From: AAAI-82 Proceedings. Copyright ©1982, AAAI (www.aaai.org). All rights reserved.

THE ROLE OF LOGIC IN KNOWLEDGE REPRESENTATION AND COMMONSENSE REASONING*

Robert C. Moore
Artificial Intelligence Center
SRI International, Menlo Park, California 94025

ABSTRACT

This paper examines the role that formal logic ought to play in representing and reasoning with commonsense knowledge. We take issue with the commonly held view (as expressed by Newell [1980]) that the use of representations based on formal logic is inappropriate in most applications of artificial intelligence. We argue to the contrary that there is an important set of issues, involving incomplete knowledge of a problem situation, that so far have been addressed only by systems based on formal logic and deductive inference, and that, in some sense, probably can be dealt with only by systems based on logic and deduction. We further argue that the experiments of the late 1960s on problem-solving by theorem-proving did not show that the use of logic and deduction in AI systems was necessarily inefficient, but rather that what was needed was better control of the deduction process, combined with more attention to the computational properties of axioms.

I INTRODUCTION

In his AAAI presidential address, Allen Newell [1980] presented his view of the role that logic ought to play in representing and reasoning with commonsense knowledge. Probably the most concise summary of that view is his proposition that "the role of logic [is] as a tool for the analysis of knowledge, not for reasoning by intelligent agents" [p. 16]. What I understand Newell to be saying is that, while logic provides an appropriate framework for analyzing the meaning of expressions in representation formalisms and judging the validity of inferences, logical languages are themselves not particularly good formalisms for representing knowledge, nor is the application of rules of inference to logical formulas a particularly good method for commonsense reasoning.

As to the first part of this position, I could not agree more. Whatever else a formalism may be, at least some of its expressions must have _referential semantics_ if the formalism is really to be a representation of _knowledge_. That is, there must be some sort of correspondence between an expression and the world, such that it makes sense to ask whether the world is the way the expression claims it to _be_. To have knowledge at all is to have knowledge** that the world is one way and not otherwise. If one's "knowledge" does not rule out any possibilities for how the world might be, then one really does not know anything at all. Moreover, whatever AI researchers may say, examination of their actual practice reveals that they do rely (at least informally) on being able to provide referential semantics for their formalisms. Whether we are dealing with conceptual dependencies, frames, semantic networks, or what have you, as soon as we say that a particular piece of structure represents the assertion (or belief, or knowledge) that John hit Mary, we have hold of something that is _true_ if John did hit Mary and _false_ if he didn't.

Now, mathematical logic (especially model theory) is simply the branch of mathematics that deals with this sort of relationship between expressions and the world. If one is going to provide an analysis of the referential semantics of a representation formalism, then, _a fortiori_, one is going to be engaged in logic. As Newell puts it [p. 17], "Just as talking of _programmerless_ programming violates truth in packaging, so does talking of a _non-logical_ analysis of knowledge." It may be objected that Newell and I are both overgeneralizing in defining logic so broadly as to include all possible methods for addressing this issue, but the fact remains that the only existing tools for this kind of semantic analysis have come from logic. I know this view is very controversial in AI, but I will

----

* Preparation of this paper was supported by the Defense Advanced Research Projects Agency under Contract N00039-80-C-0575 with the Naval Electronic Systems Command. The views and conclusions contained in this paper are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency of the United States Government.

** or at least a belief; most people in AI don't seem overly concerned about truth in the actual world.

not argue the point any further for two reasons. First, it has already been argued quite eloquently by Pat Hayes [1977], and second, I want to go on to those areas where I *disagree* with Newell.

The main point on which I take issue with Newell is his conclusion that logical languages and deductive inference are not very useful tools for implementing (as opposed to analyzing) systems capable of commonsense reasoning. Newell does not present any real argument in support of this position, but instead says [p. 17] "The lessons of the sixties taught us something about the limitations of using logics for this role." In my view, Newell has seriously misread the lessons of the sixties with regard to this issue.

It appears to me that a number of important features of commonsense reasoning can be implemented *only* within a logical framework. Consider the following problem, adapted from Moore [1975, p. 28]. Three blocks, A, B, and C, are arranged as shown:

```
| ‾‾‾‾‾ || ‾‾‾‾‾ || ‾‾‾‾‾ |
|   A   ||   B   ||   C   |
|_______||_______||_______|
```

A is green, C is blue, and the color of B is unstated. In this arrangement of blocks, is there a green block next to a block that is not green? It should be clear with no more than a moment´s reflection that the answer is "yes." If B is green, it is a green block next to the nongreen block C; if B is not green then A is a green block next to the nongreen block B.

How is a person able to solve this problem? What sort of reasoning mechanisms are required? At least three distinctly "logical" factors seem to be involved: (1) the ability to see that an existentially quantified proposition is true, without knowing exactly which object makes it true, (2) the ability to recognize that, for a particular proposition, either it or its negation must be true, and (3) the ability to reason by cases. So far as I know, none of these abilities are possessed by any AI system not explicitly based on formal logic. Moreover, I would claim that, in a strong sense, these issues can be addressed only by systems that are based on formal logic.

To justify this claim we will need to examine what it means to say that a system uses a logical representation or that it reasons by deductive inference. Then we will try to re-evaluate what was actually shown by the disappointing results of the early experiments on problem-solving by theorem-proving, which we must do if the arguments presented here are correct and if there is to be any hope of creating systems with commonsense reasoning abilities comparable to those possessed by human beings.

II WHAT IS A LOGICAL REPRESENTATION?

The question of what it means to use a logic for representing knowledge in a computer system is less straightforward than it might seem. In mathematics and philosophy, a logic is a language--i.e., a set of formulas--with either a formal inference system or a formal semantics (or both).* To use a logic in a computer system, we have to encode those formulas somehow as computer data structures. If the formulas are in "Cambridge Polish" notation, e.g.,

```
(EVERY X (IMPLIES (MAN X) (MORTAL X))),
```

we may be tempted to assume that the corresponding LISP S-expression must be the data structure that represents the formula in the computer. This is in fact the case in many systems, but using more sophisticated data structures certainly does not mean that we are not implementing a logical representation. For example, Sickel [1976] describes a theorem-proving system in which a collection of formulas is represented by a graph, where each node represents a formula, and each link represents a possible unification (i.e., pattern match) of two formulas, with the resulting substitution being stored on the link. Furthermore, Sickel notes that the topology of the graph, plus the substitutions associated with the links, carries all the information needed by the theorem-prover--so the actual structure of the formulas is not explicitly represented at all!

This example suggests that deficiencies attributed to logical representations may be artifacts of naive implementations and do not necessarily carry over when more sophisticated techniques are used. For instance, one of the most frequently claimed advantages of semantic nets over logic as a representation formalism is that the links in the semantic net make it easier to retrieve information relevant to a particular problem. Sickel´s system (along with that of Kowalski [1975]) would seem to be at least as good as most semantic net formalisms in this respect. In fact, it may even be better, since following a link in a semantic net usually does not guarantee that the subsequently attempted pattern match will succeed, while in Sickel´s or Kowalski´s system, it does.

Given that the relationship between a logical formula and its computer implementation can be as abstract as it is in Sickel´s system, it seems doubtful to me that we could give any clear criteria for deciding whether a particular system *really* implements a logical representation. I think that the best way out of this dilemma is to give up trying to draw a line between logical and

----

* For example, for several decades there were formal inference systems for modal logic [Hughes and Cresswell, 1968], but no semantics; Montague´s [1974] intensional logic has a formal semantics, but no inference system.

nonlogical representations, and instead ask what logical features particular representation formalisms posses. If we adopt this point of view, the next question to ask is what logical features are needed in a general-purpose representation formalism. My answer is that, at a minimum, we need all the features of first-order classical logic with equality.

Perhaps the most basic feature of first-order logic is that it describes the world in terms of objects and their properties and relations. I doubt that anyone in AI could really complain about this, as virtually all AI representation formalisms make use of these concepts. It might be argued that one needs more than just objects, properties, and relations as primitive notions, but it should be kept in mind that first-order logic places no limits on what can be regarded as an object. Times, events, kinds, organizations, worlds, and sentences--not just concrete physical objects--can all be treated as logical individuals. Furthermore, even if we decide we need "nonstandard" features such as higher-order or intensional operators, we can still incorporate them within a logical framework.

For me, however, it is not the basic "metaphysical" notions of object, property, and relation that are the essential features of logic as a representation formalism, but rather the kinds of assertions that logic lets us make about them. Most of the features of logic can be seen as addressing the problem of how to describe an incompletely known situation. Specifically: existential quantification allows us to say that something has a certain property without having to know which thing has that property. Universal quantification allows us to say that everything in a certain class has a certain property without having to know what everything in that class is. Disjunction allows us to say that at least one of two statements is true without having to know which statement is true. Negation allows us to distinguish between knowing that a statement is not true and not knowing that it is true. Finally, logic lets us use different referring expressions without knowing whether they refer to the same object, but provides us with the equality predicate to assert explicitly whether or not they do.

One way that logic has been criticized is not to claim that the above features are unnecessary or harmful, but rather to argue that logic lacks some other essential feature--for instance, the ability to express control information. This was the basis of the early MIT-led criticism of theorem-proving research (e.g., [Winograd, 1972, Chapter 6]), which was, I believe, largely justified. This sort of problem, however, can be addressed and, in fact, has been [Hayes, 1973] [McDermott, 1978] [Kowalski, 1979] [Moore, 1975] by extending logic in various ways (see Section III), rather than by throwing it out and starting over. Moreover, the criticism quickly turned into a much more radical attack on any use of logic or deduction at all in AI [Hewitt, 1973] [Hewitt, 1975] [Minsky, 1974, Appendix]. That assault, in my view, was tremendously detrimental to serious research on knowledge representation and commonsense reasoning and represents the position I primarily want to argue against.

The major reason I regard the features of first-order logic as essential to any general-purpose representation formalism is that they are applicable to expressing knowledge about any domain. That is, it doesn't really matter what part of the world we are talking about; it always may be the case that we have only partial knowledge of a situation and we need some of these logical features to express or reason with that knowledge. This can be seen in the example presented in Section I. Reasoning about the position and color of blocks is certainly no more inherently logical than reasoning about anything else. The logical complexity of the problem comes from the fact that we are asked whether any blocks satisfy a given condition, but not which ones, and that we don't know the color of the middle block. If we had a complete description of the situation--if we were told the color of the middle block--we could just "read off" the answer to the question from the problem description without doing any reasoning at all.

Similar situations can easily arise in more practical domains as well. For instance, in determining a course of treatment, a physician may not need to decide between two possible diagnoses, either because the treatment is the same in either case or because only one of the two is treatable at all. Now, as far as I know, none of the inference methods currently being used in expert systems for medical diagnosis are capable of doing the sort of general reasoning by cases that ultimately justifies the physician's actions in such situations. Some systems have ad hoc rules or procedures for these special cases, but the creators of the systems have themselves had to carry out the relevant instances of reasoning by cases, because the systems are unable to. But this means that, in any situation the system designers failed to anticipate, the systems will fail if reasoning by cases is needed. It seems, though, that the practical utility of systems capable of handling only special cases has created a false impression that expert systems have no need for this kind of logic.

To return to the main issue, I simply do not know what it would mean for a system to use a nonlogical representation of a disjunctive assertion or to use a nonlogical inference technique for reasoning by cases. It seems to me that, to the extent any representation formalism has the logical features discussed above, it is a logic, and that to the extent a reasoning procedure takes account of those features, it reasons deductively. It is conceivable that there might be a way of dealing with these issues that is radically different from current logics, but it would still be some sort of logic and, in any event, at the present time none of the systems that are even superficially different from standard logics have any way of dealing with them at all.

Furthermore, the idea that one can get by with only special-purpose deduction systems doesn´t seem very plausible to me either. No one in the world is an expert at reasoning about a block whose color is unknown between two blocks whose color is known, yet anyone can see the answer to the problem in Section I. Intelligence entails being able to cope with novelty, and sometimes what is novel about a situation is the logical structure of what we know about it.

III WHY DID EARLY EXPERIMENTS FAIL?

The bad reputation that logic has suffered from in AI circles for the past decade or so stems from attempts in the late 1960s to use general-purpose theorem-proving algorithms as universal problem-solvers. The idea was to axiomatize a problem situation in first-order logic and express the problem to be solved as a theorem to be proved from the axioms, usually by applying the resolution method developed by Robinson [1965]. The results of these experiments were disappointing. The difficulty was that, in the general case, the search space generated by the resolution method grows exponentially (or worse) with the number of formulas used to describe a problem, so that problems of even moderate complexity could not be solved in reasonable time. Several domain-independent heuristics were proposed to try to deal with this issue, but they proved too weak to produce satisfactory results.

The lesson that was generally drawn from this experience was that any attempt to use logic or deduction in AI systems would be hopelessly inefficient. But, if the arguments made here are correct, there are certain issues in commonsense reasoning that can be addressed only by using logic and deduction, so we would seem to be at an impasse. A more careful analysis, however, suggests that the failure of the early attempts to do commonsense reasoning and problem-solving by theorem-proving had more specific causes that can be attacked without discarding logic itself.

I believe that the earliest of the MIT criticisms was in fact the correct one, that there is nothing particularly wrong with using logic or deduction per se, but that a system must have some way of knowing which inferences it should make out of the many possible alternatives. A very simple, but nonetheless important, instance of this is deciding whether to use implicative assertions in a forward-chaining or backward-chaining manner. The deductive process can be thought of as a bidirectional search, partly working forward from premises to conclusions, partly working backward from goals to subgoals, and converging somewhere in the middle. Thus, if we have an assertion of the form (P -> Q), we can use it to generate either the assertion Q, given the assertion P, or the goal P, given the goal Q.

Some early theorem-proving systems utilized every implication both ways, leading to highly redundant searches. Further research produced more sophisticated methods that avoid some of these redundancies. Eliminating redundancies, however, creates choices as to which way assertions are to be used. In the systems that attempted to use only domain-independent control heuristics, a uniform strategy had to be imposed. Often the strategy was to use all assertions only in a backward-chaining manner, on the grounds that this would at least guarantee that all the inferences drawn would be relevant to the problem at hand.

The difficulty with this approach is that the question of whether it is more efficient to use an assertion for forward or backward chaining can depend on the specific form of that assertion. Consider, for instance, the schema

```
(EVERY X (IMPLIES (P (F X)) (P X)))
```

Instances of this schema include such things as:

```
(EVERY X (IMPLIES (JEWISH (MOTHER X))
                  (JEWISH X)))

(EVERY X (IMPLIES (LESSP (SUCCESSOR X) Y)
                  (LESSP X Y)))
```

That is, a person is Jewish if his or her mother is Jewish,* and a number X is less than a number Y if the successor of X is less than Y.

Suppose we were to try to use an assertion of this form for backward chaining, as most "uniform" proof procedures would. It would apply to any goal of the form (P X) and produce the subgoal (P (F X)). This expression, however, is also of the form (P X), so the process would be repeated, resulting in an infinite descending chain of subgoals:

```
GOAL: (P X)
GOAL: (P (F X))
GOAL: (P (F (F X)))
GOAL: (P (F (F (F X)))), etc.
```

If, on the other hand, we use the rule for forward chaining, the number of applications is limited by the complexity of the assertion that originally triggers the inference:

```
ASSERT: (P (F (F X)))
ASSERT: (P (F X))
ASSERT: (P X)
```

It turns out, then, that the effcient use of a particular assertion often depends on exactly what that assertion is, as well as on the context of other assertions in which it is embedded.

----

* I am indebted to Richard Waldinger for suggesting this example.

Other examples illustrating this point are given by Kowalski [1979] and Moore [1975], involving not only the forward/backward-chaining distinction, but other control decisions as well.

Since specific control information needs to be associated with particular assertions, the question arises as to how to provide it. The simplest way is to embed it in the assertions themselves. For instance, the forward/backward-chaining distinction can be encoded by having two versions of implication--e.g., (P -> Q) to indicate forward chaining and (Q <- P) to indicate backward chaining. This approach originated in the distinction made in the programming language PLANNER between antecedent and consequent theorems. A more sophisticated approach is to make decisions such as whether to use an assertion in the forward or backward direction themselves questions for the deduction system to reason about using "metalevel" knowledge. The first detailed proposal along these lines seems to have been made by Hayes [1973], while experimental systems have been built by McDermott [1978] and de Kleer et al. [1979], among others.

Another factor that can greatly influence the efficiency of deductive reasoning is the exact way in which a body of knowledge is formalized. That is, logically equivalent formalizations can have radically different behavior when used with standard deduction techniques. For example, we could define ABOVE as the transitive closure of ON in at least three ways:*

```
(EVERY (X Y)
       (IFF (ABOVE X Y)
            (OR (ON X Y)
                (SOME Z (AND (ON X Z)
                             (ABOVE Z Y))))))

(EVERY (X Y)
       (IFF (ABOVE X Y)
            (OR (ON X Y)
                (SOME Z (AND (ON Z Y)
                             (ABOVE X Z))))))

(EVERY (X Y)
       (IFF (ABOVE X Y)
            (OR (ON X Y)
                (SOME Z (AND (ABOVE X Z)
                             (ABOVE Z Y))))))
```

Each of these axioms will produce different behavior in a standard deduction system, no matter how we make such local control decisions as whether to use forward or backward chaining. The first axiom defines ABOVE in terms of ON, in effect, by iterating upward from the lower object, and would therefore be useful for enumerating all the objects that are above a given object. The second axiom iterates downward from the upper object, and could be used for enumerating all the objects that a given object is above. The third axiom, though, is essentially a "middle out" definition, and is hard to control for any specific use.

The early systems for problem-solving by theorem-proving were often inefficient because axioms were chosen for their simplicity and brevity, without regard to their computational properties--a problem that also arises in conventional programming. To take a well-known example, the simplest LISP program for computing the nth Fibonacci number is a doubly recursive procedure that takes $O(2^n)$ steps to execute, while a sligthly more complicated and less intuitively defined singly recursive procedure can compute the same function in $O(n)$ steps.

Kowalski [1974] was perhaps the first to note that choosing among alternatives such as these involves very much the same sort of decisions as are made in conventional programming. In fact, he observed that there are ways to formalize many functions and relations so that the application fo standard deduction methods will have the effect of executing them as efficient computer programs. These observations have led to the development of the field of "logic programming" [Kowalski, 1979] and the creation of new computer languages such as PROLOG [Warren and Pereira, 1977].

IV SUMMARY AND CONCLUSIONS

In this paper, I have tried to argue that there is an important class of problems in knowledge representation and commonsense reasoning, involving incomplete knowledge of a problem situation, that so far have been addressed only by systems based on formal logic and deductive inference, and that, in some sense, probably can be dealt with only by systems based on logic and deduction. I have further argued that, contrary to the conventional wisdom in AI, the experiments of the late 1960s did not show that the use of logic and deduction in AI systems was necessarily inefficient, but only that better control of the deduction process was needed, along with more attention to the computational properties of axioms.

I would certainly not claim that all the problems of deductive inference can be solved simply by following the prescriptions of this paper. Further research will undoubtedly uncover as yet undiagnosed difficulties and, one hopes, their solutions. My objective here is to encourage consideration of these problems, which have been ignored for a decade by most of the artificial-intelligence community, so that at future conferences we may hear about their solution rather than just their existence.

----

* These formalizations are not quite equivalent, as they allow for different possible interpretations of ABOVE if infinitely many objects are involved. They are equivalent, however, if only a finite set of objects is being considered.

ACKNOWLEDGMENTS

I wish to thank Nils Nilsson and Bill Woods for helpful comments on previous versions of this paper.

REFERENCES

de Kleer, J. et al. [1979] "Explicit Control of Reasoning," in Artificial Intelligence: An MIT Perspective, Vol. 1, P. H. Winston and R. H. Brown, eds., pp. 93-116 (The MIT Press, Cambridge, Massachusetts, 1979).

Hayes, P. J. [1973] "Computation and Deduction," Proc. 2nd Symposium on Mathematical Foundations of Computer Science, Czechoslovak Academy of Sciences, pp. 105-116 (September 1973).

Hayes, P. J. [1977] "In Defence of Logic," Proc. Fifth International Joint Conference on Artificial Intelligence, Cambridge, Massachusetts, pp. 559-565 (August, 22-25 1977).

Hewitt, C. et al. [1973] "A Universal Modular ACTOR Formalism for Artificial Intelligence," Advance Papers of the Third International Conference on Artificial Intelligence, Stanford University, Stanford, California, pp. 235-245 (August, 20-23 1973).

Hewitt, C. [1975] "How to Use What You Know," Advance Papers of the Fourth International Joint Conference on Artificial Intelligence, Tbilisi, Georgia, USSR, pp. 189-198 (September, 3-8 1975).

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

Kowalski, R. [1974] "Predicate Logic as a Programming Language," in Information Processing 74, pp. 569-574 (North-Holland Publishing Company, Amsterdam, The Netherlands, 1974).

Kowalski, R. [1975] "A Proof Procedure Using Connection Graphs," Journal of the Association for Computing Machinery, Vol. 22, No. 4, pp. 573-595 (October 1975).

Kowalski, R. [1979] Logic for Problem Solving (Elsevier North Holland, Inc., New York, New York, 1979).

McDermott, D. [1978] "Planning and Acting," Cognitive Science, Vol. 2, No. 2, pp. 71-109 (April-June 1978).

Minsky, M. [1974] "A Framework for Representing Knowledge," MIT Artificial Intelligence Laboratory, AIM-306, Massachusetts Institute of Technology, Cambridge, Massachusetts (June 1974).

Montague, R. [1974] "The Proper Treatment of Quantification in Ordinary English," in Formal Philosophy, Selected Papers of Richard Montague, R. H. Thomason, ed., pp. 188-221 (Yale University Press, New Haven, Connecticut, and London, England, 1974).

Moore, R. C. [1975] Reasoning from Incomplete Knowledge in a Procedural Deduction System MIT Artificial Intelligence Laboratory, AI-TR-437, Massachusetts Institute of Technology, Cambridge, Massachusetts (December 1975). Also published by Garland Publishing, Inc. (New York, New York, 1980).

Newell, A. [1980] "The Knowledge Level," Presidential Address, American Association for Artificial Intelligence, AAAI80, Stanford University, Stanford, California (19 August 1980), printed in AI Magazine, Vol. 2, No. 2, pp. 1-20 (Summer 1981).

Robinson, J. A. [1965] "A Machine-Oriented Logic Based on the Resolution Principle," Journal of the Association for Computing Machinery, Vol. 12, No. 1, pp. 23-41 (January 1965).

Sickel, S. [1976] "A Search Technique for Clause Interconnectivity Graphs," IEEE Transactions on Computers, Vol. C-25, No. 8, pp. 823-835 (August 1976).

Warren, D. H. D. and Pereira, L. M. [1977] "PROLOG--The Language and its Implementation Compared with LISP," in Proceedings of the Symposium on Artificial Intelligence and Programming Languages (ACM); SIGPLAN Notices, Vol. 12, No. 8; and SIGART Newsletter, No. 64, pp. 109-115 (August 1977).

Winograd, T. [1972] Understanding Natural Language (Academic Press, New York, New York, 1972).