REPRESENTING THE REAL WORLD

J.E. Dobson

<u>**Rapporteur**</u>: Mr. M.J. Elphick

REPRESENTING THE REAL WORLD

John Dobson

Computing Laboratory
University of Newcastle
Newcastle upon Tyne NE1 7RU
United Kingdom

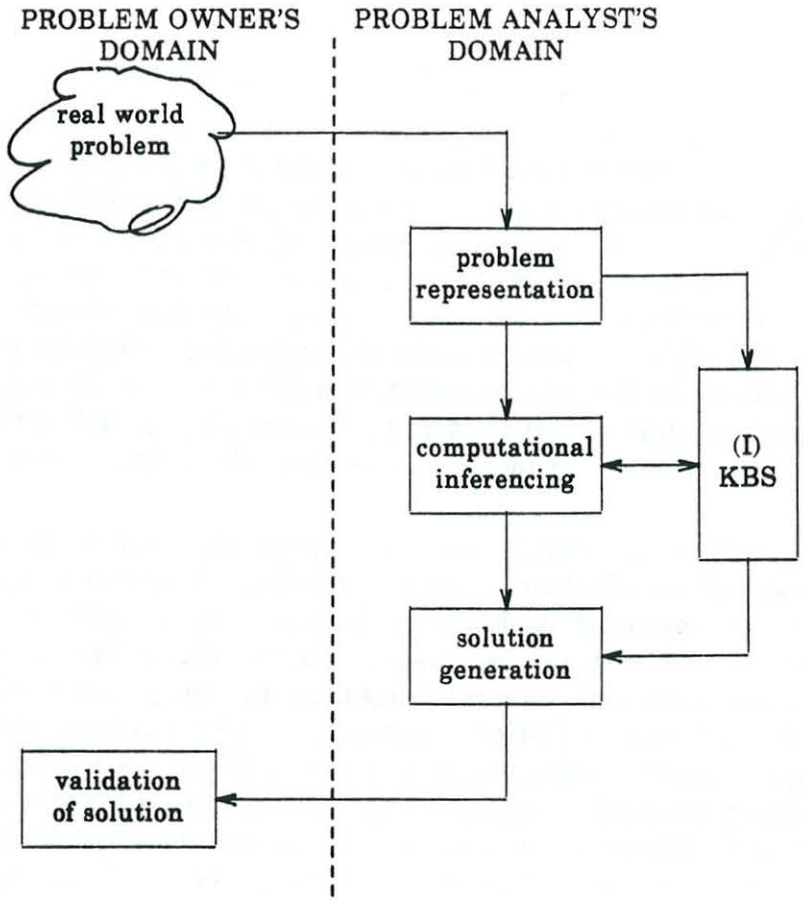
ABSTRACT

In this paper I want to take a look, from a more philosophical point of view, at some of the assumptions underlying the use of logic in knowledge-based systems, and ask - and attempt to answer - the following question: Can a knowledge based system represent the real world? The answer, briefly, is Up to a point; but that point is not very far in any interesting direction.

1. The Role of the Knowledge-Based System

I want to start by taking a simplified (but not over-simplified nor distorted) picture of the role of a knowledge-based system in assisting a problem owner determine the structure of a problem and (possibly) in determining the answer. It is, I think, fair to start by picturing something like the diagram (below).

What the picture is trying to express is that during the problem representation phase, the problem and/or knowledge about the problem domain is represented or encoded in some form of (I)KBS; during the inferencing phase, the problem is explored using the impressive apparatus of logical reasoning; during the solution generation phase, something is shown to 'satisfy' the problem statement which is then given (or sold) to the problem owner as a solution to the problem.



The picture gives rise to three interesting questions:

> What are the philosophical problems in representing a real-world problem?
>
> What are the philosophical problems in real-world inferencing?
>
> What are the philosophical problems in generating and validating real-world solutions?

Here the adjective "real-world" is taken to mean "perceived as relevant by the problem owner".

Note the qualification "*as relevant*". This implies that the relevance has to be demonstrated to the problem owner; and this in turn means that a two-way translation has to take place: from natural language into a formalism, and from a formalism back into natural language. It is the nature of these translation processes that I wish to examine because it is there that the philosophical problems lurk.

Since this paper is part of a seminar on logic in computer science, I want to concentrate on the second of these questions. Recent work here[4] and elsewhere† has examined the first question. (However, it is the third question

† see [3] for an excellent selection

that is, in my opinion, by far the most interesting (and intractable, and ignored). I shall ignore it.)

What I'm particularly interested in then is not how the system represents its data structures or sentential units - its knowledge about the factual structure of the real world - but what it does with its representation. It is impossible, certainly in practice and almost certainly in principle, to be able to formulate explicitly everything that needs to be said about the domain ("slice of reality") or task under consideration. If we were omniscient, we could explicitly represent everything that was or is or is to be, and we wouldn't need to teach logic in computing science, only data structures. But because it is not possible to foresee extra demands that may be placed on the system or changes in the environment, the system must be able to *reason:* to make explicit that which is latent. That's what justifies the use of logic. And it is about the use of logic in reasoning about the real world that I want to talk.

The general problem is that much work in computing science that exploits logical formalisms (or graph-theoretic transcriptions of these) is supposedly directed towards the general problem of 'requirements capture': i.e. the meaning of sentences in natural language. On the other hand, very little work in formal semantics of natural language is being done within the framework of formal quantification languages. The current methods of natural language analysis (Montague semantics[5] and situation semantics[2]) both employ formal languages that are quite different from first order or modal logic. There has never been a serious attempt to use a standard first-order language to provide a semantic analysis of any significant fragment of natural language.

There seem to be very good reasons why this is so. Any account of the semantics of a natural language that exploits a formal language has to make a choice between inventiveness of the *representation in* the formal language and inventiveness of the *translation to* the formal language. (The provision of guidelines for this particular tradeoff is well in the middle of the territory of deep and unsolved problems.) Montague and Barwise/Perry are inventive in the formal language and its semantics, though with (at least in the former case) a severe cost in computational tractability. This inventiveness is, however, confined to an area which at least in principle seems to allow for recursive translation procedures between English and the target formal language, though that language itself does not - in either case - appear to permit much in the way of formal reasoning. The emphasis of these languages is shifted to the precise and unambiguous representation of what is actually the case rather than to the generation of new sentences from old.

By contrast the use of standard formalisms for the expression of (the abstraction of) the natural language problem statement focusses on inventiveness in the procedures for paraphrase from the natural to the formal language. The procedures can be seen exhibited in many an introductory text to logic (programming) for computer scientists - imprecise, nonformalisable, rules of thumb. Here the emphasis is on the generation of new sentences from old rather than on precise representation of the subtleties of real-world situations.

I want now to turn to examine some particular aspects of the differences between natural and formal language that cause particular difficulties for the translation process.

2. Some Arguable Assumptions

Many formal representations of "real world" knowledge presuppose a number of logical assumptions which do not appear to be true of the informal (natural language) expression of the world. It follows that the roles of these assumptions have to be examined, and their introduction has to be made explicit and justified. In this section I want to concentrate on five assumptions related to reasoning which, to put it mildly, are by no means universally accepted except perhaps by inadequately educated computer scientists.

(1) The elements of a model of a language are sets.

(2) Valuations are bivalent

(3) Logic is monotonic

(4) The semantic theory of truth

(5) The (Fregean) principle of compositionality

It is worth emphasising that whereas all of these theses are true for most formal languages, none of them are true for most natural languages. So if the formalism of a knowledge representation language designed to support reasoning assumes them, by what authority can it be said to be a knowledge representation of the real world?

2.1. The Elements of a Model of a Language are Sets

Consider the set of all *meaningful sentences* in English.

Why does this appear a strange thing to consider? After all, it does seem to make some sense to consider the set of all wffs in a (sufficiently simple) calculus. Is it because there can be no complete (generative) specification of the rules of natural language without taking into account the context of utterance of a particular sentence? If so, the collection of all meaningful sentences can never form a set (in the mathematical sense) since the ascription of 'meaningfulness' is undecidable.

Consider instead the following quotation from Wittgenstein ([16] , § 320):

> Why don't I call cookery rules arbitrary, and why am I tempted to call the rules of grammar arbitrary? Because 'cookery' is defined by its end, whereas 'speaking' is not. That is why the use of language is in a certain sense autonomous, as cooking and washing are not. You cook badly if you are guided in your cooking by rules other than the right ones; but if you follow other rules than those of chess you are *playing another game;* and if you follow

grammatical rules other than such-and-such ones, that does not mean you say something wrong, no, you are speaking of something else.

So formal logic is cookery and language is not. In other words, formal logic has rules to generate sets of wffs because it has an end, that of convincing logicians.

It is, of course, an attempt to get away from the rigid mathematical view of sets of wffs that is one of the motivations of situation semantics. But even there we find the same preoccupation with the sentence as the basic unit of grammatical discourse. It is, I think, the tyranny of the sentence that leads us to suppose that natural language discourse is about sets and relations. The main reason why the complexities of discourse and context are not even vaguely suggested by considering the set of all meaningful *sentences* of English is that no very interesting sets exist *in isolation*. The preoccupation of formal logicians of language with sentential fragments about unicorns [Montague] or domestic animals, including children [Barwise/Perry] ignores what we all know - that only in a very superficial sense is the sentence the unit of grammar. As Wittgenstein argues (though I can't find the place where he says this succinctly) natural language is a *process* and 'grammatical' is a *relation* between a sentence and its context, not an attribute of a sentence. Formal language is a *structure* and 'grammatical' is an *attribute* of a wff in isolation. So the two are very different in a fundamental way.

2.2. Valuations are Bivalent

One of the motivations of the practitioners of alternative logic† is to try to get away from another tyranny, that of the bivalent valuation of a sentence. The literature on many-valued logics is staggeringly large, but it seems to me that many of them are, as it were, only slightly adorned variants of classical (bivalent) logic: instead of $V \in \{0,1\}$, we have $V \in [0,1]$ (i.e., the closed interval); and in any case the way in which they are introduced seems often to be parasitic on the notion of bivalence.

Not all logics, of course, fall into this category, only those with which computer scientists are most familiar. Intuitionist logic, for example, is much more radical. The intuitionist would argue that intuitionist logic ought to be employed *instead of* the classical logic whereas the trivalent or modal logician would argue that their logic ought to be employed *in addition to* the classical - i.e. the former regards classical logic as mistaken whereas the latter regards it as incomplete.

Consider the following sentence (one of the meaningful sentences of English, at least to readers of James Joyce):

---

† I am using the term "alternative logicians" in a very broad and general sense here to include all those who think that 'true' and 'false' do not completely cover all that can be said about a sentence (or proposition).

Molly Bloom was an unfaithful wife.

True or False? Aristotle would doubtless argue

A wife who has extra-marital lovers is unfaithful.

Molly Bloom had extra-marital lovers.

Therefore, Molly Bloom was an unfaithful wife.

But the whole point of the book† is summed up in its title. Penelope was faithful. Surely any sensitive reader will realise that despite the major premise as an analytic definition (pace Quine), and the empirical truth of the minor premise, the conclusion is false. It's a question of context and sentence tyranny again. Classical logic cannot account for the falsity of the above consequent; I can conceive of an alternative logic that might take the wider context into account and come to what I (as a reader) would regard as the correct conclusion: namely that the book is what it is partly because of the tension between the simultaneous truth and falsity of the above sentence about its main female character. (Note that this is different from saying that the sentence is neither true nor false, as a (classical) 3-valued logic might assert.)

I do not want here to advocate either classical or alternative logic. What I want to do is to argue that the teaching of computer science has largely ignored the supplementary logics and totally ignored the rival ones. Of course, for many purposes, classical logics are fine; but real-world inferencing is not always of that category.

It has frequently been argued that vagueness, for example, threatens the relevance of classical logic. The structure of such an argument goes (roughly) like this:

(1) Vague sentences may not be bivalent.

(2) They are, however, within the scope of (some) logic.

(3) Any supplementary logic interpretation - e.g. a decision into three classes true, false, neither - can give results as counterintuitive as those consequent on the use of a bivalent logic.

(4) The program advocated by logicians such as Carnap of 'precisifying' ordinary language arguments will not work.

(5) The most economical remaining possibility is to propose an alternative logic which behaves in a manner close to that of ordinary discourse.

All of these steps are of course arguable, and I do not wish to argue them here. What I am arguing is that there *are* alternatives which have intellectual foundation for use when classical logics are seen as inappropriate. Perhaps the most widely discussed such logic in the context of computer science is non-monotonic logic, and to this we shall now turn.

Ulysses. Penelope was the wife of Ulysses.

2.3. Logic is Monotonic

In recent - and from certain points of view, very impressive - work, writers such as Doyle[6], McDermott[13], and Reiter[14] have argued that classical logic is incapable of capturing or adequately representing certain crucial features of real-world reasoning. In particular, they note that knowledge is (almost) always incomplete, and known to be so; so that conclusions which are drawn (e.g. about unfaithful wives) may have to be withdrawn in the light of later evidence (e.g. about literary allusions). An essential point of non-monotonic logic is that the addition of axioms may make previous inferencing invalid even when the additional axioms are (or appear to be) independent of those already assumed. In the words of Doyle and McDermott[7] "Monotonic logics lack the phenomenon of new information leading to the revision of old conclusions" (and by implication non-monotonic logics remedy this lack).

Now my point about logic is this: the non-monotonic logicians are correct in stating that classical (monotonic) logic is - for all the reasons they adduce - incapable of representing the real world; but non-monotonic logic fares not much better, since it is based on the same sort of logical confusion about the role of logic in representing real-world problems.

The basic misconception, which is held by the non-monotonic reasoners as well as by the monotonic ones, is that proof-theoretic logic is appropriate for representing what goes on when we work on the basis of partial knowledge and belief.

There is a distinction to be drawn between "real world" honest-to-goodness rules of inference on the one hand and deductive rules of inference on the other. *Real* rules of inference are rules (or rather policies) of belief revision; *formal* rules of inference are two-place relations between a set of wffs in the first place and a wff in the second. (Remember that the elements of a model of a language are not sets and relations.) To see the difference, consider the following: if you believe that p, and if you believe that p entails q, then you (should) believe that q †. A logician might formalise this as

$$(Bel(p) \wedge Bel(p \Rightarrow q)) \Rightarrow Bel(q)$$

Now suppose you have good (overwhelming) reasons for rejecting belief in q - a policy decision, as it were. All the logician can suggest is that you might revise '$Bel(p)$' and/or '$Bel(p \Rightarrow q)$'. But there is another alternative, and that is to redefine '$\Rightarrow$' so that '$\Rightarrow Bel(q)$' doesn't follow from the antecedent. To do that sensibly, you would have to have a theory of real-world inferencing that fits the evidence, is coherent, simple, generally applicable, and philosophically justifiable. Non-monotonic logic is not such a real-world theory; it is a formal theory (though with a non-standard

† I am using a standard example from epistemic (belief) logic because that is what non-monotonic logicians are concerned with. A very similar example can be found in deontic (obligation) logic: see [9] for a comprehensive and penetrating discussion of the logical and philosophical issues involved.

formalism).

The point of course is that rules of logic are simply rules that permit certain syntactic transformations on sets of signs in a formal language. And although it is quite easy to say (by fiat) that the signs have certain denotations, it is a much harder thing to justify the permitted transformations by saying what the denotations of the transformations are - because they do not necessarily correspond to real world entities. *Nor do they necessarily correspond to our patterns of thought.* Our patterns of thought, be it noted, not what our patterns of thought ought to be if they are to be represented by the logicians' elvish script. The crucial point is that adherence to a set of deductive rules of transformation is neither necessary nor sufficient for acting rationally in the presence of incomplete information. (Again, there is a large literature on the subject; see for example [11] for an account of some of the factors a real-world theory of inferencing would have to account for.)

If, then, neither classical nor non-monotonic logic is capable of representing what we know or believe about the real world, what place do they have in our representations? But what else can we use? Should we not be teaching our students to think about these problems instead of constructing thousand-line formal proofs in some arbitrarily chosen unrealistic theory?

2.4. The Semantic Theory of Truth

Tarski's theory of truth is probably the most widely accepted theory of truth, at least among computer scientists. (How many know that what they are taught about what is meant by the truth of a sentence is but Tarski's definition? How many know that there are alternative definitions?) As is well known, the theory falls into two parts: he provides *adequacy conditions* (i.e. conditions which any acceptable definition of truth ought to fulfil) and a recipe which provides, for a given formal language, a definition of truth which he shows to be, by his own standards, adequate.

The Tarskian theory of truth underlies almost all work in logic in computer science. It isn't hard to see why. His adequacy condition seems eminently plausible, and his recipe can be applied to a large class of formal languages, including (probably) all those that computer scientists will ever meet. The questions I want to raise, though, are (a) Can Tarski's adequacy conditions be given an independent justification? and (b) Have his methods any interesting application to the problem of truth for natural language? I think the answer to both these questions is No; but I'm not going to try to justify that here. The fact that I may disagree with my audience means simply that there is scope for disagreement, and therefore something to be taught.

Tarski proposes, as a material adequacy condition, that any acceptable definition of truth should have as a consequence all instances of the (T) schema:

(T) S is true iff p

where p can be replaced by any sentence of the language for which truth is being defined and S is to be replaced by a name of the sentence which replaces p .

The problem with this adequacy condition is twofold: it can be satisfied by some definitions of truth which seem to most philosophers (and would seem to almost all computer scientists) definitely bizarre; and it is not satisfied by some serious and respectable theories of truth (for example those based on any non-bivalent logic and those based on any non-monotonic logic), which might be held as relevant by thoughtful computer scientists. (See Haack [8] Chapter 7 for an interesting discussion of this problem.)

As far as application to natural language itself is concerned, Tarski himself admits ([15] p. 165; emphasis in original):

> *... the very possibility of a consistent use of the expression 'true sentence' which is in harmony with the laws of logic and the spirit of everyday language seems to be very questionable, and consequently the same doubt attaches to the possibility of constructing a correct definition of this expression.*

Tarski's pessimism has two main sources: his formal correctness condition rules out languages which are neither (i) semantically open† nor (ii) formally specifiable. Natural languages, Tarski argues, fail on both scores, so there is no prospect of an adequate definition of truth for them (in his sense of the word 'adequate').

2.5. The Fregean Principle of Compositionality

The principle of compositionality says that the meaning of a complex expression is some function of the meanings of all its constituent parts. Most formal language users accept this principle without thinking about it. But there are some difficulties with it as applied to natural language, as I hope to show.

If the meaning of a complex expression E is determined on the basis of its several constituent features, the contribution of a constituent part e to the meaning of the larger whole can be identified with the meaning of e - or so it seems. (After all, the meaning of an entity is supposed to be its use in the language.) In other words, we have identified the meaning of of a component e with its contribution to the meaning of some larger complex expression E in which it can occur. But this identification needs to be safeguarded by another principle often associated with the name of Frege, according to which a word or other simple grammatical constituent has meaning only in a context. For if a constituent part e could also have a meaning in isolation, there would not be any general guarantee that its meaning in the context of a general expression E should be identical with its meaning in isolation. Frege attempted to escape from this problem by

† A language is **semantically closed** if it contains, in addition to its expressions, (a) the means of referring to those expressions, and (b) the semantic predictates 'true' and 'false'.

denying that it could arise. But the real difficulty is that the meaning of *e* might vary (even when it is thought of as the contribution of *e* to the meaning of a complex expression) according to some wider context which includes the expression.

It seems certain that this is what happens in natural language. It is seen at its best (or worst) in the case of quantifiers. Consider for example the sentence

Few men accomplish as much as many women.

This has three readings:

A strongly feminist reading ("there are many women such that for each of them you need a small group of men to accomplish as much"), a weakly feminist reading ("there are not many men each of whom accomplishes as much as each of a large number of women"), and a rampant male chauvinist reading ("a small group of men accomplishes as much as a large group of women").

The first two readings can be derived by the two possible linear orderings of the two quantifiers 'few' and 'many'. The third reading results from making the quantifiers independent of each other. Which handling of the quantifiers is chosen depends on the hearer's assumption about the personal prejudices of the speaker; i.e. the rules for handling multiple quantifiers are not clearly defined in natural language. Thus the problem of translating natural language into a quantificational one is formidable, since it is (in general) not possible to state what general principles (e.g. compositionality) the translation can rely on. Worse still, from a technical point of view, is that it is possible to construct intelligible English sentences that cannot be translated into any quantificational language that imposes a linear ordering on its quantifiers: one needs something like the branching quantifiers of Henkin[1] - but by now we are well removed from the sort of formal languages studied in computer science.

Note that I am not claiming that compositionality must fail for computer languages or for the formal languages used in computer science. It merely seems to me that it is a convenience in such languages to make them easier to handle, and not an abstraction from the properties of natural language.

3. What Should be Taught about Logic?

So what has all this to do with the application of logic to computer science and the teaching thereof? There are, it seems to me, three morals to be drawn:

(1) Formal languages are not abstractions of natural language but have quite different properties. Hence the representation step involves *significant* transformation problems which cannot be dealt with by simplistic notions of formal verification.

(2) The kind of inferencing that goes on in a KBS may be quite different from the kind of inferencing that would go on in the real world were the problem to be solved without computer assistance.

(3) Hence the importance of the final step. The reinterpretation back into natural language also involves significant transformations which cannot be verified in any formal sense.

Each of these morals has some implications for the teachers.

3.1. About Representation

A formal description is not an abstraction of reality but of a natural language interpretation of reality. Hence we should be teaching how to interpret as well as how to formally describe, and how to manage the translation from the interpretation to the description.

What is wrong, I think, with our teaching is that we often concentrate on the proof-theoretic issues at the expense of the epistemological issues. For example, no-one in their right minds would suppose that a book entitled, say, The Logic of Scientific Discovery would be a book using the apparatus of formal proof-theoretic logic. One would expect the book to talk in a rational manner about the epistemology of science. Similarly we should be teaching that logic in computer science has some epistemological implications, and showing what they are. We should extend the computer science notion of logic to include the formulation of a set of rules which could be argued - from a philosophical point of view which sees ourselves as parts of the world and computers as an extension of ourselves - to embody rational rules for extending our admittedly imperfect grasp of things.

3.2. About Inferencing

We should be teaching that just as there isn't a single representational formalism that is best, all things considered, for all plausible representational purposes, so there isn't any chance of there being a single reasoning scheme which is best, all things considered, for all plausible reasoning jobs. Hence one has to be taught how to choose. In particular, one has to be taught how to choose between the following statements:

- there is just one correct system of logic for a given problem
- there is more than one correct system of logic for a given problem
- there is no correct system of logic for a given problem. The notion of 'correctness' is inappropriate.

3.3. About Solution Validation

We should be teaching that the sequence of representation, inferencing, and validation as a (somewhat simplified) process model places a lot of weight on validation. In my experience, this is the least well taught phase. Briefly, though, although there is widespread agreement regarding the need to evaluate the product (and process) of systems development, the vehicle for undertaking such an evaluation is far from clear. As with inferencing, the *social* interpretation of evaluation has largely been ignored in the drive to provide a *rigorous* interpretation of evaluation. But that, as I tell my children, is another story.†

It has been well told in [10].

POSTSCRIPT

I can imagine it being argued that I have taken a somewhat philosophical or idiosyncratic view of 'logic', and that really the role of logic in computer science is akin to that of statistics in an experimental science. That is, although there are indeed philosophical issues underlying probability theory, they are of no relevance to practical stochastics and experimental design.

There are two points I could make in response. The first point would be to argue that the use of statistics in an experimental science is purely internal: it is a part of the science not necessarily visible to the outside observer. But this is not appropriate for logic in computer science. If a computer system is to perform some useful role in the real world, whatever logic it uses internally for inferencing must be a reflection of the kind of inferencing processes that are being performed by the rest of the actors surrounding that role and with whom it has to interact.

The second point is that there are serious questions concerning the limitations that should be placed on the use of deductive methods in a real-world context, particularly in areas of social importance (e.g. privacy). In a logical world, inferencing can indeed be captured in a set of syntactic rules, but in a social world, the *logical* conclusion of a chain of argument may be to recommend a course of action that is unacceptable for perfectly good political, economic or judicial reasons. (See Leith[12] for a powerful statement of this principle in the context of his critique of the attempts of Kowalski and others to codify the law in Prolog.) What we should equally well be teaching to students is how to reason about those issues, and to think about how to represent that reasoning.

References

1. J. Barwise, "On Branching Quantifiers in English," *Journal of Philosophical Logic*, vol. 8, pp. 47-80, 1979.
2. J. Barwise and J. Perry, *Situation Semantics,* MIT Press, 1983.
3. R.J. Brachman and H.J. Levesque, *Readings in Knowledge Representation,* Morgan Kaufmann, Los Altos, CA, 1985.
4. J.E. Dobson, "Modelling the Real Issues in Secure Systems Analysis," in *Proceedings of the 1986 CSR Safety and Security Symposium*, ed. T. Anderson, Blackwell Scientific, Oxford, 1987. (in preparation)
5. D.R. Dowty, R.E. Wall, and S. Peters, *Introduction to Montague Semantics,* D. Reidel, Dordrecht, Holland, 1981.
6. J. Doyle, "A Truth Maintenance System," *Artificial Intelligence,* vol. 12, pp. 231-272, 1979.
7. J. Doyle and D. McDermott, "Non-Monotonic Logic I," *Artificial Intelligence,* vol. 13, pp. 41-72, 1980.
8. S. Haack, *Philosophy of Logic,* Cambridge University Press, 1978.
9. J. Hintikka, "Deontic Logic and its Philosophical Morals," in *Models for Modalities*, pp. 184-214, D. Reidel, Dordrecht, Holland, 1969.
10. R. Hirschheim and S. Smithson, "Information Systems Evaluation: Myth and Reality," in *Information Analysis: Selected Readings*, ed. R. Galliers, Addison-Wesley, Reading, Mass., 1987.
11. D. Kahneman, P. Slovic, and A. Tversky, *Judgement under Uncertainty: Heuristics and Biases,* Cambridge University Press, 1982.
12. P. Leith, "Fundamental Errors in Legal Logic Programming," *Computer Journal*, vol. 29, no. 6, pp. 545-552, December 1986.
13. D. McDermott, "Non-Monotonic Logic II," *JACM*, vol. 29, no. 1, pp. 33-57, 1982.
14. R. Reiter, "A Logic for Default Reasoning," *Artificial Intelligence*, vol. 13, pp. 81-132, 1980.
15. A. Tarski, "The Concept of Truth in Formalised Languages," in *Logic, Semantics, Metamathematics*, ed. J.H. Woodger, pp. 152-278, Oxford University Press, 1956.
16. L. Wittgenstein (ed. G.E.M. Anscombe and G.H. von Wright), *Zettel,* Blackwell, Oxford, 1967.

DISCUSSION

Professor Backhouse found it significant that John Dobson's first slogan had concerned meta-logic. He would argue that most logic textbooks contained too much meta-logic; as computer scientists, we wanted to use logic. He asked whether the speaker was suggesting that teaching in logic should start with meta-logic, and if so, what relative importance should be attached to say, a study of meta-logic and the elaboration of a large proof? **Mr. Dobson** replied that he was not suggesting that one should teach meta-logic as a formal discipline; one should use it to get across what is important and what is incidental. Thus one might start off by using very simple models, then point out some of the simplifying assumptions and explore their consequences. One should in teaching expose such assumptions, ask if alternative accounts can be formulated, and in general introduce the concepts of meta-logic as one goes along.

Asked by **Professor McConalogue** to "define in one sentence the distinction between semantics and meaning", the speaker offered the view that meaning was a (non-symmetric) relation between the real world and signs, while semantics was a relation between signs and the real world. Whether one was the inverse of the other could be argued about at great length!

Professor Pnueli responded to the speaker's criticism of Computer Science teaching by saying that he was usually happy to teach students to work in a particular formal system, but felt that expecting them to be able to analyse the inadequacy of such a system, and if necessary design a new logic, was asking too much. **Mr. Dobson** replied that he was not in fact advocating this much, but only that graduates should be able to recognise when a standard system was inadequate, and when it was necessary to obtain advice (say, from a practising logician). They ought to be be taught the basic concepts of modal logic, and the ability to recognise that, for example, the phrase "It is secret that..." suggests the need for a modal logic. **Professor Pnueli** suggested that some help from logicians would be needed, perhaps in condensing the material in the texts taking a more philosophical approach. Computer Science students are pragmatists, and need to be shown the connection with reality as they see it!