SRI International

Belief and Incompleteness

Technical Note 319

July 13, 1984

Kurt Konolige
Computer Scientist

Artificial Intelligence Center
Computer Science and Technology Division

To appear in *Formal Theories of the Common-Sense World*, edited by Jerry Hobbs.

This research was made possible in part by a gift from the System Development Foundation. It was also supported in part by Contract N00014-80-C-0296 from the Office of Naval Research, and by Contract F49620-82-K-0031 from the Air Force Office of Scientific Research.

333 Ravenswood Ave. • Menlo Park, CA 94025
(415) 326-6200 • TWX: 910-373-2046 • Telex: 334-486

Contents

1. Introduction

Two artificially intelligent (AI) computer agents begin to play a game of chess, and the following conversation ensues:

S_1: *Do you know the rules of chess?*
S_2: *Yes.*
S_1: *Then you know whether White has a forced initial win or not.*
S_2: *Upon reflection, I realize that I must.*
S_1: *Then there is no reason to play.*
S_2: *No.*

Both agents are state-of-the-art constructions, incorporating the latest AI research in chess playing, natural-language understanding, planning, *etc.* But because of the overwhelming combinatorics of chess, neither they nor the fastest foreseeable computers would be able to search the entire game tree to find out whether White has a forced win. Why then do they come to such an odd conclusion about their own knowledge of the game?

The chess scenario is an anecdotal example of the way inaccurate cognitive models can lead to behavior that is less than intelligent in artificial agents. In this case, the agents' model of belief is not correct. They make the assumption that an agent actually knows all the consequences of his beliefs. S_1 knows that chess is a finite game, and thus reasons that, in principle, knowing the rules of chess is all that is required to figure out whether White has a forced initial win. After learning that S_2 does indeed know the rules of chess, he comes to the erroneous conclusion that S_2 also knows this particular consequence of the rules. And S_2 himself, reflecting on his own knowledge in the same manner, arrives at the same conclusion, even though in actual fact he could never carry out the computations necessary to demonstrate it.

We call the assumption that an agent knows all logical consequences of his beliefs *consequential closure*. This assumption is clearly not warranted for either mechanical or human agents, because some consequences, although they are logically correct, may not be computationally feasible to derive. This is in fact illustrated by the chess scenario. Unfortunately, the best current formal models of belief on which AI systems are based have a *possible-worlds* semantics, and one of the inherent properties of these models is consequential closure. While such models are good at predicting what consequences an agent could *possibly* derive from his beliefs, they are not capable of predicting what an agent *actually* believes, given that the agent may have resource limitations impeding the derivation of the consequences of his beliefs.

The chess scenario illustrates one source of logical incompleteness in belief derivation, namely, an agent may not have enough computational resources to actually derive some result. We will identify several others in Section 2, by presenting a problem in belief representation that we have called the Not-So-Wise-Man Problem, a variation of the familiar Wise Man Puzzle. Not surprisingly, this problem involves reasoning about beliefs an agent does *not* have, even though they are logical consequences of his beliefs. The representational problems posed by the chess scenario and the not-so-wise-man problem cannot be solved within the framework of any model of belief that assumes consequential closure.

In this paper we introduce a new formal model of belief, called the *deduction model*, for representing situations in which belief derivation is logically incomplete. Its main feature is that it is a symbol-processing model: beliefs are taken to be expressions in some internal or "mental" language, and an agent reasons about his beliefs by manipulating these syntactic objects. Because the derivation of consequences of beliefs is represented explicitly as a syntactic process in this model, it is possible to take into account the fact that agents can derive some of the logically possible consequences, but in many cases not all of them. When the process of belief derivation is logically incomplete, the deduction model does not have the property of consequential closure.

Symbol-processing models of belief in themselves are not new (see, for example, Fodor [10], Lycan [23], and Moore and Hendrix [31] for some philosophical underpinnings, and McCarthy [26], Perlis [33], and Konolige [19] for AI approaches). The deduction model

presented here differs significantly from previous approaches, however, in two respects. First, it is a formal model: beliefs are represented in a mathematical framework called a *deduction structure*. The properties of the deduction model can be examined with some preciseness, and we do so in Section 3. Second, we have found sound and complete logics for the deduction model. One of these, B, is presented in Section 4, and used in the solution of the problems in Section 5. An important property of the deductive belief logic B is that it can serve as a basis for building computer agents that reason about belief. We have been able to find a number of interesting proof methods for B that have reasonable computational properties. Although the exposition of these methods is beyond the scope of this paper, at the appropriate points we will show how the design of the logic was influenced by computational considerations.

The nature of the deduction model and its logic B is further analyzed by comparing B to modal logics based on a possible-worlds semantics in Section 6. An important result is that the deduction model exhibits a correspondence property: in the limit of logically complete deduction, B reduces to a modal logic with possible-worlds semantics. Thus the deduction model dominates the possible-worlds model, while retracting the assumption of consequential closure.

The material for this paper was abstracted from the author's dissertation work (Konolige [21]). Because of the limited scope of this paper, we are not able to do more than mention in passing several interesting topics that are a part of the deduction model and its logics. Among these are efficient proof methods, the formal semantics and completeness proofs, extensions to B that permit quantifying-in, and introspection properties (beliefs about one's own beliefs). Interested readers can consult the dissertation itself for a fuller exposition.

2. Two Problems in the Representation of Belief

In this section we introduce three ways in which an agent may be incomplete in reasoning from his beliefs: *resource-limited incompleteness*, *fundamental logical incompleteness*, and *relevance incompleteness*. We argue that it is important for AI systems that reason about belief to be able to represent each of these, and offer two anecdotal problems to support this contention.

THE CHESS PROBLEM. *Suppose an agent knows the rules of chess. It does not necessarily follow that he knows whether White has a winning strategy or not.*

The chess problem, on the face of it, seems hardly to be a representational problem at all. Certainly its statement is *true*: no agent, human or otherwise, can possibly follow out all the myriad lines of chess play allowed by the rules to determine whether White has a strategy that will always win. What kind of model of belief would lead us to expect an agent to know whether White has a winning strategy? As we stated in the introduction, any model that does not take resource limitations into account in representing an agent's reasoning about the consequences of his beliefs has this behavior. Within such a model, we could establish the following line of argument.

Chess is a finite game,[1] *and so it is possible, in theory, to construct a complete, finite game tree for chess, given the rules of the game. The question of White's having a winning strategy is a property of this finite game tree. If for every counter Black makes, White has a move that will lead to a win, then White has a winning strategy. Thus, White's having a winning strategy is a consequence of the rules of*

[1] The finiteness of chess is assured by the rule that, if 50 moves occur without a pawn advance or piece capture, the game is a draw.

chess that can be derived in a finite number of simple steps. If an agent believes all the logical consequences of his beliefs, then an agent who knows the rules of chess will, by the reasoning just given, also know whether White has a winning strategy or not.

The chess problem is thus a problem in representing reasoning about beliefs in the face of resource limitations. The inference steps themselves are almost trivial; it is a simple matter to show that a move is legal, and hence to construct any position that follows a legal move from a given position. But while the individual inferences are easy, the number of them required to figure out whether White has a forced win is astronomical and beyond the computational abilities of any agent. We call this behavior *resource-limited incompleteness.* A suitable model of belief must be able to represent situations in which an agent possesses the inferential capability to derive some consequence of his beliefs, but simply does not have the computational resources to do so.

THE NOT-SO-WISE-MAN PROBLEM. *A king, wishing to know which of his three advisors is the wisest, paints a white dot on each of their foreheads, tells them there is at least one white dot, and asks them to tell him the color of their own spots. After a while the first replies that he doesn't know; the second, on hearing this, also says he doesn't know. The third then responds, "I also don't know the color of my spot; but if the second of us were wiser, I would know it."*

The not-so-wise-man problem is a variation of the classic Wise Man Puzzle, which McCarthy (in [24] and [25]) has used extensively as a test of models of knowledge. In the classic version, the third wise man figures out from the replies of the other two that his spot *must* be white. The "puzzle" part is to generate the reasoning employed by the third wise man. The reasoning involved is really quite complex and hinges on the ability of the wise men to reason about one another's beliefs. To convince themselves of this, readers who have never tried before may be interested in attempting to solve it before reading the solution below.

Solution to the Wise Man Puzzle: *the third wise man reasons: "Suppose my spot were black. Then the second of us would know that his own spot was white, since he would know that, if it were black, the first of us would have seen two black spots and would have known his own spot's color. Since both answered that they had no knowledge of their own spot's color, my spot must be white."*

The difficulty behind this puzzle seems to lie in the nature of the third wise man's reasoning about the first two's beliefs. Not only must he pose a hypothetical situation (*Suppose my spot were black*), but he must then reason within that situation about what conclusions the second wise man would come to after hearing the first wise man's response. This in turn means that he must reason about the second wise man's reasoning about the first wise man's beliefs, as revealed by his reply to the king. Reasoning about beliefs about beliefs about beliefs... we call reasoning about *iterated* or *nested beliefs*. It can quickly become confusing, especially when there are conditions present concerning what an agent does *not* believe.

In the Wise Man Puzzle, nested belief contributes to the complexity of the reasoning involved. The third wise man must reason about what the second wise man does not know (the color of his own spot); in doing this, he must also reason about the second wise man's reasoning about what the third wise man does not know (the color of *his* own spot). It is particularly annoying and troublesome to keep track of who believes what after several occurrences of not-believing in a statement of nested belief. Because human agents find it so difficult, the Wise Man Puzzle is thought to be a good test of the *competence* of any model of belief. If one can state the solution to the puzzle within the framework of Model X, so the reasoning goes, then Model X is at least good enough to show what might conceivably be concluded by agents in complicated situations involving nested beliefs.

It is possible to solve the Wise Man Puzzle within the confines of belief models that assume consequential closure (see, *e.g.*, McCarthy [24], [25] or Sato [38]). Such models make the assumption that every agent believes other agents' beliefs are closed under logical consequence, and so on to arbitrary depths of belief nesting. While this is an accurate assumption if one is trying to model the competence of ideal agents (which is what the Wise Man Puzzle seeks to verify), it cannot represent interesting ways in which reasoning about complicated nested beliefs might fail for a less-than-ideal agent. This is the import

of the not-so-wise-man problem. From the reply of the third wise man, it appears that the second wise man lacks the ability to deduce all the consequences of his beliefs. The representational problem posed is to devise interesting ways in which the second wise man fails to be an ideal agent, and then show how the third wise man can represent this failure and reply as he does.

The not-so-wise-man problem does not seem to fall into the category of resource-limited incompleteness mentioned in the chess problem, since the computational requirements of the inferences are not particularly acute. We can identify at least two other types of incompleteness (there may certainly be more) that are interesting here and would be useful to represent. In one of these, the second wise man may have incomplete inferential procedures for reasoning about the other wise men's beliefs, especially if tricky combinations of *not-believing* are present. Suppose, for instance, the second wise man were to see a black spot on the third wise man, and a white spot on the first wise man (this is the hypothetical situation set up by the third wise man in solving the classic puzzle). If he were an ideal agent, he would conclude from the first wise man's reply that his own spot must be white (by reasoning: *if mine were not white, the first of us would have seen two black spots and so claimed his own as white*). But he may fail to do this because his rules for reasoning about the beliefs of the first wise man simply are not powerful enough. For example, he might never consider the strategy of assuming that his spot was black, and then asking himself what the first wise man would have said. In this case, the second wise man's inferential process, even when given adequate resources, is just not powerful enough in terms of its ability to arrive at simple logical conclusions. To apply an analogy from high-school algebra: a student who is confronted with the equation $x + a = b$ and asked to solve for x won't be able to do so if he doesn't know the rule that subtracting equals from each side leaves the equation valid. It is not that the student lacks sufficient mental resources of time or memory to solve this problem; rather, his rules of inference for dealing with equational theories are logically incomplete. To contrast this type of incompleteness with the resource-limited incompleteness described in the chess problem, we call it *fundamental logical incompleteness.*

Another way in which the not-so-wise-man might fail to draw conclusions is if he were to make an erroneous decision as to what information might be relevant to solving

his problem. Although the Wise Man Puzzle has a fairly abstract setting, it is reasonable to suppose that actual agents confronted with this problem would have a fair number of extraneous beliefs that they would exclude from consideration. For example, the not-so-wise-man might be privy to the castle rumor mill, and therefore believe that the first wise man was scheming to marry the king's daughter. A very large number of beliefs of this sort have no bearing on the problem at hand, but would tend to use up valuable mental resources if they were given any serious consideration. One can imagine an *unsure agent* who could never come to any negative conclusions at all, because he would keep on considering more and more possibilities for solving a problem. This agent's reasoning might proceed as follows: *I can't tell the color of my spot by looking at the other wise men. But maybe there's a mirror that shows my face. No, there's no mirror. But maybe my brother wrote the color on a slip of paper and handed it to me. No, there's no slip of paper, and my brother's in Babylon. But maybe ...*

McCarthy (in [27]) first called attention to the problem of representing what is *not* the case in solving puzzles. In the Missionaries and Cannibals Puzzle, why can't the missionaries simply use the bridge downstream to get across? A straightforward logical presentation of the puzzle doesn't explicitly exclude the existence of such a bridge. And, if it did, we could always come up with other modes of transportation that had not been considered beforehand and explicitly excluded. McCarthy called the general problem of specifying what conditions do not hold in a puzzle the *circumscription problem*. By analogy, we call the problem of specifying what beliefs an agent does not have, or does not use in solving a given task, the problem of *circumscriptive ignorance* (see Konolige [20]). Without a solution to this representational problem, all agents will be modeled as unsure agents – never able to reach a conclusion about what they don't believe, even though it is obvious when the set of relevant beliefs is circumscribed.

Of course, if an agent can circumscribe his beliefs, it is possible that he will choose the wrong set of beliefs, and exclude some that actually are relevant. The not-so-wise-man may decide that the beliefs of the first wise man are not germane to the problem of figuring out his own spot's color. Thus, even though he has all the relevant information, and even sufficiently powerful inference rules and adequate resources, he may fail to come to a correct conclusion because he has circumscribed his beliefs in the wrong way. We call this type of incompleteness *relevance incompleteness*.

Within a model of belief that assumes consequential closure, it is possible to represent circumscriptive ignorance, but only in a relatively limited fashion. If consequential closure is assumed, one can state that an agent is ignorant of some fact which is not a logical consequence of his beliefs (McCarthy [25] uses this technique in his solution to the Wise Man Puzzle). But this clearly does not capture the complete conditions of circumscriptive ignorance, since agents are often ignorant of some of the logical consequences of their beliefs, as in the chess scenario.

Modeling relevance incompleteness (or having the third wise man do so) is impossible if it is assumed that the beliefs of agents are consequentially complete. One simply cannot partition the set of beliefs into those that are either relevant or not to a given problem; *all* the consequences of beliefs are believed. If we try to state the conditions of relevance incompleteness within such a model, we can arrive at a contradiction, where a proposition is both believed (because of the assumption of consequential closure) and not believed (because of the condition of relevance incompleteness).

3. The Deduction Model of Belief

The two belief representation problems can be solved within the framework of a formal model of belief that we call the deduction model. In this section we define the model; in the next we introduce a logic family B as its axiomatization.

The strategy we pursue is to first examine the way typical AI robot planning systems (STRIPS [9], NOAH [37], WARPLAN [42], KAMP [1], *etc.*) represent and reason about the world. This leads to the identification of an abstract *belief subsystem* as the internal structure responsible for the beliefs of these agents. The characteristics of belief subsystems can be summarized briefly as follows.

1. A belief subsystem contains a list of sentences in some internal ("mental") language, the *base beliefs.*
2. Agents can infer consequences of their beliefs by syntactic manipulation of the sentences of the belief subsystem.
3. The derivation of consequences of beliefs is logically incomplete, because of limitations of the inferential process.

Having identified a belief subsystem as that part of an agent responsible for beliefs, our next task is to define a formal mathematical structure that models it accurately. The decisions to be made here involve particular choices for modeling the various components of a belief subsystem: What does the internal language look like? What kind of inference process derives consequences of the base beliefs? and so on. The formal mathematical object we construct according to these criteria is called a *deduction structure.* Its main components are a set of sentences in some logical language (corresponding to the base beliefs of a belief subsystem) and a set of deduction rules (corresponding to the belief inference rules) that may be logically incomplete. Because we choose to model belief subsystems

in terms of logical (but perhaps incomplete) deduction, we call it the *deduction model of belief.*

3.1. Planning and Beliefs: the Belief Subsystem Abstraction

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 properties of the real world that are important for its task; *e.g.*, it might assume that there are objects that can be stacked in simple ways (the *blocks world* domain). The state of the abstract world at any particular point in time has been called a *situation* in the AI literature.

In general, the planning system will have only incomplete knowledge of a situation. For instance, if it is equipped with visual sensors, it may be able to see only some of the objects in the world. What this means is that the system has to be able to represent and reason about partial descriptions of situations. The process of deriving beliefs is a *symbol-manipulating* or *syntactic* operation that takes as input sentences of the formal language, and produces new sentences as output. Let us call any new sentences derived by inferences the *inferable sentences*, and the process of deriving them *belief inference.*

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 process for deriving their consequences. 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 with other processes of the planning system. The belief system is composed of the base beliefs, together with the belief inference process. Belief inference itself can be decomposed into a set of inference rules and a control strategy

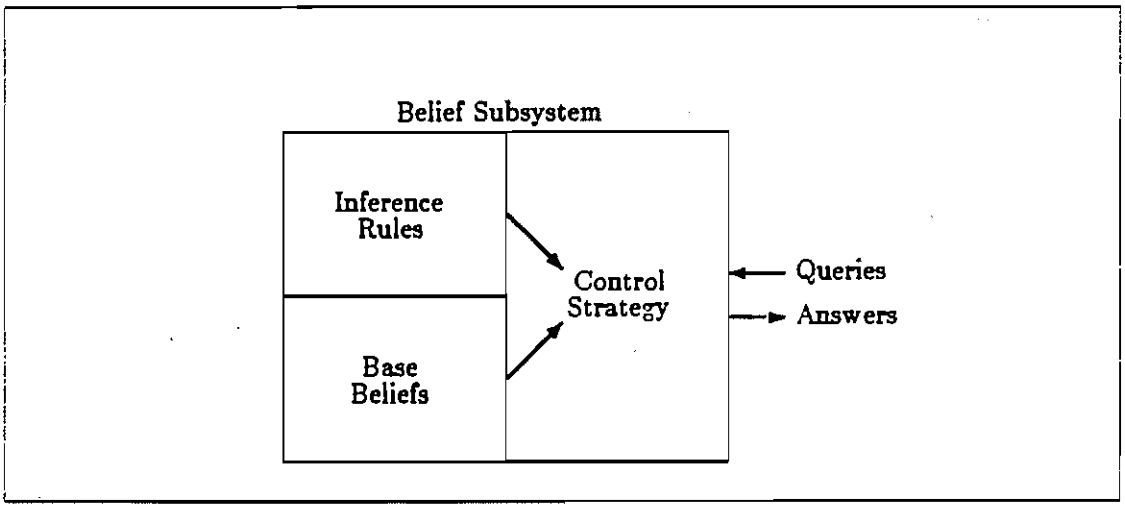

Figure 1. Schematic of a Belief Subsystem

that determines how the rules are to be applied and where their outputs go when requests are made to the belief subsystem.

A belief subsystem defines an agent's beliefs by the action of the inference rules on the base beliefs, under the guidance of the control strategy. Some, but not necessarily all, of the inferable sentences will be beliefs of the agent; which inferable sentences are actually beliefs depends on the details of the control strategy and the resources available for belief inference.

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 beliefs; this happens, for example, when the plan derivation process decides which sentences hold in a new situation. The problem of updating and revising beliefs in the face of new information is a complicated research issue in its own right, and we do not address it here (see Doyle [7] for some related AI 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 infer, using the its rules, that the sentence is follows from the base beliefs. It is this process of *belief querying* that we model in this paper.

The above description of the operation of a belief subsystem is meant to convey the idea that in most formal planning systems there is a tight interaction between belief

subsystems and planning. Different systems may deviate from the described pattern to a greater or lesser extent. In some systems, the representation of facts may be so limited, and that of actions so explicit, as to almost obviate the need for belief deduction *per se* (as in some versions of STRIPS). In others, deduction may be used to calculate all the effects of an action by expanding the representation to include situations as objects (as in WARPLAN). Here it is hard to make a clean separation between deductions performed for the purpose of deriving consequences of beliefs and those that establish the initial set of facts about a new situation. However, it is still conceptually useful to regard the belief subsystem as a separate structure and belief derivation as a separate process within the planning system.

3.2. A Formal Model of Belief

The formal mathematical object we use to model belief subsystems is called a *deduction structure*. A deduction structure is a tuple consisting of two sets and will be written as $(B, \mathcal{R})$. The set B is a set of sentences in some language L; It corresponds to the base beliefs of a belief subsystem and its members are referred to as the *base sentences* of the deduction structure. $\mathcal{R}$ is a set of deduction rules for L; these correspond to the inference rules of a belief subsystem. We demand that deduction structures satisfy the following four conditions.

Language Property.	The language of a deduction structure is a logical language.
Deduction Property.	The rules of a deduction structure are logical deduction rules. These rules are sound, effectively computable, and have bounded input.
Closure Property.	The *belief set* of a deduction structure is the least set that includes the base sentences and is closed under derivations by the deduction rules.
Recursion Property.	The intended model of deduction structure sentences involving belief is the belief set of another deduction structure.

We discuss each of these properties briefly below. For the interested reader, a more thorough treatment of the mathematical properties of deduction structures is given in the next subsection.

About the only condition we require of L is that it be a *logical* language. Logical languages are distinguished by having a constructable set of syntactic objects, the *sentences* of the language, together with an *interpretation method* (a means of assigning true or false to every sentence with respect to a given state of affairs).

$\mathcal{R}$ is a set of deduction rules that operate on sentences of L. We will leave unspecified the exact form of the deduction rules $\mathcal{R}$, but we do insist that they operate in the normal manner of deduction rules in some proof-theoretic framework. This means that there is the concept of a *derivation* of a sentence, which is a structure built from effective applications of the rules $\mathcal{R}$. If p is derivable from the set of sentences Γ in this manner, we write $\Gamma \vdash_{\mathcal{R}} p$, where $\vdash_{\mathcal{R}}$ is a derivation operator for the rules $\mathcal{R}$. For example, in terms of Hilbert systems (as defined in Kleene [18]), $\mathcal{R}$ would be a set of logical axioms (zero-premise rules) together with *modus ponens* (a two-premise rule). A sentence p would be derivable from the premise sentences $B = \{b_1, b_2, \ldots\}$ if there were a Hilbert proof of $(b_1 \wedge b_2 \wedge \ldots) \supset p$, using the logical axioms and *modus ponens*.

A deduction structures models beliefs by its *belief set*, which we define as follows.

DEFINITION 3.1.

$$\mathrm{bel}((B, \mathcal{R})) =_{\mathrm{df}} \{p | B \vdash_{\mathcal{R}} p\} \quad .$$

The belief set is composed of all sentences that are derivable from the base set B with the rules $\mathcal{R}$. The derivation operator $\vdash_{\mathcal{R}}$ thus corresponds to the belief inference process of belief subsystems.

For several technical reasons, we restrict the derivation operators allowed in deduction structures to those that satisfy a deductive closure condition. One consequence of this assumption is that the belief set itself obeys a closure property: if the sentence p can be derived from the sentences in a belief set, then it too must be present in the belief set. By making the assumption of deductive closure, the task of formalizing and reasoning about deduction structures is greatly simplified.

It is important to note that deductive closure does not entail *consequential* closure for belief derivation: a set of sentences closed under logically incomplete deduction rules need not contain all logical consequences of the set. This is an important property of

deduction structures, and it enables them to capture the behavior of belief subsystems with resource-bounded control strategies.

Finally, we single out certain sentences of the deduction structure for special treatment, namely the ones that themselves refer to the beliefs of agents. In discussing the not-so-wise-man problem in the previous section, we mentioned that one of the key tests of a belief model is its ability to handle nested beliefs by assuming that agents use the model in representing other agents' beliefs; a belief model that has this characteristic is said to have the recursion property. In terms of deduction structures, the recursion property implies that the sentences of the internal language L that are about beliefs should have another deduction structure as their intended interpretation.

3.3. Properties of Deduction Structures

In this subsection we treat the mathematical properties of deduction structures in some detail, taking care to show how they can model the behavior of belief subsystems of formal AI planning systems.

Language Property

One restriction we place on the language of deduction structures is that sentences of the language have a well-defined (*i.e.*, truth-theoretic) semantics. Such a requirement seems absolutely necessary if we are going to talk about the beliefs of an agent being *true* of the actual world, or, as we will want to do in discussing the rationality of agents, judge the *soundness* of belief deduction rules. Such concepts make no sense in the absence of an interpretation method – a systematic way of assigning meanings to the constructions of the language. As Moore and Hendrix ([31], parts IV and especially V) note, the interpretation method is not something the agent carries around in his head; a belief subsystem is just a collection of sentences, and computational processes manipulate the sentences themselves, not their meanings. One simply cannot put the referent of "Cicero" into a robot's computation device, even if he (Cicero, of course) were alive. But the attribution of semantics to sentences is necessary if an outside observer is to analyze the nature of an agent's beliefs.

How well do actual robot belief subsystems fit in with the assumption of a logical language of belief? AI systems use a variety of representational technologies; chief among these are frames, scripts, semantic nets, and the many refinements of first-order logic (FOL), including PROLOG and the "procedurally oriented" logics of μ-PLANNER, CONNIVER, QA4, and the like. The representations that fall into the latter category inherit their semantics from FOL, despite many differences in the syntactic form of their expressions. But what can we say about the first three? In surface form they certainly do not look anything like conventional mathematical logics; furthermore, their designers often have not provided anything but an informal idea of what the meanings of expressions in tbe language are. When, after all, is a pair of nodes connected by a directed arc *true* of the world? As Hayes [11] has forcefully argued, the lack of a formal semantics is a big drawback for these languages. Fortunately, on further examination it is often possible to provide such a semantics, usually by transliterating the representation into a first-order language (see Woods [44] and Schubert [39] for a reconstruction of semantic nets in FOL terms, and Brachman [4] for a similar analysis of frames).

In discussing human belief, several philosophers of mind have argued that internal representations that count as beliefs must have a truth-value semantics (see Fodor [10], Field [8], and Moore and Hendrix [31] for a discussion of the many intricate arguments on this subject, especially pp. 48ff. of Field and part V of Moore and Hendrix). However, there almost certainly is a lot more to human belief than can be handled adequately within the framework of a logical language. For example, the question of membership in the belief set of a deduction structure is strictly two-valued: a sentence is either a member of the belief set of a deduction structure, or it is not. If it is, then the assumed interpretation is that the agent believes that sentence to be true of the world. Deduction structures thus do not support the notion of uncertain beliefs directly, as they might do if fuzzy or uncertain membership in the belief set were an inherent part of their structure.[1]

One further requirement is that L contain expressions referring to the beliefs of agents. Generally we will take this to be a belief operator whose argument is an expression in L.

[1] However, uncertain beliefs could always be introduced into deduction structures in an indirect manner by letting L contain statements about uncertainty, *e.g.*, statements of the form *P is true with probability* 1/2.

Finally, it is often the case that we will want to freeze the language of deduction structures in order to study their properties at a finer level of detail, *e.g.*, when looking at the behavior of nested beliefs in general or when giving the particulars of the solution to a representational problem. It is convenient to think of the language as being a *parameter* of the formal model. For every logical language L, there is a class of deduction structures $\mathsf{D}(L, \rho)$ whose base sets are sentences of the language L (the parameter ρ will be explained in discussing the recursion property below).

Deduction Property

Rules for deduction structures are rules of inference with the following restrictions:

1. The rule is an effectively computable function of sentences of L.
2. The number of input sentences is boundedly finite.
3. The conclusion is sound with respect to the semantics of L.

These restrictions are those normally associated with deduction rules for classical logic, although, strictly speaking, deduction rules need not be sound, if one is just interested in proof-theoretic properties of a logic without regards to semantics.

The fact that belief deduction rules are effectively computable functions means that they can be very complicated indeed. Mathematical logicians are interested in logics with simple deduction rules (such as Hilbert systems) because it is easy to analyze the proof-theoretic structure of such systems. However, for the purpose of deriving proof methods for commonsense reasoning in AI, it is often better to sacrifice simplicity for computational efficiency. For example, Robinson's *resolution rule* [36], which employs a matching process called unification, is a complicated rule that has been widely employed in AI theorem-proving. Another important technique is Weyhrauch's *semantic attachment* [43], a general framework for viewing the results of computation as deductions. In this paper, we will exploit complicated rules that perform deductions that are relatively "large" with respect to the grain size of the predicates, particularly in solving the chess problem of Section 2. Although these "large" deductions could be broken down into smaller steps, it is computationally and conceptually easier to view them as single deductions.

We call an inference rule *provincial* if the number of its input sentences is boundedly finite; deduction rules are always provincial. We thus do not allow inferences about beliefs

that take an infinite number of premises. For example, the following rule of Carnap's is not a valid rule of belief deduction: *if for every individual a: $F(a)$ is a theorem, then $\forall x.F(x)$ is a theorem.*[1] Provincial inference rules have the following interesting property: if α is a consequence of a set of sentences S by the rule, then it is also a consequence of any larger set $S' \supset S$. To see that this must be so, consider that, if α can be derived by the application of provincial rules on the set of sentences S, and S' contains S, then the same derivation can be performed by using S'. Rules that adhere to this property are called *monotonic*. Technically, monotonicity is convenient because it means we can reason about what an agent believes on the basis of partial knowledge about his beliefs. A derivation made using a subset of his beliefs cannot be retracted in the face of further information about his beliefs.

Several types of nonmonotonic (and unsound) reasoning have been of interest to the AI community, specifically

Belief revision:	the beliefs of an agent are updated to be consistent with new information (*e.g.*, Doyle [7]).
Default reasoning:	an agent "jumps to a conclusion" about the way the world is (*e.g.*, McCarthy [27], Reiter [35]).
Autoepistemic reasoning:	an agent comes to a conclusion about the world based on his knowledge of his own beliefs (*e.g.*, Collins *et al.* [6], Moore [30]).

We are explicitly not trying to arrive at a theory of these forms of reasoning. Indeed, it is helpful here to make the distinction that Israel (in [16]) advocates between inference or reasoning in general (which may have nonmonotonic properties) and the straightforward deduction of logical consequences from a set of initial beliefs. It is the latter concept only that is treated in this paper.

If we wish to accommodate some nonmonotonic theory formally within the framework of the deduction model, then we can view its inferences as deduction rules operating on deduction structure theories as a syntactic whole. McCarthy [27] exploits this approach to formalize a certain type of useful default inference, which he calls *circumscription* (see the description of the not-so-wise-man problem in Section 2). In defining the logic B, we

[1] I am indebted to David Israel for pointing out this example.

will show how to formalize *circumscriptive ignorance*, a type of nonmonotonic inference, in this manner.

Deduction rules for belief subsystems must also be sound. A sound deduction rule is one for which, if the premises are true in an interpretation, then the conclusion will be also (see Kleene [18]). Informally, one would say that sound deduction rules never deduce false conclusions from true premises. *Modus ponens* is an example of such a rule: if p and $p \supset q$ are true, then q must also be.

Soundness of inference is an important property for robot agents in deriving consequences of their beliefs. We would not want a robot who believed the two sentences

(3.1) *All men are mortal.*
Socrates is a man.

to then deduce (and hence believe) the sentence

(3.2) *Socrates is not mortal.*

Soundness is not a critical assumption for the deduction model, since none of the major technical results depend on it. In some cases we may wish to relax it, for example, in modeling the behavior of human syllogistic reasoning, which is often unsound (see Johnson-Laird [17]).

To sum up: deduction structures are restricted to using inference rules which are provincial, sound, and effectively computable. Several interesting types of reasoning, such as reasoning about defaults or one's own beliefs, cannot be modeled directly as deduction rules over sentences. However, they can be incorporated into the deduction model if the input to the rules is taken to be the deduction structure as a whole.

Closure Property

The closure property states that the belief set of a deduction structure is *closed under derivations*. Formally, this amounts to the following conditions on the belief set.

1. $B \subseteq \text{bel}(\langle B, \mathcal{R} \rangle)$.
2. If $\Gamma \subseteq \text{bel}(\langle B, \mathcal{R} \rangle)$ and $\Gamma \vdash_{\mathcal{R}} p$, then $p \in \text{bel}(\langle B, \mathcal{R} \rangle)$.

Since we have defined the belief set in terms of the belief derivation operator $\vdash_{\mathcal{R}}$ (Definition 3.1), we can reexpress these as conditions on belief derivation.

(Reflexivity) $\alpha \vdash_{\rho(i)} \alpha$.

(Closure) If $\Gamma \vdash_{\rho(i)} \beta$ and $\beta, \Sigma \vdash_{\rho(i)} \alpha$, then $\Gamma, \Sigma \vdash_{\rho(i)} \alpha$.

Reflexivity guarantees that the base set will be included in bel, and the closure condition establishes closure of bel under derivation.

The chief motivation for requiring derivational closure is that it simplifies the technical task of formalizing the deduction model. Consider the problem of formalizing a belief subsystem that has a complex control strategy guiding its inferential process. To do this correctly, one must write axioms that not only describe the agendas, proof trees, and other data structures used by the control strategy, but also describe how the control strategy guides inference rules operating on these structures. Reasoning about the inference process involves using these axioms to perform deductions that *simulate* the belief inference process, a highly inefficient procedure. By contrast, the assumption of derivational closure leads to a simple formalization of deduction structures in a logic B that incorporates the belief inference process in a direct way. We need not differentiate between a belief as a member of the base set, or as a derived sentence. A sentence that follows from any members of the belief set is itself a belief. The axiomatization of B is simplified, since we need only have an operator whose intended interpretation is membership in the belief set. In Section 4, we exploit the properties of closed derivational systems to exhibit a complete axiomatization of B, using techniques that are manner similar to the *procedural attachment* methods of Weyhrauch [43].

The closure property is an extremely important one, and we should examine its repercussions closely. A point that we have already made is that derivational closure is not the same as consequential closure. The latter refers to a property of sets of sentences based on their *semantics*: every logical consequence of the set is also a member of the set. The former refers to the *syntactic* process of derivability; if the rules $\mathcal{R}$ are not logically complete, then a set of sentences that is derivationally closed under $\mathcal{R}$ need not be consequentially closed.

One of the key properties of belief subsystems that we wish to model is the incompleteness of deriving the consequences of the base set of beliefs. We have identified three sources of incompleteness in belief subsystems: an agent's belief inference rules may be too weak from a logical standpoint, or he may decide that some beliefs are irrelevant to a query, or his control strategy may perform only a subset of the inferences possible when confronted with resource limitations. The assumption of derivational closure for deduction structures affects their ability to model incomplete control strategies, since closure demands that all possible deductions be performed in deriving the belief set.

For an important class of incomplete control strategies, however, there is a corresponding complete control strategy operating on a different set of inference rules that produces the same beliefs on every base set. The criteria that defines this class is that the control strategy use only a *local cost bound* in deciding to drop a particular line of inference. By "local" is meant that the control strategy will always pursue a line of inference to a certain point, without regard to other lines of inference it may be pursuing in parallel. Control strategies with a local cost bound are important because their inferential behavior is predictable: all inferences of a certain sort are guaranteed to be made.

Deduction structures can accurately model the class of locally bounded incomplete control strategies by using an appropriate set of logically incomplete deduction rules. A good example is found in the solution to the chess problem in Section 5. The agent's control strategy applies general rules about chess to search the game tree to only a limited depth; this is modeled in a deduction structure by using deduction rules that work only above a certain depth of the game tree, and applying them exhaustively.

In belief subsystems whose control strategies have a global cost bound, the concept of belief itself is complicated, since one must differentiate between base beliefs and beliefs inferred with some amount of effort. Deduction structures are only an approximate model of these subsystems, and a language with a single belief operator is no longer sufficient for their axiomatization.

Recursion Property

If belief subsystems adhere to the recursion property, then agents view other agents as having belief subsystems similar to their own. This still leaves a considerable degree of

flexibility in representing nested beliefs. For example, an agent John might believe that Sue's internal language is L_1 and that she has a set of deduction rules $\mathcal{R}_1$, whereas Kim's internal language is L_2 and her deduction rules are $\mathcal{R}_2$. In addition, John might believe that Sue believes that Kim's internal language is L_3, and that her rules are $\mathcal{R}_3$. We call the description of a belief subsystem at some level of nesting a *view*; formally, views are sequences of agents' names, so that the view *John, Sue* is Sue's belief subsystem as John sees it. We will often use the Greek letter ν to stand for an arbitrary view, and lowercase Latin letters (i, j, *etc.*) for singleton views, which are agents' actual belief subsystems. Since the formal objects of the deduction model are deduction structures, these will be indexed by views when appropriate. For example, the $d_{John,Sue}$ is a deduction structure modeling the view *John, Sue*.

Obviously, some fairly complicated and confusing situations might be described, with agents believing 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 novice's deductive capabilities that would initially be less powerful and complete than its own, and could be modified as the novice learned about the domain.

We model the recursion property of belief subsystems within the framework of deduction structures by allowing sentences of L to refer to the beliefs of agents. A standard construct is to have a *belief operator* in L: an operator whose arguments are an agent S and a sentence P, and whose intended meaning is that S believes P. According to the recursion property, this means that the belief operator must have a deduction structure as its interpretation. Deduction rules that apply to belief operators will be judged sound if they respect this interpretation. For example, suppose a deduction structure d_ν has a rule stating that the sentence "John believes q" can be concluded from the premise sentences "John believes p" and "John believes $p \supset q$". This is a sound rule of d_ν if *modus ponens* is believed to be a rule of John's belief subsystem as viewed from the view ν, since the presence of p and $p \supset q$ in a deduction structure with *modus ponens* means that q will be derived.

Several simplifying assumptions are implicit in the use of deduction structures to model the nested views of belief subsystems. The language L contains a belief operator

that denotes membership in a belief set (its intended interpretation), and so L can describe what sentences are contained in an agent's belief set. However, there is no provision in L for talking about the deduction rules an agent uses. Instead, these nested-belief rules are implicitly specified by the rules that manipulate sentences with belief operators. Consider the example from the previous paragraph. Let us suppose that we are modeling Sue's belief subsystem with the deduction structure d_{Sue}. Because Sue believes that John uses *modus ponens*, a sound rule of inference for d_{Sue} would be the one that was stated above, namely, the sentence "John believes q" could be concluded from the premise sentences "John believes p" and "John believes $p \supset q$." All of the rules that Sue believes John uses are modeled in this way. Similarly, if, in Sue's opinion, John believes that Kim uses a certain rule, this will be reflected in a rule of John's deduction structure as seen by Sue, which in turn will be modeled by a rule in d_{Sue}. The deduction model thus assumes that the rules for each view, though they may be different, are a fixed parameter of the model. We introduce the function $\rho(\nu)$ to specify deduction rule sets for each view ν; thus, for each function ρ and each language L, there is a class of deduction structures $\mathrm{D}(L, \rho)$ that formalize the deduction model. If the rules ρ are complete with respect to the semantics of L, then the class is said to be *saturated*, and is written $\mathrm{D}_s(L, \rho)$

A final simplification that is not inherent in the deduction model, but which we introduce here solely for technical convenience, is to assume that all deduction structures in all views use the same language L. There are situations in which we might want to relax this restriction, it makes the axiomatization less complex in dealing with the problems at hand.

4. The Logic Family B

We now define a family of logics $\mathsf{B}(L, \rho)$ for stating facts and reasoning about deduction structures. This family is parameterized in the same way as deduction structures, namely by an agents' language L and an ensemble of deduction structure rules ρ. Each logic of the family is an axiomatization of the deduction structures $\mathsf{D}(L, \rho)$.

The language of B includes operators for stating that sentences are beliefs of an agent, but not for describing deduction rules of agents. Thus the deduction rules are a parameter of the logic family, and are fixed once we decide to use a particular logic of the family. The ensemble function ρ picks out a set of rules for each agent. The reason we chose to make the deduction rules a parameter of B is that it is then possible to find efficient proof methods for B. One of the interesting features of B's axiomatization is that agents' rules are actually present as a subset of the rules of B; proofs about deduction structures in B use these rules directly in their derivation.

The logic of B is framed in terms of a modified form of Gentzen systems, the block tableau systems of Hintikka. Although they may be unfamiliar to some readers, block tableaux are easy to work with and possess some natural advantages when applied to the formalization of deduction structures. Unlike Hilbert systems, which contain complex logical axioms and a single rule of inference in the propositional case (*modus ponens*), block tableau systems have simple axioms and a rich and flexible method of specifying deduction rules. We exploit this capability when we incorporate deduction structure rules into B.

In this section we first present a brief overview of block tableaux. Then we give the postulates of the family B, and a particularly simple subfamily called BK that will be used in solving the problems. By way of example, we prove some theorems of BK.

4.1. Block Tableaux

Most of this section will comprise a review for those readers who are already familiar with tableaux systems.

The Base Language L_0

The language of B is formed from a base language L_0 that does not contain any operators referring to beliefs. L_0 is taken to be a first-order language with constant terms. An interpretation of L_0 is a truth-value assignment to all sentences (closed formulas) of L_0; this assignment must be a *first-order valuation*, that is, it must respect the standard interpretation of the universal and existential quantifiers as well as the Boolean connectives.

We call L_0 *uninterpreted* if every first-order valuation is an interpretation of L_0; *partially interpreted* if some proper subset of the first-order valuations are interpretations of L_0; and *fully interpreted* (or simply *interpreted*) if there is a singleton interpretation of L_0. A sentence of L_0 is *valid* if and only if it is true in every interpretation of L_0.

We use lowercase Latin or Greek letters (p, q, α, *etc.*) as metavariables that stand for sentences of L_0. A formula of L_0 that possibly contains the free variable x will be indicated by $\alpha(x)$; the formula derived by substituting the constant a everywhere for x is denoted by $\alpha(x/a)$. Uppercase Greek letters ($\Gamma =_{df} \{\gamma_1, \gamma_2, \ldots\}$, $\Delta =_{df} \{\delta_1, \delta_2, \ldots\}$, *etc.*) stand for *finite sets* of sentences of L_0. By p, Γ we mean the set $\{p\} \bigcup \Gamma$. We also introduce the abbreviation $\neg\Gamma =_{df} \{\neg\gamma_1, \neg\gamma_2, \ldots\}$.

Sequents

Sequents are the main formal object of block tableaux systems.

DEFINITION 4.1. *A sequent is an ordered pair of finite sets of sentences, $\langle\Gamma, \Delta\rangle$. This sequent will also be written as $\Gamma \Rightarrow \Delta$, and read as "$\Delta$ follows from Γ."*

A sequent $\Gamma \Rightarrow \Delta$ is true in an interpretation *of its component sentences iff one of γ_i is false, or one of δ_j is true. A sequent is* valid *iff it is true under all interpretations, and* satisfiable *iff it is true in at least one interpretation.*

From the definition of truth for a sequent, it should be clear that a sequent $\Gamma \Rightarrow \Delta$ is true in an interpretation just in case the sentence $(\gamma_1 \wedge \gamma_2 \wedge \ldots) \supset (\delta_1 \vee \delta_2 \vee \ldots)$ is true in that interpretation. Thus, in a given interpretation a true sequent can be taken as asserting that the conjunction of γ's materially implies the disjunction of the δ's.

We allow the empty set ϕ to appear on either side of a sequent, and abbreviate $\phi \Rightarrow \Delta$ by $\Rightarrow \Delta$, $\Gamma \Rightarrow \phi$ by $\Gamma \Rightarrow$, and $\phi \Rightarrow \phi$ by $\Rightarrow$. By the above definition, $\Rightarrow \Delta$ is true (in an interpretation) if and only if one of δ_i is true, $\Gamma \Rightarrow$ is true if and only if one of γ_i is false, and $\Rightarrow$ is never true in any interpretation.

Block Tableaux for L_0

The proof method we adopt is similar to Gentzen's original sequent calculus, but simpler in form. It is called the ***method of block tableaux***, and was orginated by Hintikka [13]. A useful reference is Smullyan [40], in which many results in block tableaux and similar systems are presented in a unified form.

A block tableau system consists of axioms and rules (collectively, *postulates*) whose formal objects are sequents. Block tableau rules are like upside-down inference rules: the conclusion comes first, next a horizontal line, then the premises. Block tableaux themselves are derivations whose root is the sequent derived, whose branches are given by the rules, and whose leaves are axioms. Block tableaux look much like upside-down Gentzen system trees. (A more formal definition of block tableaux is given below).

We consider a system T_0 (see Smullyan [40], pp. 105–109) that is first-order sound and complete: its consequences are precisely the sentences true in every first-order valuation.

DEFINITION 4.2. *The system T_0 has the following postulates.*

Axioms. $\Gamma, p \Rightarrow \Delta, p$

Conjunction Rules.

$$C_1: \quad \frac{\Gamma, p \wedge q \Rightarrow \Delta}{\Gamma, p, q \Rightarrow \Delta}$$

$$C_2: \quad \frac{\Gamma \Rightarrow \Delta, p \wedge q}{\Gamma \Rightarrow \Delta, p \qquad \Gamma \Rightarrow \Delta, q}$$

Disjunction Rules.

$$D_1: \quad \frac{\Gamma \Rightarrow \Delta, p \vee q}{\Gamma \Rightarrow \Delta, p, q}$$

$$D_2: \quad \frac{\Gamma, p \vee q \Rightarrow \Delta}{\Gamma, p \Rightarrow \Delta \qquad \Gamma, q \Rightarrow \Delta}$$

Implication Rules.

$$I_1: \quad \frac{\Gamma \Rightarrow \Delta, p \supset q}{\Gamma, p \Rightarrow \Delta, q}$$

$$I_2: \quad \frac{\Gamma, p \supset q \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, p \qquad \Gamma, q \Rightarrow \Delta}$$

Negation Rules.

$$N_1: \quad \frac{\Gamma \Rightarrow \Delta, \neg p}{\Gamma, p \Rightarrow \Delta}$$

$$N_2: \quad \frac{\Gamma, \neg p \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, p}$$

Universal Rules.

$$U_1: \quad \frac{\Gamma, \forall x.\alpha(x) \Rightarrow \Delta}{\Gamma, \alpha(x/a), \forall x.\alpha(x) \Rightarrow \Delta}$$

$$U_2: \quad \frac{\Gamma \Rightarrow \forall x.\alpha(x), \Delta}{\Gamma \Rightarrow \alpha(x/a), \forall x.\alpha(x), \Delta}, \quad \textit{where } a \textit{ has not appeared in the tableau}$$

Existential Rules.

$$E_1: \quad \frac{\Gamma \Rightarrow \exists x.\alpha(x), \Delta}{\Gamma \Rightarrow \alpha(x/a), \exists x.\alpha(x), \Delta}$$

$$E_2: \quad \frac{\Gamma, \exists x.\alpha(x) \Rightarrow \Delta}{\Gamma, \alpha(x/a), \exists x.\alpha(x), \Rightarrow \Delta}, \quad \textit{where } a \textit{ has not appeared in the tableau}$$

Remarks. Note the simple form of the axioms and the symmetric nature of the inference rules (actually, each rule is a rule schema, since Γ, Δ, p, q, and α stand for formulas and sets of formulas of L_0). There is one rule that deletes each logical connective on either side of the sequent. For example, the first conjunction rule deletes a conjunction on the left side of a sequent in favor of the two conjoined sentences; informally, it can be read as "Δ follows from Γ and $p \wedge q$ if it follows from Γ, p, and q." It is easily verified that each rule is *sound* with respect to first-order valuations: if the premises are true in an interpretation, then so is the conclusion.

DEFINITION 4.3. *A block tableau for the sequent $\Gamma \Rightarrow \Delta$ in a system $\mathcal{T}$ is a tree whose nodes are sequents, defined inductively as follows.*

1. *$\Gamma \Rightarrow \Delta$ is the root of the tree.*

2. *If sequent s is the parent node of daughters $s_1 \ldots s_n$, then* $\dfrac{s}{s_1 \ldots s_n}$ *is a rule of* $\mathcal{T}$.

A block tableau is closed *if all its leaves are axioms. If there is a closed block tableau for the sequent* $\Gamma \Rightarrow \Delta$, *then this sequent is a* theorem *of the system* $\mathcal{T}$ *and we write* $\vdash_{\mathcal{T}} \Gamma \Rightarrow \Delta$.

A system $\mathcal{T}'$ *is called a* subsystem *of* $\mathcal{T}$ *if every rule of* $\mathcal{T}'$ *is also a rule of* $\mathcal{T}$. *If some subsystem* $\mathcal{T}'$ *of* $\mathcal{T}$ *has exactly the same theorems as* $\mathcal{T}$, *then the rules of* $\mathcal{T}$ *not appearing in* $\mathcal{T}'$ *are said to be* eliminable *from* $\mathcal{T}$, *or* admissible *to* $\mathcal{T}'$.

Block tableaux are similar to the AND/OR trees commonly encountered in AI theorem-proving systems (see Nilsson [32]). Rules C_2, D_2, and I_2 cause AND-splitting, while a choice of rules to apply at a tableau node is an OR-split.

Example. Here is a block tableau for the sequent $\exists x.\, Bx \wedge Ax,\ \forall x.\, Cx \supset \neg Bx \Rightarrow \exists x.\, Ax \wedge \neg Cx$.

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\exists x.\, Bx \wedge Ax,\ \forall x.\, Cx \supset \neg Bx \Rightarrow \exists x.\, Ax \wedge \neg Cx}{Be \wedge Ae,\ \forall x.\, Cx \supset \neg Bx \Rightarrow \exists x.\, Ax \wedge \neg Cx}\,E_2}{Be \wedge Ae,\ Ce \supset \neg Be \Rightarrow \exists x.\, Ax \wedge \neg Cx}\,U_1}{Be \wedge Ae,\ Ce \supset \neg Be \Rightarrow Ae \wedge \neg Ce}\,E_1}{Ae,\ Be,\ Ce \supset \neg Be \Rightarrow Ae \wedge \neg Ce}\,C_1}{\dfrac{\dfrac{Ae,\ Be,\ \neg Be \Rightarrow Ae \wedge \neg Ce}{Ae,\ Be \Rightarrow Be,\ Ae \wedge \neg Ce}\,N_2}{\times}\,I_2 \qquad \dfrac{\dfrac{\dfrac{Ae,\ Be \Rightarrow Ce,\ Ae \wedge \neg Ce}{Ae,\ Be \Rightarrow Ce,\ \neg Ce}\,C_2}{Ae,\ Be,\ Ce \Rightarrow Ce}\,N_1}{\times} \qquad \dfrac{Ae,\ Be \Rightarrow Ce,\ Ae}{\times}}
$$

The sequent to be proved is inserted as the root of the tree. By a series of reductions based on the rules of $\mathcal{T}_0$, the atoms of the sequent's sentences are extracted from the scope of quantifiers and Boolean operators. Splitting of the tree occurs at the rules I_2 and C_2; otherwise the reduction produces just a single sequent below the line. If a tree is found where the sequents at all the leaves are axioms, then the theorem is proved. Note that the logical inferences are from the leaves to the root of the tree, even though we work backwards in forming the tree. At each junction of the tree, the parent sequent is true in an interpretation if all its daughters are true in that interpretation.

An important connection between theoremhood and logical consequence for sequent systems is the following soundness theorem for tableaux.

THEOREM 4.1. *If $\Gamma \Rightarrow p$ is a theorem of $\mathcal{T}$ (where p is a single sentence of L_0), and all the rules of $\mathcal{T}$ are sound, then p is a logical consequence of Γ.*

Proof. If the rules of $\mathcal{T}$ are sound, then every theorem of $\mathcal{T}$ is valid. By Definition 4.1, this means that in every interpretation in which all of Γ are true, p must be also.∎

4.2. The Language of B

The language of B is formed from a first-order base language L_0 by adding modal operators for belief and belief circumscription. We call this language L^{B}. It is convenient to use L^{B} also as the agents' language L, since it provides a representation for nested beliefs as required by the recursion property. With this assumption, we can parameterize B by the base language L_0, and write $\mathsf{B}(L_0, \rho)$ for the logic family.

To form L^{B} from a base language L_0, we require a countable set of agents (S_0, S_1, ...).

DEFINITION 4.4. *A sentence of L^{B} based on L_0 is defined inductively by the following rules.*

1. *All formation rules of L_0 are also formation rules of L^{B}.*
2. *If p is a sentence, then $[S_i]p$ is a sentence for $i \geq 0$.*
3. *If p is a sentence and Γ is a finite set of sentences, then $\langle S_i : \Gamma \rangle p$ is a sentence for $i \geq 1$.*

An *ordinary atom* of L^{B} is a ground atom of L_0; a *belief atom* is a sentence of the form $[S_i]p$, and a *circumscriptive atom* is one of the form $\langle S_i : \Gamma \rangle p$. In the belief atom $[S_i]p$, p is said to be *in the context of* the belief operator. Note that there is no quantification into the contexts of belief atoms, since the argument of a belief operator is always a closed sentence. L^{B} can be extended to include quantification into belief contexts; such a language has greater representational power and its logic qB has a more complex axiomatization. The interested reader is referred to Konolige [21] for a description of qB. Here, the simpler B is sufficient for an analysis of the problems.

We will use the abbreviation $[S]\Gamma =_{\mathrm{df}} [S]\gamma_1, [S]\gamma_2, \ldots$.

Interpretations

Interpretations of the language of L^{B} are formed from interpretations of its base language L_0, together with an interpretation of belief and circumscriptive atoms. The intended meaning of the belief atom $[S_i]p$ is that p is in the belief subsystem of agent S_i; informally, we would say "S_i believes p." Since we are formalizing belief subsystems by means of deduction structures, an interpretation of the belief atoms $[S_i]p$ is given by a deduction structure d_i. $[S_i]p$ is *true* if p is in bel(d_i), the belief set of d_i; otherwise it is *false*.

In addition to representing beliefs of individuals, we use belief atoms to represent common beliefs. A common belief is one that every agent believes, and every agent believes every other agent believes, and so on to arbitrary depths of belief nesting. We reserve the name S_0 for a fictional agent whose beliefs are taken to be common among all agents. The belief atom $[S_0]p$ means that p is a common belief. In terms of deduction structures, its intended interpretation is that p and $[S_0]p$ are in the deduction structure d_i of every agent S_i, $i \geq 0$.

McCarthy (see, for example, [25]) was the first to recognize the common knowledge could be represented by the use of a fictitious agent FOOL whose knowledge "any fool" would know. He used a possible-worlds semantics for knowledge, and so all consequences of common knowledge were also known. The representation of common belief presented here uses an obviously similar approach; it differs only in that common belief rather than common knowledge is axiomatized (common beliefs need not be true), and in having a deduction structure semantics, so that common beliefs need not be closed under logical consequence.

The interpretation of circumscriptive atoms is also given by the deduction structure representing an agent's beliefs. The intended meaning of $(S_i : \Gamma)p$ is that p is derivable from Γ in the deduction structure d_i, that is, $\Gamma \vdash_{\rho(i)} p$. The circumscription operator elevates the belief derivation process to a first-class entity of the language (as opposed to belief operators $[S_i]$, which simply state that certain sentences are in or not in the belief set).

While it may not be apparent at first glance, the circumscription operator is a powerful tool for representing situations of delimited knowledge. For example, to formally state the condition, "the only facts that agent S knows about proposition p are F," we could use

$$(4.1) \qquad \langle S : F \rangle p \equiv [S]p \quad .$$

This assertion states that S believing p is equivalent to S being able to derive p from F. The forward implication is uninteresting, since it just says that p is derivable from F by agent S, *i.e.*, $[S]F \supset [S]p$. The reverse implication is more interesting, since it states p cannot be a belief of S *unless* it is derivable from F. This reverse implication limits the information S has available to derive p to the sentences F, and thus gives the circumscriptive content of (4.1). Note that there is no way to formulate the reverse implication as a sentence of L^{B} using only belief operators.

The reader should note carefully that the semantics of L^{B} differs completely from that of most modal languages, in which the argument to the modal operator is usually taken to denote a *proposition* that can take on a truth-value in a possible world. By contrast, arguments to modal operators in the language of B denote *sentences* of L, namely themselves. It is important to keep this distinction in mind when interpreting the modal operators of B.

4.3. A Sequent System for B

The deductive process that underlies the deduction model is characterized in very general terms by deduction structures and their associated belief sets. Until now we have been content with deliberate vagueness about the exact nature of deduction rules and the derivation process. As stated in Section 3, there are five conditions that must be satisfied: the deduction rules must be *effective*, *provincial*, and *sound*, and the derivations *reflexive* and *closed under deduction*. Consider a deduction structure $d_i = \langle B, \rho(i) \rangle$ for agent S_i. If we let the process of belief derivation for d be symbolized by $\vdash_{\rho(i)}$, these conditions are as follows.

(Effectiveness) The deduction rules $\rho(i)$ are effectively applicable.

(Provinciality)	The number of input sentences to each rule is finite and bounded.
(Soundness)	If $\Gamma \not\vdash_{\rho(i)} \alpha$, then α is a logical consequence of Γ.
(Reflexivity)	$\alpha \not\vdash_{\rho(i)} \alpha$.
(Closure)	If $\Gamma \not\vdash_{\rho(i)} \beta$ and $\beta, \Sigma \not\vdash_{\rho(i)} \alpha$, then $\Gamma, \Sigma \not\vdash_{\rho(i)} \alpha$.

Suppose we are given beforehand a derivation operator $\not\vdash_{\rho(i)}$, satisfying the above conditions, that models an agent S_i's belief subsystem. The central problem in the formulation of B is to find tableau rules that correctly implement the meaning of the belief operator $[S_i]$ and the circumscription operator $\langle S_i : \Gamma \rangle$ under $\not\vdash_{\rho(i)}$.

Consider first the sequent $[S_i]\Gamma \Rightarrow [S_i]\alpha$. Its intended meaning is that, if all of Γ are in S_i's belief set, then so is α. The only possible way that we can guarantee this condition is if α is derivable from Γ, *i.e.*, $\Gamma \not\vdash_{\rho(i)} \alpha$. If this were not the case, then we could always construct the counterexample $d_i =_{df} \langle \Gamma, \rho(i) \rangle$ in which all of Γ are in d_i, but α is not. Thus we can relate the truth of a sequent involving belief operators to derivability in an agent's belief subsystem. This relation is captured by the inference rule

$$A: \qquad \frac{\Sigma, [S_i]\Gamma \Rightarrow [S_i]\alpha, \Delta}{\Gamma \not\vdash_{\rho(i)} \alpha} \quad .$$

A is called the *attachment rule*, because it derives results involving the belief operator by attaching sentences about belief to the actual derivation process of an agent. Remembering that the premise is the bottom sequent and the conclusion the top, we can read A informally as follows: "If α is a deductive consequence of Γ in S_i's belief subsystem, then, whenever S_i believes Γ, he also believes α."

To capture the notion of common belief, we need to make a modification to the attachment rule. The intended meaning of the common belief atom $[S_0]q$ is that both q and $[S_0]q$ are in the belief subsystem of every agent. The sequent $[S_0]\Lambda, [S_i]\Gamma \Rightarrow [S_i]\alpha$ will be true if whenever $[S_0]\Lambda$, Λ, and Γ are in the belief set of d_i, α also is. By reasoning similar to that used in deriving the rule A, we can rephrase this in terms of belief derivation. This yields the revised attachment rule A^{CB}.

$$A^{CB}: \qquad \frac{\Sigma, [S_0]\Lambda, [S_i]\Gamma \Rightarrow [S_i]\alpha, \Delta}{[S_0]\Lambda, \Lambda, \Gamma \not\vdash_{\rho(i)} \alpha}$$

In A^{CB}, both A and $[S_0]A$ can be used in the derivation of α. Note that this rule is applicable to the fictional agent S_0. Because S_0's beliefs are intended to be common beliefs, and hence derivable by any agent, it should be the case that the rules $\rho(0)$ are used by every agent. We thus demand that $\rho(0) \subseteq \rho(i)$ for every i.

We can find tableau rules for the circumscription operator in a similar manner. The intended semantics of this operator relates directly to the belief derivation process: $\langle S_i : \Gamma \rangle p$ means that p is derivable from Γ in S_i's belief subsystem, *i.e.*, $\Gamma \vdash_{\rho(i)} p$. In writing sequent rules, there are two cases to consider, for a circumscriptive atom can appear on the right or left side of the sequent arrow. We thus have the following two rules.

$$Circ_1 : \qquad \frac{\Sigma \Rightarrow \langle S_i : \Gamma \rangle p, \Delta}{\Gamma \vdash_{\rho(i)} p}$$

$$Circ_2 : \qquad \frac{\Sigma, \langle S_i : \Gamma \rangle p \Rightarrow \Delta}{\Gamma \nvdash_{\rho(i)} p}$$

The second circumscription rule is the one that is used to show circumscriptive ignorance. It states that if p is not derivable from a set of sentences Γ, then the circumscriptive atom $\langle S_i : \Gamma \rangle p$ is false. Given a statement of the form 4.1, this in turn would imply that S_i was ignorant of p.

We can now give a full axiomatization of the logic family B.

DEFINITION 4.5. *The system* $\mathsf{B}(L_0, \rho)$ *has the following postulates.*

1. *The first-order complete rules* T_0.
2. *The rules* A^{CB}, $Circ_1$, *and* $Circ_2$.
3. *A closed derivation process* $\vdash_{\rho(i)}$ *for each agent* S_i, *such that* $\rho(0) \subseteq \rho(i)$ *for every* i.

This axiomatization of B is both sound and complete with respect to its deduction structure semantics, as proven in Konolige [21]. It is a compact formalization of the deduction model and useful for theoretical investigations, but we do not use it very much as a representational formalism because of the general nature of the belief deduction process $\vdash_{\rho(i)}$, which is rather opaque to further analysis. For instance, we might wish to look at

the subfamily of B in which the rules of $\rho(i)$ that govern nested belief are as strong as A. In order to explore the fine structure of S_i's belief deduction process, or to formalize the problems, we need to fix the nature of $\not\vdash_{\rho(i)}$ more precisely. The rich set of rules, and the flexibility of tableau derivations, make tableau systems a natural choice here. In the next section we define a particularization of B, the logic family BK, whose belief derivation process is defined in block tableaux terms.

4.4. The Nonintrospective Logic Family BK

In the logic family BK, the belief derivation operator $\not\vdash$ is defined as provability in a tableau system.

DEFINITION 4.6. *A sentence α is* BK*-derivable from premises* Γ ($\Gamma \not\vdash_{\mathcal{T}} \alpha$) *if and only if* $\vdash_{\mathcal{T}} \Gamma \Rightarrow \alpha$.

We need to show that tableau system derivability as just defined satisfies the five criteria of belief derivation: effectiveness, provinciality, soundness, reflexivity and closure. Consider a sequent system $\mathcal{T}$ made up of sound tableau rules. According to Theorem 4.1, the theorem $\vdash_{\mathcal{T}} \Gamma \Rightarrow p$ of $\mathcal{T}$ implies that p is a logical consequence of Γ, so we are assured that $\vdash_{\mathcal{T}}$ satisfies the soundness criterion. Provinciality and effectiveness are also satisfied, since the theorems of $\mathcal{T}$ are built by using effectively computable steps that operate on a bounded number of sentences at each step. The observant reader might object at this point that tableau rules may indeed refer to an unbounded number of premise sentences; *e.g.*, any of the rules of $\mathcal{T}_0$ have this property, since Γ and Δ can stand for any set of sentences. However, each rule of $\mathcal{T}_0$ is actually a rule *schema*: the capital Greek letters are metavariables that are instantiated with a boundedly finite set of sentences to define a rule.

The closure condition is fulfilled by a special subclass of sequent systems, namely those for which the following rule, *Cut**, is admissible:

$$Cut^*: \quad \frac{\Gamma, \Sigma \Rightarrow \alpha}{\Gamma \Rightarrow \beta \qquad \beta, \Sigma \Rightarrow \alpha} \quad .$$

To see how this rule guarantees closure, suppose that $\Gamma \Rightarrow \beta$ and $\beta, \Sigma \Rightarrow \alpha$ are both theorems of a sequent system $\mathcal{T}$ for which *Cut** is admissible. Because *Cut** is admissible

and both of its premises have closed tableaux, the conclusion $\Gamma, \Sigma \Rightarrow \alpha$ must also be a theorem.

Finally, the derivation process will be reflexive ($\alpha \vdash_{\mathcal{T}} \alpha$) if we include the following axiom in the system $\mathcal{T}$:

$$Id: \quad \Sigma, \alpha \Rightarrow \alpha, \Delta \quad .$$

Thus we only allow a system $\mathcal{T}$ to appear in a deduction structure $d\langle B, \mathcal{T}\rangle$ if the system is sound, Cut^* is an admissible rule of $\mathcal{T}$, and Id is an axiom of $\mathcal{T}$.

An interesting consequence of using tableau derivations in BK is that the attachment rule A can now be expressed wholly in terms of sequents, eliminating the derivation operator. To see how this comes about, consider first replacing the belief operator in rule A by tableau provability, as given by Definition 4.6. This yields

$$AK': \quad \frac{\Sigma, [S_i]\Gamma \Rightarrow [S_i]\alpha, \Delta}{\vdash_{\tau(i)} \Gamma \Rightarrow \alpha} \quad ,$$

where $\tau(i)$ is the set of tableau rules used by agent S_i.

Now $\vdash_{\tau(i)} \Gamma \Rightarrow \alpha$ is true precisely if there is a closed tableau for $\Gamma \Rightarrow \alpha$, using the rules $\tau(i)$. Hence we should be able to eliminate the provability symbol if we add the rules $\tau(i)$ to B for the purpose of constructing a tableau for $\Gamma \Rightarrow \alpha$. In order to keep the agents' rules $\tau(i)$ from being confused with the rules of B, we add an agent index to sequents to indicate that the tableau rules to be use are for a particular agent. The final version of the attachment rule is

$$AK: \quad \frac{\Sigma, [S_i]\Gamma \Rightarrow [S_i]\alpha, \Delta}{\Gamma \Rightarrow_i \alpha} \quad .$$

Agents' rules are expressed using the indexed sequent sign, e.g., if agent S_i were to use C_2, the following rule would be added to B.

$$C_2^i: \quad \frac{\Gamma \Rightarrow_i \Delta, p \wedge q}{\Gamma \Rightarrow_i \Delta, p \qquad \Gamma \Rightarrow_i \Delta, q}$$

Taking the recursion property of belief subsystems seriously, we can iterate the process just described for the attachment rule. Each agent treats other agents as having

a set of tableau rules. In formulating BK, there will be a tableau rule set associated with each *view* (views are discussed in relation to the recursion property in Section 3.3). Let us symbolize the set of tableau rules representing the view ν by $\tau(\nu)$.

A sequent $\Gamma \Rightarrow_\nu \Delta$, with index ν, is a statement about the belief subsystem of the view ν. For example, if $\nu = Sue, Kim$, the sequent $\Gamma \Rightarrow_\nu p$ states that p follows from Γ in Sue's view of Kim's belief subsystem. The deduction rules $\tau(\nu)$ always have sequents indexed by ν in their conclusions (above the line). This assures us that they will always be used as rules of the belief subsystem ν, and of no other.

The logic BK can thus be parameterized by a set of tableau rules for each view, and we write $\mathsf{BK}(L_0, \tau)$ to indicate this. If the sequent $\Gamma \Rightarrow_\nu \Delta$ is a theorem of the logic $\mathsf{BK}(L_0, \tau)$, it asserts that the sequent $\Gamma \Rightarrow \Delta$ is provable in the view ν. We write this as $\vdash_{\mathsf{BK}(L_0,\tau)} \Gamma \Rightarrow_\nu \Delta$. If this sequent is a theorem for every parameterization of BK, we write simply $\vdash \Gamma \Rightarrow_\nu \Delta$. Note that the presence of the index on the sequent means that we do not have to state explicitly that the set of rules used to derive the theorem were those of the view ν. Properties of the the actual belief subsystems are always stated using unindexed sequent; for example, to show formally that if an agent believes p, then he believes q, we would have to prove that the sequent $[S_i]p \Rightarrow [S_i]q$ is a theorem of BK.

Postulates of $\mathsf{BK}(L_0, \tau)$

This family is parameterized by a base language L_0 and tableau rules $\tau(\nu)$ for each view ν.

DEFINITION 4.7. *The system* $\mathsf{BK}(L_0, \tau)$ *is given by the following postulates:*

1. *The first-order complete rules* $\mathcal{T}_0$.
2. *The attachment rule*

$$AK^{\mathsf{CB}}: \quad \frac{\Sigma, [S_0]\Lambda, [S_i]\Gamma \Rightarrow [S_i]\alpha, \Delta}{[S_0]\Lambda, \Lambda, \Gamma \Rightarrow_i \alpha} \quad .$$

3. *A set of sound sequent rules* $\tau(\nu)$ *for each view* ν *which contains the axiom Id, and for which the rule Cut* is admissible. Also,* $\tau(\nu, 0) \subseteq \tau(\nu, i)$ *for all views* ν *and agents* S_i.

4. *The circumscription rules*

$$CircK_1: \quad \frac{\Sigma \Rightarrow \langle S_i : \Gamma \rangle p, \Delta}{\Gamma \Rightarrow_i p}$$

and

$$CircK_2: \quad \frac{\Sigma, \langle S_i : \Gamma \rangle p \Rightarrow \Delta}{\not\vdash \Gamma \Rightarrow_i p} \quad .$$

Remarks. There are three parts to the system $\mathsf{BK}(L_0, \tau)$. The first part is a set of rules that perform first-order deductions about the real world. These rules incorporate the nonsubscripted sequent sign ($\Rightarrow$).

The second part is the attachment rule AK^{CB}, together with a set of rules formalizing the deductive system of each view. These rules involve the sequent sign $\Rightarrow_\nu$, since they talk about agents' deductive systems. They can contain rules that have a purely nonmodal import (*e.g.*, rules of τ_0), as well as rules that deal with belief operators. The rule Cut^*, which implements the closure property of belief sets, must be an admissible rule of $\tau(\nu)$.

The rules $\tau(\nu)$ of a view ν can be incomplete in several ways. They may be first-order incomplete, in which case they cannot be used to draw all the consequences of sentences involving nonmodal operators that they otherwise might (to be first-order complete, it is sufficient for the rules τ_0 to be admissible in a view). They may also be incomplete with respect to the semantics of sentences involving belief operators. To be complete in this respect, a sufficient rule would be AK^{CB}. A view for which this rule is admissible is called *recursively complete.* If every view of a logic $\mathsf{BK}(L_0, \tau)$ is recursively and first-order complete, the logic is called *saturated.* We will symbolize the subfamily of saturated logics by BK_s.

The rule AK^{CB} is a weak version of the attachment rule A^{CB} in that it makes no assumptions about the beliefs an agent may have of his own beliefs. For example, we might argue that, if an agent S believes a proposition P, then he believes that he believes it. All he has to do to establish this is query his belief subsystem with the question, "Do I believe P?" If the answer comes back "yes," then he should be able to infer that he does indeed believe P, *i.e.*, $[S][S]P$ is true if $[S]P$ is. However, as far as rule AK^{CB} is concerned, an

agent's own belief subsystem has the same status for him as does that of any other agent. In particular, AK^{CB} allows an agent to have false and incomplete beliefs about his own beliefs. Other version of AK^{CB} with stronger assumptions about self-belief are possible (see Section 6).

The third part consists of the two circumscription rules. The provability operator can be eliminated from $CircK_1$, but not from $CircK_2$. In order to show that p does not follow from Γ for S_i, we must show that there is no closed tableau for $\Gamma \Rightarrow_i p$. One technique that we use in solving the problems is the following. If there is no closed tableau for a *saturated* logic of BK, there is no closed tableau for any logic of BK. Every theorem of saturated BK is a theorem of the normal modal system $K4$ (see Section 6), which has a decision procedure based on the methods of Sato (in [38]). Thus if a sequent is not provable in $K4$, it is not provable in any logic of BK.

Some Theorems of BK

THEOREM 4.2. *Let p be derivable from Γ in the view i of* $\mathsf{BK}(L_0, \tau)$. *Then*

$$\vdash_{\mathsf{BK}(L_0,\tau)} [S_i]\Gamma \Rightarrow [S_i]p \quad .$$

Proof. In one step, using rule AK^{CB} :

$$AK^{\mathrm{CB}} \quad \frac{[S_i]\Gamma \Rightarrow [S_i]p}{\Gamma \Rightarrow_i p} \quad \times$$

∎

THEOREM 4.3. *Let ν be a recursively complete view of* $\mathsf{BK}(L_0, \tau)$, *and let p be derivable from Γ in the view ν, i. Then*

$$\vdash_{\mathsf{BK}(L_0,\tau)} [S_i]\Gamma \Rightarrow_\nu [S_i]p \quad .$$

Proof. In one step, using rule AK^{CB} *of* $\tau(\nu)$:

$$AK^{\mathrm{CB}} \quad \frac{[S_i]\Gamma \Rightarrow_\nu [S_i]p}{\Gamma \Rightarrow_{\nu,i} p} \quad \times$$

Remarks. These two theorems show that BK has a weakened analog of the necessitation rule of modal logic (if α is provable, so is $\Box\alpha$). If a nonmodal sentence α is provable in the view i (*i.e.*, $\vdash_{\mathsf{BK}(L_0,r)} \Rightarrow_i \alpha$), then, by Theorem 4.2, $[S_i]\alpha$ is provable in the empty view. Since the theorems of $r(i)$ are assumed to be sound, α is a tautology, and so must be provable in the empty view.[1] Hence, for those tautologies provable in the view i, necessitation holds. Theorem 4.3 establishes this result for an arbitrary view in which A is an admissible rule. Depending on the exact nature of the rule sets r, necessitation will hold for some subset of the provable sentences of a particular logic $\mathsf{BK}(L_0, r)$.

THEOREM 4.4. $\not\vdash [S_i]p \Rightarrow p$

Proof. If p is a primitive sentence, then there is no applicable tableaux rule, and hence no closed tableaux for the sequent.∎

Remarks. The familiar modal logic principle $\Box p \supset p$ (if p is necessary, then p is true) is not a theorem of BK, since beliefs need not be true.

THEOREM 4.5. $\not\vdash [S_i]p \Rightarrow [S_i][S_i]p$

Proof. The only applicable rule is AK^{CB} :

$$AK^{\mathrm{CB}} \quad \frac{[S_i]p \Rightarrow [S_i][S_i]p}{p \Rightarrow_i [S_i]p}$$

According to the semantics of the deduction model, the sequent $p \Rightarrow_i [S_i]p$ is not valid: just because a sentence p is true does not mean that an agent S_i believes it. Hence, there cannot be any set of sound tableau rules for $r(i)$ that causes $p \Rightarrow_i [S_i]p$ to close.∎

[1] Care must be taken in restricting α to nonmodal sentences, since the semantics of modal operators can change from one view to another (see the discussion of the recursion property in Section 3.3). John may believe perfectly well that Sue's belief subsystem can prove a certain fact, whereas in actuality her inference rules are too weak.

THEOREM 4.6. $\not\vdash \neg[S_i]p \Rightarrow [S_i]\neg[S_i]p$

Proof. We can apply either N_2 or AK^{CB}. If we apply the latter, we obtain

$$AK^{\mathrm{CB}} \quad \frac{\neg[S_i]p \Rightarrow [S_i]\neg[S_i]p}{\Rightarrow_i \neg[S_i]p} \quad .$$

deduction model, since it would require that no agent believe any sentence. Hence there can be no set of sound tableau rules $r(i)$ that derives it.

If we apply N_2 first, we obtain

$$N_2 \quad \frac{\neg[S_i]p \Rightarrow [S_i]\neg[S_i]p}{\Rightarrow [S_i]p, [S_i]\neg[S_i]p} \quad .$$

There are now two ways to apply AK^{CB}. In one application, we generate the sequent $\Rightarrow_i \neg[S_i]p$, which cannot close. In the other, we generate $\Rightarrow_i p$, which again cannot be derived by any set of sound tableau rules. ∎

Remarks. These theorems show that no logic of BK sanctions inferences about self-beliefs. If an agent believes p, it does not follow that his model of his own beliefs includes p; this is the import of Theorem 4.5. Similarly, if he does not believe p, he also may not have knowledge of this fact, as shown by Theorem 4.6.

THEOREM 4.7.

$$\vdash [S_0]p \Rightarrow [S_0][S_0]p$$

Proof.

$$AK^{\mathrm{CB}} \quad \frac{[S_0]p \Rightarrow [S_0][S_0]p}{[S_0]p, p \Rightarrow_i [S_0]p}$$

$$\times$$

∎

Remarks. We have proven a simple fact about common beliefs: if p is a common belief, it is a common belief that this is so.

For the circumscriptive ignorance part of BK, it is an interesting exercise to show that

(4.2) $\quad \langle S_i : \Gamma \rangle p \Rightarrow [S_i]\Gamma \supset [S_i]p$

holds, but the converse doesn't. That is, if p follows from Γ for agent S_i, it must be the case that believing Γ entails believing p; on the other hand, it may be that every time an agent has Γ in his base set he also has p, which would satisfy $[S_i]\Gamma \supset [S_i]p$ without having p derivable from Γ.

THEOREM 4.8. $\vdash \langle S_i : \Gamma \rangle p \Rightarrow [S_i]\Gamma \supset [S_i]p$

Proof. We have the following two tableaux for this sentence.

$$AK^{CB}_{I_1}\ \frac{\langle S_i : \Gamma \rangle p \Rightarrow [S_i]\Gamma \supset [S_i]p}{\dfrac{\langle S_i : \Gamma \rangle p, [S_i]\Gamma \Rightarrow [S_i]p}{\Gamma \Rightarrow_i p}}$$

$$Circ_2^{I_1}\ \frac{\langle S_i : \Gamma \rangle p \Rightarrow [S_i]\Gamma \supset [S_i]p}{\dfrac{\langle S_i : \Gamma \rangle p, [S_i]\Gamma \Rightarrow [S_i]p}{\not\vdash \Gamma \Rightarrow_i p}}$$

Either p is derivable from Γ using the rules $r(i)$, or it isn't. In either case one of these tableaux closes. ∎

Example. we give an example of the use of the circumscription rules to show ignorance. Suppose the agent Sue believes only the sentences P and $P \supset Q$ in a situation; we want to show that she doesn't believe R. Thus we want to prove the sequent $\langle Sue : P, P \supset Q \rangle R \equiv [Sue]R \Rightarrow \neg[Sue]R$.

$$C_1\ \frac{\langle Sue : P, P \supset Q \rangle R \equiv [Sue]R \Rightarrow \neg[Sue]R}{\dfrac{\langle Sue : P, P \supset Q \rangle R \supset [Sue]R,\ [Sue]R \supset \langle Sue : P, P \supset Q \rangle R \Rightarrow \neg[Sue]R}{I_2\ \dfrac{\langle Sue : P, P \supset Q \rangle R \Rightarrow \neg[Sue]R}{Circ_2\ \not\vdash P, P \supset Q \Rightarrow_{Sue} R}} \qquad N_2\ \frac{\Rightarrow [Sue]R, \neg[Sue]R}{[Sue]R \Rightarrow [Sue]R}}$$

×

If the rules $r(Sue)$ are sound, there is no closed tableau for $P, P \supset Q \Rightarrow_i R$, and so both branches of the tableau close. Note that only the reverse implication half of the equivalence was needed.

5. The Problems Revisited

Using the logic BK, we present formal solutions to the two representational problems posed at the beginning of this section. In each case we have tried to avoid solutions that are trivial in the sense that they solve the representational problem, but only at the expense of excluding types of reasoning that might be expected to occur. For example, in the chess problem it would be an adequate but unrealistic solution to credit each player with no deduction rules at all. Instead, we try to find rules that allow a resource-limited amount of reasoning about the game to take place.

The Chess Problem

To approach this problem, we need to represent the game in a first-order language. Because the ontology of chess involves rather complicated objects (pieces, board positions, moves, histories of moves) we will not give a complete formalization, but rather sketch in outline how this might he done.

We use a multisorted first-order language L_e for the base language L_0. The key sorts will be those for players (S_w or S_b), moves, and boards. The particular structure of the sort terms is not important for the solution of this problem, but they should have the following information. A board contains the position of all pieces, and a history of the moves that were made to get to that position. This is important because we want to be able to find all legal moves from a given position; to do this, we have to have the sequence of moves leading up to the position, since legal moves can be defined only in terms of this sequence. For example, castling can only occur once, even if a player returns to the position before the castle; more importantly, there are no legal moves if 50 moves have been made without a capture or pawn advancement (this is what makes chess a finite game). A move contains

enough information so that it is possible to compute all successor boards, that is, those resulting from legal moves.

The game tree is a useful concept in exploring game-playing strategies. This is a finite tree (for finite games like chess) whose nodes are board positions, and whose branches are all possible complete games. A terminal node of the tree ends in either a win for White or Black, or a draw. The *game-theoretic value* of a node for a player is either 1 (a win), 0 (a draw), or -1 (a loss), based on whether that player can force a win or a draw, or his opponent can force a win. We use the predicate $M(p, b, k, l, r)$ to mean that board b has value k for player p. The argument l is a depth-of-search indicator, and shows the maximum depth of the game tree that the value is based on. We include the argument r so that M can represent heuristic information about the value of a node; when $r = \mathrm{f}$, k is the player's subjective estimate of the value of the node, *i.e.*, he has not searched to all terminal nodes of the game tree. If $r = \mathrm{t}$, then k is the game-theoretic value of the board.

We take the formal interpretation of boards, players, and the M predicate to be the game of chess, so that L_c is a partially interpreted language. The rules of the game of chess strictly specify what the game tree and its associated values will be; hence, each predication $M(p, b, k, l, \mathrm{t})$ or its negation is a valid consequence of these interpretations. Any agent who knows the rules of chess, and who has the concept of game trees, will know the game-theoretic value of every node if his beliefs are consequentially closed. In particular, he will believe either $M(S_w, I, 1, k, \mathrm{t})$ or $\neg M(S_w, I, 1, k, \mathrm{t})$, where I is the initial board; and so he will know whether White has an initial forced win or not.

We represent agents' knowledge of chess by giving tableau rules for L_c. The rules $\mathcal{T}_c$ presented below are one possible choice.

$$Ch_1: \quad \frac{\Gamma \Rightarrow M(p, b, k, l, r), \Delta}{\Gamma \Rightarrow M(p, b_1, k_1, l_1, r_1), \Delta \quad \Gamma \Rightarrow M(p, b_2, k_2, l_2, r_2), \Delta \quad \cdots \quad \Gamma \Rightarrow M(p, b_n, k_n, l_n, r_n), \Delta}$$

where b_1–b_n are *all* the legal successor boards to b
p's opponent is to move on b
k is the minimum of k_1–k_n
l is 1+ the maximum of l_1–l_n
r is t *iff* all of r_1–r_n are t

$$Ch_2: \frac{\Gamma \Rightarrow M(p,b,k,l,r),\Delta}{\Gamma \Rightarrow M(p,b_1,k_1,l_1,r_1),\Delta \quad \Gamma \Rightarrow M(p,b_2,k_2,l_2,r_2),\Delta \quad \cdots \quad \Gamma \Rightarrow M(p,b_n,k_n,l_n,r_n),\Delta},$$

where b_1–b_n are *all* the legal successor boards to b
p is to move on b
k is the maximum of k_1–k_n
l is 1+ the maximum of l_1–l_n
r is t *iff* all of r_1–r_n are t

Ch_3 : $\Gamma \Rightarrow M(p,b,k,0,\mathrm{t}),\Delta$, where $k = 1$ if p has a checkmate on his opponent on board b; $k = 0$ if board b is a draw; and $k = -1$ if p's opponent has a checkmate.

Ch_4 : $\Gamma \Rightarrow M(p,b,k,0,\mathrm{f}),\Delta$, where k is any number between -1 and 1

Ch_1 axiomatizes nodes in the game tree where p's opponent moves. The value of such a node is the minimum of the values of its successor nodes. The argument l is the maximum depth of the subtree searched. r will be t only if all the subtrees have been searched to leaf nodes. Ch_2 is similar to Ch_1, except p moves, and the maximum of the successor values is chosen.

Ch_3 is the rule for terminal nodes of the tree. Ch_4 is a rule for heuristic evaluation of any node; note that the last argument to M is f, which indicates that a terminal node has not been reached. Each agent may have his own particular heuristics for evaluating nonterminal nodes; we can accommodate this by changing the values for k in Ch_4.

As an example of the use of these rules, consider the following tableau proof.

(5.1)

$$Ch_1\ \frac{\Rightarrow M(S_w,b,1,2,\mathrm{t})}{\Rightarrow M(S_w,b_1;1,0,\mathrm{t})\ \times \qquad Ch_2\ \frac{\Rightarrow M(S_w,b_2,1,1,\mathrm{t})}{\Rightarrow M(S_w,b_3,0,0,\mathrm{t})\ \times \quad \Rightarrow M(S_w,b_4,1,0,\mathrm{t})\ \times} \qquad \Rightarrow M(S_w,b_5,1,0,\mathrm{t})\ \times}$$

This is a proof that the board b has a value 1 for White, searching to all terminal nodes. Boards b_1, b_2, and b_3 all have value 1, so an application of rule Ch_1 yields that value 1 for b (it is Black's turn to move on b). Boards b_1 and b_5 are terminal nodes that are checkmates for White. There are two legal moves from board b_2; one ends in a draw (b_3), the other in a win (b_4) for White. Since it is White's turn to move, rule Ch_2 applies.

The structure of this tableau proof mimics exactly the structure of the game tree from the board b. Indeed, for any subtree of the complete game tree of chess whose root the board b with value k for player p, there is a corresponding proof of $M(p, b, k, l, \mathrm{t})$ using the rules $\mathcal{T}_c$. In particular, if one of $M(S_w, I, 1, l, \mathrm{t})$, $M(S_w, I, 0, l, \mathrm{t})$, or $M(S_w, I, -1, l, \mathrm{t})$ is true, there is a proof of this fact. Hence the rules $\mathcal{T}_c$ are sufficient for a player to reason whether White has a forced initial win or not, given an infinite resource bound for derivations. If we model agents as having the rules $\mathcal{T}_c$, so that $\mathcal{T}_c \subseteq \tau(\nu)$ for every view ν, the conversation presented at the beginning of this paper would make sense: each agent would believe that everyone knew whether White had a forced initial win.

A simple modification of the rules Ch_1 and Ch_2 can restrict exploration of the entire game tree, while still allowing agents to reason about game tree values using the heuristic axioms Ch_4, or the terminal node axioms Ch_3 if the game subtree is small. All that is necessary is to add the condition that no rule is applicable when the depth l is greater than some constant N. S_w would still be able to reason about the game to depths less than or equal to N, but he could go no further. In this way, a deductively closed system can represent a resource-limited derivation process. The revised rules are

Ch_1' $\quad$ Ch_1, with the condition that $l \leq N$.
Ch_2' $\quad$ Ch_2, with the condition that $l \leq N$.

With these rules, the proof of (5.1) would still go through for $N \geq 2$, but a proof of $M(S_w, I, k, l, \mathrm{t})$ could not be found if N were low enough to stop search at a reasonable level of the game tree.

The solution to the chess problem illustrates the ability of the deduction model to represent resource bounds by the imposition of constraints on deduction rules. There are other workable constraints for this problem besides depth cutoff: for example, the number of nodes in the tree being searched could be kept below some minimum. Because the structure of proofs mimics the game tree, any cutoff condition that is based on the game tree could be represented by appropriate deduction rules.

The Not-So-Wise-Man Problem

For this problem we use a base language L_w containing only the three primitive propositions P_1, P_2, and P_3. P_i expresses the proposition that wise man S_i has a white spot on his forehead.

In the initial situation, no one has spoken except the king, who has declared that at least one spot is white. Axioms for this situation are

$$
\begin{array}{lll}
(W1) & P_1 \wedge P_2 \wedge P_3 & \\
(W2) & [S_0](P_1 \vee P_2 \vee P_3) & \\
(W3) & (P_i \supset [S_j]P_i) \wedge [S_0](P_i \supset [S_j]P_i), & i \neq j, \quad j \neq 0 \\
(W4) & (\neg P_i \supset [S_j]\neg P_i) \wedge [S_0](\neg P_i \supset [S_j]\neg P_i), & i \neq j, j \neq 0 \\
(C1) & \langle S_i : W2\text{–}4, P_j, P_k\rangle P_i \equiv [S_i]P_i, & i \neq j, k
\end{array}
$$

$W1$ describes the actual placement of the dots. $W2$ is the result of the king's utterance: it is a common belief that at least one spot is white. $W3$ and $W4$ are schemata expressing the wise men's observational abilities, including the fact that everyone is aware of each other's capabilities. $C1$ is the circumscriptive ignorance axiom: the only beliefs a wise man has about the color of his own spot are the three axioms W_2–W_4, plus his observation of the other two wise men's spots.

As an exercise of the formalism, especially the circumscription rules, let us show that all agents are ignorant of the color of their own spot in the initial situation.

(5.2)

$$
I_2\ \dfrac{C_1\ \dfrac{C1 \Rightarrow \neg[S_i]P_i}{[S_i]P_i \supset \langle S_i : W2\text{–}4, P_j, P_k\rangle P_i \Rightarrow \neg[S_i]P_i} \qquad CircK_1\ \dfrac{\langle S_i : W2\text{–}4, P_j, P_k\rangle P_i \Rightarrow \neg[S_i]P_i}{\not\vdash W2\text{–}4, P_j, P_k \Rightarrow_i P_i} \qquad N_1\ \dfrac{\Rightarrow [S_i]P_i, \neg[S_i]P_i}{[S_i]P_i \Rightarrow [S_i]P_i}}{\times}
$$

We have omitted some irrelevant sentences from the left side of sequents in this tableau. To show that it closes, we must be able to prove that there is no set of sound deduction rules that will enable S_i to deduce P_i from $W2$, $W3$, $W4$, P_j, and P_k. We can prove this

for any set of sound tableau rules by showing that $W2\text{–}4, P_j, P_k \Rightarrow_i P_i$ is not provable in the normal modal logic $K4$ (see Section 4.4). It is possible to find a $K4$-model in which the sequent $W2\text{–}4, P_j, P_k \Rightarrow_i P_i$ is false, using the methods of Sato [38]; hence this sequent is not provable in any logic of BK.

After the first wise man has spoken, it becomes a common belief that he does not know his own spot is white. The appropriate axioms are

$$(W5) \quad \neg[S_1]P_1 \wedge [S_0]\neg[S_1]P_1$$
$$(C2) \quad \langle S_i : W1\text{–}5, P_j, P_k\rangle P_i \equiv [S_i]P_i, \quad i \neq j, k$$

In this new situation, all the wise men are again ignorant of their own spot's color; we could prove this fact, showing that $\vdash C2 \Rightarrow \neg[S_i]P_i$, in a manner similar to the proof in (5.2). S_2 relates his failure to the others, and the new situation has the additional axiom

$$(W6) \quad \neg[S_2]P_2 \wedge [S_0]\neg[S_2]P_2 \quad .$$

The third wise man at this point does have sufficient cause to claim his spot is white, but only if the second wise man is indeed wise, and the third wise man believes he is. To see how this comes about, let us prove it in the saturated form of BK. We will take the wise men to be powerful reasoners, and set $\tau(\nu) = T_0 + AK^{CB} + CircK_1 + CircK_2$, for all views ν. The sequent we wish to prove is $W1\text{–}6 \Rightarrow [S_3]P_3$.

(5.3)

$$I_2\ \frac{\dfrac{}{P_2 \Rightarrow P_2}\times \qquad I_2\ \dfrac{\dfrac{}{P_1 \Rightarrow P_1}\times \qquad AK^{CB}\ \dfrac{C_1\ \dfrac{W2\text{–}6,\ P_1,\ P_2,\ P_3,\ [S_3]P_2,\ [S_3]P_1 \Rightarrow [S_3]P_3}{W2\text{–}6,\ P_1 \vee P_2 \vee P_3,\ P_2,\ P_1 \Rightarrow_3 P_3}}{C_1\ \dfrac{W2\text{–}6,\ P_1,\ P_2,\ P_3,\ [S_3]P_2,\ P_1 \supset [S_3]P_1 \Rightarrow [S_3]P_3}{W2\text{–}6,\ P_1,\ P_2,\ P_3,\ [S_3]P_2 \Rightarrow [S_3]P_3}}}{C_1\ \dfrac{W1\text{–}6 \Rightarrow [S_3]P_3}{W2\text{–}6,\ P_1,\ P_2,\ P_3 \Rightarrow [S_3]P_3}}}{W2\text{–}6,\ P_1,\ P_2,\ P_3,\ P_2 \supset [S_3]P_2 \Rightarrow [S_3]P_3}$$

This part of the proof is mostly bookkeeping. We have used some shortcuts in the proof, omitting some obvious steps and dropping sentences from either side of the sequent if they are not going to be used.

We now must show that S_3's belief subsystem can prove P_3 from the assumptions $W2$–6 and from the belief that the other two wise men's dots are white (note that we are now using S_3's sequent $\Rightarrow_3$).

(5.4)

$$
\dfrac{\dfrac{\dfrac{}{P_1 \Rightarrow_3 P_1}\,I_2}{\times}\qquad \dfrac{\dfrac{\dfrac{\dfrac{}{\Rightarrow_3 P_3, \neg P_3}\,I_2}{P_3 \Rightarrow_3 P_3}\,N_1}{\times}\qquad \dfrac{\dfrac{\dfrac{}{W2\text{–}6,\ P_1 \vee P_2 \vee P_3,\ P_2,\ P_1 \Rightarrow_3 P_3}\,C_1}{W2\text{–}6,\ P_1,\ P_2,\ P_1 \supset [S_2]P_1 \Rightarrow_3 P_3}\,I_2}{W2\text{–}6,\ P_1,\ P_2,\ [S_2]P_1,\ \neg P_3 \supset [S_2]\neg P_3 \Rightarrow_3 P_3}\,C_1}{W2\text{–}6,\ P_1,\ P_2,\ [S_2]P_1,\ [S_2]\neg P_3 \Rightarrow_3 P_3}\,N_2}{W2\text{–}6,\ P_1,\ P_2,\ [S_2]P_1,\ [S_2]\neg P_3 \Rightarrow_3 P_3,\ [S_2]P_2}\,AK^{\mathrm{CB}}}{W2\text{–}6,\ P_1 \vee P_2 \vee P_3,\ P_1,\ \neg P_3 \Rightarrow_{32} P_2}
$$

Note the atom P_3 on the right-hand side of the top sequent; it is equivalent to $\neg P_3$ on the left-hand side, *i.e.*, the assumption that S_3's spot is black. The sequent proof here mimics the third wise man's reasoning, *Suppose my spot were black* .... Through the observation axiom $W4$, which is a common belief, this assumption means that S_3 believes that S_2 believes $\neg P_3$. At this point, S_3 begins to reason about S_2's beliefs. Since, by $W6$, the second wise man is unaware of the color of his own spot, a contradiction will be derived if P_2 follows in S_2's belief subsystem.

(5.5)

$$
\dfrac{\dfrac{\dfrac{}{\neg P_3 \Rightarrow_{32} \neg P_3}\,I_2}{\times}\qquad \dfrac{\dfrac{\dfrac{\dfrac{}{\Rightarrow_{32} P_2, \neg P_2}\,I_2}{P_2 \Rightarrow_{32} P_2}\,N_1}{\times}\qquad \dfrac{\dfrac{\dfrac{}{W2\text{–}6,\ P_1 \vee P_2 \vee P_3,\ P_1,\ \neg P_3 \Rightarrow_{32} P_2}\,C_1}{W2\text{–}6,\ P_1,\ \neg P_3,\ \neg P_3 \supset [S_1]\neg P_3 \Rightarrow_{32} P_2}\,I_2}{W2\text{–}6,\ P_1,\ \neg P_3,\ [S_1]\neg P_3 \Rightarrow_{32} P_2}\,C_1}{W2\text{–}6,\ P_1,\ \neg P_3,\ [S_1]\neg P_3,\ \neg P_2 \supset [S_1]\neg P_2 \Rightarrow_{32} P_2}\,N_2}{W2\text{–}6,\ P_1,\ \neg P_3,\ [S_1]\neg P_3,\ [S_1]\neg P_2 \Rightarrow_{32} P_2}\,AK^{\mathrm{CB}}}{W2\text{–}6,\ P_1,\ \neg P_3,\ [S_1]\neg P_3,\ [S_1]\neg P_2 \Rightarrow_{32} P_2,\ [S_1]P_1,\ [S_1]\neg P_1}}{W2\text{–}6,\ P_1 \vee P_2 \vee P_3,\ \neg P_2,\ \neg P_3 \Rightarrow_{321} P_1}
$$

S_2's reasoning (in S_3's view) takes the assumption that the third wise man's spot is black and asks what the effect would be on the first wise man S_1. Since S_1 is also ignorant of the color of his own spot, a contradiction will ensue if the first wise man can prove that his own spot is white, under the assumption $\neg P_3$. The remainder of the proof is conducted in the view 321.

(5.6)

$$D_2 \quad \dfrac{N_2 \quad \dfrac{W2\text{–}6,\ P_1 \vee P_2 \vee P_3,\ \neg P_2,\ \neg P_3 \Rightarrow_{321} P_1}{W2\text{–}6,\ P_1 \vee P_2 \vee P_3,\ \Rightarrow_{321} P_1,\ P_2,\ P_3}}{\begin{array}{ccc} P_1 \Rightarrow_{321} P_1,\ P_2,\ P_3 & P_2 \Rightarrow_{321} P_1,\ P_2,\ P_3 & P_3 \Rightarrow_{321} P_1,\ P_2,\ P_3 \\ \times & \times & \times \end{array}}$$

In pursuing this proof, we have assumed that the second wise man is indeed wise. There are several places in which, with slightly less powerful deduction rules for the view 32, the proof would break down. Each of these corresponds to one of the two types of incompleteness that we identified in the statement of the problem: relevance incompleteness and fundamental logical incompleteness.

Consider first the notion that S_2 is not particularly good at reasoning about what other agents do not believe, a case of fundamental logical incompleteness. One way to capture this would be to weaken the rule N_2 in the following manner:

$$N_2': \quad \frac{\Gamma, \neg p \Rightarrow_{32} \Delta}{\Gamma \Rightarrow_{32} p, \Delta}, \quad \text{where } p \text{ contains no belief operators}$$

The modified rule N_2' would not allow deductions about what agents do not know. In particular, it would not allow the transfer of the sentence $\neg[S_1]P_1$ to the left-hand side of the sequent, a crucial step in the tableau (5.5) for the view $\Rightarrow_{32}$.

Note that the modified rule N_2' still allows deductions about what other agents do believe. For instance, if S_2 were asked whether S_1's believing P_1 followed from his believing $\neg P_2$ and $\neg P_3$, S_2 would say "yes," even with the logically incomplete rule N_2' (as in tableau (5.6) above).

A more drastic case of logical incompleteness would result if S_2 simply did not reason about the beliefs of other agents at all. In that case, one would exclude the rule AK^{CB} from S_2's deduction structure. Again, the proof would not go through, because the attachment rule could not be applied in the tableau (5.5).

The notion of relevance incompleteness emerges if the not-so-wise-man S_2 does not consider all the information he has available to answer the king. For example, he may

think that the observations of other agents are not relevant to the determination of his own spot, since the results of those observations are not directly available to him. The observational axioms $W3$ and $W4$ enter into the proof tableau (5.5) in two places. Both times the rule I_2 is used to break statements of the form $p \supset [S]p$ into their component atoms. Preventing the decomposition of $W3$ and $W4$ effectively prevents S_2 from reasoning about the observations of other agents. A weakened version of I_2 for doing this is:

$$I_2': \quad \frac{\Gamma, p \supset q \Rightarrow_{32} \Delta}{\Gamma \Rightarrow_{32} p, \Delta \qquad \Gamma, q \Rightarrow_{32} \Delta}, \quad \text{where } p \text{ and } q \text{ are both modal or both nonmodal.}$$

This rule is actually weaker than required for the purpose we have in mind. Consider the observation axiom $\neg P_3 \supset [S_1]\neg P_3$. There are two ways S_2 could use this axiom. If S_2 believes $\neg P_3$, he could conclude that S_1 does also. This is not the type of deduction we wish to prevent, since it means that S_2 attributes beliefs to other agents based on his own beliefs about the world. On the other hand, the axiom $\neg P_2 \supset [S_1]\neg P_2$ is used in a conceptually different fashion. Here it is the contrapositive implication: if S_1 actually does not believe $\neg P_2$, then P_2 must hold. The way this shows up in the proof tableau (5.5) is that $\neg P_3$ appears as an initial assumption on the sequent $W2$–5, $P_1, \neg P_3 \Rightarrow_{32} P_2$, while P_2 is a goal to be proved.

To capture the notion of using an implicational sentence in one direction only, we would have to complicate the deduction rules by introducing asymmetry between the left and right sides of the sequent. This is one of the major strategies used by commonsense theorem provers of the PLANNER tradition (Hewitt [12] originated this theorem-proving method). Rather than having implicational rules of the form I_2, typical PLANNER-type systems use something like the following rule.

$$PI: \quad \frac{\Gamma, p, p \supset q \Rightarrow \Delta}{\Gamma, p, q, p \supset q \Rightarrow \Delta}$$

The implicational sentence is used in one direction only in PI. If it is desired to make contrapositive inferences, then the contrapositive form of the implication must be included explicitly. The construction of PLANNER-type deduction rules within the tableau framework allows a much finer degree of control over the inference process. A full exposition of such a system is beyond the scope of this paper; the interested reader is referred to Konolige [21].

In sum, we have shown that it is possible for the deduction model to represent the situation in which not-so-wise-man has less than perfect reasoning ability, preventing the third wise man from figuring out the color of his own spot. Both relevance incompleteness and fundamental logical incompleteness can be captured by using appropriate rules for $r(32)$.

6. Other Formal Approaches to Belief

How does the deduction model and its logic B compare to other formal models and logics of belief? We examine two alternative approaches in this section: modal logics based on a Hintikka/Kripke possible-worlds semantics, and several different first-order formalizations that treat beliefs as sentences in an internal language.

6.1. The Possible-Worlds Model

The possible-worlds model of belief was initially developed by Hintikka in terms of sets of sentences he called *model sets*. Subsequent to Kripke's introduction of possible worlds as a uniform semantics for various modal systems, Hintikka rephrased his work in these terms (see Hintikka [14]). The basic idea behind this approach is that the beliefs of an agent are modeled as a set of possible worlds, namely, those that are *compatible with* his beliefs. For example, an agent who believes the sentences

(6.1) *Some of the artists are beekeepers.*
All of the beekeepers are chemists.

would have his beliefs represented as the set of possible worlds in which some artists are beekeepers and all beekeepers are chemists.

Representational Issues

In a possible world for which the sentences (6.1) are true, anything that is a valid consequence of (6.1) must also be true. There can be no possible world in which some artists are beekeepers, all beekeepers are chemists, and no artists are chemists; such a world is a logical impossibility. If beliefs are compatible with a set of possible worlds (*i.e.*, true of each such possible world), then every valid consequence of those beliefs is also compatible

with the set. Thus one of the properties of the possible-worlds model is that an agent will believe all consequences of his beliefs – the model is consequentially closed. Hintikka, recognizing this as a serious shortcoming of the model, claimed only that it represented an idealized condition: an agent could justifiably believe any of the consequences of his beliefs, although in any given situation he might have only enough cognitive resources to derive a subset of them.

The assumption of consequential closure limits the ability of the possible-worlds model to represent the cognitive state of agents. Consider, for example, the problem of representing the mental state of agents as described by belief reports in a natural language. Suppose the state of John's beliefs is at least partially given by the sentence

(6.2) *John believes that given the rules of chess, White has a forced initial win.*

Since the statement, *given the rules of chess, White has a forced initial win* is either a tautology or inconsistent, this would be equivalent in the possible-world model to one of the following belief reports:

(6.3) a. *John believes* t.
b. *John believes everything.*

Clearly this is wrong; if it turns out that John's belief in White's forced initial win is correct, John has a good deal of information about chess, and we would not want to equate it to the tautology t. On the other hand, if John's belief is false and no such strategy for White exists, it is not necessarily the case that all of his beliefs about other aspects of the world are incoherent. Yet there are no possible worlds compatible with a false belief, and so *every* proposition about the world must be a belief.

The representational problems of the possible-worlds approach stem from its treatment of belief as a relation between an agent and a proposition (*i.e.*, a set of possible worlds). All logically equivalent ways of stating the same proposition, no matter how complicated, count as a report of the same belief. By contrast, the deduction model treats belief as a relation between an agent and the *statement* of a proposition, so that two functionally different beliefs can have the same propositional content.

There is a large philosophical literature on the problems of representing propositional attitudes using possible worlds. Perry (in [34]) gives an account of some of the more subtle

problems inherent in equating belief states with propositions; his analysis does not depend on consequential closure. Barwise (in [2]) critiques consequential closure in possible-worlds models of perception. By comparison, a good account of the relative advantages of a symbol-processing approach to representing belief can be found in Moore and Hendrix (in [31]).

The Correspondence Property

It is reasonable to ask how the deduction and possible-worlds models compare in respects other than the assumption of consequential closure. That is, are the saturated deduction models $D_s(L, \rho)$ (whose rules are consequentially complete) significantly different from possible-worlds models for the purpose of representing belief?

The last phrase, "for the purpose of representing belief," is important. The two models are composed of different entities (expressions *vs.* propositions), so we can always use a language that distinguishes these entities, and has statements that are valid in one model and not the other. So the answer to this question depends on the type of language used to talk about the models. Fortunately, the language standardly used to axiomatize possible-worlds models is the same as that of B: a modal calculus containing atoms of the form $[S]p$, in which p refers to a proposition.[1] Thus it is possible to compare the possible-worlds and deduction models by comparing their axiomatizations in modal logic. We have proven the following general property about the two approaches.

Correspondence Property. For every modal logic of belief based on Kripke possible-worlds models, there exists a corresponding deduction model logic family with an equivalent saturated logic.

The correspondence property simply says that possible-worlds models are indistinguishable from saturated deduction models from the point of view of modal logics of belief. To the author's knowledge, this is the first time that the symbol-processing and possible-worlds approaches to belief have been shown to be comparable, in that the possible-worlds model

[1] Historically, the axiomatization of modal systems preceded Kripke's introduction of a unifying possible-worlds semantics.

is equivalent to the limiting case of a symbol-processing model with logically complete deduction.

Although space is too short here to give a full proof of this claim, we will give an overview of the most important of the propositional modal logics with a possible-worlds semantics, and their corresponding deductive belief logics (a full exposition and proofs of results mentioned here are in Konolige [21]).

Modal calculi for the possible-worlds model differ, depending on the particulars of their intended domains. For propositional modal calculi, these particulars center around whether knowledge or belief is being axiomatized, and what assumptions are made about self-beliefs or self-knowledge (a survey of these calculi may be found in Hughes and Cresswell [15]). The standard propositional modal calculi contain a single modal operator (which we write here as $[S]$) and are expressed as Hilbert systems. Their rules of inference are modus ponens (from p and $p \supset q$, infer q) and necessitation (from p, infer $[S]p$). Axioms are taken from the following schemata.

$M1$. p, where p is a tautology
$M2$. $[S](p \supset q) \supset ([S]p \supset [S]q)$
$M3$. $[S]p \supset p$
$M4$. $[S]p \supset [S][S]p$
$M5$. $\neg[S]p \supset [S]\neg[S]p$

$M1$ are the purely propositional axioms. $M2$, also called the *distribution axioms*, allow modus ponens to operate under the scope of the modal operator. $M3$ are axioms for knowledge: all knowledge is true. $M4$ and $M5$ are called the positive and negative introspection axioms, respectively: if an agent believes p, then he believes that he believes it ($M4$); if he doesn't believe p, then he believes that he doesn't believe it ($M5$).

Any modal calculus that uses modus ponens and necessitation, and includes all tautologies and the distribution axioms, is called a *normal modal calculus*. Normal modal calculi have the following interesting property (see Boolos [3]): if $p \supset q$ is a theorem, then so is $[S]p \supset [S]q$. Interpreting the modal operator $[S]$ as belief, this asserts that whenever

q is implied by p, an agent S who believes p will also believe q. As expected, normal modal calculi assume consequential closure when the modal operator is interpreted as belief.

The simplest normal modal calculus is K, which contains just the schemata $M1$ and $M2$. To axiomatize knowledge, $M3$ is included to form the calculus T. Assumptions about self-knowledge lead to the calculi $S4$ ($T + M4$) and $S5$ ($S4 + M5$). McCarthy (in [24] and [25]) was the first to recognize the utility of modal calculi for reasoning about knowledge in AI systems, and defined three calculi that were extensions to T, $S4$, and $S5$, allowing belief operators for multiple agents. Sato ([38]) has a detailed analysis of these calculi as Gentzen systems, and calls them $K3$, $K4$, and $K5$, respectively. He also gives decision procedures for these logics. $K4$ is the calculus used by Moore in his dissertation on the interaction of knowledge and action ([29]).

The so-called weak analogs to $S4$ and $S5$ are formed by omitting the knowledge axiom $M3$ (this terminology is introduced by Stalnaker [41]). The weak versions are appropriate for axiomatizing belief rather than knowledge, since beliefs can be false. Levesque [22] has an interesting dissertation in which he explores the question of what knowledge a data base can have about its own information. Because he makes the assumption that a data base has complete and accurate knowledge of its own contents, the propositional calculus he arrives at is weak $S5$, with the addition of a consistency schema $[S]p \supset \neg[S]\neg p$.

How does the family of logics **B** compare with these propositional modal calculi? As with the possible-worlds logics, the deductive belief logics formed from **B** will depend on the assumptions that are made about self-beliefs. In this paper we have developed the logic family **BK**, which assumes that an agent has no knowledge of his own beliefs. The saturated logic $\mathbf{BK}_s$, restricted to a single agent, is provably equivalent to K, the weakest of the possible-worlds belief calculi.

We have developed a theory of introspection within the deduction model framework that accounts for varying degrees of self-knowledge about one's own beliefs. This theory is based on the idea that an agent's belief subsystem can query a model of itself (an *introspective belief subsystem*) to answer question of self-belief. Depending on constraints placed on the introspective belief subsystem, it is possible to arrive at any one of eight

different logic families. Two of these, BS4 and BS5, have saturated logics that are equivalent in the single-agent case to the modal systems weak $S4$ and weak $S5$.

While we have been interested in the concept of belief throughout this paper, it is possible to define a deductive belief logic based on the related concept of knowledge. One property that distinguishes knowledge from belief is that if something is known it must be true, whereas beliefs can be false. The appropriate tableau axiom for knowledge is

$$K_0: \quad \frac{\Sigma, [S_i]\Gamma \Rightarrow \Delta}{\Sigma, \Gamma, [S_i]\Gamma \Rightarrow \Delta}$$

Adding K_0 to B forms the logic family K. Particularizations of K with varying degrees of self-knowledge correspond to the propositional modal systems T, $S4$ and $S5$.

We summarize these results in the following table.

	Normal Modal Calculus	Deduction Model Family
Belief	K	BK
	weak $S4$	BS4
	weak $S5$	BS5
Knowledge	T	KT
	$S4$	KS4
	$S5$	KS5

6.2. Syntactic Logics for Belief

There are a number of first-order formalizations of belief or knowledge in the symbol-processing tradition that have been proposed for AI systems. We have labeled these "syntactic" logics because their common characteristic is to have terms whose intended meaning is an expression of some object language. The object language is either a formal language (*e.g.*, another first-order language) or an internal mental language. The logic B is also a syntactic logic, although it uses a modal operator; the argument of the operator denotes a sentence in the internal language. We have chosen to use a modal language for B because

it has a relatively simple syntax compared to first-order formalizations. It is also less expressive, in that quantification over sentences of the object language is not allowed by the modal syntax.

McCarthy [26] has presented some incomplete work in which *individual concepts* are reified in a first-order logic. Exactly what these concepts are is left deliberately unclear, but in one interpretation they can be taken for the internal mental language of a symbol-processing cognitive framework. He shows how the use of such concepts can solve the standard representational problems of knowledge and belief, *e.g.*, distinguishing between *de dicto* and *de re* references in belief sentences.

A system that takes seriously the idea that agent's beliefs can be modeled as the theory of some first-order language is proposed by Konolige [19]. A first-order metalanguage is used to axiomatize the provability relation of the object language. To account for nested beliefs, the agent's object language is itself viewed as a metalanguage for another object language, and so on, thereby creating a hierarchy of metalanguage/object language pairs. Perlis [33] presents a more psychologically oriented first-order theory that contains axioms about long- and short-term memory. The ontology is that of an internal mental language.

These axiomatic approaches are marred by one or both of two defects – the lack of a coherent formal model of belief, and computational inefficiency. Regarding the first one: the vagueness of the intended model often makes it difficult to claim that the given axioms are the correct ones, since there is no formal mathematical model that is being axiomatized. In arriving at the deduction model of belief, we have tried to be very clear about what assumptions were being made in abstracting the model, how the model could fail to portray belief subsystems accurately, and so on. In contrast, the restrictions these syntactic systems place on belief subsystems are often obscure. What type of reasoning processes operate to produce consequences of beliefs? How are these processes invoked? What is the interaction of the belief subsystem with other parts of the cognitive model? These types of questions are begged when one simply writes first-order axioms and then tries to convey an intuitive idea of their intended content. (To some extent this criticism is not applicable to the formalism of Konolige in [19], because here the intended belief model is explicitly stated to be a first-order theory).

A second shortcoming is that efficient means of deduction for the syntactic axiomatizations are not provided. As we have mentioned, a system that is actually going to reason about belief by manipulating some formalization can encounter severe computational problems. Many of the assumptions incorporated into the deduction model, especially the closure property, were made with an eye towards deductive efficiency. The end result is a simple rule of inference, the attachment rule A, that has computationally attractive realizations.[1] On the other hand, formalizations that try to account for complex procedural interactions (as in Perlis's theory of long- and short-term memory), or that use a metalanguage to simulate a proof procedure at the object language level (as in Konolige [19]), have no obvious computationally efficient implementation.

[1] Several efficient proof methods are given in Konolige [21]: a decision procedure for propositional BK based on the Davis-Putnam procedure (see Chang and Lee [5]), which is sufficient to solve the Wise Man Puzzle automatically; a resolution method for the quantifying-in form of B; and a PLANNER-type deduction system.

7. Conclusion

We have explored a formalization of the symbol-processing paradigm of belief that we call the deduction model. It is interesting that the methodology employed was to examine the cognitive structure of AI planning systems. This methodology, which we might term *experimental robot psychology*, offers some distinct advantages over its human counterpart. Because the abstract design of such systems is open and available, it is possible to identify major cognitive structures, such as the belief subsystem, that influence behavior. Moreover, these structures are likely to be of the simplest sort necessary to accomplish some task, without the synergistic complexity so frequently encountered in studies of human intelligence. The design of a robot's belief subsystem is based on the minimum of assumptions necessary to ensure its ability to reason about its environment in a productive manner, namely, it incorporates a set of logical sentences about the world, and a theorem-proving process for deriving consequences. The deduction model is derived directly from these assumptions.

The deduction model falls within that finely bounded region between formally tractable but oversimplified models and more realistic but less easily axiomatized views. On the one hand, it is a generalization of the formal possible-worlds model that does not make the assumption of consequential closure, and so embodies the notion that reasoning about one's beliefs is resource-limited. On the other hand, it possesses a concise axiomatization in which an agent's belief deduction process is incorporated in a direct manner, rather than simulated indirectly. Thus, the deduction model and its associated logic B lend themselves to implementation in mechanical theorem-proving processes as a means of giving AI systems the capability of reasoning about beliefs.

Acknowledgements

Many people contributed their time and effort to reading and critiquing this paper. I am especially indebted to Stan Rosenschein, Nils Nilsson, and Jerry Hobbs in this regard.

References

[1] Appelt, D. E., "Planning Natural-Language Utterances to Satisfy Multiple Goals," *SRI Artificial Intelligence Center Technical Note 259*, SRI International, Menlo Park, California (1982).

[2] Barwise, J., "Scenes and other Situations," *Journal of Philosophy* LXXVIII, 7 (July, 1981).

[3] Boolos, G., *The Unprovability of Consistency*, Cambridge University Press, Cambridge, Massachusetts, 1979.

[4] Brachman, R., "Recent Advances in Representational Languages," Invited lecture at the National Conference on Artificial Intelligence, Stanford University, Stanford, California (1980).

[5] Chang, C. L. and Lee, R. C. T., *Symbolic Logic and Mechanical Theorem Proving*, Academic Press, New York, New York, 1973.

[6] Collins, A. M., Warnock, E., Aiello, N. and Miller, M., "Reasoning from Incomplete Knowledge," in *Representation and Understanding*, Bobrow, D. G., and Collins, A. (eds.), Academic Press, New York (1975).

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

[8] Field, H. H., "Mental Representation," *Erkenntnis* **13** (1978), pp. 9–61.

[9] Fikes, R. E. and Nilsson, N. J., "STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving," *Artificial Intelligence* **2**, 3-4 (1971).

[10] Fodor, J. A., *The Language of Thought*, Thomas Y. Cromwell Company, New York, New York, 1975.

[11] Hayes, P. J., "In Defence of Logic," *Proceedings of the 5th International Joint Conference on Artificial Intelligence*, Massachusetts Institute of Technology, Cambridge, Massachusetts (1977).

[12] Hewitt, C., *Description and Theoretical Analysis (Using Schemata) of PLANNER: A Language for Proving Theorems and Manipulating Models in a Robot*, Doctoral dissertation, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1972.

[13] Hintikka, J., "Form and Content in Quantification Theory," *Acta Philosophica Fennica* 8 (1955), pp. 7–55.

[14] Hintikka, J., *Knowledge and Belief*, Cornell University Press, Ithaca, New York, 1962.

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

[16] Israel, D. J., "The Role of Logic in Knowledge Representation," *Computer* **16**, 10 (October, 1983).

[17] Johnson-Laird, P. N., "Mental Models in Cognitive Science," *Cognitive Science* 4 (1980), pp. 71-115.

[18] Kleene, S. C., *Mathematical Logic*, John Wiley and Sons, New York, 1967.

[19] Konolige, K., "A First Order Formalization of Knowledge and Action for a Multiagent Planning System," in *Machine Intelligence 10*, J. E. Hayes, D. Michie, and Y-H Pao (eds.), Ellis Horwood Limited, Chichester, England (1982).

[20] Konolige, K., "Circumscriptive Ignorance," *Proceedings of the Second National Conference on Artificial Intelligence*, Carnegie-Mellon University, Pittsburgh, Pennsylvania (1982).

[21] Konolige, K., *A Deduction Model of Belief and its Logics*, Doctoral thesis in preparation, Stanford University Computer Science Department, Stanford, California, 1984.

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

[23] Lycan, W. G., "Toward a Homuncular Theory of Believing," *Cognition and Brain Theory* **4**, 2 (1981), pp. 139–59.

[24] McCarthy, J., Sato, M., Hayashi, T., and Igarashi, S., "On the Model Theory of Knowledge," *Stanford Artificial Intelligence Laboratory Memo AIM-312*, Stanford University, Stanford (1978).

[25] McCarthy, J., "Formalization of two puzzles involving knowledge," unpublished note, Stanford University, Stanford, California (1978).

[26] McCarthy, J., "First Order Theories of Individual Concepts and Propositions," in *Machine Intelligence* **9**, B. Meltzer and D. Michie (eds.), Edinburgh University Press, Edinburgh, England (1979), pp. 120–147.

[27] McCarthy, J., "Circumscription–A Form of Non-Monotonic Reasoning," *Artificial Intelligence* **13**, 1,2 (1980).

[28] McDermott, D. and Doyle, J., "Non-Monotonic Logic I," *Artificial Intelligence* **13**, 1,2 (1980).

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

[30] Moore, R. C, "Semantical Considerations on Nonmonotonic Logic," *SRI Artificial Intelligence Center Technical Note 284*, SRI International, Menlo Park, California (June, 1983).

[31] Moore, R. C. and Hendrix, G. G., "Computational Models of Belief and the Semantics of Belief Sentences," *SRI Artificial Intelligence Center Technical Note 187*, SRI International, Menlo Park, California.

[32] Nilsson, N., *Principles of Artificial Intelligence*, Tioga Publishing Co., Palo Alto, California, 1980.

[33] Perlis, D., "Language, Computation, and Reality," *Department of Computer Science TR95*, University of Rochester, Rochester, New York (May, 1981).

[34] Perry, J., "The Problem of the Essential Indexical," *NOÛS* **13** (1979).

[35] Reiter, R., "A Logic for Default Reasoning," *Artificial Intelligence* **13**, 1–2 (1980).

[36] Robinson, J. A., *Logic: Form and Function*, Elsevier North Holland, New York, New York, 1979.

[37] Sacerdoti, E. D., *A Structure for Plans and Behavior*, Elsevier, New York, 1977.

[38] Sato, M., *A Study of Kripke-type Models for Some Modal Logics by Gentzen's Sequential Method*, Research Institute for Mathematical Sciences, Kyoto University, Kyoto, Japan, July 1976.

[39] Schubert, L. K., "Extending the Expressive Power of Semantic Nets," *Artificial Intelligence* **7**, 2 (1976), pp. 163–198.

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

[41] Stalnaker, R., "A Note on Nonmonotonic Modal Logic," unpublished manuscript, Department of Philosophy, Cornell University (1980).

[42] Warren, D. H. D., "WARPLAN: A System for Generating Plans," *Dept. of Computational Logic Memo 76*, University of Edinburgh School of Artificial Intelligence, Edinburgh, England (1974).

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

[44] Woods, W., "What's in a Link?," in *Representation and Understanding*, Bobrow, D. G., and Collins, A. (eds.), Academic Press, New York (1975).