAI Proceedings may be made to:

IJCAI-77 (Fifth IJCAI -- these volumes):
IJCAI-77
Department of Computer Science
Carnegie-Mellon University
Pittsburgh, PA 15213, USA

IJCAI-75 (Fourth IJCAI):
The Artificial Intelligence Laboratory
Publications Department
545 Technology Square
Cambridge, MA 02139, USA

IJCAI-73 (Third IJCAI):
SRI International
Publications Department
333 Ravenswood Avenue
Menlo Park, CA 94025, USA

IJCAI-71 (Second IJCAI):
The British Computer Society
29 Portland Place
London W1N 4AP, England

IJCAI-69 (First IJCAI):
The MITRE Corp.
Bedford, MA 01730, USA

REASONING ABOUT KNOWLEDGE AND ACTION

Robert C. Moore
Artificial Intelligence Laboratory
Stanford University
Stanford, California 94305

Abstract

This paper discusses the problems of representing and reasoning with information about knowledge and action. The first section discusses the importance of having systems that understand the concept of knowledge, and how knowledge is related to action. Section 2 points out some of the special problems that are involved in reasoning about knowledge, and section 3 presents a logic of knowledge based on the idea of possible worlds. Section 4 integrates this with a logic of actions and gives an example of reasoning in the combined system. Section 5 makes some concluding comments.

1. Introduction

One of the most important concepts an intelligent system needs to understand is the concept of knowledge. AI systems need to understand what knowledge they and the systems or people they interact with have, what knowledge is needed to achieve particular goals, and how that knowledge can be obtained. This paper develops a formalism that provides a framework for stating and solving problems like these. For example, suppose that there is a safe that John wants to open. The common sense inferences that we would like to make might include:

> If John knows the combination, he can immediately open the safe.
>
> If John does not know the combination, he cannot immediately open the safe.
>
> If John knows where the combination is written, he can read the combination and then open the safe.

In thinking about this example, consider how intimately the concept of knowledge is tied up with action. Reasoning about knowledge alone is of limited value. We may want to conclude from the fact that John knows A and B that he must also know C and D, but the real importance of such information is usually that it tells us something about what John can do or is likely to do. A major goal of my research has been to work out some of the interactions of knowing and doing.

That this area has received little attention in AI is somewhat surprising. It is frequently stated that good interactive AI programs will require good models of the people they are communicating with. Surely, one of the most important aspects of a model of another person is a model of what he knows. The only serious work on these problems in AI which I am aware of is a brief disscussion in McCarthy and Hayes (1969), and some more recent unpublished writings of McCarthy. In philosophy there is a substantial literature on the logic of knowledge and belief. A good introduction to this is Hintikka (1962) and papers by Quine, Kaplan, and Hintikka in Linsky (1971). Many of the ideas I will use come from these papers.

In representing facts about knowledge and actions, I will use first-order predicate calculus, a practice which is currently unfashionable. It seems to be widely believed that use of predicate calculus necessarily leads to inefficient reasoning and information retrieval programs. I believe that this is an overreaction to earlier attempts to build domain-independent theorem provers based on resolution. More recent research, including my own M.S. thesis (Moore, 1975), suggests that predicate calculus can be treated in a more natural manner than resolution and combined with domain-dependent control information for greater efficiency. Furthermore, the problems of reasoning about knowledge seem to require the full ability to handle quantifiers and logical connectives which only predicate calculus posseses.

Section 2 of this paper attempts to bring out some of the special problems involved in reasoning about knowledge. Section 3 presents a formalism which I believe solves these problems, and Section 4 integrates this with a formalism for actions. Section 5 makes some concluding comments.

2. Problems in Reasoning about Knowledge

Reasoning about knowledge presents special difficulties. It turns out that we cannot treat "know" as just another relation. If we can represent "Block1 is on Block2" by On(Block1,Block2), we might be tempted to represent "John knows that P" simply by Know(John,P). This approach glosses over a number of problems. We might be suspicious from the first, since P is not the name of an object but is rather a sentence (or proposition). The semantics of predicate calculus forbid the arbitrary intermingling of sentences and terms for good reason. For one thing, the second argument position of Know is a *referentially opaque context*. Ordinarily in logic we can freely substitute an expression for one that is extensionally equivalent (i.e., one that has the same referent or truth value), without affecting the truth of the formula that contains the expression. This is called *referential transparency*. For example, if X + Y = 7 and X = 3, then 3 + Y = 7. This pattern of reasoning is not valid with Know. We cannot infer from Know(John,(X + Y = 7)) and X = 3 that Know(John,(3 + Y = 7)) is true, since John might not know the value of X.

One possible solution to this problem is to make the second argument of Know the name of a formula rather than the formula itself. This is essentially the same idea as Goedel numbering, although it is not necessary to use such an obscure encoding as the natural numbers. We won't specify exactly how the encoding is done, but simply use "P" to represent a term denoting the formula P. The representation of "John knows that P" now becomes Know(John,"P"). We are no longer in any danger of infering Know(John,"P(A)") from Know(John,"P(B)") and A = B, because A is not contained in "P(A)". Only the name of A, i.e. "A", is contained, and since "A" does not equal "B", there is no problem.

There is, however, a more serious problem, the fact that people can reason with their knowledge. We would expect a reasoning system to have built into it the ability to conclude B from A and A ⊃ B. But if we treat Know as just an ordinary predicate, we will have no reason to suppose that Know(John,"A") and Know(John,"A ⊃ B") might suggest Know(John,"B"). This problem is emphasised by the fact that there is no formal connection between a formula and its name. The fact that we regard "P" as the name of P is entirely outside the system. To get around this, it is necessary to re-axiomatize the rules of logic within the system, e.g. ∀a,p,q(Know(a,"p ⊃ q") ∧ Know(a,"p") ⊃ Know(a,"q")). But if we hope to do automated reasoning, this amounts to re-programming the deductive system in first-order logic, and using the top-level inference routines as the interpreter. When we consider the complexities of quantification and matching, it seems likely that this would be an inefficient process.

A different idea which initially seems very appealing is to use the multiple data-base capabilities of advanced AI languages to set up a separate data base for each person whose knowledge we have some information about. We then can record what we know about his knowledge in that data base, and simulate his reasoning by running our standard inference routines in that data base. This idea seems to have wide currency in AI circles, and I advocated it myself in an earlier paper (Moore, 1973).

Unfortunately, it doesn't work very well. It can handle simple statements of the form "John knows that P," but more complicated expressions cause trouble. Consider "John knows that P or John knows that Q." We can't represent this by simply adding "P or Q" to the data base representing John's knowledge, because this would mean "John knows that P or Q," - something quite different. We could try setting up two data bases, DB1 and DB2, add "P" to one and "Q" to the other, and then assert in the main data base "DB1 represents John's knowledge, or DB2 represents John's knowledge." However, if we also wanted to assert "John knows that C, or John knows that D, or John knows that E," we would need *six* data bases to represent all the possibilites for John's knowledge - one for each of the combinations "A" and "C", "B" and "C", "A" and "D", etc. As we add more disjunctive assertions, we get a combinatorial explosion in the number of data bases.

We also have a problem in representing "John doesn't know that P." We can't add "not P" to John's data base, because this would be asserting "John knows that not P," and simply omitting "P" from John's data base means that *we* don't know whether John knows that P. So it seems that what John doesn't know has to be kept separate from what he does know. But there are inferences that require looking at both. For example, if we have "John doesn't know that P," and "John knows that Q implies P," we might want to conclude that "John doesn't know that Q," is probably true. This is representative of a class of inferences that the data base approach doesn't capture. There seems to be a fundamental problem in saying things about a person's knowledge that go beyond simply enumerating what he knows.

3. Reasoning about Knowledge via Possible Worlds

While there may be ways to directly attack the difficulties we have been discussing, there is a way to avoid them entirely by reformulating the problem in terms of possible worlds. When we want to reason about someone's knowledge, rather than talking about what facts he knows, we will talk about which of the various possible worlds might be, so far as he knows, the real world. A person is never completely sure which possible world (or possible state of *the* world) he is in, because his knowledge is incomplete. We will be willing to conclude that a person knows a particular fact, if the fact is true in all the worlds that are possible according to what he knows. This idea is due to Hintikka (1969), and is an adaptation of the semantics for modal logic developed chiefly by Kripke (1963).

Hintikka uses these ideas about possible worlds to provide a model theory for a modal logic of knowledge. In order to use this theory directly for reasoning, we will axiomatize it in first-order logic. To do this, we must encode a language that talks about knowing facts (which we will call the object language) into term expressions of a first-order language that talks about possible worlds (which we will call the meta-language). Then we will have a relation T, such that T(W,P) means the object-language formula denoted by P is true in the possible world denoted by W. So that we can talk more easily about truth in the actual world, we will have a predicate True, such that True(P) ≡ T(W0,P), where W0 is a constant which refers to the actual world. We will also have a relation K(A,W1,W2), which means that W2 is a world which is possible according to what A knows in W1. The fundamental axiom of knowledge is then ∀w1,a,p(T(w1,Know(a,p) ≡ ∀w2(K(a,w1,w2) ⊃ T(w2,p)). This simply says that a person knows the facts that are true in every world that is possible according to what he knows.

One problem with this axiom is that it is not universally true. For a person to know everything that is true in all worlds which are possible as far as he knows, he would have to know all the logical consequences of his knowledge. Of course, he can know only some of them. But in any particular case, if we can see that a certain conclusion follows from someone's knowledge, we are probably justified in assuming that he can see this also. So we can regard this axiom as a rule of plausible inference, using it when needed, but being prepared to retract our conclusions if they generate contradictions. I will not attempt here to devlop a general theory of plausible reasoning, but I believe that a theory can be worked out that will allow us to use this axiom in essentially its current form.

I should clarify what type of possible worlds I have in mind. Rather than all logically possible worlds, we will consider only those worlds which are possible according to "common knowledge". So, I will feel free to say that facts like "Fish live in water," are true in all possible worlds. This gives us an easy way of saying that not only does everyone know something, but everyone knows that everyone knows it, and everyone knows that everyone knows that everyone knows, etc.

We can now give the full axiomatization of knowledge in terms of possible worlds:

L1. True(p1) ≡ T(W0,p1)
L2. T(w1,(p1 And p2)) ≡ (T(w1,p1) ∧ T(w1,p2))
L3. T(w1,(p1 Or p2)) ≡ (T(w1,p1) ∨ T(w1,p2))
L4. T(w1,(p1 => p2)) ≡ (T(w1,p1) ⊃ T(w1,p2))
L5. T(w1,(p1 <=> p2)) ≡ (T(w1,p1) ≡ T(w1,p2))
L6. T(w1,Not(p1)) ≡ ¬T(w1,p1)

K1. T(w1,Know(a1,p1)) ≡ ∀w2(K(a1,w1,w2) ⊃ T(w2,p1))
K2. K(a1,w1,w1)
K3. K(a1,w1,w2) ⊃ (K(a1,w2,w3) ⊃ K(a1,w1,w3))
K4. K(a1,w1,w2) ⊃ (K(a1,w1,w3) ⊃ K(a1,w2,w3))

Axioms L1 - L6 just translate the logical connectives from the object language to the meta-language, using the ordinary Tarski definition of truth. For instance, according to L2, (A And B) is true in a world if and only if A is true in the world and B is true in the world. K1 is the fundamental axiom of knowledge which we already looked at. K2 says that each world is possible as far as anyone in that world can tell, which is another way of saying that if something is known then it is true. Although it may not be obvious, K3 and K4 imply that everyone knows whether he knows a certain fact. K2 - K4 imply that for fixed A, K(A,w1,w2) is an equivalence relation. This makes our logic of knowledge isomorphic to the modal logic S5. The correspondence between various modal logics and and possible-worlds models for them is discussed in Kripke(1963).

This representation gives us what we need. The meta-language translations of the object-language statements have a structure that reflects their logical properties. To illustrate the use of these axioms, we can prove that people can do simple inferences:

Given: True(Know(A,P) And Know(A,(P => Q)))

Prove: True(Know(A,Q))

```
 1. True(Know(A,P) And Know(A,(P => Q)))             Given
 2. T(W0,(Know(A,P) And Know(A,(P => Q))))           L1,1
 3. T(W0,Know(A,P)) ∧ T(W0,Know(A,(P => Q)))         L2,2
 4. T(W0,Know(A,P))                                  3
 5. K(A,W0,w1) ⊃ T(w1,P)                             K1,4
 6. T(W0,Know(A,(P => Q)))                           3
 7. K(A,W0,w1) ⊃ T(w1,(P => Q))                      K1,6
 8.    K(A,W0,w1)                                    Ass
 9.    T(w1,P)                                       5,8
10.    T(w1,(P => Q))                                7,8
11.    T(w1,P) ⊃ T(w1,Q)                             L4,10
12.    T(w1,Q)                                       11,9
13. K(A,W0,w1) ⊃ T(w1,Q)                             Dis(8,12)
14. T(W0,Know(A,Q))                                  K1,13
15. True(Know(A,Q))                                  L1,14
```

Proofs in this paper use natural deduction. The right hand column gives the axioms and preceding lines which justify each step. Indented sections are subordinate proofs, and **Ass** marks the assumptions on which these subordinate proofs are based. **Dis** indicates the discharge of an assumption.

This proof is completely straight-forward. Lines 1 - 7 simply expand the given facts into possible-worlds notation. Then we pick w1 as a typical world which is possible according to what A knows. In lines 9 - 12, we do the inference that we want to attribute to A. Since this inference can be done in an abitrarily chosen member of the set of worlds which are possible for A, it must be valid in all of them (line 13). From this we conclude that A can probably do the inference also (lines 14 - 15).

So far I have avoided dealing with the problem of quantifiers. Exactly what do expressions like ∃x(Know(A,P(x))). mean? This is not a simple assertion that someone knows a certain fact, so its intuitive meaning may not be clear. The best paraphrase seems to be "There is something that A knows has property P." It is a matter of great dispute in philosophy exactly how to handle this. I will take a pragmatic approach. To say that a person knows of something that it has property P means that he can name something that has property P. Furthermore, just any sort of name won't do. "The thing that has property P" is no good, for instance. We will say that A must know the standard name of the thing that has P. This is, of course, a simplification. Not all things have standard names, and some things have different standard names in different contexts, but we will ignore these difficulties to preserve the simplicity of the ordinary case. Abstract entities usually have unproblematical standard names - "23" is the standard name of 23, "15 + 8" is not.

Turning to the model theory, the interpretation of the formula we are considering would be that there is something that is P in all worlds compatible with what A knows. That means that standard names must refer to the same thing in all possible worlds. There is a term for this in philosophy, *rigid designator.* We can greatly simplify our formalism if we require that all ordinary terms in the object language be rigid designators. We would then have to have a special notation for non-rigid designators, but this will not come up in our examples, so I will not develop that idea here. We can now give the axioms for quantifiers and equality:

L7. T(w1,Exist(v1,P)) ≡ ∃x(T(w1,P[x/v1]))
provided x is not free in P

L8. T(w1,All(v1,P)) ≡ ∀x(T(w1,P[x/v1]))
provided x is not free in P

L9. T(w1,Eq(x1,x2)) ≡ (x1 = x2)

L7 and L8 are axiom schemas relative to P and x, and v1 is a meta-language variable that ranges over object-language variables. P[x/v1] means the result of substituting x for v1 in P.

These three axioms may seem somewhat peculiar in that they appear to say that individuals in the world can be part of object-language expressions. In L7 and L8, we took x, a variable ranging over real objects, and inserted it into P, the name of a sentence, implying that objects can be contained in sentences. To preserve the simplicity of the notation, without this apparent absurdity, we will make the interpretation that all functions which represent atomic predicates in the object language (e.g. Eq) take individuals as arguments and return expressions containing the standard names of those individuals.

4. Integrating Knowledge and Action

In order to integrate knowledge with actions, we need to formalize a logic of actions in terms comparable to our logic of knowledge. Happily, the standard AI way of looking at actions does just that. Most AI programs that reason about actions view the world as a set of possible situations, and each action determines a binary relation on situations, one situation being the outcome of performing the action in the other situation. We will integrate knowledge and action by identifying the possible worlds in our logic of knowledge with the possible situations in our logic of actions.

First, we need to define our formalism for actions exactly parallel to our formalism for knowledge. We will have an object-language relation **Res(E,P)** which says that it is possible for event E to occur, and P would be true in the resulting situation. In the meta-language, we will have the corresponding relation R(E,W1,W2) which says that W2 is a possible situation/world which could result from event E happening in W1. These two concepts are related in the following way:

R1. T(w1,Res(e1,p1)) ≡
(∃w2(R(e1,w1,w2)) ∧ ∀w2(R(e1,w1,w2) ⊃ T(w2,p1)))

The existential clause on the right side of R1 says that it is possible for the event to occur, and the universal clause says that in every possible outcome the condition of interest is true. There is a direct parallel here with concepts of program correctness, the first clause expressing termination, and the second, partial correctness.

We can extend the parallel with programming-language semantics to the structure of actions. We will have a type of event which is an actor performing an action, Do(A,C). (C stands for "command".) Actions can be built up from simpler actions using loops, conditionals, and sequences:

R2. T(w1,Res(Do(a1,Loop(p1,c1)),p2) ≡
T(w1,Res(Do(a1,If(p1,(c1;Loop(p1,c1)),Nil)),p2))

R3. T(w1,Res(Do(a1,If(p1,c1,c2)),p2) ≡
((T(w1,Know(a1,p1)) ∧ T(w1,Res(Do(a1,c1),p2))) ∨
(T(w1,Know(a1,Not(p1))) ∧ T(w1,Res(Do(a1,c2),p2))))

$$\begin{aligned}&\text{R4. } T(w1,Res(Do,a1,(c1;c2)),p1)) \equiv \\ &\qquad T(w1,Res(Do(a1,c1),Res(Do(a1,c2),p1)))\end{aligned}$$

$$\text{N1. } R(Do(a1,Nil),w1,w2) \equiv (w1 \equiv w2)$$

R2 defines the step-by step expansion of while-loops: if the test is true, execute the body and repeat the loop, else do nothing. To prove general results we would need some sort of induction axiom. R3 defines the execution of a conditional action. Notice that being able to execute a conditional requires *knowing* whether the test condition is true. This differs from ordinary program conditionals, where the test condition is either assumed to be a decidable primitive, or is itself a piece of code to be executed. R4 says that the result of carrying out a sequence of actions is the result of executing the first action, and then executing the rest. N1 simply defines the no-op action we need for the definition of Loop.

One of the most important problems we want to look at is how knowledge affects the ability to achieve a goal. Part of the answer is given in the definition of the notion Can. We will say that a person can bring about a condition if and only if there is an action which he knows will achieve the condition:

$$\text{C1. } T(w1,Can(a1,p1)) \equiv \exists c1(T(w1,Know(a1,Res(Do(a1,c1),p))))$$

The idea is that to achieve something, a person must know of a plan for achieving it, and then be able to carry out the plan.

We have seen a couple of ways that knowledge affects the possibility of action in R3 and C1. We now want to describe how actions affect knowledge. For actions that are not information-acquiring, we can simply say that the actor knows that he has performed the action. Since our axiomatization of particular actions implies that everyone knows what their effects are, this is sufficient. For information-acquiring actions, like looking at something, we will also add that the information has been acquired. This is best explained by a concrete example. Below, we will work out an example about opening safes, so we will now look at the facts about dialing combinations:

$$\begin{aligned}&\text{D1. } \exists w2(R(Do(a1,Dial(x1,x2)),w1,w2) \equiv \\ &\qquad (T(w1,Comb(x1)) \wedge T(w1,Safe(x2)) \wedge T(w1,At(a1,x2)))\end{aligned}$$

$$\begin{aligned}&\text{D2. } R(Do(a1,Dial(x1,x2)),w1,w2) \supset \\ &\qquad ((T(w1,\text{Is-comb-of}(x1,x2)) \supset T(w2,Open(x2))) \wedge \\ &\qquad ((\neg T(w1,\text{Is-comb-of}(x1,x2)) \wedge \neg T(w1,Open(x2)) \supset \\ &\qquad\quad \neg T(w2,Open(x2))) \wedge \\ &\qquad (T(w1,Open(x2)) \supset T(w2,Open(x2)))\end{aligned}$$

$$\begin{aligned}&\text{D3. } R(Do(a1,Dial(x1,x2)),w1,w2) \supset \\ &\qquad (K(a1,w2,w3) \equiv ((T(w2,Open(x2)) \equiv T(w3,Open(x2))) \wedge \\ &\qquad\quad \exists w4(K(a1,w1,w4) \wedge R(Do(a1,Dial(x1,x2)),w4,w3)))\end{aligned}$$

D1 says that an actor can perform a dialing action if the thing he is dialing is a combination, the thing he is dialing it on is a safe, and he is at the same place as the safe. D2 tells how dialing a combination affects whether the safe is open: if the combination is the combination of the safe, then the safe will be open; if it is not the combination of the safe and the safe was locked, the safe stays locked; if the safe was already open, it stays open.

D3 describes how dialing affects the knowledge of the dialer. Roughly it says that the actor knows he has done the dialing, and he now knows whether the safe is open. More precisely, it says that the worlds that are now possible as far as he knows are exactly those which are the result of doing the action in some previously possible world, and in which the information acquired matches the actual world. Notice that by making the consequent of D3 a bi-conditional, we have said that the actor has not acquired any other information by doing the action. Also notice that D3 is more subtle than just saying that whatever he knew before he knows now. This is not strictly true. He might have known before that the safe was locked, and now know that the safe is open. According to D3, if the actor knows before the action "P is true", after the action he knows "P was true before I did this action."

Having presented the basic formalism, I would now like to work out a simple example to illustrate its use. Simply stated, what I will show is that if a person knows the combination of a safe, and he is where the safe is, he can open the safe. Besides the axioms for Dial, we will need two more domain-specific axioms:

$$\begin{aligned}&\text{A1. } T(w1,\text{Is-comb-of}(x1,x2)) \supset \\ &\qquad (T(w1,Comb(x1)) \wedge T(w1,Safe(x2)))\end{aligned}$$

$$\text{A2. } T(w1,At(a1,x1)) \supset T(w1,Know(a1,At(a1,x1)))$$

A1 says that if one thing is the combination of another, the first thing is a combination and the second thing is a safe. A2 says that a person knows what is around him. The proof is as follows:

Given: $True(At(John,Sf))$
$\qquad True(Exists(X1,Know(John,\text{Is-comb-of}(X1,Sf))))$

Prove: $True(Can(John,Open(Sf)))$

1.	$True(Exists(X1,Know(John,\text{Is-comb-of}(X1,Sf))))$	Given
2.	$T(W0,Exists(X1,Know(John,\text{Is-comb-of}(X1,Sf))))$	L1,1
3.	$T(W0,Know(John,\text{Is-comb-of}(C,Sf)))$	L7,2
4.	$K(John,W0,w1) \supset T(w1,\text{Is-comb-of}(C,Sf))$	K1,3
5.	$True(At(John,Sf))$	Given
6.	$T(W0,At(John,Sf))$	L1,5
7.	$T(W0,Know(John,At(John,Sf)))$	A2,6
8.	$K(John,W0,w1) \supset T(w1,At(John,Sf))$	K1,7
9.	$\quad K(John,W0,w1)$	Ass
10.	$\quad T(w1,\text{Is-comb-of}(C,Sf))$	4,9
11.	$\quad T(w1,Comb(C))$	A1,10
12.	$\quad T(w1,Safe(Sf))$	A1,10
13.	$\quad T(w2,At(John,Sf))$	8,9
14.	$\quad \exists w2(R(Do(John,Dial(C,Sf)),w1,w2)$	D1,11,12,13
15.	$\qquad R(Do(John,Dial(C,Sf)),w1,w2)$	Ass
16.	$\qquad T(w1,\text{Is-comb-of}(C,Sf)) \supset T(w2,Open(Sf))$	D2,15
17.	$\qquad T(w2,Open(Sf))$	16,10
18.	$\quad R(Do(John,Dial(C,Sf)),w1,w2) \supset T(w2,Open(Sf))$	Dis(15,17)
19.	$\quad T(w1,Res(Do(John,Dial(C,Sf)),Open(Sf)))$	R1,14,18
20.	$K(John,W0,w1) \supset$ $\quad T(w1,Res(Do(John,Dial(C,Sf)),Open(Sf)))$	Dis(9,19)
21.	$T(W0,Know(John,Res(Do(John,Dial(C,Sf)),Open(Sf))))$	K1,20
22.	$T(W0,Can(John,Open(Sf)))$	C1,21
23.	$True(Can(John,Open(Sf)))$	L1,22

The proof is actually simpler than it may look. The real work is done in the ten steps between 10 and 19; the other steps are the overhead involved in translating between the object language and the meta language. Notice that we did not have to say explictly that someone needs to know the combination in order to open a safe. Instead we said something more general, that it is necessary to know a procedure in order to do anything. In this case, the combination is part of that procedure. It may also be interesting to point out what would have happened if we had said only that John knew the safe had a combination, but not that he knew what it was. If we had done that, the existential quantifier in the second assertion would have been inside the scope of Know. Then the Skolem constant C would have depended on the variable w1, and the step from 20 to 21 would have failed.

5. Conclusions

In summary, the possible-worlds approach seems to have two major advantages as a tool for reasoning about knowledge. First, it allows "lifting" reasoning in knowledge contexts into the basic deductive system, eliminating the need for separate axioms or rules of inference for these contexts. Second, it permits a very elegant integration of the logic of knowledge with the logic of actions.

This approach seems to work very well as far as we have taken it, but there are some major issues we have not discussed. I have said nothing so far about procedures for reasoning automatically about knowledge. I have some results in this area which appear very promising, but they are too fragmentary for inclusion here. I have also avoided bringing up the frame problem, by not looking at any sequences of action. I am also working in this area, and I consider it one of the largest IOU's generated by this paper. However, the possible-worlds approach has an important advantage here. Whatever method is used to handle the frame problem, whether procedural or axiomatic, knowledge contexts will be handled automatically, simply by applying the method uniformly to all possible worlds. This should eliminate any difficulties of representing what someone knows about the frame problem.

6. References

Hintikka, J. (1963) *Knowledge and Belief.* Ithica, New York: Cornell University Press.

Hintikka, J. (1969) Semantics for Propositional Attitudes, in Linsky (1971), 145-167.

Kripke, S. (1963) Semantical Considerations on Modal Logic, in Linsky (1971), 63-72.

Linsky, L. (ed.) (1971) *Reference and Modality.* London: Oxford University Press.

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

Moore, R. C. (1973) D-SCRIPT: A Computational Theory of Descriptions. *Advance Papers of the Third International Joint Conference on Artificial Intelligence*, 223-229.

Moore, R. C. (1975) *Reasoning from Incomplete Knowledge in a Procedural Deduction System.* MIT Artificial Intelligence Laboratory, AI-TR-347.