Logicism and Meaning: The Case Against (Draft)

Iain D. Craig
Department of Computer Science
University of Warwick
Coventry CV4 7AL
UK EC

January 24, 1995

Abstract

This paper argues, contrary to the claims of other workers, that formal semantics in the sense of model theory cannot provide an adequate basis for the ascription of meanings in AI programs.

Note: The original version of this paper was written in March and April, 1991. It therefore pre-dates *Meanings and Messages* and *Programs that Model Themselves.*

1 Introduction

AI programs are obviously *about* things, more explicitly so than conventional ones. A payroll program is certainly *about* salaries, employees and tax law, but the representations it employs, together with the ways in which those representations are manipulated, are more *obviously* encoded in the program code than are the representations of most AI programs—one has to say 'most' because there are AI programs that represent their knowledge in terms of procedures in some programming language (often LISP or as Prolog clauses). The observable behaviour of a payroll program also suggests that it is different in kind from an AI program: payroll suites tend to be run overnight in batch mode, and their output is, typically, a form-like listing. We are used to seeing *interactive* AI programs that enter into dialogues with their users. There is nothing, of course, to prevent one building an 'expert' payroll program with representations that are more explicit than those currently found, nor is there any real reason why payroll programs could not be interactive.

The observable behaviour of a payroll program does *not* correspond to an AI program: it can be objected that the fact that payroll suites often run in batch is an irrelevance—this is correct. It can also be objected that one of the criteria that I have just presented—*explicit* representation—is rather operational. Explicit representation, according to the logicist view of AI (e.g., [32]), brings with it the benefit that represented items can be used more flexibly—a hint at meaning-as-use (which has turned out to be notoriously difficult for logic). One should be able to determine whether one has an AI program on the grounds of the behaviour it exhibits: one should not have to look at the code to say what one has. The explicitness of representation is often taken by AI workers to be part of the claim that AI programs represent things in a modular fashion (Waterman and Hayes-Roth's 1978 collection [43] is full of statements about the 'modular representation of knowledge'): modularity in representation entails, it is claimed, greater ease in extension or revision[1]. Another claim that is made for explicit representation is the one that if something is not represented, it cannot be reasoned about. Quite obviously, this last point is, in some sense, correct, especially if one wants to engage in *formal* reasoning about something. It can be argued that what makes the AI program distinct from the payroll suite is that the information that is represented (encoded) in the AI program can be used more flexibly than can the information in the payroll program.

The comparison with payroll programs is quite interesting because it compares two different approaches to programs (and that is what AI systems are) which are *about* something. We would probably agree that a payroll suite does not have 'representations' of the employees whose records it processes; we would also probably agree that a payroll program does not have 'knowledge' of tax and social security

[1] One wonders whether UK local authorities would have been more able to cope with the Poll Tax and its changes if they had used AI systems—perhaps not

law, nor that it held 'beliefs' about the employees who depend upon it for correct payment. One reason why we would make these negative decisions about a payroll suite (and correspondingly positive ones about an AI program) is related to the level at which information is represented. In a payroll program, an employee is represented as a record—the employee's name is typically represented as a string. In an *expert* payroll program, an employee would be represented by a frame or by a set of assertions; what makes these representations different is that more can be done with them than can be done with a character string. In a frame system, information can be inherited and defaults over-ridden; in a logic-base system, deductions can be performed.

The most important facts about an *expert* payroll[2] program are that:

1. It performs *inferences*.
2. It can (be said to) hold beliefs or to have knowledge—the beliefs which it holds or asserts can be (taken to be) *true*.

In other words, that an *expert* payroll program can do things with symbolic representations implies that its representations have *meanings*[3]. This claim of semantics is implicit in much of AI. It does not matter that the individual symbols that are used in the representation of, say, an employee are not given an interpretetation: what matters is that such an interpretation *could* be given, and that the interpretation would be about the world external to the program. The symbols that comprise the representation are manipulated by inference rules in order to derive new symbols, and this process *respects the semantics*. In a logic-based system, the concept of respecting the semantics explicitly shows up as the general logical principle that a *sound* inference licences one to draw a true conclusion from true premises: a false conclusion can never be inferred from true assumptions unless the inference rule that is used is *unsound*. In AI programs that are not explicitly logic-based, a similar principle applies, although it might be described as drawing 'reasonable' or 'rational' conclusions from assumptions or data; nevertheless, what is still desired is a conclusion that is at best true or, if that cannot be achieved, one that is 'possible' (in the logical sense) or that is 'adequate' (in some *pragmatic* sense—see [11] for an example of an AI program that is required only to provide adequate answers).

2 Semantics and Models

The issue of semantics is central to the AI enterprise. Most theoretical accounts of knowledge representation (perhaps, one ought to say 'knowledge representation language proposals' because there has been little work in AI, or elsewhere, on the

[2] The emphasis on 'expert' is not ironic—or is it? See below.

[3] It is not at all surprising that one of the earlier collections of papers on AI research was the influential *Semantic Information Processing* [27].

concept of *representation* itself, although Hayes [14] and Sloman [37] have made illuminating comments) deal, ultimately, with truth. As in logic, the foundational idea is to determine when a sentence of a formal language is true: that is, in the present context, when a sentence or expression of the knowledge representation language is true. As with formal logic, the idea is reduced (following Tarski) to that of determining the conditions under which a sentence is *satisfied* by a model (the 'intended' or 'standard' model). Whether a traditional Tarskian account, or a Kripke-style possible worlds account is given, what is sought is a relation between the representation language and the models which make its sentences true.

The reader should not think that I have any objections to either account of semantics[4]: indeed, model theory is not used *enough* in AI. As will become clear below, my objection is to the conception of a logical semantics in connection with representation. Certainly, what is needed is *something* that licences the move from givens to conclusions. What is also needed is a way of determining the *content* of a representation.

In mathematics, the concept of a 'model' is something akin to what logicians call a theory: it is a collection of structures and axioms, together with inference rules which licence the application of axioms in the derivation of new sentences. This view corresponds closely with some views of 'semantics' in formal logic. Indeed, it can be argued that the concept of a 'model' in logic is very similar: one version of model theory (the one usually ascribed to Tarski) says that a model is a set-theoretic structure which provides an interpretation of the sentences of the formal theory whose semantics are being given; another account (due, basically, to A. Robinson) is that a model is a mathematical structure of some kind (usually set theory, although it might be, for example, a topos) and that there is a mapping from the formal language to the other mathematical structure. Under either interpretation, what is going on is that the sentences of the theory expressed in the formal language whose semantics are to be given are translated into sentences in some other formal structure: once the translation has been achieved, reasoning can proceed in the other structure using its objects, relations, functions, axioms and inference rules (here, I include theorems under the general heading of 'inference rules'). A sentence of the formal language is true if and only if the sentence in the model (the translation, that is) is also true—the latter is the case if the sentence is true with respect to the axioms of the model. The two versions are equivalent, although the second seems to allow greater range in the structures that can be chosen to serve as models. It is an important observation that models *represent* in various ways (one way is that they represent the sentences of the formal language).

The appeal of 'semantics' in the sense of the last fews paragraphs to knowledge representation should be made clear. A sentence (expression) of a knowledge representation language is useful only when it represents something that is *true* (the

[4] The quotation marks are intended to be signs that the term is being mentioned and not used—there is no irony, and there should be no criticism of these concepts imputed to the author.

hedge 'something' is important here because it is not, in general, possible to say that one is dealing with a 'fact', a 'proposition', a 'state-of-affairs', or, maybe, 'situation'). What is important is to give an account of the semantics of the knowledge representation language in such a way that its sentences can be used assertorically: that is, to make true assertions. Truth, as has been noted, is important for the concept of sound inference. If the representation is to be construed as dealing with the beliefs of the artificial agent, there is a double task for the semantics. For belief systems, two properties are important:

1. The semantics must give the conditions under which the agent may truthfully assert a belief (i.e., the conditions under which $\vdash_{\mathcal{A}} \mathbf{B}\phi$ can be veridically stated).

2. The *content* of ϕ in $\vdash_{\mathcal{A}} \mathbf{B}\phi$.

The content of a proposition, ϕ, is usually taken to be its interpretation in all possible worlds (or, at least, all those accessible from the reference world): this involves determining the referent of ϕ and also its truth-value. This interpretation is justified by the argument that if one believes that ϕ, one will assert it (as being true); in symbols:

$$\mathbf{B}\phi \rightarrow \phi$$

which is often taken to be an axiom[5]. (Note that the material equivalence derived from the converse, namely, $\mathbf{B}\phi \leftrightarrow \phi$ is not taken as an axiom—it is not even true: just because something happens to be the case, one need not *believe* that it is the case.) In other words, in order to determine whether the belief that ϕ is a true belief, one needs to know about ϕ[6]. If the theory does not deal with beliefs (or knowledge), it is still important to know about the content of ϕ. In both cases, one needs to know about the *intension* of ϕ.

Both Tarski-style satisfaction and Kripke-style possible world semantics have been proposed by workers in knowledge representation. Hayes is, perhaps, the most notable proponent of using Tarskian semantics. Moore [29] and Levesque [22] have argued for possible world semantics for doxastic and epistemic logic. More recently, Levesque [22, 23] has used a *situation* semantics: for present purposes, situation semantics can be considered to be a variation on possible worlds in which each situation is a 'small' possible world (see [2])[7]. Below, the reasons why semantics of these kinds for knowledge representation languages have been proposed will be considered: the discussion will focus on the central issues of my argument.

[5] It is frequently referred to as T: it states that the accessibility relation between possible worlds is reflexive.

[6] The reader is given the almost traditional health-warning about true belief and knowledge: true belief is not, ordinarily, to be construed as knowledge. The latter can be analysed as *justified* true belief—justification is important.

[7] Even allowing the novelty of situation semantics, Levesque's work still falls foul of the arguments to be presented below.

3 Semantics and Knowledge Representation

There has been much activity in the semantics of knowledge representation since Woods [47] argued that most representation languages were confused formalisms that lacked a coherent semantics. Hayes [15, 16, 17] has long argued that first-order logic (FOL) should serve as the basis for the representation of knowledge.

The aim of this section is to determine why FOL is so often cited as *the* knowledge representation par excellence. This discussion will lay the ground for the argument that is to follow on *what* is necessary for a representation language to be adequate.

To be fair to Hayes, in [17], he argues that FOL should serve a role similar to that of a specification language in software engineering: what he advises is that domain knowledge be encoded in FOL because:

1. FOL is domain-independent (hence, it does not bias one in any way).

2. FOL has well-understood inference rules (modus ponens and generalisation in an axiomatic presentation).

3. FOL has a semantics (the one supplied by Tarski).

Hayes' idea is that FOL can serve as a medium for communication between workers on the naive physics (and, presumably, other) projects, and he does not suggest that FOL be the *implementation* language: he says quite explicitly that the theories discovered or constructed during the naive physics enterprise can be encoded in *any* representation language whatsoever[8].

Despite the fact that what Hayes actually advocates in [17] is somewhat milder than what many take him to be recommending, in this section, I will concentrate on the case that he presents, *as well as* the stronger position that he is often supposed to take. The stronger position (i.e., the one that FOL *must* be used as *the* representation language) has been adopted by others, for example Moore [28], Genesereth and Nilsson [13], and Nilsson [32], and which was adopted by McDermott for a number of years (e.g., [24, 25], and [26]). Of the various arguments in favour of FOL, I consider those of Hayes to be the clearest, the best-articulated and, possibly, the most easily available: the critique that follows is, if anything, intended as a compliment to him.

With these preliminaries out of the way, a consideration of the reasons for adopting FOL (with or without intensional connectives or operators) as the paradigm representation system for AI can begin.

[8]He does not say that such an encoding must be supported by a meaning-preserving mapping between FOL and the target representation language. A semantics is *still* needed for the target language, and that, by Hayes' arguments, must be a formal semantics. By the remainder of Hayes' arguments, what would be needed would be a semantics of the representation language in terms of its properties as a syntactic variation on FOL, so one would be translating FOL and its semantics into FOL and its semantics! Perhaps I am being a little too harsh in this.

The discussion can start with an argument which, although I have never seen it in print, appears to have some bearing on logic and representation. I will call this argument the *symbol argument*[9]. In essence, the symbol argument runs as follows. AI programs perform tasks that require 'intelligence', and part of being intelligent is the ability to perform reasoning tasks. At bottom, AI programs are about the transformation of input symbols into output symbols (the physical symbol hypothesis [31]). That is, the reasoning that an AI program performs is represented as transformations of symbols. The symbols are arranged according to syntactic rules, so it is possible to talk about a 'well-formed' expression or sentence or formula (what one *actually* calls it is of no importance—'formula' seems as good as anything else). The transformations can be cast in the form of rules: they have a mild syntactical context-dependency. (Those who find this remark about dependency perplexing should consider the transition rules in finite-state automata or the rule of modus ponens in logic. Modus ponens can be stated as:

$$\frac{\vdash \phi, \vdash \phi \to \psi}{\vdash \psi}$$

Quite clearly, anything will do for ϕ, but not everything will do for $\phi \to \psi$—it *must* be an implication. It is context-dependency in this sense that I intend.) Thus, an AI program is constructed from symbols and performs reasoning tasks.

Logic is the paradigm of reasoning: its roots lie in the attempts by the ancient Greeks to codify human reasoning and thought. Because of this, and because of the more recent work by philosophers on reasoning, belief and on the semantics of natural language, logic appears to be a good candidate as a representation language. Logic also has some useful constructs, such as quantification and negation: it deals with truth, so it deals with exactly the kind of material that is needed to support the symbol manipulations of an AI program. In addition, a good deal of logic can be done by just applying the inference rules: it can be treated as a symbol-manipulation game (as can a lot of mathematics). The analogy between the symbol manipulations of logic and those of AI programs is striking, though not surprising, given the work of Turing and of Church (the work of these two also shows that there is a strong connection between logic and computation). Because of proof theory, it is possible to get a long way merely by manipulating the symbols according to the inference rules (this is used to ward off initial questions about where semantics enters the matter); even better, there is the completeness theorem which states that:

$$\vdash \phi \leftrightarrow \models \phi$$

This *shows* that whatever can be done in semantics can also be done in syntax (just using the inference rules): in fact, because of the completeness theorem, semantics

[9]Whether anyone would *openly* express this view is not in question: the point of stating the argument is that it highlights some essential aspects of the representational task. The fact that some might object to this argument as a satire or as cynicism is irrelevant.

can be thought of as a 'nicety'. When pressed for a semantics, a 'theorist' of this kind can wave his (or her) hands and point to model theory: *that* is the semantics of logic—there is no need to do anything about semantics because Tarski et al. have done it all for us.

I very much doubt that anyone would seriously put this argument forward, even though it might seem *plausible*. I would very much doubt that anyone who was serious in proposing logic as a representation language, or even as a meta-language, would merely allude to Tarski and claim that all the necessary work on semantics has been done—witness the number of papers that propose a model for some new and exotic logic. As I warned the reader above, the argument has been stated because it makes a number of points that will be useful below, the most important of which are that:

1. An account of meanings *must* be given (hence the insistant semanticist).

2. Symbols are used to perform the representational task.

3. Inference is a species of transformation

4. The processes must be effective (have to be performed by a machine, so, by Church's thesis, *we* have to obey this constraint if we are to accept the information-processing account of mind).

Hayes' argument in [17] is rather more apposite. The first makes the (not unreasonable) assertion that most, if not all, representation languages[10] that have been proposed to date are variants of FOL. In favour of this is his own work on translating frames into FOL [15], although he remarks (p. NN) that hierarchical representation similar to frames and semantic networks may require default logics [34, 3] (an opposing view on frames, which treats frames as *terminological* definitions, is to be found in the work of Brachmann, Levesque et al. on KRYPTON [4, 5], [33]). If this assertion is correct (and I believe that, essentially, it is), it does not mean that we should stop trying to develop new representational formalisms and write everything down in FOL, but that FOL should be used as the gold standard against which we measure our proposals.

The main thrust of Hayes' argument is that FOL has a *semantics*. The semantics is model theory. The semantics is supported by a view on truth (which is, ultimately, a correspondence-theoretic account). Unless a semantics is given for a representation language, there is no sense in which we can ascribe 'meanings' to the formulae that do the job of representing. What this amounts to is that we can do all the inference we want (i.e., apply all the inference rules in any combination), but it only amounts to symbol transformation *until* we have given an account of what the symbols are intended to stand for. Now, it is simply not enough to say 'symbol `fido1`[11] stands

[10] Remember that connectionist representations are *not* at issue.

[11] Typewriter face—courrier font—will be used for anything that is assumed to be part of a program.

for my dog' and 'symbol `fido2` stands for your dog'—what if I am addressing my wife, and we only have one dog? It seems natural to assume that `fido1` and `fido2` co-refer: what is there to back that claim up? What happens if the person who utters the first sentence does not have a dog (neither my wife nor I own a dog, so neither `fido1` nor `fido2` refers to anything). As it stands, nothing. Another problem comes with substitutions. Clearly, not all substitutions are meaningful: for example, and taking FOL syntax, what if we have the predication $\phi_1(\overline{a}_1)$, where ϕ_1 is a predicate symbol and $\overline{a}_1$ is a constant, with the interpretation that ϕ_1 is 'has four legs', and $\overline{a}_1$ is interpreted as 'red'? This is semantically anomalous, but there are no restriction criteria on the syntax of FOL (or of most representation languages I know of, many-sorted logics being one exception) which will forbid this kind of sortal error (for that is what it is). The claim is that the explicit provision of models for the *theories* that we develop using our representation languages will allow us to *infer* co-reference or sortal and detect category mistakes. In order to provide a semantics for the theories to be expressed in a representation language, the language itself must have a semantic framework.

By taking the assertion that most representation language proposals have been syntactic variations on FOL together with the fact that FOL has such a semantic framework, one naturally (deductively?) arrives at the conclusion that FOL is *the* representation language.

Hayes comments that, for any theory expressed in FOL, there are many non-equivalent models, but he sees this as a virtue: it becomes possible to *choose* a model. If one needs a *unique* model, what one does is to add (non-logical) axioms to the theory. The more axioms there are in a theory, the more constrained the interpretation of those axioms because axioms place constraints on the interpretations of the non-logical symbols (the relation, function and constants of the theory, that is). As the number of axioms increases, the number of models decreases: eventually, one reaches a stage at which there is a unique model—this becomes the 'standard' model or preferred interpretation. Thus, it is possible, using the model theory of FOL to reach a position in which one can point to a model and state "this is the *unique* model of the theory".

Unfortunately, I doubt that it is possible to restrict the interpretation of a theory by adding new axioms: the Löwenheim-Skolem theorem seems to defeat this. The reason is that, by the Löwenheim-Skolem theorem, any finite axiomatisation will not necessarily have a finite number of models. As the number of axioms increases, the number of models will also increase. Even when equivalent models (in the sense that two models are equivalent if and only if they assign the same truth-value to a sentence) are removed, I suspect that the number of resulting models will be greater than one. A sure-fire way to obtain a unique model is to use the Herbrand interpretation of the theory (Hayes describes the Herbrand interpretation as 'ghostly' in [17]): for any theory, there is always a *unique* Herbrand model. However, as Hayes also observes when he makes his remark about them, Herbrand interpretations do not really help. The reason for this is that the Herbrand interpretation of a theory is constructed from all and only the function and constant symbols that appear in that theory: that is, a Herbrand interpretation is *only* composed of the function and constant symbols. Thus, by building a Herbrand model, nothing has been gained because the symbols in the theory are interpreted by terms composed of those very symbols. Hayes then goes on to say that there is a way of obtaining meanings by means reflection into the meta-language. If this is a claim that meanings can be salvaged from a Herbrand model by recourse to reflection, it is false, for one can construct a Herbrand interpretation for the meta-language and reflect the Herbrand model of the object-language into the Herbrand model of the meta-language, thereby gaining absolutely nothing; even if this is not what Hayes intends, there are problems with his account of meaning. Furthermore, the reason why reflection into a formal meta language is of no help is again because of models: the theory cannot be pinned down to just one model.

The inability to produce a unique model of a theory is one problem. There are others which Hayes mentions and solves in summary fashion: these relate to the connection between the representation language and the external objects which it is supposed to represent.

The point of representing, say, the commonsense world is that a program can reason about it. As has been seen, a semantics has to give an account of *content*. In other words, it has to say what the representations are *about*. Now, Hayes sees this quite clearly. His argument is that the models of a formal theory give a semantics: that semantics can be in terms of abstract objects (sets, numbers, etc.), or it can be in terms of the external reality the representation is intended to capture. Here is where the problem resides. Hayes insists that a semantics *must* be given, and he is exactly correct about this. However, as will be seen below, the problem comes when one wants to use the model-theoretic semantics as an account of 'meaning'—in other words, as an account of content. Hayes in [17] remarks that the 'real world' can be used as a model of a formal system (in other words, he will admit models which do not consists of formal structures), as well as models of the kind more usually encountered in model theory. The problem comes in giving the grounds on which a model, a formal system and the external reality can be connected in such a way as to confer content.

The all too brief account which Hayes gives is that *we* confer upon a formal representation the content which *we* want it to have. In other words, *we* determine what the representations are about: there is no need for the representation to incorporate 'connection' conditions. A point which needs to be raised is that model theory deals only with concepts such as completeness, validity and so on: it does *not* deal with 'meaning' in its full sense, nor does it actually deal with truth. What a model of a formally presented theory amounts to is a guarantee that certain, purely formal, conditions are met: for example, that the theorems are valid. What the presentation of a model-theoretic account of a formal theory gives is an assurance that the theory whose semantics is being presented has certain properties which are desirable if one wants a deductive (or, perhaps, even one day an inductive or abductive) theory. A model does not deal with truth, nor, really, with interpretation, because it, actually, only involves the translation of a theory expressed in one formal language into the structures of another formal language[12]: the concept that is employed in model theory is not truth, but the weaker one of *satisfaction*. The fact that model theory involves a translation is not of any importance, and there is rather more to the status of model theory than translation would seem to imply. The term *interpretation* is also, quite frequently, mis-used: the interpretation is, more properly, the *mapping* between the formal theory and the model. The essential point is that truth is externally imposed upon a model by us: the reasoning is something along the lines of 'if ϕ is translated into a sentence ψ of the model, and if ψ is true, then so is ϕ' (where ϕ is any formula of the formal language, and ψ is a formula in the model)—this depends upon our intuitions about and knowledge of the structures in the model. In other words, a model does not *confer* truth, it only gives equivalence under some mapping. That is to say that truth is only *relative* to some model.

The above is not to be construed as an attack on model-theoretic semantics: it is intended only for clarification. I agree that some kind of semantics must be provided for a formally presented theory, and that model theory is one form of semantics (perhaps, when talking of model-theory in this context, the word 'semantics' ought to be quoted, for its use in logic does not correspond to its use in other contexts—i.e., the 'semantics' in "model-theoretic semantics" is a technical use) which is of considerable use: it is very nice indeed to have consistency proofs, completeness proofs are yet better. My point is that a semantics of another kind, one which goes beyond the abstract entities of model theory (and of the formal theory itself) is required: this point is acknowledged by Hayes, of course.

By the argument about unique models, it is we, again, who determine the 'standard' or 'intended' model. By determining the standard model, we are not going about the task of connecting the formal theory with the outside world. Any symbol (and that is what, at base, a formal theory consists of) can be interpreted in a myriad of different ways: pick up two books on logic and compare the various interpretations of the proposition symbol p, or take, as another example, the usual statement that p is a propositional variable and, so, can range over *all* (expressible)

[12] Strictly speaking, the relation is many-many and not, as one might think one-one. However, in the translation of ordinary languages, one can still have a many-many relation which one calls 'translation'. For example, *il pleut*, *piove* and *es regnet* all literally translate into English as *it is raining*; they can also be translated as *it does rain*, *it rains*, or *it will rain*—some languages use what looks like a present tense to stand for a future. These examples show that translation in the ordinary sense can be many-many. The content of the French, Italian and German sentences is mapped onto the same content in the English rendering, but the point I want to get across is simply that the many-many relation between theory and model does not disqualify one from calling it a 'translation' if one understands by it the concept that we use in everyday speech. Hayes says that the relation between a formal theory and its models is many-many *unlike* a translation.

propositions. By the Löwenheim-Skolem theorem, there are many (sometimes infinitely many) non-equivalent models, so the non-logical symbols of a theory have many non-equivalent interpretations[13]. Without some way of fixing an interpretation, a symbol is just a mark on a page (or an address in memory): what one does with the symbol depends upon the symbol-manipulation game one is playing at the time. To make mathematics work, we supply the meanings; for AI programs, this luxury may not always be available. In the next section, I will investigate this aspect of semantics in more detail.

4 Inference, Reference and Descriptions

In this section, I want to examine the ways in which representations can be connected to what they represent: in other words, I will be interested in a form of symbol grounding. My remarks are aimed at AI *programs* and *machines*, not at people: if it happens that the arguments transfer to the case of people, I will be most pleased; if not, too bad.

The problem to be dealt with can be summarised as follows. Knowledge epresentation languages are symbolic systems that are typically composed of a set of sentences or formulae. In order to get the representation to do anything, inference rules are applied by a processor called the *inference engine*. The job of the inference rules is sometimes stated as drawing out what is *implicit* in the (declarative) representation. For example, if the declarative database contains the formulae (which are expressed in predicate calculus for simplicity):

```
sister(sue,anne)
mother(anne,sarah)
```

the inference rule:

```
if sister(X,Y) and mother(Y,Z) then aunt(X,Z)
```

can be applied to yield the fact that `sue` is `sarah`'s aunt: this information is not explicit in the declarative database, it has to be inferred. The extent to which inference really *is* making implicit information explicit (also whether problem solving is of this kind) is moot, but I prefer not to go into the arguments here.

Inference rules are, like the declarative component, symbolic structures. The inference engine. I will use the term *processor*, following Smith [38, 39], to denote the inference engine. The reason for this is that it is all too easy to refer to the inference engine as the “interpreter”: I feel that this, last, term is too reminiscent of the interpretation mapping in logic, and that possible confusions can result from this.

[13] Wittgenstein remarks in a number of places, for example [46], that symbols do not have a *unique* interpretation, meaning or use—symbols are *arbitrary*.

There are two main points I want to make about the processor. The first deals with truth and the constraints that processors are usually believed to satisfy. The second concerns action. Both of these points will be of use below.

The processor is charged with applying the rules in such a way that solutions are found to problems, implicit information made explicit in a way similar to that shown above. A not unreasonable constraint which the processor must satisfy is that rules be applied in a way that preserves truth (i.e., the application of rules is at least *sound*). In the above example, if the program is said to believe that Sue is Anne's sister, then any conclusions drawn on the basis of this belief should not falsify that belief (this still applies even if the processor uses a refutation strategy to establish truth—it would *assume* that Sue is not Anne's sister and aim to draw a contradiction). Other, more practical, constraints may apply: for example, the processor must give up attempts to find answers after a given time has elapsed or after a given amount of store (measured somehow, in some units—say, cons cells) have been used. The actual constraints that are applied to the processor do not matter for present purposes: that there are constraints in addition to the soundness one (which seems, in any case, to be basic, or at least, the one that is implicit in the descriptions of just about every processor of which I am aware). Of course, we always *want* truth to result from inference: we might accept an undefined value in some cases, though.

The other important point I want to make about the processor is that, without it, there would be no interpretation of symbols by the machine[14]. The latter, I take, by an large, to be software, whereas the former is taken to be either software or hardware), and there would be no way of solving problems (satisfying goals, in other parlance). *In order to* solve a problem, it is necessary to apply at least one inference rule: the processor must *act* in a certain way in order to exhibit the behaviour that we call "solving a problem". Without these actions, there is nothing: no behaviour—all there is is a collection of structures in memory which, of themselves, do nothing. I want to assume that *all* the behaviours of the processor are at least in principle observable by some external observer: for programs, this can often be arranged (it is harder for parallel programs, but that does not alter matters for present purposes). What I want to argue is that there is no way of interpreting the declarative representation *without* performing acts of various kinds: this turns out to be an important point, and I will return to it below when discussing the proposition that processors be considered as *causal* entities—causally efficacious and causally embodied, that is.

Above, I described as symbolic structures the declarative database and the inference rules that, together, are often taken as comprising the "knowledge base" of an AI program. What I really should have said is that they are composed of *uninterpreted* symbols. For the vast majority of AI programs, it makes no difference to the way in which the program behaves if one changes all the symbols it contains

[14] I will use the term 'machine' in a more global sense than I do 'processor'.

in its knowledge base: in the limit, one can uniformly substitute `gensym`d symbols for the symbols in the knowledge base. (As will become clear, the unrestricted use of `gensym`d symbols can lead to horrific problems, particularly when there is no evidence for the exisence of the 'object' named by the new symbol.) When run, the program's behaviour will be identical to its behavior before the substitution, the only difference being that different symbols will be output (uniform substitution seems warranted because any other regime could lead to non-equivalent behaviour). The difference in output might make the program harder to understand, but one can imagine a translation table that one reads in order to convert the output to a form that means something to the user.

The last point is important: the symbols that are actually used in a program are chosen because they mean things to the user or builder—for example, above the symbol `aunt` was used to denote a relation between two individuals, one of whom has to be female; it would have been possible to use the symbol `g101`, but that would have been more opaque, less understandable to the reader. In other words, it is *we* who choose the symbols, and it is *we* who give them a meaning; as far as the program is concerned, the choice of symbols is *irrelevant*. If the choice of symbol were important, the program would behave in different ways when its constituent symbols were changed, and this is not observed. The "interpretation" that is performed via the inference rules in an AI program amounts to no more than the transformation of *uninterpreted* symbolic structures. Thus, the actions taken by a program in solving problems deal with entirely uninterpreted symbols. In any case, the fact that symbols (or words) do not uniquely refer should not be a surprise; Wittgenstein [46] states quite clearly that there are many possible interpretations of a word; we can also imagine the situation in which many of the words that we use have different meanings—we would understand different things by the utterance of these words in the alternative world. This shows that what is meant, referred to, or denoted by a word is not a necessary property of that word.

There is a problem here, or so it would seem. MYCIN is clearly, for us, about medical diagnosis, HEARSAY-II is clearly about speech understanding, and R1 is clearly about computer configuration. If one were to provide R1 with an absurd requirement for a computer, it would not be surprising if the output were equally absurd. By the above argument, the program itself cannot determine what makes sense. It is we who determine what is reasonable and what is not. The sequence of inferential steps (the "acts" performed by the processor) together add up to a path from the initial to the solution state, yet these steps are manipulations of uninterpreted symbols. Even if one were to give a model-theoretic semantics for the representation, there would still be no guarantee that the program really does operate on structures which actually do refer properly to the external objects we believe it to be reasoning about.

At this point, it seems necessary to make a distinction. What the AI program deals with is a formal structure that is intended to represent something or other.

This formal structure is a *theory* about the external world (or whatever the program is supposed to reason about): sometimes, the word 'representation' is used to denote the theory. The 'representation' in more general terms is slack usage, and denotes the representation *language*—the general linguistic framework that is used to state (or articulate) the theory. At the end of the last paragraph, it is the first sense that we intended.

The last few paragraphs are intended to show that AI programs contain theories that are composed of uninterpreted symbols. The processor does not interpret those theories, nor does it interpret the symbols the theory contains. My argument is that we supply the necessary meanings. In [17], Hayes is concerned with the concept of meaning in AI programs. This is one reason why he (correctly) rejects the idea that the Herbrand interpretation can serve as the 'meaning' of a theory in a representation language. Hayes goes on to consider reflection principles, and assigns the meta-language the role of providing meaning (which is where "meaning" is usually assumed to reside in logic, but more of this anon). Finally, Hayes considers the ultimate source of meaning to be external to the representation and its processor, and says that people can provide meanings for their programs. This last claim is similar to Kripke's concept of "reference-borrowing"[21].

"Reference-borrowing" is a way for an agent (a person in Kripke's original paper) to be able to refer correctly to objects with which they are not directly acquainted. I am able to refer correctly to Cicero in sentences such as "Cicero was a Roman lawyer" or "Cicero criticised Caesar" even though I have never encountered Cicero, and all my evidence for his existence is indirect (for example, a collection of his letters were the set text for my O-Level Latin course, so I have read what I believe to be some of his writings). What happens when I utter a sentence about Cicero is that I "borrow" the reference from others who have had more direct contact with him. There is a long chain of borrowings from me right back to the people who actually knew Marcus Tullius Cicero: a chain which stretches across two thousand years. What Hayes suggests is that programs are able to "borrow" reference in a way similar to this: if I interpret a particular symbol, `aunt`, in the knowledge base as representing the relation *aunt*, the program can borrow that reference from me—because the symbols that we use in our AI programs tend to be words in ordinary language, the representations that we employ "borrow" their references from us as language users. If reference borrowing works, we can allow our programs to have meanings in a straightforward way; if it does not, then we are no better off.

Sterelny [42] argues that the reference-borrowing theory is inadequate. It is inadequate when the agent (person or machine) that is doing the borrowing does not have a sophisticated cognitive apparatus with which to know what is being referred to. AI programs at the moment, and young children learning language do not possess the cognitive sophistication to borrow references. For example, if I add the formula `man(cicero)` to my program's knowledge base, reference-borrowing requires that the program already have borrowed the referent of `man`. It is not enough for the program to contain a formula along the lines of:

```
man(X) if-and-only-if human(X) and male(X)
```

because this just means that the referents of `human` and `male` must also be known: for the average AI program, this is not the case. As Sterelny and others, for example, Fodor[12], have argued, this would *never* be enough: what prevents a program from classifying *chair* as something wooden as opposed to something one sits on?

The main alternative account of reference is the *description* theory. Referents can be identified, according to this theory, if a sufficiently "good" description can be obtained. A description, here, is a sentence in some language (logic, if you will). However, reflection indicates that description theories are no better off than the Kripke-style reference "borring" theory: they too presuppose cognitive sophistication, or, at least, they presuppose that all the terms that appear in the description have been individuated in an adequate fashion. For artificial objects, typically those of mathematics and logic, descriptions can be provided (although one might want, eventually, to relate everything back to numbers or to sets, which would make the descriptions rather large): these concepts have crisp boundaries. A problem which has to be faced for natural kinds is that the kind of sentence which uniquely descibes them is, in general, an infinite conjunction: clearly, such a sentence is not representable in a finite device. The formula that appears in the last paragraph can be viewed as a description of the class of men (male humans, that is). Although this concept is relatively crisp (there might be borderline cases—it is conceivable[15]), to make this description work, one needs to have the descriptions for `male` and `human`, the latter is certainly quite a long sentence.

Descriptions can be used to classify, but they are not the same as the things they describe. A chair is a chair, but a description of a chair is a description: a sentence in some language, formal or not. As aids to classification, they can succeed or fail; they may not even have a truth-value (as in "The present king of France"); as descriptions, they stand at one remove from the objects they are meant to capture. However, descriptions still require sophisticated cognitive powers.

5 Models, Truth-conditions and Reference

Hayes appears be incorrect on a number of points:

1. He confuses model-theoretic semantics with 'meaning'.
2. He mistakenly believes that there can be a unique model for a theory expressed as a set of first-order sentences.

[15] There have been reports of infants born with both male and female genitalia, so here is a case in point.

3. He believes that reference can be "borrowed" by a representation or processor that is not endowed with sophisticated cognitive apparatus.

He also makes the claim that 'meanings' can be fixed by reflecting into the meta-language. This last claim is of interest, and I will be returning to it below.

In this section, I will begin to present an alternative to Hayes' account, while still bearing in mind the various readings of Hayes' position. Before doing so, I want to point out a possible confusion: content is not the same as reference, and reference is not the same as meaning (whatever the last is). To see the difference between content and reference, one only need consider the case of an utterance (or proposition) which has content, yet does not refer: "The present King of France is bald" or "Mathilda's unicorn is delighted" have clear senses, yet neither refers to any extant object (certainly, there *was*, once, a King of France, but there is none now in 1991; as far as I know, unicorns have never been said to exist in the sense that lions, ring-tailed lemur or hippopotomus[16] exist). As has been pointed out since Frege, propositions can fail to refer, yet make perfect sense; even when they refer, there is a sense in which we would assent to their having 'content'—in the unicorn example, we would all, I belive, agree that the sentence was 'about' unicorns. A similar kind of argument can be used to show that, even if the words in a sentence all refer, the sentence need not *necessarily* be meaningful (Chomsky's "Colourless green ideas sleep furiously" is one such).

What, then, is meaning? It seems relatively clear that accounts of reference and of content can be given. This is not the place to give an account of meaning (even if I held a definite position): space forbids it, for one thing. Instead, I offer an account very close to that of Barwise and Perry[1]. Consider the case of an argument between A and B. At some stage, both A *and* B utter the sentence "I am right; you are wrong". Each means it of the other, so, in one case 'I' refers to A, and in the other it refers to B; similarly, 'you' refers to both A and B, depending upon who is speaking. Clearly, these two sentences cannot state propositions which are both simultaneously true (that both A and B claim that their *token* sentence is true is another matter). Equally clearly, the two utterances refer and they both have meaning: in fact, they both have the *same* meaning—both sentences exhibit a kind of regularity across contexts of utterance. For the present discussion, I will associate meaning with this kind of regularity (see [1] for a more detailed discussion)[17].

The point of discussing these distinctions is because I want to be clear that the three concepts are not the same. What I want is to distinguish them, and for reasons other than clarity.

Now, Hayes, like all logicists (for example, [13]) wants the objects in his representations to refer to objects in the external world, and he says that one can have a model (in the technical sense) of a theory which *is* the external world. He clearly

[16] Or any rare and/or exotic creature.

[17] I am not entirely happy with the account given by Barwise and Perry for a number of reasons—some metaphysical.

sees that there has to be a relationship between the entities that are represented and the entities themselves. If there is no such relationship between the objects of the representation and the "corresponding" external objects, there is no clear sense in which it can be claimed that the representation is of anything[18]. What is clear is that if something is a representation of something else, the representation must bear some relation to the thing that it represents (one such relation is *represents*, but what is needed is an analysis of representing relations). The problems begin when one wants to represent some *unique* object and when the world changes. In the former case, one must give an *individuating* description; in the latter, those propositions which are true (the facts, one might say) may change.

Hayes is quite clear that there is no *formal* or *logical* principle to which one can appeal in order to guarantee the connection between formalism and external world: that is why he suggests the reference-borrowing account. Now, to be fair, it must also be said that he suggests 'attaching' symbols to the output of sensors if the representation is being used by a robot; he also discusses the use of the meta-language for giving meanings to the symbols that are *internal* to the formal system (the reading that we gave above was the one which he might be *claimed* to prefer, his statements to the contrary). Reference-borrowing is used as the way in which representations acquire meanings (referents, actually) when the system in which they are embedded (I can think of no other term) does not have sensors. Whatever the representation, it will be *about* things that are *external* to the representation itself: this is because a symbol in a representation language is only a symbol until it is *interpreted* in some way—it is how this 'interpretation' is done that is the issue.

The suggestion that symbols can be 'attached' to the outputs of sensors does not seem, at first glance, to be particularly strange: this is because, perhaps, we are used to machines sensing some external environment. What is it, though, that makes this attachment? One answer is that we, the builders of systems, do it: we make the connection between the sensor's output and what it is supposed to denote (or maybe represent). If one follows the strict model-theoretic account of semantics, what one has is the fact that, for any formal theory, there are many (potentially infinitely many) models—leaving out, that is, the Herbrand models, but they are just collections of symbols, and are, thus, no genuine account of meaning or denotation for they merely give an interpretation in terms of the symbols of the theory. Because there is such a choice of model, in one sense, there is no concept of a definitive or actual model of a theory. In other words, the models *do not* give an account of meaning (or even of reference). One consequence of this is that symbols are still 'meaningless' even if one has given a model. A counter-argument is that amongst the models, there will be one which corresponds with reality: but who is to say that, especially if there is an infinite collection of models? (I.e., there is no effective procedure for deciding that there is at least one such model.) Furthermore, if one restricts oneself to those models which are used in model theory, there must be at

[18] A standard hedge is to claim that the representation contains 'abstractions'.

least level of interpretation before one can say that one has a model of reality (or some piece of it): this leads to an infinite regression.

The moral is that, by concentrating on *purely* formal structures, one will never *directly* give a sense, reference, interpretation or model in terms of the external world the theory is supposedly 'about'. What *must* be done in such circumstances is for some agent external to the formal theory to *decide* that the model is a suitable one. In a similar fashion, it is not possible to give a *unique* interpretation to the sentences of the meta-language because it suffers from exactly the same problem: at some stage, a semantics must be given for the meta-language[19].

One might want to give a model-theoretic semantics for the meta-language, but this just postpones decisions. One might want to give an alternative semantics (say, an algorithmic one for an intuitionistic meta-language), but this does not solve things, either. One might want, like Davidson, to give a truth-conditional account: truth-conditions are inadequate for giving meanings. A truth-conditional reading is along the following lines. In the meta-language, there is a special predicate, T, such that $T(\phi)$ is read as "ϕ is true", for any sentence ϕ in the object-language. A truth-condition is another sentence, ψ such that:

$$T(\phi) \leftrightarrow \psi$$

Unfortunately, if $\psi \leftrightarrow \theta$, then

$$T(\phi) \leftrightarrow \theta$$

also. There are no conditions imposed upon what ψ and θ are, so *any* sentence θ will do as long as (i) it is logically equivalent to ψ, and (ii) as long as it is true. There is no circularity here, note, for we can assume that ψ and θ belong to the meta-language, and they are assigned truth by the predicate $T_{\mathcal{M}}$, the truth-predicate for the meta-language ($T_{\mathcal{M}}$ belongs to the meta-meta-language). This is the case for any language whose semantics is being presented along Tarskian lines. All that matters is that ψ and θ have the same truth-value: ψ and θ need not even mention the same things! (That is, ψ could be "The sentence 'Grass is green' contains three words", and θ might be $\forall x.x = x$. Strengthening by modal operators does not work, either. The account still fails even if we impose the condition that $\psi \prec \theta$.) Nothing *in* the theory guarantees the same subject-matter, that is.

What one *can* do is to talk about the structures which are entirely internal to the object-language theory in ways that do not refer to the outside world[20]. The simple answer is that the symbolic structures of a formal theory are not the external objects they are intended to represent—they are symbols *simpliciter*—and it is *we*

[19]The semantics might include denotations for the sentences or symbols of the object-language that have been included in order to give them an interpretation.

[20]Even here, though, I have doubts, and I find these doubts disappointing, being one who is interested in the language/meta-language distinction, and being one who finds considerable appeal in the idea of using the meta-language to give interpretations. However, see below for more on this.

who assign the interpretations. In other words, *symbols are not the same kinds of thing as the objects they represent.*

The main point of this section is to add weight to the case that it is *we* who assign meanings to formal representations. None of the moves suggested by Hayes (and the other logicists) helps in this respect. With the exception of reference-borrowing, what they lack is some kind of connection between the representation language and the world which is being represented. In other words, there is nothing in the scheme which they favour which gives the kinds of connection that one wants unless there is a person there to give the connection on behalf of the representation. Another way of saying this is that there can be no possibility of semantic originality (nor, incidentally, are there ways in which semantics—or meanings—can be enforced by the representation itself[21].

6 Causes and States

This section is to be re-written.

After the negative, I will try to present the positive. The positive amounts to a way of viewing things that *might* put representations in a position where they can do the things we want of them.

The first observation I want to make is that computers (processors of knowledge representations) are *causal.* That is, they are not the purely mathematical entities that is often assumed (see [19] for an exposition of the formalist, mathematical position). Certainly, one can give *descriptions* of the behaviour of programs and machines in terms of logical concepts and by means of proofs. What the logicists assume is that the purely logical properties are all that need be considered. What they forget is that machines are causally *efficacious.* Robot arms can move cups or weld car bodies; sensors of various kinds can detect changes in the non-computational environment which is external to the machine.

Even within a computer, the state of the store changes (I do not want to argue that one *must* attempt descriptions or accounts that descend as far as the quantum level). The behaviour of a processor depends changes of state. This can be seen in some descriptions of logic processors—for example, the descriptions of intelligent agents in [13]. The clauses in the database of a resolution theorem-prover represent the state in abstract terms: the deduction of a new clause alters the state. In formal specification, states are an important aspect: for example, in Hoare logics, specifications have the form $\{P\}S\{Q\}$, where S is a statement of the programming language, $\{P\}$ is a description of the state before S is executed (the precondition), and $\{Q\}$ is a description of the state after S has run (the post-condition). Functional programming is also state-dependent because functional programs have to be

[21] In my earlier[6], I expressed the view that the meta-language might be called on to do this. In some ways, I want this still to be the case, although the scope of such enforcement might be far less than I had hoped.

executed by a physical processor.

One can argue quite convincingly, I believe, that programs change the state of the processor: that is, programs are methods for changing states. The clearest example of state change occurs in an assignment. In Hoare logic, the state before the execution of a statement is transformed into the state described by the states described by a formal specification have an interesting status because the specification must be invariant with respect to time and location (the also have to be neutral with respect to lots of other things, for example, word length): one might say that a specification is about equivalence classes of states. The remarks I have just made about specification might be objected to by some: they consider the precondition as determining those states which *permit* the execution of S, and that the post-condition determines the resulting state. In other words, if the precondition satisfies P then S may be executed, and the state that results therefore must satisfy Q. Conversely, for predicate transformers [10], the account is that for any state Q that obtains after the execution of S, the precondition will satisfy $\mathbf{wp}(S, Q)$. I tend to think that the two accounts are equivalent because they both deal with the relationships between the two states P and Q—one describing the state before and the other describing that after execution of S. The use of words like "permits" tends to suggest a way of viewing the relationship between the states which is more to do with use; in any case, what is wanted is that $P \rightarrow Q$ (which can either be read as material implication or state transition).

The important things are that the notation (Hoare logic, predicate transformers, Z[41] or VDM[20]) relates descriptions of states, and that the programming language statement is considered to change the state. At a macroscopic level of analysis, one might want to view (simple) statements as events (strictly, that they represent classes of event)[22]. What is essential to remember is that descriptions of states are not the states themselves, and that descriptions of or prescriptions for events are not, themselves, events.

Apart from establishing that states are part of the computational story, what this is intended to show is that *causality* is an essential ingredient of that story. The fact that one can abstract away from causality and work within a system of formal descriptions (or in a functional language) does not detract from this. In a similar way, the processor for a knowledge representation language can be thought of as being an entity which depends upon causality for its operation. One of the factors that was identified above for a representation system is that it makes inferences: inferences are a kind of action. The application of an inference rule causes the processor to change state: what was said above was that an inference rule can be viewed as a kind of transformation—what can now be seen is that an inference rule can be thought of as a relation between states in a non-derivative sense. This point is, perhaps, obscure and some might argue that it has more to do with the way

[22] Events figure, too, in theories of concurrent processing, for example [18] or in descriptions of message-passing systems in general.

in which we implement our representation systems and their processors. However, consider the proposition that A tells B that his shoe-lace is undone: if B thinks that A is telling the truth, B will believe A (and may also do up his shoe-lace[23]). What A utters to B elicits a behaviour, a physical act. One can say, to an approximation, that B's belief (engendered by A's utterance) *caused* the behaviour of tying up the shoe-lace[24]. The relationship between belief and action has almost never been in serious doubt (nor, since Descartes, at least, has the concept of one idea's causing another).

One can hold a belief as a result of direct experience: for example, one might see a magpie in a tree and believe that there is a magpie in that tree on that day and at that time. What one would say is that the perception caused the belief: this does not seem objectionable. Apart from the 'input' of perception, there is the production of action. Actions change the world in various ways by causing things to happen in the ways that we want (for example, pulling up weeds in the garden so that they do not cause our plants to be overgrown). This is what I mean by *causally efficacious*: we can change the world in ways that we want or in ways that we believe will improve matters as far as we are concerned. Inference can also be analysed as a (special, i.e., mental) kind of action: it changes one's beliefs. If a dispositional account of belief is required (so as to avoid problems such as maintaining an active belief state while one is asleep), inference can still be seen as altering one's *disposition* to believe something (assert it as true, which is what most analyses amount to)—see [40], for example, for an discussion of dispositions. For my own money, the difference between a belief state and a belief disposition is that the latter incorporates the notion that beliefs can be inactive or latent—one can have them without being aware that one has them.

Part of the point I am trying to make is that causality is an ingredient (perhaps *the* ingredient) in meaning and reference. Although Kripke's reference-borrowing theory has attractions, it fails because of what it presupposes. Description theories presuppose also that there are adequate categories for us to understand descriptions (hence the attempts by many, Russell and Wittgenstein, amongst them, to ground descriptions on deictics such as "here" and "this"—the so-called 'logically' proper nouns). What any theory of reference must do is account for the ways in which we can refer to distal as well as proximal objects. The theory proposed by Devitt and Sterelny[42] attempts to steer a course between the extremes of reference-borrowing

[23] My original example was that B's flies were undone: under normal circumstances, B will *certainly* attempt as soon as possible to do them up—here, the resulting behaviour is almost automatic.

[24] Note that I am not suggesting that all beliefs cause behaviour in such an immediate way: there are beliefs that we hold which do not cause behaviour. For example, beliefs that one is not attending to. If one's beliefs *always* caused one to act, one would never rest! Also, one might have a belief and not be conscious of it, yet it may cause action of some kind—the kind of behaviour that is pointed out to us, an explanation offered (which seems just right), and to which we reply "I never realised that!"

and description theories by proposing a hierarchy of causal and descriptive references. Causal reference is most prominantly associated with lower-level (or 'earlier' in the developmental sense) objects. The main attraction of the theory is that is enables one to build from proximal reference into distal ones by providing a context within which cognitive capacities can be developed.

The whole point of considering *causal* theories of reference is that they provide a way of connecting representations with what is being represented. Without causal grounding, descriptions do not capture the connection: this is so because there may be many objects that fit a given description. If one is causally involved[25], then there is one object which is causally involved in the process of referring. If one is situated in the world, one can do a lot more than engage in inference about objects: one can touch, smell, see them, for example. In other words, perception and sensation can directly cause beliefs: in a formalist account, one has of necessity to explain how percepts and sensations are encoded in the formal language—in other words, the *linguistic* assertion "I am in pain" is indissoluably linked to the state of being in pain, whereas one can be in pain without ever saying so to anyone (including oneself).

At this juncture, a hard-line logicist might reply in one of a number of ways. The first way is that causality is notoriously hard to understand: Hume, for example, could find nothing more than constant conjunction to explain causality. The axioms of causality are nigh impossible to work out (there have, of course, been a number of—mostly unsuccessful—attempts to give formal theories of causation: inductive theories have been prominent[30], and a more recent proposal even involves *non-monotonic* logic[36]). This reply simply gets everything back-to-front: the claim is not that we should try to *model* causation, but, more simply, that it all rests on causality—causes and effects are the bottom line, so to speak[26]. The second reply is that causality is simply irrelevant. The second reply breaks down into two components:

1. Causation is not required to account for representation and behaviour.

2. Logic is not supported by causal relations.

My comment in response to (1) is that there is no other mechanism that I know of that will give the kind of connection between representation and represented that is necessary in order for any agent that possesses such a representation to use it in meaningful or efficacious ways. Without causation, nothing happens. It has already been argued at some length that *purely logical* relations cannot account for reference or for meanings: thus, any agent that relies on logic alone cannot change the environment in which it resides. My reply to (2) is that *logic* may not be

[25] I prefer the term *involvement* to *connection* or *connectedness* because it also carries connotations of action; it is, in any case, a weaker term without connotations of determination.

[26] One can describe a mouse as much as one wants, but the descriptions do not make it move.

dependent on causation, but the *people* who do it (i.e., the people who use and do logic) are: is there logic without people who do logic?

The above arguments suggest, I believe quite strongly, that at some point, to mis-quote Wittgenstein[46]: *formalism must come to an end.*

All of this leads, eventually, to understanding. I do not want, here, to begin to discuss the various theories that have been proposed over the years: that would take far too long. However, I want to suggest that theories which take the external world seriously provide a better framework than do others. What I want to suggest is that we only know that someone understands something by the way in which they behave in critical situations—the mere assertion "I understand that" leads to nothing unless the person who utters it can actually *show* that they understand. If the claim is that X understands *how to* ϕ, then we cannot be convinced of this until X actually does it. Equally, if X claims that he or she understands ϕ, we would only be convinced if X behaved in such a way that the implications of ϕ were known: i.e., that the implications of ϕ alter X's behaviour in some significant way. For example, if X understands arithmetic, we would expect X to be able to add and to exhibit an understanding of numbers. In a similar way, we would assent to the proposition that X understands ϕ (the sense in which 'understand' is taken to be a cognate or near-cognate of 'know' or 'believe') if X behaves in a certain way. If X understands that walking too close to the edge of the river entails running the risk of falling in, we would expect X to stay away from the edge of the bank. In other words, all we have is behaviour on which to judge understanding: the behaviour can be verbal (X can give justifications or explanations in terms of ϕ, or can use ϕ in ways which allow us to infer an understanding), or they can be non-verbal (for example, completing a proof in first-order arithmetic). In the case of understanding taken as a near-cognate of 'believing', what we are interested in is the implications that X draws from this belief.

For a program, merely getting to the right answer need not convince us that it *understands* something (for example, the fact that either MYCIN or PUFF produces a correct diagnosis need not convince us that these programs in any way understand meningitis or pulmonary diseases). Even the 'explanations' that they give do not allow such a conclusion to be drawn (in these cases, there is not the luxury of asking further, more probing, questions)—we require depth as well as breadth in explanation. In general, though, we want more than the right answer: we want justifications, explanations and use of concepts—one can get to the right answer by the wrong route (recall the paradox of material—or of strict—implication). The ways in which ϕ is *used* contributes to our claiming that "X understands ϕ".

Now, I am *not* claiming that meaning *is* use. What I am saying is that use contributes to meaning: there are ways of using concepts that are accepted as meaningful, and there are others which are not. In other words, there are some ways of using concepts that we count as indicative of meaningful use, and there are others that we would reject (some uses do not immediately seem to satisfy this: for example, the coining of a new metaphor). What I am saying, ultimately, is that the ways in which words, in particular, are used is the criterion which we use in determining whether or not the person who utters them understands them. That is, the outward behaviour of the person, the actions which they perform, determine for us whether they understand or not. The context in which the behaviour is elicited is also important in determining whether there is understanding. Some actions will be prohibited the situation. In some contexts, what is perceived to be appropriate also enters into matters. However, it remains the case that actions provide the best way of determining understanding.

In order to make the claim that "X understands ϕ", it is necessary for there to be someone else who can make that claim: understanding depends upon factors external to the agent (cf. Wittgenstein's *Private Language Argument*[46, 45]). For a processor to apply a rule, it is necessary for there to be criteria which guide its application, and there must be someone or something which can determine *that* the rule has been correctly applied. One can build a rule-based program and allow its processor to apply rules, but there is nothing in the processor *on its own* that will allow it to assert that its rules are correctly applied. What is lacking is the social dimension: there are conventions that are applied.

The above discussion of understanding and rule-application relates to the distinction I drew early on concerning the two main components of an AI program: the knowledge base (or database) and the inference engine. The knowledge base, it will be recalled contains an explicit representation of the knowledge that the inference engine applies in solving problems. The interpretation of the declarative representations that the knowledge base contains was the issue that prompted the entire discussion of model-theoretic semantics and its inadequacies. What do the structures in the knowledge base mean, and to what do they refer? How is the representational relation maintained?

The account which is given by the logicists is that the inference engine is a necessary component only because, without it, one would not have a program that worked: the processor is an empirical requirement. The fact that it is supposed to be domain-independent suggests that its workings are not the *real* concern of the knowledge representation enterprise, for it contains no knowledge (perhaps it contains knowledge, in some sense—a sense closer to the way in which payroll programs might be said to have knowledge—of how to process the declarative structures in the representation proper). It is the contents of the declarative database that are the real concern, it is argued, for it is they that actually represent what the program is about, and it is they, together with sound inference rules, that guide the processor in giving up correct answers. In a sense, the only interesting property of the inference engine—the processor—is that it gives the *right* answer.

It has been argued that the account usually given of the declarative representation falls far short of what is required: the standard account (in terms of model theory) ignores the relationship between what is represented and what is doing the representation. If model theory is all that can be given by way of an account of meaning, reference and content ('semantics' in intuitive sense, or in the sense of ordinary, natural, language), then one might as well be using the Herbrand model. The argument I have given requires that semantics be *explicit* related to the external world. I have also argued at some length that causal theories need to be used in order to illucidate the relationships between what is being represented and what is representing—taking the external world seriously, in other words. I have also argued for an account of representations in artificial processors in terms of causal relationships (this is point of the argument about what software specification is about). It is the *behaviour* of the processor that determines the ways in which the representation are viewed, not whether the representation language or the theories expressed in it have models in the sense of the model theory of first-order theories.

Finally, and this point is worth making, the purely behavioural aspects that have been the focus of attention are, in themselves, inadequate. An intelligent machine could be *simulated* by getting the population of China to pass messages written on pieces of paper. No-one would claim that the Chinese population when engaged in this activity *was* an intelligent machine, nor would the population be doing anything more than passing messages in an attempt at simulation. The point of this argument is that the global behaviour (when viewed by an experimenter who merely posed questions and waited for answers) might be such that the experimenter would be prepared to say that an intelligent artifact was responsible for producing the answers thus obtained. What the argument is saying, though, is that causal relations of a different kind can be used in simulation, and the simulation may be as good as the real thing. The Chinese population argument is aimed at showing that a functionalist view of mind and meaning is inadequate because many, non-equivalent processes or devices can be used to simulate the behaviour of a given process or device: the components of such a simulating engine may not bear the same relationships as corresponding (or supposedly corresponding) entities in the original. Furthermore, the causal relationships exhibited by the Chinese population while engaged in the simulation exercise—while they wait for and pass messages—are *internal* to the simulation, not external. The moral for functionalists is that their theories of internal causal relationships are inadequate—a more ecological view might bring the closer alignment that is sought.

7 Conclusions

Throughout this paper, I have been concerned with *formal* theories and with their model-theoretic semantics. Some readers may have formed the impression, despite my protestations to the contrary, that my aim has been to argue that formality, formalism and mathematics have no role in AI: for the last time, let me state quite clearly that I am not criticising logic and model theory; nor am I claiming that logic has *no* role in knowledge representation. It is my opinion that the use of logic has a great clarificatory role: by dint of being formal, one must think carefully about what one wants to say. Use of logic also brings with it the advantage that inference of a well-understood kind can be used in determining the implications of one's formal theory; if a model is given for the theory, one has the choice of where one's inference is performed (very often, it turns out to be easier in the model).

What I have tried to criticise is the view that a formal account (a theory and a model) is *all* that one needs. One can give a formal account, certainly, but one should not be deluded into believing that, once the formal stuff has been done, there is nothing else to do. The formal account can be used as a *specification* of what is wanted: it can serve as a *meta*-theory, that is. What I have argued is that representations are representations of *something*, and that *something* is not a formal structure in the way that, say, number theory is. In order to represent, one needs some kind of *connection* between what is doing the representing and what is being represented. This connection determines the content of the representation. The problem with the purely formal accounts of semantics that are to be found in the knowledge representation literature (e.g., [29, 25]) is that it does not go far enough along the road to connection with the external world. One can, of course, present a model in terms of external physical objects and the relations which obtain between them: however, the relations that we want are part of a model[27] that we impose on the external world—try to find an instance of "to the right of" in the world.

There is, in addition, the problem of finding the right (mathematical) conception which is not only faithful to what one is trying to capture in one's non-logical axioms, but which is also easy to manipulate as a formula. By choosing an inappropriate conceptualisation, one can end up with paradoxical or just silly results (for example, the consequence that something which is inside a spatial region can also be outside it, or 'implausible' or 'impossible' objects). As far as logic and model theory are concerned, there is no question of a paradoxical or silly result: in a sense, the ontology is outside of logic. The conceptualisation that one employs depends, naturally, upon an adequate choice of object to populate the world one is trying to represent. For purely formal accounts, there is nothing except our own judgement to determine what makes sense.

The conclusions I have reached in this paper are, in many ways, surprising to me: I have spent a long time trying to find ways of getting meanings out of representations without having to deal with messy details of causal mechanisms. I almost view the conclusions reached in the last section as a partial vindication of the approach that I took with my ELEKTRA rule interpreter [7, 8, 9]: what worried me most about ELEKTRA was that it seemed not to have an elegant, mathematical, theory of representation, and attempts to provide one foundered. The problems I had in producing a *denotational* semantics for ELEKTRA were also a worry: what resulted from the attempts I made looked more like a causal analysis than anything

[27]The use of 'model' here is of a different category from that we have been using in connection with logic.

else. Apart from under-determining too much, ELEKTRA is also a solipsistic processor: it is not connected with the world, and has to borrow references from its user (so, by the argument against Hayes, it must fail). The approach I initially took to ELEKTRA was to give it a formal specification in the Z language[41]: this seemed to be a good place to start if one wanted an implementation, but seemed to be completely wrong if one wanted a deep and elegant mathematical theory. By the argument about processors, it seems that the specification is along the right lines: it deals, albeit indirectly, with states and events. None of the above should be construed as a claim that ELEKTRA is a representative of *the* correct way of viewing things: I tend to believe that it is all *wrong*, but that some of the basic intuitions were correct.

In a changing world, the models that we might want for our theories will change as well. When I started writing the notes for this paper, the large tree which I can see when I look out of the window was not in bud: eight weeks later, it has leaves. Furthermore, what we want is for representations to connect with what they represent in such a way that I can represent the proposition that there is a large tree outside my study window: that representation must capture the property that there is one *large* tree, a smaller one, and, to the left and just out of sight from the study, there is a large oak. As has been argued, the *referential* aspects of representation are very important, but a model-theoretic account ignores this.

The position that I argue for can be summarised as "it is necessary to take the world seriously". In other words, the metaphysical choices one makes should not be determined entirely by what one can do in a formal system (consider the results of Russell's logical atomism[35], or the metaphysics of the *Tractatus*[44]). Another way of putting this is to say that model theory in representation only serves to mystify the semantic enterprise, positing as it does, a platonic realm of objects which we can never perceive and which do not enter into causal relationships with other objects (even Hayes' remarks about the real world serving as a model are subject to this accusation, for can we take the intersection of a pile of three apples and another pile of four apples—physical piles of real apples, that is? we can only do this when we *represent* the piles as sets, and this requires that we leave the world and enter the realm of sets and collections).

References

[1] Barwise, J. and Perry, J., *Situations and Attitudes*, MIT Press, Cambridge, MA, 1983.

[2] Barwise, J., Situations and Small Worlds, in Barwise, J., *The Situation in Logic*, CSLI Lecture Notes No. 17, pp. 79-92, CSLI, Stanford University, 1989.

[3] Besnard, P., *An Introduction to Default Logic*, Springer-Verlag, Berlin, 1989.

[4] Brachman, R., Fikes, R. and Levesque, H., KRYPTON: A Functional Approach to Knowledge Representation, *IEEE Computation*, Vol. 16, pp. 67-73, 1983.

[5] Brachman, R. and Levesque, H., A Fundamental Tradeoff in Knowledge Representation and Reasoning, *Proc. CSCSI-84*, pp. 414-152, London, Ontario, 1984.

[6] Craig, I.D., *Meta-Knowledge and Introspection*, Research Report, Department of Computer Science, University of Warwick, 1990 (*in prep.*).

[7] Craig. I.D., *ELEKTRA: A Reflective Production System*, Research Report No. 184, Department of Computer Science, University of Warwick, 1991.

[8] Craig, I.D., *Rule Interpreters in ELEKTRA*, Research Report No. 191, Department of Computer Science, University of Warwick, 1991.

[9] Craig, I.D., *The Formal Specification of ELEKTRA*, Department of Computer Science, University of Warwick, 1991 (*in prep.*).

[10] Dijkstra, E. *A Discipline of Programming*, Prentice Hall, New Jersey, 1976.

[11] Feigenbaum E.A., Nii, H.P., Anton, J.J. and Rockmore, A.J., Signal-to-signal transformation: HASP/SIAP case study, *AI Magazine*, Vol. 3, pp. 23 - 35, 1982.

[12] Fodor, J.A., *Psychosemantics*, MIT Press, 1988.

[13] Genesereth, M.R. and Nilsson, N.J., *Logical Foundations of Artificial Intelligence*, Morgan Kaufmann, Palo Alto, 1987.

[14] Hayes, P.J., Some Problems and Non-Problems in Representation Theory, *Proc. AISB*, pp. 63-79, University of Sussex, 1974.

[15] Hayes, P.J., The Logic of Frames, in Metzing, D. (ed.), *Frame Conceptions and Text Understanding*, pp. 46-61, de Gruyter, Berlin, 1979.

[16] Hayes, P.J., In Defense of Logic, *Proc. IJCAI-5*, pp. 559-565, Morgan Kaufmann, Los Altos, CA, 1977.

[17] Hayes, P.J., The Second Naive Physics Manifesto, in Hobbes, J.R. and Moore, R.C. (eds.), *Formal Theories of the Commonsense World*, pp. 1-36, Ablex Publishing Corp, Norwood, NJ, 1985.

[18] Hoare, C.A.R., *Communicating Sequential Programs*, Prentice Hall, Englang, 1985.

[19] Hoare, C.A.R., Inaugural Lecture, University of Oxford, 1985.

[20] Jones, C.B., *Systematic Software Development Using VDM*, Prentice Hall, England, 1986.

[21] Kripke, S.A., Naming and Necessity, in Harman, G. and Davidson, D., *Semantics of Natural Language*, Reidel, Dortrecht, 1972.

[22] Levesque, H.J., Foundations of a Functional Approach to Knowledge Representation, *Artificial Intelligence Journal*, Vol. 23, pp. 155-212, 1984.

[23] Lakemeyer, G. and Levesque, H.J., A Tractable Knowledge Representation Service, in Vardi, M.Y. (ed.), *Proc. 2nd Conf. on Theoretical Aspects of Reasoning about Knowledge* 145-159, Morgan Kaufmann, Los Altos, CA, 1988.

[24] McDermott, D., Planning and Acting, *Cognitive Science*, Vol. 2, pp. 71-109, 1978.

[25] McDermott, D. and Doyle, J., Non-monotonic Logic I, *Artificial Intelligence Journal*, Vol. 13, pp. 41-72, 1980.

[26] McDermott, D., A Temporal Logic for Reasoning about Processes and Plans, *Cognitive Science*, Vol. 6, pp. 101-155, 1982.

[27] Minsky, M.L. (ed.), *Semantic Information Processing*, MIT Press, Cambridge, MA, 1968.

[28] Moore, R.C., The Role of Logic in Knowledge Representation and Commonsense Reasoning, *Proc. AAAI-82*, Morgan Kaufmann, Los Altos, CA, pp. 428-433, 1982.

[29] Moore, R.C., *A Formal Theory of Knowledge and Action*, CSLI Report No. CSLI-85-31, CSLI, Stanford University, 1985.

[30] Mortimer, H., *The Logic of Induction*, eds. Craig, I.D. and Cohn, A.G., Ellis Horwood, England, 1988.

[31] Newell, A., The Knowledge Level, *Artificial Intelligence Journal*, Vol. 18, pp. 87-127, 1982.

[32] Nilsson, N.J., Nilsson, N.J., Logic and Artificial Intelligence, *Artificial Intelligence*, Vol. 47, pp. 31-56, 1991.

[33] Patel-Schneider, P., A Decidable First-Order Logic for Knowledge Representation, *Proc. IJCAI-85*, pp. 455-458, 1985.

[34] Reiter, R., A Logic for Default Reasoning, *Artificial Intelligence Journal*, Vol. 13, pp. 81-132, 1980.

[35] Russell, B., *The Philosophy of Logical Atomism*, Open Court, La Salle, Illinois, 1985.

[36] Shoham, Y., Nonmonotonic Reasoning and Causation, *Cognitive Science*, Vol. 14, no. 2, pp. 213-252, 1990.

[37] Sloman, A., Interactions between Philosophy and AI – the Role of Intuition and Non-logical Reasoning in Intelligence, *Artificial Intelligence Journal*, Vol. 2, 1971.

[38] Smith, B.C., *Reflection and Semantics in a Procedural Language*, Tech. Report MIT/LCS/TR-272, Computer Science Laboratory, MIT, 1982.

[39] Smith, B.C., *The Correspondence Continuum*, Report CSLI-87-71, CSLI, Stanford University, 1987.

[40] Smith, P. and Jones, O.R., *The Philosophy of Mind*, CUP, 1986.

[41] Spivey, J., *The Z Notation: A Reference Manual*, Prentice Hall, England, 1989.

[42] Sterelny, K., *The Representational Theory of Mind*, Blackwell, Oxford, 1990.

[43] Waterman, D.A., Lenat, D.B. and Hayes-Roth, F., *Pattern-Directed Inference Systems*, Academic Press, New York, 1978.

[44] Wittgenstein, L., *Tractatus Logico-Philosophicus*, trans, McGuinness, B.F. and Pears, D., Routledge and Kegan Paul, London, 1961.

[45] Wittegenstein, L., *The Blue and Brown Books*, tr. R. Rees, Blackwell, Oxford, 1958.

[46] Wittgenstein, L., *Philosophical Investigations*, trans. G. E. M. Anscombe, Blackwell, Oxford, 1958.

[47] Woods, W., What's In A Link: Foundations for Semantic Networks, in Bobrow, D. and Collins, A. (eds.), *Representation and Understanding: Studies in Cognitive Science*, pp. 35-82, Academic Press, New York, 1975.