# A DEDUCTIVE MODEL OF BELIEF

Kurt Konolige

Artificial Intelligence Center
SRI International

## Abstract

Representing and reasoning about the knowledge an agent (human or computer) must have to accomplish some task is becoming an increasingly important issue in artificial intelligence (AI) research. To reason about an agent's beliefs, an AI system must assume some formal model of those beliefs. An attractive candidate is the *Deductive Belief model:* an agent's beliefs are described as a set of sentences in some formal language (the base sentences), together with a deductive process for deriving consequences of those beliefs. In particular, a Deductive Belief model can account for the effect of resource limitations on deriving consequences of the base set: an agent need not believe all the logical consequences of his beliefs. In this paper we develop a belief model based on the notion of deduction, and contrast it with current AI formalisms for belief derived from Hintikka/Kripke possible-worlds semantics for knowledge.[1]

## 1. Introduction

As AI planning systems become more complex and are applied in more unrestricted domains that contain autonomous processes and planning agents, there are two problems (among others) that they must address. The first is to have an adequate model of the cognitive state of other agents. The second is to form plans under the constraint of resource limitations: i.e., an agent does not always have an infinite amount of time to sit and think of plans while the world changes under him; he must act. These two problems are obviously interlinked since, to have a realistic model of the cognitive states of other agents, who are presumably similar to himself, an agent must reason about the resource limitations they are subject to in reasoning about the world.

In this paper we address both problems with reference to AI planning system robots and one part of their cognitive state, namely beliefs. Our goal is to pursue what might be called *robot psychology:* to construct a plausible model of robot beliefs by examining robots' internal representations of the world. The strategy adopted is both descriptive and constructive. We examine a generic AI robot planning system (from now on we use the term agent) for commonsense domains, and isolate the subsystem that represents its beliefs. It is then possible to form an abstraction of the agent's beliefs, that is, a *model* of what the agent believes. This is the descriptive part of the research strategy. Among the most important properties of this model is an explicit representation of the deduction of the consequences of beliefs, and so we call the model one of *Deductive Belief.*

It is assumed that the beliefs of the agent are about conditions that obtain in the planning domain, *e.g.,* what (physical) objects there are, what properties they have, and what relations hold between them. Thus the descriptive model of Deductive Belief has an obvious shortcoming. Although agents can reason about the physical world, they don't have any method for reasoning about the beliefs of other agents (or their own). By taking the descriptive model to be the way in which agents view other agents' beliefs, we can construct a more complex model of belief that lets agents reason about others' beliefs. This is the constructive part of the research strategy.

There are two main sections to this paper. In the first, the concept of a *belief subsystem* is introduced, and its properties are defined by its relationship to the planning system as a whole. Here we discuss issues of deductive closure, completeness, and the resource limitations of the belief subsystem. We also characterize the constructive part of the model by showing how to expand a belief subsystem to reason about the beliefs of other agents. In the second section, we formalize the Deductive Belief model for the propositional case by introducing the belief logic B, and compare it with other formalizations of knowledge and belief. Because the treatment here must be necessarily brief, throughout the paper proofs established by the author, but not yet published, are referenced.

## 2. Deductive Belief

What is an appropriate model of belief for robot problem-solving systems reasoning about the world, which includes other robot problem-solving systems? In this section we discuss issues surrounding this question and propose a model of Deductive Belief as a suitable formal abstraction for this purpose.

### 2.1 Planning and Belief: Belief Subsystems

A robot planning system, such as STRIPS, must represent knowledge about the world in order to plan actions that affect the world. Of course it is not possible to represent all the complexity of the real world, so the planning system uses some abstraction of real-world properties that are important for its task, *e.g.,* it might assume that there are objects that can be

'This paper describes results from the author's dissertation research. The work presented here was supported by grant N0014-80-C-0296from the Office of Naval Research.

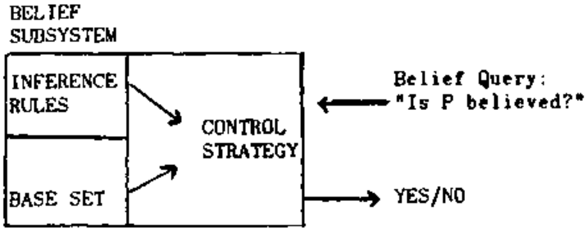

**Figure 1.** A Schematic Belief Subsystem

stacked on each other in simple ways (the *blocks-world* domain). It is helpful to view the representation and deduction of facts about the world as a separate subsystem within the planning system; we call it the *belief* subsystem. In its simplest, most abstract form, the belief subsystem comprises a list of sentences about a situation, together with a deductive process for deriving consequences of these sentences. It is integrated with other processes in the planning system, especially the *plan derivation process* that searches for sequences of actions to achieve a given goal.

In a highly schematic form, Figure 1 sketches the belief subsystem and its interaction modes with other processes of the planning system. The belief system is composed of the base *sentences,* together with the belief deductive process. Belief deduction itself can be decomposed into a set of *deduction rules,* and a *control strategy* that determines how the deduction rules are to be applied and where their outputs will go when requests are made to the belief subsystem.

There are two types of requests that result in some action in the belief subsystem. A process may request the subsystem to add or delete sentences in its base set; this happens, for example, when the plan derivation process decides what sentences hold in a new situation. Although this process of belief updating and revision is a complicated research problem in its own right, we do not address it here (see Doyle [I| for related research). The second type of request is a query as to whether a sentence is a belief or not. This query causes the control strategy to try to prove that the sentence is a consequence of the base set, using the deduction rules. It is this process of *belief querying* that we model in this paper.

We list here some further assumptions about belief subsystems. The internal language of a belief subsystem is a formal language, which must include a (modal) belief operator, *e.g.,* a propositional or first-order modal language would be appropriate. It is assumed that there is a Tarskian semantics for the language, that is, sentences of the language are either true or false of the real world. The belief subsystem doesn't *inherently* support the notion of uncertain beliefs, although this idea could be introduced if the internal language contained statements about uncertainty, *e.g.,* statements of the form P is true with *probability* 1/2.

The deduction rules of a belief subsystem are assumed to be sound (with respect to the semantics of the internal language), effectively computable, and to have bounded input. In particular, this forces deduction rules to be *monotonic.* It is our view that nonmonotonic or default reasoning should occur in the belief updating and revision process, rather than in querying beliefs.

The process of belief derivation is assumed to be *total.* This means that the answer to a query will be returned in a finite amount of time; i.e., the belief subsystem cannot simply sit and continue to perform deductions without returning an answer.

It is possible to define several types of consistency for beliefs. Deductive consistency requires that no sentence and its negation be simultaneous beliefs. *Logical consistency* requires that there be a world in which all the beliefs are true. Note that deductive consistency does not entail satisfiability, because the deductive process may not be complete. That is, a set of beliefs may be unsatisfiable and thus logically inconsistent, but, because of resource limitations, it may be impossible for an agent to derive a contradiction. Deductive consistency is the appropriate concept for belief subsystems. The assertion that rational agents are consistent is compatible with, but not required by, the model. It gives rise to a slightly different axiomatization (see Section 3).

The results of this paper depend only on the most general features of a belief subsystem as depicted in Figure 1: namely, that there is a formal internal language in which statements about the world are encoded; that there is a finite set of base beliefs in this language; and that there is some process of belief deduction that applies sound and effectively computable deduction rules to the base sentences at appropriate times, in response to requests by other processes in the planning system. A belief subsystem with these properties (along with the amplifications and restrictions given above) is a model of belief for planning agents, which we call *Deductive Belief.*

## 2.2 Resource Limitations and Deductive Cloture

One of the key properties of belief deduction that we wish to include is the effect of resource limitations. If an agent cannot deduce all the logical consequences of his beliefs, then we say that his deductive process is *incomplete.* Logical incompleteness arises from two sources: an agent's deduction rules may be too weak, or his control strategy may perform only a subset of the derivations possible with the deduction rules. Both these methods can be, and are, used by AI systems confronted with planning tasks under strict resource bounds. For several reasons, both conceptual and technical, we do not include incomplete control strategies in the Deductive Belief model. Instead, we make the following assumption:

CLOSURE PROPERTY. *The sentences derived in a belief subsystem are closed under its deduction rules.*

One advantage of requiring that beliefs be closed under deduction is conceptual clarity and predictability. If beliefs are not closed, then there is some control strategy that guides the deductive process, making decisions to perform or not to perform deductions. If this control strategy uses a global effort bound, then behavior of such a subsystem is hard to predict. Theoretically there may be a derivation of a sentence, but the control strategy in a particular case decides not to derive it, because it tried other derivations first. Closed systems, on the other hand, behave more dependably. They are guaranteed to arrive at all derivations possible with the given deduction rules.

The concept of "belief is also complicated by the introduction of control strategy issues. For example, it makes a difference to the control strategy as to whether a sentence is a member of the base set, or obtained at some point in a derivation. One cannot simply say, "Agent S believes *P** because such a statement doesn't give enough information about *P* to be useful. If *P* is derived at the very limit of deduction resources, then nothing will follow from it; if it is a base sentence, then it might have significant consequences.

In terms of formalizing the model of Deductive Belief, the assumption of closure is technically extremely useful. Consider the task of formalizing a belief subsystem that has a complex global control strategy guiding the deductive process. To do this correctly, one must write axioms that describe the agendas, proof trees, and other data structures used by the control strategy, and how the control process guides deduction rules operating on these structures. Reasoning about the deductive process involves making inferences using these axioms to *simulate* the deductive process, a highly inefficient procedure. By contrast, the assumption of closure leads to a simple formalization of belief subsystems that incorporates the belief deductive process in a direct way (the Deductive Belief logic, B, is presented in the next section). We have found complete proof techniques for B that involve running an agent's deductive system directly, in a manner similar to the semantic attachment methods of Weyhrauch [6].

Having argued that control strategies that use a *global* effort bound are undesirable, we now show that weak (but closed) deduction can have the same effect as control strategies with a *local* effort bound. We define a local bound as a restriction on the type of derivations allowed, without regard to other derivations in progress, i.e., all derivations of a certain sort are produced. An example of this sort of control strategy is *level-saturation* in resolution systems. Here we give a simpler example.

Suppose an agent uses *modus ponens* as his only deduction rule, and has a control strategy in which only derivations using fewer than *k* applications of this rule are computed; this is a local effort bound. To model this situation with a closed belief subsystem, consider transforming the base set so that each sentence has an extra conjunct tacked onto it, the predicate *DD(0)* *(DD* stands for "derivation depth"). Instead of *modus ponens,* the belief subsystem has the following modified deduction rule:

$$MP2: \quad \frac{DD(n)\wedge\alpha \qquad DD(m)\wedge(\alpha\supset\beta)}{DD(n+m+1)\wedge\beta}, \quad n+m\leq k$$

*MP2* is sound and effectively computable, so it is a valid deduction rule for a belief subsystem. The closure of the base set of sentences of the belief subsystem under *MP2* will be the same (modulo the *DD* predicate) as the set of sentences deduced by the nonclosed control strategy of the agent.

The Closure Property, together with the assumption of totality for the belief derivation process, imply that the deduction rules are *decidable* for all base sets of sentences.

### 2.3 Views

Up to this point, we have specifically assumed that agents don't have any deduction rules dealing with the beliefs of other agents. Now, however, we form the constructive part of the Deductive Belief model: adding to the belief subsystem model so that an agent can reason about its own and other belief subsystems.

We can arrive at deduction rules that apply to beliefs by noting that the obvious candidate for the intended interpretation of the belief operator is another *belief subsystem.* That is, the modal sentence *[S]α* is intended to mean "the sentence a is derivable in agent S's belief subsystem." The new deduction rules that apply to belief operators will be judged *sound* if they respect this intended interpretation. For example, suppose a deduction rule states that, from the premise sentences *\S]p* and *[S}(p>q),* the sentence *[S]q* can be concluded. This is a sound rule if *modus ponens* is believed to be a deduction rule of S's belief subsystem, since the presence of p and *pz>q* in a belief subsystem with *modus ponens* means that *q* will be derived.

We summarize by postulating the following property of Deductive Belief:

> RECURSION PROPERTY. *The intended model of the belief operator in tbe internal language of a belief subsystem is another belief subsystem. The intended model for an agent's own beliefs is his own belief subsystem.*

The Recursion Property of belief subsystems leaves a large amount of flexibility in representing nested beliefs. Each agent might have his own representational peculiariaties for other agents' beliefs. An agent John might believe that Sue has a set of deduction rules $R_1$, whereas he believes that Kim's rules are $\mathrm{R}_2$. In addition, John might believe that Sue believes that Kim's rules are $\mathrm{R}_3$. We call a belief subsystem as perceived through a chain of agents a view, and use the Greek letter *v* to symbolize it. For example, John's perception of Sue's perception of Kim's belief subsystem is the view *v* = *John, Sue, Kim.*

Obviously, some fairly complicated and confusing situations might be described with views, in which agents believe that other agents have belief subsystems of varying capabilities. Some of these scenarios would be useful in representing situations that are of interest to AI systems, *e.g.,* an expert system tutoring a novice in some domain would need a representation of the deductive capabilities of the novice that would initially be less powerful and complete than its own, and could be modified as the novice learned about the domain.

Having slated the Recursion Property, we now ask if there is a way to implement it within the confines of belief subsystems. At first glance it would seem so: suppose the agent *S* wishes to know whether he believes some statement p, *i.e.,* whether *[S]p* is one of his own beliefs. If we assume he can query his belief subsystem, he simply submits *p* to it; if it answers "yes," he believes *{S]p,* and if "no," then he believes -*[S]p.* Similarly, if he wishes to know whether another agent *S'* believes p, he simply queries a subsystem supplied with (his version of) *S'* deduction rules, and uses the answer to conclude either *[S]p* or -[S']p.

The problem with this strategy is that we haven't shown that *S* will receive an answer from the subsystems he queries. In the case of querying his own subsystem, there may be another occurrence of the modal operator *[S]* that will cause a recursive call to his belief subsystem, and so on in an unbounded manner. Although we assumed that the initial subsystem without the Recursion Property was decidable, we have not shown that this is also true for the expanded subsystem.

In the case of querying *S's* subsystem, S doesn't have the complete subsystem in hand, since he has incomplete knowledge of the base set. So, in effect, 5 must try to prove that, in each of *S"s* base sets that are consistent with S's beliefs, p is derivable. But even if we assume that individual subsystems that faithfully implement the Recursion Property are decidable, we haven't shown that the *theory* of a set of such subsystems is decidable, which is what is needed for *S* to receive an answer to [S']p.

We now give a formal interpretation of these issues. Let *6* be a belief subsystem for agent *S* characterized by a set of deduction rules R, and let *6(B)* be the set of sentences deduced by the belief subsystem from a base set *B.* We say that *6* is *decidable* if *6(B)* is decidable for all *B.* An extension *of 6* is a subsystem whose deduction rules are a superset of R. Now suppose $ is decidable, and consider the following questions:

1. *Is there an extension $\delta'$ of $\delta$ such that, for all base sets $B$ and all sentences $\alpha$,*
   a) *if $\alpha \in \delta'(B)$, then $[S]\alpha \in \delta'(B)$, and*
   b) *if $\alpha \not\in \delta'(B)$, then $\neg[S]\alpha \in \delta'(B)$ ?*
2. *Is $\delta'$ decidable?*
3. *Is the theory of $\delta'$ decidable?*

We have proven the following about these questions. In general, (1) must be answered negatively, as not all subsystems are extendable. There are specific types of subsystems for which extensions satisfying (1) exist, however *(e.g.,* if the base set contains no instances of the self-belief operator).[2] If an extension exists, it is decidable. But the *theory* of a decidable extension is not, in general, decidable; there exist counter-examples to (3).[3]

Even though a complete, decidable implementation of the Recursion Property does not exist in all cases, we can find incomplete approximations. The idea is that the undec id ability results from the unboundedness of belief recursion, that is, reasoning about an agent reasoning about an agent..., in an unbounded manner. Suppose, however, we place a bound on the depth of such reasoning: as the deductions involve higher embeddings of belief subsystems, the rules become weaker, and eventually the line of reasoning is cut off at some finite depth. Belief subsystems satisfying this property are said to have *Bounded Recursion.* Bounded Recursion subsystems are a nice example of resource limitations in belief deduction.

## 3. A Propositional Deductive Belief Logic

We present a logic, **B**, for Deductive Belief. For simplicity and ease of comparison with other modal systems, we assume a (modal) propositional internal language. The logic is capable of representing belief subsystems with or without the Bounded Recursion property. It is sound and complete with respect to these models.

### 3.1 Sequent Systems and Views

The general model of deduction that we assume is a block tableau sequent system. Block tableaux have much in common with PLANNER-type theorem-provers, and also have nice formal properties; the interested reader is referred to Smullyan [5]. The treatment here will be necessarily brief.

Let $S_i$ be a set of (names for) agents, and let $L$ be a modal propositional language, with unary modalities $[S_i]$ ($[S_i]\alpha$ means "agent $S_i$ believes $\alpha$"). Let capital Greek letters stand for finite sets of sentences of $L$ (lower Greek letters stand for single sentences). A *sequent* is an ordered pair of sets written as $\Gamma \Rightarrow \Delta$, and read as "$\Delta$ follows from $\Gamma$." A sequent $\Gamma \Rightarrow \Delta$ is true in a Boolean valuation *iff* the sentence $(\gamma_1 \wedge \gamma_2 \wedge \ldots) \supset (\delta_1 \vee \delta_2 \vee \ldots)$ is true.

A block tableau system $\mathcal{T}$ consists of a set of axioms and deduction rules. A sentence $\alpha$ is a *theorem* of a system $\mathcal{T}$ if there is a closed block tableau whose root is the sequent $\Rightarrow \alpha$. A set of sentences $\Gamma$ is *inconsistent* if there is a closed tableau for $\Gamma \Rightarrow$. We make use of a set of rules $\mathcal{T}_0$ that are propositionally complete, *i.e.*, all tautological consequences can be derived using $\mathcal{T}_0$ (see Smullyan [5]).

Although we have given a semantic treatment of sequents above, they have a natural interpretation in terms of derivability, and it is this interpretation we exploit to formalize belief subsystems. Consider the belief subsystem of agent $S_i$. This subsystem has a set of effectively computable deduction rules, represented as a block tableau system; call the set of rules $\tau(i)$. Suppose $\Gamma \Rightarrow_i \alpha$ is a theorem of this system (we use a subscript on the sequent sign to denote that this sequent refers to $S_i$'s belief subsystem). If all of $\Gamma$ are beliefs of $S_i$, then, by the Closure Property, $\alpha$ must be also, since $\alpha$ is deducible from $\Gamma$, according to $S_i$.

For any view $\nu$, we can thus characterize the belief subsystem by a set of tableau rules $\tau(\nu)$. These, together with tableau rules that relate sentences in one view to another, constitute the system **B**, which we give below (the following abbreviation is used: $[S]\Gamma =_{df} [S]\gamma_1, [S]\gamma_2, \ldots$).

$\Rightarrow$ *The propositional rules $\mathcal{T}_0$.*

$\Rightarrow_\nu$ *Rules $\tau(\nu)$ for each view $\nu$.*

*Cut*:* $$\frac{[S_i]\Pi, [S_i]\Gamma \Rightarrow_\nu [S_i]\alpha}{[S_i]\Pi \Rightarrow_\nu [S_i]\beta \qquad [S_i]\Gamma, [S_i]\beta \Rightarrow_\nu [S_i]\alpha}$$

*$B_\delta$:* $$\frac{\Sigma, [S_i]\Gamma \Rightarrow_\nu [S_i]\Delta, \Pi}{[S_i]\Gamma, \Gamma \Rightarrow_{\nu,i} \delta, [S_i]\Delta}$$

*Remarks.* There are three parts to the theory **B**. The first part is a set of rules formalizing the outside observer's

[2] Tbe work of Levesque [2] is helpful in finding classes of extendable systems.

[3] The proof of this uses Kripke's well-known result that monadic 55 is undecidable.

Boolean system. These rules incorporate the nonsubscripted sequent sign ($\Rightarrow$). A propositionally complete deductive system is employed here (*i.e.*, $\mathcal{T}_0$), since we are interested in deducing all we can concerning the belief systems of the agents. Properties of belief systems in general are always stated using the observer's sequent; for example, to show formally that, if any agent believes $p$, he believes he believes $p$, we prove that the sequent $[S_i]p \Rightarrow [S_i][S_i]p$ is a theorem of **B**. The second part is a set of rules formalizing the propositional component of each view ($\Rightarrow_\nu$). These rules involve the sequent sign $\Rightarrow_\nu$, since they talk about agents' deductive systems.

The third part of **B** formalizes the constructive component of the Deductive Belief model, characterizing the way in which the outside observer views agents' deductive systems, and the way in which agents view their own and other agents' systems. In general these rules have intermixed occurrences of sequent signs with different view indices. Rule *Cut** implements the Closure Property for each view, by saying that, if $\beta$ follows from believing $\Pi$, and $\alpha$ from $\beta$ (and possibly additional beliefs $\Gamma$), then $\beta$ follows from believing $\Pi$ (and $\Gamma$). Rule $B_5$ formalizes the deductive system of an agent from the point of view of an outside observer and other agents. The key part to this rule is that the modal operators get dropped from $[S_i]\Gamma$ and one of $[S_i]\Delta$ in going from the top to the bottom sequent. This part of the rule can be informally read as, "in any view, $S_i$ believing $\delta$ follows from his believing $\Gamma$ if, in that view of $S_i$'s deductive system, $\delta$ follows from $\Gamma$." When put this way, rule $B_5$ is simply a formal statement of the basic concept of deductive belief given in the last section.

There are two other interesting components to $B_5$: both $[S_i]\Delta$ and $[S_i]\Gamma$ get repeated on the bottom sequent. The reason for this is to capture the introspective properties of belief; namely, if an agent believes $\Gamma$, then he believes that he believes it, and if an agent doesn't believe $\Delta$, then he believes that he doesn't believe it. Note that $B_5$ is appropriate only if agents actually have complete knowledge of their own beliefs. As we indicated in the last section, this is not always even theoretically possible. In this case, weaker versions of $B_5$ must be used. If an agent doesn't know all the things he doesn't believe, then $[S_i]\Delta$ is dropped from the bottom sequent. If he doesn't know all the things he does believe, then $[S_i]\Gamma$ is dropped.

Finally, we need a separate rule to state a consistency condition on beliefs, if this is desired.

$$B_c \quad : \quad \frac{\Sigma, [S_i]\Gamma \Rightarrow_\nu \Pi}{[S_i]\Gamma \Rightarrow_\nu [S_i]\alpha \qquad [S_i]\Gamma \Rightarrow_\nu [S_i]\neg\alpha}$$

This rule states that $S_i$ believing $\Gamma$ is logically inconsistent, if $\Gamma$ is deductively inconsistent in $S_i$'s deductive system (recall that $\Pi \Rightarrow_i$ means that the $\Pi$ are logically inconsistent).

We have proven the system **B** to be both sound and complete with respect to the Deductive Belief model.

### 3.2 Comparison to Other Modal Systems

Most AI research on formal representations of belief and knowledge is based on Hintikka's adaptation of Kripke's *possible-worlds model* (*e.g.*, [4], [3]). Possible-worlds models, by their very nature, require that all logical consequences of an agent's beliefs are also beliefs; a possible-worlds model cannot take into account resource limitations that might be present in an agent's belief system. The propositional modal logic that formalizes the possible-worlds model of belief is weak 55, that is, 55 without the condition that all beliefs are true. We have proven that B reduces to this system under the following conditions:

J. The propositioned rules r(v)) for each view v are complete, and

2. Belief recursion is unbounded.

In addition, if a modified form of $B_5$ is used in which an agent doesn't know everything he doesn't believe, then under the same conditions B reduces to weak 54. Thus, under the assumption of deductive completeness and an infinite resource bound, the B reduces to more familiar belief logics.

## 4. Conclusion

We have introduced the concept of robot belief subsystems parameterized by a finite set of base sentences and a set of deduction rules. This Deductive Belief model is a viable alternative to possible-worlds models of belief and has the attractive property of taking resource limitations into account in deriving consequences of beliefs. We have formalized the Deductive Belief model for the propositional case with the logic B, which is sound and complete with respect to our model.

## References

[I) Doyle, J., "Truth Maintenance Systems for Problem Solving," Artificial Intelligence Laboratory Technical Report 419 , Massachusetts Institute of Technology, Cambridge, Massachusetts (1978).

[2] Levesque, H. J.., "A Formal Treatment of Incomplete Knowledge Bases," FLAIR Technical Report No. 614 , Fairchild, Palo Alto, California (1982).

[3] Moore, R. C, "Reasoning About Knowledge and Action," Artificial Intelligence Center Technical Note 191 , SRI International, Menlo Park, California (1980).

[4] Sato, M., A Study of Kripke-type Models for Some Modal Logics by Gentzen '$ Sequential Method, Research Institute for Mathematical Sciences, Kyoto University, Kyoto, Japan, July 1970.

[5] Smullyan, R. M., First-Order Logic, Springer-Verlag, New York, 1968.

[6] Weyhrauch, R., "Prolegomena to a Theory of Mechanized Formal Reasoning," Artificial Intelligence 13 (1980).

# KNOWING INTENSIONAL INDIVIDUALS,<br>AND REASONING ABOUT KNOWING INTENSIONAL INDIVIDUALS

Anthony S. Maida

Center for Cognitive Science
Box 1911, Brown University
Providence, Rhode Island 02912, USA

## ABSTRACT

This paper outlines an approach toward computationally investigating the processes involved in reasoning about the knowledge states of other cognitive agents. The approach is Fregean and is compared with the work of McCarthy and Creary. We describe how the formalism represents the knowing of intensional individuals, coreferentiality, iterated propositional attitudes, and we describe plans to test, the scheme in the domain of speech act recognition.

## I INTRODUCTION

Humans quite effectively reason about other humans' knowledge states, belief states, and states of wanting. Unfortunately, the processes by which humans do this are not well understood. This paper outlines an approach toward computationally investigating these processes. This approach involves two components, the first of which involves adequately representing knowledge about others' knowledge; and the second of which involves describing implementable processes by which it is possible to reason about such knowledge. Our approach is Fregean to the extent that the kind of cognitive system we propose puts emphasis upon the representation of Fregean senses. However, the approach is not entire]y Fregean because we do not represent denotations. This contrasts with the purely Fregean approaches of McCarthy (1979) and Creary (1979).

### A. McCarthy's Approach

McCarthy begins with the simple example of Pat knowing Mike's phone number which is Incidentally the same as Mary's phone number, although Pat does not necessarily know this. This example immediately exposes one of the difficulties of reasoning about knowledge, namely, the problem of inhibiting substitution of equal terms for equal terms in referentially opaque contexts. McCarthy's approach toward solving this problem involves explicitly representing senses and denotations.

### B. Creary's Extension

Creary extended McCarthy's system to handle iterated propositional attitudes. McCarthy's system fails for iterated propositional attitudes because propositions are represented but not their concepts. Creary's extensions involve introducing a hierarchy of typed concepts. Thus for individuals such as the person Mike, this scheme would have the person Mike, the concept of Mike, the concept of the concept Mike, and so forth. The higher concept is the Fregean sense of the lower concept, which reciprocally is the denotation of the higher concept. A similar situation holds for propositions. The hierarchy would consist of a truth value, the proposition which denotes the truth value, the concept of that proposition, and so on. This scheme allows for the representation of iterated propositional attitudes because all objects in the domain of discourse (most notablv propositions) have senses.

### C. The Maida-Shapiro Position

Our starting point is the observation that knowledge representations are meant to be part of the conceptual structure of a cognitive agent, and therefore should not contain denotations. The thread of this argument goes as follows: A cognitive agent does not have direct access to the world, but only to his representations of the world. For instance, when a person perceives a physical object such as a tree, he is really apprehending his representation of the tree. Hence, a knowledge representation that is meant to be a component of a "mind" should not contain denotations. A more elaborate statement of this position can be found in Maida and Shapiro (1982) and the system for representing knowledge, *called* Lambda Net, described in the remainder of this paper is described in Maida (1982). For our purposes, refraining from representing denotations achieves two goals: 1) the problem of substitution of equal terms for equal terms goes away because distinct terms are never equal; and 2) we can represent iterated propositional attitudes without invoking a hierarchy of types.

## II LAMBDA NET

### A. Intensional Individuals

There is a class of intensional individuals for which it can be said that they have a value as seen in assertions such as:

a) John-bear knows where Irving-bee is.
b) John knows Mike's phone number.
c) John knows the mayor's name.

What does John know in each of these sentences? He knows the value of some intensional individual. We can characterize these individuals by observing that they each involve a two-argument relation; namely, location-of, phone-no-of, and name-of, respectively. In each case, one argument is specified; namely: Irving-bee, Mike, and the mayor. The other argument is unspecified. We make the assumption that context uniquely determines the value of the unspecified argument. This value is the value of the intensional expression. The expressions themselves can now be represented as:

```
d) (the (lambda (x) (location-of Irving x)))
e) (the (lambda (x) (phone-no-of Mike   x)))
f) (the (lambda (x) (name-of     mayor  x)))
```

## B. Knowing Intensional Individuals

Since each of these expressions has a value, someone can know their values. We will express this via a relation called "know-value-of" which takes a cognitive agent and an intensional individual as arguments. To represent "John knows Mike's phone number," we write:

```
g) (know-value-of John
       (the (lambda (x) (phone-no-of Mike x))))
```

Observe that we treat propositional attitudes, and attitudes toward intensional individuals, as being relational and not as intensional operators. Knowing is viewed as correct (but not necessarily justified) belief.

The meaning of "know-value-of" entails that if John knows the value of Mike's phone number, and the value of Mike's phone number is 831-1234, then John "knows-that" the value of Mike's phone number is 831-1234.

## C. Iterated Propositional Attitudes

Reasoning about the knowledge states of others necessarily involves iterated propositional attitudes because the cognitive agent doing the reasoning is generating beliefs about another agent's knowledge state which itself may contain beliefs about the beliefs of other cognitive agents. Thus it is useful to show how Lambda Net represents such assertions. Creary (1979) offers three semantic interpretations of the ambiguous sentence:

h) Pat believes that Mike wants to meet Jim's wife.

He suggests that the task of representing these interpretations provides a strong test of the representation. In order to allow the reader to compare the Lambda Net scheme with Creary's we list the representations below. In each case, we give a rendering of the interpretation in English, our representation, and Creary's representation.

1) Pat believes that Mike wants to meet Jim's wife as such.

```
a) (believe-that Pat
       (wants Mike
          (meet
             (the (lambda (x:person)
                     (wife-of Jim x))))))

b) believes (pat, Wants{Mike, Meet$
       {Mike$, Wife$ Jim$}})
```

2) Pat believes that the person Mike wants to meet is Jim's wife, although Mike doesn't necessarily believe that.

```
a) (believe Pat
       (wife-of Jim
         (the (lambda (x:person)
           (wants Mike (meet Mike x))))))

b) believes (pat, Exist P$.Wants {Mike,
   Meet${Mike$, P$}} And Conceptof
   {P$, Wife Jim})
```

3) There is a specific person Pat believes Mike wants to meet. Neither necessarily believes this person is Jim's wife, but it incidentally is.

```
a) (wife-of Jim
       (the (lambda (x:person)
          (believe Pat (want Mike
                            (meet Mike x))))))

b) ∃P$ P.believes(pat, Wants {Mike,
   Meet${Mike$, P$}}) & conceptof(P$,P) &
   conceptof(P, wife jim)
```

The reader should refer to the original papers, Creary (1979) and Maida (1982), to make the proper comparison. One of Creary's goals is to stay within the confines of a first-order logic. Lambda Net does not have that constraint.

## D. Knowing Coreferential Intensional Individuals

To assert that two intensional individuals are coreferent, we write:

```
i) (equiv individual-1 Individual-2)
```

The relation "equiv" is mnemonic for extensional equivalence, and is the only reference to extensionality used in Lambda Net. One of our performance goals is to design a system which reacts appropriately to assertions of coreference. This involves specifying a method -to treat transparent and opaque relations appropriately. A relation, or verb, such as "dail" or "value-of" is transparent whereas a relation such as "know" is opaque with respect to its complement position. We can express this as:

```
(transparent dial)
(transparent value-of)
(conditionally-transparent know lst-arg 2nd-arg)
```

"Dial" and "value-of" are unequivically transparent, whereas "know" (either know-that or know-value-of) is transparent on the condition that the agent doing the knowing also knows that two entities are coreferent. We can partially express

### E. Axiom of Rationality

A system that reasons about the beliefs of another cognitive agent must make assumptions about the rationality of that agent in regard to what he considers legitimate rules of inference. We shall assume that all cognitive agents utilize the same set of inference schema. This is the Axiom of Rationality and we further assume that this set of schema is exactly the set given in this paper. A statement of the Axiom of Rationality is:

> Axiom of Rationality - If a cognitive agent knows or is capable of deducing all of the premises of a valid inference, then he is capable of deducing the conclusion of that inference.

The Axiom of Rationality enables one cognitive agent to determine by indirect simulation whether another cognitive agent is capable of inferring something. It implies, "If I figured it out and he knows what 1 know, then he can also figure it out if he thinks long enough." We will assume that the situations involved in knowing about telephone numbers are simple enough to make plausible the stronger rule, "If 1 figured out and he knows what I know, then he has definitely figured it out."

### F. Reasoning about Knowing

In this section we give an example of how reasoning about knowing can take place in Lambda Net by modeling the following situation involving a propositional attitude.

Premises: 1) John knows that Pat knows Mike's phone number.
2) John knows that Pat knows that Mike's phone number is the same as Mary's phone number.

Conclusion: John knows that Pat knows Mary's phone number.

By the definition of knowing as correct belief, it follows that: 1) Pat knows Mike's phone number; and, 2) Pat knows that Mike's phone number is the same as Mary's phone number. From conditional transparency and the Axiom of Rationality, the conclusion follows.

## III SUMMING UP

### A. What has been Achieved?

A system which can reason validly about knowledge must have at least the following three performance characteristics: 1) The system must be able to represent assertions involving iterated propositional attitudes and reason from these assertions; 2) The system must react appropriately to assertions involving coreference between distinct intensional individuals; and, 3) The system must felicitously represent that another cognitive agent can know the value of some intensional individual without the system itself necessarily knowing the value. Lambda Net has these characteristics just as Creary's (1979) does. However, Lambda Net offers the advantage of not invoking a hierarchy of conceptual types in order to achieve these performance characteristics.

### B. Current Work

We are implementing this system to process speech acts using the general strategy described by Allen (1979). This approach views speech acts as communications between cognitive agents about obstacles and potential solutions to achieving some goal. Therefore, comprehending and appropriately reacting to a speech act necessarily requires the capacity to reason about another cognitive agent's goals (wants), planning strategy, and knowledge states.

## REFERENCES

1 Allen, J. A plan-based approach to speech act recognition. Ph.D. Thesis, Computer Science, University of Toronto, 1979.

2 Creary, L. "Prepositional attitudes: Fregean representation and simulative reasoning." In Proc. IJCAI-79. Tokyo, Japan, August, 1979, pp."176-181.

3 Maida, A. "Using lambda abstraction to encode structural information in semantic networks." Report //1982-9-1, Box 1911, Center for Cognitive Science, Brown University, Providence, Rhode Island, 02912, U.S.A.

4. Maida, A. and Shapiro, S. "Intensional concepts in propositional semantic networks." Cognitive Science 6:4 (1982) 291-330.

5 McCarthy, J. "First order theories of individual concepts and propositions." In J. Hayes & D. Michie (Eds.) Machine Intelligence 9, New York: Halsted Press, 1979.