Notre Dame Journal of Formal Logic
Volume 36, Number 3, Summer 1995

Avoiding Omnidoxasticity in Logics of Belief: A Reply to MacPherson

KIERON O'HARA, HAN REICHGELT and NIGEL SHADBOLT

Abstract In recent work MacPherson argues that the standard method of modeling belief logically, as a necessity operator in a modal logic, is doomed to fail. The problem with normal modal logics as logics of belief is that they treat believers as "ideal" in unrealistic ways (i.e., as omnidoxastic); however, similar problems re-emerge for candidate non-normal logics. The authors argue that logics used to model belief in artificial intelligence (AI) are also flawed in this way. But for AI systems, omnidoxasticity is impossible because of their finite nature, and this fact can be exploited to produce operational models of fallible belief. The relevance of this point to various philosophical views about belief is discussed.

1 Introduction: modeling beliefs with logic and the problem of omnidoxasticity A recurring problem in the philosophy of mind has been the logic of reasoning and belief (cf., e.g., Rescher [17], pp. 3–96 for some examples). The most important recent paradigm was first canvassed by Hintikka [11], in which the belief operator **BEL** is seen effectively as the necessity operator of a modal logic. The operator would then be relativized to a subject, so we can say that i believes that p by writing '$\mathbf{BEL}_i\, p$'. This then allows the logic of belief to be given a semantics by exploiting familiar possible worlds semantics for modal logics. For example, one might understand a believer to be allowing for, or expecting, one of a number of situations to obtain; each situation which the believer thinks is possible can then be seen as a possible world accessible from the actual one, where the accessibility relation is defined relative to the believer (for an agent i, we will call this relation *i-accessibility*). Then we can say that $\mathbf{BEL}_i\, p$ in a world w iff $\forall w'$ i-accessible from w, $p \in w'$. The belief operator can then be embedded, so that beliefs about the beliefs of oneself and others can be represented (e.g., $\mathbf{BEL}_i\mathbf{BEL}_j\, p \equiv i$ believes that j believes that p).

Although this is an important paradigm, there are some familiar problems with it. To begin with, most well-behaved modal logics are normal, that is they admit the

Received March 22, 1995; revised July 13, 1995

axiom **K** ($\mathbf{BEL}_i(A \supset B) \supset (\mathbf{BEL}_i A \supset \mathbf{BEL}_i B)$), and the rule of necessitation (if $\vdash A$ then $\vdash \mathbf{BEL}_i A$). However, it is clear that, for the purposes of a doxastic logic, such normal logics are hardly going to be appropriate. The rule of necessitation means that any believer would believe all theorems of classical logic. Worse, the addition of **K** means that any believer would believe all the logical consequences of all his beliefs, or, as we shall say, would be *omnidoxastic*.[1] Normal modal logics are the logics of unbounded rationality, and that is not an assumption that one would wish to make about real believers. Any doxastic logic which intends to model the structure of belief for boundedly rational subjects must be nonnormal. The distinction is usually described by saying that normal logics give the structure of belief for *ideal* believers (cf. [17], p. 99), whereas the nonnormal logics must be used to describe the structure of belief for *actual* believers (cf. Konolige [15], p. 13).

However, using nonnormal logics to model belief makes things harder. To begin with, choosing a nonnormal logic means that the standard possible world-based model theory cannot be exploited fully (Williamson [19] discusses this issue in the context of developing syntactically-based ways of determining admissibility of rules in nonnormal modal logics); basically, the problem is that possible worlds are logically well-behaved, so that tautologies are always true in them (hence making it difficult to block the rule of necessitation), and they are closed under deduction (which will let **K** in).

But technical questions about modal semantics aside, in a recent paper [15], MacPherson raises a series of questions about various actual attempts to develop theories of belief that use nonnormal logics and shows that such attempts remain unsatisfactory. The issues that MacPherson addresses turn on the oft-noticed distinction between doxastic logics which allow unbounded rationality and those which do not. What we wish to claim in this paper is that there are ways to investigate the relation between bounded and unbounded rationality that do not involve the development of two kinds of logic, and that clues to alternatives can be garnered from Artificial Intelligence (AI). In the next section, we look at approaches to doxastic logic in AI and put forward positive proposals in Section 3. Finally, we discuss briefly the philosophical implications of our claim in Section 4.

2 Approaches to the problem in Artificial Intelligence

2.1 The need for doxastic logics in AI The logic of belief is an important problem for AI systems. There are many examples of areas where agents need to be able to reason both about the beliefs[2] they hold, and about the beliefs that other agents hold. One example is multi-agent problem-solving or planning systems, where artificial agents need to coordinate their actions with those of other (artificial and possibly natural) agents. Which actions an agent is likely to perform will depend on its beliefs about the world, so if agents *A* and *B* need to interact in a complex way, and *A* needs to anticipate *B*'s likely actions, *A* will also need to reason about what *B* believes.

Doxastic logic is a fairly natural place for AI researchers to turn for hints on how to perform such reasoning. However, it is absolutely essential for AI purposes that doxastic logics avoid omnidoxasticity. The logic of ideal reasoners is not a matter with which, at first blush at least, AI is concerned. That a believer ought, given ideal

resources to believe thus-and-so is at best irrelevant from the AI point of view, and at worst misleading. Since, in AI, one is interested in building artificial believers,[3] and interested also in such systems' interactions with other boundedly rational agents, the logical problem is precisely that of modeling the fallible belief structures of the finite. It is essential that the resource limitations of believers are taken into account in AI, and therefore a logic that models only the beliefs of ideal agents will not be acceptable.

Another way of making this point is to say that a logic which does not acknowledge the resource limitations of agents will lead to an intuitively wrong definition of logical consequence when reasoning about agents' beliefs. Because we know that agents are resource limited, we will not necessarily conclude that an agent believes $\neg p$ even if we know that it believes $\neg q$ and $p \supset q$. Therefore, an epistemic logic which leads to this conclusion unrestrictedly cannot be said to capture adequately the notion of belief that underlies both reasoning about the beliefs of resource-limited reasoners, and the design of artificial agents.

The problem of omnidoxasticity has two main aspects as far as AI research is concerned. First, according to omnidoxastic logics, an agent is assumed to believe all tautologies. $\mathbf{BEL}_i\, p$ is true in world w if and only if p is true in all worlds i-accessible from w. Since a tautology is true in all possible worlds, *a fortiori* it is true in all possible worlds i-accessible from w, and therefore if p is a tautology, $\mathbf{BEL}_i\, p$ is true in w. This is a very strong assumption as an account of natural reasoning.

Second, an omnidoxastic logic assumes that belief is deductively closed. Suppose $S = \{p : \mathbf{BEL}_i\, p \text{ is true in } w\}$, and that $S \models q$. Then every member of S is true in every world i-accessible from w. So, therefore, q must be true in every world i-accessible from w. So $\mathbf{BEL}_i q$ is true in w, and hence $q \in S$. Once again, this is too strong an assumption about resource-limited belief.

Two consequences of omnidoxasticity which have special importance for AI might also be mentioned. Omnidoxasticity implies that agents with inconsistent beliefs believe everything. This will not do as an assumption about boundedly rational believers from the AI perspective. Certainly, faced with a recognizable contradiction, any minimally rational agent would take action. For example, in the case of a contradiction of the form 'p & $\neg p$', it would withdraw one of the two conjuncts; in the case of a contradiction of the form p iff $\neg p$, no doubt such an agent would withdraw either 'if p then $\neg p$' or its converse. So there is no immediate problem when the contradiction is clear and recognizable. But it does not seem unreasonable that an entire set of beliefs might imply a contradiction which has not been discovered, but that that set of beliefs might also be a reasonable basis for suitably limited interaction with the world.[4] Furthermore, if belief is deductively closed, logically equivalent beliefs turn out to behave identically. The sentences *it is raining outside* and *arithmetic is incomplete and it is raining outside* are true or false together in all possible worlds. Hence in an omnidoxastic logic, the one is believed just in case the other is. But a system that reasoned about the beliefs of bounded agents should treat them differently.

Since the omnidoxasticity problems impinge on AI applications so acutely, it is not surprising that a number of efforts have been made within AI itself to meet the difficulties. In the remainder of this section, we will briefly review some of the major approaches and note that they are not obviously more successful than the straight

logical approaches reviewed, and rejected, by MacPherson.

2.2 Konolige's deduction structures A first example of a nonnormal logic is the logic proposed by Konolige [13]. Konolige associates with each agent a so-called *deduction structure*, consisting of a set of sentences B in propositional logic and a set of inference rules R. B represents the set of basic beliefs of the agent, whereas R is the set of inference rules that the agent is able or willing to apply. An agent is then assumed to believe p if p can be derived from B using the inference rules in R. Since R is not necessarily complete with respect to the logic in which the sentences of B are expressed, an agent does not always believe all the logical consequences of its base beliefs.

Konolige's approach deals with only one source of bounded rationality, namely a lack of inference rules. But other serious problems remain (clearly it is not the case that the *only* source of bounded rationality is a lack of inference rules, since all logicians are able to apply a basic set of rules in FOPC, yet are still only boundedly rational). In particular, the beliefs of an agent are still closed under deduction. It is simply that deduction now depends on the set of inference rules in an agent's deduction structure rather than on the logic in which its base beliefs are expressed. For example, it is just as objectionable to claim that an agent is omnidoxastic with respect to intuitionistic logic as it is to claim that it is omnidoxastic with respect to classical logic.[5] Yet intuitionistic logic is a *bona fide* deduction structure in Konolige's sense, since it results from classical logic by the removal of the rule of double negation elimination.

2.3 Levesque's logic of implicit and explicit belief Another attempt at dealing with the problem of omnidoxasticity is Levesque's logic of implicit and explicit belief [14]. Levesque draws a distinction between *explicit beliefs*, defined as those beliefs actively held by the agent (so that the agent will answer "yes" if asked whether it believes them), and *implicit beliefs*, which logically follow from what the agent believes, whether it realizes it or not. Levesque argues that resource limitations, and hence the problem of omnidoxasticity, apply only to explicit belief (in other words, implicit beliefs are the beliefs of the ideal believer, whereas explicit beliefs are affected by resource limits and bounds of rationality). He therefore proposes to maintain the possible world analysis for implicit belief and proposes a new logic for explicit belief.

In his logic of explicit belief, Levesque uses the notion of a *situation* in Barwise and Perry's [3] sense. Intuitively, a situation is a partial possible world. That is, whereas a possible world supports the truth or falsity of every sentence in the language, a situation supports the truth or falsity of only some. Levesque then associates with each agent a set of situations, S, and says that an agent explicitly believes p if every situation in S supports the truth of p. This idea is similar to MacPherson's own solution [15], pp.19–26.

Levesque's logic of explicit belief clearly avoids the various omnidoxasticity problems because (i) a situation which supports the truth of, for example, p and $p \supset q$ need not necessarily support the truth of q, and (ii) a situation may support both the truth and falsity of some sentence without supporting the truth of all other sentences in the language. As a consequence, an agent need not believe all the logical conse-

quences of its beliefs, and it is not committed to believing all propositions when it is inconsistent.

Levesque's solution does have some intuitive strength behind it. However, its solution to the omnidoxasticity problem leaves other problems behind in its wake. For example, there is a question of the structure of situations. These partial possible worlds must support inference rules or not. If not then there are no inferences to be made with respect to explicit belief, and, in effect, the attempt to give a doxastic logic, however attenuated a logic it may be, has been given up.[6] This is worse even than Lemmon's pessimistic view of epistemic logic, on which the only conclusion that can be drawn from someone's knowing something is that that thing is true, since even that does not follow in the case of belief. If no inference rules are available at all, then any belief states are merely lists of propositions believed; but the whole point of a doxastic logic is that one could have some small handle on what else an agent believes given that it believes p, q, r, etc.. Even if it is philosophically correct to claim that no rules are available, it is a disappointing view for those who wish logic to provide some input for AI (though many in AI, such as Ballim and Wilks [2], are quite happy to do without inputs from logic).

On the other hand, if the partial possible worlds do support inference rules, then really all we have is a special case of a Konolige deduction structure, in which case the criticisms of that theory also apply. The agent would still be omnidoxastic with respect to the rules available.

2.4 Can the extent of omnidoxasticity be limited sufficiently to assuage doubts? One response that the supporters of both deduction structures and logics of explicit beliefs could make here is that omnidoxasticity might not matter too much if its extent were limited. After all, the major problems to be met are arguably the belief in (or knowledge of) all tautologies and the entailment by an inconsistency of every proposition. These are easily blocked, in the first place by restricting the rule of necessity and in the second place by being careful in the treatment of inconsistency. Having blocked those difficulties, what then needs to be watched is what is called "belief clutter" (cf. [10], p. 15). The logic needs to ensure that it keeps manageable the number of beliefs that the subject has.

One route to doing this is to make sure that only elimination rules are available in the partial possible worlds. For instance, MacPherson's own preferred system BEL can be seen as a description of belief in which the partial possible worlds support modus ponens (i.e., $\supset$-elimination), &-elimination, and do not allow sentences of the form $p \,\&\, \neg p$ (cf. [15], p. 22). So, although relative to this small set of rules, a believer would still be omnidoxastic, there are no problems with belief clutter. Because only elimination rules are allowed, a finite set of "core" beliefs could be expanded, through omnidoxasticity, only to another finite set. Every application of every rule reduces the degree of the new beliefs formed, and so it is not the case that infinitely many trivial beliefs could be deduced as it would be if introduction rules were available.

This may point the way to a solution for the truth-functional case.[7] And it might well be a very interesting philosophical result. But it does not solve the AI problem (in the general case); planning problems often involve substantially more than truth-

functional inferences. For example, Aitken et al. [1] discuss reasoning about beliefs, time (in particular, modeling the future through a branching model of time), and individuals. Even if we ignore the special problems set by the doxastic and temporal components of that particular theorem prover, we still need to address the problem of how to deal with quantification over individuals.

The problem is this: once the logical apparatus gets beyond simple truth-functions, the tactic of reducing the degree of the logical formulas by allowing elimination rules only in the partial possible worlds fails to prevent belief clutter. For example, suppose we had an axiom on which $\mathbf{BEL}_i(\forall x)P(x)$ is allowed to entail $\mathbf{BEL}_i P(a)$ for each a that i knows about (call this axiom **U**), so that, on this axiom, if an agent believed that everything was purple, then it would believe of everything it knew about that it was purple. In that event, the number of extra beliefs "created" as a result of omnidoxasticity with respect to this rule could be very large indeed, though finite. Suppose the agent was aware of the existence of n objects; then the addition of the belief that everything was purple would automatically lead to the addition of n beliefs (viz., to the effect that each of the n objects it recognized was purple). This number n might of course be very large, and so the belief clutter could well become very extensive.

Because there is no upper bound to n in this case, it is clear that the strategy of restricting all inference rules to elimination rules will not help with belief clutter for agents which recognize a large number of objects. Only in the truth functional case is the strategy going to pay dividends. Each application of &-elimination will result in two extra beliefs (i.e., if the agent believes that p & q, it will also believe that p and that q); each application of modus ponens will result in only one extra belief (i.e., if the agent believes that p and that $p \supset q$, it will also believe that q). But an application of **U** could result in orders of magnitude more beliefs.

Further, if a system included a function for creating new objects (e.g., the successor function for the natural numbers), then an infinity of new (nontrivial) beliefs could be created using **U**. Although the previous argument shows that indefinitely many new beliefs might be formed using the axiom **U** alone, the scale of the increase will always be limited by the number of objects that the agent knows about. But when the number of objects is not limited, neither is the number of beliefs that **U** will sanction.

For example, suppose that i believes that if a number is greater than 0, then so is its successor. That is not an unreasonable belief; most human agents with a rudimentary knowledge of arithmetic would assent to it. In our logical terminology, we say that $\mathbf{BEL}_i(\forall x)(x > 0 \supset s(x) > 0)$. Suppose also that i believes that 1 is greater than 0 (i.e., $\mathbf{BEL}_i(1 > 0)$). Now we are on the slippery slope to an infinite chain of beliefs, thanks to MacPherson's axiom **AS2** and our own **U**. **AS2**, translated into our own terminology is $\mathbf{BEL}_i p$ & $\mathbf{BEL}_i(p \supset q) \supset \mathbf{BEL}_i q$.

The proof is simple: since we have $\mathbf{BEL}_i(\forall x)(x > 0 \supset s(x) > 0)$, we have, by **U**, $\mathbf{BEL}_i 1 > 0 \supset s(1) > 0$. We also have $\mathbf{BEL}_i(1 > 0)$. Hence, by **AS2**, we have $\mathbf{BEL}_i(s(1) > 0)$. Until now, our agent was aware of two objects only, 0 and 1. Now it is aware of another, $s(1)$ (i.e., 2). So **U** can be applied again, since $\mathbf{BEL}_i(\forall x)(x > 0 \supset s(x) > 0)$, to give us $\mathbf{BEL}_i(s(1) > 0 \supset s(s(1)) > 0)$, and **AS2** can be applied, since $\mathbf{BEL}_i(s(1) > 0)$, to give us that $\mathbf{BEL}_i(s(s(1)) > 0)$. This process of applications of

U followed by **AS2** will give us an infinite series of beliefs $1 > 0$, $s(1) > 0$, $s(s(1)) > 0$, $s(s(s(1))) > 0$, $s(s(s(s(1)))) > 0, \ldots$, together with the corresponding (infinite) series of conditionals of the form $(s^n(1) > 0 \supset s^{n+1}(1) > 0) : s \geq 0$.

The alternative to such infinitary belief clutter, of course, is to avoid using an axiom such as **U**. But this will restrict severely the scope of the logic. For it is not clear what the point is of allowing such sentences as $\mathbf{BEL}_i(\forall x)P(x)$ into the logic if no conclusions are to be drawn from this with respect to i's beliefs about the individuals over which it is reasoning. If an agent believes that everything is a P, and yet treats some individual as if it were not a P, then it is not clear that the logic is not equivocating about the semantics of the quantifier (i.e., one's implicit beliefs would be classical, and one's explicit beliefs would be deviant), which MacPherson strongly (and rightly) objects to (cf. [15], pp. 16–7).

3 Taking control into account So, having reviewed some likely candidates for doxastic logics from AI, we can endorse MacPherson's result that there seems to be, at present, a singular lack of nonnormal modal logics which can model belief. Even the apparently promising tack of using partial possible worlds, favored by MacPherson, has important problems with it. Now, neither our survey nor MacPherson's pretends to be exhaustive, but it is yet to be demonstrated *either* that (a) *all* the obstacles can be surmounted without creating new problems elsewhere (the minimum result which would satisfy MacPherson), *or* that (b) such a logic could be embodied in a machine that could work in or close to real time (the extra result required for AI applications). Our aim in this paper is to suggest a new way of conceptualizing implicit and explicit belief (or ideal and fallible believers) to sidestep the problems with doxastic logics.

3.1 The sources of fallibility To begin with, consider why it is that an agent does not believe all the logical consequences of its beliefs. After all, all things being equal, the survival value of believing the consequences of one's beliefs is likely to be high. Following Fagin and Halpern [8], we might identify four reasons why real agents (both natural and artificial) fail to be omnidoxastic.

Lack of awareness: the agent may be unaware of some concept or object and therefore cannot have any beliefs about it.

Resource boundedness: the agent does not have the computational resources to derive all the logical consequences of its beliefs. Or, even if it did, it does not have enough memory to store them all. Or, even if it did, it could not access an arbitrary belief from its knowledge base[8] in real time due to the cost of searching such a vast store.

Lack of inference rules: the agent may not be able to apply (perhaps through lack of awareness) certain inference rules (for example $\vee$-elimination).

Limited focus of attention: the agent may fail to put together sentences that were derived in different contexts. The agent's reasoning might well be locally perfect, but the limited focus of attention may not enable it to make connections with other inferentially relevant beliefs which are currently out of focus.

Note that all these problems are problems with the agent itself. Hence the sources of fallibility are not logical but psychological. When an agent departs from

the beliefs of the ideal believer, it is not because its "logic" is nonnormal but because it is incapable of deriving the best results.

Suppose an agent were also a bit of a metatheorist and observed and commented on its own beliefs. Suppose also that it believed that p and that q, and that it believed that it believed that p and it believed that it believed that q. It is clear that it would not reason like this: *hmm, p is true (I believe), and so is q, but the logic governing my beliefs is nonnormal, and in particular does not contain the adjunction schema, so I should not believe that* $p \,\&\, q$. It *would* reason like this: *p is true (I believe), and so is q, and therefore* $p \,\&\, q$ *is true, so I will add that to my set of beliefs*. The conjunction is added to the agent's set of beliefs because it performed an inference. Any doxastic logic that denied the agent access to the inference would be failing to model belief properly. The important point is that agents sometimes (but only sometimes) fail to perform inferences; this is a psychological fact, not a point of logic.

3.2 Control heuristics in theorem provers The question now is: how can such psychological facts be modeled? As we have discovered, this may be tricky using logic. But in AI, what we do have are systems whose limitations might well be congruent to those of human agents. Normally this is not something to boast of, but in this case, fallibility may well be of philosophical interest.

We will attempt to show how a particular type of system can be used to model the limitations of resource-bounded agents. A *theorem prover* for a logic is a system that attempts to find proofs automatically for statements using that logic. This process turns out to be a process of *searching*; a theorem prover for a logic is a program that searches the space of possible proofs in that logic. Because of the recursive nature of many inference rules, it is in general impossible to search this space exhaustively. Moreover, even when an exhaustive search is in principle possible, practical considerations tend to rule it out (e.g., because of the need to produce results in real time). Therefore, a theorem prover will need to use a number of *heuristics* to determine the way in which it searches the space of possible proofs. We shall discuss these various heuristics, using Prolog (as in Clocksin and Mellish [5]) to illustrate them where appropriate.

For those readers who are unfamiliar with theorem-proving techniques in AI in general, and with Prolog in particular, it might be helpful to review some the principles of Prolog briefly, just so the flavor of automatic theorem proving is sampled.[9] Readers who are more *au fait* with theorem proving could skip the next few paragraphs.

Prolog is a logic-based programming language, widely used within the AI community, that can determine whether or not a particular proposition (classically) follows from a set of propositions. The basic type of expression in Prolog is a predicate expression, which consists of a predicate name followed by its arguments enclosed in parentheses. So, examples of Prolog predicate expressions would be `president(clinton)` or `brother_of(scott,virgil)`.

More complex expressions can be formed in Prolog by arranging them in the form:

```
p :- p1,p2,...,pn.
```

The expression before the '`:-`' symbol is the *head*, and the `pi` are the *subgoals*. The commas separating the subgoals are conjunctive. The full stop completes the expression. The expression is interpreted as saying that *p if* p_1 *and* p_2 *and* … *and* p_n. If a predicate expression appears followed by a full stop (i.e., as a head without any subgoals), as in

```
brother_of(scott,virgil).
```

then that is interpreted as simply asserting that the expression is true (i.e., it is unconditionally the case and does not depend on any subgoals).

Rules governing the use of expressions can be set up using the Prolog syntax. For example, a bachelor can be defined as an unmarried man.

```
bachelor(X) :- unmarried(X), male(X).
```

A single man can be defined as either unmarried or a divorcee or a widower. This gives us three rules.

```
single(X) :- unmarried(X).
single(X) :- married(X,Y),divorced(X,Y).
single(X) :- married(X,Y),deceased(Y).
```

To use Prolog, you send a *query*. Prolog will then search the *predicate database* which contains all the rules and facts that have been asserted (this is a form of knowledge base). A query is a rule without a head, such as

```
:- single(scott).
```

Prolog will then try to prove the expression in the query from the expressions in its predicate database. It will look for an expression whose head matches the query. In the simplest case, this would just be the fact that Scott was single.

```
single(scott).
```

Having matched these up, Prolog could assert that the query was proven. That case is simple enough, but it hardly counts as theorem proving. Now suppose that there is no simple fact corresponding to the query. Then the query might be matched to one of our rules for the predicate `single`. In the first, there is a match if `scott` is substituted for the variable `X` through the expression. Given there is a match to the rule's head, Prolog then sets up subgoals, which are to prove each statement on the right hand side of the rule (with suitable substitutions made). So, in the first instance, Prolog will set up a sub-query to prove `unmarried(scott)`, which it will attempt to prove in the same way. If it fails to prove it, it will remove it from its list of queries, and go to the next option, which is to prove that `married(scott,Y)` and then `divorced(scott,Y)`. It will attempt to prove each in turn. It will try to match each query against the heads of all the expressions in its predicate database, substituting for `Y` (consistently through the two queries) where appropriate. If it fails to prove either expression, it will then go on to the third possibility, trying to show that `married(scott,Y)` and `deceased(Y)`.

Having briefly reviewed the operation of Prolog, we can go on to observe that it is clear from this description that one source of problems for theorem proving is the fact that a theorem prover has to search through a knowledge base, such as Prolog's

predicate database, which may be large and unwieldy. This can make the process of proving queries very time (and memory) consuming indeed. The process may even go around in circles (and thereby enter an infinite loop). Imagine if the following rule had been asserted and was placed in such a way as to be the first expression used for matching purposes.

```
married(X,Y) :- married(Y,X).
```

In that event, when Prolog matched its query `married(scott,Y)` against the head of the rule, it would then set up a subgoal `married(Y,scott)`. But this would also then match against the head of the rule, since `scott` could match against the variable `Y`, and `Y` would match against `X`, since both are variables (one can think of Prolog rule expressions as being effectively universally quantified). This would then set up the next subgoal, which would be `married(scott,Y)`, and we would be back where we started. This loop could go on until the machine ran out of resources.

Even assuming such infinite loops are avoided, it could always happen that a query sparks off a series of subgoals, which eventually, after much computation, lead to a dead end. In this case, the machine would have to go back to the beginning, and all that time and effort would have been wasted — with no guarantee that the next trail that Prolog pursued would be any more successful.

The set of decisions that a theorem prover like Prolog can take can be arranged in a tree-like structure. At each point, or *node*, a number of possible actions are available, and each action determines a different *branch* of the tree (in the case of Prolog, there is a branch of the tree corresponding to each expression whose head matches the query). Down each branch, a different set of actions is available; these actions each then determine their own sub-branches, and so on. This branching structure is called a *search tree*. A search tree can be seen as a representation of the space of possible proofs and attempted proofs of the query in the knowledge base.

So, using our example above, the root node of the tree corresponds to the asking of the query `single(scott)`. There are then four branches from that node in the search tree, one corresponding to the assertion in the predicate database and one each corresponding to the three rules for `single`, and each branch leads to a new node. The first branch leads to a successful proof (a *terminal node*), the second leads to a node corresponding to a new query, `unmarried(scott)`, and so on. Each new query will set off its own subtree.

But the search tree is only half the story. The branches leading from a node in the tree determine what the system *can* do, but not what the system *will* do. The issue of what a machine does at each point is called the issue of *control*. The flow of control, in a computational system, defines what the machine is going to do. In the case of a theorem prover, the major control issue is to determine how it will navigate through the search tree. Whenever a node with a set of branches is reached, the theorem prover has to decide which branch to go down. If the branch then leads down a long trail which appears to have no end, another control issue is to decide when to stop and go back to the beginning, trying another branch. Clearly, as one can see from the discussion above, the order in which such a system performs its actions can have a large effect on how long a derivation of a result can take. Indeed, the order in which actions are performed can affect whether a result is produced at all. Where possible,

the control in a system will ensure success. But often it is the case that there is no sequence of actions which will work in all cases. In these circumstances, the system is programmed with control *heuristics*, which determine a strategy for selecting the next action to be performed.

Four types of heuristic can be distinguished that are used by a logic-based theorem prover:

1. heuristics to decide which propositions to retrieve from the knowledge base;
2. heuristics to generate the next layer in the search tree;
3. heuristics to decide which open node in the search tree to expand; and
4. heuristics to decide whether a branch should be pruned or not.

These control heuristics define what the theorem prover is intended to do, and when (i.e., which propositions it should prove, how, and in what order). In the remainder of this section, we explain in detail the effects of each of these types of heuristic.

First, in general, such a theorem prover will contain a number of inference rules, such as modus ponens. The rule will be stored in a form which looks something like: IF ((ASSERTED? P) AND (ASSERTED? (=> PQ))) THEN ASSERT Q, where P and Q are metavariables for propositions. Therefore, before an inference rule can be applied, the metavariables first have to be instantiated with actual propositions in the knowledge base. This is usually done by searching for and retrieving suitable formulas. The first heuristic that logic-based theorem provers use is a heuristic to decide the order in which formulas should be retrieved.

The heuristic that Prolog uses for retrieving propositions from the knowledge base is simply textual order. Propositions are retrieved in the order in which they were asserted into the knowledge base. However, one can imagine more sophisticated heuristics. For example, Socrates (cf. Corlett et al. [6]) contains a partitioning mechanism that allows the user to partition the knowledge base and thus restrict retrieval of propositions to only certain parts of the knowledge base. Hence, to extend our example above, one could make a partition with all the propositions to do with the marital status of the people in question, and then when a query like `single(scott)` comes in, restrict the search to that partition (or at least promote efficiency by searching that partition first). Or alternatively, all the propositions mentioning `scott` might be partitioned together, and that partition would then be searched first. All the general propositions, with variables like `X` and `Y` in them, would be in a separate partition, and that partition might be searched next.

As we have seen, a theorem prover's proof can be seen as a traversal of a search tree. At any node in the tree, the theorem prover can apply any one of its repertoire of inference rules; the application of each rule will open up a new branch. The second class of heuristics then concerns the operators that are used to build the next layer of the search tree. Given that one has decided to expand a particular node of the tree, one has to decide how to expand it.

The third class of heuristics concerns the decision as to which node to expand next. At any given point, more than one node may remain unexplored, and a theorem prover has to decide where to go next.

Again, the heuristics that Prolog uses for deciding which node to expand next are relatively simple: left-to-right, depth-first (i.e., choose the leftmost unexplored

node, and go down a level where possible). Other systems have more sophisticated heuristics. Indeed, many systems allow the user to define explicitly the heuristics that they want to use (e.g., MRS of Genesereth et al. [9] and Socrates of Corlett et al. [6]). This would allow, for instance, the user to apply a breadth-first heuristic (i.e., never go down a level in the tree unless all the nodes at higher levels are expanded), which can be slower than depth-first search but is guaranteed to find the shortest proof.

The final class of heuristics in theorem provers is used to prune the search tree—i.e., to make the decision that a given node looks so bad that it should not be considered for further expansion. This is of particular importance for real-time theorem proving. Indeed, since most interesting logics are semi-decidable, it is impossible in general to perform an exhaustive search for a proof of a proposition. Heuristics are therefore necessary to decide whether the current node is a suitable candidate for expansion or should be considered a dead end.

The heuristics that Prolog uses for deciding to prune a branch are very weak. Prolog will prune a branch as a dead end only if there are no clauses in the knowledge base against which the current query can be matched. In all other cases, Prolog will continue until it runs out of memory space. As a more sophisticated example, the UT theorem prover of Bledsoe [4] prunes a branch from the search tree if the same goal occurs more than once in the same branch. The rationale is that if in trying to prove some goal g, we reduce g to itself, we are never going to find a proof for g along the current path. Recall our example where the rule `married(X,Y) :- married(Y,X)` caused an infinite loop; this heuristic would rule that out, since once `married(scott,Y)` appeared for the second time as a query, the whole branch of the tree would be pruned (i.e., this query would be withdrawn), and another route would be tried.

3.3 Avoiding omnidoxasticity with control So, what does all this mean in practice? The idea is that modeling belief should be a two-stage process. Firstly, the structure of warranted belief has to be worked out as a logic. This logic needs to be inclusive, to take account of all the inferences a resource-bounded believer is justified in making. It does not need to be exclusive, and so it would not matter if some inferences were allowed that were psychologically unrealistic (as long as those inferences were warranted). In such a logic, an expression such as $\mathbf{BEL}_i\, p$ is interpreted as "i has a warrant to believe p." After this stage has been completed, the second stage is to develop a theorem prover for that logic whose control heuristics model the psychological fallibility of a human agent with respect to the logic. If an agent being modeled by this process fails to believe something it is warranted in believing, then the idea would be that that limitation (assuming it to be systematic) would be coded into the theorem prover. Then, if the theorem prover proves $\mathbf{BEL}_i\, p$, and inserts it into its knowledge base, this (act) is interpreted as meaning that i actually believes p.

There is a rough consensus that any logic of warranted belief must be normal (although this needs to be argued for). We do not particularly want to make a stand over this, although we think it is a reasonable assumption. It seems likely that if an agent has a warrant to believe p and a warrant to believe $p \supset q$, then it has a warrant to believe q; this would allow **K** into the logic of warranted belief. And similarly, an agent presumably has a warrant to believe any theorem of classical logic, which

would allow the rule of necessitation. These additions to the logic would make it normal.

The logic of warranted belief could allow further refinements that would be unavailable for a logic of actual belief. If an agent is warranted in believing p, then it is probably warranted in believing that it is warranted in believing p (i.e., $\mathbf{BEL}_i\, p \supset \mathbf{BEL}_i(\mathbf{BEL}_i\, p)$). i's warrant to believe p consists of some evidence for p that i possesses (this evidence is of course defeasible, since this is a logic of belief, not knowledge). Then it is arguable that i's possession of such evidence must be known to i in order for that evidence to warrant i's belief that p. Then i must be warranted in believing that that evidence is available to itself, from which it follows that i is warranted in believing that i has a warrant for belief in p.

We do not want to get into the fine details of a logic for warranted belief. But the main point is that such a logic would be substantially stronger than any of the solutions to the omnidoxasticity problem discussed in Sections 1 and 2 above. The question then is: will the modeling of belief by a theorem prover for this logic avoid omnidoxasticity?

When belief is interpreted as the theorem prover's coming out with the output asserting that a proposition is believed, the result is automatically that omnidoxasticity must be avoided, since any theorem prover can only output finitely many propositions. This will obviously rule out the possibility that belief will be deductively closed (assuming a classical propositional logic). Even if the machine has inserted $\mathbf{BEL}_i\, p$ into its knowledge base, it does not follow that it will ever get around to inserting $\mathbf{BEL}_i\, p \vee q$. It need not insert $\mathbf{BEL}_i\mathbf{BEL}_i\, p$. It may or may not; it depends on the control heuristics. It certainly will not in every case. If the knowledge base is large, there will be many (almost certainly infinitely many) warranted beliefs derivable from it. This is where the dangers of omnidoxasticity arise, since logically there is no reason why, for any warranted belief, that belief should not be held. But an artificial agent is in no danger of deriving all those beliefs, since it is limited in time and memory, and therefore clearly will not be omnidoxastic. The beliefs derived by the theorem prover would all be warranted, since the underlying logic would be doxastic, but not all warranted beliefs would be derived. This, we believe, makes the solution already as good as the nonnormal doxastic logics discussed above.

In the same way, the other serious problem with normal doxastic logics, that any contradiction implies all other propositions, is also circumvented. A theorem prover may well not discover an inconsistency in its knowledge base, although it will withdraw one when one is found (in this respect its behavior will mirror that of a human agent). Most theorem provers contain some sort of consistency checking; checking each proposition for consistency with the knowledge base is generally quite expensive in time and memory, however, and usually this facility can be turned off. One particular way in which an inconsistency might remain undiscovered with a human agent occurs when the two inconsistent beliefs are not connected because they are associated with different contexts. This can be modeled by a partitioned knowledge base, where the propositions in such a knowledge base are separated from each other; when two inconsistent propositions are kept in separate partitions, it might be the case that the inconsistency goes undiscovered for some time.

Given that the two basic problems of omnidoxasticity are circumvented by the

use of a theorem prover, the question arises as to how good a model of an actual believer it would be. In other words, the additional problem is that not only does a human agent not believe everything, what it believes is circumscribed in particular ways. This is the area where a theorem prover is likely to score over nonnormal doxastic logics. Nonnormal logics tend to rule out particular types of inference. But in general, most inference types are used by human agents; even quite complex inferences like *modus tollendo ponens* $((A \vee B) \supset (\neg B \supset A))$ or *reductio ad absurdum* are used fairly frequently in relatively unsophisticated contexts. Human agents tend to fail to draw all the conclusions that they might; it is not that they are omnidoxastic with respect to particular restricted sets of inference rules. Hence the solutions that nonnormal doxastic logics provide for omnidoxasticity are quite arbitrary and *ad hoc*.

On the other hand, theorem provers can be biased in ways similar to the ways in which human agents can be biased. In Section 3.1, we followed Fagin and Halpern in distinguishing four different sources of doxastic fallibility. In the remainder of this section, we will show how these different sources correspond directly to various properties essential to any theorem prover implementing a logic sufficient to capture doxastic reasoning.

The first source is lack of awareness. This is the situation where an agent may not be aware of a concept or object. In this case, it should not be held to be entertaining, for example, any of the tautologies involving that concept or object. Even though it is a tautology that either Mario Andretti can fly a helicopter or Mario Andretti cannot fly a helicopter, it would be deeply counterintuitive to assume that Perkin Warbeck believed that. We would propose that this limitation could simply be modeled as a limitation in the language available for the theorem prover. A system that cannot reason about an expression because it has not been *declared* cannot be said to believe any sentences involving such expressions.

The second source is resource-boundedness. An agent does not have the computational resources to derive all the logical consequences of its beliefs. Recasting the problem in terms of theorem proving we can say that the problem is what to do given that infinite resources of memory and time are not available to the agent. Resource-boundedness is addressed in theorem provers via the heuristics used to decide which node to expand next and which branches in the search tree to prune. Deciding which states are interesting, or likely to be fruitful, is a way of coping with the fact that it is not feasible to search the entire tree. The heuristics to prune the search tree could be altered to take account of likely strategies used by human agents; in this way, the ways that human resource-boundedness manifests itself can be mimicked by the artificial agent.

The third source is lack of inference rules. We can model this problem by noticing the fact that any theorem prover must embody heuristics that allow it to generate the next layer of the search tree. Such operations can be seen as the application of inference rules in our doxastic application. If the heuristics apply only certain operations then we will see behavior which amounts to a lack of inference rules. This offers a natural way of implementing Konolige's deduction structures and MacPherson's partial possible worlds semantics.

The final source is limited focus of attention. Often agents do not use all the beliefs that they have which are relevant to a given query. Such behavior is reflected in

the heuristic that is used to determine which propositions to retrieve from the knowledge base. This heuristic will attempt to retrieve only those propositions which are likely to be relevant, such as all those propositions that mention terms that are used in the query explicitly. However, such a heuristic is not guaranteed to retrieve all relevant propositions, and therefore we are likely to get behavior that mimics limited focus of attention.

It is an open question how far all the psychological strategies for coping with resource-boundedness used by human agents can be modeled in this way. However, the advantage that theorem provers have over nonnormal doxastic logics is that they allow the logic of belief to be very simple, while allowing the complex reasons for the lack of omnidoxasticity to be modeled. As more information is gathered from psychology about human reasoning practices, that information can be incorporated into an operational model built around a theorem prover merely by altering its priorities; the changes required by a logic would be much more complex. A logic, being a bounded and abstract object, is better suited to modeling relatively well-behaved structures (such as warranted belief). The unpredictable types of human behavior would better be modeled by machine behavior that is not easy to predict either. A human agent might believe p and $p \supset q$ and therefore believe q on Monday, while on Thursday no longer believe q, despite still believing that p and $p \supset q$. It is hard to imagine how that sort of capricious behaviour could be modeled by a logic at all, yet it is at the heart of the problem with belief. On the other hand, there is at least a chance of using the machine and the context in which the machine is used to model the nonlogical conditions governing Thursday's failure and Monday's success.

4 Discussion: belief, control, and omnidoxasticity Effectively, then, our claim is that the distinction between fallible and ideal believers can be modeled by a simpler method than the traditional solution of exploiting the distinction between nonnormal and normal modal logics. Our claim is that, if ideal believers are correctly modeled using a normal modal logic, then fallible believers can be modeled using a theorem prover for that logic. The relevant distinction would then be between a logic and a finite embodiment of a logic.

Note how the distinction can be of relevance to both types of AI we mentioned above—recall from Note 3 how AI can be seen as a branch of cognitive science or as a branch of engineering. If AI is an engineering discipline, then the aim is to get good results in real time without using too many computational resources. Then using a (relatively simple and well-understood) normal logic for modeling belief must be a saving over systems that try to use complex nonnormal systems. On the other hand, if AI is a psychological discipline, then the heuristics discussed in Section 3.2 can be used specifically to model the limitations mentioned in Section 3.1. For instance, the heuristic governing the pruning of the theorem prover's search tree, for example, could be based on the investigation and discovery of exactly when a human agent fails to apply a particular rule, which is uncontroversially a matter of *psychological* investigation. Although we make no suggestion that existing theorem provers model fallible human believers exactly, there would seem to be no reason why the heuristics governing a theorem prover might not be made more psychologically realistic in this way. Indeed, if a strong model of human belief is required, then the control heuristics

for the theorem prover could even reflect flawed reasoning patterns, such as incorrect reasoning with probabilities (see Kahneman et al. [12]), or biases in expert reasoning (see Silverman [18]).

However, even if our solution were deemed to be adequate for the engineering discipline of AI or the empirical discipline of cognitive science, it might be objected that the solution is unlikely to be adequate as an *analysis* of belief. The distinction between the logic and the theorem prover cannot be an analysis or explication of the concept of belief, this objection would run, because of the following dilemma. Either the theorem prover (which embodies a normal modal logic and control heuristics which are psychologically realistic to the degree required) always does the same thing or it does not. In the first case, the theorem prover in effect is merely a representation of a nonnormal logic; the set of propositions that the theorem prover will prove will define a nonnormal doxastic logic (although this may be a logic with very nontrivial inference rules). In such a logic, '$\vdash$' would be equivalent to 'provable with the theorem prover'. Hence the use of control heuristics to model resource-bounded belief is merely a short cut to the correct analysis, not the analysis itself. In the second case, where the theorem prover is inherently unpredictable, that very unpredictability means that we have no stable analysis of the concept of belief.

In fact, the first case, where the theorem prover always does the same thing, and therefore will effectively specify a nonnormal logic, is relatively improbable. A theorem prover is likely to have various dynamic properties, as will be seen in the discussion of the second case below. These dynamic properties would tend to make it unlikely that theorem provers are guaranteed to function in the same way across computational contexts, so a criticism of our approach based on the hypothesis that the first case obtained would lack force on that ground. Furthermore, it is not clear that the claims we have made for the philosophico-logical interest of theorem provers in doxastic logics would be inconsistent with the criticism. After all, we could certainly insist that the theorem prover would be an important tool for the investigation of the concept of belief if it was equivalent to the logic; indeed, it might be the *only* tool available for such an investigation, since such a circumscribed logic as is envisaged here would lack generally applicable and easily formulated rules of inference. One might even go further and claim that without extensive tests on such a theorem prover, the logic would almost certainly never have been discovered in the first place.

The second case we have to look at is the case where the psychologically realistic theorem prover does not specify a nonnormal logic. The first point to note is that we should resist claims that a computational system could not be *explanatory* of belief simply because it is a piece of hardware, and that therefore the analysis of the concept of belief would not have been advanced. For instance, the distinction between the *logic* of belief (the logic of ideal or implicit belief) and the *psychology* of belief (the beliefs that individuals actually hold) is a perfectly respectable one, and many philosophers have held that which beliefs *actually* get held by resource-limited agents is a matter of psychology and not entirely determined by the concept of belief itself. Logic determines which beliefs are *warranted*; psychology determine which beliefs are held.

Indeed, to claim that an analysis of belief could not use input from psychology (or any other empirical discipline, for that matter), is an over-strong claim. For in-

stance, a psychologically realistic theorem prover for a normal modal logic might be used to disprove the claim that that logic was a good doxastic logic for an *ideal* believer. For if the heuristics that the theorem prover used were *bona fide* psychologically and the output of the theorem prover did not respect fairly basic intuitions about what fallible believers are likely to believe, this might be taken as evidence of a failure of the embodied logic to define what the *ideal* believer would believe. It might even be said that that it is not obvious how else one could discover the properties of *ideal* belief.

So we should resist claims that a functioning computational system *per se* must fail to explicate an abstract concept. What we have to do finally is to explain why the output of a theorem prover may vary from context to context. After all, computers are supposedly deterministic machines; how could such a thing happen? There are at least three types of circumstance in which such variance could occur, and we will argue that each circumstance is analogous to similar sources of variation in natural belief.

To begin with, a theorem prover, like any computational system, can do only whatever is permitted at any time (i.e., whatever is consistent with its specification). If it turns out that nothing is permitted, the program will stop. If only one thing is permitted, then that action is performed. But there will be a problem of control if there is more than one permissible action. In that event, the class of actions that the system could perform is called a *conflict set*. The system must then choose between the items in the conflict set; the strategy it uses to make this choice is called a *conflict resolution strategy*. In the case of a theorem prover, one type of situation in which a conflict resolution strategy is likely to be needed occurs when it is ready to apply an inference rule. There may be one or more inference rules applicable, or alternatively there may be more than one formula to which a rule will apply. The theorem prover then has to choose one particular action to perform, and, as discussed in Section 3.2, it will have heuristics to make that choice. But note that the heuristic which determines the conflict resolution strategy need not be deterministic. For example, the strategy may be to use a randomizing function to choose the action to perform next, in the event that there is a conflict. Or, more realistically, it may be dependent on other concerns not directly related to the logical issues (e.g., the ease of performing the action, or the order in which the members of the conflict set are listed).

Thus the theorem prover may in different circumstances resolve conflicts differently, which means that given the same premises, it need not always give the same output, since some ways of resolving conflicts could lead to an efficient solution, while others may simply cause the system to run out of time. Indeed, if the system's knowledge base were actually inconsistent, different ways of resolving conflicts could lead to inconsistent solutions being reached in different circumstances to the same problem. This would certainly be of interest if it turned out that natural believers were similarly variant in the ways in which they approached problems. But even if not, such differing conflict resolution strategies could still supply results of philosophical, psychological, and logical interest. If it turned out that a theorem prover which was psychologically realistic in the appropriate ways could always reach the same result even with, say, a random conflict resolution strategy, then that is suggestive; that is good evidence (given the psychological realism of the systems in question) that a

natural believer should also achieve such results, come what may. On the other hand, if the achievement of a result varied with the conflict resolution, that is good evidence that a natural believer might not be expected to reach the result in all circumstances. In other words, a theorem prover might be useful in discovering whether the order in which things are done is significant. And if the purpose of doxastic logics is to reason about real, resource-bounded believers (whether natural or artificial), then it is hard to see why that is not relevant.

A second type of circumstance where the output of a theorem prover might be expected to vary will depend on factors to do with memory. The efficiency of a theorem prover will be limited by the amount of memory that is free at any point; that in turn will depend on the type of machine that the system is being run on. The memory size itself may vary; furthermore, the quality of the machine's *garbage collection* (i.e., the way the machine routinely frees parts of its memory by erasing old data that are no longer in use) will also influence how much memory the system gets to use. The amount of available memory will obviously have an effect on whether the theorem prover manages to complete its reasoning. But it may also be the case that the actual results of a theorem prover's investigations will depend on the quantity of memory available, so that, for example, it will reach an approximate solution, followed by a series of refinements which lead to increasing accuracy. In that case, the number and quality of the refinements may also be determined by garbage collection. Hence theorem provers may be used as tools for the investigation of *human* memory limitations if it is shown that psychologically realistic heuristics require a certain amount of memory to work efficiently.

The third type of circumstance in which variance might be anticipated occurs when the order that the beliefs appear in the system's knowledge base varies. Doxastic logics make play with a notion of timelessness which would appear to be unrealistic for a fallible agent. If I come to believe something, there is a time at which I believe it, and a time in the past at which I did not.[10] However, in a (static) doxastic logic, this phenomenon is rather glossed over; all the statements of the logic must be made in what Hintikka [11], p. 7 (or Hintikka's friend, to be precise) called a "logically specious present." But beliefs might well interact; the order in which beliefs are acquired may be an important factor in judgments about, for example, which of an inconsistent set of beliefs to give up (a long-held belief might become embedded). Whether contradictions and inconsistencies are even noticed may be a function of an ordering effect. Such effects are modeled easily by a theorem prover (since a theorem prover is also operating in time, and order effects can be modeled in control heuristics), whereas a logic, being static, can model such effects only at the cost of a further set of complications to the axioms and the semantics.

So, to conclude, we have suggested that theorem provers embodying logics of ideal or implicit belief can, by an artful choice of their control heuristics, be used to model the explicit beliefs of fallible believers, thereby gaining in simplicity over static nonnormal modal logics, which are standardly used. We have argued that use of such heuristics in operational dynamic systems is preferable to the production of complex and arcane logics from the points of view of AI conceived as an engineering discipline, as a sub-discipline of cognitive psychology, and even as a contributor to the philosophy of mind.

Acknowledgments HR and NS would like to thank Joe Downs and Sean Wallis for a number of discussions on this topic. An early (unpublished) version of this paper was partly supported by SERC grants no GR/F 28618 and GR/F 35968.

NOTES

1. Suppose that $\mathbf{BEL}_i\, p$ and that q is a logical consequence of p ($p \vdash q$). Then $\vdash p \supset q$. So, by necessitation, $\vdash \mathbf{BEL}_i(p \supset q)$. So, by **K**, $\vdash \mathbf{BEL}_i\, p \supset \mathbf{BEL}_i q$. But, by assumption, $\mathbf{BEL}_i\, p$. Hence, on that assumption alone, $\mathbf{BEL}_i q$.
2. For an artificial system, a "belief" is a proposition stored in its knowledge base upon which it may act. Opinion is divided as to whether the beliefs of human agents can be conceptualized similarly. For example, Harman [10], pp.13–4 sees beliefs as being written in "mentalese" in the mind; in AI terms mentalese can be seen as the human knowledge representation language, and on this view human and artificial agents are not too disanalogous in principle at least. On the other hand, Dennett [7] emphasizes the role of interpretation in belief attribution. For the purposes of this paper, our account will be neutral between these two views; our claim is only that at the level of logic, the logics of natural belief and artificial "belief" can and should influence each other (which, it should be emphasized, is consistent with the idea that philosophical theories of belief are independent of AI theories—cf. Ballim and Wilks [2], p. 451). For this reason, for the rest of this paper, we shall drop the scare quotes around the term "belief," which we shall use to refer both to the natural and the artificial versions. The argument will not be affected. Similarly, we will not use quotes to distinguish between natural action and artificial "action."
3. This characterization is neutral between a view of AI as cognitive science, where psychological computational models of existing (natural) believers are to be built, and the view of AI as an engineering discipline, where the aim is to build working systems that conform to given specifications.
4. One example of this would be Kripke's Paderewski example, where someone has contradictory (but not disastrously so) beliefs about Paderewski (the musician) and Paderewski (the politician). Another example might be naïve set theory, whose inconsistencies become apparent only in relatively specialized circumstances, viz., those where sets might be members of themselves. One could certainly imagine there being decades between Frege's development of the theory and the discovery of the Russell set without any problem arising.
5. Though some would claim that omnidoxasticity with respect to intuitionistic logic is at least slightly more realistic psychologically since intuitionistic logic is premised on the finite abilities of agents to verify or prove statements.
6. Cf. [17], p. 99 on knowledge: "epistemic logic cannot concern itself with actual *occurrent* knowledge, nor with *dispositional* knowledge: these biographical and psychological approaches to knowledge simply lack a 'logic.' "
7. What concerns us here is the *general* structure of MacPherson's solution. We do not want to get into arguments about the particular axioms that MacPherson has chosen. His aim is not to model belief exactly using his axioms but to show that it is possible to develop a nonnormal modal logic which will avoid the problem of omnidoxasticity, and, for the propositional case, we do not wish to dissent from that.
8. The knowledge base is the portion of memory in which all the current beliefs of the theorem prover are stored. It is called a *knowledge* base as opposed to a *belief* base because AI makes no distinction between what a system *knows* and what it *believes*. See Reichgelt [16] for a review of the representation of knowledge (= beliefs) in AI systems.

9. We only skim the surface of Prolog in this account; it is a substantially more sophisticated system than would appear from our brief review. Automatic theorem proving is a process of searching for proofs, and what we want to show is that controlling the search process can lead to interesting behavior. To that end, we hope to give the flavor of proof as search to an audience which is relatively unfamiliar with that idea. We have therefore (to save space as much as any other reason) only described as much of Prolog as would serve that purpose. For a full account of Prolog, see [5].

10. Those who claim that belief attribution is not determinate are not, of course, committed to the claim that there is a determinate time at which the belief came to be believed. As before, we intend our account to be neutral between accounts of beliefs as determinate and more interpretative accounts. See Note 2.

REFERENCES

[1] Aitken, J. S., H. Reichgelt and N. Shadbolt, "Resolution theorem proving in reified modal logics," *Journal of Automated Reasoning*, vol. 12 (1994), pp. 103–129. Zbl 0810.03009 MR 95k:68195 2.4

[2] Ballim, A., and Y. Wilks, *Artificial Believers: The Ascription of Belief*, Lawrence Erlbaum Associates, Hillsdale, 1991. 2.3, 4

[3] Barwise, J., and J. Perry, *Situations and Attitudes*, M.I.T. Press, Cambridge, 1983. Zbl 0946.03007 MR 2001h:03051 2.3

[4] Bledsoe, W., *The UT Interactive Theorem Prover*, Technical Paper ATP-17B, University of Austin, 1983. 3.2

[5] Clocksin, W., and R. Mellish, *Programming in Prolog*, Springer-Verlag, Berlin, 1981. Zbl 0466.68009 3.2, 4

[6] Corlett, R., N. Davies, R. Khan, H. Reichgelt and F. van Harmelen, "The architecture of Socrates," pp. 135–164 in *Logic-Based Knowledge Representation*, edited by P. Jackson, H. Reichgelt, and F. van Harmelen, M.I.T. Press, Cambridge, 1989. 3.2, 3.2

[7] Dennett, D. C., *The Intentional Stance*, M.I.T. Press, Cambridge, 1987. 4

[8] Fagin, R., and R. Halpern, "Belief, awareness and limited reasoning," pp. 491–501 in *Proceedings of International Joint Conference on Artificial Intelligence 1985*, Morgan Kaufman, San Mateo, 1985. Zbl 0634.03013 MR 88k:03030 3.1

[9] Genesereth, M., R. Greiner and D. Smith, *MRS Manual*, Memo HPP-80-24, Stanford University, 1980. 3.2

[10] Harman, G., *Change in View*, M.I.T. Press, Cambridge, 1986. 2.4, 4

[11] Hintikka, J., *Knowledge and Belief: An Introduction to the Logic of the Two Notions*, Cornell University Press, Ithaca, 1962. 1, 4

[12] Kahneman, D., P. Slovic and A. Tversky, *Judgment Under Uncertainty: Heuristics and Biases*, Cambridge University Press, Cambridge, 1982. 4

[13] Konolige, K., *A Deduction Model of Belief*, Pitman, London, 1986. Zbl 0683.68080 MR 89i:03050 2.2

[14] Levesque, H., "A logic of implicit and explicit belief," pp. 198–202 in *Proceedings of AAAI-84*, Morgan Kaufman, San Mateo, 1984. 2.3

[15] MacPherson, B., "Is it possible that belief isn't necessary?" *Notre Dame Journal of Formal Logic*, vol. 34 (1993), pp. 12–28. Zbl 0800.03008 MR 94h:03041 1, 1, 2.3, 2.4, 2.4

[16] Reichgelt, H., *Knowledge Representation: An AI Perspective*, Ablex, Norwood, 1991. 4

[17] Rescher, N., *Studies in Modality*, Blackwell, Oxford, 1973. 1, 1, 4

[18] Silverman, B. G., "Critiquing human judgment via knowledge acquisition systems," *AI Magazine*, vol. 11 (1990), pp. 60–79. 4

[19] Williamson, T., "Some admissible rules in nonnormal modal systems," *Notre Dame Journal of Formal Logic*, vol. 34 (1993), pp. 378–400 Zbl 0803.03008 MR 94i:03056 1

Artificial Intelligence Group
Department of Psychology
University of Nottingham
University Park
Nottingham NG7 2RD
U.K.
email: koh@psyc.nott.ac.uk

Department of Computer Science
University of the West Indies
Mona, Kingston 7
Jamaica
email: han@uwimona.edu.jm

Artificial Intelligence Group
Department of Psychology
University of Nottingham
University Park
Nottingham NG7 2RD
U.K.
email: nrs@psyc.nott.ac.uk