

217
45

Deceptive Robots

by

Brian Lee Klock

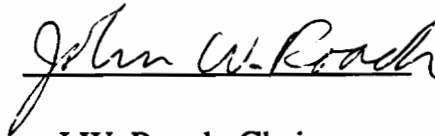
Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

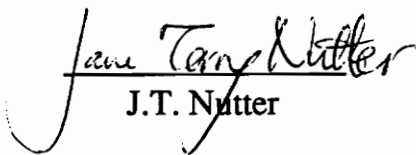
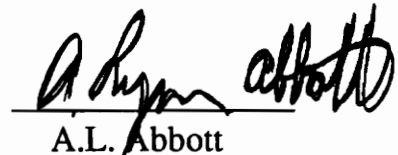
in

Electrical Engineering

APPROVED:



J.W. Roach, Chairman


J.T. Nutter
A.L. Abbott

August 9, 1990

Blacksburg, Virginia

LD

5655

V855

1990

K 576

C.2

Deceptive Robots

by

Brian L. Klock

Committee Chairman: John W. Roach

Electrical Engineering

(ABSTRACT)

Plan recognition in cooperative or adversarial situations requires the ability to reason with beliefs. The problem is complicated in adversarial situations because an opponent may employ deception. In this case, an agent must also be able to reason about an opponent's beliefs (nested beliefs) as well as his own beliefs. A system has been developed that permits agents to reason with nested beliefs using possible worlds semantics; consistency maintenance is employed to allow agents to revise their beliefs when an inconsistency occurs. The advantages over prior systems are that belief revision occurs without user interaction and that beliefs are treated as objects having equal status with facts; this permits complex interaction between beliefs and actions. The theory includes treatment of many areas necessary to create a system of multiple autonomous reasoning agents. These agents are given the ability to deceive each other and to predict when they are being deceived. The system is shown to be practical by its implementation in a simulation.

ACKNOWLEDGEMENTS

I would like to thank all of my fellow students who provided information and assistance throughout my research. I would also like to thank Lynn Abbott for serving on my committee and Terry Nutter for serving on my committee and for many helpful criticisms. I would especially like to thank my advisor, John Roach, for his dedicated guidance.

This research was supported in part by a contract from the Office of Naval Research, contract number N62269-89-9081-8295.

TABLE OF CONTENTS

	<u>Page</u>
List of Figures	x
1. Introduction	1
1.1. Theoretical Basis	4
1.2. Example	5
1.3. Topics Covered in the Thesis	7
1.4. Summary Outline	10
2. Survey of Previous Work	12
2.1. Konolige's Deduction Model	12
2.2. Moore's Theory of Action and Belief	16
2.3. Marins and Shapiro's SNeBR	18
2.4. Conclusions	21
3. Counterfactual Reasoning	23
3.1. Uses of Counterfactual Reasoning	24
3.1.1. Our Use of Counterfactual Reasoning	26
3.2. The Virginia Tech Counterfactual Engine	28
3.2.1. Operation of the Counterfactual Engine	30
3.2.1.1. Operation of the Rules	33
3.2.1.2. Possible Worlds and Timestamps	38
3.2.2. Causality	40
3.2.3. Formalization	42
4. Representation of Beliefs	45
4.1. A First Attempt	45
4.2. Requirements for a Representation	47
4.3. Development of a Representation	50

4.3.1. Modal Logic and the Accessibility Relation	54
4.3.2. Extension of Accessibility to Include Multiple Agents	59
4.4. Implementation of the Representation Scheme	62
4.5. States of Belief	72
4.5.1. Distinction Between Truth Condition and Epistemic Status	73
4.5.2. Uncertain Beliefs	74
4.5.3. Beliefs About the Unknown	75
4.5.4. Beliefs that Something is False	76
4.6. Conclusions	80
5. Uncertainty and Unknownness	83
5.1. Negative Beliefs	83
5.1.1. Representing Disjunctions	87
5.2. Unknownness	89
5.3. Representing Uncertainty	94
5.3.1. Meaning of Opinions	95
5.3.2. Explanations	99
5.3.3. Uses of the Representation	105
5.3.3.1. Combining Uncertainty With Other Beliefs .	105
5.3.4. Implementation	110
5.3.4.1. Combination of Opinions	111
5.4. Conclusions	117
6. Theoretical Basis of Our System	118
6.1. Ontology	118
6.1.1. Real World	119

6.1.2. Concepts	120
6.1.3. Propositions	125
6.1.3.1. Abstract Situations	125
6.1.3.2. Beliefs	127
6.1.3.3. Goals	128
6.1.3.4. Communication	128
6.1.4. Statements	129
6.1.5. (Partial) Possible Worlds	131
6.2. Use of Possible Worlds	135
6.2.1. Relations Between Statements and Propositions ...	135
6.2.2. Comparison to Situation and Possible Worlds	
Semantics	138
6.2.3. Theory of Truth	140
6.3. Reasoning with Concepts	142
6.4. Imaginary and Fictional Objects	142
6.5. What Exists	147
6.6. The King of France	155
6.7. Conclusions	158
7. Formal Basis	159
7.1. Time Characterization	160
7.2. Relations Among Nested Beliefs	166
7.3. Memory Relation	171
7.4. Conclusions	175
8. (Mental) Event Logic and the Frame Problem	176
8.1. Model of Time	176
8.1.1. Instants vs. Intervals	178

8.1.2. Events Related to Instants	182
8.1.3. Formalization	184
8.1.4. Causality	186
8.2. The Frame Problem	189
8.2.1. Theoretical Solution	190
8.2.2. Internal Representation	193
8.2.3. Components of the Frame Problem	195
8.2.3.1. The Qualification Problem	195
8.2.3.2. The Ramification Problem	201
8.2.3.3. The Persistence Problem	203
9. Substitution of Equality	206
9.1. Representations	207
9.2. Our Approach to Substitution	209
9.2.1. Our Definition of Equality	212
9.2.1.1. Our Solution	213
9.2.1.2. Comparing Situations	216
9.2.1.3. Examples of Our Solution	218
9.2.2. Comparison to Other Solutions	220
9.3. Uses of Substitutions	223
9.3.1. Substitutions in Derived Beliefs	223
9.3.2. Substitutions in Communication	225
9.4. Variables	226
9.5. Additional Problems	229
9.6. Conclusions	231
10. Perception, Communication, and Action	233
10.1. Perception	233

10.1.1. Rules of Perception	235
10.2. Communication Model	245
10.2.1. Structures	245
10.2.1.1. Structure for P	249
10.2.1.2. Structure for S	251
10.2.2. Interactions Between Structures	252
10.2.3. Partiality of Structures	256
10.3. Action and Belief	257
10.4. Conclusions	264
11. Deception	265
11.1. Detecting Deception	266
11.1.1. Detecting Conflicts	267
11.1.2. Determining the Relevant Information	270
11.1.3. Distinguishing Deception from Mistakes	273
11.1.4. Detecting Lies	273
11.1.4.1. Types of Lies	276
11.1.5. Deceptive Actions	279
11.1.6. Theory of Deception	283
11.2. Implementation	286
11.2.1. A Bilevel Reasoning System	286
11.2.2 Implementation of Deception Reasoning	289
11.3. Conclusions	294
12. A Multiple-Agent Simulation	296
12.1. Structures	296
12.1.1. Statements in the Simulation	297
12.1.1.1. Beliefs and Facts	297

12.1.1.2. Actions	299
12.1.2. Schedules	300
12.1.2.1. Timing	303
12.1.3. Rules	305
12.1.4. Counterfactual Engine Control Rules	308
12.2. Capabilities	309
12.2.1. Communication	309
12.2.2. Perception	311
12.2.3. Substitution	313
12.2.4. Nested Beliefs and Deception	314
12.3. Conclusions	315
13. Results	317
14. Conclusions	341
14.1. Contributions of This Thesis	341
14.2. Future Work	343
15. Literature Cited	345
Vita	349

List of Figures

Figure	Page
1. A possible scenario of beliefs at different levels of nesting	57
2. Accessibility relations depicted graphically	63
3. Relation between belief sets from distinct levels of nesting	65
4. Division of a world into distinct sets of beliefs ..	70
5. Types of belief status	79
6. Accessibility between belief sets expressed in "set" notation	82
7. (a) a probability scale for certainty	98
(b) separate scales for true and false beliefs	98
8. Representation of nested negations as a single active belief	102
9. Possible output table showing justification for a result	109
10. Mapping from world to beliefs via statements	132
11. A cat on a mat	134
12. Accessibility between possible worlds representing nested beliefs	167
13. Memory relation	172
14. Temporal relations between events, statements, and possible worlds	181
15. (a) Relation between instants, possible worlds, and truth values	194
(b) Internal representation	194

1. Introduction

Whenever multiple agents coexist in a world, they may find it necessary or advantageous to reason about the plans and goals of other agents. This consideration adds to the complexity of developing a system for reasoning about beliefs. Any intelligent agent must have the ability to reason about his beliefs; he must be able to predict the possible consequences of an action and observe changes in the actual world. Furthermore, he often must reason about objects or actions separated from him in space and time.

The problem of goal recognition is further complicated because it requires an agent to reason about the beliefs of other agents. Both cooperative and adversarial situations involving multiple agents require the ability to reason about other agents' beliefs. In a situation where multiple agents are working on a cooperative task, they must be able to reason about each other's beliefs to determine what action each must perform to complete the task.

For example, suppose that two agents wish to lift a heavy log. If each agent is standing at one end of the log, then each should be able to predict that the other agent will lift the end of the log nearest him. The task cannot be completed efficiently if the agents switch ends or both attempt to lift the same end.

In an adversarial situation, the ability to reason about another agent's beliefs will assist an agent in determining when his opponent is deceiving him; deception is an important factor that must be considered as part of goal recognition. Deception is very dependent on

the beliefs of others, just as competition is. For an agent to believe that competition exists with another agent, he must believe that the other agent has a goal that conflicts with his own; he must also believe that the other agent believes there is a conflict of goals. We assume in our treatment of deception that agents will not lie from sheer orneriness.

Similarly, an agent who wishes to deceive an opponent must affect the opponent's beliefs, and he may achieve this by communicating false beliefs to the opponent as if they were true beliefs. If an agent has some belief that he wants an opponent to be deceived about, then the agent wants the opponent to believe that the agent believes something other than his actual belief. Essentially, the agent is trying to communicate to his opponent a concept that exists in the agent's mind, but that the agent does not believe maps to the physical world. If the opponent believes that the agent holds this belief, then the opponent may also hold the same (incorrect) belief.

For example, agent A may believe that a map is in room1. If he does not want agent B to have the map, then he may tell agent B that the map is in room2. He actually believes "INROOM room1 map" but he wants agent B to believe that he believes "INROOM room2 map." His purpose is make agent B also believe "INROOM room2 map."

On the other hand, we want agents to make guesses about when they are being deceived. This also requires that agents be able to reason about what other agents believe. If there is a reason to be suspicious of another agent (e.g. there is a conflict between the agents' goals) and if there are discrepancies between what an agent perceives and what

an agent believes, then the agent may guess that he is being deceived.

For example, agent A may tell agent B that the map is in room2, but if agent B is aware of the conflict over possession of the map, then agent B may guess that agent A is lying to him. After all, it would be strange for agent A to tell agent B the location of the map if he does not want agent B to have it. Agent B should use his belief that agent A is lying to him when he plans his search for the map, i.e., he should look in other places before looking in room2.

For an agent to reason about other agents' beliefs he must be able to represent several distinct models of the world. Such models will enable an agent to reason with different views of the world. These models should include an agent's own beliefs, his model of another agent's beliefs, and his model of another agent's model of the first agent's beliefs (we hypothesize that this depth of nested beliefs is sufficient for reasoning about deception). With this capability, an agent can (hopefully) predict when an opponent is using deception by reasoning about that agent's (presumed) beliefs as well as his actions.

In addition, an agent could make plans to achieve his opponent's (presumed) goals using his model of the opponent's beliefs about the world. This would result in beliefs about how an opponent might act to achieve his goals. When the opponent's actions differ from those expected, it could be taken as a clue that deception is involved.

Previously, no single system has integrated the necessary capabilities to reason with nested beliefs and revise beliefs automatically when contradictions arise. This thesis presents a system that can determine a plausible way of resolving inconsistencies that

occur when reasoning with beliefs and nested beliefs. This system has been implemented in a simulation that illustrates how to detect deception in an adversarial situation.

1.1. Theoretical Basis

Our system is based on consistency maintenance because we use a coherence theory of belief revision. This theory holds that humans reflect on the justification for their beliefs only when inconsistencies arise; they recognize conflicts among their beliefs and consider the justifications for the relevant beliefs to determine how the inconsistency should be resolved. The resolution of conflicts "should involve minimal changes in one's beliefs in a way that sufficiently increases the overall coherence" [Harman, 1986]. There have been psychological experiments that support a coherence theory for belief revision [Ross and Anderson, 1982].

Work done with epistemic counterfactuals at Virginia Tech has resulted in a counterfactual reasoning engine that provides a mechanism for our system. This mechanism allows us to maintain a knowledge base of beliefs along with rules for generating new beliefs by resolving inconsistencies with minimal changes. While the determination of a minimal change is a complex problem, the Virginia Tech counterfactual engine incorporates Rescher's theory of modal categories [Rescher, 1963] as a basis for this determination. A minimal change is one that involves rejecting beliefs that have the least justification or the

greatest uncertainty.

Our system is based on a possible worlds* semantics. This approach works well when reasoning with beliefs because an agent's beliefs form a partial possible world that he believes models a portion of the actual world. As the agent reasons or acquires new information, there will be several ways that his beliefs might change; the counterfactual engine generates new partial possible worlds that include the new beliefs and as many of the old beliefs as possible while maintaining consistency. The agent must adopt one of these partial possible worlds as his new model of the world.

Although a possible worlds semantics is not necessary for a system that reasons about beliefs, it is a widely accepted approach. We have chosen this approach because the counterfactual engine gives us the ability to operate directly on the possible worlds. This permits our system to use the semantics when determining the most plausible way to resolve an inconsistency; the relevance and certainty of beliefs can be determined more easily than in a system that must operate on syntactic entities that cannot be easily differentiated.

1.2. Example

A simple example will illustrate the problems associated with reasoning about deception and nested beliefs. In the following situation, a Hero is attempting to locate key2 so that he can unlock a door and rescue the Damsel. The Villain knows that key2 is in room4,

* See Chapter Six for our definition of possible world.

but he does not want the Hero to find the key and rescue the Damsel. Hence, the Villain knows that his goal and the Hero's goal create a conflict; the Hero is unaware of the conflict because he does not realize that the Villain has evil intentions. The Hero sees the Villain and asks him where key2 is located. The Hero's action of querying the Villain is introduced as a new counterfactual that leads to the following possible world (note that some unimportant details have been omitted from this possible world):

cf: (ACT (ASK Villain (INROOM where key2)) Hero 6)

Possible World : w6

6 100 (ACT (ASK Villain (INROOM where key2)) Hero 6)

6 100 (GOAL Hero (POSSESS key2) True set4)

6 100 (INPUT QUERY (INROOM where key2) Hero Villain 6)

6 100 (ACT (SPEAK_TO Hero (INROOM DUNGEON key2)) Villain 6)

6 100 (INPUT REPORT (INROOM DUNGEON key2) Villain Hero 6)

6 100 (INPUT REPORT (INROOM room5 key2) Villain Hero 6)

6 100 (NEW_INFO 6 INROOM room5 key2 set1)

6 100 (REVISE SCHEDULE Hero ((MOVE_TO room5)))

6 100 (INROOM room5 key2 True set1)

6 100 (TIME 6)

This simple example illustrates many capabilities that an agent must have to reason about this situation. The Hero must be able to reason about actions to conclude that he should attempt to ask the Villain for the key's location, and he must actually be able to communicate (note that the Villain's action of speaking occurs in the same time frame as the question because it is a response to the

stimulus of the question). The second statement in the world indicates that the Hero believes that the Villain believes that the Hero wants to possess the key; the Hero believes the Villain will assume this since the Hero is asking for the key's location. This involves an ability to represent and reason with nested beliefs.

The Villain must be able to reason about the request and determine that he should lie about the location of the key since he does not want the Hero to have it. The Hero cannot understand the Villain's response because he does not identify the concept of the Dungeon with any object; the Hero does know what room the Villain is denoting and can therefore substitute the concept of room5 that he is familiar with. The Hero must then be able to revise his beliefs because he now thinks he knows the location of key2.

When the Hero actually goes to room5, he must be able to perceive that the key is not where it is expected. At a later time, he must use this information in determining that the Villain is lying to him. Thus, this example illustrates some of the problems involved in simulating a complex interaction between agents. Our system integrates solutions to each of these problems.

1.3. Topics Covered in the Thesis

We have noted several of the problems encountered in developing a system for reasoning about deception. We will now examine these in greater detail.

A system that deals with changing beliefs encounters many

problems. There must be a well defined process for determining how beliefs should be revised, and there must be a representation that permits changes in beliefs to be computed and stored. In effect, we are seeking a solution to the frame problem applicable to beliefs; by the frame problem we mean the problem of computing all changes in the world that are relevant to an event. The results of these changes must be represented, and there must be a way to determine the current status of what has not changed. Since beliefs exist in the world (inside an agent's mind), changes in beliefs are changes in the world. We will discuss our mental event logic and the frame problem in Chapter Eight.

The determination of how beliefs should be revised uses uncertainty reasoning; these uncertainties represent an agent's degree of belief in a statement with a specific interpretation. Also, an agent often needs to represent that he is aware he does not know something; this is especially true of situations when action and belief must interact. For example, if an agent wants to open a safe, he must be able to recognize that he does not know the combination; this realization should motivate him to obtain the combination.

Agents acting in the world must be able to perceive changes in the world, and if there are multiple interacting agents, they should be able to communicate with each other to facilitate cooperation. A theory of substitution of equalities is necessary for communication when agents refer to the same objects with different descriptions (e.g., one agent may call a room the den while another agent refers to it as the study). Substitution of equalities may also be important when these different concepts are used in deception. Deception

requires a representation that permits nested beliefs because it depends on the ability to reason about an agent's goals and detect conflicts between goals.

Another capability needed is temporal reasoning to distinguish past beliefs from those currently held or from those that were never believed, temporal reasoning is needed in the system. Thus, the system must be able to keep track of which beliefs are held at a given time. This is required to give the system the ability to reason about the following kinds of issues: 1) is a fact currently believed, 2) did the agent begin believing one fact before he believed another, and 3) does the agent believe that certain facts were ever true together at the same time. Reasoning about the temporal ordering of events is also needed for an agent to determine how his beliefs should be revised.

As an example of this last case, consider that agent B would believe that a window was broken if he looked at the window sometime after it was broken. Agent A may need to determine if B is aware of the broken window. Thus, A must be able to determine if B looked at the window at a time after it was broken; if so, then B probably noticed that it was broken.

In addition, the temporal ordering of certain actions may rule out some possible explanations of another agent's actions. For example, if a person buys an expensive new car *after* attending a high school reunion, it is unlikely that he purchased the new car merely to impress his former classmates. This kind of reasoning may be needed for recognizing an agent's goals based on his actions.

These are the technical problems that we have encountered while

developing a system with possible worlds semantics that uses consistency maintenance to revise beliefs when an inconsistency arises in a world. This thesis describes and formalizes our belief reasoning system, presents our solutions to the technical problems, and presents the results of the simulation developed to implements our theory.

1.4. Summary Outline

Chapter Two contains a review of the important literature related to our work. Chapter Three contains a description of the counterfactual engine's operation and a formalization of its operating rules. Chapter Four shows the development of the representation we use for beliefs and nested beliefs, and Chapter Five discusses uncertainty in the system and how we represent various states of belief.

Chapter Six explains the interpretation of statements in our system; it also describes the ontology that we use as a basis for our system, and how the system reasons with objects that do not exist in the physical world. Chapter Seven contains the formalization of the tense logic that describes our model of time, and it also contains a formalization of the accessibility relations between possible worlds in terms of a modal logic system (S5).

Chapter Eight contains the (mental) event logic and a discussion of the frame problem. Chapter Nine addresses the problem of substitution of equalities. Chapter Ten discusses a model for perception, communication between agents, and the relation between actions and beliefs. Chapter Eleven discusses deception. Chapter Twelve

discusses the simulation. Chapter Thirteen discusses our results and Chapter Fourteen discusses conclusions and areas for future work.

2. Survey of Previous Work.

The system described in this thesis uses theories from many different areas of research. Rather than exhaustively review each of these areas in this chapter, we will examine several earlier systems that have been used for reasoning with beliefs. Previous work in individual areas of research will be discussed as each topic is introduced in the thesis.

The systems we examine have offered solutions to several important problems in the area of belief reasoning. The problems include the derivation of new beliefs from an agent's current beliefs, the integration of action and knowledge (which are codependent), the propagation of supports for derived beliefs, and the revision of beliefs when contradictions occur. Each of these problems must be solved for an agent who must act in a changing world and reason about beliefs and deception.

2.1. Konolige's Deduction Model

Konolige's 'deduction model' uses logical deduction to model belief subsystems and derive new beliefs [Konolige, 1985]. The system is designed to deduce the logical consequences of an agent's beliefs, but Konolige's goal is to develop a system that need not assume consequential closure, i.e., an agent is not necessarily aware of all logical consequences of his beliefs. At the same time Konolige wishes to permit relevance incompleteness; this essentially gives an agent an

ability to circumscribe the beliefs that he thinks are relevant to a problem.

Beliefs are represented in the framework of a formal model; Konolige develops a logic to deduce the consequences of beliefs. The logical system, which he calls B, is designed to have reasonable computational properties. B is a syntactic system that uses modal operators. In the limit of logically complete deduction (consequential closure), B is a modal logic with possible world semantics.

Konolige creates *deduction structures* that consist of a set B and a set R. The set B is a set of sentences that correspond to the base set of beliefs of a belief subsystem. The set R is a set of deduction rules that correspond to the inference rules of the belief subsystem. A belief set is the base set of beliefs along with all beliefs that can be deduced using the accompanying rules. The belief set from one deduction structure becomes the base set of beliefs for another.

Konolige demands that his deduction structures satisfy the following four conditions [Konolige, 1985]:

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.

Konolige can represent nested beliefs and introspection through the recursion property. He can also allow agents to make deductions with another agent's beliefs and deduction rules by using his attachment rules. His attachment rule for the language of the logic B is defined as follows:

A branch with the signed formulas $T[S_i]A$ and $F[S_i]B$ closes if A proves B using $p(i)$.

$P(i)$ is the set of deduction rules for the belief subsystem containing the statements S_i . This means that if an agent can prove B from A using a belief subsystem, then he cannot believe that A is true and B is False; branches of possible deductions cannot contain contradictions.

An example of a proof using this attachment rule for an agent R is:

(1) F [R](A $\supset$ B) $\supset$ ([R]A $\supset$ [R]B)	
(2) T [R](A $\supset$ B)	(1)
(3) F [R]A $\supset$ [R]B	(1)
(4) T [R]A	(3)
(5) F [R]B	(3)

x

A, A $\supset$ B prove B for R	(2), (4), (5)

Thus, Konolige's system allows agents to reason about their own beliefs and other agents' beliefs, and it allows them to make only those deductions supported by their inference rules; agents are not

aware of all the logical consequences of their beliefs. These are desirable properties in a belief reasoning system, and Konolige presents a valid solution that supports reasonably complex deductions using an auxiliary tableau deduction system.

Konolige's logic allows him to solve some interesting problems. One problem he refers to as the chess problem. Since chess is a finite game and all of the moves are known, it should be possible to determine if the player who moves first has a forced initial win. The difficulty arises because there are too many possible combinations of moves to analyze; there is a solution to the problem, but agents do not have the resources (memory, computational power) to find the solution. Konolige can reason about problems where an agent is resource limited.

Another problem Konolige refers to as the not-so-wise-man problem. This problem is similar to the well known wise man problem where three agents try to determine the color of a spot on their foreheads. In Konolige's problem, the agents reason about each other's beliefs, but they also reason about what beliefs the other agent's do not have. Thus, the deduction model can use nested beliefs and incomplete information.

Unfortunately, the deduction model prohibits inferences that lead to contradictions; it has no way of resolving contradictions if they were allowed to occur. This limits valid deductions to those that are monotonic; an agent cannot use additional information to revise his belief about the truth value of a proposition. Since beliefs often change over time, an agent should be able to recognize and resolve contradictions.

2.2. Moore's Theory of Action and Belief

Moore proposes a system that integrates knowledge and action [Moore, 1985]. He argues that this is necessary because actions can affect an agent's knowledge and his knowledge may limit his possible actions. In fact, Moore points out that nearly all actions that are not automatic responses require some prior knowledge. For example, walking somewhere requires that an agent knows the location of where he is going.

Moore's approach is to develop separate logics for knowledge and action and then merge them; his system is based on possible worlds semantics. He first determined the properties of knowledge that he wanted his system to have. These are defined by the following axioms:

M1. P , such that P is an axiom of ordinary propositional logic.

M2. $\text{KNOW}(A, P) \supset P$

M3. $\text{KNOW}(A, P) \supset \text{KNOW}(A, \text{KNOW}(A, P))$

M4. $\text{KNOW}(A, (P \supset Q)) \supset (\text{KNOW}(A, P) \supset \text{KNOW}(A, Q))$

closed under the principle that

M5. If P is an axiom, then $\text{KNOW}(A, P)$ is an axiom.

Moore determines that these properties correspond to an S4 modal logic system which he then interprets using possible world semantics. Next, he formalizes the relations among the possible worlds by translating them into first order logic statements, using the relation K to capture the accessibility relations between worlds.

Moore also develops axioms for an action logic. He does this by representing the situation calculus in modal logic. There are four

operators in his system: DO, CAN, RES, and RES1; two result operators (RES and RES1) are needed because the results of actions may or may not yield the results an agent expects. These operators are translated into extensional predicates (D,R,etc.) that encode the relations between possible worlds and are used to define axioms for an action logic. There are three conditions that Moore uses to define the effect of actions on an agent's knowledge; these conditions are:

- (1) The agent knows that Q will be true after he performs the action just in case P is true before he performs the action.
- (2) After the agent performs the action, he knows that he has just performed the action.
- (3) After the agent performs the action, he knows whether Q is true.

Moore uses the accessibility relations of possible worlds to represent both knowledge axioms and action axioms; in the first case the accessibility relation captures the information that an agent has knowledge of, and in the second case it relates actions to their results. By overlaying the two possible worlds analyses, Moore can represent an agent's knowledge about the results of actions. The possible worlds structure is translated into first order logic using the extension predicates; this allows Moore to generate proofs.

Unfortunately, Moore's use of first order logic requires a theorem prover that must be able to generate long proofs and such a theorem prover cannot (currently) be implemented. Many steps of these proofs involve translations between Moore's operators and first order predicates; rigid designators must also be manipulated. The required

proofs are extremely complicated and difficult to follow. For example, a proof that an agent can read a combination from a slip of paper and dial it to open a safe requires one hundred thirty-one steps.

In spite of its complexity, Moore's system does integrate action and knowledge. The method of integrating them, however, does not treat thinking as an event. Thinking should be an event that can have results just as physical actions do; Moore's use of knowledge only permits an agent to capture the relation between actions and their results. Thinking cannot result in new beliefs.

2.3. Martins and Shapiro's SNeBR

Martins and Shapiro have developed a system specifically designed to handle contradictions that require belief revisions. Their system automatically computes dependencies of propositions on other propositions, discovers contradictions among them, records the contradictions, and switches between belief spaces.

Martins and Shapiro use a specially developed logic for formalizing their system [Martins and Shapiro, 1988]. This SWM system (for Shapiro, Wand, and Martins) can capture relevance between propositions and keeps track of propositional dependencies. Each belief is associated with a list of information; this information indicates the origin of the belief (hypothesis, derived, etc.), the set of supporting beliefs, and the sets of beliefs with which the belief is inconsistent.

The inference rules of the SWM system can deal with

contradictions, but modifications are made to permit practical implementation. These modifications involve the addition of non-standard connectives that allow deductions within the representation structure rather than merely about the representation.

Martins and Shapiro implement their system using semantic networks; they build upon their previously developed system, SNePS (Semantic Network Processing System). The implementation of their belief revision system is referred to as SNeBR (SNePS with Belief Revision).

This system keeps track of propositional dependencies and informs the user when a contradiction occurs; it then identifies the hypotheses that led to the contradiction so that the user may reject or revise one of them to eliminate the contradiction. The process can be repeated until the contradiction is eliminated.

Martins and Shapiro give an example using Summers' problem of 'Who will marry Freeman.' The statement of the puzzle is as follows:

Freeman knows five women: Ada, Bea, Cyd, Deb and Eve.

(1) The women are in two age brackets: three women are under 30 and two women are over 30.

(2) Two women are teachers and the other three women are secretaries.

(3) Ada and Cyd are in the same age bracket.

(4) Deb and Eve are in different age brackets.

(5) Bea and Eve have the same occupation.

(6) Cyd and Deb have different occupations.

(7) Of the five women, Freeman will marry the teacher over 30.

The system represents facts as wff's. For example:

wff1: Woman(Ada)

wff2: Woman(Bea)

wff3: Woman(Cyd)

wff4: Woman(Deb)

wff5: Woman(Eve)

It uses wff's (well-formed formulas) to represent the other rules given and uses these to deduce a result. The notation $_{2xx_1}^1$ means that exactly one of the two arguments of a function is true. The 0 operator is called thresh. The thresh operator takes a set of n propositions as arguments and asserts that there is a connection between them; the notation $_n0_i$ indicates that either fewer than i of the n arguments are true, or they are all true. In other words, if i or more arguments are true, then they are all true. The system will output information that looks like the following:

I wonder if marry(Freeman, ?who) holds withing the BS (belief space) defined by the context (wff31 wff28 wff15 wff13 wff89 wff6).

Let me try to use the rule $A(?x)_{20_1}(\text{marry}(\text{Freeman}, ?x),_{2xx_2}^2(\text{age}(?x, o30), \text{worker}(?x, \text{teacher})))$. (This rule captures the idea that the person who marries Freeman must be both a teacher and over thirty, and there is only one such person).

I wonder if age(?x,o30) holds within the BS defined by the context (wff31, wff28, wff15, wff13, wff89, wff6).

I know age(Cyd, o30).

I know age(Ada, o30).

I wonder if worker(?x,teacher) holds within the BS defined by the

context (wff31, wff28, wff15, wff13, wff89, wff6).

I know worker(Deb, teacher).

I know worker(Ada, teacher).

Since worker(Ada,teacher) and age(Ada, o30) I infer marry(Freeman, Ada).

This illustrates that the system can make deductions to arrive at an answer. During operation, the user can interact with the system; if a contradiction is found, it identifies the hypotheses that led to it and prompts the user for a decision. Thus, the system is able to keep track of the supports for propositions as it reasons with them.

2.4. Conclusions

The systems that we have examined have each solved important problems in belief reasoning. Konolige's deduction model makes deductions based on an agent's beliefs but makes only the deductions that the agent's inference rules allow. It also allows introspection and reasoning about nested beliefs. Moore's system integrates knowledge and action, which have significant effects on each other. Martins and Shapiro's system is able to propagate supports for propositions as it reasons, and it can identify contradictions and the hypotheses that lead to them.

All of these abilities are needed to function in the world as a human would. We have based much of our system on the concepts developed in these earlier systems. Our system allows the integration of beliefs and actions and combines this with an ability to reason with

an incomplete model of the world and to automatically revise beliefs to achieve the most plausible new model.

3. Counterfactual Reasoning

Any system that operates on a complex, dynamic knowledge base must have some type of control structure. In a system that reasons with beliefs, this control structure should enable us to generate new beliefs and revise previous beliefs; hopefully, this will be done in a consistent manner.

We have based the framework that allows our system to generate new possible worlds on the theory of counterfactuals. Much work has been done at Virginia Tech in the area of counterfactual reasoning. This work has resulted in a counterfactual reasoning engine that we will use as the mechanism for our system.

Our system uses possible worlds semantics to represent agents' beliefs. Since beliefs may change over time, we must generate new possible worlds to reflect these changes. The counterfactual engine performs hypothetical reasoning to infer new propositions. When these new propositions create inconsistencies in the database, the counterfactual engine can resolve the inconsistency.

The counterfactual engine operates as a syntactic system; it treats its inputs and outputs as literals and their negations. We have integrated the consistency maintenance capability of the counterfactual engine with our semantics by using propositions that represent beliefs as the inputs and outputs. Sets of these propositions are interpreted as partial possible worlds that model agents' beliefs about the real world. The addition of new counterfactuals allows us to extend the partial possible worlds.

We should be clear that not every new proposition that we assert is a proposition; we are actually performing hypothetical reasoning. In many cases, however, our hypotheses correspond to counterfactuals and result in an inconsistency; we use the term 'counterfactual'* to represent any hypothesis that is asserted in the counterfactual engine, whether or not it results in an inconsistency. The occurrence of an inconsistency is the semantic process that causes belief sets to be revised (rather than merely extended). This system for belief revision enables us to incorporate reasoning about nested beliefs and conflicting goals for detecting deception.

This chapter will discuss counterfactuals and the formalization of counterfactual engine that will define how inconsistencies are resolved (and thereby how belief revisions occur). Chapter Four describes how we represent beliefs in the counterfactual engine, and Chapter Six explains how we interpret the representation.

3.1. Uses of Counterfactual Reasoning

A counterfactual is the type of statement we usually consider as a "what if?" question, where the antecedent is assumed to contradict a proposition known to be true. An example would be a statement beginning "If George Washington were alive today" We know that George Washington is not still living, but we reason about what he would think or do today based on what we know about him. In other

*The term 'counterfactual' is commonly used to indicate a counterfactual conditional.

words, we are employing hypothetical reasoning; we propose some statement and then predict what would be true if that statement were true.

Hypothetical reasoning need not be limited to situations that seem impossible or highly improbable; George Washington would be more than 200 years old if he were alive today. Consider instead the statement "If I had gone to the store yesterday (instead of today)" It is very possible that a person could have made a trip to the store a day earlier, but this type of problem is not of great use for plan recognition; we are interested in what agents will do, not what they have not done. Nevertheless, it might be interesting to consider what might have happened if an agent had done something differently in the past; this use of counterfactuals for hypothetical reasoning could be used in an analysis of our mistakes. It would be interesting to apply this use of counterfactuals in a system that could learn from its mistakes.

An alternative to the previous example would be the statement "If I go to the store tomorrow (instead of today)" This allows us to reason about different possible futures that depend on our actions. This type of reasoning is very important in planning; the ability to consider the outcomes of various plans allows an agent to select the best possible plan. A planning system has already been implemented using the Virginia Tech counterfactual engine [Pendergraft, 1988].

A similar type of hypothetical reasoning may occur when an agent is unsure of the truth value of a proposition. In this case, he can create two distinct possible worlds; one in which the proposition is

true and one in which it is false. Hence, he is not hypothesizing different possible outcomes based on his own actions, but he is considering possible worlds that could match the actual world based on what he believes.

It is this type of reasoning that concerns us primarily. We are reasoning with beliefs but may sometimes be unsure of our beliefs; however, we can create alternative worlds to account for multiple possibilities. Thus, we are essentially saying "What if I am mistaken and this is actually the case?"

3.1.1. Our Use of Counterfactual Reasoning

Although inconsistencies do not always occur when we assert the new propositions that we refer to as counterfactuals, the counterfactual engine resolves inconsistencies when they occur. The idea of consistency is important because we are working with possible worlds; an inconsistency, such as a world containing p and $\sim p$, would create an impossible world [Bradley and Swartz, 1979]. An impossible world could never correspond to the real world, but neither could a possible world that contains false propositions. The distinction is important because a rational agent is aware that an impossible world can never correspond to the real world; on the other hand, an agent may not be aware that a possible world contains false propositions, or he may be aware of the inconsistency but accept it for lack of a way to resolve it.

We should not generate impossible worlds because agents are aware

that they could never correspond to the actual world. There may be inconsistencies, however, that are not logical inconsistencies but physical inconsistencies or some other type where the inconsistency arises as a result of the interpretation. We must define which of these inconsistencies an agent is aware so that he may resolve his beliefs accordingly; we do this by using some counterfactual rules to define consistency constraints that an agent is aware of.

For example, let p denote the proposition that an agent is in room1, and let q denote the proposition that the same agent is in room2. It is not a logical inconsistency to have p and q in a belief set, but a rational agent would recognize that it is a physical inconsistency because an agent cannot be in two different rooms at the same time. Thus, we must have an explicit rule specifying that p and q are inconsistent.

If there is no rule to handle a particular non-logical inconsistency, then the agent can believe the inconsistency and may or may not be aware of it. This is a valid state because we explained above that human beliefs can behave in the same manner.

Other counterfactual rules specify which deductions an agent can make based on semantic interpretations; an agent can only make deductions that are relevant to the current counterfactual. Thus, an agent does deduce all of the logical consequences of his beliefs, just as humans do not (we discussed this problem of consequential closure in Chapter Two). Agents make only those deductions that counterfactual rules permit based on semantic interpretations, and the rules that act as consistency constraints place an upper bound on what deductions can

be made. Our system prohibits inconsistencies that we have defined as such. Although consistency is undecidable in general, the counterfactual engine makes it decidable at the expense of representation power by requiring all counterfactual statements to be of a form proven decidable by Bernays and Schonfinkel [Bernays and Schonfinkel, 1928].

When a counterfactual (representing a new belief) is introduced into the system, the effort to maintain consistency generates changes in beliefs and new possible worlds. If it is inconsistent to assert the implied conditions when the antecedent conditions are present, then there is a mistake in some belief or in the rule. Thus, an agent must think "Something is inconsistent so I must change something. What if I reject this belief or that belief or this rule?" Different possible worlds are generated to account for the different ways of resolving the inconsistency.

3.2. The Virginia Tech Counterfactual Engine

There are two distinct theories of counterfactuals. The theory of epistemic counterfactuals has been developed by Rescher and Pollock. An alternate theory, the theory of metaphysical counterfactuals, has been developed by Stalnaker, Lewis, and Kvart. The metaphysical theory emphasizes the truth conditions placed on counterfactuals regardless of peoples' beliefs. The epistemic theory, on the other hand, attempts to account for beliefs in the analysis of counterfactuals. We base our system on epistemic counterfactuals since we are concerned with

reasoning about knowledge and beliefs.

The original Virginia Tech counterfactual engine was developed in [Whitehead, 1988]. A proof of its properties and a description of its solution to the frame problem appears in [Pendergraft, 1988]. A second version of the counterfactual engine that incorporates time reasoning has been developed and is discussed in [Khambati, 1989]. We will provide a formal characterization of counterfactual engine's operation, which has not been given for either version.

The counterfactual engine uses a non-standard form of implication rather than strict *modus ponens*. This approach is based on Rescher's theory of epistemic counterfactuals [Rescher, 1964]. A primary goal of the theory is to produce the minimal change necessary to resolve inconsistencies. This leads, however, to the important question of determining what we mean by "minimal."

A minimal change could be taken to mean the change that involves changing the fewest propositions. This does not seem to be the best solution since one important belief may be changed instead of two beliefs that were less believable in the first place. Some other way of defining minimal change seems necessary.

Rescher's approach groups beliefs into categories depending on how much support they have. Each of these categories receives a number referred to as its modal category. Thus, modal category 0 contains beliefs that cannot change, such as the basic axioms of logic. As the modal category number increases, beliefs become less certain and are more likely to change. The resolution of inconsistencies, then, involves changing those beliefs that have the highest modal category.

At the same time, as few of these are changed as possible.

Different possible worlds are generated because there may be several equally minimal ways to resolve the inconsistency. These possibilities represent maximally consistent subsets; this means they are the largest subsets of the inconsistent set to which consistency can be restored. We will next examine specifically how the counterfactual rules operate to resolve inconsistencies using the modal categories.

3.2.1. Operation of the Counterfactual Engine

We will describe the operation of the counterfactual engine when its propositions are interpreted in terms of our belief system. This description of the way rules resolve inconsistencies will lead to an easier understanding of the formalization. A counterfactual rule has the general form:

$$\begin{array}{l} P_1 \\ P_2 \\ . \\ . \\ P_n \rightarrow \begin{array}{l} q_1 \\ q_2 \\ . \\ . \\ q_i \end{array} \end{array}$$

In many cases, we will have only one statement on the right hand

side so that $i=1$. Both the right side and the left side should be interpreted as a conjunction of statements. The important fact to note is that the symbol " $\rightarrow$ " does not indicate implication or logical entailment. We will explain its use after defining some additional notation.

As indicated above, the p_n 's and q_i 's represent statements in our system's language; they do not represent propositions about the world; they represent an agent's beliefs about those propositions. In effect, these are propositions about propositions, and they may have a truth value of true or false. A statement is considered true in the system if it is contained in the system. Each statement, however, contains a proposition about the world that has some truth value. The statement maps a proposition and its truth value to a particular agent's set of beliefs.

In the development of the representation, we will specify that an agent may be uncertain about information; he does not necessarily believe that a proposition is strictly true or false. Instead, we believe that an agent has an opinion regarding his degree of confidence in the truth value of a proposition. These opinions depend on the agent's justification for believing the proposition to be true or false.

The opinion associated with a statement in our system indicates an agent's degree of certainty regarding the truth value associated with the proposition embedded in that statement. A higher number indicates a lower certainty. Although this may seem counterintuitive, we have arranged our system this way to correspond to the ranking of modal

categories; the higher the modal category, the more likely it is that a fact will change. Thus, the notation:

$\text{opinion}(p_n(t))$

denotes the number indicating the strength of opinion an agent associates with the proposition of statement p_n . The opinions are determined and justified based on the type of rules used to make an inference and the opinions about supporting propositions. Chapter Five will discuss how we determine the strength of opinions for propositions.

These opinions are then used as the modal categories in our system, and they are used by the counterfactual engine to determine its actions. The ranking of opinions coincides with the ranking of modal categories; the lower the number representing the opinion, the more certain the agent is about a proposition and the less likely it is that the proposition will change.

Note that although these opinions are not an explicit part of our statements, they are contained in the internal PROLOG representation of statements. Thus, the system has access to them internally and can operate on them automatically, even though they are not explicit in the representation. Thus, an observer will not see these opinions during the reasoning process, but he can assume that when a statement p_k appears, the system can retrieve $\text{opinion}(p_k(t))$.

With this notation for $\text{opinion}(p_n(t))$, we can define how the counterfactual engine operates on worlds. Although the symbol " $\rightarrow$ " does not precisely indicate implication, we can define it. The definition given below is similar to McDermott's M operator [McDermott,

1982]; the counterfactual engine can not only identify inconsistencies but can also resolve them.

Def. $\rightarrow^*$: If the conjunction of the statements $p_1, p_2, \dots, p_n$ is true, then the conjunction of the statements $q_1, q_2, \dots, q_i$ should be asserted to be true if it is consistent to do so.

Thus, the counterfactual rules are actually consistency rules, and the counterfactual engine acts as a system for consistency maintenance.

When a counterfactual rule operates on possible worlds, it is said to have fired. A rule will only fire if every p_n on its left side is true. There are several ways that a rule will operate when it fires. Essentially, the " $\rightarrow$ " is a conditional implication, with the conditions depending on consistency and the values of opinions.

3.2.1.1. Operation of Rules

The statements in our belief reasoning system have the following form:

PREDICATE list-of-arguments truth-value belief-set

This form is necessary because we are representing an agent's beliefs and nested beliefs, rather than simple propositions about the world. We will explain how this representation was developed in Chapter Four. The PREDICATE and list-of-arguments denote a proposition about the world,

* " $\rightarrow$ " will be used throughout this thesis as the symbol for counterfactual reasoning; it will never be used as an implication sign.

but the statement as a whole concerns an agent's belief about that proposition.

Counterfactuals may be new inputs to the system or statements asserted by the right side of counterfactual rules that have already fired. If the counterfactual does not match any statements on the left side of rules, it is added to the world; if it contradicts an existing statement, the old statement is rejected. If a counterfactual matches a statement on the left side of a rule, it is temporarily assigned an opinion (i.e. modal category) of -1; this prevents the counterfactual statement from being rejected. When the effects of that rule have been determined, the counterfactual becomes a normal statement and then regains the normal opinion (modal category) that it should have.

When the match is made, the counterfactual engine instantiates matching variables in other left hand statements of that rule. It then attempts to match all statements on the left side of the rule to statements in the current world, i.e. statements contained in its current knowledge base. If every statement on the left side is true, then the rule fires. When it fires, the variables contained in the q_i statements on the right side of the rule are instantiated. The rule may then operate in several different ways as follows (these cases are mutually exclusive):

If no inconsistency results, then

- 1) If the conjunction of the q_i statements can be asserted as true without creating an inconsistency (i.e. causing a contradiction with another statement in the world), then this is done.

If an inconsistency results:

2) If the opinions of the left side statements are all lower (lower modal category means more accepted) than all the modal categories of the right side statements, then the right side statements are asserted as counterfactuals that will attempt to fire other rules, and the previously held statements that contributed to the inconsistency are rejected.

3) If the opinion of a statement on the left side is higher (higher modal category means less accepted) than the opinion of all statements on the right side, then none of the right side statements are asserted and that left side statement with the highest modal category is retracted. If two statements on the left side have the same modal category which is higher than that of all right side statements, then two possible worlds can be generated to resolve the inconsistency (although only one is asserted at a time since the search is depth first).

4) If the modal category of the rule is higher than the opinions of all the statements, the rule may be rejected. The system maintains lists of which statements are supported by which rules. If a rule is rejected, then the supports for all statements previously inferred from that rule are checked to determine if the statements have some other basis for belief. If the rejected rule is the only support, then the statement must also be rejected. If the statement is also supported by other rules, then it is not affected because it has alternative justification.

Note that action 3 may result in several possible worlds if there is more than one left side statement having an opinion higher than than

the right side statements. The system, however, operates with a depth first search; only one of these worlds will be generated unless we later backtrack to look at other possibilities.

Another important feature is the " $\rightarrow$ " operator. While the " $\rightarrow$ " operator applies only a restricted *modus ponens*, the " $\Rightarrow$ " operator applies both restricted *modus ponens* and restricted *modus tollens*. Of course, when we say that these logical operations are applied, we mean they are applied if it is consistent to do so. Otherwise, the inconsistency must somehow be resolved.

If we have a statement such as:

$p \Rightarrow q$

it is logically equivalent to having the two statements:

$p \rightarrow q$

$\neg q \rightarrow \neg p$

There may be some cases where this property is desired. At the same time, *modus tollens* may not be necessary with some rules, and " $\rightarrow$ " can be used to avoid generating undesired possibilities. This is especially important in dealing with the frame problem; we do not want to make unwanted inferences that will slow down the system and that do not add new, important information.

The advantage of inhibiting *modus tollens* can be illustrated with an example. McCarthy gives an example involving a car with a potato in the tailpipe; Pendergraft's planning system also utilizes the Virginia Tech counterfactual engine but uses a different representation. We will also use an example involving a disabled car.

If the car is in some way disabled, we should be able to realize

that the car will not run. Yet, if the car runs fine, there is no need to enumerate all of the possibilities that have not gone wrong. For example, we may have the simple rule:

DISABLED CAR True set1 $\rightarrow$ RUNS car False set1.

If the car is disabled, then we recognize that it will not run. We do not want to apply restricted *modus tollens* to this rule, however, because if the car runs (RUNS CAR True set1) we do not need to conclude that it is not disabled (DISABLED CAR False set1). Furthermore, we may not want to apply restricted *modus tollens* because of the problem of non-controversion. This problem occurs when we attempt to take the contrapositive of a causal relation (we are assuming that a cause is sufficient but not necessary). For example, if p causes q we may write:

$p \supset q$

but it would not make sense (with the stated view of causality) to write:

$\sim q \supset \sim p$

because $\sim q$ does not cause $\sim p$. Hence, the two types of operators give the system greater flexibility.

The counterfactual engine does not use either of the open or the closed world assumptions. The closed world assumption [Reiter, 1978] means that all positive literals are known; from this it follows that a literal is false if it is not known. The weaker version of the closed world assumption means that all positive literals are known or derivable. The open world assumption [Nutter, 1983] means that all positive and negative literals are known, so literals that are not represented are unknown.

Our system explicitly represents positive and negative propositions, but this does not mean that propositions that are unknown in a possible world are necessarily unknown. Thus, information may be unknown to an agent at a particular time (i.e., in a particular possible world), but the agent may extend his model to include that proposition (or its negation) in some other possible world.

3.2.1.2. Possible Worlds and Timestamps

A partial possible world is defined by a set of statements that represent what an agent believes. When counterfactual rules are fired and statements change value (i.e. agents revise their beliefs), new possible worlds are generated. Statements indicate a relationship between a proposition (or its negation) and an agent; the statement attaches a set label to a proposition to indicate that the proposition is part of that agent's belief set at a particular time. The entire set of an agent's statements in a given possible world form that agent's model of a portion of the real world. The relationship between propositions and statements will be explained in Chapter Four and Chapter Six.

If no statement containing a given proposition occurs in a possible world that represents an agent's beliefs, then the truth value of that proposition is unknown to the agent. Since there are many things about the world that an agent does not know, the possible worlds we generate are partial possible worlds. The partiality of these possible worlds enables us to avoid making either of the closed or open

world assumptions, and it allows agents to extend their models of the world. If we assume that agents have complete knowledge of the types of causal relations possible and that the relevant entities to a situation are all named, then given sufficient information an agent can extend his model of the world to include the relevant propositions (or their negations). These assumptions sacrifice some representational power, but they insure that all propositions have a truth value that can be determined (even if an agent may never acquire enough information to determine the truth value).

Since the set of statements changes with time, a possible world can be associated with a time; the timestamp of a possible world indicates when that set of statements and only that set of statements are believed by an agent. Each possible world has a timestamp. This means that timestamps are an integral part of our system.

A system using counterfactuals requires a sense of time because both the epistemic and metaphysical schools of thought about counterfactuals agree that time is inherently a part of counterfactuals [Lewis, 1986b]. The timestamps we use are not absolute; they indicate relative time between possible worlds. Our system has several interesting abilities that result directly from the use of these timestamps.

One important characteristic of our system is that it has a memory of past beliefs. Unlike systems that store only the current model of the world, ours keeps track of past worlds and can inspect them at later times. This leads to an ability to reason about time and to deduce temporal relations between beliefs. Also, the use of timestamps

enables us to model the time taken by the reasoning process by ordering mental events and the inferences they produce.

Timestamps are not represented explicitly in our statements. Like the numbers indicating opinions, they are represented internally and are accessible to the system. We will examine the frame problem in Chapter Eight and see how the timestamps enter into the solution of the frame problem. We will see that timestamps enable us to transform non-monotonic belief maintenance into a monotonic system.

3.2.2. Causality

The counterfactual engine utilizes the principle of local causality; it assumes that rules need only specify the immediate effects of events. This locality does necessarily imply closeness in time and space or theoretical relationship; it may be either. Local causality means that an event will directly affect only a subset of all facts or beliefs that are relevant to the event. Counterfactual rules are used to capture these direct causal relationships; by transitivity, all effects of an event will be realized after intermediate effects are asserted and cause additional changes. This is in contrast to STRIPS [Fikes and Nilsson, 1971] type systems that must specify every change in the world that results from an event. The result is that rules are easier to write because extended effects need not be considered. We should note that the level of detail represented in a situation may affect what effects are considered local and direct.

As an example, consider a switch turning on a lightbulb. When the

switch is closed it allows electricity to flow through a wire. The wire guides the electricity through a filament in the bulb; the resistance of the filament to the electricity causes the filament to heat up, and the heat causes the filament to glow. A STRIPS type approach might require the following model:

OPEN SWITCH -> THROUGH WIRE ELECTRICITY
THROUGH FILAMENT ELECTRICITY
HEATS FILAMENT
GLOWS FILAMENT

Our system, on the other hand, would model the system as:

OPEN SWITCH -> THROUGH WIRE ELECTRICITY
THROUGH WIRE ELECTRICITY -> THROUGH FILAMENT ELECTRICITY
THROUGH FILAMENT ELECTRICITY -> HEATS FILAMENT
HEATS FILAMENT -> GLOWS FILAMENT

Thus, every change in facts or beliefs is assumed to have a direct cause. The assumption that every change of belief has a cause is necessary to avoid random inferences; beliefs (or facts) should not change unless there is *some* cause. We are not implying that there is a one-to-one relationship between causes and effects; the relation is many-to-many.

Note that causality extends only to the way rules are written and operate; it does not extend to the way modal categories (or opinions in our case) are assigned. Simon and Rescher linked a theory of causality to the theory of epistemic counterfactuals [Simon and Rescher, 1966]. In this theory, they assign a statement p_i to a modal category lower than statement p_j just in the case that p_i causes p_j .

The theory for assigning modal categories would not work for our system because statements on the left side of our rules are assumed to cause statements on the right side of rules. Hence, Simon and Rescher's scheme would always assign lower modal categories to the statements on the left side of rules than to statements on the right side because of the causal relations indicated by the rules; the counterfactual engine would never reject statements from the left side of rules. Our method of determining opinions, which we will explain in Chapter Five, assigns the highest modal categories to the propositions that are least certain and therefore most likely to change. Causality is not specifically considered in this determination.

3.2.3. Formalization

We will now formalize the operation of the counterfactual rules; while previous work has formalized the properties of the counterfactual engine, there has been no formalization of the way the rules resolve an inconsistency. This formalization will be given using the language of counterfactual rules in a manner similar to the way MLISP can be used to describe LISP's operation.

The statement p_i occurring at time t will be denoted as $p_i(t)$. As previously, statements on the left side of rules will be represented by p_i 's and statements on the right side of rules will be represented as q_j 's. The opinion of a statement p_i at time t will be indicated as $\text{opinion}(p_i(t))$. Now, assume we have a rule, R_k , of the form given in section 3.2.1. Note that the counterfactual engine treats the

statements on one side of a rule as a conjunction, even though they are written on separate lines without connectives; we will use a shorthand representation by letting '&' denote conjunction. R_k then becomes:

$$R_k: p_1(t) \& p_2(t) \& \dots \& p_i(t) \rightarrow \\ q_1(t') \& q_2(t') \& \dots \& q_j(t').$$

The modal category of the rule R_k will be designated as $\text{opinion}(R_k)$. The modal category of a rule will not change in the present system, so there is no time associated with it.

The arrow of R_k indicates ordinary counterfactual reasoning. As long as no inconsistency results from asserting the q_j 's, they will be asserted whenever the p_i 's are satisfied. This means that the counterfactual rules operate as ordinary implication rules as long as it is consistent to do so.

If an inconsistency does result, then the counterfactual engine detects it and resolves it; the operation of the counterfactual engine in that case can be formalized by the following mutually exclusive counterfactual rules:

- 1) $p_1(t) \& p_2(t) \& \dots \& p_i(t) \&$
 $\sim q_x(t) \&$
 $\text{opinion}(q_x(t')) > \text{opinion}(p_y(t)) \text{ for all } y \rightarrow$
 $q_1(t') \& q_2(t') \& \dots \& q_j(t')$
- 2) $p_1(t) \& p_2(t) \& \dots \& p_i(t) \&$
 $\sim q_x(t) \&$
 $\text{opinion}(p_y(t)) > \text{opinion}(q_x(t')) \text{ for some } y \rightarrow$
 $\sim p_y(t')$

3) $p_1(t) \& p_2(t) \& \dots \& p_i(t) \&$
 $\sim q_x(t) \&$
 $\text{opinion}(R_k) > \text{opinion}(p_y(t))$ for all y &
 $\text{opinion}(R_k) > \text{opinion}(q_j(t'))$ for all j $\rightarrow$
 $\sim R_k$

3.3. Conclusions

Our system consists of possible worlds containing sets of beliefs. The counterfactual rules specify how combinations of beliefs may generate new beliefs. In addition, they specify how the beliefs should be revised to create new possible worlds if there is some inconsistency in a belief set. With this knowledge of how our system generates new worlds, we next show the development of the representation for the statements that make up our possible worlds.

4. Representation of Beliefs

One of the fundamental problems in designing a system to reason about beliefs is choosing a representation that permits nested beliefs. This problem was further complicated because we desired a system with multiple agents reasoning about other agent's beliefs. Several systems have been developed that permit nested beliefs, but we developed our own representation to make explicit use of the possible worlds semantics and the use of counterfactuals for belief revision. We will begin by developing a representation for use with a single agent's beliefs and will later expand it to accomodate multiple agents' beliefs.

4.1. A First Attempt

We made an initial attempt to solve this problem using a belief predicate similar to the method used in [Genesereth and Nilsson, 1987]. Genesereth and Nilsson use predicates for knowing and believing to replace the modal operators. One possibility for such a representation is as follows:

(*agent *belief *endorsements)

where *endorsements indicate the uncertainty and justification for the belief. This representation would allow nested beliefs of the form:

(*agent1 (*agent2 *belief *endorsements2) *endorsements1)

which would be read as: agent one believes with endorsements1 that agent two believes *belief with *endorsements2.

In this scheme, beliefs are treated as objects and are associated with a label (argument) representing the agent who holds that belief. In a similar manner, labels for agents could be associated with counterfactual rules. This would allow different agents to have different reasoning capabilities. Furthermore, each agent could have a model of what rules he believes another agent may use. In this way, an agent could make deductions in another agent's belief system with what he believes the other agent believes. This would enable him to construct a view of how another agent sees him.

Konolige's deduction model also uses predicates for knowing and believing, and it handles nested beliefs using an attachment rule [Konolige, 1985]. This rule attaches the external meanings of belief atoms to computations within his deduction structures, allowing an agent to make a deduction using another agent's deduction rules if he knows that agent's deduction rules. We can implement a similar feature by attaching set labels to counterfactual rules. These labels indicate that some agent believes another agent can use that rule for making deductions.

Unfortunately, a representation that uses belief predicates (a syntactic method) is cumbersome and the intended meaning of a statement using nested predicates is difficult to comprehend when several belief predicates are scattered throughout the statement. Nested predicates also make it difficult to manipulate statements during the reasoning process because the belief predicates are embedded within the statement. In addition, the length and complexity of statements represented this way prevents their meaning from being very intuitive.

Thus, we wish to develop an alternative representation that does not use belief predicates.

4.2. Requirements for a Representation

Before proceeding with the development of a representation, we should consider the requirements of a representation that will support belief revisions. Some specific axioms are needed to determine what may be deduced from beliefs. Peter Gardenfors sets forth general axioms for the revision of belief sets in [Gardenfors, 1988]. These axioms are:

K*1) For any sentence A and any belief set K, K^*A is a belief set.

K*2) $A \in K_A^*$

K*3) $K_A^* \subseteq K_A^+$

K*4) If $\neg A \notin K$, then $K_A^+ \subseteq K_A^*$

K*5) $K_A^* = K_{\perp}$ if and only if $\vdash \neg A$

K*6) If $\vdash A \leftrightarrow B$, then $K_A^* = K_B^*$

K*7) $K_{A \& B}^* \subseteq (K_A^*)_B^+$

K*8) If $\neg B \notin K_A^*$, then $(K_A^*)_B^+ \subseteq K_{A \& B}^*$

where K_A^* means the revision of set K by the proposition A, K_A^+ means the addition of proposition A to set K, and $\neg A$ means the negation of A.

These axioms are very general and mainly specify set relations among revised belief sets; they do not help us determine which beliefs to revise. They do, however, provide a general framework of constraints on the sets of beliefs that may be generated when beliefs

are revised, and the counterfactual engine satisfies the constraints of this framework. The system presented in this thesis addresses the more specific problem of deciding which beliefs to revise.

While general constraints do not help us solve specific problems, it is important to note that the counterfactual engine does fit the general framework; otherwise, it would not be a good tool to use in solving the more specific problems. The counterfactual engine restores consistency to the world with the minimum number of changes to the world, and the changes that do occur affect those beliefs that are least strongly believed (i.e., they have the highest modal categories).

The counterfactual engine is essentially a mechanism that finds maximal size, maximally consistent subsets. The counterfactual engine prohibits an agent from believing contradictions, but it resolves inconsistencies by rejecting the smallest number of the weakest beliefs that will restore consistency. Prohibiting beliefs in contradictions is one of the valid axioms for belief that Genesereth and Nilsson describe in [Genesereth and Nilsson, 1987]. However, there are few axioms specific to beliefs contained in the literature.

Note that Gardenfors' axioms actually concern knowledge rather than beliefs. One of the reasons that knowledge axioms in general are invalid for beliefs is that beliefs are not necessarily true. Hence, the non-monotonicity of belief revision causes problems. The counterfactual engine handles this problem by allowing beliefs to be retracted, but there are other difficulties with reasoning about beliefs. One of these is the problem of logical omniscience [Genesereth and Nilsson, 1987]. Because agents have finite mental

capacities, they cannot deduce all of the logical consequences of their beliefs, and they cannot test consistency reliably. Still, they must be able to determine some of these consequences. Control over what deductions are allowed must be used to prevent an explosion of possible worlds.

Although there are few belief axioms to be found in the literature, common sense leads to some conclusions. For example, a rational agent will usually have a stronger belief in what he discovers for himself than in what is learned secondhand. Also, he will have more confidence in beliefs that are corroborated than those that are not. In addition, if something is believed to be true but later evidence suggests that it may not be the case, then an agent must weigh the supports and decide to accept the new beliefs, retain the old beliefs, or suspend belief in the old and the new propositions. If the old beliefs are rejected, then they must be modified to indicate that they are no longer believed. Thus, an agent must be able to assign varying degrees of support to beliefs, to compare these supports, and to revise beliefs.

It appears, then, that there are several problems to be solved. If the counterfactual engine is to be used effectively, a good representation must be found that will enable reasoning about nested beliefs. In addition, a method of representing and reasoning about uncertain beliefs must be used, and a method for representing what is unknown is needed when neither a proposition nor its negation is in the database. Finally, endorsements (or rationales) should be used to justify inferences and to determine which beliefs should be inferred

and which beliefs should be retracted (e.g. because premisses are uncertain or deception may be involved). We do not use endorsements to the extent that [Cohen, 1985] proposed using them (nor do we know of any system that has), but we do determine how to reason using levels of certainty and to explain how those certainties were determined. Chapter Five discusses how we represent uncertain states of belief and how the level of uncertainty is determined.

We will next develop a representation that will enable us to reason with nested beliefs.

4.3. Development of a Representation

One alternative to nested predicates uses distinct sets of possible worlds to represent nested beliefs. Each of these sets of possible worlds represents beliefs at a different level of nesting. Assuming agent A1 to be the primary agent, one set of possible worlds represents A1's beliefs at different stages of reasoning. A second set of possible worlds represents what A1 believes a second agent, called A2, believes. A third set represents what A1 believes A2 believes A1 believes.

An accessibility relation permits each possible world to have access to certain possible worlds in other sets. For example, an agent reasoning about his own beliefs could take into account some model of an opponent's beliefs. The accessibility relation indicates which model of another agent's beliefs should be used at a particular time. Essentially, the accessibility relation establishes a physical link

between sets at different levels of nesting that allows them to interchange information during the reasoning process. Thus, reasoning at one level uses beliefs from a different level of nesting, and the accessibility relation is used to capture nesting of beliefs. The following definitions will aid in describing this system:

Let A_1 denote a primary agent (i.e. the agent whose viewpoint is being described).

Let A_2 denote a secondary agent (i.e. another agent whose beliefs A_1 wishes to reason about).

Recall that we are developing the representation for a single agent's beliefs, so we will not yet represent A_2 's actual beliefs; instead, we are representing A_1 's model of A_2 's beliefs. One way to look at this is that the system views everything from A_1 's point-of-view, and A_2 is the only other agent in the system. When we develop the multiple agent representation, we will be able to represent the beliefs of several agents and their nested beliefs about each other.

One of the problems we face is representing and storing changes to a world; the counterfactual engine addresses this problem by representing a possible world by the changes it makes to a previous world. In our system (where the worlds are mental worlds consisting of beliefs), these changes will represent new beliefs the agent has adopted and old beliefs that have been retracted. Thus, each possible world will represent a set of beliefs that an agent holds after some new information is reconciled with the agent's previous beliefs. Each world can therefore be thought of as a stage of reasoning or the

agent's beliefs at a particular instant in time.

Consider a set of partial possible worlds $B(0)$ where each element of $B(0)$ is a possible world representing A_1 's set of beliefs at some stage of reasoning. Let $B(i,0)$ denote the i th possible world from the set $B(0)$, where i is an integer used to refer to a particular element of $B(0)$ (for shorthand, we tend to refer to $B(n)$ as B_n). In practice, this designation could be replaced by a distinct name for each world (e.g. gensym11, gensym12, etc.).

Next, consider a set of worlds B_1 where each element of B_1 is a possible world representing A_1 's model of A_2 's mental state at some stage of reasoning. Let $B(j,1)$ denote the j th possible world from the set B_1 , where j is an integer used to refer to a particular element of B_1 .

Also, consider a set of worlds B_2 where each element of B_2 is a possible world representing A_1 's model of A_2 's model of A_1 's mental state at some stage of reasoning. Let $B(k,2)$ denote the k th possible world from the set B_2 , where k is an integer used to refer to a particular element of B_2 .

Next, define an accessibility relation as follows:

$$\text{ACC} \{B(n,m), B(i,j)\} \quad n,m,i,j \text{ integers}$$

indicating that from the world $B(n,m)$ A_1 has access to the world $B(i,j)$. If we consider any world P we know that the accessibility relation for P will be contained within P . Hence, we may abbreviate the representation to indicate only the world that may be accessed. The world P that has the access will be implicitly indicated by the presence of the accessibility relation in P . This relation now has the

form:

$$\text{ACC } \{B(i,j)\} \quad i,j \text{ integers.}$$

If A1 has access to some world then A1 believes all of the beliefs in that world to be true with respect to the level of nesting. For example, if $B(x,2)$ contains the belief that "HAS John book" then A1 believes that A2 believes that A1 thinks John has the book. A1 believes that A2 believes that A1 holds this fact to be true even if, in fact, A1 does not think that John has the book. This indicates that inconsistencies are permitted between worlds related by an accessibility relation. This notation appears cumbersome, but it leads to a simplification (that we discuss below) in which the ordering of B's parameters and the definition of the various B's need not be memorized.

It is important to remember that the worlds we are discussing consist of beliefs rather than facts about the real world. This means that the statements comprising a world contain propositions whose truth values are interpreted relative to a set of beliefs. In other words, a proposition may be true but an agent may believe it is false; this would result in the proposition being false in that agent's belief set. Thus, the proposition is false on an interpretation (the agent's set of beliefs) but is not false in the real world.

This shows that beliefs exist in an agent's mind; if we are dealing with A1's beliefs, the possible worlds are constructed in A1's mind. Since a person is capable of distinguishing his own beliefs from those that he believes belong to others, A1 can change his model of

what he believes A2 believes without changing his own beliefs. He can also modify what he believes A2 believes A1 believes without changing what A1 himself actually believes. This indicates that the beliefs contained in a world are independent of beliefs at a different level of nesting. Thus, the critical aspect of our representation is that it allows the counterfactual to operate on distinct possible worlds by explicitly defining which statements comprise a given world.

4.3.1. Modal Logic and the Accessibility Relation

It is important that we consider our accessibility relation with respect to modal logic systems. Hughes and Cresswell discuss an interpretation of possible worlds based on the ability of an agent in one world to conceive of another world [Hughes and Cresswell, 1968]. A similar approach can be applied to this representation. It should be noted, however, that the analysis is complicated by the presence of worlds at different levels of nesting.

Our accessibility relation indicates which belief sets an agent may extract information from when reasoning. Although the reasoning at nested levels actually takes place in the primary agent's mind, the primary agent can reason as if he had the beliefs of some other agent. For example, when A1 wants to reason about A2's beliefs, changes in A1's model of A2's beliefs will be based on beliefs in A1's model of A2's beliefs and A1's model of A2's model of A1's beliefs. This indicates that each world must have access to itself and that the relation is reflexive.

The relation is not symmetric, however, because a world at a deeper level of nesting cannot have access to a world at a lower level of nesting. It is obvious that what A1 believes A2 believes A1 believes could not depend on what A1 actually believes unless A2 is telepathic; although A2's beliefs about A1's beliefs could be very strong, A2 can never have access to A1's actual beliefs, and A1 is aware of A2's limitation. Since the worlds of set B1 and B2 do not have accessibility relations to the worlds of set B0, the accessibility relation is not symmetric.

We will now determine if the relation is transitive. Transitivity holds when $ACC(P1, P2)$ and $ACC(P2, P3)$ implies that $ACC(P1, P3)$. Since we have established that only a world from B0 may contain an accessibility relation, we need only examine the cases where some world $B(i, 0)$ has access to some other world $B(j, 0)$ which then has access to a third world $B(m, n)$ where i, j, m, n are integers. The accessibility of $B(j, 0)$ to $B(i, 0)$ means that A1 can recall previous mental states and reason about the beliefs he held at that time. $B(n, m)$ is either a model of A1's beliefs at some other stage of reasoning or a model of some nested level of beliefs.

For transitivity to hold, we must allow A1 to reason not only about his own previous beliefs, but also about his previous models of other agents' beliefs. This is something that people are quite capable of doing, e.g. "When you saw me fighting with John yesterday I think you must have believed that I dislike John." Although it may be difficult to implement and control, transitivity of the accessibility relation is valid. Since the relation is reflexive and transitive but

not symmetric, it too coincides with the possible worlds structure associated with an S5 modal logic.

An example will help to clarify the structure. Figure 1 depicts a possible scenario. Note that $B(0,0)$ represents John's initial mental state, and $B(0,1)$ represents John's model of Mary's initial mental state. If a new belief is introduced into some world, the arrows leaving that world will indicate possible worlds that are acceptable revisions. To qualify as an acceptable revision, a world must incorporate the new belief and maintain consistency in the belief set with minimal change.

Initially, John believes that a table is in the kitchen, Mary is in the kitchen, and a chair is in the hall. He also believes that Mary believes she is in the kitchen and that the table is in the kitchen. Now, assume that someone tells John that the chair is next to the table. Since John believes that two things that are next to each other must be in the same room, there is an inconsistency.

One way to resolve the inconsistency is to assume that the chair is actually in the kitchen. This is represented in world $B(1,0)$. Since Mary is in the kitchen, John believes that she would be able to see the chair there. So, John's model of Mary's mental state also changes and is represented by $B(1,1)$.

A second way of resolving the inconsistency is to assume that the table is actually in the hall. This new mental state of John's is represented in world $B(2,0)$. John also believes that Mary would not believe the table was in the kitchen if it were not there. His new model of Mary's mental state is represented by world $B(2,1)$.

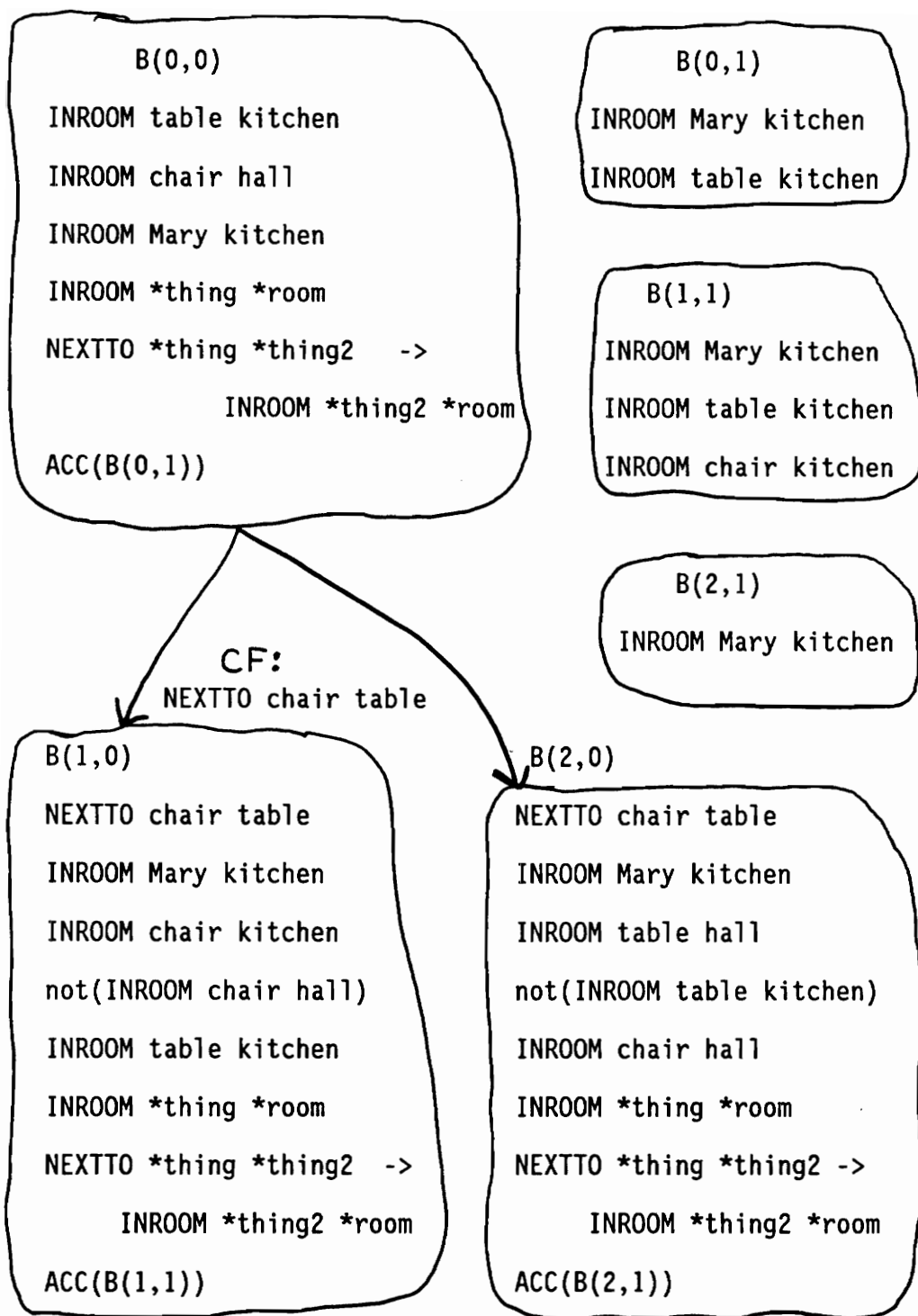


Figure 1. A possible scenario of beliefs at different levels of nesting.

There is, however, more than one possibility for generating world $B(2,1)$. If John believes that the table really was in the kitchen at some point and has been removed, then world $B(2,1)$ would be correct as shown in Figure 1. From John's point of view, Mary would not know where the table had been taken. If, on the other hand, John believes he was initially mistaken and the table was always in the hall, then he might now believe that Mary does know the location of the table, perhaps because she passed it on her way into the kitchen.

The system will be simplified if we say that for every world in B_0 and for each $m \geq 1$ there is a unique n that satisfies the relation $ACC(B(n,m))$. This means that an agent may only reason about one model of another agent's beliefs at a time. A_1 may consider different models of A_2 's beliefs, but he can accept only one of them as correct at a given time. If A_1 wishes to do hypothetical reasoning about A_2 's beliefs, he must introduce a hypothesis into his model of A_2 's beliefs and determine the resulting changes. If he is unsatisfied with the inferences, he may assert a different hypothesis in the next world and reject the previous hypothesis.

By limiting the number of worlds at nested levels that A_1 may access, we restrict him from comparing different models of another's beliefs; we have required that each model of another's beliefs be associated with a model of A_1 's beliefs. A_1 will be able to compare different models of another's beliefs consciously because he has access to each of his own sets of beliefs that are associated with those models. This is the accessibility relation among nested beliefs; we will formally define this relation in Chapter Seven.

Each of A1's mental states will consist of three worlds: A1's beliefs, A1's beliefs about A2's beliefs, and A1's beliefs about A2's beliefs about A1's beliefs; this level of nesting is sufficient to perform some deception reasoning without becoming tremendously complex. Any of these worlds may be identical to another world at the same level of nesting, but the total mental state will be different than every other group of associated worlds.

4.3.2. Extension of Accessibility to Include Multiple Agents

We next address the extension of our representation to handle multiple agent scenarios. This extension is straightforward. Our new relation will have the form:

$$ACC \{B(m,n,p)\} \quad m,n,p \text{ integers.}$$

In this representation, m is an index to indicate an individual world in a set, n indicates the level of nesting, and p indicates the agents between whom the nesting occurs. The system must be informed of the relation between the integers p and pairs of agents.

As an example, consider that following table:

p	primary agent	secondary agent
x	John	Bill
y	Bill	John
z	John	Mary

With these relations in mind, the following worlds would represent

these levels of nesting:

representation	level of nesting
$B(i,0,x)$	John believes
$B(i,1,x)$	John believes Bill believes
$B(i,0,y)$	Bill believes
$B(i,1,y)$	Bill believes John believes
$B(i,2,z)$	John believes Mary believes John believes

where i is simply an integer indicating a single world from the set.

Since agents are not able to read each other's minds, there can be no accessibility relations between worlds with different values for p , the third argument. For example, world $B(0,0,x)$ could not contain the relation:

$$ACC \{B(0,0,y)\}.$$

This would allow John to reason with Bill's actual beliefs. We do not permit this. John may only reason with his own model of Bill's beliefs, $B(i,1,x)$. The sets of worlds associated with each value of p will interact exactly as in the single agent model that we developed. Since no accessibility relations are permitted between sets with different values of p , the reflexive and transitive properties of the relation still hold.

We can now specify our accessibility relation formally as:

$$ACC(B(i,j,k), B(l,m,n)) \text{ if and only if}$$

$$1) i=l$$

$$2) j \leq m$$

$$3) k = n$$

The first requirement indicates that an agent can only believe one model of another agent's beliefs at a time. The second requirement indicates that beliefs at a given level of nesting are not accessible from belief sets at a deeper level of nesting. The third requirement means that accessibility must be between belief sets of the same primary agent; there is no accessibility between the actual beliefs of separate primary agents.

Furthermore, we have that:

$$ACC(B(i,j,k), B(i,j,k))$$

and if: $ACC(B(i,j,k), B(l,m,n))$ and $ACC(B(l,m,n), B(x,y,z))$

then: $ACC(B(i,j,k), B(x,y,z))$

and if: $ACC(B(i,j,k), B(l,m,n))$

then: $\text{not } (ACC(B(l,m,n), B(i,j,k)))$ which indicates that the relation is reflexive, transitive, and anti-symmetric.

We are now able to represent a system with any number of agents. This is a powerful capability and arises from our explicit representation of an accessibility relation. This approach differs from typical possible world systems. Typically, possible worlds grow in a tree-like fashion and accessibility is shown graphically by connecting branches. Accessibility is usually implicit. A world p generally has access to all the worlds that grow from it and all the worlds that grow from those and so on. Our system becomes complex because we allow nested beliefs and also the ability to reflect on other possible mental states. The use of an explicit accessibility relation helps to handle

the complexity and insures that agents cannot gain access to another's actual beliefs. Figure 2 shows how the accessibility relations might be drawn for our system.

The explicit accessibility relation we have developed gives us powerful control over what information can be used to make inferences. This control will enable us to attack a problem that often plagues reasoning systems. This problem involves substitution under equality which is discussed in Chapter Nine.

The representation presented gives us a way to express our sets of worlds at different levels of nesting. It also provides a way of specifying the accessibility relation between worlds in different sets. We next examine a method for implementing this representation in the counterfactual engine, and we determine what form the accessibility relation should have.

4.4. Implementation of the Representation Scheme

There are several requirements of the representation and accessibility relation. Direct inconsistencies among beliefs in a given world cannot be allowed; the nature of the counterfactual engine prevents this if the rules and initial worlds are consistent. This means that the initial facts or beliefs cannot contain a contradiction (or implied contradiction) and the rules cannot imply a contradiction. Inconsistent beliefs at separate levels of nesting must be allowed, however, to model cases where agents have differing opinions. Also, there must be an explicit way to indicate which worlds a given world

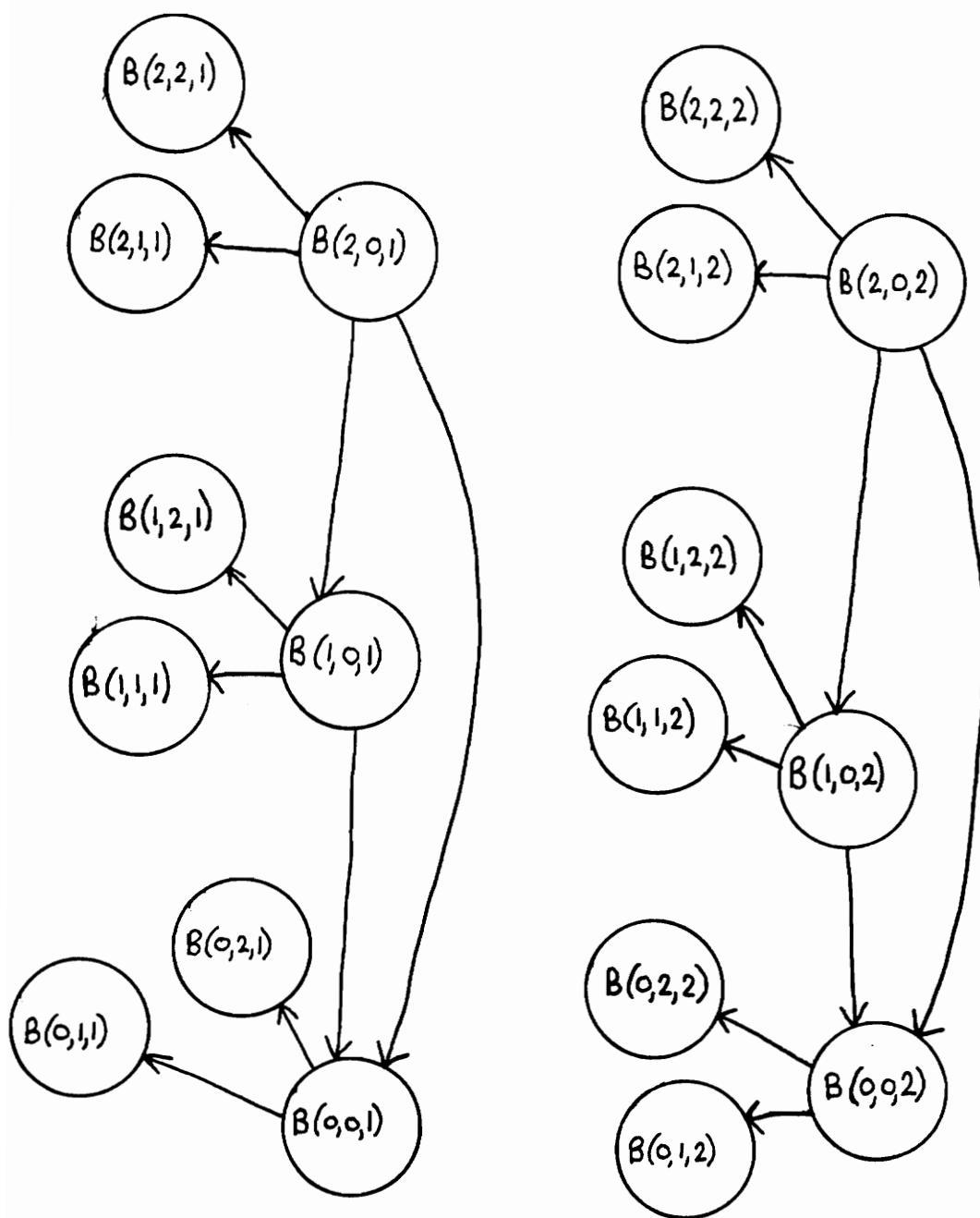


Figure 2. Accessibility Relations Depicted Graphically

can extract information from when reasoning (i.e., the accessibility relation).

At the level of the counterfactual engine, it would be ideal to separate worlds at different levels of nesting into distinct possible worlds. This would correspond more closely with the theoretical model of representation. Without major revisions to the counterfactual engine, however, it is unclear how the accessibility relations could be made explicit and how information could be passed between worlds with the current system. For these reasons, we have separated the levels of nesting within a single possible world, and we will let different possible worlds represent different stages of the reasoning process. At some future time, we may wish to modify the counterfactual engine to try a different approach.

Currently, an agent has a single model of another agent's beliefs at any given time; he also has a single model of the second agent's beliefs about the first agent's beliefs. Thus, an agent will have his own beliefs and these two models for another agent. We may now simplify our representation by considering these three related worlds to be part of the same possible world, $P(0)$. Thus, agent A1 may use any of these beliefs in the reasoning process, since they are all accessible to him. Figure 3 illustrates how the sets of beliefs from different levels of nesting are combined.

We will use the counterfactual engine to generate worlds at different stages of reasoning, but each world generated will contain all of the the belief sets $B(i,j,k)$ having identical values for i . Thus, we will require some other means of implementing the distinctions

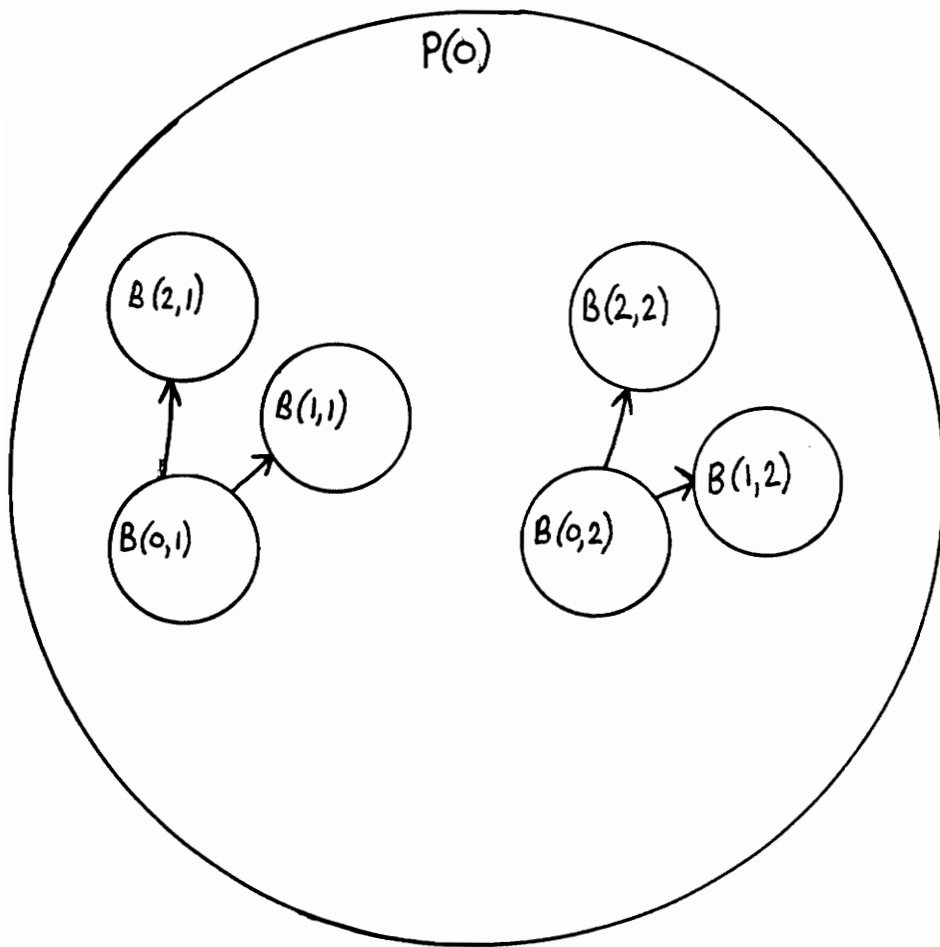


Figure 3. Relation between belief sets from distinct levels of nesting.

among the j and k parameters, i.e., some way of identifying which belief set a statement belongs in.

For example, if we have the belief

INROOM map room1

and there are two agents, John and Bill, then how do we interpret the belief? Does John believe the map is in room1? Does John believe that Bill believes the map is in room1? Perhaps John believes that Bill believes that John believes the map is in room1.

This ambiguity is resolved by grouping beliefs from the same level of nesting into sets and labelling each belief with the name of the set to which it belongs. For example, the set of statements that A1 (or John) believes, formerly called B(1,0), may be labelled set1. The model of what John believes Bill believes may be set2, and the model of what John believes Bill believes John believes may be set3.

These nesting relationships will be represented by, respectively:

SET John nil nil set1

SET John Bill nil set1

SET John Bill John set3

Nil is used to maintain the same number of parameters in each SET statement.

These set labels perform the function of defining the p value and n value discussed in the previous section. This means that a given level of nesting between a distinct pair of agents has a unique set label. An advantage of this representation is that p values can be related to a tertiary agent as well as the primary and secondary agents. For example,

SET John Bill Mary set6

can be used to refer to the set of John's beliefs about Bill's beliefs about Mary's beliefs.

The use of set labels precisely defines the set to which a belief belongs. When rules are written, a SET statement must usually be included for each set label used in that rule; this requirement is not necessary for rules that apply to all belief sets (e.g., rules that deal with common knowledge). This inclusion of SET statements enables the system to determine if the rule should apply for a given pair of agents and level of nesting. An example of how a rule would operate will be presented after we discuss the interpretation of statements containing set labels.

If John believes the map is in room1, we have

INROOM map room1 set1.

Likewise, if John believes Bill believes the map is in room1, then we have

INROOM map room1 set2

Statements about the actual world can use the label "world" in place of a set designation. Agents could not have access to these statements, but a simulator for the system would require them. If we want to represent information contained in the system but not necessarily available to an agent we can use the world set label. If we want to store the whereabouts of the map, it could be represented as

INROOM map room1 world.

It is important to note that the world set label does not indicate global knowledge.

There are several important observations to be made about the representation. The first is that a statement may be believed at multiple levels of nesting simultaneously. This is handled by having multiple statements in the world that differ only in their set label. An example is

INROOM map room1 set1

INROOM map room1 set2

Although this may appear redundant, it allows beliefs at different levels of nesting to be treated independently. Suppose that John no longer believes the map is in room1, but he thinks that Bill still believes it is there. The statement in set1 may be changed without affecting the one in set2.

A second observation concerning the representation is that it allows inconsistencies between different levels of nesting, but not within the same level. Since we (hopefully) do not believe contradictions, the statements

INROOM map room1 set1

and INROOM map room2 set1

should cause an inconsistency in the system, since we are aware that an object cannot be in two different places at once, and we assume for convenience that John never calls any room by more than one name. It is possible, however, to have the statements

INROOM map room1 set1

INROOM map room2 set3

and INROOM map room2 set2

simultaneously. This concurs with the realistic fact that what we

think does not always agree with what other people believe we think. Also, what we think does not necessarily agree with what we believe someone else thinks.

Another observation is that our representation allows us to model different agents' beliefs and beliefs about each other. Suppose we add the statements

SET Bill set4 nil nil

and SET Bill John set5 nil

to our world. Now we can also represent what Bill actually believes and what Bill believes John believes. Again, there may be inconsistencies between different sets. For example, set4 may not agree with set2 because what Bill actually believes may differ from what John thinks Bill believes. Figure 4 illustrates how a single possible world is divided into several sets of beliefs.

The arrows in Figure 4 indicate which sets can extract information from which other sets when reasoning. Since all of these sets are within the same world within the counterfactual engine, the accessibility relation is controlled by the way that rules are written. Consider the following situation. An agent (*agent1) is in a room containing an object. A second agent (*agent2) is also in the same room. Now, barring any unusual circumstances (e.g. *agent1 is hiding something behind his back), *agent1 can assume that *agent2 is also aware that the object is in the room. Hence, perception can be handled by counterfactual rules. An example will illustrate simple perception and indicate how counterfactual rules use set labels.

Consider the following perception rule:

P(0)

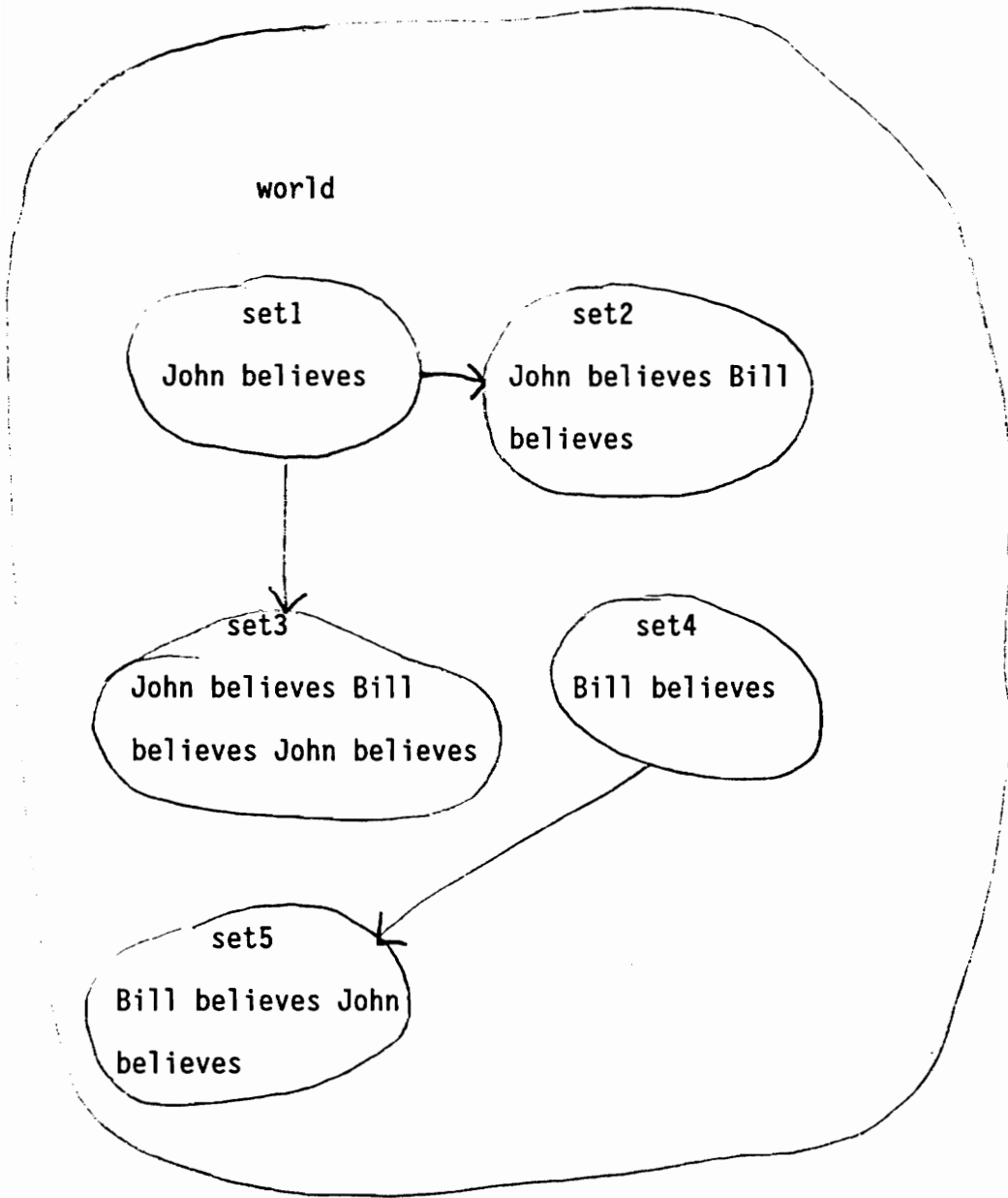


Figure 4. Division of a world into distinct sets of beliefs

```
INROOM *object *room *set1
INROOM *agent2 *room *set1
SET *agent1 *set1 nil nil
SET *agent1 *agent2 *set2 nil ->
      INROOM *object *room *set2
```

This rule simply allows agent's to perceive objects in the same room.

Note that the set label can be used to retrieve the name of the agent to whom the belief belongs. Also, the names of agents can be used to retrieve the label of the set that contains nested beliefs between those agents. This will be clarified by tracing through the rule.

Assume we initially have

```
INROOM map room1 set1
```

and now we learn that

```
INROOM Bill room1 set1
```

These match the first two conditions of the rule and instantiate *set1 as set1 and *agent2 as Bill. The third statement becomes

```
SET *agent1 set1 nil nil.
```

This statement matches the statement in our database that says

```
SET John set1 nil nil
```

which tells us that it is John who believes that both the map and Bill are now in room1. With *agent1 instantiated as John, the fourth statement becomes

```
SET John Bill *set2 nil
```

which matches our database statement that

```
SET John Bill set2 nil.
```

With `*set2` equal to `set2`, we can reach the conclusion that

```
INROOM map room1 set2
```

or, in other words, John believes that Bill assumes that the map is in `room1`.

This example demonstrates how different levels of nesting can interact and how such interaction can be controlled by using the set labels.

We should also point out that this schema for using labels for belief sets gives us flexibility in writing counterfactual rules. This is so because the counterfactual rules can be written to fire only when certain belief sets are used to match the left side of a rule. Thus, we can have different reasoning rules for different agents, and we can allow agents to reason about another agent's beliefs using different rules than he would use to reason with his own beliefs. In this thesis we will assume that all agents use the same counterfactual reasoning rules and will not utilize this capability.

4.5. States of Belief

To reason effectively with and about belief, some consideration must be given to the different kinds of status a belief may have. Propositions are either true or false. Beliefs, however, may have a status that is more ambiguous. Beliefs are propositions that are either true or false, but an agent may not know or be certain of the value; this is because agents do not have complete knowledge of the

world or the beliefs of other agents. As a result, an agent may have varying degrees of confidence in his beliefs. Therefore, an agent must be able to distinguish between unknown information, uncertain beliefs, and facts that are believed to be false.

4.5.1. Distinction Between Truth Condition and Epistemic Status

An agent must associate a truth condition (true or false) with every proposition in his belief set; in addition, there must be some epistemic status associated with that truth condition (e.g., unknown or some degree of uncertainty).

In some instances, an agent may have no belief at all concerning a fact. Notice that this is quite different from the case where an agent believes the proposition to be false. For example, suppose someone asks John whether Mary has blonde hair. If John believes that Mary has brown hair, then he will answer no. If, instead, John has never met Mary, then the information will be unknown to him, and he will be unable to answer. In either case, we would like John to justify his answer with additional information. For example, he could say, "No, Mary's hair is not blonde because it is brown." If he did not know the answer, he might say, "I do not know because I have never met Mary," but he would at least agree that it must be either blonde or not blonde. In other words, an agent is aware that he should not believe both a proposition and its negation.

Another example is McCarthy's well known problem, "Is the President standing or sitting?" We will explain our solution to these

problems in Chapter Five.

An agent's set of beliefs will contain some beliefs that he holds to be true. This does not mean that the belief is true in the actual world, nor does it mean that the agent has no doubt about that belief. For example, a person standing on Earth holds a map in front of him, four feet above the ground, with nothing between the map and ground. Because he has confidence in the laws of physics, this person believes that the map will fall to the ground if he releases it. He does not think it might fall or most likely will fall, but that it will fall; this is because he strongly believes that the proposition concerning the law of gravity is true. There are always possibilities that the map will not fall (e.g., it is caught in an updraft and blows away). Thus, the epistemic status is distinct from the truth condition because "believing something does not make it so."

4.5.2. Uncertain Beliefs

An agent will often be aware of his uncertainty. For example, suppose that John met Mary and Sally at the same time and believes that one of them had blonde hair. John may believe that Sally was the blonde, but he is slightly unsure because he only met the two briefly several years ago. If John met a third woman at the same time who had red hair, he might be even more unsure of which one was blonde. As another example, consider that Mary had brown hair, but someone told John that she was planning to bleach it to blonde. John now believes Mary is blonde, but he is uncertain because the report may have been

unreliable or Mary may have changed her mind.

In each of these examples, John has reason to believe something, but also has reason to be uncertain. Since these beliefs are possible, they must be represented in his belief set. There must be some qualifier, however, based on the support for the belief that indicates the level of confidence in the belief. The level of confidence in the belief will then affect the level of confidence in any beliefs inferred from that one.

4.5.3. Beliefs About the Unknown

An obvious problem is how to determine when something should be considered unknown instead of very uncertain. This problem is related to the size of the search space and to the agent's consciousness of that size. If there are only a few possibilities in a situation, the agent may choose to believe the most likely one until it is determined to be incorrect; he then may try another possibility.

If, however, there are many possibilities to make this practical, or if there is no basis to consider one possibility most likely, then the agent may simply consider the information unknown. Furthermore, if there are only a few possible solutions but the agent does not realize it, he will consider the answer unknown. Thus, an important consideration in distinguishing uncertain and unknown information will be the size of the search space, the agent's awareness of the size of the search space, and evidence to indicate one possibility is more likely than others.

As an example, suppose we wish to determine which of three boys a girl will choose to go out with, and there is no evidence (such as previous dating history with these boys) to indicate that one selection is more likely than the others. In this case, we may be able to determine the most likely candidate and believe that she will go out with him, but not with absolute confidence. On the other hand, if there are fifteen boys who ask this girl out, we may just decide that we do not know which one she will date. Some threshold for the size of search space must be chosen to make this distinction.

4.5.4. Beliefs that Something is False

Another problem is in representing that an agent believes something to be false, or that he believes another agent believes it to be false. When we say false we *do not* mean that the agent thinks his belief is incorrect; we mean that he believes the negation of the proposition. There are many situations that require an agent to believe the negation of a proposition. For example, if an agent is searching for an object and has looked for it in a specific room, then he should be able to remember that the object is not in that room. Even if he does not know which room contains the object, he will know which rooms do not contain the object and can avoid searching there again.

An agent should be able to represent explicitly that he believes something is false, i.e., John believes the negation of some proposition. Another case occurs when John does not believe something

is true. This is not equivalent to saying that John believes it to be false, i.e., disbelieves it. The problem with representing negations arises due to the representation used by the counterfactual engine; we will describe the problem and its solution below.

A very important case involves representing that one agent believes another agent has some information but the first agent does not know what that information is. Consider the example of a safe combination. How can we specify that John thinks Bill knows the combination without using the combination itself? This is a problem because John must refer to a value that he himself does not know. This will be important in competitive situations because an agent must determine if an opponent has better information without knowing what that information is. For example, his actions would be different if he thinks his opponent has the combination to the safe, so he must be able to reason about this even though he himself may not know the combination. We discuss this problem in Chapter Nine.

There are two cases remaining. Consider that John believes that Bill does not believe some proposition. This does not mean that John believes that Bill believes the negation of that proposition. It means that John believes either Bill is uncertain of the belief or that the information is unknown to Bill. Depending on the situation, John will either consider it unknown to Bill or John will decide Bill believes something possible with a low level of confidence.

The last case occurs when John does not believe that Bill believes some proposition. Again, this does not mean that John believes that Bill believes the negation of that proposition. It means that Bill's

belief on the subject is unknown to John. Figure 5 shows the different kinds of status that a belief may have.

Representation of the negations of propositions with a "not" prefix will not work with the counterfactual engine because the counterfactual engine uses this prefix for another purpose (we will explain this in Chapter Five). Representation of unknown information could be accomplished by absence from the database, but only if negative propositions are explicitly represented (or somehow derivable). We will examine these problems in Chapter Five.

A final, major problem is the unclear boundary between unknown and uncertain information. Intuitively, we know that people make educated guesses about situations even when they are uncertain. Consider the case where John believes that Bill does not believe some proposition. It is true that this does not mean that Bill believes the negation of the proposition; John might be mistaken, and Bill could actually believe the proposition. Yet, there may be reasons for John to believe that Bill believes the negation of the proposition; in this case, John will assume that Bill believes the negation of the proposition.

If the reasons for John's belief about Bill's belief are incorrect, his assumption may be wrong. Certain guesses must be made, however, to have enough information to reason with. John's belief that Bill believes the fact to be false can be qualified to show that John is not certain of his assumption. Chapter Five examines a better method for representing this uncertainty and for explicitly representing things believed to be false.

<u>Status</u>	<u>Example</u>
True	The map is in room1.
False	The map is not in room1.
Unknown	Nothing at all is known about the whereabouts of the map.
Uncertain	Many gradations of certainty are possible

Figure 5. Types of belief status.

4.6. Conclusions

Since the counterfactual engine timestamps possible worlds, we can use this timestamp to refer to belief sets instead of a separate integer numbering system. Furthermore, at any instant of time there will exist a group of belief sets for different agents and at different levels of nesting. For this reason, we will associate this entire group with an instant of time (a particular timestamp). Thus, this "superworld" will be a possible world composed of many smaller possible worlds; there will be one possible world for each belief set.

For example, if there are two agents in the world there will be six belief sets (or possible worlds) in the superworld. This follows since each agent is associated with three belief sets: his own beliefs, his model of the other agent's beliefs, and his model of the other agent's model of his own beliefs. Two agents times three belief sets equals six belief sets. Note that there is not a linear relation between number of agents and total number of belief sets. For example, if there are three agents then each agent is associated with five belief sets. This is because there are now two other agents in the world for whom the agent must have models of nested belief sets. Hence, three agents times five belief sets equals fifteen possible worlds in the superworld.

Since we are grouping belief sets together with timestamps, we now need only two of the indices of $B(i,m,n)$ to indicate a particular belief set; one index to indicate the agents involved, and another to indicate the level of nesting. We can redefine our set notation to

account for both of these indices. Let $\text{set}(x,y,z,)$ represent agent x 's beliefs about agent y 's beliefs about agent z 's beliefs. If y or z are nil instead of representing an agent, then there is one less level of nesting.

With this notation, the statement

SET John Mary John set1

in our system would mean that the label set1 is associated with the $\text{set}(\text{John}, \text{Mary}, \text{John})$. We then define our accessibility relations as follows:

$\text{set}(x,y,z)$ has access to $\text{set}(a,b,c)$ iff

$y=\text{nil}$

$z=\text{nil}$

$x=a$

Note that this notation enables us to represent nested beliefs involving three agents, such as John believes Mary believes Bill believes. We will concern ourselves only with nested beliefs between two agents. Figure 6 shows the relations between possible worlds in a superworld using this set notation.

SET John nil nil set1
SET John Bill nil set2
SET John Bill John set3
SET Bill nil nil set4
SET Bill John nil set5
SET Bill John Bill set6

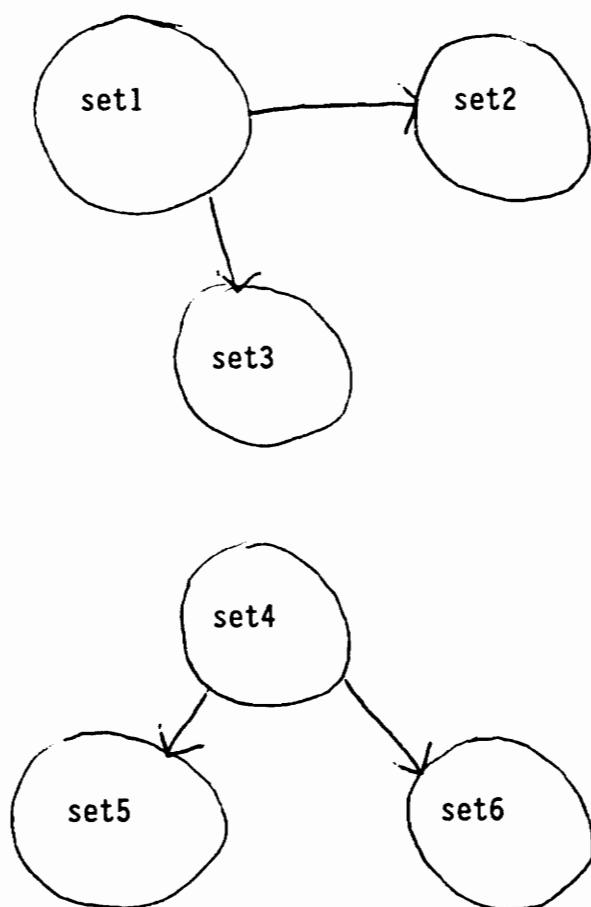


Figure 6. Accessibility between belief sets expressed with "set" notation.

5. Uncertainty and Unknownness

We described the various states of belief that we must represent in Section 4.5. The four major categories of information that concern us are: positive beliefs, negative beliefs, unknown information, and uncertain beliefs. The representation of positive beliefs is straightforward using the method of representation that we have developed. We will now focus on the problem of representing the other three categories of information.

5.1. Negative Beliefs

The first problem we will address is the representation of negative beliefs. Note that negative beliefs are not the same as false beliefs. A false belief is one that does not correspond to the real world. A negative belief, which may or may not be false, occurs when an agent actively believes that some proposition is not true. A false belief is represented by including a proposition that is not true in the real world in the possible world that represents an agent's beliefs. This possible world cannot correspond to the real world since it contains at least one false belief, but the agent is unable to judge whether his beliefs correspond to the real world because of his incomplete knowledge. Hence, an agent cannot distinguish when his current beliefs are false and needs no representation for false beliefs. When we refer to true or false beliefs, we are referring to propositions and their negations, respectively; we are not referring to

the truth condition of the proposition in the real world.

We need to represent not only that an agent believes the negation of a proposition, but also that he may think another agent believes the negation of that proposition. If an agent has the ability to reason with nested beliefs, then he should be able to represent any beliefs that he attributes to other agents as well as he represents his own beliefs.

If we wish to represent explicitly the negation of some proposition, this can most easily be done with a label such as "not" used as a prefix to the fact. Since there are many situations that require an agent to believe the negation of a proposition, we should examine each of them to see if a "not" prefix will work as desired. For this discussion, we assume that John has a set of beliefs about the world (set1) and a set of beliefs about what Bill believes about the world (set2). If John believes the negation of a proposition, the proposition would appear with the "not" prefix in set1. For example, if John believes that Bill is not his brother, the statement "not brother_of John Bill set1" would be used.

If John believes that Bill believes the negation of a proposition, then the belief appears with the "not" prefix in set2. If John believes that Bill believes that Bill is not John's brother, then the statement "not brother_of John Bill set2" is used.

This representation is therefore capable of handling negative beliefs, and it is not very complex to implement. Unfortunately, this representation will not work with the counterfactual engine. In the counterfactual engine, a 'not' prefix indicates that an agent does not

hold a belief; it does not indicate a negated belief.

We have tried using a different word as a prefix to indicate negative beliefs, but this presents other problems. The counterfactual engine treats the first word of a statement as the predicate that indicates a relation or attribute (e.g. INROOM, NEXTTO, HOLDS, BIG). Since the NOT prefix is an operator used by the system, the counterfactual engine can look beyond it to find the actual predicate of the statement. This would not occur if some other prefix were used; the result is that every statement concerning a negative belief has the same predicate. Hence, another method of representing negative beliefs must be found.

Fortunately, we can adapt our representation slightly to incorporate negative beliefs. Our method for representing negative beliefs involves a modification to the representation used for positive beliefs. This modification also allows a simple solution to the problem of unknown information. The remaining problem, representing uncertainty, will require significant additions to the representation scheme.

Ordinarily, negative beliefs are represented with a 'not' prefix or a tilde (~) before the proposition. Since we cannot use a prefix with the counterfactual engine because the prefix is used to distinguish predicates, it makes sense to place an identifier for negated propositions elsewhere in the statement. This label indicates the same operation as a tilde, but does not occur in front of the proposition. This label is not an argument of the embedded proposition in a statement; it is merely a flag to distinguish between the

proposition and its negation. The label should be considered a part of the proposition and is not the distinguishing factor between a proposition and a statement. Chapter Six describes in detail the distinction between propositions and statements.

In order to maintain a parallel structure, we should also label positive propositions with a descriptor. This approach is similar to the use of "yes" and "no" labels in [Barwise and Perry, 1983]. With our True and False labels, we can now give examples of our representation for negative beliefs.

If set1 represents John's set of beliefs, then John's belief that a map is in room1 would be represented as:

INROOM MAP ROOM1 True set1.

If John has searched room1 and knows that the map is not there, it would be represented as:

INROOM MAP ROOM1 False set1.

This indicates that an added descriptor will occur with each statement and will contain a flag, True or False. This explicit representation provides a simple way to distinguish between positive and negative beliefs. It requires us to include a rule that prohibits a statement from being true and false at the same time (since p and $\neg p$ is a contradiction that the counterfactual engine prohibits). This rule can easily be added to the list of consistency rules.

An example of such a rule is:

```
(*PREDICATE *ARG True *set
  *PREDICATE *ARG False *set -> (inconsistent))
```

The use of explicit flags allows agents to represent both

propositions and their negations; we wish to make clear that these flags do not refer to the truth condition of the proposition. An agent cannot know the truth condition of a proposition; he can only indicate an epistemic attitude towards the truth condition by including the proposition (or its negation) in the possible world representing his beliefs. His beliefs may be incorrect or false if they do not correspond to the real world, but the agent will not be aware of the false beliefs (although he may realize at a later time that his beliefs were false).

This points to a distinction between 'p' and 'p is true'. If a proposition is true in the real world then we say p; if the agent believes the proposition is true we say p is true. Thus, p is true indicates an epistemic status that an agent assigns to p; it means that p is true on the interpretation of the agent's beliefs. We can represent false beliefs by having $\neg p$ in the world and 'p is true' in the world corresponding to an agent's beliefs. In our representation this would be:

INROOM room1 robot True world

INROOM room1 robot False set1

A proposition's presence in the world indicates that its truth value is true; its presence in a belief set indicates that its truth value on an interpretation (the agent's beliefs) is true.

5.1.1. Representing Disjunctions

Since our representation can now support both p and $\neg p$, we can

also represent the disjunction (p or $\neg$ p). We can represent that John believes either p or $\neg$ p as:

```
OR (p True) (p False) set1.
```

This belief does not represent a split into two distinct possible worlds (one having p True and one having p False). Instead, it represents that in the current world, the agent believes that one or the other is true but does not know which. This does not give any information about which of the two is a true proposition, which is logically correct.

There are, however, deductions that can be made using this belief. For example, if John believes that p implies q and that $\neg$ p implies r, then the belief (p or $\neg$ p) leads to the conclusion that (q or r). This would be represented by the rule (this rule is specific to the example but could be implemented with variables):

```
OR (p True) (p False) set1
CAUSES (p True) (q True) set1
CAUSES (p False) (r True) set1 ->
    OR (q True) (r True) set1
```

The system can also support rules that will assert which part of a disjunction must be true if the other part is determined to be false. For example:

```
OR (*p *value1) (*q *value2) *set
    NEGATE *value1 *value3
    *p *value3 *set -> *q *value2 *set
```

where NEGATE is a function that negates the flag of a proposition. Note that this rule does not prohibit both parts of the disjunction from

being true, i.e., it is an inclusive OR. An exclusive OR can be created by a consistency constraint that prohibits both parts of the disjunction from being believed at the same time.

5.2 Unknownness

Having decided to represent explicitly that a belief is positive or negative by using true or false flags, the problem of unknown information is easily solved. If information about something is contained in the database, regardless of how uncertain the information is, then a statement about that information will appear in the knowledge base with either a true or false flag. If nothing is known about a subject, then there will be nothing stored about it. This is a natural way to represent unknown information; if it is not stored in memory (i.e. represented in the system), then it is not known.

We should point out that there will be no inferrable propositions that have not been inferred; the counterfactual engine will fire rules until all possible deductions have been made. This means that the initial world must contain all propositions inferrable from the initial propositions and rules. If the initial world is too complex to accomplish this manually, the initial facts can be introduced as a list of counterfactuals. The counterfactual engine will assert the desired initial propositions and will deduce any other propositions inferrable from them using the counterfactual rules.

Although it seems logical simply to omit statements about unknown topics, we must consider how the counterfactual engine works to

determine if this method is practical. There may be times when we wish to be aware that we are ignorant of some fact. This often occurs when we require information to achieve a goal. For example, we may not have the combination to a safe, but we believe that one exists and are aware that we do not possess it. A more common example involves simply moving from one place to another; we cannot go somewhere if we do not know how to get there.

By representing that information is unknown, we can redirect our goals to determine this information, e.g., ask someone the combination or buy a map of the area we wish to go to. This is where the size of a search space (and the agent's awareness of the size) becomes important. If we are trying to find a ball beneath one of three cups, we can try each one until we choose the correct cup. In the case of the combination, however, we realize (hopefully) that there are far too many possible combinations to try in a reasonable amount of time.

Thus, if we are simply to omit any statements about unknown information, we must have some way to detect that such information is missing. In addition, we need a way to represent that we know that something is unknown. The latter problem can be solved by use of the predicate DON'T_KNOW. This predicate will have the form:

DON'T_KNOW *proposition *belief-set

and can be included in the same belief sets as any other statement. This allows us to represent that agent A believes he does not believe something, but it also allows us to represent that agent A believes that agent B believes that agent B does not have a belief about something.

If John does not know the combination to safel, then we use the statement:

DON'T_KNOW combination safel True set1

We should note that the concept of the combination of safel is treated as an object. This will be discussed in detail in Chapter Six. Since the combination is treated as an object, we can refer to the combination without referring to its value, as in the example above. This means that John can represent whether or not he believes that Bill has the combination even if John does not have the combination himself. The value of the combination can be expressed by a statement like:

combination safel 38-25-36 True set1

which can be thought of as shorthand for the statement:

IS combination safel 38-25-36 True set1

It is important to note that DON'T_KNOW indicates that some agent is aware of his ignorance; this is not the same case as when an agent does not believe that he does not believe something. Thus, we do not assume that an agent always believes that he does not believe things. Instead, he will only be conscious that he does not have some information if he requires it and cannot find it in his memory (or has tried before and remembers that he does not know).

There may also be cases when we wish to represent that some agent is missing information but is not aware of his ignorance. For example, agent A may realize that agent B does not believe his house is on fire. Agent B is away from home and it has never occurred to him that his house is on fire. The fact that agent A is aware of agent B's ignorance should prompt agent A to inform agent B that his house is on

fire.

This ability will also be useful in deception when an agent has secret information that he believes another agent does not have. We will represent that some belief is missing from an agent's beliefs with the predicate MISSING, as in the statement:

MISSING *proposition *belief-set.

Both MISSING and DON'T_KNOW are used to represent unknown information. The difference is that DON'T_KNOW shows that an agent is aware of his own ignorance while MISSING shows that another agent is aware of the agent's ignorance.

The belief that some information is MISSING should not appear in an agent's own beliefs because that would indicate that he is aware of his ignorance; the DON'T_KNOW predicate should be used instead. Thus, a MISSING predicate should only be used with nested beliefs about other agents' beliefs. Of course, not all information that is unknown to agents will be represented with MISSING statements; agents will not be conscious of the fact that information is missing if it is not important or relevant information.

Although we now have a representation, we still need a method for detecting that information is missing. If the system were queried, it would simply fail to match information that was not present, but it would not indicate that the information was unknown.

Fortunately, this complication may be circumvented by rules written at the Prolog level. Since Prolog's negation-as-failure operator succeeds when the goal it operates on fails, it is ideal for indicating when some statement is not present. Consider the statement:

DON'T_KNOW INROOM MAP *room true set1.

The statement above indicates that John (whose beliefs are contained in set1) does not know of any room where that map is located. Thus, the location of the map is unknown. This statement, however, is of little use unless we have some way to place it in the belief set. This can be done using a Prolog rule. Such a rule has the form:

```
((user_fact (0 100) name (DON'T_KNOW INROOM MAP *room true
                                set1) *mc) if
  (not(user_fact (0 100) *name2 (INROOM MAP
                                *room true set1) *modcat))).
```

If we wish some reasoning to take place only if the location of the map is unknown, the statement

DON'T_KNOW INROOM map *room true set1

should appear on the left-hand side of a rule. When the statement is encountered, it will be evaluated as true only if the backtracking of Prolog detects no belief in the knowledge base about the location of the map. Hence, an agent has a way to become aware that he does not know something.

The counterfactual engine is not designed to provide answers to queries, but we can develop an interface to help answer queries. When a query is made, we will assume it uses a QUERY predicate and has the form:

QUERY *proposition *set.

We can then write a counterfactual rule such as:

QUERY *proposition *set

DON'T_KNOW *proposition *set -> print "no belief about that "

When the query is made, the DON'T_KNOW predicate will cause the Prolog statement to backtrack and search for information about the proposition. If no information is found it will respond that that agent does not have any belief about that information. If information is found, then DON'T_KNOW proposition will be false, and the rule will not fire.

If we want the system to learn about what is unknown as we query, we must add another DON'T_KNOW *proposition *set to the right side of the above rule; this will then assert the statement that the information is unknown, and the backtracking will not be needed the next time that query is made.

5.3 Representing Uncertainty

We next focus our attention on the problem of uncertainty. The problem of representing uncertainty is the most complex issue because of its broad scope. No belief can be held with absolute certainty, although many are held with great confidence. For this reason, every belief in the system must have some degree of certainty (or uncertainty) associated with it.

Cohen has developed a system using endorsements that are included with statements [Cohen, 1985]. His endorsements indicate degrees of uncertainty and justification for beliefs, but even his complex combination rules result in lengthy lists of endorsements whose complete meaning is hard to grasp and compare to endorsements for other beliefs. We will attempt to represent and justify uncertainty in a way

that allows simple computations and comparisons.

By using flags, we have presented the way that we distinguish between propositions and their negations. The presence of a proposition or its negation in a possible world indicates that the agent believes that proposition corresponds to the real world; the agent cannot know if it is an incorrect (false) belief that does not correspond to the real world. If the agent knew the belief were false, then that belief would not be part of his beliefs (although the belief that the belief is false might be part of his beliefs). As we pointed out earlier, however, most propositions are not believed true or false absolutely; an agent will have an epistemic attitude (opinion) concerning the proposition that indicates his uncertainty about whether it really corresponds to the real world. We next examine the way we represent and reason with this uncertainty.

5.3.1. Meaning of Opinions

Chapter Three discusses how the counterfactual engine uses opinions about the certainty of beliefs to determine which beliefs to revise; this is the basis of the epistemic thesis. The use of opinions is based on plausibility theory [Rescher, 1976]. Kowalski [Kowalski, 1979] and Rescher both point out that logic can only indicate when there are inconsistencies in beliefs (Rescher refers to this as cognitive dissonance); logic cannot determine why the inconsistency arises or how to resolve it.

Plausibility theory moves "from the realm of formal into that of

"material" considerations" [Rescher, 1976]. Our opinions are based on the confidence we have in the premises and rules used to derive the belief, and they indicate an agent's opinion about the truth values of a proposition. This is precisely what Rescher requires of plausibility theory:

The core of the present conception of plausibility is the notion of the extent of our cognitive inclination towards a proposition - of the extent of its epistemic hold upon us in the light of the credentials represented by sources from which it derives.

Recall that we are using an epistemic theory of counterfactuals.

There is generally some uncertainty associated with any belief that we must be able to represent. An agent may believe something to be false, probably false, possibly false, etc. Similarly, he may have varying degrees of strength in his belief that something is true. We will say then that an agent has an opinion regarding the truth value of a proposition.

Besides having opinions to indicate uncertainty we require some way of justifying the opinion and showing why we have that level of certainty. We accomplish this by storing more than merely a number representing the opinion; we also keep track of the sources of beliefs. Thus, we use the sources of a belief to determine the opinion for it, and we can trace the history of those sources to explain how a particular opinion was derived.

Since we already represent explicitly whether a proposition is true or false, it should be understood that the opinion associated with a statement indicates the agent's opinion about the certainty that the

chosen truth value is correct. Thus, we can use the same range of opinions for positive and negative beliefs. There is no need to use negative values for the opinions about negative beliefs.

If we were to use probabilities, we would say that 1.0 represents that an agent is certain of the truth value; 0 represents that an agent thinks the truth value is wrong. Figure 7a shows how a probability scale might correspond to degrees of certainty. Of course, an agent would not retain this statement because it would not represent a belief if he were certain it was wrong. He may, however, have beliefs about which he is very uncertain of the truth value.

Probabilities are not a good choice for indicating plausibility because they cannot be defined relative to an inconsistent set of beliefs [Rescher, 1976]; although inconsistent worlds are prohibited in a stable state, they do and must occur during the reasoning process to necessitate belief revisions. It is when these transient inconsistent states exist that a decision must be made on how to resolve the inconsistency. Hence, probabilities for the various propositions would at that time need to be defined relative to an inconsistent set, which Rescher pointed out cannot be done.

Rather than use probabilities, we will use a scale of integers to represent opinions. This scale is open-ended so that we have a wider range of opinions. Figure 7b shows such a scale, with the lowest value on the scale representing the strongest opinion.

The numbers on these scales do not represent probabilities; they are meant to indicate a decreasing degree of certainty when moving from low values to higher values. Adjectives may be substituted for these

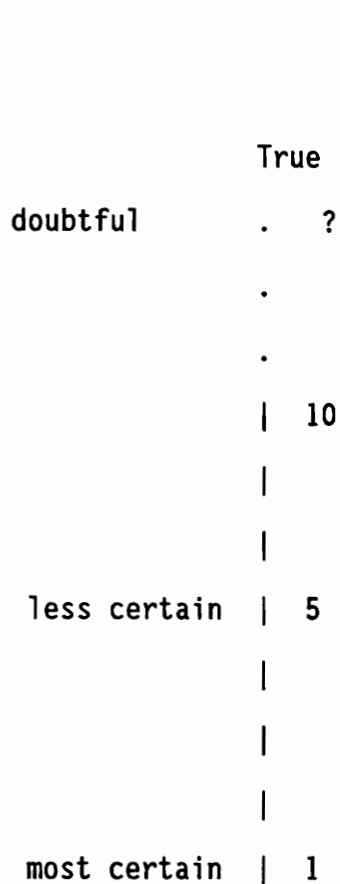
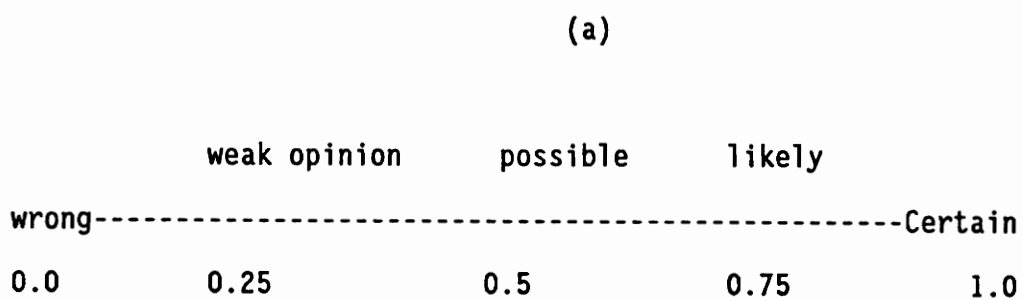


Figure 7. (a) a probability scale for certainty (b) separate scales for true and false beliefs

numbers as shown at the left side of Figure 7b. This seems unnecessary, however, because the relationship between numbers is more concrete and easier to compute.

Regardless of whether numbers or adjectives are used, the point is that an agent adopts an opinion about the certainty he has for a given truth value of a proposition. This opinion, and therefore the level of certainty, will depend on the reasons for being inclined towards a specific truth value. In other words, the certainty depends on why the agent adopts the belief in the first place.

The certainty also varies when beliefs are nested. We can include opinions with nested beliefs because we may have varying degrees of certainty about what we think others think. In fact, weaker opinions for nested beliefs can capture the fact that we are less certain of what others think than we are of what we think; the deeper the nesting of beliefs, the less certain we are and the weaker the opinions should be.

Our treatment of uncertainty is not intended to be merely quantitative, so we will turn our attention to the justification of opinions before we examine how they are computed and used.

5.3.2. Explanations

Rescher discusses several different schemes for assigning plausibility status to beliefs [Rescher, 1976]. In his basic model, status is based on the reliability of the source of information. He places sources in categories such as other agents, common knowledge,

perceptions, reasoning, principles of uniformity, etc. Beliefs are assigned plausibilities based on the category of their source.

Another schema he presents involves categorizing groups of axiomatically stipulated theses. For example, beliefs may be graded according to whether they are definitions, fundamental principles, well-supported theses, or hypotheses and conjectures. A third schema that he presents assigns plausibilities based on a hierarchy of modal categories. For example, epistemic modalities may be categorized as certain, reasonably sure, somewhat sure, or uncertain.

Our use of opinions is a combination of these three schema. The opinion levels themselves are numbers that indicate epistemic modalities; recall that we use integers to simplify comparisons of rank and to increase the range of opinions that we can assign. The modalities are determined by considering the source of the belief. In the case where the belief is derived, the opinions of the supporting beliefs and the type of rules used ("axiomatic contention") are considered to determine the opinion.

We must be able to explain how this modality was determined. Along with an opinion for each belief we keep track of the type of rule from which it was derived. Currently, the counterfactual engine maintains a list of which beliefs were used with the rule to derive the new belief. The justification for reasoning in a particular manner will then be formed from the opinion for a belief, the rule and its type from which it was derived, and the supporting beliefs along with their opinions and origin. Comments may also appear in certain places to add information about where second-hand information came from or qualify a

rule or its exceptions. These comments are not needed during the reasoning process because the information in them is encoded into the numerical level of the opinions; they are, however, included in the table which shows the support for a belief to assist in indicating how the opinion (modal category) was arrived at.

When a desired goal has been concluded, the hierarchical support for it can be recovered. Figure 8 shows what such a support table might look like. Martins and Shapiro distinguish between Justification-based systems and assumption-based systems [Martins and Shapiro, 1988]. Justification-based systems store the direct support for a belief while assumption-based systems store the base assumptions (those that are not derived) that support a belief. Our explanation table is a combination; it stores the direct supports, but it also traces these back to non-derived information (e.g., perceptions, common knowledge, initial beliefs, etc.). This enables the system operator to see how the opinion for a belief was calculated and also to see how and when the beliefs/facts that led to the belief were introduced into the agent's set of beliefs.

There are a few other requirements to be mentioned. In many cases, derived beliefs will not be derived from direct inputs or initial facts; they will be derived through multiple layers of derivations, and supports for each stage of deduction must be combined and propagated. Also, the possibility of exceptions to the rules must be considered.

It may often be the case that a belief that is known is rederived from new information or an alternative rule. This is simply discarded in the current counterfactual engine because it causes no

```

"belief": from physical law R1: 3
  support:"belief": from causal relation R2: 3
    support:"belief": perception: 2:"comment"
      "belief": basic assumption (initial world:
        1: "comment"
      "belief": from physical law R3: 3
        support:"belief": perception: 2:"comment"
          "belief": report from other
            agent: 4:"comment"
    "belief": from correlation: 5
      support:"belief": common fact: 0 "comment"
  "belief": from physical law R1

```

Figure 8. Output Table Showing Explanation
for a Derived Belief

inconsistency. If we wish to give the strongest justification for our conclusions, however, we must compare the uncertainty of the separate derivations.

If we have previously concluded some belief and later find we can conclude it with greater certainty, we should adopt the stronger set of support for the belief. This, in turn, may affect the certainty of other results that were previously derived based on the former opinion and support of the belief. The counterfactual engine makes propagation of these changes simple.

When a new opinion and support for a belief is adopted, the counterfactual engine must search for beliefs whose support includes the belief whose opinion has been modified. If such beliefs exist then their opinions must be recomputed to reflect the stronger opinion of one of their supports. Thus, we are assured that all reasoning is done with the most certain version of a belief, and we end up with the strongest justification for our final conclusion.

We must consider how to handle exceptions to rules. If there are specific exceptions, these may be incorporated into the writing of a rule. A familiar example is that of flying birds. If we know that *x* is a bird, then we know that *x* flies, unless *x* is a bird that does not fly (e.g., a penguin). This could be captured by a pair of rules such as:

```
PENGUIN *x True *set1 -> not(FLIES *x True *set1)
```

```
BIRD *x True *set1 -> FLIES *x True *set1
```

If *x* is a bird then we assume that *x* can fly without assuming that it is not a penguin. When *x* is a penguin, however, we assert that it does

not fly so that a contradiction will result if we say that it flies. The second rule does not care if x is a penguin if we do not have any information to cause us to believe x is a penguin. In other words, we assume that a bird can fly unless we are explicitly aware that a type of bird is an exception. This prevents us from having to specify that x is not a penguin whenever we refer to a bird x.

In general, there may be too many exceptions to a rule to allow us to use this technique. The category to which a rule belongs and the opinion assigned to it will implicitly capture the number of exceptions because a rule with many exceptions (e.g. a correlation) will be less certain and will hold less often than a rule with few exceptions (e.g. a physical law).

Thus, this technique will allow reasoning to take place based on opinions that represent uncertainty and that are invisible to the user during reasoning. Once a conclusion has been reached, however, the user may retrieve a detailed table showing the origin and justification for each decision made during reasoning.

The approach presented here is intended to simulate the way people weigh evidence in their minds when reaching conclusions. We assume that a person's opinion concerning the certainty of a belief is related to the certainty of evidence supporting it and the confidence in the rule used to reach the conclusion. When inconsistencies occur, the first statements rejected are those believed least. This system will enable us to use the built-in capabilities of the counterfactual engine to simulate our model of how people reason. The method of combining opinions will be explained in section 5.3.4. when implementation is

discussed.

5.3.3. Uses of the Representation

We have described the basis for determining opinions and the way that we record sources to explain opinions. Now we will examine how we use those opinions.

In general, the various opinions or levels of certainty give us a basis for discarding or accepting derived conclusions during the reasoning process. Hence, there is a strong similarity to the modal categories that the counterfactual engine currently uses. We can use opinions and the explanation tables associated with them to explain why particular beliefs are retained or retracted. Thus, it makes sense to use the opinions as the modal categories that the counterfactual engine can operate on directly. Before discussing the implementation, however, we will discuss how uncertainty fits in with our representation of different types of beliefs.

5.3.3.1. Combining Uncertainty With Other Belief States

We now have a representation for uncertainty that is distinct from the representation of whether a proposition or its negation (or neither) is believed. The use of opinions allows us to combine uncertainty with the representations of positive and negative beliefs. This is essential because, as we indicated earlier, an agent has some uncertainty associated with each of his beliefs. We will first look at

the combination of uncertainty with negative beliefs.

The ability to represent negative beliefs explicitly is necessary in order to represent many of the beliefs we naturally hold. When this is combined with the ability to attribute different opinions about beliefs, the result is a powerful and useful representation. Returning to our example of searching for a map, we may now plan an efficient search. The combination of negative beliefs and uncertainty enable us to plan the most efficient search based on what we believe about the world. We realize, however, that this is a heuristic approach to planning a search. We only search in rooms where we think the map is not located after we have searched elsewhere. Thus, if we have a belief of the form:

INROOM map *room False set1

then we will search the other rooms before searching in *room.

If there are several rooms that we do not think the map is in, we can be most efficient by searching first in the room that we have the weakest opinion concerning. In other words, if we must search rooms that we thought the map was not in, we should first search the room of which we were the least certain that the map was not in it.

Positive beliefs have the same structure as negative beliefs, and opinions are associated with both in the same way. Since positive and negative beliefs have the same structure, the use of uncertainties has a common effect on both; uncertainties allow an epistemic attitude concerning the agent's beliefs. The opinion associated with a statement indicates how an agent believes the embedded proposition corresponds to the real world (i.e., it is a measure of the agent's

confidence that his beliefs are correct). The flag indicating whether a proposition is negated or not is separated from the representation of uncertainty; this enables us to make direct comparisons between the strength of positive and negative beliefs, since each is represented with the same system of modal categories.

The only requirement that we have for combining uncertainty with belief status is that a belief status be expressed positively; an agent will believe something to be 'True' rather than 'not False'. This is not to say that a truth value is assumed for propositions that are believed to be neither true nor false (i.e., those that are unknown). We are merely stipulating that when an agent has an opinion about a truth value it will be expressed positively. Since False is merely a flag to indicate negation, 'not False' contains the double negation 'not not True'; we will cancel such double negations and simply use True to indicate that the proposition is positive.

We can show that flags that are expressed with a not prefix can be rewritten without it. If John believes that something is not positive then he believes it to be negative (i.e., it has a False flag); the definition of the False flag is that the agent believes the negation of the proposition. He adopts the opinion that he believes the negation of the proposition instead of the opinion that he believes the proposition itself is not true. This may be an incorrect belief if his reason for believing it not true is invalid, but the counterfactual engine permits revision of this belief if a mistake is later discovered. It is important to note again that believing some proposition is not true is different from not believing that it is

true. In the latter case, we cannot assume that an agent will believe the negation of the proposition because the proposition may be completely unknown to him in this case.

Similarly, if John believes something is not negated, then he must believe the positive proposition. There are only two truth values, so there are four possibilities: a proposition may be believed true, believed false, believed to be both, or believed to be neither. We have prohibited propositions from being both true and false. If it is not believed true, it must be believed false or neither (unknown). If a proposition is believed not to be true, however, then the agent has indicated that he has some opinion about the truth of the proposition; it cannot then be unknown, so it must be believed false. Hence, to believe something is not true is equivalent to believing that it is false (as we would intuitively believe should be the case). Likewise, to believe something is not false is equivalent to believing that it is true. Although the two are equivalent, we require that either the flag True be used or the flag False be used.

Figure 9 shows how all possible statements for one level of nesting can be expressed as active, positive beliefs that a proposition is true or false. The different forms will differ in the opinion they are given when translated into our desired format. We have no strict rule for assigning these opinions, but we can make some assertions about their relative strengths based on our intuition.

Our use of flags to distinguish propositions from their negations and our use of opinions to represent uncertainty is not only adequate but also powerful. Since we use opinions as our modal categories, they

I believe not True
should be

I believe False

I believe not False
should be

I believe True

I believe you believe not False
should be

I believe you believe True

I believe you believe not True
should be

I believe you believe False

I believe you believe I believe not True
should be

I believe you believe I believe False

I believe you believe I believe not False
should be

I believe you believe I believe True

Figure 9. Representation of Nested Negations as
a Single Active Belief

are not represented explicitly in our statements. Instead, the counterfactual engine keeps track of them internally and uses them to make decisions. However, opinions alone will not suffice for our purposes because we wish to justify our opinions. We have referred to the reasons or basis that an agent has for adopting a specific truth value with some accompanying opinion. To justify the truth value and opinion selected, we must have some way of representing and propagating these justifications in the reasoning process.

5.3.4. Implementation

We turn now to the problem of implementing a system of opinions and justifications for them in the counterfactual engine. As mentioned previously, we will retain integers to be used as opinions to indicate uncertainty. We chose to avoid fractions because the counterfactual engine uses integers to specify modal categories. If possible, we wish to utilize the current capabilities without major changes.

During the reasoning process, beliefs with a higher modal category are rejected first when an inconsistency must be resolved. When reasoning with uncertain beliefs, we also wish to reject first those with the highest number (or least degree of certainty). The main difference between opinions and modal categories is simply how they are derived.

In the current counterfactual engine, a new fact is assigned a default modal category depending on the type of fact it is. For example, if

INROOM map room1 true set1

is derived, it will be assigned the modal category of other facts of the form

INROOM *object *place *value *set.

The opinions that we will use as modal categories for derived beliefs will depend on the opinions of beliefs used to derive the result, the certainty of the rule used to derive the result, and the level of nesting at which the result was derived. Thus, the assignment of modal categories will need to be modified to use opinions as the modal categories. We ability to dynamically assign modal categories has not yet been implemented; default modal categories are used in the current system and these are used to determine how beliefs should be revised.

5.3.4.1. Combination of Opinions

We must decide how to combine opinions from several beliefs that occur on the left hand side of a rule to assign an opinion for the belief on the right side. In general, the certainty we have in a belief will be related to the origin of the information. We can categorize beliefs into several categories. From most to least certain, beliefs can be categorized as: based on direct observation, based on second-hand information, deduced from physical laws, deduced from causal relationships, or deduced from a correlation.

It is clear that in most cases, we are fairly certain of things we observe ourselves. When information is related to us by others, we must consider the reliability of the source (e.g., if we think someone

is deceiving us, we will not be very certain about the information they give us). The other categories arise because there is some degree of confidence associated with the rules we use for deductions. In general, the more exceptions there are to a rule, the less certain we are that it holds for a particular case.

It appears, then, that we must have opinions concerning the certainty of rules we use. Once again, there is a similarity to the counterfactual engine since it allows modal categories to be assigned to rules.

Rules that capture physical laws, such as allowing an object to be in only one place at a time, will generally have an opinion (modal category) of 1. Causal relationships, such as "if my skin is punctured, then I will bleed," will have an opinion of 2. Correlations, such as "if a person is sick, then he will have a fever," are the least certain and will have an opinion of 3. Physical laws often have a causal nature, but they are considered laws rather than relations because of repeated experimental or mathematical confirmation over an extended period of time.

We must still specify how several opinions are combined. There are several ways that we can do this. We could simply select the weakest opinion among those to be combined. Alternatively, we could use a straight average or a weighted average. The method we use to combine opinions will depend on the type of rule that fires.

There is no way to determine precisely how humans combine and weigh various uncertainties when making deductions. We base our combination rules on epistemic considerations, but we do not claim that

they are necessarily the best way. It makes sense, however, that the type of deduction or rule we are using will affect the way supports are combined.

For example, a physical law is a very strict rule that supposedly will never be violated. It seems that the conclusion drawn from such a rule can never be more certain than the least certain of its supports, and we assume that all of the supports are necessary for the conclusion. The rule itself, however, should not add to the uncertainty of the conclusion since the rule is solidly believed. Thus, for the case of physical laws (rules with an opinion of 1) we will assign to the conclusion an opinion about its truth equal to the least certain (highest modal category) of its supporting beliefs.

This convention conforms to Rescher's general rule that "the plausibility-ranking of a plausible thesis that is derivable from some group of mutually consistent theses is never to be less than that of the least plausible thesis operative in the derivation" [Rescher, 1976]. We agree that this is the case for laws with the lowest modal categories, but we not use the same procedure when the rule itself may be questionable.

The convention also conforms to Theophrastus' Rule, which says [Rescher, 1976]:

The modality of the conclusion (of a valid argument all of whose premises are essential to drawing this conclusion) must follow that of the weakest premis.

Rescher modifies Theophrastus' rule to say "cannot be weaker than that of the weakest premis (but could possibly be stronger)" [Rescher, 1976]. For physical laws we use Rescher's modification, but for other

types of rules our system allows a conclusion to be weaker than the weakest premis because we use the uncertainty of the rule in determining new opinions (Rescher and Theophrastus merely include the rule in the set of premises, but they do not relate its modality to that of the conclusion). This does not directly contradict Rescher's work, however, because we do not assume that all the left side conditions are *essential* to the conclusion of some rules.

We can say this because our causal relations are meant to be sufficient but not necessary (we discuss the causal theory thoroughly in Chapter Eight). When we are using rules that represent either causal relations or correlations, we are generally drawing a conclusion about what we think should be the case given a set of circumstances. These supporting beliefs may not all be required to make the conclusion of the rule possible; some of them may simply add support to make the conclusion more believable.

We cannot be sure of how the various uncertainties are combined, but it makes sense to assume that an agent will be more certain of a conclusion that has a strong explanation; a strong explanation requires that the supporting beliefs have low modal categories (strong opinions). If most of the supporting circumstances are fairly certain and have low modal categories, then we should be confident about the conclusion. On the other hand, if the supporting beliefs have high modal categories (weak opinions), the explanation is shaky and we cannot be confident about the conclusion.

Since the supporting beliefs are a set of sufficient conditions (as opposed to necessary), we may be uncertain about some of these

beliefs without forcing ourselves to reject the conclusion entirely. The certainty of the conclusion should be comparable to the certainty of most of the supporting beliefs. If we average the opinions of the supports, we are effectively averaging the explanations for the supports to determine the strength of explanation for the conclusion. Hence, an average of their opinions will be used. This average will be rounded to an integer since modal categories must be integers. We will discuss below the additional factors that affect the explanation of the conclusion.

The justification for using an average is based on the realization that the supporting statements are sufficient but need not be necessary; we can make them necessary by assigning a modal category to a rule that indicates that it is a physical law. This is why physical laws are treated differently; for a physical law we assume that all of the antecedent conditions must be met. Thus, we admit that a causal relation may hold even if not all antecedent conditions are met, but we wish the opinion of our conclusion to reflect the opinion of the majority of the antecedent conditions; this is why we use an average. By no means do we claim that this is the best or the only method that could be used for combining opinions.

To this average opinion we must add some other factors. The opinion of the rule will affect the opinion of the right hand side. As we noted earlier, a physical law will be considered certain and will add nothing to the average. A causal relation will increase the opinion of the right hand side by 1 and a correlation will increase it by 2. This illustrates how the source and rules used to derive a

belief affect its opinion; the combination of the supporting factors leads to a numerical opinion that can be used as the modal category of the belief. By modifying the opinion of a conclusion based on the modal category of the rule, we will usually violate Rescher's rule; this does not bother us because we believe the opinion of a conclusion should reflect the uncertainty of both the supports and the rule.

We should also increase the opinion when we are attempting to predict beliefs of other agents based on our own observations and beliefs. This is because we are less certain of what we think others believe than what we believe ourselves. To account for this we will add one to the opinion for each level of nesting. Consider the following case as an example. John believes Bill is in the kitchen. John also believes that the cat is in the kitchen. John assumes that Bill can see the cat so he believes that Bill believes that the cat is in the kitchen. Since this is only a prediction John can not be as certain of his conclusion; it may be that Bill has his back turned to the cat.

The level of nesting will not affect conclusions made when an agent attempts to predict what beliefs another agent makes while reasoning. We assume in this case that the second agent will use the same rules for combining opinions that would be used by the first agent.

Thus, we have included several factors in the way we determine the opinion or certainty of a newly derived belief. This opinion is used just as a modal category would be in determining the order in which to discard beliefs when resolving inconsistencies. Like modal categories,

opinions are transparent to the user during the reasoning process. There are different ways to combine opinions based on the type of reasoning involved; this makes intuitive sense because humans can change the way they reason depending on the situation.

5.4. Conclusions

The schema we have adopted to represent uncertainty involves a combination of a level of certainty and background information on why that certainty was chosen. The opinions we use are numerically based like the belief functions of [Dubois and Prade, 1988]. Like belief functions, our opinions represent a level of uncertainty and are combined using special combination rules. Unlike belief functions, however, we include with an opinion the justification for that opinion. Although our justifications are not as rich as Cohen's system of endorsements, they are easier to reason with in the system. Also, we do not have to deal with lengthy lists of endorsements that may appear to contain contradictions; a given belief may have "maybe too general" and "maybe too specific" as endorsements. Our justifications are essentially a history of the support for a belief. We chose this approach because humans seem to base the certainty of their inferences on the certainty of the beliefs and rules used to derive them.

6. Theoretical Basis of Our System

Now that we have a representation for beliefs in our system, we need to understand what we are representing. A thorough understanding of the system is necessary to study problems such as substitution of equalities and detection of deception. Barwise and Perry argue against possible worlds approaches [Barwise and Perry, 1983], but our use of possible worlds is a method for collecting possible situations.

We will first describe our ontology. The ontology is similar to that of Barwise and Perry [Barwise and Perry, 1983] except that our abstract situations are defined differently, and we extend the ontology with various types of statements. We will then use this ontology to describe how our system uses possible worlds, and we will compare our use of possible worlds to Cresswell's use of possible worlds and Barwise's use of situations. Next, we will identify the theory of truth we are using and will define our ontology. We conclude this chapter by discussing our representation of concepts that do not map to the real world.

6.1. Ontology

We have discussed our system in great detail, but we have not thoroughly described our ontology. This ontology allows us to represent an agent's beliefs about the world and his nested beliefs about other agents' beliefs. A possible world is a set of statements specifying what is believed true at a particular time. Our system

stores these statements in the memory of a machine, but they are intended to represent concepts within a human's mind. This means we must somehow relate these intangible beliefs to the actual world. We will now explain the various entities in our system, where they exist in the world, and how they interact to create beliefs.

Our system is based on the existence of many types of entities in the real world. The interaction of these various structures is essential to the operation of the system. An understanding of the system requires comprehension of the purpose of these structures and the relations between sets of them.

6.1.1. Real world

The real world is the world that surrounds us and includes us. It is composed of physical objects that interact with each other, and we can perceive relations between these objects. It also contains objects, relations, and attributes that exist only in the minds of agents, since those minds are a part of the real world. In other words, we use the term real world to refer to the actual world we live in; it describes the way things really are. We will refer to the set of objects that exist in the real world as O_p . The members of this set of objects are anything that physically exists in the world, such as a cow, a watermelon, a bicycle, a cloud, etc.

The set of relations that exist in the world will be R_p . Relations may have many arguments or only one. We should have a strong intuitive idea of what relations are; they can be described by predicates like

ON, NEXTTO, ABOVE, etc.. We must also be able to include actions and events in our system; we will consider actions to be a form of relation between an individual and objects or the world (e.g. IS_WALKING John).

Objects may also have attributes, such as big, good, or ugly. The set of attributes that real objects may have will be referred to as A_p .

If we describe the world in terms of these objects, relations, and attributes, then we are enumerating facts about the world. Thus, a fact is some combination of objects, relations, and attributes that reflects some information about those objects in the real world. Facts are themselves described by natural language statements; these statements may contain language that refers to an object in a specific manner or merely assures that correct grammar is used in the statement. Ordinary discourse generally uses natural language statements, but our system operates at a more basic level using propositions. These natural language statements should not be confused with the statements we use in our system.

Reasoning in our system takes place with concepts that are usually intended to map to the real world; we should note, however, that we will also use concepts that do not map to the real world. An example is a unicorn, which we can think about easily although it does not exist physically. These imaginary concepts pose a special problem that we will examine in the latter part of this chapter.

6.1.2. Concepts

Since we are reasoning with beliefs and this reasoning takes place

in an agent's mind, the mental world is more interesting to us than the real world. The primary structures of the mind are concepts.

A concept is a thought that exists within an agent's mind. The way a concept is represented in the mind is unknown; it may be a mental image or a description of the properties of the concept or a combination of both. An agent can understand and reason with concepts; concepts are the primary thoughts that provide a mental representation of the objects, relations, and attributes of the real world. Concepts, therefore, can be classified as object concepts, relation concepts, or attribute concepts; they are used to define classes of objects, relations, or attributes. When a concept is very specific, it may define a class that has only one member.

We will refer to the set of object concepts as O_C . An object concept can map to a physical object in the set O_p . It is not, however, necessary that this be the case. Since we can imagine things such as unicorns, and pixies, it is clear that there are some object concepts that do not map to objects in the real world. Regardless, there is still a definite image or description in our minds that allows us to understand what a unicorn is. Hence, we can reason about object concepts as well as physical objects; this allows us to answer questions about how many horns a three-headed unicorn must have.

The set of relation concepts will be referred to as R_C . These relations will generally map to a relation in the real world (the set R_p). It is possible, though, to develop concepts of relations that do not exist in the real world. At first, these relations might not appear useful since they would not apply to physical objects. The

ability to conceive of relations that do not yet exist in the actual world is very important, however, because it enables humans to be inventive. Such concepts would be particularly hard to grasp without a physical example of the relation to define what it meant; this may be why not all humans have the imagination to be inventors.

It is also possible to imagine relation concepts that could not exist as physical relations but could only relate object concepts to other object concepts. Perhaps these relations would violate some physical law if they existed in the real world, e.g., a rocket travelling faster than the speed of light. Such relations are especially interesting in the context of science fiction.

We refer to the set of attribute concepts as A_C . These are interesting concepts because they can vary from agent to agent; for example, what is beautiful to one may not be beautiful to another. Thus, attribute concepts like beautiful or good are very subjective, while others like big or small are relative. This means that the physical attributes of the set A_p could not be associated with the objects of O_p without the corresponding attribute concept of A_C ; relations, on the other hand, can exist even if an agent does not understand the concept of the relation.

For example, two people may be next to each other in the physical world even if an agent does not understand the relation concept of "next to." On the other hand, a person could not be beautiful unless the agent has an attribute concept of what beauty means.

This is not to say that all attributes are relative or subjective; mass and kinetic energy are properties that are neither. Also, this

does not mean that no relations are subjective; many comparatives, such as 'wiser' or 'more beautiful' are subjective. By reasoning with concepts, we allow agents to use subjective concepts in reasoning, but we assume that many concepts have standard definitions that are common to all agents.

For example, two agents may not agree on classifying a given object as 'large', but they would probably agree on which of two objects is 'larger'. Thus, relative size is a rigid concept just as the labels referring to the concept may be rigid designators (a rigid designator is a label that everyone agrees on for an object). In fact, agents must have rigid concepts to be able to communicate information; they may disagree on whether an object can be classified by a given concept, but they will (hopefully) both agree on the meaning of the concept. If they are using different concepts, it becomes moot whether they agree that the object can be classified by that concept. Humans sometimes misunderstand each other when they use the same designator but mean different concepts. This usually occurs when one of the agents has misunderstood the commonly accepted meaning of the concept, i.e., he has failed to grasp the rigid concept.

Since concepts are taken as the primary mental structures, it is difficult to define them other than through their relation to actual objects, relations, and attributes. Although we may not be able to define the concept precisely, we can reason with it to determine that Mary is beautiful or that Fred is a unicorn.

Concepts exist as mental thoughts that may actually be impossible to define completely in words. We encounter this difficulty when we

try to understand what makes an object correspond to a particular object concept. As an example, consider what makes a bicycle a bicycle. We are generally able to classify a bicycle as such if we see one [Lakoff, 1988], yet we cannot easily define the concept of being a bicycle. A bicycle might be described as having two wheels, pedals, a chain, handlebars, a seat, a frame, etc. If a wheel is removed, however, we do not consider the object to be a unicycle; it is a bicycle with a missing wheel. If we remove the seat and handlebars, it is no longer functional as a bicycle, but we could still identify it as such. It seems, then, that humans develop some intuition that helps them classify objects, relations, and attributes in the real world using certain concepts.

The ability to manipulate concepts allows us to use our imagination. For example, we know what a unicorn is even though we may not believe that one exists in the world. If we saw a unicorn we could identify it, even though we have never seen one before. We can also classify pictures that represent unicorns and can state how many horns a unicorn has. This reasoning may deal only with mental structures or concepts and may not involve the real world at all.

It is also important to remember that the label we use for a concept is not identical with the concept. Labels are merely strings of characters that we use to represent objects and concepts; they are a shorthand representation we use to avoid lengthy descriptions of what we are referring to. In fact, we have already pointed out that concepts may be impossible to describe completely with linguistics.

6.1.3. Propositions

A proposition is an assertion concerning a relationship between objects or attributes of an object. In other words, a proposition describes a situation in the real world. Propositions take on truth values of true or false. The value will be true if the relation asserted by the proposition corresponds to the actual relation between the objects referred to. Thus, a true proposition is also a fact.

Since the real world contains mental entities, a proposition can also concern objects that do not exist physically, such as the number of horns a unicorn has. This means that an agent's beliefs about another agent's beliefs are really propositions about that other agent's beliefs.

Since the world may change, there must be time associated with the truth value of a proposition to specify when it is valid. We will refer to a proposition as $p_i(t)$. Since a proposition refers to a situation (a relation between objects and/or attributes), we can refer to the proposition independently of its actual truth value. Thus, we may use a proposition to describe a situation without caring whether that situation is true or false of the real world.

6.1.3.1. Abstract situations

An abstract situation is a mental representation of a situation in the real world. We think of abstract situations as images because a picture can represent the details of a situation efficiently; we do not

claim to have proof that humans represent information in this manner. An agent may generate two abstract situations for a proposition; one corresponds to the proposition and the other to its negation. An agent will have no abstract situations for propositions that he has not thought about. Abstract situations, therefore, present a way for an agent to think about both a proposition and its negation simultaneously; the agent can represent both but does not believe that both correspond to the real world. In effect, the agent is considering the possible truth conditions that the proposition could have on different interpretations (i.e., in different possible worlds). Since these are mental representations of the propositions, there is no contradiction created by an agent thinking about both.

The agent must select one of these abstract situations as the one he believes corresponds to the real world. When he does so, that abstract situation becomes part of his beliefs, and the proposition corresponding to the abstract situation is added to the possible world representing the agent's beliefs. An agent's beliefs, then, consist of a set of abstract situations that correspond to the propositions he believes are true in the world.

A set of abstract situations would appear to be a set merely consisting of $\{p, \neg p\}$. The important distinction is that propositions are about the world independent of mind while abstract situations are an individual agent's mental representations of those propositions.

We will represent an abstract situation as $AS_i(t)$. Abstract situations are considered to be mental images in our theory; this is why we use statements to represent them in a machine. An example would

be a cat concept and a mat concept related by an ON concept. This abstract situation would correspond to the proposition: ON cat mat True. Another abstract situation would correspond to the negation of the proposition:

ON cat mat False.

If the agent has a belief about this proposition or its negation, then the appropriate abstract situation will be part of his beliefs.

We will think of abstract situations as mental images. Since we do not represent images directly in our system, we use statements that correspond to abstract situations. These statements will be described below.

6.1.3.2. Beliefs

A belief is an abstract situation that an agent has accepted as corresponding to the world. This abstract situation corresponds to a proposition about the world or its negation. By admitting an abstract situation into his beliefs, an agent indicates that he believes that corresponding proposition (or its negation) is true in the real world. Thus, the proposition is true on the interpretation of the agent's beliefs.

Of course, the agent's beliefs have no effect on the actual truth value for the proposition. The truth value that an agent maintains for a proposition need not be the correct value. That is, an agent may have incorrect beliefs; his belief about the truth value of a proposition may not correspond to the actual truth value of the

proposition. In addition, an agent's beliefs may change, so there must be time associated with the belief to indicate when an agent holds that belief. We can represent a belief as $B(i,t)$ where i indicates the agent (or nested belief set) and t indicates time.

If an agent thinks the abstract situation given in the previous section (ON cat mat True) corresponds to the real world, then it will be asserted within a statement in the possible world corresponding to the agent's beliefs. The time associated with the belief is implied by the timestamp of the possible world that contains the belief as a statement.

6.1.3.3. Goals

A goal is a situation that an agent wants to exist in the physical world. As with beliefs, a situation in the real world is described by an abstract situation. The difference is that a belief describes what the agent thinks is true of the world while a goal describes what the agent wants to be true. It is possible that an agent's goal is already true in the world and he may not have noticed. If the agent determines that his goal is not true, he will act on the world in an attempt to achieve his goal, i.e., make the situation described by the goal correspond to the real world. In other words, an agent believes he has achieved a goal when the goal corresponds to a belief.

6.1.3.4. Communication

Communication occurs when one agent transmits information to another agent. This information will be a concept proposition when an agent is communicating a belief (we refer to this as a report); the information will be a goal if the agent is communicating a request to another agent (we refer to this as a query). Note that we are not dealing with communication at a natural language; we are communicating ontological entities directly without interpretation.

6.1.4. Statements

A statement maps a proposition (or its negation) into a belief, a goal, or a communication. The structure of these entities differs so that they may be distinguished.

A belief is represented by a statement that consists of a proposition and a set label specifying membership in a particular belief set. Since an abstract situation corresponding to the belief occurs in the mind of an individual agent, the set label is necessary to indicate which agent the belief belongs to. Thus, statements explicitly indicate the interpretations on which a given proposition is true (relative to an agent's beliefs). We use statements to represent the abstract situations that make up an agent's beliefs because our system cannot represent images.

Since propositions change their truth value over time, and since beliefs also change, statements must also have time associated with them. We can represent a statement as $s_i(t)$ to indicate that statement s_i is true at time t on the interpretation (belief set) indicated by

the statement's set label.

If set1 represents John's beliefs and John believes the cat is on the mat, then we would use the statement:

ON cat mat True set1

This indicates that John believes that the abstract situation in which the cat is on the mat corresponds to the real world.

A goal is identified by the predicate GOAL and consists of the agent who is the subject of the goal, the goal (in the form of a proposition or its negation), and a set label to identify which belief set contains the goal (i.e., whose goal the statement describes). The explicit representation of whose goal it is along with the set label means that one agent can think about what another agent believes about a third agent's goals.

An example of a goal is:

GOAL JOHN HOLDING BOX True set1

which says that John's goal is for John to be holding the box.

A communication is identified by the predicate INPUT along with a label of REPORT or QUERY to distinguish between the two. This is followed by a proposition that can represent a belief or a goal depending on whether the communication is a report or query. Finally, the speaker and listener are identified. Listener refers to the person to whom the communication is directed; there may be other agents present who can hear the communication.

An example of a report is:

INPUT REPORT INROOM ROOM1 BOX True JOHN BILL

which says that John is telling Bill that the box is in room1.

An example of a query is:

INPUT QUERY COMBINATION SAFE1 JOHN BILL

which means that John is asking Bill for the combination to safe1. Since propositions change their truth value in time, and since beliefs and goals also change, statements must also have time associated with them. We can represent a statement as $s_i(t)$ to indicate that statement s_i is true at time t .

6.1.5. (Partial) Possible Worlds

A possible world is a set of all an agent's beliefs at a particular point in time. Thus, there is a time associated with every possible world, and the world is defined by the collection of statements that represent the agent's beliefs at that time. A possible world $W(t)$ is then defined as:

$$W(t) = \{s_i(t)\}.$$

A general schematic showing the relation among our structures is shown in Figure 10. Note that there are two distinct paths shown for Figure 10 that map a situation into a belief. One way is via a statement, and the other is directly by way of an abstract situation. Theories for both linguistic and mental image models have been suggested. It is likely that memory contains a combination of images and descriptions.

The mapping in Figure 10 via abstract situations is similar to the idea behind situation semantics [Barwise and Perry, 1983] except that our definition of abstract situation is different. In either case,

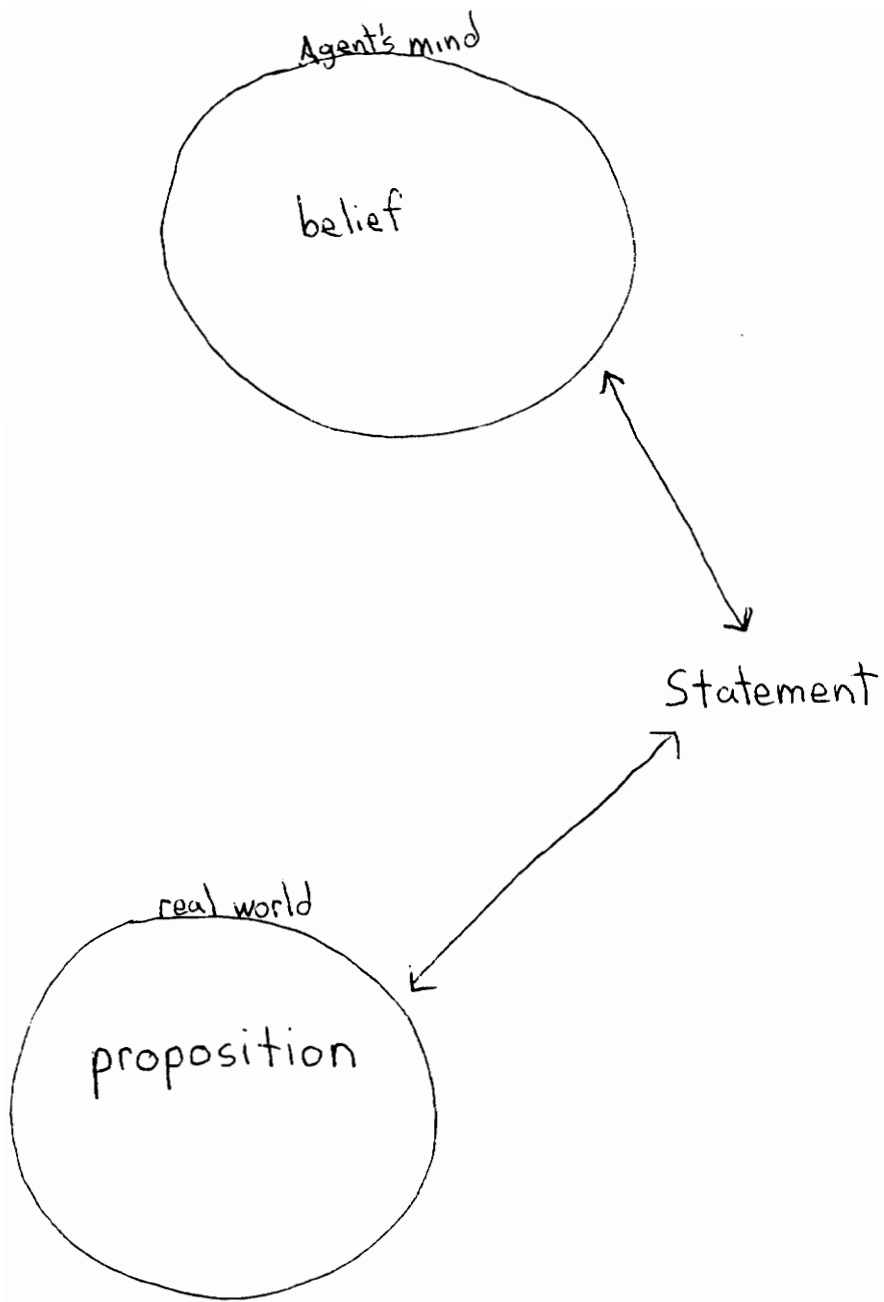


Figure 10. Mapping from world to beliefs via statements.

abstract situations are thought of as images; Johnson-Laird [Johnson-Laird, 1983] provides support that memory utilizing images is used by humans. To map abstract situations directly into beliefs, our system would need a memory capable of representing and manipulating images; we have not developed this capability. Our system instead uses a linguistic approach by mapping via our statements; statements are formed from labels that are easy to represent in a machine.

Ideally, a statement should correspond to an image of an abstract situation. The limited structure of statements, however, makes them less detailed than an image could be.

Figure 11 depicts how a situation in the real world is mapped into a belief. There is a physical relation in the real world between a cat and the mat. An agent would identify the objects and relation in the real world and match these with concepts in his mind. He then creates a concept proposition in his mind using the object concepts of cat and mat and the relation concept of ON. The abstract situation in which the concept proposition of ON cat mat is true is then chosen to correspond to the agent's belief that the proposition ON cat mat is true in the actual world. This abstract situation then becomes part of the agent's belief set. Our implementation uses the statement that corresponds to this abstract situation to represent the belief.

In summary ('--->' represents a mapping):

- 1) combinations of physical objects, relations, and attributes map into propositions

$$O_A \& R_A \& A_A \text{ ---> } p_i$$

- 2) combinations of object, relation, and attribute

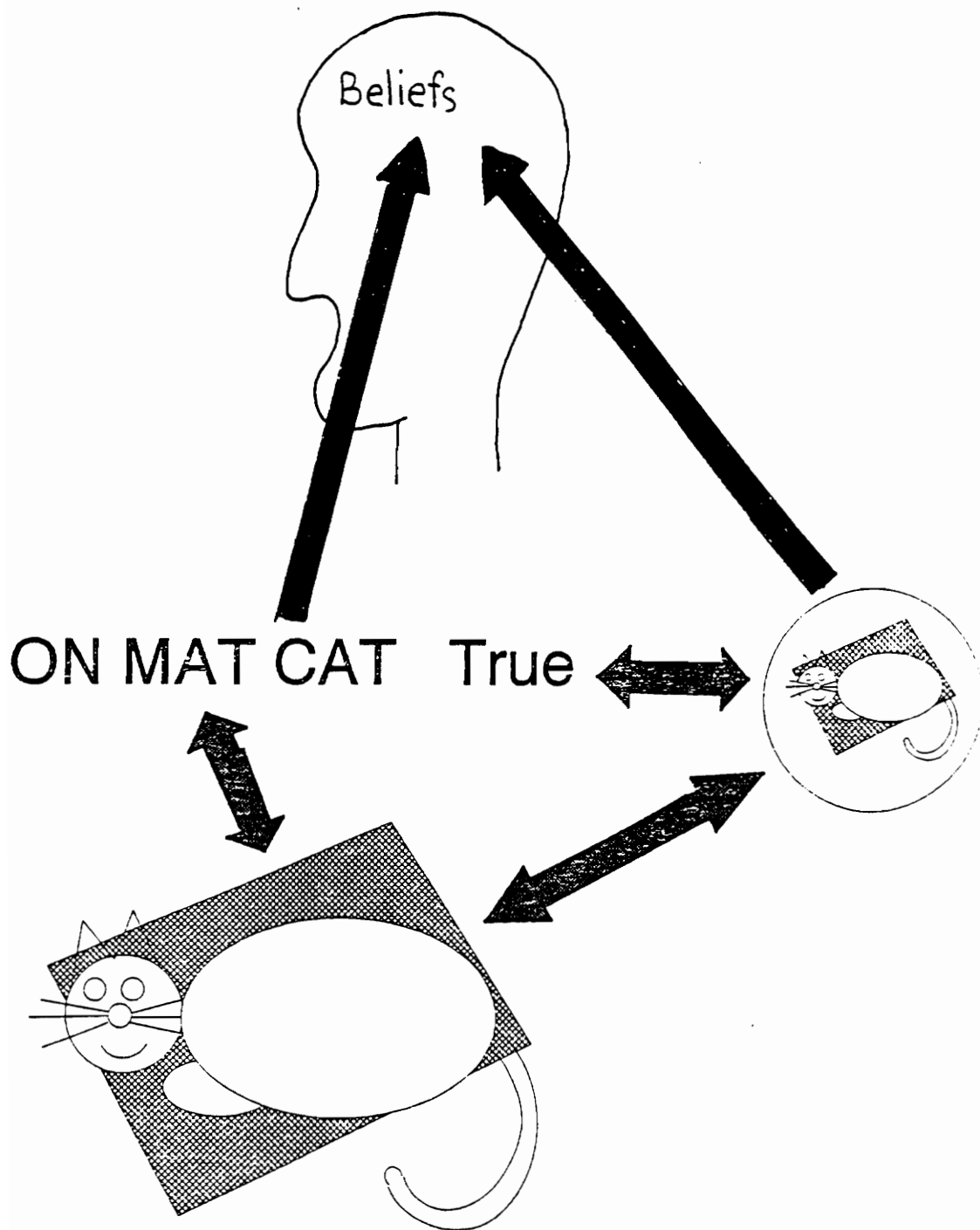


Figure 11. A cat on a mat.

concepts map into abstract situations; these
are mental representations of propositions and their negations

$$O_C \& R_C \& A_C \dashrightarrow AS_i$$

- 3) statements correspond to abstract situations

$$s_i \Leftrightarrow AS_i$$

- 4) statements indicate which abstract situations are
accepted as beliefs.

$$s_i \dashrightarrow B_i$$

- 5) (partial) possible worlds are sets of beliefs that are
represented by statements.

$$W_i = \{s_i\}$$

6.2. Use of Possible Worlds

Our system represents a possible world by a list of statements; these statements indicate the propositions about the world that are believed to be true or believed to be false. If information about a proposition is simply unknown, then there are no statements represented concerning that proposition. Since we do not include statements about everything in the world, our possible worlds are actually partial possible worlds; we realize that there are facts about the world that we do not know about.

6.2.1. Relations Between Statements and Propositions

The distinction between statements and propositions is important

in our system. Propositions represent information about the world while statements represent information that an agent believes. The only representational difference between the two is that statements contain a set label; propositions are embedded within statements, and the set label indicates the belief set to which the proposition belongs. Propositions that are true of the real world have the label 'world' attached to maintain the same number of arguments as statements. Statements that represent nested beliefs are representing propositions about propositions (e.g., the proposition that one agent believes the proposition that another agent believes some proposition). Thus, the set label of a statement indicates the belief set (possible world) that defines the interpretation on which the embedded proposition is true.

The use of set labels means that the same proposition may occur embedded in several statements; for each distinct belief set that contains that proposition there is a separate version of the proposition with a corresponding set label. Although it seems redundant to have multiple copies of a proposition in the system, each is a separate entity and the existence of multiple versions is intuitively correct; this is because each agent has his own version of the concept in his beliefs. Agent's do not share beliefs even though they may believe the same propositions; each has constructed his own abstract situations corresponding to propositions and has accepted some of these as beliefs.

A group of objects, relations and attributes in the real world comprise a situation, including time and location. The concepts that

map to them comprise a proposition, and an agent's mental representation of this proposition (and its negation) are abstract situations. When the abstract situation is believed to model the real situation correctly, the abstract situation (i.e., the proposition it represents) becomes a belief. Since statements are used to represent abstract situations, a set of these statements describes one partial possible world that an agent believes corresponds to a portion of the real world.

We can understand our system better by considering the description of possible worlds given in [Bradley and Swartz, 1981]. According to this description, a possible world is one possible state of affairs for everything that has ever existed (either mentally or physically) as long as it does not contain a contradiction. For example, a world that asserts George Washington to be alive in 1990 and George Washington to be dead in 1990 would be an impossible world.

The real world is one of these possible worlds, but an agent cannot really tell which one because he has only partial information about the physical and mental world around him. Thus, we can construct several partial possible worlds from his beliefs that might correspond to the real world. An agent in our system uses his beliefs and counterfactual reasoning rules to select the possible world that seems most likely to correspond to the real world.

The counterfactual engine allows an agent to revise his beliefs when a mental event occurs (we will discuss our mental event logic in Chapter Eight). Because of uncertainty and incomplete information, however, an agent may revise his beliefs in several possible ways to

account for new information. Thus, he may have several possible belief sets or possible worlds that he may adopt to reflect his revised view of the world; he must have a way of choosing the best new possible world.

Lewis [Lewis, 1986a] discusses the idea of closeness between possible worlds; the closest possible worlds to another are those with the "fewest" differences. The opinions that our system uses as modal categories provide a way to measure closeness. The counterfactual engine generates the new possible world that is closest to the previous one; it requires the fewest changes involving the least certain information, and it thereby retains the most certain information. Thus, the agent's beliefs about how the world is changes as little as possible in response to inconsistencies or new information.

6.2.2. Comparison to Situation and Possible Worlds Semantics

The basic properties of possible worlds semantics and situation semantics are described in [Barwise, 1989]. Barwise then compares the two semantic theories by contrasting these basic assumptions. Our theory is based on a combination of these two theories.

The first difference that Barwise points out is the primitives of the two theories. Possible worlds semantics takes objects and possible worlds as primitives; it uses sets of these to represent everything else. Situation semantics allows objects, relations, and properties to be primitives, as well as events, locations, and situations. Our theory closely resembles this because we take as primitives object

concepts, relation concepts, and attribute concepts, as well as concept propositions and situations.

A major difference is that the primitives we use to build our abstract situations are all concepts. Our abstract situations model very small portions of the world that can be described by single propositions; Barwise implies that situations are intended to describe larger portions of the world. Barwise makes this clear in [Barwise, 1989] so that he can refute Cresswell's claim that situations are conflated with propositions.

Situations are generally more complex than propositions; if they are the same entity referred to by different names, then the use of situations is neither new nor exciting. This is why Barwise emphasizes that 'situation' is not just an alternate term he uses for 'proposition' and that he does not confuse the two. We do not deny that our system may do this, except we should point out that we base our system on concept propositions rather than propositions or facts.

Situation semantics allows only the model of a single world, the real world. Possible worlds allows many different worlds, but distinguishes one as the real world to settle basic facts. Lewis [Lewis, 1986a] claims that possible worlds are alternative realities. We do not make this claim, but our system allows multiple (partial) possible worlds; one of these corresponds to the physical world, although it is often not the one represented by an agent's beliefs.

Possible worlds semantics claims that these other worlds are the other ways that things might have happened. Situation semantics says that legitimate assertions are about portions of the real world. We

combine both ideas in our theory; we build abstract situations that are intended to correspond to the real world; we then use these to create different possible worlds (sets of beliefs) that might correspond to the real world based on our incomplete knowledge.

Thus, our use of possible worlds semantics serves two purposes. It allows agents to create a model from partial information that they believe corresponds to the real world, and it allows agents to reason hypothetically for reasons such as planning. At a given time, an agent has a single set of beliefs that he thinks is his best currently available model of the real world. When his information about the world changes, so does his set of beliefs that are meant to correspond to the real world. The agent may also reason about sets of beliefs that he thinks would be true given some hypothetical fact.

As we stated at the beginning of the chapter, our system provides a method for collecting different possible (abstract) situations into partial possible worlds.

6.2.3. Theory of Truth

We have mentioned several times that an agent's beliefs are intended to correspond to the real world. For completeness, we will expand on this and explain the theory of truth we are using.

[Haack, 1978] gives a thorough discussion of several theories of truth. Two major theories of truth are the correspondence theory, defended by Russell, Wittgenstein, et. al., and the coherence theory, defended by Bradley and Rescher.

Our system is based on a correspondence theory of truth; beliefs are intended to correspond to situations in the actual world. Since agents cannot verify with certainty that all (or perhaps any) of their beliefs correspond to the real world, we must have some other mechanism by which to gauge correspondence.

The counterfactual engine uses a coherence theory of truth when revising beliefs. This makes sense because human beings believe that the real world is consistent; thus, any model of the real world should also be consistent.

The beliefs that can most easily be compared to the actual world through perceptions are given greatest confidence (lowest modal categories). These beliefs can best be tested for true correspondence; beliefs deduced from rules or obtained from questionable sources (lying opponents) end up with higher modal categories. Thus, beliefs are revised to be consistent with those that best correspond to the real world.

We are therefore using correspondence as a definition of truth, but coherence is a criterion for the acceptability of possible worlds representing revised beliefs; coherence is a necessary but not sufficient condition for the possible world to correspond to the real world (i.e., for the possible world to correspond to the real world it must be coherent, but mere coherence does not mean it will be the real world). We should note that this is similar to the pragmatic theory of truth defended by Peirce, Dummett, et. al., which is also a combination of the correspondence and coherence theories.

6.3. Reasoning with Concepts

The use of concepts as independent entities with which we can reason is important for a system that reasons with beliefs. This is true because we often reason with concepts of objects that do not physically exist in the real world or exist at some other time or location; we seldom think about only the things before our eyes or only about the present time. Even though most of our object concepts may map to physical objects, we reason with the concepts themselves so it does not matter if they do not map to physical objects; this allows us to reason about concepts like unicorns.

For example, we believe that George Washington physically existed once upon a time but does not anymore. Yet, we can reason about George Washington because we have a concept of him. Thus, object concepts enable us to reason with objects that are separated from us in space or time, even if they do or have existed physically. Object concepts are more important to our reasoning than physical objects because we do most of our reasoning with them; object concepts are not necessary only for reasoning about imaginary objects, they are needed for reasoning about physical objects as well.

6.4. Imaginary and Fictional Objects

We have allowed our system to contain object concepts that do not necessarily map to physical objects. This enables us to reason about imaginary or fictional objects that exist only in our minds. At the

same time, we must distinguish whether an agent realizes that the object exists only as a mental object or whether he believes it exists in the real world. We refer to concepts that do not map to physical objects, relations, or attributes as imaginary or abstract concepts.

Humans can perform many operations with numbers, such as counting and arithmetic, even though numbers are abstract concepts. It is not surprising, then, that they should also be able to reason about imaginary objects such as unicorns. Just as we have used distinct labels to represent concepts and the objects they denote, we can apply the same scheme to imaginary objects such as unicorns. The difference is that an agent is capable of knowing that unicorns do not exist. Thus, he may be able to think about a unicorn and may say that the subject of a picture is a unicorn, but he would not search for an actual, physical unicorn.

Since reasoning takes place with concepts, this reasoning would not differ significantly for an agent who believes in unicorns from one who does not. The major dilemma occurs when an agent attempts to assign the concept of being a unicorn to a particular physical object. We will consider a physical object referred to as Fred. If the agent believes unicorns actually exist, he can make the assignment without any problems from his point of view.

For example, he might have the belief:

IS_A unicorn Fred True set1

Since unicorns do not exist in the real world, however, his belief that Fred is a unicorn is a mistake; Fred may actually be a goat. Since agents are capable of making mistakes, there is no problem with the

agent having the belief that Fred is a unicorn.

Since the reasoning in our system is done with concepts, there is no difference in the mechanism of reasoning when the concepts used do not map to physical objects. There will be cases, however, when the agent should not perform certain kinds of reasoning with purely imaginary concepts.

Consider the unicorn example above. It is valid for the agent to have wrong perceptions of the world, but if he believes that unicorns are imaginary he should not really believe that the object he sees is a unicorn. Thus, we must have a distinct way of representing concepts that do not map to the real world. We must also be able to control the types of inferences that may be made with these concepts. At the same time, an agent who does not realize that the concept is imaginary should be able to make inferences with it as if it mapped to a physical object.

Our approach to this problem is similar to McCarthy's use of a "denotes" predicate [McCarthy, 1985b]. This predicate explicitly gives the relation between a concept and the object it denotes. We will henceforth adopt the notation for indicating a physical object by enclosing a string representing it in quotation marks. This means that DOG refers to the concept of a dog, while "DOG" refers to a particular physical dog. We can represent the mapping between the two using McCarthy's predicate, denotes:

```
denotes(Lassie, ("dog1"))
```

We will expand on this by including a set label as an argument of the predicate. This allows us to represent that different agents map

concepts to different objects. If John maps the concept DOG to a physical "DOG" we would represent it as:

`denotes(Lassie, ("dog1"), set1)`

or in our representation

`denotes Lassie ("dog1") True set1`

provided that set1 is John's set of beliefs.

We can now represent concepts that do not map to physical objects by the following notation:

`denotes(X,nil,set1)`

where X is a concept of John's that does not map to the real world. An example with a unicorn (an imaginary concept) would look like:

`denotes(unicorn,nil,set1).`

Thus, we can now distinguish between concepts that map to the real world and those that do not. For those that do map, we can represent the object they map to; for those that do not map, we represent that they denote 'nil'. If an agent does not know whether or not the concept maps to a physical object, then we may omit the denotes predicate for that concept.

We can also represent that an agent believes a concept has a mapping when it does not. If John believes that physical unicorns really exist then he would have a denotes predicate concerning 'unicorn' (we use single quotation marks to set a word or phrase off from the rest of the sentence since we are using double quotation marks to indicate physical existence). We could have either of the following statements in the system:

`denotes(unicorn,nil,world)`

`denotes(unicorn,"horsel",set1)`

The same notation can also be used for relation concepts and attribute concepts. These concepts may or may not map to relations and attributes that exist physically. If they do map, we may wish to refer to the relation or attribute itself, rather than the concept that maps to it.

We can describe the implications of a mapping between a concept and the physical object, relation, or attribute it represents using the following schema:

- (i) $\text{denotes}(x, "x", \text{set1}) \ \& \ \text{denotes}(*\text{PREDICATE}, "*\text{PREDICATE}", \text{set1}) \supset$
 $(*\text{PREDICATE} \ x \ \text{True} \ \text{set1} \equiv \text{"*PREDICATE" } "x" \ \text{True} \ \text{set1})$
- (ii) $\text{denotes}(x, \text{nil}, \text{set1}) \supset$
 $(*\text{predicate} \ x \ \text{True} \ \text{set1} \neq \text{"*predicate" } "x" \ \text{True} \ \text{world})$

For concepts that map to objects, relations, or attributes in the real world, (i) applies and for concepts that do not (ii) applies. Statement (i) says that if an agent believes object concepts and predicates (relation and attribute concepts) denote things in the real world, then if the agent believes the predicate holds for the concepts he also believes the physical objects satisfy the predicate in the real world. This is not the case for imaginary concepts because there are no physical objects to satisfy the predicate.

We stated earlier that there may be inferences that an agent should not make with imaginary concepts. We do not attempt to categorize the types of rules that describe these inferences; this is domain dependent and can be done by the programmer setting up the domain rules. We can, however, provide a schema in the form of

counterfactual rules to distinguish the type of concepts (real or imaginary) that a counterfactual rule uses. A rule that applies to concepts that map to the real world would have the form:

```
(Predicate1 x True set  
denotes(x,"x",set) -> Predicate2 x True set)
```

Rules that apply to objects that do not map to the real world would have the form:

```
(Predicate1 x True set  
denotes(x,nil,set) -> Predicate2 x True set)
```

Thus, the denotes predicate enables us to distinguish whether or not concepts map to the physical world (and whether or not an agent believes that they do). The distinction between what exists physically and what is purely imaginary is important, but there are finer distinctions about existence that can be made. We examine these in the next section.

6.5. What Exists

We will now further subdivide the concepts in our ontology so that we may represent a finer distinction among different types of existence. The distinction of the type of existence attributed to an object will sometimes be useful in deception reasoning. This additional subdivision fits the framework of the ontology discussed so far.

We have taken what at first appears to be a Meinongian approach [Meinong, 1904]; we say that everything that is a concept exists. This

means that anything we can think about exists, at least as an idea. Anything that does not have an object concept can never be referred to in our system, and can therefore never exist in our beliefs.

Our approach is not quite the same as Meinong's because we agree strongly with Hirst's point [Hirst, 1989] that there are different types of existence. He lists the following nine kinds of existence:

- 1) Physical existence in the present real world, with causal interaction.
- 2) Physical existence in a past world (with causal interaction therein, and some indirect causal connection to the present world
- 3) Abstract, necessary existence, as of mathematical objects
- 4) A sort of doubly abstract existence, granted to objects such as "imaginary" numbers
- 5) Existence outside a world, but with causal interaction with that world.
- 6) Abstract, contingent existence in the real world.
- 7) Existence as a concept, which is abstract but contingent
- 8) Unactualized existence
- 9) Existence in fiction

We assume everything exists in the sense that we can reason with the concept, but some things do not physically exist at all or only in the past or future, as illustrated by the above list. The major distinction we use is whether something physically exists in the world. This is handled by the upper and lower case representations. Our system can account for each of the above types of existence; the

ability to handle many variations of existence gives the system flexibility.

Existence of type number one is merely represented by object concepts that map to physical objects, that is, the agent's belief set contains a denotes predicate for the concept. Existence of type two actually has two cases. The first case is when the object existed in a past world in which the agent did not yet exist. These objects must be described using predicates that indicate a past tense, such as:

`lived_during 1706 1790 Bdn_Franklin True set1.`

We would not have a denotes predicate for Ben Franklin because he does not physically exist anymore. The past tense of the predicate indicates past tense relative to the current timestamp (the timestamp of the world in which the statement appears). We can, however, include the past tense predicate 'denoted' to represent that Ben Franklin existed physically at one time. This enables us to distinguish Ben Franklin from Merlin; we might wish to represent that Merlin lived during the middle ages (in fiction), but we do not believe he physically existed even during his fictional lifetime.

The second case is when the object existed in a past world along with the agent but does not exist any longer. In this case, the agent will have a memory of the object. To discuss this representation, we must introduce a function `TIME(x)` that returns the timestamp associated with a possible world `x`. This function is necessary for the same reasons we introduced the `opinion(x)` function in Chapter Three; the timestamps are represented internally by the counterfactual engine, and we do not have access to them. The `TIME(x)` function allows us to refer

to the timestamp of a possible world.

We can now deal with things that existed in an agent's past. Consider a typical statement of the form

PREDICATE x True set

We represent that this existed in the past by the presence of this statement in some world w such that $TIME(w) < TIME(\text{current world})$. An agent can reason about previously existing objects by looking back into his previous possible worlds (i.e., the memory relation). The ability of the counterfactual engine to timestamp possible worlds gives us this ability to keep track of an agent's personal history.

Existence of types three and four is handled by object concepts that have no denotes predicate. Type five existence is possible in our system but we choose not to implement it because of our model of causality. If we wished to model some type of God-like interference, we could input information directly into an agent's beliefs rather than allow him to derive or perceive the information.

Type six existence is similar to type one except that the object is contingent. Most physical objects are contingent on some condition, but since we are using possible worlds, we deal with contingency by including the object in some worlds but not in other possible worlds. Hence, it is not a necessary object. Type seven is dealt with the same way; concepts exist in some worlds but not in others.

Type eight existence is an interesting case because it deals with projected events or objects in the future that have not existed and may never exist. These concepts are represented in an agent's mind by associating some future time with the object. Thus, the agent is

anticipating that the object will exist in some future time. These are not the same as goals because the agent may not necessarily desire these anticipated situations.

The fact that these anticipated situations exist as concepts means that an agent can reason about them, whether or not they ever begin to exist physically. To represent this we must include an explicit reference to a future time in the statement; this is because an agent cannot look for information in future possible worlds that have not been generated.

In order to reason with these anticipated events or objects, we must actually implement the $TIME(x)$ function and have access to the possible world timestamps. We can then include a time as an argument of a statement, and we can use the $TIME(x)$ function in counterfactual rules to compare this explicit time to the timestamp of the current possible world. This representation also enables us to deal with substitution of equalities involving different descriptions of the same time.

An agent can also look back from future worlds to see if the anticipated event or object actually came into existence. If so, it will exist in those worlds and will no longer be anticipated, but the record will be maintained that it was anticipated in the past. It is also possible that the agent's reason for anticipating something will be removed; in this case the object will no longer be anticipated and the belief that it is anticipated will no longer be true.

We can now introduce a predicate `ANTICIPATED` to represent things expected to exist at a later time. We will say that

ANTICIPATED X if and only if $\text{TIME}(X) > \text{TIME}(\text{current world})$

The final type of existence mentioned above, existence in fiction, is handled by considering fictional existence to be an attribute of something. Thus, we would have statements such as:

fictional Tom_Sawyer True world

Tom_Sawyer would not have a denotes predicate since he does not physically exist. A concept like Tom_Sawyer can be used as any other imaginary concept is used; if the fact that the concept is fictional is significant, rules can be written to accept only objects that have the fictional attribute.

We can represent the formalization of these different types of existence using counterfactual language rules. We will use the functions $\text{opinion}(X)$ to determine the modal category of statement and $\text{TIME}(X)$ to determine the timestamp of the possible world containing the statement. We will let T represent the timestamp of the current world and let t be a variable that can represent any time. A modal category (opinion) of 0 indicates that something is necessary because it can never be rejected; a higher modal category indicates that something is contingent (it may or may not exist in a possible world, depending on the rest of the possible world). We define the types of existence as:

1) ($\text{denotes}(x, "x")$)

$\text{opinion}(x) > 0$

$\text{TIME}(x) = T \quad \rightarrow \text{TYPE 1 EXISTENCE}$

2) ($\text{denotes}(x, "x")$)

$\text{opinion}(x) > 0$

$\text{TIME}(x) < T \quad \rightarrow \text{TYPE 2 EXISTENCE}$

- 3) (denotes(x,nil)
 opinion(x) = 0
 TIME(x) = t -> TYPE 3 EXISTENCE)
- 4) (denotes(x,nil)
 opinion(x) = 0
 TIME(x) = t -> TYPE 4 EXISTENCE)
- 5) not used
- 6) (denotes(x,nil)
 opinion(x) > 0
 TIME(x) = t -> TYPE 6 EXISTENCE)
- 7) (denotes(x,"x")
 opinion(x) > 0
 TIME(x) = t -> TYPE 7 EXISTENCE)
- 8) (denotes(x,"x")
 opinion(x) > 0
 TIME(x) > T -> TYPE 8 EXISTENCE)
- 9) (denotes(x,nil)
 fictional x
 opinion(x) > 0
 TIME(x) = t -> TYPE 9 EXISTENCE)

Note that types 3 and 4 have the same formalization; they are distinguished by the type of rules in which they are used. For example, you cannot assign an imaginary number as the number of physical objects that are in a room.

We should justify why we place so much emphasis on the existence of object concepts, whether or not they map to actual objects. An

important reason is that the distinction is important for deception reasoning; deception involves one agent trying to convince another that imaginary objects or relations actually exist. Another reason for the emphasis is that we can reason with these concepts, so it follows that they must exist, even if only as imaginary concepts.

Most people have a very clear, definitive concept or idea of what a unicorn is. We can identify a unicorn as easily as an elephant; unicorns may be more recognizable than some animals that physically exist. The point, however, is that we have a clear concept of the attributes of a unicorn; we can reason about how many horns it has, how many tails, etc.. There are definitive attributes about a unicorn just as there are about objects that physically exist.

As an example, think about the current President. There are several attributes that are intrinsic properties of the man. Thus, we cannot answer the classic question about whether he is standing or sitting except to say that we do not know (see Chapter Five). We can, however, speak confidently of his hair color when he is not present. Attributes like hair color, eye color, shoe size, etc. are fairly permanent and can be considered attribute concepts linked to the object concept of the President.

Since imaginary objects have the same type of attribute concepts associated with them, there is no reason to treat them differently in general or to say they do not exist. An interesting problem is suggested in [Hirst, 1989]. He claims that Meinong's approach, which gives status to objects like gold mountains that do not physically exist, has problems with statements like "The gold mountain is 1000

metres tall." Hirst says that the 1000 metres refers to the object, not the height of the idea, but there is no object to refer to.

This analysis does not apply to our system for two reasons. First, we represent what an object denotes in the real world, so the attribute concept should be assigned to this physical object when a mapping exists. The second reason applies when a concept does not map to a physical world; the attribute should not be assigned to the object concept, but instead an entirely new concept is created. Thus, we would conceive of a 1000 metre tall gold mountain. If we can conceive of a gold mountain that does not physically exist, then we should be able to conceive of a gold mountain that is 1000 metres tall.

A similar situation occurs if we say "Cousin Bill is about six feet tall" even though we have not seen him since he was a child. We have an object concept of Bill that corresponds to an actual person, but we have never seen that person as an adult. Still, it is possible to assume that he grew to the same height as his father; he certainly is no longer three feet tall. Hence, people may assign attribute concepts to object concepts that map to the real world, and they create new concepts that define a smaller class of concepts by combining object and attribute concepts.

6.6. The King of France

A classic problem concerns the truth values of statements about objects that do not exist. Can a statement like "The King of France is wise" be evaluated? It can in our system because we are working in the

context of beliefs. Thus, we avoid the complex issue of whether this statement has a truth value in the real world. If we have an object concept for the King of France, even an imaginary or fictional concept, then we can see if it is linked to an attribute concept for "wise." Thus, an agent may incorrectly believe there is a King of France, or he may consider the general concept of what the King of France ought to be like if he were to exist physically.

This point indicates the importance of determining what object concepts are intended to represent. Does "the King of France" mean "the present King of France" or some more general concept? Sometimes the context will give a clue to the intended meaning. Olsen [Olsen, 1987] uses as an example the question "Does the pitcher always try to hit the batter with the ball?" The problem lies in determining if "the pitcher" refers to a particular pitcher in a particular game or to the general concept of the position of pitcher. Some additional information about the situation or context is needed. If a particular pitcher is intended, the object concept will map to an individual in the world. If the position of pitcher is intended, then the object concept will embody the concept of what a pitcher does and will not map to the real world.

With information about the context, we must decide if the King of France is a general concept or a particular King. If a general concept is intended, then we are free to have any belief we like about what a King of France is like, even if we are mistaken. This should surprise nobody because we can think of our President as a general concept rather than the present individual, and we hope that we may say "The

President is a wise man." Thus, people are free to hold whatever beliefs they wish about a general concept.

If, on the other hand, a particular King of France is indicated, then we must decide which one. Since no time is indicated in this example, we will assume the present King of France is referred to. Thus, our object concept will be that of the present King of France, and we may reason with this concept the same way we would reason with the concept of a unicorn; the present King of France is an imaginary object, and we may or may not be aware that he does not exist physically.

If we know that there is no King of France now, then we will indicate with a denotes predicate that the King_of_France maps to nil in the real world; we will not permit ourselves to have beliefs like "the King of France is wise" that would assume there is a physical object who is King of France and is wise. This is accomplished by letting 'wise' be an attribute associated with concepts that map to the real world. We would have:

WISE x True *set

denotes x nil *set -> inconsistent

Such beliefs do not appear at all in possible worlds; we do not presume that the negation of the proposition is true because there is no King of France whose wisdom we can judge. We can include additional rules that will permit object concepts that do not physically exist but are fictional to be wise (such as Solomon).

If, on the other hand, we do not know that there is no King of France, then there is nothing to prevent us from having the belief.

This is possible because we can believe that the King of France is wise whether or not the proposition can actually be evaluated. This is because our beliefs concern concept propositions and are not dependent on the value of a real proposition. Although we intend them to correspond to propositions, we may be mistaken.

Hence, if we do not know that there is no King of France, then we simply have a mistaken belief and can reason about him as if he were real. We believe that the object concept actually maps to the world in the same way a child believes that Santa Claus is a real man. The child can say that Santa Claus is fat and wears a red suit, and an adult may say that the King of France is wise.

Thus, our representation enables us to distinguish concepts that map to the real world from those that do not; in addition, we can refer to the denotations of the concepts that do map by using a variation of McCarthy's predicate 'denotes'.

6.7. Conclusions

Our system is defined around a complex web of physical and mental structures. We use concepts as primitive mental structures that allow us to build abstract situations and reason about physical and imaginary objects. The intermediate steps between the real world and beliefs are important for being able to generate beliefs that are independent of the real world, although they are intended to reflect it. Thus, we may make mistakes as well as reason about imaginary objects.

7. Formal Basis

We will now develop the axioms that formally characterize our system. This will enable us to relate our semantics-based possible worlds to syntactic modal logic axiom systems. Note that this development is intended to explain the relations between possible worlds; we do not use syntactic methods to make deductions. Moore has previously translated possible worlds into first order logic. An inference engine could then be used to make deductions, and the results could be related back to possible worlds [Moore, 1985]. This method, however, results in long proofs for which there is no known effective theorem prover. Our system uses a set of semantic rules that permit direct operation on and generation of the possible worlds.

There are many levels at which our system may be characterized. Before proceeding with this characterization, it is necessary to define our terminology to avoid confusion. We have previously referred to an agent's belief set and his models of other agents' belief sets as possible worlds. In addition, we have used the term world to refer to a collection of these belief sets that occur at a single stage of reasoning. Thus, we have used worlds to distinguish the beliefs of different agents and beliefs held at different times.

To avoid confusion, we will henceforth refer to single belief sets as possible worlds, and we will refer to a collections of worlds at some point in time as a possible superworld. Possible worlds will be represented as single circles while superworlds will be indicated by double circles. In Figure 12 (page ####), therefore, W1, W2, etc.

indicate superworlds and a_1 , a_2 , b_1 , b_2 , etc. indicate worlds corresponding to single belief sets. Each possible superworld, then, may contain many possible worlds. Some of these will be core worlds, i.e., the belief sets of distinct agents. Other worlds will contain models of nested beliefs constructed by the core agents.

With this structure in mind, we may proceed with our formal characterization. The complexity of our structure results in many relations. This chapter examines the framework of relations between distinct possible worlds and superworlds. Chapter Six discussed the interpretation of our (partial) possible worlds, and Chapter Three discussed the way the counterfactual engine operates on possible worlds to generate new possible superworlds.

There are several types of relations between possible worlds and superworlds in our system. The first of these is the temporal framework of the generated superworlds. The second relation we will characterize is the accessibility between possible worlds that captures the nesting of beliefs. Finally, we will characterize the memory relation that allows an agent to look back into his belief sets from previous times.

7.1. Time Characterization

The first relation between worlds that we will characterize is the time ordering of the generated superworlds. The formal basis of this relation will be cast in the tense logic discussed in [Rescher and Urquhart, 1971]. After we determine the tense logic system that

corresponds to this relation, we can find the modal logic system (S4, S4.3, etc.) equivalent to the desired tense logic system.

The tense logic that characterizes the temporal ordering relation of our system will depend on the properties of time (e.g. linear, dense, infinite) that we choose to model time in our system. We will first examine the minimal tense logic system, K_t , that was proposed by E. J. Lemmon. We can then determine the appropriate extension as we decide what properties our system should have.

In developing tense logic systems, Rescher and Urquhart use two operators, R_t and U . R_t is an operator that evaluates a proposition at time t . In other words, it gives temporal context to propositions. The U operator defines the relation of temporal precedence. Thus, $U_{tt'}$ represents that the time t is before the time t' . A special constant n is also introduced that essentially indicates the present instant of time.

With these operators to define present time and precedence, Rescher and Urquhart define four tense operators: F , P , G , and H . These tense operators are interpreted as:

Fp : It will be that p .

Pp : It has been that p .

Gp : Henceforth always p .

Hp : Heretofore always p .

and they are defined as [Rescher and Urquhart, 1971]:

Fp : There exists t such that $[U_{nt} \ \& \ R_t(p)]$

Pp : There exists t such that $[U_{tn} \ \& \ R_t(p)]$

Gp : For all t $[U_{nt} \supset R_t(p)]$

H_p : For all t [$U_{tn} \supset R_t(p)$]

where $\&$ represents conjunction and $\rightarrow$ represents material implication.

These tense operators are used in formulating the axioms that characterize tense logic systems. In terms of these tense operators, K_t is axiomatized as [Rescher and Urquhart, 1971]:

$G(p \supset q) \supset (Gp \supset Gq)$ (G1)

$H(p \supset q) \supset (Hp \supset Hq)$ (H1)

$\neg H \neg Gp \supset p$ (G2)

$\neg G \neg Hp \supset p$ (H2)

This system is minimal in the sense that it has the properties a tense logic should have, but it does not depend on any assumptions about the structure of time. We will extend this system with additional axioms as we determine the structure of time for our system.

The first property we must examine is whether time should be linear or branching. Since our system may branch into several possible superworlds, it appears that time branches in our system. This is not so, however, because we are actually just representing different possibilities. There is only one future and one past even if we can represent different possibilities of what could be or could have been. Since we do not believe that time branches in the actual world, we do not want time to branch in our system. Thus, we want our system to be linear in time.

To disallow branches in time, we must assert that:

$t < t' \vee t = t' \vee t' < t$.

This simply says that two instants of time must have an ordering or must be identical; they cannot occur concurrently without being

identical. The axiomatization of K_l consists of K_t plus the following four axioms [Rescher and Urquhart, 1971]:

G3: $Gp \supset GGp$

H3: $Hp \supset HHp$

G4: $[G(p \vee q) \ \& \ G(p \vee Gp) \ \& \ G(Gp \vee q)] \supset (Gp \vee Gq)$

H4: $[H(p \vee q) \ \& \ H(p \vee Hp) \ \& \ H(Hp \vee q)] \supset (Hp \vee Hq)$

Another property we must determine for our system is whether time is infinite in either direction. Based on our physical knowledge, we believe that time had a beginning and will eventually come to an end. We do not, however, have any way of determining when that beginning or end occurred or will occur. Since our system will not need to capture the idea of either endpoint, we will assume for our purposes that time is infinite in either direction. This gives us the system $K_l^{00\pm}$ and requires the addition of the axioms [Rescher and Urquhart, 1971]:

G5 $Gp \supset Fp$

H5 $Hp \supset Pp$

The next property we must determine is whether our system should use a dense or discrete model of time. If we use a discrete model of time, then the "ordering of temporal instants is isomorphic to that of the integers" (Rescher and Urquhart). Dense time, on the other hand, corresponds to the rational numbers.

We have chosen a dense model of time because it seems intuitive that our system should have some flexibility in reasoning about the order of changes in the world. This analysis is like that given by Pendergrast [Pendergrast, 1988] because our mental events are treated just like physical events. If an event is determined to follow another

immediately, we may determine later from additional information that there was yet another state of affairs that occurred between the two. In that case, we must be able to assign the new state of affairs to a time between the previously determined states. If the previous states had been assigned to consecutive, discrete instants, however, this would not be possible. It therefore is best to use a dense model of time.

Since we are using a dense model of time, we must decide if it should also be continuous. A continuous model of time would correspond to the real numbers. Intuitively, we may have difficulty comprehending the difference between the rational and real numbers because the granularity of the rationals is so fine. We can make our decision, however, on the basis of what we are trying to model. Our system captures changes in the truth value of propositions. We do not attempt to specify the state of truth values of propositions at every instant of time. In fact, humans certainly do not take note of everything about them at every instant of time.

Specifying every value at every instant would be an inefficient solution to the persistence problem, a part of the frame problem to be examined in a later chapter. Since we do not need to capture continuous values, we will not require a continuous model of time.

Density in time is captured by adding the following axioms [Rescher and Urquhart, 1971]:

G6 $GGp \supset Gp$

H6 $HHp \supset Hp$

This completes the axiomatization of the tense logic model of our

system. The system we have arrived at is $K1_d^{00\pm}$. According to Rescher and Urquhart, this is equivalent to an axiom system presented in [Prior, 1966].

We next wish to determine how this tense logic system relates to modal logic systems. Necessity and possibility can be defined in terms of the tense operators. If these definitions are added to the tense logic system, then the axioms can be shown to correspond to one of the modal logic systems (S4, S4.3, S5, etc.). Rescher and Urquhart give two sets of definitions for necessity and possibility in terms of tense calculus. Since we are concerned with the past as well as the future, we will use the definitions based on the Aristotelian/Megarian construction of modality. Rescher and Urquhart give the following definitions:

$$D\Box 2: \Box A \supset A \ \& \ GA \ \& \ HA$$

$$D\Diamond 2: \Diamond A \supset A \vee FA \vee PA$$

From this definition we determine that we have an S5 modal system. This follows from Rescher and Urquhart's assertion that any linear time system will produce an S5 system based on $D\Box 2$, regardless of assumptions about the density of time. Adding $D\Box 2$ to our tense logic system results in the following axiomatization:

$$Ax1: \Box p \supset p$$

$$Ax2: \Box(p \supset q) \supset (\Box p \supset \Box q)$$

$$Ax3: \Diamond \Box p \supset p$$

$$Ax4: \Box p \supset \Box \Box p$$

Thus, we have determined that the relation between possible worlds is characterized by an S5 modal logic system, and we have determined

the axioms that describe it.

7.2. Relations Among Nested Beliefs

The relation between worlds that we examined in the preceding section determined the framework for generating new possible superworlds. The relations that capture nesting of beliefs and memory define which sets of worlds may be used together to make deductions. Since a person may always use his own beliefs in reasoning, both of these relations are reflexive. Because we use distinct worlds to represent separate sets of beliefs, our system allows an agent to use information from other worlds to make deductions. This section characterizes the relation between worlds within a superworld.

The accessibility relation that captures the nesting of beliefs specifies an agent's access to the worlds that represent his model of another agent's beliefs. Figure 12 shows how this might be represented in a single superworld. Note that there are certain "core" worlds that have access to several other worlds. Each of these core worlds represents a different agent. The worlds related to the core worlds are that agent's models of what other agents believe and what other agents believe that the core agent believes. Each agent can therefore reason with his own beliefs and with his own models about what others believe; he cannot reason with another agent's actual beliefs since he cannot read minds.

Changes made to a belief set will be dependent on the previous beliefs in that belief set; this means that this relation is reflexive.

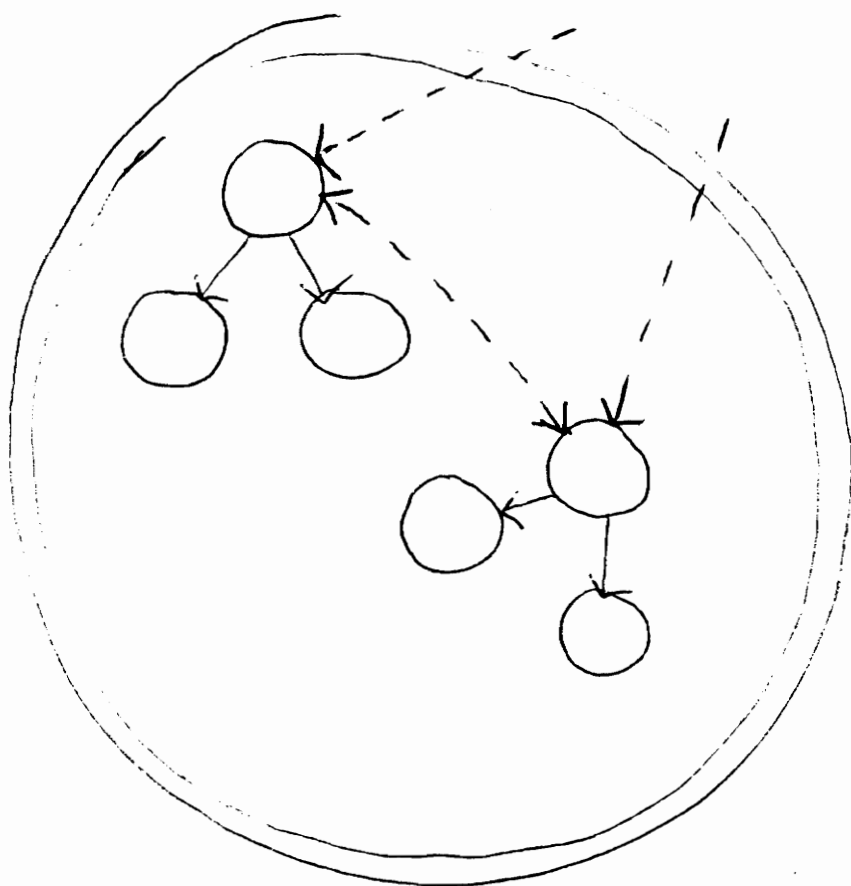


Figure 12. Accessibility between possible worlds representing nested beliefs.

It is also antisymmetric because a nested model of another agent's beliefs may not have access to the beliefs of the core agent who constructed the model. Transitivity is also required because some worlds represent models of other agent's beliefs about the core agent's beliefs. Since these worlds are actually models within the mind of the core agent, this allows him to reason about what other agents think about his beliefs, and therefore the relation is transitive.

The dashed arrows in Figure 12 represent information coming to agents either from perception or by communication with other agents. Note that this input can only be directed to core worlds because they correspond to the distinct agents. The solid arrows, on the other hand, represent the accessibility relations between possible worlds. These accessibility arrows capture the essential information concerning the nesting of beliefs. Since this relation is only required to be reflexive and transitive without being symmetric, it appears to correspond to the modal system S5. We will next attempt to formalize some requirements on the relation to see if the results actually do belong to the system S5.

The first assumption we make is that all agents have some common knowledge. What this common knowledge is will depend on the problem domain, but it implies that an agent will assume that other agents share these beliefs when he constructs his model of their beliefs. Essentially this means that if p is necessary then p is present in each world. This corresponds to axiom A5 of the system T [Hughes and Cresswell, 1968]:

$$[A5] \Box p \supset p.$$

The next assumption we make is the simplifying assumption that all agents have equal reasoning capabilities. An agent will use this assumption when making deductions about how another agent's beliefs will change. If it is necessary that p implies q , then if p is necessary then q is necessary. The way the system enforces such constraints is explained in Chapter Three.

This is not a good assumption because it gives agents ideal reasoning capabilities that are unrealistic; humans do not make all deductions they are capable of, and some agents make deductions that other agents do not make given the same information. This assumption is necessary, though, because our system has no way of determining which deductions not to make; if the conditions of a rule are met, then the rule will fire. We should note, however, that agents will make only those deductions for which they have rules; this solves the problem of logical omniscience the way that Konolige does [Konolige, 1985] by associating a particular set of reference rules with each belief set.

At present, we must maintain the assumption that agents will make all assumptions they are capable of relevant to a new counterfactual. The assumption is equivalent to axiom A6 of system T [Hughes and Cresswell, 1968]:

$$[A6] \Box(p \supset q) \supset (\Box p \supset \Box q).$$

To this point, we have shown that our system satisfies all of the axioms characterizing the modal logic system T. The properties of the accessibility relation, however, led us to believe that it corresponds to an S5 modal logic system. If the correct system is S5 rather than

T, then we must have the following additional axiom [Hughes and Cresswell, 1968]:

$$[A8] \Diamond p \supset \Box \Diamond p$$

This axiom says that if a belief is possible in one world, then it is necessary that it is possible in all worlds related through accessibility. This makes sense intuitively for our system because all of the accessible worlds are really models in a single agent's mind. Thus, if it is possible for that agent to have that belief in one of his models, then it must be at least possible that he can have the same belief in any other model he creates. This means that the nesting relation can be characterized by an S5 modal system.

These are the only assumptions needed to capture the nesting of beliefs. When an agent constructs a model of another agent's beliefs, he will assume that that agent has basic knowledge and reasoning capabilities similar to his own. The differences in the beliefs held by two agents result because the two agents may receive different inputs. For example, an agent sees an event take place but believes that another agent did not because he is not present.

Intuitively, the reflexive and transitive properties seem sufficient to capture the nesting of beliefs if we look at the problem a different way. We have required that an agent can reason with all of his own beliefs due to reflexivity. The additional worlds that an agent has access to are really models of belief sets that he has constructed in his mind. If we draw a circle about a core world and all its related worlds, we can consider everything inside to be contained in the core agent's mind. If the reflexivity is then applied

to the entire contents of that circle, then the agent is reasoning entirely with his own beliefs. This implies that he has direct access even to the models of his own beliefs that he believes another agent constructs. Hence, the transitivity relation exists but does not have to be used to access the innermost nested beliefs.

The major advantage of separating the models into separate possible worlds is that it shows more clearly how these models can change independently of each other. That is, John may think that Mary received some new input and changed her beliefs, but Bill did not receive that input and therefore retains his beliefs.

7.3. Memory Relation

We next examine the relation that captures the memory of the system. This memory relation allows an agent to examine his beliefs in previous worlds. Of course, he still has access only to the worlds he could access if he were actually in that previous superworld. In other words, an agent still cannot read another agent's mind when looking back into the past; he can only remember his own thoughts.

The double arrows in Figure 13 represent this relation. The single arrows indicate the direction of the flow of time. It is important to remember in this discussion that this relation is distinct from the one that captures the nesting of beliefs. We are not concerned with the level of nesting of a world; instead, we are concerned with the time ordering of the worlds.

It is important to note that an agent may only retrieve

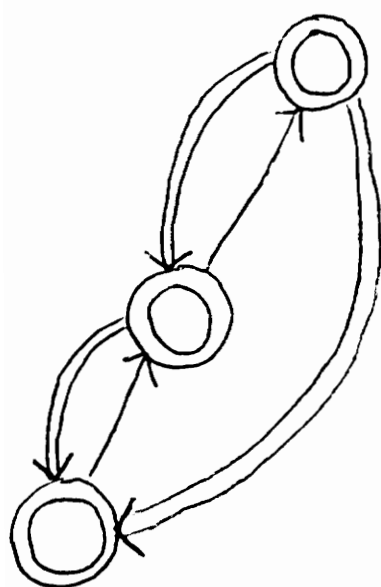


Figure 13. Memory relation.

information from previous superworlds; he can never change past information. The result of this observation is that our system is actually monotonic in its reasoning with beliefs. Although changes of beliefs appear to be nonmonotonic, they really are not because the new beliefs have a different timestamp associated with them. It is possible for people to change their beliefs about the past, but it is never possible for them to change their beliefs in the actual past. We will return to this topic in Chapter Eight when we discuss the frame problem applied to belief systems.

The memory relation has properties similar to the relation discussed in the first section of this chapter. The arrows in Figure 13 and some intuitive reasoning demonstrate the basic properties of the relation. As with the nested beliefs, this relation is clearly reflexive because an agent can reason with his own beliefs at the current time. Since we are attempting to model memory, an agent can only look back in time. This means that the relation is antisymmetric.

We also assume that it is a transitive relation. If you can remember yesterday and yesterday you remembered the day before, then today you should be able to remember the day before yesterday. Obviously, this does not take into account events such as forgetting information. We have not yet attempted to model forgetting because a procedure for choosing what things to forget and when to forget them would be a major project in itself. This problem would be similar in scope to developing a method for determining what deductions an agent should and should not make.

The ability to remember allows an agent to reason about what he

used to think and what he used to think other agents thought. To avoid problems with branching time, we should consider the relation only among worlds associated with a given timeline. The reflexivity and transitivity of the relation suggests that this corresponds to an S_5 modal system. We will examine the tense logic properties to determine if this is accurate.

The memory relation shares many of the properties of the time ordering relation discussed earlier because we will also consider time infinite and linear. While an agent obviously cannot remember events looking back to infinity, we do not wish to create some arbitrary starting time at which his memory began. The major difference from the time ordering relation is that we are now looking back in time rather than forward. Also, the function of the relation is different since it determines what information can be accessed for use in reasoning.

Time is assumed to continue forward from the present even though we can have no memory of the future. This allows a well-defined timeline even though we do not have access to the future portion. With this in mind, the relation is nearly identical to the time ordering relation for new superworlds. The basic axioms of the tense logic system K_t are needed along with the assumptions of linear and transitive time.

The only difference is that we do not require the assumption of dense time; this is because we are remembering discrete mental states at past points in time. Since time is still assumed dense in the future, however, we will include dense time in the past so that our timeline remains symmetric. Hence, we arrive at the same result as for

the first relation and find this equivalent to the $K1d^{00\pm}$ system in tense logic. It is also therefore equivalent to an S5 modal logic system.

7.4. Conclusions

Thus, we have found that each of our relations among possible worlds is reflexive, transitive, and anti-symmetric; each is equivalent to an S5 modal logic system. Although this result appears to be coincidental, it is important to the operation of the system. The result is quite remarkable, in fact, considering the diversity of types of relations between worlds. One relation describes the generation of new worlds in the future, one describes the nesting of beliefs among agents at the present, and one describes the memory of past worlds. The common modal logic system that these three relations share enables them to interact and creates a theoretically uniform system that performs as desired.

8. (Mental) Event Logic and the Frame Problem

Since thoughts are events in our system, we must define an event logic that will specify when and how an agent's beliefs will change. Our system reasons with concepts rather than physical entities; the events that we base our reasoning on will be mental events. For our purposes, we will consider a mental event to be a change in truth value of an agent's belief. This change may result from the agent's perception, communication with other agents, or introspection and hypothetical reasoning.

We will examine how our model of time and our assumption of causality result in an event logic that allows our system to model change in a consistent manner; we will be able to predict all possible worlds resulting from an event occurring in a given world because we have precisely specified the logic of the system. Following that discussion, we will examine the frame problem and the theory of our solution. Finally, we will show how our theoretical framework is implemented.

8.1. Model of Time

Time is an important consideration in a system of beliefs because: 1) we must reason about time, and 2) the reasoning process itself takes time. We will also find that time is closely related with events and changes in our system.

When we discussed the formalization of our time ordering

relationship in tense logic, we assumed several properties for our model of time. The assumptions that we made were:

- 1) Instants of time map to the rational number line.
- 2) There is a linear, total ordering of instants.

These assumptions give us a model of time that is dense, linear, and infinite.

We discussed the reasons behind these assumptions in Chapter Seven when we axiomatized our model of time. We believe our model of time is useful because we represent sets of beliefs with (partial) possible worlds that correspond to mental states. These mental states change as a person's beliefs change; thus, a possible world will be associated with a particular instant of time, and that particular possible world will consist of all statements held true at the associated instant of time.

The assumption of linearity means that time instants are fully ordered. We will represent this with the "<" operator, where:

$t_1 < t_2$ if and only if t_1 precedes t_2 .

Intuitively, we know that there is an ordering on events as well as time. Two events must either occur at the same time or one of them must occur before the other. We will find that events can be directly related to instants of time to obtain their ordering. This presupposes that events are instantaneous, but we can adjust the granularity of representation so that the events that interest us for a given world may be considered instantaneous. First, however, we will specify how beliefs relate to our model of time.

8.1.1. Instants vs. Intervals

We specified in Chapter Six how our system's statements are to be interpreted. At that time, we noted that concept propositions take on truth values to generate abstract situations, but we did not associate time with the truth values of concept propositions to indicate when those truth values were believed to be valid. To specify an event logic, we must now relate those truth values to our model of time.

As previously defined, a mental event is an agent's perception of the change in truth value of a proposition. Thus, we are concerned only with the agent's belief about the truth value of the proposition; the actual truth value may differ from the agent's belief, but the agent must not be aware of this or he would change his belief. In Chapter Six, we also defined the relation between the statements of our system and the propositions embedded in those statements. Since our statements correspond to agents' beliefs about propositions, then, an event will correspond to a change in the truth value of a statement.

To assign some temporal meaning to truth values, we must decide whether those values are defined by intervals or instants of time. Since intervals and instants may be defined in terms of each other, it is possible to take either as primary. We will use both in defining our system. Our solution is that truth values change at instants of time, and they maintain their state during the intervals between changes.

Mental events correspond to changes in an agent's beliefs; this means that the truth value of a proposition changes in the possible

world representing that agent's beliefs. Since a truth value is either true or false it must change instantaneously, so we will say that events occur instantaneously. We also may say that events correspond to the instants of time in our model. We will not be concerned with any instants occurring between events, since all truth values remain unchanged between events.

Our mental event logic parallels the event logic used in [Pendergraft, 1988]. An event occurs at some instant and produces changes in the world. These changes effect other changes until a stable, consistent system is again obtained. Each of these changes is an event and also corresponds to an instant of time. At each instant of time, we begin to believe that some statements are true, and we stop believing that other statements are true. If more than one event occurs simultaneously then the events are associated with the same instant of time, and the conjunction of the events must be considered in determining the appropriate changes to the world.

Every statement, then, changes value at some instant of time (an event) and maintains the new value until it changes again. The interval over which the truth value persists will contain at least one instant of time. These intervals will be left-open intervals; this choice is the most logical one intuitively. Completely open intervals would leave undefined points at the instants where changes occur, and completely closed intervals might cause contradictions at those instants. This leaves only left-open or right-open as choices. We choose left-open because right-open intervals would imply that the value of the statement had already changed when the event causing the

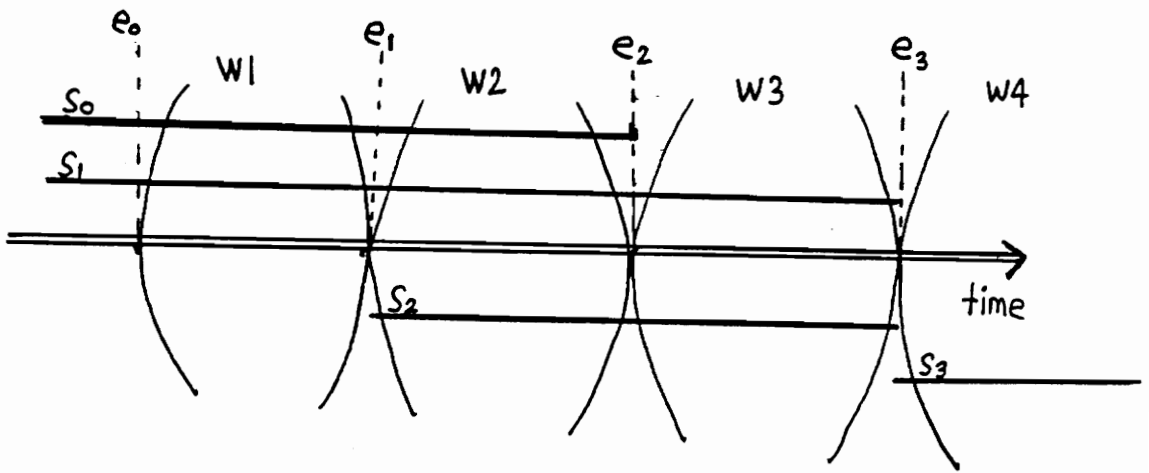
change occurred.

As specified in Chapter Seven, our possible worlds are defined over intervals of time; each possible world is associated with an integer timestamp that indicates the time at which the world begins to exist. The possible world exists until there is a change that causes a new possible world to be generated. The timestamp of the newly generated world marks the end of the previous possible world. We define our possible worlds as the set of all statements true at the instant of time corresponding to the timestamp of the world. Since the values of these statements persist over an interval, the possible world persists also. Conversely, we can define a statement to be true in a possible world if it has a true value at the instant corresponding to that possible world.

Figure 14 depicts a timeline and the temporal relation between mental events, statements, and possible worlds. Events, statements, and possible worlds will be denoted by $e(t)$, $s(t)$, and $W(t)$, respectively to indicate explicitly the instant of time associated with them.

Note that although events are defined only at instants of time, the truth values of statements are defined over intervals. This feature is what enables the truth values of statements to persist in new possible worlds without mentioning that they have not changed. If the truth value of a statement changes, the change is assumed to occur immediately after the event that causes the change.

Since statements are defined over intervals, we need two arguments to refer to the interval in which the statement is true, such as



$W_1:\{s_0, s_1\} \implies^{e_1} W_2:\{s_0, s_1, s_2\}$

$W_2:\{s_0, s_1, s_2\} \implies^{e_2} W_3:\{s_1, s_2\}$

$W_3:\{s_1, s_2\} \implies^{e_3} W_4:\{s_3\}$

where $\implies$ is a causality arrow mapping one possible world to another via an event

Figure 14. Temporal relations between events, statements, and possible worlds.

$s(t, t')$, where t is taken to be the time associated with the first possible world where s is true and t' is the time associated with the last possible world where s is true. When we write $s(t)$ with a single argument, however, we are referring to the truth value of the statement s at instant t .

We can define:

$$W(t) = s_1(t) \& s_2(t) \& \dots \& s_n(t)$$

and

$$E(t) = e_1(t) \& e_2(t) \& \dots \& e_m(t)$$

This indicates that a world $W(t)$ is defined by the set of all statements s_i that are true at time t . We use only one argument to refer to a possible world, and this argument is the timestamp associated with the beginning of that world. $E(t)$ is the conjunction of all events that occur simultaneously at time t .

We then have:

$$W(t) \implies^{E(t)} W(t')$$

indicating that the events at time t map the world at time t into a new world at time t' . " $\implies$ " is in this case a causality arrow indicating that $E(t)$ causes a mapping from $W(t)$ to $W(t')$. Below we shall investigate the relation of the causality and counterfactual arrows.

8.1.2. Events Related to Instants

We have defined events to have a direct correspondence to instants. We will next explain why this is possible even though it may seem counter-intuitive. After that, we will investigate a formal

attempt toward unification of time and events.

Although we defined statements over intervals, we stated that events occur at a given instant. This does not make sense for many events that seem to occur over an interval. An example of such an event would be "John walked to the store." Obviously, there was some time involved in John's walking; he did not flash to the store in an instant.

Such an event seems to consist of a series of events that occur over an interval. The example above might consist of subevents like "John started to walk to the store", "John was walking to the store", and "John arrived at the store." These subevents may or may not be important in the problem domain. Thus, the problem becomes one of granularity: how precisely must the event be defined? Whether we refer to the action as an interval or instant in English is controlled by the aspect of the verb of an utterance [Venkateshwaran, 1990].

In either case there is a solution. If the subevents are important, i.e., references must be made to John while he is walking, then the subevents should be specified explicitly. For example, the following statements could be used:

- 1) STARTS_WALKING John to_store True set1
- 2) IS_WALKING John to_store True set1
- 3) STOPS_WALKING John at_store True set1

What we have done is to pick out some instant or instants during the interval that John is walking and generate new possible worlds. We then have worlds that allow us to refer specifically to the action of walking. Our predicate IS_WALKING will become true and remain true

until some event at another instant causes it to change value. The same event should cause `STOPS_WALKING` to become true.

If not concerned with subevents that occur during the walk to the store, then the solution lies in looking at the problem properly. When the interval during John's walk is unimportant, we should simply interpret the statement literally. Since it says "John walked to the store," it becomes true only when he completes his walk and is at the store. With this more general description of the statement, the only important information is that John has arrived at the store and that he walked there. We do not need to know about what happened along the way. Thus, we treat the event as occurring at the instant John arrives at the store, since it could not be true that he walked to the store unless he has arrived.

8.1.3. Formalization

We next turn our attention to a formal attempt at unifying concepts of time and events. This formalization is proposed by Turner in [Turner, 1984]. The important difference to note is that events for our system are mental events. In many cases mental events may be triggered by physical events; either type of event is merely some incident that causes changes in the world. Since we have defined our mental events to parallel a system dealing with physical events, the application of Turner's formalization is valid for our system.

To examine Turner's definitions, we need two relations: temporal precedence and temporal overlap. The relation of temporal precedence,

"<", and the relation of temporal overlap, "0", are defined as:

Def <: $e_1 < e_2$ iff $t_1 < t_2$

Def 0: $e_1 0 e_2$ iff $t_1 = t_2$

where e_i are distinct events and t_i is the time associated with event e_i . These definitions are straight forward since we are using a linear, ordered model of time.

Turner's Definition 4.1 consists of seven postulates based on the above definitions of temporal precedence and overlap. Since these definitions hold for our system, the postulates also hold for our system. Note that the t_i associated with each e_i is implicit since the events correspond to instants of time. The postulates are:

(E1) $e_1 < e_2 \rightarrow \neg(e_1 0 e_2)$ REPLACE $\rightarrow$ with horseshoes

(E2) $e_1 < e_2 \ \& \ e_2 < e_3 \rightarrow e_1 < e_3$

(E3) $e_1 0 e_2 \rightarrow e_2 0 e_1$

(E4) $e_1 0 e_1$

(E5) $e_1 < e_2 \rightarrow \neg(e_1 0 e_2)$

(E6) $e_1 < e_2 \ \& \ e_2 0 e_3 \ \& \ e_3 < e_4 \rightarrow e_1 < e_4$

(E7) $e_1 < e_2 \vee e_1 0 e_2 \vee e_2 < e_1$

Turner's Definition 4.2 defines instants of time in terms of events. It says that instants are maximal subsets of pairwise overlapping events. Since we have defined our events to correspond to instants, this definition applies to our system. Each instant is the overlap of corresponding event with itself or with any other simultaneous event.

Turner's Theorem 4.3 states that instants defined in terms of events are ordered temporally in a strictly linear fashion. This is

obviously true for our system since we have defined our events to correspond to instants. Also, the temporal precedence relation for events and for instants is the same. Thus, our system fits nicely with the formal framework of Turner.

8.1.4. Causality

We use the assumption of causality in our system because we do not believe that changes occur randomly without cause. Humans would have great difficulty functioning if the beliefs in their minds simply changed without reason. Just as one can argue that changes in the world are caused by events, we believe that changes in a person's beliefs are also caused by events such as perception, reasoning, and hypothesis generation.

As defined earlier, these mental events are changes in truth values that the agent perceives. Whether these events are triggered through direct perception, communication with other agents, or conscious reflection, they must occur before any other changes take place.

We will define a cause for a belief in the following way (this also parallels the causal schema used for physical events in [Pendergraft, 1988]):

Def. A cause for a belief is a set of conditions that
are sufficient to justify including that belief
in a belief set.

The set of conditions referred to will consist of other beliefs and the

level of confidence in them. Thus, we are saying that certain beliefs should cause a person to have other, relevant, beliefs. If it is inconsistent to accept the new belief, the counterfactual engine will determine how to amend the beliefs to restore consistency.

Note that we have not stated that a cause must be a necessary condition. That is because there may be several distinct causes for believing something. We may have that:

$P \implies Q$ and

$R \implies Q$.

If we believe P , it is possible that this will lead us to conclude Q , even if we do not believe R . This definition of causality has an important consequence for the qualification problem that we will discuss below in Section 8.2.3.1.

We will next define a causal schema to capture the notion that changes do not occur without a cause. We do not need to use explicit start and stop labels to indicate when statements are believed; the counterfactual engine keeps an internal record of the intervals during which statements are true. A statement d begins to be believed when b is asserted into a possible world; d stops being believed when $\neg d$ is asserted into the possible world.

If we consider $c_1, c_2, \dots, c_n$ as sets of sufficient conditions to believe d , we can define C as the set of all conditions that could cause d . We can write C as

$$C = c_1 \vee c_2 \vee \dots \vee c_n$$

and we have that:

$$d \ \& \ \neg C \implies \text{inconsistent}$$

unless d was given in the initial world.

Similarly, we can assume that there are sets of conditions c_1' , c_2' , . . . c_n' , each of which is sufficient to cause $\neg d$ to be believed. We can write C' as:

$$C' = c_1' \vee c_2' \vee \dots \vee c_n'$$

and we have that:

$$\neg d \ \& \ \neg C' \implies \text{inconsistent.}$$

Alternatively, we could write:

For all j , d_j there exists i such that $c_{i,j}$
where $c_{i,j}$ is the i th cause for d_j .

Note that the counterfactual engine satisfies this schema because a rule asserting d will not fire unless a cause c matches the left side statements. This shows that the counterfactual rules specify the causal relations between beliefs, and therefore the counterfactual engine fits the event logic. The counterfactual rules fire when an event occurs and cause necessary changes in beliefs; so we have an event logic that fits our assumption that every change in belief has a cause.

There are three properties that we want our causal schema to have. It should define a cause as a sufficient condition for a change to occur, it should fit an event logic describing how those changes take place, and it should not allow controversion [Simon and Rescher, 1977]. Our definition of a cause satisfies the first property, and we have shown that the counterfactual engine fits the causal schema shown above and thus satisfies the definition of causality. We have also shown that the counterfactual engine fits an event logic to satisfy the

second property. We explained in Chapter Three how we satisfy the third property by using the $\rightarrow$ operator instead of the $\Rightarrow$ operator. Thus, our counterfactual arrow is a function mapping causes to their consequences. This agrees with the idea that causal relations are functions [Simon and Rescher, 1977].

We have shown that our system satisfies the three desirable properties of a causal schema. Since the event logic describes the semantics of the causality arrow ($\Rightarrow$) and since the counterfactual arrow ($\rightarrow$) satisfies the event logic, the two arrows specify the same function. Thus, the counterfactual rules specify the changes to an agent's model of the world that are caused by physical and mental events.

8.2. The Frame Problem

The frame problem [Hayes, 1973] deals with how to represent changes effectively that occur in the world. When an event occurs, we must be able to calculate and represent the changes that it causes. In addition, we must somehow represent that those statements that have not changed have maintained their truth values. The representation chosen to solve the frame problem is a key factor in determining how successful a system is, i.e., it affects the ability to represent complex ideas and the efficiency with which changes may be computed and stored.

Previously, the frame problem has been studied with respect to systems that model the world rather than beliefs about the world. Since

changes in belief are necessary in our system and since we consider mental events as events in the world, the frame problem becomes an important issue for our system; we must be able to characterize the way we represent changes in our beliefs. Hence, our analysis of the frame problem specifically considers the subproblems with respect to changing beliefs.

Gardenfors [Gardenfors, 1988] mentions that a solution to the frame problem describes "a form of the world . . . that would enable us to translate efficiently our knowledge and beliefs about the world into a computer program." He then discusses Doyle's Truth Maintenance System and claims that it is not a good solution to the frame problem, but he fails to discuss the frame problem in any detail. We believe that ordinary reasoning is an event, and so we provide the first detailed analysis of the frame problem relative to mental events and changes in beliefs. We will discuss the frame problem as it relates to our system of beliefs, and we will explain how our representation solves the components of the frame problem.

8.2.1. Theoretical Solution

We will first describe the special aspects of our system that distinguish it from other systems. After that, we will describe how our theory is translated into an internal representation for a computer. Finally, we will examine how our system handles each subproblem of the frame problem.

We have described our model of time and event logic at the

beginning of this chapter. With this framework, our system appears to work like systems that deal directly with the world, even though we are reasoning with beliefs. There are two aspects of our system that contribute greatly to its usefulness. These are the ability to reason with time, and the ability to remember previous beliefs.

Both abilities are related to the representation our system uses. Our event logic is based on two assumptions: 1) events occur at instants and 2) truth values persist over intervals until changed by some event. Since possible worlds correspond to instants of time, an interval corresponds to a path through one or more possible worlds along a timeline.

Thus, statements become true in some possible world and remain true in later possible worlds until they are explicitly retracted. Since these possible worlds correspond to instants of time, they are each effectively time stamped. The statements are also timestamped by virtue of having a true value in a possible world.

Also, since statements are assumed to persist, only changes in truth values need be recorded in later possible worlds. In other words, we save space because we do not need to copy a statement into later worlds; its interval is assumed to be open to the future until specifically closed.

The combination of timestamps and intervals over possible worlds allows us to think about time and to remember previous sets of beliefs. Patrick Hayes [Hayes 1973] says there are actually two related aspects of reasoning about time. These are:

- a) there must be beliefs about time.

b) the robot lives in time.

To which we add two conditions:

c) the robot's thought processes take time,

d) thoughts are a part of the world.

The timestamps on possible worlds and statements allow us to reason about time and temporal relations. They also enable an agent to maintain some idea of the passing of time. For example, an event causes changes in belief that may in turn be events that cause other changes. Each change results in a new possible world at a later instant. If an event causes only minor revisions to beliefs, the system will quickly reach a steady state and no more changes will take place until another event occurs. If, on the other hand, the first event triggers a long chain of changes, then many more worlds will be generated before reaching that steady state.

This allows an agent to get a relative idea of the effect that events have on his beliefs. The more worlds that are generated, the greater the scope of the changes, and the longer it takes him to compute all of the changes. Thus, the system actually accounts for the time that it takes to reason and this time depends on the complexity and scope of the reasoning involved.

Our implementation using the counterfactual engine provides a solution to the frame problem that includes an agent's thoughts as part of the world, and it also represents the time taken by an agent's reasoning.

8.2.2. Internal Representation

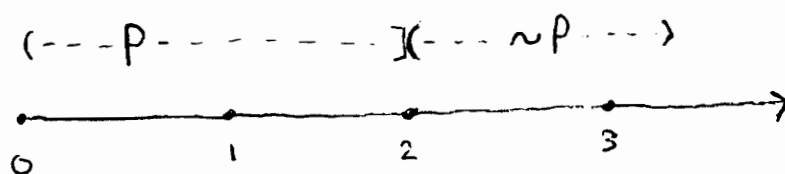
We will briefly describe the internal representation of statements. It is obvious that the actual possible worlds cannot be represented directly in a computer. Instead, we store a list of our statements and use arguments to specify intervals. The interval is specified as a pair of integers representing the worlds in which the statement becomes true and when it becomes false. When a statement becomes true, the end of the interval is set to a large number to indicate that it is presently open to the right. If the statement becomes false, the end of the interval is reset to the appropriate value.

There is a one-to-one correspondence between the numbers used and the possible worlds. Thus, a statement is true in all possible worlds whose corresponding number falls within the interval defined for the statement. As mentioned before, statements do not need to be copied to other worlds; no operation is performed on statements that do not change. Only when a statement changes truth value is its representation changed. Even then, all that changes is the interval that defines when it is true. Thus, only one copy of any statement is ever stored in memory; all operations on the statement are performed by changing or examining its defined interval.

Figure 15a shows how the truth value of a statement changes in time, and Figure 15b shows how this is represented internally.

f 194

(a)



(b)

user_fact (0 2) *name (P True) *mc
 user_fact (3 100) *name2 (P False) *mc

Figure 15. (a) Relation between instants, possible worlds and truth values (b) Internal representation

8.2.3. Components of the Frame Problem

We will now examine the frame problem by dividing it into three subproblems: the qualification problem, the ramification problem, and the persistence problem. An analysis for physical changes in the world using this type of division is given in [Pendergraft, 1988]. The qualification problem deals with limiting the scope of search and reasoning to relevant beliefs. The ramification problem deals with computing and storing all of the changes necessary when a belief is changed. Lastly, the persistence problem deals with representing the fact that all irrelevant beliefs have remained unchanged.

There is much in the literature concerning general analyses and solutions to these subproblems; our discussion does not propose a novel solution to the frame problem but instead applies the prior art specifically to belief revision.

8.2.3.1. The Qualification Problem

The qualification problem deals with limiting the effect of a change in belief to relevant beliefs. For example, suppose that John has beliefs about American history. If he learns that his former history professor has been arrested for driving while intoxicated, he is unlikely to revise his beliefs about history. If, on the other hand, his history professor was arrested for fraud and is not really a certified history professor, then John may doubt many of the historical facts he has been taught.

In the first case, it is unlikely that John would even consider changing his beliefs about history. It is possible, however, that John would mistrust a person who drives while intoxicated and might question what the professor had taught him. This shows that the scope of beliefs affected by another belief will depend on the rules relating beliefs.

It is almost certain that John would not change his beliefs about history if a former mathematics professor had been arrested. We do not want John to consider his history beliefs and consciously decide that he should keep them. Instead, he should never consider changing them because they are not relevant to the belief that does change. This is ensured by the way we write counterfactual rules and by the way they operate.

For example, we may have the statements:

TAUGHT John teacher history True set1

IS_SURE_OF history True set1

but adding the statement:

ARRESTED teacher DWI True set1

has no effect whatsoever on these beliefs.

When rules are written, it is under the assumption that beliefs affect only other beliefs that are closely related. In other words, we assume the principle of locality. This simplifies the writing of rules because the effects of one belief with respect to every other belief do not need to be considered. Roughly, this is similar to the idea of local causality where actions are concerned. Now, however, we are saying that the results of changing a belief affect only closely related beliefs.

We have all heard the expression "What does that have to do with the price of tea in China?" In general, the answer is usually nothing, so our system does not consider that the price of tea might have changed whenever it changes some other belief. This limitation of scope is due to the way the rules operate. Recall that a rule will not fire unless all of the left hand side statements are true. If there is not a specific rule relating the arrest of a history professor to the price of tea in China, then the professor's arrest will not cause John to consider the price of tea.

The ability to make only relevant inferences is important since we want our system to be consistent and predictable. If irrelevant inferences were allowed from arbitrary true propositions, then we could not predict which inferences would be asserted and when they would change. Propositions would be asserted at random, destroying the idea of causality in the system. Causality inherently requires relevance, and the manner in which we have structured our counterfactual rules ensures that only relevant inferences can be made. As Haack points out, however, the concept of relevance is difficult to formalize [Haack, 1978].

Although our system is semantic-based, we will attempt to understand relevance by examining a "relevance logic," the logic system E developed in [Anderson and Belnap, 1975]. We will show that the counterfactual arrow ($\rightarrow$) fits a relevance logic format.

Relevance logic is based on the idea that:

$A \supset B$

should only be valid if A is relevant to B. Emphasis is placed on the

distinction between deducing B under A and deducing B from A. B is deduced under A when the assumption of A does not prevent the deduction from being made but is not necessary for the deduction.

Anderson and Belnap say that it is not valid to claim that B is deduced from A if A is not used in the proof. Thus, their definition of validity requires that the antecedent be relevant. They go further and include that A must be necessary to the proof as well. This follows because if A were not necessary it could be eliminated without affecting the proof. Hence, Anderson and Belnap require that A be both relevant and necessary for deducing B if we wish to say that $A \rightarrow B$.

Under these criteria, Anderson and Belnap show that *modus ponens* is invalid in standard logics. They then develop their own logic based on relevant implication. The relevance aspect of implication is captured by the system they call R, while the aspect of necessity is captured by an S4 modal logic system. These two systems are combined to create the composite system E.

If our system corresponds to E then we have an inconsistency because E is an S4 modal system and we showed in Chapter Seven that our accessibility relations to correspond to S5 modal systems. Our system, however, does not correspond to the logic system E because our definition of causality states that causes are sufficient to believe the antecedent; the conditions are not necessary for the belief, if there is another set of causal conditions that are satisfied for the belief.

Thus, we do not require the S4 axioms for necessity, and we assume that we need only the axioms of R. The axioms of R are [Anderson and

Belnap, 1975]:

R1. $A \supset A$	identity
R2. $A \supset B \supset ((B \supset C) \supset (A \supset C))$	transitivity
R3. $(A \supset (B \supset C)) \supset (B \supset (A \supset C))$	permutation
R4. $(A \supset (A \supset B)) \supset (A \supset B)$	contraction

We will show below that the operation of the counterfactual engine follows these principles; Martins and Shapiro showed that relevance logic can be used to solve the frame problem [Martins and Shapiro, 1988]. For this analysis, it is important to remember that the counterfactual rules do not operate by strict implication. A belief on the right side of a rule is not always asserted into a possible world when a rule fires; the relative modal categories of the right and left side statements may cause a belief on the left side to be retracted to resolve an inconsistency. The counterfactual rules do, however, specify which beliefs can be inferred, i.e., which possible worlds can be generated. The rules are structured so that only relevant beliefs can be derived. Thus, the counterfactual rules act as a limit on which inferences are possible, and it limits valid inferences to those that are relevant to a counterfactual.

The identity axiom is satisfied since any statement $s(t)$ that is true at t remains true until some other event changes it. If $s(t)$ is true then $\neg s(t)$ will cause a contradiction. Transitivity also holds because the counterfactual engine continues to fire rules until no more can be fired. Assume that we have $A \supset B$ and $B \supset C$. When A is asserted, B is inferred and becomes a new event. B is then asserted as a counterfactual and C is inferred. Hence, whenever A is asserted we

will eventually infer C , so that $A \supset C$ follows from $A \supset B$ and $B \supset C$.

Our system also obeys the axiom of permutation. We can rewrite $(A (B \supset C))$ as $(A \& B) \supset C$. Likewise, we can rewrite $(B \supset (A \supset C))$ as $(B \& A) \supset C$. These two statements correspond respectively to the counterfactual rules:

(A
B $\rightarrow$ C)

and

(B
A $\rightarrow$ C)

but these two rules are the same counterfactual rule because the order of statements on the left side is unimportant. Thus, we obey the axiom of permutation.

Finally, our system obeys the axiom of contraction. We can rewrite $(A \supset (A \supset B))$ as $(A \& A) \supset B$. The redundancy of A is not necessary because if A is asserted once it can match both occurrences of A . Hence, we simply require $A \supset B$.

Thus, we have shown that the operation of the counterfactual engine is a relevance logic. It is not the logic system E of Anderson and Belnap because we do not require the antecedent to be necessary for the consequence. Instead, we use a theory of causality that requires the antecedent to be sufficient for accepting the consequent. Our relevance logic then corresponds to an $S5$ modal logic system and not an $S4$ modal logic system.

Since our counterfactual rules satisfy the axioms of R , it is reasonable that our system makes only relevant inferences. Intuitively,

it seems obvious that this is the case because we write our rules specifically to identify a causal, relevant relationship between statements. Since a rule cannot fire and deduce an inference unless the left-hand side is satisfied, inferences will always be based on the relevant statements of the left-hand side of a rule.

The specific counterfactual rules, therefore, enable the system to consider all and only those beliefs relevant to a subject. Irrelevant beliefs are not considered. This means that the system spends the minimum time necessary in searching for other beliefs that may change. There may be indirect relationships between beliefs that must be taken into account, but these are handled by the solution to the ramification problem.

8.2.3.2. The Ramification Problem

The ramification problem involves determining changes and representing them. Our system consists of possible worlds that contain belief sets. When a change occurs in a belief set, a new possible world is created to represent the revised belief set. This new world represents an agent's beliefs at some later time. Thus, worlds represent not only changes in beliefs, but also the passage of time as an agent reasons.

As mentioned, indirect relationships will be determined even though rules are written concerning only directly related beliefs. This is because the results of the rules are transitive; the right side of one rule is asserted as a counterfactual that matches the left side of

another rule and causes it to fire. Thus, every effect of a change will eventually be determined.

As an example, consider the arrest of John's history professor for driving while intoxicated. We mentioned that this could cause John to revise his beliefs about history, but it would not be a direct result. For this to occur, there would probably need to be two rules as follows:

- 1) If someone drives while intoxicated, then John does not trust them.
- 2) If John does not trust someone, then he does not believe what they have told him.

Thus, these two rules together may permit John to change his beliefs about history that the professor taught him if the the professor is arrested for DWI.

The following rules added to our example would cause the change in belief:

```
(ARRESTED *person DWI True *set ->
      MISTRUST *person True *set)
(MISTRUST *person True *set
      TAUGHT *person2 *person *subject True *set ->
      IS_SURE_OF *subject False *set)
```

This discussion shows that our system will efficiently compute all the changes necessary using the counterfactual rules and the transitivity necessary in a relevance logic. The other part of the ramification problem involves the representation of changes. Our system's representation makes the representation of changes simple.

True statements are represented by their presence in the system. If a change is made, then the statement is retracted. This is represented by a NOT prefix to indicate that the statement was believed but is no longer. In addition, the system keeps track of the worlds in which a statement is true. Thus, for any given world, the system can quickly determine if the statement is true by checking the interval that defines when that statement is believed true. The ability to examine the interval of a statement is part of the solution to the persistence problem.

8.2.3.3. The Persistence Problem

The persistence problem deals with efficiently representing that beliefs have not changed. We need to show that all irrelevant beliefs that have not changed have persisted into a new world. Yet, we do not wish to copy these beliefs into the new world or even acknowledge them since they are not relevant to changes in belief.

The solution to this is possible because the counterfactual engine keeps track of the interval during which a statement is true. In other words, the worlds in which a statement becomes true or false are stored; since these worlds correspond to instants of time, there is a record of the time interval during which the statement is true. The interval begins when a statement is asserted as true and is closed when the statement is retracted. Thus, statements are assumed to persist until they are explicitly retracted. If the system needs to determine if a given statement is true, it checks the current world's time index

to see if it falls within the interval defining when that statement is true.

Clearly, this is very efficient because we need not even mention beliefs unless they change. Otherwise, they are assumed to remain unchanged. Also, it is important to note that beliefs are not copied many times as new worlds are created; only one copy of a statement appears in the system, and its interval changes to indicate that it has started or stopped being true. Furthermore, this scheme has the advantage that statements are never erased from the knowledge base. When they are no longer true in a world, this is indicated by closing the interval. The statement is still present, however, so that an agent can remember that it used to hold the statement as true and in which particular worlds it was true. So, our system is monotonic because beliefs are never deleted from the counterfactual engine, even though an agent may stop believing them.

We must point out, however, that there is a serious problem with this solution; the problem is that this solution assumes that all states will tend to remain unchanged over time. There are many examples where this is clearly not the case (the weather, a person's age, a person's height, etc.) While it is reasonable to assume that an object will not move (unless an agent moves it), it is definitely not reasonable to assume that the weather in Blacksburg will remain sunny all day, even if it is sunny when we enter a building in the morning.

We will accept this limitation and hope that the types of situations that we model do not require unreasonable assumptions of persistence.

8.3. Conclusions

Our system operates directly on the possible worlds to compute changes and generate new possible worlds. The representation scheme and counterfactual rules enable us to consider only changes that are relevant to a belief. Yet, because the rules can be applied transitively, we are assured of determining all relevant changes.

The use of internal intervals gives us many advantages. When a new belief is added to a world, the statement is added to the knowledge base and the statement's interval is opened. The statement is assumed to persist in future possible worlds until it is retracted, at which time the interval is closed.

The system is capable of determining precisely which statements are true in which possible worlds by using the stored intervals along with timestamps. Thus, the representation of our many worlds in the computer occupies a minimum amount of space, but we must sacrifice some time when computing and representing changes and determining persistence. [Pendergraft, 1988] discusses the trade-off between the efficient use of space and time. The most important factor in the solution to the frame problem, however, is the representation. A powerful representation leads to efficiency in space and time.

9. Substitution of Equality

A class of problems that should be addressed is the substitution of equalities, or referential transparency. This is a problem that has plagued both philosophers and AI researchers [McCarthy, 1985a]; it concerns the validity of replacing one description with another that denotes the same object. In a complex system, such as the real world, there may be many different labels that can be used to refer to an object or concept. In some cases these labels may be interchanged without confusion; in other cases, using the wrong label in the wrong situation can create confusion or lead to incorrect assumptions. It is therefore important to define when the system may make substitutions and which substitutions may be made.

As an example, suppose we have sentence (1) below:

(1) The Lincoln Memorial is in Washington, D.C.

Since we know that Washington is the capital of the United States, it seems valid to have sentence (2) as well:

(2) The Lincoln Memorial is in the capital of the
United States.

Now, however, assume that a foreign visitor named Boris is confronted with the same two statements. Boris does not realize that Washington, D.C. is the capital of our country. Hence, the two statements are not equivalent from Boris' point of view. In the first case, we are dealing with true facts, so the substitution is valid. In Boris' case, on the other hand, we are dealing with propositions within Boris' set of beliefs, and the substitution fails to make sense.

This example points out that whether a substitution can be made depends on the context in which it occurs. The context is important because some contexts refer to concepts about objects while other contexts refer to the objects themselves. If a system cannot handle this problem, then there is clearly some weakness inherent in the chosen representation. In this chapter we will give the definition of equality that we use for concepts in our system. We will then explain how our system handles substitution of equalities and the theory behind how it works.

9.1. Representations

To attempt a solution to the problem of substitution of equalities, we must first have a model of how mental representations of objects relate to the objects themselves. There are several competing theories that suggest ways that mental objects correspond to the actual world. Frege developed a theory based on sense and reference; the sense of a word or phrase consists of the way it refers to an object while the reference is the physical object that a sense denotes [Frege, 1892]. Barwise and Perry, on the other hand, proposed a theory based on situations [Barwise, 1983]. They suggest that people use mental representations to classify abstract situations corresponding to the real world.

Each of these theories contains useful ideas, but neither seems complete by itself. The idea of sense and reference is the older and more established of the two; it has inspired several attempts at a

solution. McCarthy [McCarthy, 1985a] uses the terms 'sense' and 'denotation' to refer to the concept of an object and the object itself, respectively. He says that in direct contexts we intend a name to refer to the denotation; in opaque contexts we intend the name to refer to the sense or concept itself. This is exactly what Frege proposed using the terms 'sense' and 'reference' in [Frege, 1892].

An example that McCarthy gives involves the concept of Mary's telephone number and the actual digits that make up the number. He says that an example of an opaque context is a statement such as "John knows Mary's telephone number." This statement refers to the concept of a telephone number. A direct context, on the other hand, would be a statement such as "John dialed Mary's telephone number." This statement refers to the actual digits of the number.

The distinction becomes important when we try to substitute equalities. Suppose that Bill's telephone number is the same as Mary's telephone number. We can substitute this in a direct context to get, "John dialed Bill's telephone number." This is true since the actual digits are the same, even though John may not know that he dialed Bill's telephone number. It is not true, however, to say that "John knows Bill's telephone number" because John may not be aware that the two numbers are the same. The problem in the latter case is that the context is referring to the concept of a person's telephone number.

In the second case, the act of knowing something is an "attitude report." McCarthy explains that attitude reports such as knows, believes, sees, wants, etc. are examples of opaque contexts. Direct contexts are identified by predicates such as dialed; these predicates

indicate an action or attribute of the actual object that does not depend on a person's opinion or attitude about the object. McCarthy's solution to the problem is to treat senses and denotations as distinct objects with separate names [McCarthy, 1985b]. He proposes a scheme to specify precisely whether a name refers to the concept or the actual object.

Although McCarthy's idea leads to some possible solutions, it is based on first order logic. For this reason it cannot directly be applied to our system of possible worlds. The general principle, however, can in effect be applied to our system.

9.2. Our Approach to Substitution

We will now describe how our system handles substitutions and why we feel this is an appropriate solution for our purposes. The validity of a substitution obviously depends on whether the substitution affects the truth value of a statement. Ordinarily, we would expect a statement to have a true value if it corresponded to the actual world. In our system, since we are dealing only with beliefs, we have no way to check a statement against the actual world. Instead, since we are reasoning with perceptions* of and beliefs about the world, we assign a statement a true value if it corresponds to a belief held by an individual. This means that a statement's truth value is a function of both the belief and the individual whose beliefs are inspected. A statement with a certain truth value for the actual world may have

* See Chapter Ten for a discussion of perception.

different truth values for different agents' sets of beliefs.

The theories we have mentioned are concerned with how truth values are assigned to propositions that represent information about the world. This problem is easily solved for our system because we need only compare a statement to an agent's set of beliefs. If the statement matches one of the agent's beliefs, then the statement is true. It is tempting then to say that substitution is never allowed; an exact match cannot occur if the agent's belief uses a different label for a concept than the label we wish to substitute.

Consider McCarthy's example with the telephone numbers. Suppose we query to see whether John knows Bill's telephone number. This will not match exactly with the statement representing that John knows Mary's telephone number, so it is not true. Note that we are not referring to the truth value of a proposition, as in the previous section. We are now referring to the truth value of a statement rather than a proposition. Recall that the explicit truth value that appears in a statement is associated with the concept proposition to indicate whether the belief is positive or negative. Statements have only implicit truth values; if a belief is held then the statement representing it is true and appears in a possible world. If an agent does not have that belief the statement is false and does not appear in the possible world.

Our solution to the problem, then, is to assume that all propositions are contained in an opaque context. This means that substitutions cannot take place automatically; they may only occur when the agent is aware that the substitution can be made. We will discuss

which substitutions an agent can make below.

It makes sense not to allow transparent contexts when we recall that we are reasoning with beliefs, and those beliefs consist of concept propositions. Thus, we do not refer directly to physical objects or relations; instead, we are reasoning with concepts that cannot be automatically replaced.

The impact of this is that every statement is embedded in an attitude report, so that every concept necessarily falls within an opaque context. Consider the example of dialling the telephone numbers, which we previously said was a direct context. In our system of beliefs, however, the statements should be interpreted differently.

There are actually three states involved when dialling. These concern the belief about the telephone number, the intent to dial the number, and the act of dialling. The first state is what is handled when we say that "John knows Mary's telephone number." The third state, the act of dialling, is not explicitly represented because we are concerned with beliefs. When an agent believes that he dialed a number, he believes that he intended to dial a particular number and believes that he actually dialed that number. Since we have not developed a theory of intentionality, we will allow a single statement to represent both the intention and the belief that the action was performed.

If, however, we wish to represent the actual act of dialling, then we must use a separate predicate (e.g. `PERFORM DIAL`). Even then, this statement cannot guarantee that the action is performed successfully; it can only indicate that the agent attempts the action. Some other

process must accept the attempt, record that the action is performed in the world, determine if it is successful, and record changes in the world that it causes. The simulator that we discuss in Chapter Twelve represents an agent's actions with a statement of the form:

ACT *action *agent *time

The agent can determine the results of the action by perceiving changes in the world, e.g. he hears the telephone begin to ring. Thus, beliefs about actions are distinct from attempts to perform actions.

The statement "John dialed Mary's telephone number" should then be interpreted as meaning that John intended to dial Mary's telephone number and believes that he actually did. Thus, we can no longer substitute the phrase "Bill's telephone number" for the phrase "Mary's telephone number" in predicates concerning the act of dialling. Even if the digits that John dialed were those of Bill's telephone number, John neither intended to dial Bill's telephone number nor does he believe that he did. Hence, contexts that were previously thought to be direct become opaque in our system.

9.2.1. Our Definition of Equality

In general, we wish to disallow substitutions from automatically taking place within opaque contexts. We stated above, however, that we will allow substitutions if an agent is aware that they can be made. This requires the agent to have some definition of equality that can be applied to concepts. It is important to define the type of equality we desire. As Olsen [Olsen, 1987] points out, there are different types of

equality. Two things may be considered equal if they denote the same object. Alternatively, two concepts may be considered equal only if they share every property or convey the same information about what they denote.

A classic example of a problem involving substitution of equalities is as follows:

a) I saw the morning star at 7 a.m.

b) I saw the evening star at 7 p.m.

Since 'the morning star' and 'the evening star' both denote the planet Venus, we might substitute 'the evening star' for 'the morning star' in a) to get:

c) I saw the evening star at 7 a.m.

This is problematic because an agent may not be aware that the two senses denote the same object.

The example of the morning star and the evening star tells us that we cannot equate two concepts simply because they denote the same object. 'The morning star' and 'Venus' both denote the object "Venus" but cannot be interchanged in every context. Thus, a more careful definition of equality is needed.

If we say that two concepts share every property, then we are saying they are equivalent under Leibniz's law. Formally, this says:

$$\forall x, y: ((x=y) \supset (\forall \phi: \phi x \equiv \phi y))$$

This definition is too strong; concepts that share every property are identical, and there is no reason to make a substitution.

9.2.1.1. Our Solution

One problem that must be solved is how to define the problem of substitution of equalities. A formal approach to the problem could use a first order modal logic with equality; a logic such as this is well defined, but the issue of equality becomes clouded when we move to the domain of beliefs. The problem is that beliefs are not first order so the formal logic does not fit to beliefs precisely.

We do not believe that a formal, logical approach will provide a satisfactory solution. A practical solution requires that we consider the meaning and purpose of a substitution in beliefs; whether a substitution is made depends on many factors, including the individual agent and the culture in which he lives. Our solution allows substitutions to depend on the individual agent.

We define equality to mean that two concepts are equal if they denote objects that are identical at the level of resolution used to describe them. The concepts themselves, however, cannot be identical; concepts must have some differences or there would be no substitution. The goal of substitution is for an agent to gain information through the differences in the substituted concepts.

Many concepts do not denote merely objects; they denote objects having specific properties at some time and location. Hence, concepts are sometimes shorthand representations for propositions and correspond to situations. For example 'the morning star' means 'the heavenly object that is visible in the sky at morning' which we might write as

VISIBLE object sky mornings

Concepts represent a class of objects, and complex concepts represent

the class of objects formed by the intersection of other classes. Thus, concepts will be equal if they represent the same class (and that class may be the intersection of other classes).

These differences result from the different senses or connotations associated with the concepts. Not all differences are of the same type. Some differences result from different uses of what is denoted. This is the case when Mary's telephone number and Bill's telephone number denote the same number; the concepts identify two distinct uses for that number. Another case is when the difference depends on the customs of different people. Both 'hello' and 'howdy' denote a greeting, but the two may be used by people living in different locations; the difference in which is used depends on the individual and the situation.

Clearly, substitution depends greatly on the individual and his culture. We have also mentioned a dependency on level of resolution in our definition of equality. By 'level of resolution' we mean the degree of granularity used to describe a situation. The situation denoted by a concept can be described by a list of objects, relations, and attributes. This list may have only one element or may have many; the level and amount of detail describing the situation in this list determines the level of resolution. Normally, we might have:

```
denotes(pencil,("pencil"))
```

but it would be possible to have:

```
denotes(pencil,("wooden cylinder","graphite core","pointed end"))
```

This is a finer resolution that we generally do not need because the class of objects that are pencils are usually well known; the finer

resolution describes the class of pencils as the intersection of those classes of objects defined by the list of attributes.

The level of resolution of what a concept denotes may differ between cultures and individuals; the substitutions that a given agent makes will depend on that agent's own denotations of concepts. The resolution of a situation is defined by the information in the list used to describe the denotation of a concept. The properties that will most often concern us will be the object denoted, the time associated with what is denoted, and the implied location of what is denoted. We have not developed a theory about how to set the granularity for describing a particular situation, but it depends on the agent's purpose for using the concept that denotes that situation.

9.2.1.2. Comparing Situations

We must be clear about how to determine whether concepts map to the same situation; this will depend on cultural or individual conventions and not some universal principle. Thus, our solution is a pragmatic, epistemic approach rather than a purely logical approach. Since everything in our ontology can correspond to a situation, all of our entities fit our theory and may be potentially substituted. This means concepts may also be equal. For example, the number six and a half dozen may be substituted for each other. Heaven and Utopia might also be considered equal by some individuals.

We must be wary to remember that we require concepts to convey the same information (and for the agent to be aware that they do). Concepts

must only imply the same time or location if they imply any time or location at all; not all concepts imply anything more about a situation than the objects denoted. If time is not specifically implied by a concept, then it cannot be implied by another if they are equal. What this means is that two concepts must denote situations at the same level of resolution and must include the same type of information in order to be equal. The equality of concepts can be determined by comparing the lists that describe the situations denoted by those concepts; if the lists have every element in common, then the situations denoted are identical and the concepts are equal.

It is important to realize what information a concept implies. The planet "Venus" certainly exists in time and changes its location in an orbit. When we use the concept 'Venus', however, we normally refer only to the planet and not to its particular location at a particular time. This is not true of 'the morning star', which associates a particular time with "Venus"; the resolution of 'the morning star' is finer than the resolution of 'Venus'. It is also true that the surface of the planet may change and that it is not the same morning every day; if this were important then a finer granularity could be used.

Another important question is exactly what object (or concept) is referred to in a situation. Montague discusses individual concepts and gives the following example [Montague, 1973]:

The temperature is ninety.

The temperature rises.

Therefore,

Ninety rises.

The problem here is that the concepts of temperature and ninety are not being equated; instead, the two concepts are grouped to form a new, single concept. This concept is of a particular valued temperature. When this temperature rises, the value of the concept changes, but we cannot separate it into its individual components and apply the change to each. The general concept of a temperature does not change, and neither does the concept of ninety; only the specific temperature of ninety changes.

The same thing occurs with combinations and safes. A combination by itself is a general concept and safe1 may indicate a particular safe; when we say 'combination safe1' we group the two together and obtain the concept of the combination of a particular safe. So, we must be careful of what concepts or objects are being referred to in a situation.

9.2.1.3. Examples of Our Solution

We can now give an example of how our theory of substitution is implemented. Assume that John knows Mary's phone number and Bill's phone number are the same, but he does not know what it is. We can represent this as:

Statements in the knowledge base

(i) denotes (PHONE_NUMBER Mary) ("x") True set1

(ii) SET JOHN nil nil set1

(iii) denotes (PHONE_NUMBER Bill) ("x") True set1

We also have a counterfactual rule that says if an agent believes two

things have the same value and he knows the value for one then he knows the value for the other.

Counterfactual Rule

*PREDICATE *value True *set

denotes (*PREDICATE) (*x) True *set

denotes (*PREDICATE_2) (*x) True *set ->

*PREDICATE_2 *value True *set

Statements (i) and (iii) indicate John is aware that Mary's phone number equals Bill's telephone number (they denote the same number). If John learns that

PHONE_NUMBER Mary 555-5555 True set1

the counterfactual rule enables us to make the substitution that John knows the value of Bill's telephone number as well:

(iv) PHONE_NUMBER Bill 555-5555 True set1

This illustrates the idea of how substitutions can be made. There is considerable flexibility in the form the statements may take, as long as the statements in the counterfactual rule have the same form. Note that PHONE_NUMBER Mary is a single concept and fits our schema for substitution.

We can simplify our notation with a rule that says:

denotes (x) (*X) True *set

denotes (y) (*X) True *set -> EQUALS x y True *set

Substitutions can occur if an agent believes two concepts are equal because he believes their denotations are the same. By using this definition of equality our system avoids unwanted substitutions.

We can illustrate this for the case of the morning star and the

evening star. In our system, we would represent the relevant denotations as:

denotes (Venus) ("Venus") True *set

denotes (the morning star) ("Venus","visible in mornings") True *set

denotes (the evening star) ("Venus","visible in evenings") True *set

Here we say the concept of the morning star represents the class of objects in space and time that are both Venus and are visible in mornings. None of these denotations are identical so none of the concepts are equal.

With this implementation, substitutions will be made only if the agent is aware of the equality. Whether two objects are equal in the actual world is not necessarily a factor. This also allows agents to make mistakes, as humans sometimes do. For instance, John may correctly believe that Bill is Mary's brother, as in:

denotes (Mary's brother) ("Bill") True set1

but he may also incorrectly believe that Bill is the mailman:

denotes (mailman) ("Bill") True set1

Thus, he can substitute Mary's brother for mailman even though this is not the case. This shows, therefore, that the substitutions an agent permits depend only on what concepts he believes are equal, regardless of what objects are equal in the actual world.

9.2.2. Comparison to Other Solutions

We will now compare our solution to the solution proposed by Maida and Shapiro [Maida and Shapiro, 1985] and the solution proposed by Gat

[Gat, 1987]. A system containing only opaque contexts was first proposed by Maida and Shapiro. Although they are concerned with defining the nodes in a propositional semantic network, many of their results can be applied to our possible worlds approach.

Maida and Shapiro have a special structure in their semantic networks that allows them to represent that two concepts, or 'intensions' as they refer to them, have the same denotation, which they refer to as the extension. This convention fails to take into account that two concepts may contain different information about the same object. The classic example of this is the case of "the morning star" and "the evening star." Even though each phrase implies additional information about the time that the object is seen, they both denote the planet Venus.

This means that Maida and Shapiro's network represents an indirect relation between "the morning star" and "the evening star" via two EQUIV connections. On Maida and Shapiro's interpretation of the network this means only that the two concepts denote the same object, but it presents prima facie evidence that the two concepts are equivalent; to understand the network correctly, a person must know that the EQUIV relation is not transitive. This apparent equivalence requires that there be exceptions to the rules of their network to prevent undesired substitutions such as:

(3) John saw the morning star at 7 p.m.

which may be incorrect if John is not aware that the same planet is denoted by both phrases.

Erann Gat attempts a solution involving only extensions and

relations [Gat, 1987]. By using distinct representations for concepts and their denotations and using relations rather than functions, many different relationships can be expressed and problems can be avoided that arise when a function's value must be manipulated. Relations have the same advantage in our system since we have distinct representations for concepts and denotations; we can describe relations between them, whereas a function does not yield an object that can be manipulated well in relations.

Gat associates each extension with a label; extensions are equated when they have the same label. Although Gat's system does not really solve the substitution problem, his idea of using only relations works in our system. The difference appears because we use intensions in our system, whereas Gat's approach is purely extensional. His system of labels would not work well for us, because we are using intensions. An example with the morning star and evening star problem will illustrate the problem.

In Gat's system, we can consider "the morning star" and "the evening star" to be different labels indicating Venus. If we allow substitutions to take place from the label to what it represents, then we will make undesirable substitutions.

For example, we could substitute 'Venus' for 'the morning star' in:

(4) I saw the morning star at 7 a.m.

and get the following:

(5) I saw Venus at 7 a.m.

If we then substituted 'the evening star' for 'Venus' we would end up

with

(6) I saw the evening star at 7 a.m.

(7 a.m. and evening are inconsistent and the belief would not make sense if the agent were not aware of the equality).

As shown in section 9.2.1.3., our system does not make this invalid substitution. To work like Maida and Shapiro's system, ours would have to equate THE MORNING STAR and THE EVENING STAR, but the denotations are not identical. To work like Gat's, it would have to equate VENUS with both THE MORNING STAR and THE EVENING STAR, but these denotations are not even at the same level of resolution. Hence, the substitution will never be made.

9.3. Uses of Substitutions

There are at least two cases where substitutions are needed. One is when ordinary reasoning occurs (e.g., knowing Bill and Mary's telephone numbers), and another is when communication occurs. We will now discuss each of these. Note that we are categorizing substitutions by the type of situation involved.

9.3.1. Substitutions in Derived Beliefs

Since explicit knowledge about equalities makes substitutions possible, our system requires a rule to handle such cases. This could be implemented so that the system would check for substitutions when

attempting to match statements to an agent's set of beliefs. Matching may be done more efficiently, though, if we implement a rule that checks for and derives substitutions when information is derived about an object. Statements can then be matched directly during a later search.

Assume that a statement concerning concept A enters the system, either as direct input or the result of another rule firing. The system checks its list of EQUALS statements to determine if another concept may be substituted for concept A. Assume that the agent who believes the statement about concept A also believes that concept A equals concept B. The statements would cause a counterfactual rule to fire, and the statement concerning concept A would be duplicated except that concept B would be substituted for concept A. When a query about concept B is made later, the system can directly match the statement in the knowledge base.

We will characterize the substitution as follows:

Let $p(x)$ denote a proposition concerning x and let A and B denote concepts that refer to objects in some situation.

Then, $p(A)$ represents a proposition about A . Also, let $EQUALS(A,B)$ denote the proposition that concepts A and B are equal.

We represent a statement as $(x, *set)$ where x is a proposition and $*set$ denotes the belief set in which the proposition is held. Our schema for substitution then becomes:

$$(p(A), set1) \ \& \ (EQUALS(A,B), set1) \rightarrow (p(B), set1).$$

In other words, if an agent believes a proposition about some concept (e.g., the value of PHONE_NUMBER Mary), and the same agent

believes that the concept is equal to a second concept (PHONE_NUMBER Bill), then the agent should believe the same proposition about the second concept (the value of PHONE_NUMBER Bill is the same as the value of PHONE_NUMBER Mary).

The actual implementation in our system was illustrated by the telephone number example in section 9.2.1.3. Note that there is no explicit representation of the timing context in our statements. Statements are timestamped according to the possible superworld in which they occur; different superworlds coincide with different times. Thus, timing of statements can be reasoned with by examining the timestamps of superworlds, but the timestamp does not occur explicitly in the statements.

9.3.2. Substitutions in Communication

We need substitutions in communication because other agents may use different concepts than we do. If we know what that concept denotes for the other agent, we may be able to substitute our own concept that denotes the same thing. For example, suppose there is a particular room in a house, "Room1". If another agent refers to this as the study, we may not understand if we do not use the concept of STUDY ourselves. If we believe that

denotes(study,"room1",set2)

about the other agent's beliefs, and if

denotes(den,"room1",set1)

then we will understand that the other agent is referring to room1,

(the den to us).

Using the previous notation, we can formalize this substitution as the following schema:

$$p(A) \ \& \ \text{denotes}(A, "A", \text{set2}) \ \& \ \text{denotes}(B, "A", \text{set1}) \\ (p(B), \text{set1})$$

where $\text{denotes}(X, "X", \text{set})$ is shorthand for $\text{denotes}(X, "X") \text{ True set}$. Note that we are using a rule for equality that applies to denotations from different sets of beliefs. This rule says

$$\text{denotes}(A, "A",) \text{ True } * \text{set2}$$

$$\text{denotes}(B, "A",) \text{ True } * \text{set1} \rightarrow \text{EQUALS } A \ B \text{ True } * \text{set1}$$

We can now make substitutions in communication to aid us in matching the concepts of messages to our own concepts.

9.4. Variables

Gat [Gat, 1987] also discusses problems with quantifiers, such as when we know that something exists but we do not know what it is. Although we do not have quantifiers in our system, his idea of labels helps us when we wish to indicate with some variable that something exists, even though we do not know the value that the variable assumes. A label is precisely the sort of variable we can use because a label is really just a nametag. If we wish to refer to some unknown quantity, we can simply call it X . We used this technique in our example of Mary and Bill's telephone numbers. We used " X " to represent the telephone number until we learned that it was "555-5555".

Note that the variables we are speaking of are formal variables or

unknowns that we are representing in our system; they should not be confused with counterfactual variables. In other words, we do not allow a counterfactual statement to contain an uninstantiated variable *x. Instead, we instantiate the counterfactual engine variable *x with some value such as x. Even though x may have no meaning to us, the counterfactual engine can continue to operate.

The representation we are using seems, therefore, to solve problems with substitution of equalities and also to allow us to refer distinctly to concepts and denotations. Thus, by using variables such as X we may refer to telephone numbers (or safe combinations) without knowing their values. For example, COMBINATION SAFE1 X refers to the particular string of digits that represent the combination of SAFE1. Alternatively, we may simply use COMBINATION SAFE1 to refer to the combination of this particular safe.

Another problem we might encounter, though, is the extent to which substitutions should be made when they are valid. Maida and Shapiro [Maida and Shapiro, 1985] discuss a process called merging that can occur when two concepts are found to be equivalent. This involves transferring all information about one concept to its equal, and then predominantly using only one concept to refer to the object. They cite psychological evidence suggesting that merging does not occur in opaque contexts. We will also adopt this convention since we have already determined that our system operates only within opaque contexts. Also, the built-in operation of the counterfactual engine searches for direct matches. If we merged information about equal concepts, then we would also have to look for statements that matched when a valid substitution

was made.

The main problem that Maida and Shapiro's system encounters is "memory confusion". They do not mean memory of previous beliefs or the order in which beliefs become believed. Instead, they are referring to the problems that occur when their system allows properties of one node to be inherited by another node that should not really do so. A system that does not keep track of time, however, will be subject to memory confusion because it will not know the order in which events occur.

For example, if a question is asked before the answer has been discovered, an agent will not be able to give the correct answer. It becomes important, then, to keep track of the order in time in which the answer is discovered and the question is asked. Since our system keeps track of previous possible worlds, and since it uses a stricter definition of equality, we can avoid both problems.

Consider the following partial example from [Maida and Shapiro, 1985]: Pat knows Mike's telephone number, and Ed dials Mike's telephone number.

It would be undesirable to infer that Pat knows that Ed dials Mike's telephone number. This type of problem cannot occur in our system since we do not use nodes or connections between them. Thus, we have the distinct statements:

```
SET JOHN nil nil set1
```

```
SET ED nil nil set2
```

```
TELEPHONE_NUMBER Mike Mike's_phone_number True set1
```

```
DIALS Ed Mike's_phone_number True set2
```

where Mike's_phone_number is simply a variable to us; we could use X

instead but that would not indicate what the variable represents. From these statements, there is no way to mistakenly derive:

DIALS Ed Mike's_phone_number True set1.

Furthermore, since we create new worlds at different times as information changes, our system keeps track of the order of changes. This means that our system will give the answer to a question that is correct for the circumstances at the time it is asked, in so far as our system can determine with its information at that time; the answer may change with additional information. Our system, therefore, does not suffer from any problems with memory confusion.

9.5. Additional problems

There are, in addition, two other problems we will address. These occur when either time is involved or when we encounter a context that seems transparent even though it is embedded within an opaque context

As an example of a problem when time is involved, consider the statements:

I visited the capital of the United States.

I visited Washington, D.C.

If an agent is aware that Washington is the capital of the United States, then this is a valid substitution. Suppose, however, that we have the following two modified statements:

I visited the capital of the United States in 1790.

I visited Washington, D.C. in 1790.

Here, a substitution is no longer valid because the capital of the

United States in 1790 was Philadelphia. Thus, it is important that a concept be evaluated with respect to the time of its context when determining if it is equivalent to another concept.

If an agent has sufficient knowledge of history, then he will be aware that there is no equality between Washington in 1790 and the capital of the U.S. in 1790. If the agent is unaware of the significance of the date, then an invalid substitution may be made. This is not a defect in the representation scheme; it is simply a result of incomplete knowledge of the agent. Humans make the same kind of mistakes.

An example of the second problem we wish to address is the statement:

I launched a rocket to the morning star at 7 p.m.

If this statement occurs in our system, then it is embedded in an opaque context because this is an "attitude report" about what we believe occurred. At first, there again seems to be a problem because we refer to the morning star at 7 p.m. In this case, however, we have not made a substitution and must assume that the agent making the statement is aware of the meaning of the concept 'the morning star' and does not see a problem with launching a rocket to it in the evening. This example implies that there may be some contexts that are transparent even when they are embedded within an opaque context.

There are several possible reasons why this is so. We may simply understand that the 7 p.m. refers to the time of the launch, and we do not consider the time relevant to the morning star. Alternatively, we may consider the act of launching to be a process including the travel

time to get to "the morning star," and the exact time of launch does not really matter because the rocket will not actually be at "the morning star" at 7 p.m. When we refer to seeing an object, on the other hand, the time of our statement is simultaneous with the time we see the object.

In any case, the important distinction we discover is that we do not use the phrase "the morning star" to indicate Venus at a particular time; it is not morning when we launch the rocket, and we should not assume that it will be morning when the rocket arrives at Venus. Instead, we are referring only to a particular location. There is really no reason to say "the morning star" instead of simply saying Venus. Thus, it conveys no additional information in this context but may sound more poetic.

This problem, then, really becomes a problem of natural language. It is not our intention to solve such problems here. If such a solution were necessary, however, it might be possible to define a category of relations that could maintain transparency even when embedded in an opaque context.

9.6. Conclusions

Our approach of working with opaque contexts seems well-suited to our system and handles the problems we would expect to encounter in reasoning with beliefs. Our examples show that the results should hold for attitudes such as knowing and seeing as well. In any case, statements will always be embedded in the opaque context of a belief

because of our system's design.

We can represent a variety of relations between concepts and the objects they denote. At the same time, we have close control over what substitutions can be made because a substitution will only be made in the opaque context if the agent is aware of the equality. Furthermore, the agent will be aware of only those equalities where the concepts have a common denotation.

10. Perception, Communication, and Action

Multiple agents must be able to interact with the world and with each other. We have distinguished three abilities that are necessary for an agent to interact intelligently. These abilities are:

- 1) The ability to perceive the world and develop new beliefs as a result of sensory stimuli.
- 2) The ability to communicate information to another agent or to receive information from another agent.
- 3) The ability to perform actions that will change the world.

Clearly, there are interrelationships between these abilities. For example, communication involves the perception of written or oral statements. Also, an agent performing an action must be able to perceive the effects of that action to determine if he has been successful. We will first discuss these abilities separately, and then we will integrate them when we discuss the relation between action and belief.

10.1. Perception

Since we will treat communication between agents separately, our discussion of perception will relate only to visual information; auditory information will be considered a part of inter-agent communication. Although our discussion of perception relates to vision, what we describe is a general scheme for inputting information from the world into an agent's beliefs. In other words, the input is

causally related to the real world. An agent should perceive new or changing objects, relations, and attributes and should integrate the information into his beliefs.

The process of perception is a complex issue; we do not have an exact model describing how light images on the retina are transformed into beliefs about the world. Imagine, for example, a person standing in a dimly lit room and staring at a wall. Gradually he becomes aware of the distinct shades of grey and is able to distinguish outlines of objects. Then he realizes there is a doorway in the wall, and there is a darker blur in the middle of the doorway. This blur he recognizes as a person, but he does not know who it is. The person in the doorway appears tall, and since John was the only other person in this house, the conclusion is that John is standing in the doorway. Then, the person moves closer until more details can be seen, and it turns out that it was Bill standing in the doorway.

We see from this example that perception involves many factors; the conclusions drawn depended on the image seen, the perceiver's beliefs about individual attributes (John is tall; image is tall), and some informal logic about possibilities (John is believed to be the only other person here, this image is here, so the image must be John). Somehow, the perceiver transformed the stimulus image of variegated shadows into the final, correct belief that Bill was standing there. Even this is a simplification because we have not discussed how Bill's actions and attributes were perceived (e.g. is he entering the room or merely standing in the doorway and watching).

We will not attempt to explain the process of perception; instead,

we will concern ourselves with some principles relating what is perceived to what is true in the actual world.

10.1.1. Rules of Perception

Much work has been done in the area of perception because it is of fundamental importance to agents interacting with their world. Most of the work, however, has concentrated on relating what an agent sees to what is true in the world. When we say 'sees' we are referring to the raw image data that enters the agent's visual system, independent of how the data is interpreted or perceived. Since we are dealing with beliefs in our system, our concern is to relate what an agent thinks he sees to what actually happens. From an agent's point of view, he can never be sure of what he actually sees (i.e., the precise visual image that enters his eyes) because his perception of that image is biased by his other beliefs. In other words, light enables an image of the structure of the physical world to enter an agent's eyes as raw data, but the agent must interpret that image to perceive objects, relations, and attributes of the physical world.

We will consider perception to be the process of developing beliefs about an image that is seen. Thus, what is seen are images with color, intensity, etc. while what is perceived are objects, relations, and attributes. Perception is influenced by emotions, prior beliefs, and state of mind. For example, Barwise and Perry give the following example [Barwise and Perry, 1983]:

Ralph saw a spy hiding a letter under a rock.

From an omniscient view, we may wish to make such a statement, but it really does not pertain to an agent's beliefs. It may be true that Ralph saw the spy (i.e., the image of the spy entered his eyes), but he may not have perceived the spy or his actions; Ralph may have been gazing absently into space as people sometimes do.

From our view of perception, we might have the following statements:

- (1) John saw a spy hiding a letter and John perceived a spy hiding a letter.
- (2) John saw a spy tying his shoe and John perceived a spy hiding a letter.

Alternatively, we might have statement (2) and the following statement:

- (3) John saw a spy tying his shoe and John perceived a spy tying his shoe.

Statement (2) may result from John's suspicious nature and overactive imagination. These examples show that agents do not always perceive what they see, and they sometimes imagine that they perceive what they have not seen.

Barwise and Perry give examples of the distinction between seeing and seeing that [Barwise and Perry, 1983]. They state that "seeing" is a relation between a person and a scene while "seeing that" is a relation between a person and a situation. Consider the two statements:

- (4) John saw a spy hiding a letter.
- (5) John saw that a spy was hiding a letter.

It appears that (5) is intended to indicate perception, but it is clear that this is incorrect when Barwise and Perry argue that the Principle of Veridicality holds for "sees that."

Their veridicality principle says:

If p sees that s , then s .

We have already explained why

If p perceives s , then s

does not hold. For our purposes, however, it would contradict our intuition to say that whatever we perceive has actually occurred. Hence, perception cannot be the same as "sees that."

We can also determine whether Barwise and Perry's other principles hold for our theory of perception. The Principle of Substitution states [Barwise, 1989]:

If a sees $s(t_1)$ and $t_1=t_2$, then a sees $s(t_2)$.

For us, this principle would be:

If a perceives $s(t_1)$ and $t_1 = t_2$, then a perceives $s(t_2)$.

This principle relates to the problem of substitution of equalities we discussed at length in Chapter Nine. In this form it is not valid; we can make it valid by modifying it as follows:

If a perceives $s(t_1)$ and a believes that $t_1=t_2$, a perceives $s(t_2)$.

Actually, it is not precisely correct to say that a perceives $s(t_2)$; what happens is that a perceives either $s(t_1)$ or $s(t_2)$ and then makes a substitution to create the second belief. Since the result is the same either way, and since our system makes the substitution immediately if a believes that $t_1=t_2$, we will consider the modified principle of

substitution as valid.

If we substitute "perceives" for "sees" in the following principles from [Barwise and Perry, 1983], we can easily see that they also hold. We have included examples to help clarify the principles. The examples of existential generalization and negation are taken directly from [Barwise and Perry, 1983].

Existential Generalization: If b sees s (the z), then there is something₁ such that b sees $s(it_1)$.

Example: If Helen saw the chairman of the department answer the phone, then there is someone whom Helen saw answer the phone.

Negation: If b sees $\neg s$, then b does not see s .

Example: James saw someone not leave the room. So, James did not see everyone leave the room.

Conjunction distribution: If b sees s and w , then b sees s and b sees w .

Example: Ralph saw Mary open the door and leave the room. So, Ralph saw Mary open the door and Ralph saw Mary leave the room.

There are two more principles, however, that do not hold for perceives.

The first is Disjunction distribution:

If b sees s or w , then b sees s or b sees w .

Consider the example:

John perceives Mary laughing or crying.

This implies that Mary is exhibiting some type of behavior that John believes is laughing or crying, but he is not certain which. It would make little sense to say:

John perceives Mary laughing or John perceives Mary crying.
Since John cannot distinguish what she is doing, he certainly cannot perceive either action specifically.

The final principle that does not apply is the principle of Distribution of Indefinite Descriptions, which says:

If b sees $s(z)$, then there is a z_1 such that b sees $s(it_1)$.

Barwise and Perry give the following example [Barwise and Perry, 1983]:

If Hans saw a tall, thin man climb in the window of 333

Miramonte, then there was a tall, thin man whom Hans saw climb
in the window.

This cannot be valid for perception because we have already stated that not everything an agent perceives actually occurs. Hans may perceive that a tall, thin man climbed in the window, but actually a tall, thin woman climbed in the window; there was no tall, thin man.

We have seen that our view of perception follows some of the principles of "sees" and "sees that"; overall, however, it is very different from Barwise and Perry's theory. Lakoff has proposed a cognitive theory [Lakoff, 1988] that differs greatly from Barwise and Perry's. Our theory of perception resembles Lakoff's in that it depends on the cognitive perception of the agent rather than the objective situation in the world.

Lakoff bases his theory on what he calls idealized cognitive models (ICM's). He argues that humans have these ICM's in their minds and use them to categorize what they see. With these general models, an agent can also distinguish when and why something fails to agree with the ICM. Our theory of perception also involves categorization of

images into beliefs, but it is less strict than Lakoff's theory.

Perception for us categorizes images of the actual world into propositions that are used to generate an agent's beliefs about the world. Thus, the images are mapped into object concepts, relation concepts, and attribute concepts that are combined to form many propositions that we believe describe a scene in the world. This mapping is less general for us than for Lakoff because we are mapping to concepts rather than idealized models. Thus, we would not perceive a penguin as a bird that is far removed from the prototypical bird. Instead, we perceive a penguin as a penguin, but our concept of a penguin includes the fact concepts that it is a bird but that it has special qualities for a bird (e.g., it does not fly).

Lakoff's theory differs from ours a great deal because it too deals with seeing rather than perceiving. He presents three statements as his ICM of seeing:

1. You see things as they are.
2. You are aware of what you see.
3. You see what's in front of your eyes.

These do not translate to perception very well because they can be used to derive Barwise and Perry's principles (veridicality, substitution, etc.), some of which we distinguished above as being invalid for our view of perception. For example, Lakoff's first statement is not valid because we do not always perceive things as they are. The second is valid because we must be aware of what we perceive since perception is a cognitive process that requires awareness. The third is not valid because, as we mentioned earlier, people sometimes gaze into space

without perceiving what is in front of them. The third statement is valid in that it implies that we do not perceive things that are not in front of our eyes.

Many of the problems that Lakoff raises can be handled by our theory simply because we are focused on perception rather than vision. For example, he gives the following example:

- (a) The princess saw the frog jump into bed with her.
- (b) The frog was really a prince.
- (c) The princess saw the prince jump into bed with her.

The problem is whether (c) can be concluded from (a) and (b). This is an example involving the principle of substitution. Since we are concerned with perception, then the princess could not have perceived that the prince jumped into bed with her because he looked like a frog at the time. If the princess believed that the frog was a prince, then she would have perceived that the prince jumped into bed with her. On the other hand, if the princess did not believe the frog was a prince until after he had jumped into bed with her, then she would have realized later that the prince had jumped into bed with her, but she would not have perceived it at the time.

Lakoff also discusses an example where two lights flashed rapidly in succession are perceived as a single moving light. Again, since we are dealing with perception, we care only that the observer perceives a single moving light; it does not matter from his point of view that the image he saw was of two separate lights.

Thus, we have discussed our view of perception and the principles that it follows. Images are mapped into concepts by the process of

perception, but we have not attempted to define how this complex mapping occurs. Such an analysis would require a complex model detailing how we recognize objects, relations, and attributes from images and how our emotions and beliefs affect that recognition.

Since our world will be represented in terms of propositions, we will assume that an agent can accurately perceive the world. We must therefore focus on the problem of determining what an agent sees and which things he actually perceives.

Our model will assume that an agent sees the things that do not agree with his mental model of the world. An agent has a set of beliefs about the world that constitute his model of the room; when he enters a room, his model of the room (its contents, whether doors are open or closed) are compared to the actual room. Differences between the two become the things that an agent sees. If those differences include supplemental information (things the agent perceives that he did not realize were there), then the agent's model of the world, and of the room in particular, is extended. If the differences indicate that something is missing that the agent expected to see, then his model is revised.

We may not want an agent to see differences that are trivial or irrelevant to the agent's goals. We can model this by assigning modal categories to the differences and allowing the agent to see only those things that are of sufficiently great importance. Thus, we effectively filter out insignificant changes that an agent would not notice. For example, an agent looking for a broom might not notice a fly buzzing around a room if the broom were there. If the room was empty except

for the fly, then the agent probably would see it. If the room were on fire, the agent probably would not see the broom even he were looking for it.

Thus, an agent will see things that he did not expect to see, and he will perceive them if they are significant, i.e., if they are not overshadowed by more important objects or events that are present. When he perceives these objects, the propositions about their locations are introduced into the system as counterfactuals. The counterfactual engine then revises the agent's beliefs to reflect that he now believes that those objects are in this location and not in some other location.

If the differences between a room and an agent's model of the room indicate that something is missing from the room that the agent thought was there, then we may want the agent to perceive that the object is missing. Again, these differences can be filtered using modal categories to ignore minor details (does the agent really care that the pesky fly is gone?). The important missing objects, however, are not automatically perceived. Instead, we use a goal-directed approach to detect that objects are missing.

This goal-directed approach will allow an agent to recognize that objects are missing if they are relevant to a goal. For example, if an agent believes a broom is in the kitchen and he goes to the kitchen to look for it, then he will realize that it is not there (if it is not in the kitchen). In this case, the goal of the robot is to hold a broom and that broom is one of the differences between the kitchen and the agent's model of the kitchen. The proposition that the broom is not in the kitchen becomes a counterfactual in the system, and the robot's

beliefs change to reflect that the broom is not in the kitchen.

In summary, an agent will perceive an object that he did not expect to see in a room if it is significant compared to the other differences between the actual room and the agent's model of the room. An agent will perceive that an object is missing if it is significant and if it is relevant to his goals. These perceptions are introduced as counterfactuals that cause revisions to the agent's beliefs.

Perceptions can also lead to new goals for an agent. If an agent has a goal X and he perceives a condition Y that prevents X , then the agent will have a new goal of $\neg Y$. For example, suppose that an agent wants to move through a doorway into another room. If the agent perceives that the door is closed, his new goal will be for the door to be open. If the agent tries to open the door and perceives that it is locked, his new goal will be for the door to be unlocked. This combination of perceptions and goals to generate new goals enables an agent to alter plans when necessary; if an agent had planned to walk through a doorway and discovers that the door is closed, he amends his plan to first include opening the door. Thus, perceptions can influence an agent's actions as well as influencing his beliefs.

We have referred mainly to the presence or absence of objects in our examples. We should point out that the differences between an actual room and an agent's model of the room can include things such as the status of doors (a door that was closed is now open) and the relative positions of objects (John was standing next to the door, but is now next to the window). It should also be noted that this is a sensing (hearing, feeling, etc.) model as well as a seeing model. The

emphasis of the model lies in determining when information should be interchanged and not interpretation of the information. In this case, the interchange of information is between the world and an agent's beliefs, via perception. We will see that our communication model is similar to the perception model in this regard.

10.2. Communication Model

Agents must have the ability to communicate with each other if they are to interact. Communication can become a very complex issue due to the subtleties of natural language. We will not attempt to incorporate natural language into our model; we will assume that agents will communicate only well-defined beliefs that will not be subject to misunderstanding. Essentially, agents will be communicating via propositions. Thus, the emphasis is again on information interchange rather than the interpretation of information. In this case, the interchange is between the beliefs of two agents.

10.2.1. Structures

We develop a model of the structures associated with agents' beliefs that we may use to model communication. The development of this model parallels the communication model developed by Virkar [Virkar, forthcoming] except that our model is simplified since we are not concerned with natural language, and we do not attempt to model metacommunication concepts, such as intonation or volume of an

utterance.

As we noted in the preceding section, our agents communicate through propositions. Thus, the primitive entities in our model will be the same as those with which we reason in our system: object concepts, relation concepts, and attribute concepts. When a statement is made by an agent, he is trying to communicate information about concepts that may or may not map to the physical world. We will call this agent the sender. He communicates via labels that represent concepts for him.

A second agent, whom we will call the perceiver, attempts to correlate the transmitted labels to concepts in his own mind and thereby understand the meaning of the statement. By meaning, we refer to the information concerning relations and attributes of particular objects. Thus, we will see that communication depends a great deal on overlapping sets that have common designators (labels) for certain concepts. Since we use distinct possible worlds to model agents' beliefs, we are saying that we must have rigid designators that are the same in all worlds; that is, they represent the same concepts for every agent. Our model must account for the different concepts and labels that agents may have, and the way that communication and understanding depend on these concepts and their labels.

There are three major components in our model: the perceiver (P), the sender (S), and the content of a statement (C). The perceiver is of fundamental importance because he must try to match the labels communicated by the sender to concepts. For example, if the sender uses a label that the perceiver cannot match to one of his concepts,

then the communication fails. Every statement will have some meaning m . Following [Virkar, forthcoming], we will say that the perceiver has received the statement u if the structure of P is satisfied by m . In other words, P must be able to find the reference of every concept label in the statement u . We refer to the communicated information as a statement rather than an utterance since it reflects a concept proposition rather than a natural language statement. Recall that we are communicating with words that are labels referring to concepts.

When we say that the Perceiver P has received a statement, we mean that he has been able to match every label in the statement with a concept. This does not mean that P has received the statement intended by the Sender S because the two agents may map the same label to different concepts. For example, S may refer to a person named John. The perceiver will understand the concept of John, but he may not understand which person named John is being referred to.

Another example occurs when S uses a foreign word that P is not familiar with. P may recognize that the word is a label from a foreign language, and he will be able to perceive the statement, but he will not truly understand the statement. Even if he understands the gist of the statement from the context, he does not truly understand the intended meaning. This is because he attempts to substitute labels that he understands (from his own language) for the ones he does not. There are many cases, though, where foreign words may not have a precise translation in another language.

It is possible, however, that P will come to understand the meaning of the foreign word and will use it as a label for that

concept. Essentially, P has then extended the structure of his beliefs to include a new label for a concept; he may even use the foreign word himself thereafter. We will discuss later why it is important that the structure of P be extendible.

The second component in our model of communication is the structure of S, the Sender. This structure includes all of the concepts that S has knowledge of, including object concepts, relation concepts, attribute concepts, and abstract situations composed from them. A statement is formed by S using the labels that he associates with these concepts. Since an agent cannot have complete knowledge of all the concepts and combinations of concepts in the world, the structures of P and S are partial structures that do not contain complete information.

The third component is the structure of the content. In theory, this structure contains all of the concepts and combinations that are possible in the world. Since this would be an infinite set, it cannot be implemented in our system. Instead, we assume that all the statements used will belong to this set.

The structures associated with P and S are both subsets of the structure associated with C. Furthermore, the structure of C defines which combinations of object concepts, relation concepts, and attribute concepts are valid. For example, it should not be valid to make an statement "A is ON B and B is ON A" because we cannot conceive of this relation. Granted, the content is clear, but the meaning is not.

We will next describe the structures we have referred to based on the following definition which is similar to the definition of a

many-sorted language given in [Monk, 1976]: we have a non-empty set S of sorts that correspond to our primitives, i.e., object concepts, relation concepts, and attribute concepts. For each s in S we allow variables $v_0^s, v_1^s, \dots$ that range over the set A_s of individuals associated with sort s . For example, the set A_{s_0} would be the set of object concepts, and the variables that range over A_{s_0} would be all the labels used to refer to those object concepts.

Since our system is based on possible worlds, our model differs from the one presented in [Virkar, forthcoming]. Since the relations themselves are primitives, we do not have relations operating on objects. Instead, we have groups of concepts that associate relation concepts with groups of object concepts and attribute concepts. The functions contained in the structures then map these groups of primitives into propositions.

10.2.1.1. Structure for P

We will refer to the structure associated with P as U_P and define it as follows:

If S is the set of sorts, then

$U_P = (G_P, L_P, R_P)$ where

(i) G_P is a function that assigns to each sort s a non-empty set A_s . Thus, this determines the concepts that the perceiver is familiar with.

(ii) L_P is a function that assigns variables $v_0^s, v_1^s, \dots$ to each individual A_s for each sort s . These, then, are the labels by

which the perceiver is able to refer to a concept.

(iii) R_p is a function that associates individuals A_s from the different sorts to form propositions. The propositions are then referred to by the variables v_i 's associated with the individuals used in the propositions. If a relation concept sort has a given rank, then there must be that number of individuals of an object concept sort associated with the relation concept by R_p .

We can use these predicates to specify what meanings can be perceived by P . If m is the meaning of the statement u , then

U_p supports m iff

m_e	$\text{range}(A_p)$
m_o	$\text{range}(A_p)$
m_r	$\text{range}(A_p)$
m_a	$\text{range}(A_p)$
m	R_p

where m_e , m_o , m_r , m_a are respectively the event, object, relation, and attribute components of the meaning m . We are therefore saying that P can perceive m if and only if all the components of m are contained in the structure of U_p and the meaning m is a valid function of those components. This is how we determine whether or not information is interchanged, similar to the way model differences and goals were used in the perception model to determine if information transferred from the world to an agent's beliefs.

In other words, P must be able to match all of the concepts referred to by the labels contained in m , and P must also recognize the relation between those concepts as valid. For example, P would not

perceive the meaning of the statement

ON SIDEWALK BIG

because the function R_p should associate the relation concept ON with two object concepts.

10.2.1.2. Structure for S

We will refer to the structure associated with S as U_s and define it as follows:

If S is the set of sorts, then

$U_s = (G_s, L_s, R_s)$ where

(i) G_s is a function that assigns to each sort s a non-empty set A_s . Thus, this determines the concepts that the sender is familiar with.

(ii) L_s is a function that assigns variables $v_{0s}, v_{1s}, \dots$ to each individual A'_s for each sort s . These, then, are the labels by which the sender is able to refer to a concept.

(iii) R_s is a function that associates individuals A'_s from the different sorts to form propositions. The propositions are then referred to by the variable v_i 's associated with the individuals used in the propositions. If a relation concept sort has a given rank, then there must be that number of individuals of an object concept sort associated with the relation concept by R_p .

We can use these predicates to specify what meanings can be sent by the sender. If m is the meaning of the statement u , then

U_s supports m iff

m_e	range(A's)
m_o	range(A's)
m_r	range(A's)
m_a	range(A's)
m	R_s

where m_e , m_o , m_r , m_a are respectively the event, object, relation, and attribute components of the meaning m . Thus, we are saying that S cannot communicate any meaning concerning concepts with which he is unfamiliar or using labels that he does not know are associated with a concept. Furthermore, we assume that he will only communicate meanings that are legitimate combinations of relations, objects, and attributes. That is, we will not allow him to communicate a statement such as

ON sidewalk big.

Similarly, we can define the structure for the content C . The important difference is that G_c contains all concepts in the world, L_c assigns all possible labels to the appropriate concept, and R_c operates on all valid combinations of concepts that make sense. R_p and R_s operate only on combinations that make sense, but they do not contain all combinations that make sense because an agent could not be aware of all the possible propositions about all concepts.

10.2.2. Interactions Between Structures

Since we have not modelled metacommunication in our system, perception will depend only on the content of structures; if the structure of P contains the labels of the statement u and maps them to

the same concepts as S does, then P will perceive the message. This means that we do not allow misunderstandings in which a part of the statement is not understood; we do not have any way of negotiating over the meaning of the statement. Since we are communicating with statements that represent propositions, we do not need to worry about the ambiguities of natural language utterances that could cause a misunderstanding.

In our model, U_c defines the possible meanings that a statement can have, while U_s and U_p respectively define those meanings that the sender and receiver are able to use and comprehend. The meaning of a statement will be perceived by P only if the following conditions are met:

1. The meaning is a valid combination of concepts that could be found in the set U_c . This simply means that the sender does not communicate nonsense, such as the earlier example:
ON sidewalk big
2. The structures U_c and U_p must both contain the individuals contained in the meaning m . (U_s will necessarily contain these individuals because S generates the meaning.)
3. The perceiver must be able to associate the same individuals with the labels that the sender does.

The first condition ensures that only valid meanings are sent. The second condition requires that S and P have knowledge of the same concepts. For example, a scientist may have trouble communicating with a non-scientist. The third condition requires both S and P to associate the same concept with a common label. For example, people

who do not speak the same language cannot communicate (verbally) even if they understand the same concepts.

Even when the perceiver is successful in perceiving the meaning of an statement, it is not guaranteed that he understands the true meaning. Recall the example referring to a person named John; the sender and perceiver may each know several individuals named John. We can define true understanding to occur only when the perceiver associates labels with the same concepts that the sender intends to be associated with them.

Up supports true understanding of m iff

Us m and

m_{es} range(L_s)

m_{os} range(L_s)

m_{rs} range(L_s)

m_{as} range(L_s)

m R_s

Up m and

m_{ep} range(L_p)

m_{op} range(L_p)

m_{rp} range(L_p)

m_{ap} range(L_p)

m R_p

$m_{es} = m_{ep}$

$m_{os} = m_{op}$

$m_{rs} = m_{rp}$

$m_{as} = m_{ap}$

Obviously, an agent cannot tell whether or not he has understood the true or correct meaning because of his perspective. From an omniscient perspective, the model describes what is necessary for true understanding.

An interesting aspect of our system is that we have nested beliefs. This property could assist us in determining true meaning. For example, if the sender refers to a person named John, the perceiver must determine which John it is. If the perceiver knows more than one John, then he may inspect his beliefs about what the sender believes. If he finds that the sender also knows one of the Johns that P knows, then P might assume this common John is the one referred to. Of course, it is possible that the sender is referring to a John that the perceiver does not know. It is also possible that they both know more than one common John. Thus, there still may be confusion, but it may be prevented in some cases.

The practical issues involved in an implementation are not complex because we have simplified our model to avoid natural language statements. An agent will speak if he has a goal to request information from another agent or to give information (correct or incorrect) to another agent; the other precondition of a communication action is that the agent must have an established communication link.

A communication link requires that an agent either be in the same room as the agent he wishes to speak to or has some device (such as a telephone) for remote communication. In addition, when a device such as a telephone is used, there is no communication link unless the agents are actually using the phone; they cannot communicate just

because each has a telephone in their room unless they are using the telephones to talk to each other. A goal to speak can be part of plan to achieve some other goal (asking an agent for the combination to the safe) or it can be a response to a stimulus (responding to a request).

In our system an agent will hear what is said if he is in the same room as the speaker, and he will be able to distinguish the speaker and the intended listener. If an agent cannot map all of the words in the speech to familiar concepts, he can use his model of the speaker's beliefs to try to determine what physical things those concepts denote for the speaker; the listener may be able to use substitution of equalities to make sense of the speech. If not, the message will not be usable to the listener, but he may later recognize the words if the speaker uses them again.

Emotions and state of mind can also affect communication. An agent may not say something or may ignore something said to him if he feels a certain way. He may hear a request but decide not to respond. These are important factors, but we have not integrated them into our simulator.

10.2.3. Partiality of Structures

We defined the structure of C to contain all possible concepts, labels, and valid combinations of concepts. For our purposes we can consider this complete. On the other hand, both Up and Us are necessarily partial structures because our agents are not omniscient. We wish to examine the implications this partiality has for

understanding.

Since we are not concerned with natural language meanings, the work done on the semantics of partiality for modelling natural language meanings [Landman, 1988; Langholm, 1988] is not useful in our communication model. We are, however, concerned with the implications of structural partiality. Since we have two partial structures, U_p and U_s , in our model, they may overlap or they may be disjoint. Hopefully, they will overlap; if two individuals shared no common concepts or labels they could not communicate at all. This is why common knowledge and 'rigid designators' are important in communication.

We mentioned earlier the importance of being able to extend the structure of P . In our example, the perceiver learned a foreign word; that word became a new label to associate with a concept. Thus, originally L_s mapped this v into some A 's, but L_p did not map this v into anything, even though this A 's corresponded to a A_p . U_p was extended by adding this v to the structure, and the perceiver modified L_p to map this new v into the appropriate A_p .

Once U_p was extended by v , the perceiver was able to understand the meaning m intended by the sender. Hence, we see that true understanding requires that either U_p and U_s are elementarily the same or that U_p can be extended to U_s .

10.3. Action and Belief

The relationship between action and belief is important because 1) actions can affect beliefs and 2) certain beliefs may be necessary to

perform some actions. Agents acting in a world must be able to change their beliefs when actions alter the world. They also must consider the information they believe when considering possible actions.

Moore [Moore, 1985] has proposed a formal theory of knowledge and action in a first order system that uses the relations B and K for believing and knowing. He also gives examples of the two cases above. An example of when actions can affect beliefs occurs when an agent reads something from a piece of paper; the act of reading adds information to the agent's knowledge base. Actions will also result in changes in the world that may be perceived by an agent.

It is also common for the performance of an action to depend on the agent's beliefs. An example is when an agent wishes to move to some location; he must have some belief about where that location is in space in order to move there. John cannot walk to the store unless he knows where the store is. Another example occurs when an agent is opening a safe; the agent must have the combination in order to be successful. As Moore points out, the act of dialling the safe's combination is a physical action that most agents could perform; however, they must have knowledge of the correct combination before they can perform the action of opening the safe.

It is clear that the relationship between action and belief (or knowledge) is important. Here we will present a schema that will allow agents in our system to interact with the world.

We wish to have an event logic for actions in the physical world and will assume that the theory of causality holds in the world; changes in the world do not occur randomly but are caused by events.

These events are actions that are performed in the world. The event logic we will use is that developed by Turner [Turner, 1984] that we described in Chapter Eight. The difference is that an event e is now considered to be a physical event (action) rather than a mental event (change in belief). Thus, we have two types of events that we will denote as e_m , for a mental event, and e_p , for a physical event. A physical event causes changes in the actual world and a mental event causes changes in beliefs. Note that the changes in the world caused by an e_p may themselves trigger mental events, and mental events may cause physical events through agents' actions.

In general, an agent develops a plan to achieve some goal and then performs an action as part of that plan. The action causes changes in the world and the agent perceives these changes as mental events. The mental events cause changes in the agent's beliefs, and he must then reassess his plan based on his new beliefs. We must be able to model how actions may depend on beliefs and how beliefs are affected by actions.

We first address the problem of the safe combination; this is one of Moore's examples of how action depends on knowledge. The solution for this problem in our system is not complex because of the causal theory that we use for changes in belief. We assume that an agent believes that he must have the combination before he can open the safe*. We know from Chapter Five that we can represent that he does not know the combination. Thus, the agent will not try to open the safe

* What we are actually assuming is that the agent is aware of the search space size for the combination; he needs the combination because the chances are small that he would succeed randomly.

unless he has the combination.

The conditions that will cause the agent to dial the combination are that he wants the safe open and he has the combination. Hence, the agent's goals are a cause for dialling the combination, but having the necessary knowledge (the combination) is also part of the cause. We will assume that the agent is aware that he must have the combination and will obtain it; we will assume that the goal of obtaining the combination will be a response to the goal of wanting the safe open. The agent understands that to open a safe he must have the combination and must dial it; if he wants to safe open, then he will want to have the combination.

The counterfactual rule to implement the act of dialing would look like:

GOAL open safe True set1

HAS combination True set1

NEXTTO safe *person True set1 SET *person nil nil set1 ->

ACT DIAL combination *person

Note that the conclusion here is preceded by the predicate ACT; this means that the agent wants to perform this action in the world, not merely to think about it. The action must be sent to an interface that will interpret it, attempt to implement, and make any changes in the world (such as OPEN safe). The agent must then perceive changes in the world to determine the outcome of his action. If the ACT predicate were not present, then DIAL would be treated as a belief predicate and the agent would merely be thinking about the possible actions or consequences of an action. The ability to distinguish thinking about

actions from performing those actions is necessary for planning.

A counterfactual rule that could act as an interface to implement the action would be:

ACT DIAL combination safe *person

SET *person nil nil *set

HAS combination *person True world

NEXTTO safe *person True world -> OPEN safe True world

OPEN safe True *set

This illustrates how a physical event (the act of dialling the combination) can cause a physical change (the safe is open) and a mental change (the agent believes that the safe is open). These changes would not occur without the physical event (or some other event that could cause the change). Thus, the rule operates as the event logic specifies.

This example also shows how the frame problem is handled. The beliefs on the left side should not be affected by the action of dialing; they do not appear on the right side, and their truth values simply persist. The statements on the right side reflect changes (in the world and in an agent's beliefs); the counterfactual engine internally opens an interval that will indicate the statements are true until they are retracted. Also note that this rule will fire when all the left side conditions are satisfied, but it will never fire otherwise.

In general, physical events should only cause immediate changes in the physical world; perceptions of these changes become mental events that cause changes in an agent's beliefs. When the results of an

action are immediately apparent we will assume that the agent perceives them as a direct result of the action.

For example, an agent opening a door will recognize that he has opened it, while an agent throwing an object must wait until it lands to determine the results. It is clear that there is sometimes an interval between the start of an action and the determination of the results. When there is no significant interval, the perception of the results of the action will be assumed to be automatic. In the case of dialling a combination, it makes sense to insert the belief that the safe is open immediately into the agent's beliefs (if he succeeds); the agent is attempting to open the safe, is standing next to the safe, and would undoubtedly perceive that it is open (if he succeeds).

When the agent needs to obtain the combination of the safe, he might do so by reading it from a piece of paper [Moore, 1985]; he could also ask another agent for it, which leaves him open to being deceived by lies. This is an example of how an action (reading a paper) can affect beliefs (finding out the combination). This can be accomplished by rules that act as an interface between the actual world and the mental world of an agent's beliefs. For example, if an agent performs the ACT of reading the paper, then the system should alter his beliefs so that he has the information from the paper in his belief set. For example, a counterfactual rule that indicates the results of reading a paper containing a combination would look like this:

```
ACT READ paper True *person
HAS paper *person True world
SET *person nil nil set1
```

ON paper combination True world

COMBINATION safe 38-25-36 True world ->

HAS combination *person True set1

COMBINATION safe 38-25-36 True world

The system keeps track of the facts that John has the paper, the paper contains the combination, and the value of the combination. When John performs the action of reading the paper, the system allows the combination to enter his beliefs, along with the belief that he now has the combination.

We can define a general schema as follows: Let R be a set of results (changes in the world or in an agent's beliefs). These results are caused by an event e (physical or mental) in conjunction with a set of conditions C . We can then write that:

$$e \ \& \ C \implies R$$

or, in the form of a counterfactual rule:

$$(e \\ C \rightarrow R).$$

This is a simple model of the relation between actions and beliefs. It serves to illustrate that the specific relations between action and belief that are required by a problem domain can be specified by the counterfactual rules. Furthermore, we have also implemented a simulator (discussed in Chapter Twelve) using counterfactual rules.

It is clear that perception, communication, and action are interrelated. Communication and perception are themselves actions, and perception is needed for agents to understand communication. Agents

must be able to perceive changes in the world and the results of their own actions. All three are needed for agents to interact with their environment and with each other.

10.4. Conclusions

We have specified a model for perception of an agent, communication between agents, and a relation between action and belief. Each of these is necessary for agents acting intelligently in a world. These described models could be used as the basis for developing a completely simulated system involving multiple agents. A planner and goal recognizer would also be needed for an automated system.

11. Deception

In an adversarial situation, there is a possibility that an opponent may try to deceive us. We must therefore have the ability to reason about deception if we wish to avoid being deceived. The material we have developed in previous chapters is necessary for the ability to reason about deception. Deception requires the ability to use nested beliefs, hypothetical reasoning, uncertainty, imaginary concepts, and substitution of equalities.

It is clear that the material in the previous chapters provides a foundation to study deception. There are several aspects of deception that we now must consider. Any reasoning about deception requires that we be aware that deception is possible. This means that an agent must have knowledge concerning the possibility of deception and must have the capability of modifying his reasoning process when he believes it occurs.

An agent must be capable of determining when deception might be occurring, what information is involved in the deception, and what can be inferred if this information is known to be inaccurate. The first of these problems is difficult to solve because we do not want agents to become paranoid and assume that everyone is always deceiving them. This would increase the amount of memory and computation time needed by a robot and would impede its ability to function. On the other hand, agents should not be easily deceived. Thus, agents must have some reasoning capabilities that let them make good guesses about when deception might be involved.

If we want to allow agents to recognize when deception may be involved in a situation, we must have a clear idea of what deception is. All deception involves distortion of the truth in some manner; the theory of deception, however, is more complicated than this simple statement would indicate. J. Barton Bowyer [Bowyer, 1982] has precisely classified different forms of deception. He suggests that all deception involves either hiding the truth or showing some false situation as if it were the truth.

Hiding the truth may take the form of masking (to eliminate a pattern), repackaging (to modify a pattern), or dazzling (blurring a pattern). Showing a false situation can take the form of mimicking (to recreate an old pattern), inventing (to create a new pattern), or decoying (to give an alternative pattern). This theory is precisely developed but is too detailed to be implemented practically. It is more useful for analyzing deception after the fact.

An analysis of deception to aid in recognizing it as it occurs must focus on the methods people use for deceiving others. These methods usually involve lying, hiding objects, or misleading others through actions. Thus, we will devote our analysis to these methods of deception. Before attempting to recognize whether a specific form of deception is involved in a situation, we should determine if it is likely that an opponent has reason to deceive us.

11.1. Detecting Deception

A representation that allows agents to reason about nested beliefs

was a basic requirement for deception detection. The model theory, however, is also important because it allows us to reason with concept propositions that may or may not correspond to the physical world. An agent's own beliefs should correspond to the physical world, but deception involves concept propositions that purposely do not correspond to the actual world. Furthermore, deception may even involve imaginary concepts that do not exist physically. Thus, deception may involve false statements concerning physical objects or statements about objects that do not exist.

There are several clues that may tell us deception is involved. If we can determine that deception would be to an opponent's advantage, then it has a greater likelihood of occurring. Also, if what we expect to occur disagrees with what we observe, it may be an indication that we are being misled and should modify our beliefs. Hence, our agents should be able to examine their beliefs and reason about whether deception would be to an opponent's advantage. For example, if an opponent knows where an object is, wants the object for himself, and also knows that an agent is looking for that object, it would be to his advantage to direct the agent away from the object's location.

11.1.1. Detecting Conflicts

We will examine deception as if we ourselves (as humans) are trying to reason about it. We will then examine our own thought processes and try to simulate them in automated agents. Obviously, a person considered as an opponent will have general goals which differ

from our own. We do not, however, wish to assume that they are constantly deceiving us. This would clutter the reasoning process. We will therefore analyze our opponent's immediate goals individually to decide if we would consider deception as a possibility in that instance.

For example, if we are fighting a war, it would not be helpful to recognize merely that we want to win and they (our opponents) want to win. This type of conflict is too general to be of use. It may, however, be useful to know that we want our troops to occupy some particular island and our opponent also wishes to occupy that island. This illustrates why we must narrow the scope of goals when identifying conflict.

It is also important to note that there is not always a conflict when goals differ. If one person wishes to cross a bridge from east to west, and another person wishes to cross from west to east, then there is no conflict as long as the bridge is wide enough for the two people to pass each other. If, on the other hand, the bridge is only wide enough for one person, then the differing goals present a conflict (in particular, this is a resources conflict). Hence, we must determine if a given situation would cause a conflict (or resources conflict) between us and our opponent. These conflicts occur when we are competing to achieve mutually exclusive goals.

These conflicting goals can take many forms. Possession of objects is often a source of conflict. If we wish to possess some valuable gold and our opponent also wishes to possess it, there is a conflict since we cannot both possess it simultaneously. Perhaps, though, we do

not want to possess the gold ourselves, but we do not want him to possess it either. There is still a conflict because he wants to possess something and we want him not to possess it. A similar situation occurs when he wants to occupy some position and we want him not to be there.

An agent must therefore be capable of examining his beliefs and his model of his opponents' beliefs and determining whether there might be conflicting desires. For example, if an agent believes his opponent wants the gold, and he believes that the opponent believes that he wants the gold, then he can believe that his opponent is aware that there is competition for the gold. If, on the other hand, the agent wants the gold and believes the opponent wants the gold, then the agent believes there is competition for the gold, but his opponent may be unaware of the competition. Whenever an agent's immediate goals conflict with his opponent's goals, and the agent believes his opponent is aware of the conflict, there is reason for the agent to suspect that the opponent may try to deceive him.

Another factor relating to the possibility of deception is the source of information. For example, something we learn from a used car salesman is less credible than something we learn from a friend. An agent should keep track of the source and context of information as well. If he thinks someone may be lying to him, the speaker's credibility would help the agent decide whether to believe him or not.

The rules that determine what goals present a conflict may be domain specific and are written as counterfactual rules. General conflicts, such as two agents trying to possess a single object, can be

included whenever we wish to detect conflicts.

11.1.2. Determining the Relevant Information

When deception is involved, we must be able to determine what information we believe is incorrect. This involves hypothetical reasoning; if deception is suspected, we must determine the implications. We might be deceived about the location of an object or who possesses it. We might also be misled about the intended goal of an opponent, such as which room he is moving towards. In all cases, deception occurs through the actions and/or words of another agent. That is, an opponent must do or say something to deceive us. Otherwise, we are reasoning with accurate reflections of his beliefs, and any incorrect conclusions are simply mistaken inferences or perceptions we have made.

Since we are analyzing deception in terms of immediate goals, we will assume that deception will relate to the source of the immediate conflict. For example, suppose that we want a window to be open and an opponent wants the window to be closed. We will recognize that there is a conflict over the state of the window and that our opponent may attempt to deceive us. If our opponent then tells us that the window is open and the door is closed, we will question only his report about the window, since there is no conflict over the status of the door. If we believe he is deceiving us, we will guess that the window is actually closed.

Admittedly, it may not always be the case that deception is

concerned with the immediate source of conflict. Consider the case where an opponent recognizes the conflict over the status of the window. In order to discourage us from entering the room to check the window's status, he may tell us the door is locked. If the door actually is locked, he may tell us that he does not have the key, when in fact he does. In either case, our opponent uses deception about the door or key to aid indirectly his desired goal about the window. While such occurrences are recognized, we will restrict our first attempt at identifying deception to deal with deception related to the immediate cause of goal conflict.

We have not developed a detailed theory for goal recognition; that is not the subject of this thesis. The problems of deception and goal detection are interleaved, however, because deception detection depends on beliefs about agents' goals and goal detection requires an agent to consider the possibility of deception. We use a general schema to predict what goals or actions we expect from an opponent based on his other goals, but our system does not allow agents to predict those goals. Instead, we assume that goals may could be predicted if goal recognition were implemented. We simply input what we believe to be an opponent's general goals by hand, and we analyze them to determine if there is a conflict. We allow agents to identify conflicts using the following rule:

Agent X believes that agent Y believes that there is a conflict concerning y's goal, goal1, and x's goal, goal2, if the following conditions are satisfied:

- 1) goal1 and goal2 are mutually exclusive,

2) agent X believes that agent Y has goal1, and

3) agent X believes that agent Y believes that agent X has goal2.

We can write this as a counterfactual rule as:

(SET X nil nil set1

SET X Y nil set2

SET X Y X set3

GOAL Y goal1 set2

GOAL X goal2 set3

MUTUALLY_EXCLUSIVE goal1 goal2 -> CONFLICT X Y goal1 goal2 set2

CONFLICT X Y goal1 goal2 set1)

This definition only identifies conflicts that an opponent is aware of because those are the cases in which we must be wary of deception. If we are concerned with deceiving an opponent ourselves, then we only need to identify conflicts that we are aware of; it does not matter if our opponent knows there is a conflict. In fact, we can deceive our opponent more easily if he is unaware that there is a conflict between us. We can identify these conflicts with the rule:

Agent X is aware that his goal, goal1, conflicts with y's goal, goal2, if the following conditions are true:

1) goal1 and goal2 are mutually exclusive,

2) agent X has goal1, and

3) agent X believes agent Y has goal2.

We can write this as a counterfactual rule as:

(SET X nil nil set1

SET X Y nil set2

GOAL X goal1 set1

GOAL Y goal2 set2

MUTUALLY_EXCLUSIVE goal1 goal2 -> CONFLICT X Y goal1 goal2 set1)

11.1.3. Distinguishing Deception from Mistakes

The system must also be able to distinguish between deception and mistakes. Something is only deceptive if an opponent presents to us some information that he does not believe himself. If he gives us information that he believes to be true but which is actually false, it is merely a mistake on his part. Thus, the difference depends on what we believe he believes [Vasek, 1986].

The importance of the distinction between a mistake and a lie involves what we may infer from the false information. If an opponent lies about something, we might wish to assume that the opposite is true. For example, if we believe he is lying when he tells us to look for something to the north, we may decide to look for it to the south. At the very least, we may eliminate searching in the north. If our opponent simply made a mistake, on the other hand, he did not intend to mislead us and we can only infer that he does not know where the object is.

11.1.4. Detecting Lies

As mentioned earlier, an opponent may deceive us either through words or actions. Of these, words are the easier case to analyze. We will refer to any communication from another agent, whether written or

oral, as a report. Deception involving reports is simply lying, and can be analyzed by comparing our beliefs with a model of another agent's beliefs.

Our theory uses the idea of suspicion. We define suspicion as the state of mind that causes an agent to believe that a contradiction between perceptions and beliefs indicates deception. An agent becomes suspicious when he recognizes that he and another agent may both be aware of a conflict concerning their goals. If an agent is not suspicious of another agent, then there is no reason to think deception is taking place, i.e., the agent does not believe his opponent has motivation for lying. Recall that we assumed that deception has a motivation (to gain an advantage); there are no pathological liars in our system. Suspicion is represented by the presence of a belief that there is a conflict that an opponent is aware of.

First, we can check any report against our own view of the world. If his report agrees with our beliefs, then there is no reason to think deception is taking place. We will act on the assumption that our current beliefs are correct. In this case, even if he were lying, his lie agrees with what we actually believe. It follows that we would consider him mistaken about whatever he was lying about, which is advantageous for us. This is the easiest situation to handle.

If the report contradicts what we believe, we may again assume him to be mistaken, unless the support for our belief is very weak. If his report does not relate to anything in our belief set, it may be new information about the world we have gained through our opponent. In either case, we next match the report against our model of his belief

set.

If our opponent's report contradicts what we believe him to believe, there are several possible explanations for the contradiction. The first is simply that he is lying. If the report relates to something about which there is a conflict, we may be reinforced in believing this explanation. In other words, our opinion that he is lying is stronger if we are suspicious of this opponent; it will be even greater if the contradiction concerns point of conflict that aroused suspicion.

The second possibility is that we were mistaken in what we assumed him to believe. In this case, we might examine our support for what we thought him to believe. The last possibility is that he simply changed his belief about something. This is the hardest possibility for us to examine, unless we observe something that might cause him to change his beliefs.

This type of situation could cause us to split into three possible worlds. In each world, one of these possible explanations would be assumed and the assumed belief would be given a level of certainty (opinion) based on the support for believing that case. Instead of this approach, our system selects one of these possible worlds based on the relative opinions of the supporting beliefs, along with factors such as conflicting goals. We continue to believe this world corresponds to what occurred until a contradiction forces us to change our beliefs.

The selection of one of these possible worlds is one of the reasons we must be able to reason with uncertainty. The opinions about beliefs are used to determine the most likely possible world just as in

ordinary reasoning. We do not want to assume deception is occurring if there is stronger support that something else is happening (e.g., maybe there was a mistake or an agent just changed his beliefs).

The second purpose of uncertainty is to mark as questionable new beliefs that we may have been deceived about. We do this by assigning an opinion to input that reflects that we are uncertain about the information. For example, we ordinarily might give an initial opinion of 2 to something a friend tells us. If we suspect someone is lying to us, we assign an opinion of 5 so that the belief will be easily rejected if necessary.

It is possible that an opponent's report will neither contradict nor match what we believe him to believe. In this case, it is still possible that he is lying if he has invented some information. This may occur because our model of his beliefs is incomplete. It may also occur if he invents some imaginary concept and refers to it in communication.

For example, he may have no belief about where some gold is located, but he may invent a fact and tell us that it is in some particular location, even though that location may not exist. Again, the certainty that this is the case increases if the report pertains to a situation over which there is conflict. On the other hand, this report may simply provide us with new information about an opponent's beliefs.

11.1.4.1. Types of Lies

While there is no method for determining absolutely that another agent is being deceptive, there are procedures we use for making good guesses. Recall that a statement contains a concept proposition and a truth value that we believe holds for it. The object concepts, relation concepts, and attribute concepts correspond to objects, relations, and attributes in the actual world. Thus, the concept proposition will generally have a true value if we believe that the corresponding proposition about the actual world is true.

This shows us the simplest case of lying; the truth value of the statement is simply changed when communicating the concept proposition. For example, suppose John breaks a window. He then holds the belief corresponding to the statement:

BROKE WINDOW JOHN True set1.

If he is asked about the window, he may try to deceive someone simply by asserting the opposite concept proposition:

BROKE WINDOW JOHN False.

Although he communicates this concept as if it were true of the world, the agent represents it in his own mind as

broke window john false

since it is an imaginary concept that he made up.

Deception can be much more complex than simply changing the truth value of a statement; often other components of the concept proposition are changed. For example, suppose that John believes a map is in room1:

INROOM MAP ROOM1 True set1.

If John wishes to deceive someone about the map's location, he could

simply communicate that:

INROOM MAP ROOM1 False.

This is not the most effective deception, however, because it asserts a negative concept proposition rather than a positive one; positive assertions are generally more believable. It would be more effective for John to communicate that:

INROOM MAP ROOM2 True.

Thus, he has pointed the deceived person in another direction to look for the map. Again note that he communicates the abstract situation as if it were true, but he represents it in his mind as:

inroom map room2 True

Since these are the ways we might deceive others, it makes sense that these are the tactics we should suspect others will use to deceive us. Hence, our theory for detecting lies is to compare the concept propositions communicated to us with the ones that we expect. For example, if we already believe an agent believes one thing, and he tells us something different, then he may be lying to us. Of course, we must be aware of other possibilities, such as the chance that we are mistaken (either in our previous beliefs about his beliefs, or in our perception of the communication). If we are suspicious of the opponent, however, then deception will probably be the most plausible explanation of the contradiction.

Whatever our conclusions about the truthfulness of a report, they can be drawn from comparisons between sets of beliefs, situational analysis of the possibility for deception, and analysis of the supports for specific beliefs. In this case, the detection is simplified

because we have communication from the agent that informs us of his intentions; we can compare this communication to our beliefs about what he believes. Deception resulting from an opponent's actions is more complex because we must perceive the meaning of his actions.

11.1.5. Deceptive Actions

As with the case of reports, deception in an opponent's actions must be detected by comparing observations of what he does with what we believe he ought to do, i.e., what we think his goals are. When dealing with deceptive actions, we cannot compare his actions to our model of his beliefs. Instead, we must predict what we believe he ought to do to achieve his immediate goals and compare this with our perceptions of his actions.

We have not implemented a detailed model for predicting an agent's actions. Certain general rules, however, can be implemented with counterfactual rules. For example, if we believe an agent wants something, and we think he knows where it is, then we believe that he will move to that location.

The counterfactual rule describing this reasoning looks like:

SET X nil nil set1

SET X Y nil set2

GOAL Y HOLDING *object Y True set2

INROOM *room *object True set2 ->

EXPECTS PERFORM MOVE_TO *room Y True set1

Note that goals are involved not only for detecting conflicts but also

for predicting an agent's actions. We can describe this prediction model with the following schema:

Assume that ACTION1 is an action that can lead to goal1. If agent X believes that agent Y has goal1, and agent X believes that agent Y believes that ACTION1 can lead to goal1, then agent X expects agent Y to perform ACTION1.

As a counterfactual rule, this is:

SET X nil nil set1

SET X Y nil set2

GOAL Y goal1 True set2

LEADS_TO goal1 ACTION1 True set2 ->

EXPECTS PERFORM ACTION1 Y True set1

Since we are checking for mismatch against our expectations, rather than what we actually believe he believes, there is greater uncertainty involved in the analysis. This is handled by assigning a higher modal category to the rule; this will be added to the uncertainty of the conclusion if the rule fires and asserts the right side. Also, since deception through actions is more difficult to predict, we must be more suspicious of all his actions when there is a conflict of goals.

Consider, as an example, that we believe our opponent is looking for gold and that he knows we are also looking for gold. Furthermore, suppose that we believe there is gold in a place we will call room1. If we think our opponent also knows of this gold, we would expect him to move to room1. If he does not, there are two possibilities.

First, he may be trying to deceive us about his goal. For some

reason he may want us to believe that he is not interested in the gold. Alternatively, we may be mistaken, and he may not know where it is. We must therefore consider each of these possible worlds and determine their degree of possibility by analyzing our support for the two cases. If we are more certain of his desire for gold than we are of his knowledge of its location, we will assume the first case to be his intent.

As an example of our increased suspicions, consider the previous case but assume that we have no knowledge of the gold's location. This means we will have no specific expectations of our opponent. By observing him, however, we may guess that he may know where gold is located and is either unwittingly leading us to it or intentionally decoying us away from it. These suspicions will only be aroused if we know of an immediate conflict. If we believe he is aware of competition for the gold, we will assume he is leading us away. This may not indicate where we should look, but as with the case when he directly lies to us, we may decide not to look in the place he is moving to.

Detecting deception that involves an agent hiding an object also depends on our suspicion. Suppose we observe what we consider behavior that is not ordinary, an opponent holding his hands behind his back (this may only appear out of the ordinary to us if we are suspicious of the opponent). Furthermore, assume we know of some conflict concerning an object; there is an object that he believes we want or that we do not want him to have.

Since the conflict of goals has made us suspicious of our opponent

we will analyze his unusual behavior, and we might hypothesize that he has the object and is concealing it behind his back. On the other hand, he may be holding his hands behind his back to make us think he is concealing an object. We must select the possible world that has the best support; if possible, we can decide from later actions if we were correct and revise our beliefs if necessary.

An agent can also deceive us by communicating information that is not untruthful but is misleading. This is where a theory concerning substitution of equalities is important. A theory of substitution of equalities is necessary for two reasons. The first is that it is needed for any communication between agents who might use different concepts to denote the same object. For example, there may be a room in a house that one agent calls the study and another agent calls the den. If they are aware that the other's concept maps to that room, they will be able to communicate about the room.

The second reason is that deception may center around the substitutions of equalities. For example, an agent may refer to a person by a nickname when he does not want another agent to know to whom he refers; he uses this approach when he is aware that the second agent cannot map the nickname to the person it denotes. In other words, the second agent may be unaware that 'BILL' and 'SHORTY' both map to a person "BILL". Hence, he will not be able to make the substitution.

It is clear that there are many factors in deciding whether deception is involved. In addition, complex analysis is needed to determine exactly what information we are being deceived about and what

we can infer about our opponent's intent.

When we reason about information and beliefs rather than with them, we refer to it as metalevel reasoning. We will discuss how agent's in our system use metalevel reasoning in section 11.2.1.

11.1.6. Theory of Deception

We do not have an algorithm for detecting deception because the problem is not deterministic; in fact, it is impossible to detect deception with perfect confidence; usually we find out after the fact that we have been deceived. Even if someone gives us information that we are certain is incorrect, it is not conclusive that the person is lying to us. We may be incorrect in our own beliefs, or the other person may be mistaken rather than intentionally lying.

We do, however, have a precise theory of deception that we implement. A theory of deception must contain two key elements: a theory defining when deception occurs (from an objective point of view), and a theory of uncertainty to measure an agent's confidence in detecting deception. We will first examine our definition of when deception occurs. Afterwards, we will explain how our previously defined theory of uncertainty is integrated into deception reasoning.

Our method of detecting deception is heuristic. We base our approach on the way people suspect deception; if we have reason to suspect deception and the world does not match our model of it, then we must decide if we are being deceived.

The idea behind deception is to alter someone else's beliefs; this

is usually done by an agent misleading another about his beliefs. Deception in general involves one person intentionally communicating information about the world that he does not believe is true. In terms of our theory, he is communicating an abstract situation that he does not believe corresponds to the actual world. If we wish to deceive someone about a subject, we communicate to them statements relevant to that subject that we do not believe.

For example, I may believe one thing, but I want my opponent to believe that I believe something else so that he will also believe it. There are various ways that we may try to convince him of our false beliefs; misleading actions or outright lies may be used. Whatever method is used, our goal is the same; we want our opponent to believe something that does not correspond to the actual world.

It is clear that both nested beliefs and 'imaginary concepts' (concepts that an agent believes do not map to the physical world) are needed for deception. Note that an imaginary concept proposition does not need to be entirely composed of imaginary concepts; it may concern real objects but assign them relations or attributes that they do not have in the physical world. In this sense, we do not mean that an imaginary concept is necessarily imaginary; we mean that an agent does not believe it corresponds to the physical world at a particular time.

For example, John may believe

INROOM ROOM1 MAP True set1

but at the same time he can represent in his mind the concept

inroom room2 map True set1

John does not believe the second concept proposition corresponds to the

world so it is imaginary and represented as such. Note that the entire concept proposition is treated as imaginary; this does not imply that John thinks the map, room2, or being in room2 are imaginary by themselves.

In terms of our model theory, these are just two of many abstract situations that John can represent in his mind; one of them he believes corresponds to the world, and the other he believes does not. He can reason about either concept. Deception, however, involves convincing another agent that the latter concept is the one that John believes is true in the world. We can define deception as follows:

Let set1 be agent1's beliefs, set2 be agent1's model of agent2's beliefs, and set3 be agent1's model of agent2's model of agent1's beliefs. Set1 contains CONCEPT1 and concept2, where CONCEPT1 and concept2 are contradictory. We say that agent1 is deceiving agent2 if his goal is to include concept2 in set3 for the purpose of inserting concept2 (correctly) into set2 and eliminating CONCEPT1 from set2.

The structure of our system allows us to include uncertainty easily in deception reasoning. Although this reasoning takes place at a metalevel because it is about beliefs (discussed below), deception reasoning still uses statements and counterfactual rules. To each statement and rule for deception reasoning has a modal category. These modal categories, or opinions, are used the same way as in ordinary reasoning with beliefs (see Chapter Five for a discussion of uncertainty).

If there is a contradiction, the opinions of supporting statements are compared to determine the most plausible explanation for the

inconsistency (e.g., deception, mistake, change in belief). If deception is decided on as the explanation, then the confidence an agent has in the explanation will depend on the modal categories of the supporting statements and rules. The modal categories of the rules can be altered to create agents who range from naive to paranoid.

Detection of deception depends on an agent's awareness of this definition. For agent2 to avoid being deceived, he must first identify the conflict so that he will be alert to the possibility of deception. He then must be wary of any communication from agent1 that attempts to alter his beliefs about what agent1 believes.

11.2. Implementation

We will next turn our attention to the implementation of our theory of deception.

11.2.1. A Bilevel Reasoning System

The capabilities that we need require a system that can reason on a baselevel as well as a metalevel. This is a bilevel system, as described in [Genesereth and Nilsson, 1987]. The system described by Genesereth and Nilsson has both a baselevel database with symbols to represent objects and a metalevel database that can represent the baselevel symbols and relationships among them. This allows metalevel reasoning about the baselevel reasoning process. This type of system is what we need because human reasoning seems to take place at different

levels. Metalevel reasoning allows us to reason *about* our reasoning process and to "step back" from a problem in order to get a broader perspective. A system that detects deception could make good use of this idea by analyzing goals and reports for possible contradictions or conflicts before using them to draw conclusions.

The baselevel of our system includes most of the beliefs contained in a possible world. These worlds contain beliefs, rules for deriving new beliefs, and rules for maintaining consistency. Counterfactuals are used as inputs to determine the next database (possible world) to be generated based on the current database (world) and consistency rules.

At the metalevel, counterfactual rules describe reasoning rules that agents can use to compare their own beliefs along with their models of other agents' beliefs. These rules allow an agent to look at baselevel beliefs and his perceptions of the world and communication with other agents to determine if there are conflicts. New baselevel beliefs are then generated that either add information to the baselevel beliefs (such as perceptions or new beliefs of other agents), indicate that a conflict of goals is believed to exist, or indicate that an opponent has lied or acted in a way that aroused our suspicions. Reasoning at this level also can fire counterfactual rules to determine what information may be inferred from deception.

In addition, the metalevel affects an agent's models of other agents' beliefs. For example, if new input causes John to modify his set of beliefs, it may also cause his model of another agent's beliefs to change. John must determine whether or not to change his models of other agents' beliefs, and he must decide which models to change and

how to change them. Hence, he is reasoning about beliefs (whether or not to revise them) but he is not reasoning with the beliefs.

For example, if Tom hits John and A1 witnesses the event, then A1 changes his beliefs to include the belief that Tom hit John. This belief may have further implications that A1 can derive, e.g., the realization that Tom must dislike John. If A2 were also in the room when Tom hits John, then A1 must decide if he should revise his model of A2's beliefs. In this case, he would probably assume that A2 witnessed the event as well, so A1 believes that A2 also believes that Tom hit John. If A2 is not in the room when the hitting occurs, however, then A1 should not alter his model of A2's beliefs because there is no reason for A1 to think that A2 believes the hit occurred.

We refer to this as metalevel reasoning because A1 must decide whether A2 witnessed the event before he adds the belief to his model of A2's beliefs. After A1 has decided to modify his model of A2's beliefs, he can then reason with the belief to conclude that A2 must also realize that Tom dislikes John. Another possibility is that Tom and John are in collusion to deceive A1 and A2 into thinking that they dislike each other. In this case, if A1 is deceived then he will also believe that A2 is deceived, even though A2 may not actually be convinced that Tom dislikes John. Hence, A1 may make a mistake about A2's beliefs.

There are several capabilities that this metalevel reasoning gives us. Agents are able to detect goal conflicts and thereby become suspicious. They can also compare their perceptions to their beliefs and decide if discrepancies should be treated as deception. If they

decide deception is involved they will use metalevel reasoning to determine the implications. Finally, they can use metalevel reasoning to decide when they should alter their models of other agents' beliefs.

It can be seen that the use of such metalevel reasoning creates a powerful system. Of course, the speed of such a system is decreased because of the amount of processing. It seems that this sacrifice in time is justified to obtain a more powerful system that can represent and compute what we desire. Metalevel reasoning is separated from bilevel reasoning only by the use of different predicates and the ability to access nested belief sets; the two levels of reasoning fit smoothly together and work as one system.

Deception reasoning is enhanced by the ability to reason about types of existence (discussed in Chapter Six) with metalevel reasoning. For example, when an agent realizes that communication from an opponent contradicts another belief, he must decide if the opponent is lying or is mistaken. If the agent believes that the subject of communication exists in fiction or used to exist physically but no longer does, he is more likely to think his opponent is mistaken.

11.2.2. Implementation of Deception Reasoning

We now discuss our implementation of deception reasoning. Since recognition of conflict is essential for detecting deception, that topic is discussed first. Deception involving lying, which requires direct input, is dealt with next. Following that is a discussion of how deceptive actions might be detected.

People generally deceive others to gain an advantage. Such an advantage will be desired when there is a conflict over some issue, whether it be an object or an idea. It follows that we must examine our own goals and those that we believe an opponent has to determine if they conflict. This reasoning about goal conflicts is an example of metalevel reasoning.

We defined goals as part of our ontology in Chapter Six. A goal will be treated as an object and will have the form:

GOAL *person1 *goal *set

where *person1 is the person whose goal is represented, *goal is the goal (consisting of a concept proposition and a truth value), and *set determines who it is that believes *person1 has that goal. For example, if John believes that Bill's goal is to possess the map, it would be represented as:

GOAL BILL POSSESS MAP BILL True set1.

If, instead, John believes that Bill believes John's goal is to have Mary possess the map, it would appear as:

GOAL JOHN POSSESS MAP MARY True set2.

As a final example, suppose that John believes Bill does not want John to have the map. This would be written as:

GOAL BILL POSSESS MAP JOHN False set1.

The concept of POSSESS(ion) is very useful for representing goals. It can be used to represent the goal of possessing some information as well as possessing an object. There are two general situations that will cause a conflict between Agent1 and Agent2 when dealing with goals of possession. Since we are concerned with detecting deception, we

must identify those situations where Agent1 will believe that Agent2 believes a conflict exists. If there is a conflict that Agent2 is unaware of, then Agent1 has no basis for suspecting Agent2 of deceiving him. The situations in our system that will cause Agent1 to believe Agent2 feels there is a conflict are as follows:

1. One of the agents wants a person to possess an object, and the other agent does not want that person to possess that object.
2. One agent wants one person to possess an object, and the other agent wants a different person to possess that object.

Note that the goals themselves resemble any other belief, but they are embedded in the GOAL predicate. In this sense, we can think of the GOAL comparison taking place at another level above normal reasoning. The system begins metalevel reasoning when a new GOAL is identified and compares it to other known goals to determine if a new conflict over an object has appeared. Two separate goals will fire a rule only if the goals create a conflict according to the rules above. If the rule fires, the person detecting the conflict makes a record of who and what the conflict involves.

Identifying the possibility of deception will also take place at a different level. We will first consider the detection of lies. For a lie to be identified, there must be some direct input from another agent. This input will be in the form of a report and will have the syntax:

INPUT REPORT *report *speaker *listener.

Input from communication with another agent is distinguished as a report so that it may be differentiated from other input, such as perceptions. The *report will have a similar form as any other belief, except that the set to which it belongs will not be labeled. Instead, *speaker indicates the agent telling the information, and *listener is the agent it is being told to. If Bill tells John that the map is in the library, we represent this as:

INPUT REPORT INROOM MAP LIBRARY True Bill John.

When a report is received, the system can "step back" and compare it to the listener's sets of beliefs at different levels of nesting. This process occurs because reports are received at the metalevel. They therefore have different leading predicates than statements at the baselevel and will not trigger any baselevel reasoning rules until they have been analyzed and transformed at the metalevel. Since it is more important to identify a contradiction between the report and the speaker's beliefs, this will be given priority.

The first step in the analysis is to determine whether the listener already believes that the speaker has some belief about the subject of the report. The system also checks to see whether the listener has a belief about the subject of the report. If the listener finds that he already believes the speaker to have some belief, he compares what the speaker is reporting with what he thought the speaker believed. If there is a contradiction, there are several possibilities for why the contradiction occurred.

If there is an inconsistency between what the speaker says and what we think he believes, the listener checks to see if he is

suspicious of the speaker concerning the subject of the report; this is accomplished by checking to see if he has a belief that a conflict of goals exists. If so, he determines that the speaker is lying with a certainty assigned to this belief in the usual way based on opinions of rules and supporting beliefs.

A general counterfactual rule for this looks like:

```
CONTRADICTION *speaker *listener *report *set1
SET *speaker nil nil *set1
SET *speaker *listener nil *set2
CONFLICT *speaker *listener *report *goal2 *set2 ->
        LYING *speaker *report *set1
```

If there is no conflict, he must decide whether the speaker has changed his belief and is reporting a new belief, or whether the listener was originally mistaken. This is done by examining the support or attitude of the speaker's previous belief. If the listener feels he had good reason for believing the speaker's previous belief on the subject, he will assume the speaker's belief has changed. If, however, the support was weak to begin with, he may assume that he made a mistake earlier.

A counterfactual rule for this looks like:

```
CONTRADICTION *speaker *listener *report *set1
SET *speaker nil nil *set1
SET *speaker *listener nil *set2
*previous_belief *set2
opinion(*previous_belief) > threshold ->
        *report *set2
```

If the support for the previous belief is strong, then the report will be accepted as a revised belief. Another rule can identify the previous belief as a mistake if it has weak support. The threshold can be varied and is set by the system operator when defining an initial world.

11.3. Conclusions

Our theory of deception is heuristic because the problem is not deterministic; we have, however, presented a precise theory for detecting deception. We believe that an important aspect of the problem is identification of situations where deception may occur. These situations are determined by recognizing when there is a conflict between the goals of two agents.

When we wish to deceive others, we know that it involves communicating statements that do not reflect our own beliefs. This can be implemented by first identifying through goal analysis those situations when it would be advantageous to deceive another agent. We can then communicate a statement concerning the relevant concept proposition, but we first modify the statement so that it does not reflect our own beliefs.

Our basic approach to detecting deception (when it is suspected) is to watch for communication of suspicious statements. By suspicious, we mean communication that does not match our expectations. This communication may be verbal or through actions, but verbal communication is easier to interpret.

Essentially, our system detects deception by finding inconsistencies between what we believe and what we perceive. These inconsistencies must then be explained as deception, mistakes, or changes in the world. This is an ideal approach for our system because our counterfactual engine is designed to discover and resolve inconsistencies.

12. A Multiple-Agent Simulation

Using the theory of this thesis as a basis, we have implemented a simulation that permits multiple agents to act in a small world and to reason about their beliefs and each other. The simulation has been implemented in the counterfactual engine; this has simplified the integration of our belief reasoning system, which has also been implemented in the counterfactual engine. In this chapter we will explain how our simulation operates, and in Chapter Thirteen we will provide examples of the results.

12.1. Structures

The simulation technique we use is general purpose and event-driven; changes in both beliefs and the physical world are determined when a physical or mental event occurs. The simulator rules that increment simulation time, perform schedule maintenance, and determine the effects of actions on the world operate independently of the rules and facts used to model the simulated world. Thus, the problem domain can be changed and modeled with any facts and rules without affecting the simulator's basic operation; the world that is simulated depends only on the definition of the initial world and the counterfactual rules that define the effects of physical and mental events.

The two basic entities of our simulation are statements and schedules. Statements correspond to either facts or beliefs and are

operated on by rules. Schedules represent lists of actions that an agent believes he will perform. The simulation rules perform the functions of incrementing time, revising schedules, removing actions from schedules to be performed, and determining if actions have been performed so that the next actions can be removed from agents' schedules.

12.1.1. Statements in the Simulation

There are several types of statements in the simulation. Statements concerning facts and beliefs describe the state of the world and the contents of agents' beliefs. Statements about actions can be part of an agent's beliefs or actual actions performed by an agent.

12.1.1.1. Beliefs and Facts

Many of the statements used in the simulation describe states that exist either physically or mentally. Statements concerning beliefs are of the form described throughout this thesis; they represent what different agents believe about the world and what agents believe other agents believe about the world (nested beliefs). Statements may also specify physical states that exist in the world (i.e., facts). These statements have the same form as belief statements except that the belief set label attached to them is always 'world'. When agents have correct beliefs, their beliefs will correspond to facts about the world.

The similarity between facts and beliefs allows them to be treated in the same way by the simulation. All statements can be asserted, retracted, or revised by counterfactual rules. This allows both the world and an agents' beliefs about it to change over time. The rules may be written with variables that will match only statements of a certain type, i.e., facts or beliefs; they may also be written to incorporate statements of both types. This is necessary because actions depend on both facts and beliefs, and actions affect both the world and agents' beliefs about it; the need for this interaction between action and belief was pointed out by Moore (see Chapters Three and Eleven).

The merging of belief reasoning with a changing world is possible because the mental event logic and physical event logic that we use are the same. Since both the belief reasoning system and the simulation are implemented in the counterfactual engine, they share a common solution to the frame problem. The simulation treats beliefs and facts as the same type of entity; they are statements that can be classified by the set labels attached to them.

As mentioned previously, a statement represents either the state of some physical object or the state of an agent's belief (i.e., the state of belief the agent holds for a proposition). A state begins when the statement is asserted and persists until the statement is retracted. The interval that a belief statement is true indicates when that belief is held to be true, and the interval that a fact statement is true indicates when that fact is true in the world. The assumption that truth values persist makes it unnecessary to check each statement

at every step of the simulator.

12.1.1.2. Actions

Statements about actions are distinguishable because they have no set label identifying them with the world or a belief set. An agent can reason about actions by embedding them in statements of beliefs; an agent can also plan actions by revising his own schedule. Actions are asserted as counterfactuals to indicate that they are being performed; when this occurs the asserted actions match and fire rules to begin a chain of changes. Actions need not be placed on a schedule to be performed, but they are the only type of statement that appears on a schedule.

Not all actions need to be scheduled. Many actions occur as responses to stimuli; a person who is pinched will shout immediately, not schedule a shout to occur in the future when he is not busy. These actions do not occur spontaneously; they are caused by scheduled actions or other intermediate actions such as perceptions, communications, etc.

Although actions need not be scheduled, there is a general sequence of events for those that are scheduled. An agent can reason about goals as part of their belief sets; they can also reason about knowledge with statements that contain an action as the predicate, such as:

`MOVE_TO *room *agent`

When this action is placed on a schedule, it appears only as (MOVE_TO

*room) because the entire schedule is associated with a particular agent. The predicate PERFORM and a time to be performed are added to the statement when the action is taken from the schedule to be performed (section 12.1.2. shows the simulator rules that handle this). An example of such a statement would be:

```
PERFORM (MOVE_TO *room) *agent *time
```

When the simulation time equals the time an action is to be performed, the action becomes the argument of the predicate ACT. The simulator rule that makes this transition is:

```
((PERFORM *action *agent *start_time)
 (TIME *time)
 (add 1 *time *new_time)
 (= *new_time *start_time) ->
 (ACT *action *agent *start_time))
```

An example of this type of statement is:

```
ACT (MOVE_TO *room) *agent *time
```

The ACT predicate indicates that the action is being attempted and the simulation will determine if it is successful and what the results are. In other words, ACT indicates that the action is asserted as an event. The counterfactual rules determine the effects of the action, and the frame problem associated with the effects is handled by the counterfactual engine's implementation of the frame problem.

12.1.2. Schedules

The second basic entity that the simulation contains is a

schedule. A schedule consists of a list of actions to be performed by an agent in a particular order. Each agent has his own independent schedule of actions. A schedule is associated with a particular agent and is part of his belief system. Our representation with set labels allows agents to represent beliefs about other agents' schedules and reason about those schedules. We have not implemented rules to perform such reasoning in the current simulation.

As stated earlier, this simulation is time driven. Time is incremented at each possible world generated so that the simulation time corresponds to the timestamps of the possible worlds. Since possible worlds change due to events occurring at instants of time, actions are asserted at the instants of time when the time is incremented.

A schedule specifies only an ordered list of actions to perform; it does not indicate when to perform them. This gives the simulation flexibility, because revising the schedule does not require recomputing the times to begin actions. Revising a schedule only requires that actions be appended to the beginning or end of a schedule; all actions will be performed when they eventually appear at the top of the list.

When the time is incremented, the next action on each agent's schedule is removed. A relative delay is determined for that type of action and is used as an offset from the current time. This absolute time for the action to occur is only computed after it is removed from the current schedule. Flags for each agent indicate whether or not he is busy; he is busy if he has planned the time of his next action, and he cannot plan another action until he has completed it. This does not

mean that he cannot do anything in the meantime because he can still reason with beliefs and perform unscheduled actions.

The rules for performing this function are:

```
((BUSY False ?agent ?sometime)
 (SCHEDULE ?agent (?h.?rest) ?time)
 (<> ?rest nil)
 (= ?time ?sometime)
 (TIME_REQUIRED ?h ?delay)
 (add ?sometime ?delay ?start_time)
 (add 1 ?start_time ?new_time) ->
      (SCHEDULE ?agent ?rest ?new_time)
      (PERFORM ?h ?agent ?start_time))

((BUSY False ?agent ?sometime)
 (SCHEDULE ?agent (?h.nil) ?time)
 (= ?time ?sometime)
 (TIME_REQUIRED ?h ?delay)
 (add ?sometime ?delay ?start_time)
 (add 1 ?start_time ?new_time) ->
      (SCHEDULE ?agent ((WAIT)) ?new_time)
      (PERFORM ?h ?agent ?start_time))
```

The second rule says that an agent who has no more actions schedule will WAIT (i.e., he will do nothing).

12.1.2.1. Timing

The time associated with each action insures that it is only performed at the appropriate simulation time. Schedules also have an associated time to indicate when they are valid. When an action is removed from or added to a schedule, the old schedule is rejected and the new schedule replaces it. Thus, an action is only removed from a schedule if that schedule is valid at the simulation time and the agent is not busy.

As an example, consider an agent John who has the following schedule at TIME 2:

```
SCHEDULE John ((OPEN_UP door1) (MOVE_TO room2) 2)
```

If John is not busy the first action is removed and the following statement is asserted:

```
PERFORM (OPEN_UP door1) John 3
```

since the delay for the OPEN_UP action is 1 time cycle. John's new schedule, which becomes valid after he performs (or attempts) the action, is:

```
SCHEDULE John ((MOVE_TO room2) 4)
```

The simulation continues and at TIME 3 the statement:

```
ACT (OPEN_UP door1) John 3
```

is asserted as an event. The counterfactual rules then determine the effects of the event.

The TIME_REQUIRED for an action can be specified for each type of action; this capability is needed because different actions require varying amounts of time to complete. The delay time represents the

length of time needed to complete an action.

The simulation uses the predicates `SUCCESSFUL` and `FAILED` to indicate whether the action has actually been performed. If an action fails, the agent may or may not know why the action failed. If he does not know the reason, he only knows that he must still perform the action and must replan for it. If he does know why the action failed he can revise his schedule to perform necessary actions before reattempting the failed action. For example, an agent may try to open a door and finds that it is locked. He then believes that he must unlock the door before he can open it; it is possible, though, that he is mistaken and the door is actually stuck rather than locked. A rule that shows this revision of a schedule as the result of a failed action is:

```
((ACT (OPEN_UP ?door) ?agent ?act_time)
 (TIME ?time)
 (= ?time ?act_time)
 (INROOM ?room ?agent True world)
 (LOCKED ?door True world)
 (BLOCKED ?door ?value world)
 (SET ?agent nil nil ?set)
 (CONNECTS ?door ?room ?room2 True world) ->
      (FAILED (OPEN_UP ?door) ?agent ?time)
      (GOAL *agent (LOCKED *door) False *set)
      (REPLAN ((UNLOCK *door)(OPEN_UP *door)) *agent))
```

The counterfactual engine's ability to accept multiple counterfactuals enables us to simulate simultaneous actions by

different agents. Since each agent has an individual schedule, it is possible that the next action on several agents' schedules will be performed at the same time; the actions are introduced as counterfactuals into the same possible world. All effects of each action are determined by the simulation; if the actions affect each other (e.g., two agents pushing on the same door from different sides), counterfactual rules are used to determine which action succeeds.

We should also point out that agents need not perform actions simultaneously. One agent may be busy performing an action while another is removing the next action from his schedule and thinking about when to perform that action. All operations on schedules (revising, selecting the next action, etc.) occur when the simulation time is incremented. These operations result in counterfactuals that trigger a chain reaction of changes in beliefs and facts that become new mental and physical events (intermediate counterfactuals). The possible world is generated when all changes have been resolved and no new intermediate counterfactuals are generated.

12.1.3. Rules

Statements are revised by the firing of counterfactual rules. The simulation contains rules of several different types. These rules may affect beliefs, facts, or schedules (which actually represent an agent's beliefs about the actions he wishes to perform).

This thesis has discussed rules for revising beliefs at length; the rules for revising facts operate the same way, except that they

operate with facts. Some of these rules are consistency rules. For example, we assume that agents believe an object cannot be in two places at the same time. We must also have such a rule for facts to prevent the simulation from allowing this to happen; if the simulation sees that an object or agent in the world is asserted to be in a location that differs from its previous location, the statement about its previous location is retracted.

Rules concerning the effects of actions may also change facts; these rules specify the effects that an action will have, given certain preconditions. For example, the action of moving to another room will result in an agent being in the new room, provided that the rooms are next to each other and there is a clear doorway between them. Some rules will cause changes in beliefs and actions; the act of moving will cause an agent to believe that he is in a new room.

Changes in beliefs can be caused by several types of mental events; these include perception, communication, and responses to stimuli from the surrounding world. Rules about perception allow agents to change their beliefs about their surroundings. Communication with other agents can add new information to an agent's beliefs or cause other beliefs to be revised. These will be discussed in greater detail below.

Responses usually occur as the result of actions. If an agent is spoken to he may respond without scheduling the action of speaking. The simulation is flexible, however, because the actions of speaking and asking questions may be scheduled like any other action. This is useful when an agent wants to communicate but must first satisfy some

other conditions (e.g., moving to another room or making a telephone call).

Reflex responses may also occur without an agent scheduling them. An agent does not schedule an act of falling down when he is pushed. Neither should he schedule the act of jumping out of the way of a speeding truck that is about to hit him; these actions are the results of perceptions or other agent's actions and may be performed immediately. If such actions are inconsistent with a scheduled action that an agent is performing, consistency rules will determine the results; unscheduled actions have priority because they generally involve situations that involve immediate actions.

Responses may also occur as the result of an agent's own actions. An agent who opens a door believes that the door is open. If the simulation determines that an agent fails at an action, the agent may realize why he failed (or think he does). If a door will not open, the agent may conclude that it is locked (even though it may only be stuck). When an agent fails at performing an action, he reacts by revising his schedule. If he thinks he knows why he failed he can revise his schedule accordingly; if he does not have any idea why he failed, he must make a new plan and attempt it. We have not implemented a planner in the current simulation. Initial goals and plans are inserted into the initial world. An agent's ability to reason provides flexibility because he can determine new goals and revise his schedule of actions.

An agent may also revise his goal as the result of thinking; he may consider goals and (what he believes is) the current state of the

world and conclude that he must perform some action. Thus, the integration of thinking and acting allows both physical and mental events to cause changes. Reasoning takes time just as actions take time, and the delay between selecting an action and performing it accounts for the time required to think about how to act.

12.1.4. Counterfactual Engine Control Rules

Control rules are special rules that operate after a stable world has been generated; they examine the state of the world and agents' beliefs to determine what counterfactuals to assert into the next world. These rules increment the time and perform schedule revisions specified by an agent. They also check whether the previously scheduled action was successful or failed, and they determine when an agent is no longer busy performing an action. By asserting that the agent is not busy, the control rules fire the rule that performs the next action on an agent's schedule. If all agents are busy, the time is merely incremented and the old time rejected.

Control rules are also used to handle some metalevel reasoning involved in communication and deception. If an agent receives a report that contradicts previous beliefs but has no cause to suspect deception, then the agent must decide if the contradiction is a mistake or new information. This is done by comparing the opinion supporting the report to a variable threshold that is preset by the programmer before beginning the simulation.

Control rules have the same form as other rules but are defined as

control rules. The right side of a control rule is always asserted if it fires; neither the left side nor the control rule can be rejected. Control rules enable the simulator to perform bookkeeping and monitoring outside of the world and belief sets.

12.2. Capabilities

The similarity between our theory of belief reasoning and the model we use for actions allows them to interact as we discussed in Chapter Ten; actions depend on beliefs and beliefs are affected by actions. The simulation also allows agents to communicate and perceive the environment as discussed in Chapter Ten. We have also implemented the other theories discussed in this thesis, with the exception of uncertainty; the counterfactual engine has not yet been modified to allow the dynamic assignment of modal categories that we need for uncertainty reasoning. We have implemented some simple substitution of equalities in communication, and agents may reason with nested beliefs and detect deception.

12.2.1. Communication

Communication involves actions that may be responses or performed as scheduled. We distinguish two types of communication, reports and queries. An agent may perform a communication action whenever he chooses; the simulation will determine if the message is successfully communicated, but this success does not guarantee that the listener

perceived the message correctly.

An agent will generally ask questions when his goal requires him to obtain information; he will speak in response to a query, but either of these actions can be scheduled for any reason that is encoded as the result of a rule. When an agent communicates, he must specify whether he is asking or speaking, to whom he wishes to communicate, and the content of the message. The simulation determines if the communication can take place; in our simple model this means the agents must be in the same room (since we have no telephones in our model world). If successful, the simulation identifies an input of the appropriate type (report or query) and specifies the content, the sender, and the receiver of the message.

The rules that perform these functions are:

```
((ACT (SPEAK_TO *person *message) *agent *time)
 (INROOM ?room ?agent True world)
 (INROOM ?room *person True world)
 (<> *agent *person)
 (TIME *act_time)
 (= *act_time *time) ->
      (INPUT REPORT *message *agent *person *time))
```

```
((ACT (ASK *person *question) *agent *time)
 (INROOM ?room ?agent True world)
 (INROOM ?room *person True world)
 (<> *agent *person)
 (TIME *act_time)
```

```
(= *act_time *time) ->
```

```
(INPUT QUERY *question *agent *person *time))
```

The decision to speak to another agent is usually a response that an agent makes when spoken to. The decision to ask another agent for information is handled by a counterfactual rule that causes an agent to ask another agent for information if the first agent needs the information to find an object, is in the same room as the second agent, and trusts the other agent.

Counterfactual rules determine how an agent responds to input. If the input is a query then the agent may or may not respond, and his response may be truthful (according to his beliefs) or a lie. In our simulation, an agent will respond with what he believes is the correct answer if he has a belief and has no conflict with the asking agent over the subject of the question. If he has a conflict, then he will give an answer that is different from what he believes; if he has no belief about the subject, he will not respond. Input reports may be accepted as new information if they concern beliefs or facts that the listener is ignorant about. On the other hand, an agent may perform deception reasoning on reports; this reasoning will be discussed below.

12.2.2. Perception

Our model of perception allows agents to notice objects, other agents, and statuses of objects (e.g., a door is open or closed, an agent is holding an object). The two major cases of perception are

when a new object is discovered and when an object thought to be at a particular place is missing. These cases require only a few rules. The complexity of the perception model could be increased with more detailed rules.

An agent has a model of the world, and perception is determined by comparing that model with the actual world. The simulation does this comparison whenever the state of objects change. If an agent walks into a new room and an object is there that he did not know about, then his model of the world will be extended and he will perceive that object. If an agent is in a room and another agent enters it, the first agent will notice the new agent and any objects that he is carrying. Agents may also perceive other agents who are in adjoining rooms if the connecting door is open. Thus, agents can revise and extend their model of the world simply by moving about and perceiving changes.

The case of perceiving objects that are missing is handled by a single rule. If an agent thinks that an object is at some location and moves to that location, he will notice that the object is missing if the object is relevant to his goals. For example, an agent may notice that a fly is buzzing about a room before the agent leaves the room. If the fly is gone when the agent returns, he will not notice the fly is missing unless the agent has returned with a fly swatter with the intention of removing the pest. Thus, agents do not notice that objects are not in the expected location unless the location of the object is important to a goal.

The rule for this is:

```
((ENTER *time *room *agent)
 (SET *agent nil nil *set)
 (INROOM *room *thing True *set)
 (INROOM *room *thing False world)
 (GOAL *agent (POSSESS *thing) True *set) ->
      (INROOM *room *thing False *set))
```

Thus, an agent who moves to a location to retrieve an object that he thinks is there will perceive that it is not where he thought when he enters the room.

12.2.3. Substitution

We have implemented substitution of equalities in communication. This is necessary because agents think with different concepts that denote the same situations. When an agent receives an input through communication, he will receive a list of concepts. If the concepts are not all familiar to him, then the communication will simply be stored but will not trigger any responses.

If an agent does not understand a concept involved in communication, he can try to make a substitution for that concept. Such a substitution is possible when the agent has beliefs about the situation the speaker denotes with the concepts. In this case, the agent can check for a concept of his own that denotes the same situation; if he finds one, he substitutes his own concept for the speakers and will then be able to process the input.

The rule that handles substitutions in communication is:

```
((INPUT REPORT (INROOM *Z *W) *speaker *listener *act_time)
  (TIME *time2)
  (= *act_time *time2)
  (SET *listener nil nil *set)
  (SET *listener *speaker nil *set2)
  (denotes *X *Y *set)
  (denotes *Z *Y *set2)
  (denotes *Q *P *set)
  (denotes *W *P *set2) ->
  (INPUT REPORT (INROOM *X *Q) *speaker *listener *act_time))
```

12.2.4. Nested Beliefs and Deception

Since set labels are attached to beliefs, rules can be written to allow agents to reason with beliefs from different sets. This allows them to reason with nested beliefs. They can also reason about information considered to be common knowledge; this information includes what certain actions mean and the layout of the world.

Nested beliefs are important for reasoning about other agents goals in determining conflicts and detecting deception. Conflicts are determined by discovering mutually exclusive goals belonging to two separate agents. Two types of conflict are distinguished according to whether an agent believes only he is aware of the conflict or believes the other agent is also aware of it.

Reasoning about deception involves reasoning about conflicts and

behavior, either communication or suspicious behavior such as blocking a door. When a report from an agent contradicts either the listener or the listener's previous beliefs about what the speaker believes, the agent must do some deception analysis. If the agent knows there is a conflict between himself and the speaker that the speaker knows about, and the contradictory report relates to the point of conflict, then the listener will suspect deception.

If there is no conflict that the listener is aware of, then he has no reason to suspect deception. In this case, he must determine whether there was a mistake or the report contains information that has changed. If the contradiction is with the listener's own beliefs, he must decide if the report is a mistake on the part of the speaker or represents new information that should cause the listener to revise his beliefs. If the contradiction is with the listener's model of the speaker's beliefs, the listener must change his model of the speaker's beliefs. He must also determine if the change is necessary because he was wrong about the speaker's beliefs or because the speaker has changed his beliefs. In either case of a contradiction, the decision is made by comparing the support of the statements to a threshold value. There are different threshold values depending on whose beliefs the report contradicts; this is because agents are less likely to change their own beliefs when they are contradicted.

12.3. Conclusions

The facts that describe the initial world are specific and

determine how the simulation will behave. Although some rules are domain specific and are applicable only in limited solutions, all rules are written using variables. This makes the rules as general as their subject matter will permit; rules are only instantiated with specific objects when they are applied.

We have succeeded in implementing a simulator that incorporates most of the theory presented in this thesis. The use of the counterfactual engine allows a simulation that incorporates the similar mental and physical event logics. The simulation allows agents to perceive, communicate, make substitutions, reason with nested beliefs, and reason about deception. Chapter Thirteen gives the results of our simulation of a model world.

13. Results

This chapter outlines the results obtained from the implemented simulator. The simulation we generated ran through 31 steps (distinct possible worlds). This simulation was implemented in VPI Prolog and version 2 of the VPI Counterfactual Engine running on a VAX/VMS 8800. The CPU time used for the simulation was approximately 24 minutes and 7 seconds.

The simulation contains 101 facts and beliefs in the initial world. There are 83 rules in the simulator. Of these, 12 are actual simulator rules that perform operations on schedules and maintain the correct simulation time. Another 12 rules deal with deception reasoning. There are 18 rules that act as consistency constraints, and the remaining rules specify the results of events.

The facts and beliefs of the initial world are:

```
(THRESHOLD1 2)
(THRESHOLD2 3)
(CONNECTS doora room1 room2 True *set)
(CONNECTS doora room2 room1 True *set)
(CONNECTS doorb room1 room3 True *set)
(CONNECTS doorb room3 room1 True *set)
(CONNECTS doorc room2 room4 True *set)
(CONNECTS doorc room4 room2 True *set)
(CONNECTS doord room2 room5 True set1)
(CONNECTS doord room5 room2 True set1)
(CONNECTS doord room2 room5 True world)
```

```
(CONNECTS doord room5 room2 True world)
(CONNECTS doord room2 DUNGEON True set2)
(CONNECTS doord DUNGEON room2 True set2)
(TIME_REQUIRED (MOVE_TO *h) 1)
(TIME_REQUIRED (OPEN_UP *h) 1)
(TIME_REQUIRED (SHOVE *h) 1)
(TIME_REQUIRED (WAIT) 1)
(TIME_REQUIRED (PICK_UP *h) 1)
(TIME_REQUIRED (UNLOCK *h) 1)
(TIME_REQUIRED (BLOCK *door) 1)
(TIME_REQUIRED (FIND *key) 1)
(SET Hero nil nil set1)
(SET Villain nil nil set2)
(SET Hero Villain nil set3)
(SET Hero Villain Hero set4)
(SET Villain Hero nil set5)
(SET Villain Hero Villain set6)
(AFRAID_OF Villain Hero False world)
(INROOM room2 Hero True world)
(INROOM room3 damsel True world)
(OPEN doora False world)
(OPEN doorb True world)
(OPEN doorc False world)
(OPEN doord True world)
(LOCKED doora True world)
(LOCKED doorb False world)
```

(LOCKED doorc False world)
(LOCKED doord False world)
(INROOM room2 Villain True world)
(INROOM room5 key2 False world)
(INROOM room2 key1 True world)
(INROOM room4 key2 True world)
(UNLOCKS doora key2 True world)
(UNLOCKS doora key1 False world)
(BLOCKED doora False world)
(BLOCKED doorb False world)
(BLOCKED doorc False world)
(BLOCKED doord False world)
(HOLDING key1 Hero True world)
(HOLDING key2 Hero False world)
(TIME 1)
(KEY key1 world)
(KEY key2 world)
(DAMSEL damsel world)
(denotes key2 ("key2") set1)
(denotes key2 ("key2") set3)
(denotes room5 ("room5") set1)
(denotes DUNGEON ("room5") set3)
(NO_CONFLICTS key2 Hero Villain set1)
(NO_CONFLICTS key2 Villain Hero set2)
(KEY key1 set1)
(KEY key2 set1)

```

(damse1 damse1 set1)
(KEY key1 set2)
(KEY key2 set2)
(damse1 damse1 set2)
(TRUSTS Hero Villain True world)
(EXCLUSIVE *agent (POSSESS *object) True *agent (POSSESS
    *object) False *object)
(EXCLUSIVE *agent (POSSESS *object) False *agent (POSSESS
    *object) True *object)
(EXCLUSIVE Hero (POSSESS *object) True Villain (POSSESS
    *object) True *object)
(EXCLUSIVE Villain (POSSESS *object) True Hero (POSSESS
    *object) True *object)
(GOAL Hero (POSSESS key2) False set2)
(GOAL Hero (POSSESS key2) False set3)
(INROOM room2 Hero True set1)
(INROOM room2 key1 True set1)
(INROOM room2 Villain True set1)
(INROOM room3 damse1 True set1)
(HOLDING key1 Hero True set1)
(UNLOCKS doora key1 True set1)
(UNLOCKS doora key2 True set1)
(OPEN doora False set1)
(OPEN doorc False set1)
(OPEN doord True set1)
(INROOM room2 Villain True set2)

```

```

(INROOM room2 Hero True set2)
(INROOM room2 key1 True set2)
(INROOM room4 key2 True set2)
(INROOM room3 damsel True set2)
(UNLOCKS doora key2 True set2)
(UNLOCKS doora key1 False set2)
(UNLOCKS doora key2 True set5)
(OPEN doora False set2)
(OPEN doord True set2)
(OPEN doorc False set2)
(BLOCKED doorc True set2)
(BLOCKING doorc Villain True set2)
(SCHEDULE Hero (
    (OPEN_UP doora)
    (MOVE_TO room1)
    (MOVE_TO room3)
) 1)
(SCHEDULE Villain ((WAIT)) 1)

```

What follows is a partial listing of the output of the simulation. We have edited the lengthy output to show only the statements that illustrate something interesting about a particular world. The worlds themselves have also been edited, and narration has been added to explain what is occurring.

The simulated world contains five rooms, four doors, a Hero, a Villain, a Damsel, and two keys. Figure 16 shows the layout of the initial world. The Hero and Villain are initially in room2 and the

Hero is holding key1. The damsel is in room3, and key2 is in room4. Doora connecting room2 and room1 is locked. Doorc connecting room2 and room4 is closed. We should also note that room5 is known as room5 to the Hero, but it is called the Dungeon by the Villain, and the Hero is aware of this difference in vocabulary.

The Hero's initial schedule involves opening doora, moving to room1, and then moving through doorb to be in room3 with the damsel. He does not realize that the Villain is actually a villain who has locked the damsel away.

The simulation is started by asserting that the agents are not busy.

cf: (BUSY False Hero 1)

This causes the Hero to prepare for the next (first, in this case) action on his schedule. Note that the instantiated rules are output when they fire; the actual rules contain variables as arguments.

Rule:g329

```
(BUSY False Hero 1)
((SCHEDULE Hero ((OPEN_UP doora) (MOVE_TO room1)
(MOVE_TO room3)) 1) (0 100) g438 5)
(((<> ((MOVE_TO room1) (MOVE_TO room3)) ())) (0 100) a2
0)

((= 1 1) (0 100) a1 0)
((TIME_REQUIRED (OPEN_UP doora) 1)·(0 100) g356 0)
((add 1 1 2) (0 100) a7 0)
((add 1 2 3) (0 100) a7 0)
->
```

```
(SCHEDULE Hero ((MOVE_TO room1) (MOVE_TO  
room3))) 3)
```

```
(PERFORM (OPEN_UP doora) Hero 2)
```

```
cf: (SCHEDULE Hero ((MOVE_TO room1) (MOVE_TO room3))) 3)
```

```
cf: (PERFORM (OPEN_UP doora) Hero 2)
```

```
cf: (BUSY False Villain 1)
```

The Villain initially has nothing to do, so he will continue to wait.

Rule:g335

```
(BUSY False Villain 1)
```

```
((SCHEDULE Villain ((WAIT)) 1) (0 100) g439 5)
```

```
((= 1 1) (0 100) a1 0)
```

```
((TIME_REQUIRED (WAIT) 1) (0 100) g358 0)
```

```
((add 1 1 2) (0 100) a7 0)
```

```
((add 1 2 3) (0 100) a7 0)
```

->

```
(SCHEDULE Villain ((WAIT)) 3)
```

```
(PERFORM (WAIT) Villain 2)
```

We also assert as an initial counterfactual that the Villain knows the Hero wants doora unlocked. The Villain's initial goal is for the door to remain locked.

```
cf: (GOAL Hero (LOCKED doora) False set5)
```

The Villain knows that the Hero has the wrong key to unlock the door, and he reasons that the Hero will want to possess key2 (the correct key). Thus, some goals are input directly as counterfactuals while others are inferred.

Rule:g220

```
(GOAL Hero (LOCKED doora) False set5)
((UNLOCKS doora key2 True set5) (0 100) g432 1)
((SET Villain Hero () set5) (0 100) g367 0)

->

(GOAL Hero (POSSESS key2) True set5)
```

cf: (GOAL Hero (POSSESS key2) True set5)

Since the Villain does not want the Hero to possess key2 and be able to unlock doora, the Villain recognizes a conflict.

Rule:g190

```
(GOAL Hero (POSSESS key2) True set5)
((SET Villain () () set2) (0 100) g364 0)
((SET Villain Hero () set5) (0 100) g367 0)
((EXCLUSIVE Hero (POSSESS key2) False Hero (POSSESS
key2) True key2) (0 100) g410 1)
((GOAL Hero (POSSESS key2) False set2) (0 100) g413 1)

->

(CONFLICT1 key2 Villain Hero set2)
```

cf: (CONFLICT1 key2 Villain Hero set2)

Rule:g215

```
(CONFLICT1 key2 Villain Hero set2)
((NO_CONFLICTS key2 Villain Hero set2) (0 100) g401 1)

->

(inconsistent)
```

Since the Villain knows that key2 is in room4, he revises his schedule so that he will block the door to room4 (doorc).

Rule:g213

```
((CONFLICT1 key2 Villain Hero set2)
((GOAL Hero (POSSESS key2) True set5) (1 100) g449 -1)
((SET Villain () () set2) (0 100) g364 0)
((SET Villain Hero () set5) (0 100) g367 0)
((AFRAID_OF Villain Hero False world) (0 100) g369 1)
((INROOM room4 key2 True set2) (0 100) g428 1)
((INROOM room2 Villain True world) (0 100) g380 1)
((INROOM room2 Hero True world) (0 100) g370 1)
((CONNECTS doorc room2 room4 True world) (0 100) g347
0)

((BLOCKED doorc False world) (0 100) g388 1)
```

->

```
(REVISE SCHEDULE Villain ((BLOCK doorc)))
```

cf: (REVISE SCHEDULE Villain ((BLOCK doorc)))

The first possible world generated is:

Possible World : w1

```
1 100 (BUSY False Hero 1)
1 100 (SCHEDULE Hero ((MOVE_TO room1) (MOVE_TO room3)) 3)
1 100 (PERFORM (OPEN_UP doora) Hero 2)
1 100 (BUSY False Villain 1)
1 100 (SCHEDULE Villain ((WAIT)) 3)
1 100 (PERFORM (WAIT) Villain 2)
1 100 (GOAL Hero (LOCKED doora) False set5)
1 100 (GOAL Hero (POSSESS key2) True set5)
1 100 (CONFLICT1 key2 Villain Hero set2)
```

```
1 100 (not (NO_CONFLICTS key2 Villain Hero set2))
1 100 (REVISE SCHEDULE Villain ((BLOCK doorc)))
```

At the next time cycle, the Hero performs the action of opening the door and the Villain changes his schedule so that (BLOCK doorc) is the next action on his schedule.

```
cf: (SCHEDULE Villain ((BLOCK doorc) (WAIT)) 3)
```

```
cf: (ACT (OPEN_UP doora) Hero 2)
```

Since the door is locked, the Hero fails but he realizes that the door is locked and sets a new goal to have the door unlocked.

Rule:g50

```
(ACT (OPEN_UP doora) Hero 2)
((TIME 2) (2 100) g461 -1)
((= 2 2) (0 100) a1 0)
((INROOM room2 Hero True world) (0 100) g370 1)
((LOCKED doora True world) (0 100) g376 1)
((BLOCKED doora False world) (0 100) g386 1)
((SET Hero () () set1) (0 100) g363 0)
((CONNECTS doora room2 room1 True world) (0 100) g344 0)
```

->

```
(FAILED (OPEN_UP doora) Hero 2)
(GOAL Hero (LOCKED doora) False set1)
(REPLAN ((UNLOCK doora) (OPEN_UP doora)) Hero 2)
(LOCKED doora True set1)
```

```
cf: (FAILED (OPEN_UP doora) Hero 2)
```

```
cf: (GOAL Hero (LOCKED doora) False set1)
```

The Hero revises his schedule to unlock the door.

cf: (REPLAN ((UNLOCK doora) (OPEN_UP doora)) Hero 2)

Rule:g142

(REPLAN ((UNLOCK doora) (OPEN_UP doora)) Hero 2)

((TIME 2) (2 100) g461 -1)

((= 2 2) (0 100) a1 0)

->

(REVISE SCHEDULE Hero ((UNLOCK doora) (OPEN_UP doora)))

cf: (REVISE SCHEDULE Hero ((UNLOCK doora) (OPEN_UP doora)))

cf: (LOCKED doora True set1)

cf: (BUSY True Villain 2)

cf: (BUSY True Hero 2)

The time indicates that we are in the second time cycle.

cf: (TIME 2)

Since we are at TIME 2, it can no longer be TIME 1.

Rule:g338

(TIME 2)

((TIME 1) (0 100) g392 1)

((<> 2 1) (0 100) a2 0)

->

(inconsistent)

cf: (not (TIME 1))

Possible World : w2

2 100 (SCHEDULE Villain ((BLOCK doorc) (WAIT)) 3)

2 100 (not (SCHEDULE Villain ((WAIT)) 3))

```

2 100 (ACT (WAIT) Villain 2)
2 100 (SUCCESSFUL (WAIT) Villain 2)
2 100 (ACT (OPEN_UP doora) Hero 2)
2 100 (FAILED (OPEN_UP doora) Hero 2)
2 100 (GOAL Hero (LOCKED doora) False set1)
2 100 (REPLAN ((UNLOCK doora) (OPEN_UP doora)) Hero 2)
2 100 (REVISE SCHEDULE Hero ((UNLOCK doora) (OPEN_UP
                                                                    doora)))

2 100 (LOCKED doora True set1)
2 100 (BUSY True Villain 2)
2 100 (BUSY True Hero 2)
2 100 (TIME 2)
2 100 (not (TIME 1))

```

The Villain prepares to block doorc (i.e., he takes the next action from his schedule and sets a time for it to occur).

cf: (PERFORM (BLOCK doorc) Villain 4)

Possible World : w3

```

3 100 (PERFORM (BLOCK doorc) Villain 4)
3 100 (TIME 3)
3 100 (not (TIME 2))

```

The Villain begins blocking doorc at TIME 4. This alters facts about the world about the status of the door.

cf: (ACT (BLOCK doorc) Villain 4)

Rule:gl13

```

(ACT (BLOCK doorc) Villain 4)
((SET Villain () () set2) (0 100) g364 0)
((INROOM room2 Villain True world) (0 100) g380 1)
((TIME 4) (4 100) g491 -1)
((= 4 4) (0 100) a1 0)
((CONNECTS doorc room2 room4 True world) (0 100) g347
0)

```

->

```

(SUCCESSFUL (BLOCK doorc) Villain 4)
(BLOCKED doorc True set2)
(BLOCKING doorc Villain True set2)
(BLOCKED doorc True world)
(BLOCKING doorc Villain True world)

```

The Hero attempts to unlock doora but fails. He then changes his belief about key1 unlocking doora. Since he knows another key exists, he revises his schedule to find and possess that key (key2).

Possible World : w4

```

4 100 (ACT (BLOCK doorc) Villain 4)
4 100 (SUCCESSFUL (BLOCK doorc) Villain 4)
4 100 (BLOCKED doorc True set2)
4 100 (BLOCKING doorc Villain True set2)
4 100 (BLOCKED doorc True world)
4 100 (not (BLOCKED doorc False world))
4 100 (BLOCKING doorc Villain True world)
4 100 (BLOCKING doorc Villain True set1)
4 100 (BLOCKED doorc True set1)

```



```
4 100 (ACT (UNLOCK doora) Hero 4)
4 100 (FAILED (UNLOCK doora) Hero 4)
4 100 (REPLAN ((FIND key2) (UNLOCK doora)) Hero 4)
4 100 (REVISE SCHEDULE Hero ((FIND key2) (UNLOCK doora)))
4 100 (GOAL Hero (POSSESS key2) True set1)
4 100 (CONFLICT1 key2 Hero Villain set1)
4 100 (not (NO_CONFLICTS key2 Hero Villain set1))
4 100 (UNLOCKS doora key1 False set1)
4 100 (not (UNLOCKS doora key1 True set1))
4 100 (BUSY True Villain 4)
4 100 (BUSY True Hero 4)
4 100 (TIME 4)
4 100 (not (TIME 3))
```

Since the Hero trusts the Villain and they are in the same room, he asks the Villain about the location of key2. He also realizes that the Villain will then know that the Hero wants key2.

The Villain lies to the Hero about the location of key2 because he does not want him to have it. He tells the Hero that the key is in the dungeon. This report does not immediately have meaning for the Hero. However, the Hero realizes that the Villain uses the term 'dungeon' to denote room5 and makes the substitution.

The Hero then takes the information that key2 is in room5 to be new information, and he accepts it as his own belief because he knows of no conflicts that would cause the Villain to lie to him. The Hero therefore revises his schedule to move to room5 and retrieve the key.

Possible World : w6

```
6 100 (ACT (ASK Villain (INROOM where key2)) Hero 6)
6 100 (GOAL Hero (POSSESS key2) True set4)
6 100 (INPUT QUERY (INROOM where key2) Hero Villain 6)
6 100 (ACT (SPEAK_TO Hero (INROOM DUNGEON key2)) Villain 6)
6 100 (INPUT REPORT (INROOM DUNGEON key2) Villain Hero 6)
6 100 (INPUT REPORT (INROOM room5 key2) Villain Hero 6)
6 100 (NEW_INFO 6 INROOM room5 key2 set1)
6 100 (REVISE SCHEDULE Hero ((MOVE_TO room5)))
6 100 (INROOM room5 key2 True set1)
6 100 (TIME 6)
```

When the Hero moves to room5, he and the key he carries move into room5. This causes changes to his own beliefs, the real world, and the Villain's beliefs. The Hero perceives that key2 is not where he expected it, and this causes him to believe that the Villain must want to possess the key and that there is a conflict over the key which they are both now aware of. The Hero revises his schedule to go back to room2 to again ask about the key.

Possible World : w8

```
8 100 (ACT (MOVE_TO room5) Hero 8)
8 100 (ENTER 8 room5 Hero)
8 100 (INROOM room5 key2 False set1)
8 100 (not (INROOM room5 key2 True set1))
8 100 (INROOM room5 key1 True world)
8 100 (not (INROOM room2 key1 True world))
```

```

8 100 (INROOM room5 key1 True set1)
8 100 (not (INROOM room2 key1 True set1))
8 100 (INROOM room5 Hero True set1)
8 100 (not (INROOM room2 Hero True set1))
8 100 (INROOM room2 Hero False world)
8 100 (not (INROOM room2 Hero True world))
8 100 (INROOM room5 Hero True world)
8 100 (REVISE SCHEDULE Hero ((MOVE_TO room2) (FIND key2)))
8 100 (GOAL Villain (POSSESS key2) True set3)
8 100 (CONFLICT2 key2 Hero Villain set1)
8 100 (CONFLICT2 key2 Hero Villain set3)
8 100 (INROOM room5 Hero True set2)
8 100 (not (INROOM room2 Hero True set2))
8 100 (INROOM room2 Villain True set1)
8 100 (SUCCESSFUL (MOVE_TO room5) Hero 8)
8 100 (TIME 8)

```

The Hero asks the Villain about key2 once again.

cf: (ACT (ASK Villain (INROOM where key2)) Hero 12)

Rule:g239

```

(ACT (ASK Villain (INROOM where key2)) Hero 12)
((INROOM room2 Hero True world) (10 100) g636 1)
((INROOM room2 Villain True world) (0 100) g380 1)
(((<> Hero Villain) (0 100) a2 0)
((TIME 12) (12 100) g661 -1)
((= 12 12) (0 100) a1 0)

```

->

(INPUT QUERY (INROOM where key2) Hero Villain 12)

The Villain again lies to the Hero about the key because he knows of the conflict over it.

cf: (INPUT QUERY (INROOM where key2) Hero Villain 12)

Rule:g266

(INPUT QUERY (INROOM where key2) Hero Villain 12)

((TIME 12) (12 100) g661 -1)

((= 12 12) (0 100) a1 0)

((SET Villain () () set2) (0 100) g364 0)

((CONFLICT1 key2 Villain Hero set2) (1 100) g451 2)

((INROOM room4 key2 True set2) (0 100) g428 1)

((INROOM room2 Hero True world) (10 100) g636 1)

((CONNECTS doord room2 DUNGEON True set2) (0 100) g353

0)

((OPEN doord True world) (0 100) g375 1)

((<> room4 DUNGEON) (0 100) a2 0)

->

(ACT (SPEAK_TO Hero (INROOM DUNGEON key2)) Villain 12)

cf: (ACT (SPEAK_TO Hero (INROOM DUNGEON key2)) Villain 12)

Rule:g233

(ACT (SPEAK_TO Hero (INROOM DUNGEON key2)) Villain 12)

((INROOM room2 Villain True world) (0 100) g380 1)

((INROOM room2 Hero True world) (10 100) g636 1)

((<> Villain Hero) (0 100) a2 0)

((TIME 12) (12 100) g661 -1)

((= 12 12) (0 100) a1 0)

->

(INPUT REPORT (INROOM DUNGEON key2) Villain Hero 12)

cf: (INPUT REPORT (INROOM DUNGEON key2) Villain Hero 12)

The Hero again makes the substitution of room5 for dungeon because the Villain tells him the same location again for key2.

Rule:g281

(INPUT REPORT (INROOM DUNGEON key2) Villain Hero 12)

((TIME 12) (12 100) g661 -1)

((= 12 12) (0 100) a1 0)

((SET Hero () () set1) (0 100) g363 0)

((SET Hero Villain () set3) (0 100) g365 0)

((denotes room5 (room5) set1) (0 100) g398 1)

((denotes DUNGEON (room5) set3) (0 100) g399 1)

((denotes key2 (key2) set1) (0 100) g396 1)

((denotes key2 (key2) set3) (0 100) g397 1)

->

(INPUT REPORT (INROOM room5 key2) Villain Hero 12)

cf: (INPUT REPORT (INROOM room5 key2) Villain Hero 12)

Since there is a contradiction between what the Villain says and what the Hero knows (from actually having gone to room5) and since the Hero now believes there to be a conflict over the key (this was determined by another rule in world w8), the Hero concludes this time that the Villain is lying to him.

Rule:g288

(INPUT REPORT (INROOM room5 key2) Villain Hero 12)

```
((TIME 12) (12 100) g661 -1)
((= 12 12) (0 100) a1 0)
((SET Hero () () set1) (0 100) g363 0)
((denotes room5 (room5) set1) (0 100) g398 1)
((CONFLICT2 key2 Hero Villain set1) (8 100) g594 2)
((CONTRADICTION set2 (INROOM room5 key2)) (0 100) a14 0)
```

->

```
(LYING Villain key2 set1)
```

cf: (LYING Villain key2 set1)

Since the Hero believes that the Villain is lying to him and since the Hero has observed that the Villain is blocking doorc to room4, the Hero concludes that key2 is in room4. He revises his schedule to shove the Villain out of the way in order to open doorc and move into room4.

Rule:g298

```
(LYING Villain key2 set1)
((SET Hero () () set1) (0 100) g363 0)
((BLOCKING doorc Villain True set1) (4 100) g497 1)
((INROOM room2 Hero True world) (10 100) g636 1)
((INROOM room2 Villain True world) (0 100) g380 1)
((CONNECTS doorc room2 room4 True world) (0 100) g347 0)
```

->

```
(INROOM key2 room4 True set1)
(REVISE SCHEDULE Hero ((SHOVE Villain) (OPEN_UP doorc)
```

```
(MOVE_TO room4))
```

cf: (INROOM key2 room4 True set1)

cf: (REVISE SCHEDULE Hero ((SHOVE Villain) (OPEN_UP doorc)

(MOVE_TO room4)))

Also, since the Hero believes the Villain is lying, he no longer trusts the Villain.

Rule:g290

(LYING Villain key2 set1)

((SET Hero () () set1) (0 100) g363 0)

->

(TRUSTS Hero Villain False world)

cf: (TRUSTS Hero Villain False world)

Rule:g291

(TRUSTS Hero Villain False world)

((TRUSTS Hero Villain True world) (0 100) g408 1)

->

(inconsistent)

cf: (not (TRUSTS Hero Villain True world))

Possible World : w12

12 100 (ACT (ASK Villain (INROOM where key2)) Hero 12)

12 100 (INPUT QUERY (INROOM where key2) Hero Villain 12)

12 100 (ACT (SPEAK_TO Hero (INROOM DUNGEON key2)) Villain
12)

12 100 (INPUT REPORT (INROOM DUNGEON key2) Villain Hero 12)

12 100 (INPUT REPORT (INROOM room5 key2) Villain Hero 12)

12 100 (LYING Villain key2 set1)

12 100 (INROOM key2 room4 True set1)

12 100 (TRUSTS Hero Villain False world)

12 100 (not (TRUSTS Hero Villain True world))

12 100 (TIME 12)

When the Hero shoves the Villain, the Villain moves so that the door is no longer blocked (and both agents perceive this). The Villain is afraid of the Hero after being shoved and does not attempt to block the door again.

cf: (ACT (SHOVE Villain) Hero 14)

Possible world: w14

14 100 (ACT (SHOVE Villain) Hero 14)
14 100 (SUCCESSFUL (SHOVE Villain) Hero 14)
14 100 (AFRAID_OF Villain Hero True world)
14 100 (not (AFRAID_OF Villain Hero False world))
14 100 (BLOCKED doorc False world)
14 100 (not (BLOCKED doorc True world))
14 100 (BLOCKED doorc False set1)
14 100 (not (BLOCKED doorc True set1))
14 100 (BLOCKED doorc False set2)
14 100 (not (BLOCKED doorc True set2))
14 100 (not (BLOCKED doorc True set2))
14 100 (BLOCKING doorc Villain False set1)
14 100 (not (BLOCKING doorc Villain True set1))
14 100 (BLOCKING doorc Villain False world)
14 100 (not (BLOCKING doorc Villain True world))
14 100 (TIME 14)

The Hero opens up doorc successfully.

Possible World : w16

```
16 100 (ACT (OPEN_UP doorc) Hero 16)
16 100 (OPEN doorc True world)
16 100 (not (OPEN doorc False world))
16 100 (OPEN doorc True set1)
16 100 (not (OPEN doorc False set1))
16 100 (OPEN doorc True set2)
16 100 (not (OPEN doorc False set2))
```

The Hero moves into room4 and perceives that key2 is there. Since his goal is to possess key2, he revises his schedule to pick up the key.

Possible World : w18

```
18 100 (ACT (MOVE_TO room4) Hero 18)
18 100 (ENTER 18 room4 Hero)
18 100 (REVISE SCHEDULE Hero ((PICK_UP key2)))
```

When the Hero picks up the key he is then holding the key.

cf: (ACT (PICK_UP key2) Hero 20)

Rule:g107

```
(ACT (PICK_UP key2) Hero 20)
((SET Hero () () set1) (0 100) g363 0)
((TIME 20) (20 100) g812 -1)
((= 20 20) (0 100) a1 0)
```

->

```
(HOLDING key2 Hero True world)
```

(HOLDING key2 Hero True set1)

(SUCCESSFUL (PICK_UP key2) Hero 20)

cf: (HOLDING key2 Hero True world)

Possible world: w20

20 100 (ACT (PICK_UP key2) Hero 20)

20 100 (HOLDING key2 Hero True world)

20 100 (not (HOLDING key2 Hero False world))

20 100 (HOLDING key2 Hero True set1)

Since the Hero has the key he needs, he returns to the room containing the door he wishes to unlock.

Possible World : w22

22 100 (ACT (MOVE_TO room2) Hero 22)

Using the correct key, the Hero successfully unlocks doora.

Possible World : w24

24 100 (ACT (UNLOCK doora) Hero 24)

24 100 (LOCKED doora False world)

24 100 (not (LOCKED doora True world))

24 100 (LOCKED doora False set1)

24 100 (not (LOCKED doora True set1))

24 100 (SUCCESSFUL (UNLOCK doora) Hero 24)

The Hero then opens the door and moves to room1.

Possible World : w28

28 100 (ACT (MOVE_TO room1) Hero 28)

Finally, the Hero moves through doorb into room3, where the damsel is waiting.

Possible World : w30

30 100 (ACT (MOVE_TO room3) Hero 30)

30 100 (ENTER 30 room3 Hero)

30 100 (INROOM room3 Hero True set1)

Thus, starting with a simple schedule of opening a door and moving through two rooms, the Hero completed the relatively complex task of obtaining the correct key and unlocking the door. The Hero performed actions, changed his beliefs, perceived the results of actions, revised his plans, communicated with another agent, and reasoned about the other agent's beliefs.

14. Conclusions

14.1 Contributions of This Thesis

We have thoroughly described a system that enables multiple agents to reason about their own beliefs and the beliefs of other agents. The major new contribution of this thesis is a belief revision system that treats beliefs as objects in the real world, allows reasoning to result in new beliefs just as physical events result in new states of the world, and enables agents to revise their beliefs in a plausible way automatically when an inconsistency occurs.

The possible worlds semantics we have used in conjunction with counterfactual theory enables an agent to maintain consistency in his own beliefs and his nested beliefs independently. We have incorporated opinions into the system that aid an agent in deciding how to resolve inconsistencies; these opinions indicate the agent's uncertainty about a belief and his justification for that uncertainty.

The representation we have developed enables us to use the Virginia Tech counterfactual engine to implement our theory. It also gives us flexibility because the counterfactual rules may be tailored to fit a specific problem domain, and they may be written to operate independently for different agents.

We have described how deception may be carried out or detected by an agent. Also, we have specified a model for agents to communicate, perceive, and relate their beliefs to actions. The system described has the following capabilities:

- 1) Multiple agents can reason with their own beliefs and can reason about other agents' beliefs.
- 2) Agents can reason about their past beliefs. This is important because many decisions are biased by an agent's memory of prior events. Many systems do not contain a memory of past worlds.
- 3) Our representation and ontology permit agents to reason about many states of belief, including beliefs that they are missing information.
- 4) Agents base the uncertainty of an inference on its support, and they use uncertainty as a basis for resolving conflicts.
- 5) Agents may reason about many different types of existence.
- 6) The system solves the problem of referential transparency for beliefs by allowing substitutions only when the agent is aware that two concepts map to the same objects, relations, and attributes in the actual world.
- 7) The counterfactual rules can be written to include specific information about a problem domain, particularly relations between action and belief.
- 8) The counterfactual rules can be written so that different agents may use different rules for reasoning. Agents may also reason with their nested beliefs using different rules than they used to reason with their own beliefs.
- 9) Agents can use metalevel reasoning to determine when nested beliefs should be revised.

- 10) Agents can analyze goals to determine if there is a conflict with another agent.
- 11) Agents can perform some elementary deception and detection of deception.
- 12) Agents can communicate with each other and perceive information from the world.

We have also thoroughly defined the system formally in terms of tense, modal, and event logics and a theory of causality. The most important aspect of this thesis may be that it provides a formal basis of a theory that can also be implemented practically. A theory is not as useful if it cannot be applied.

14.2. Future Work

There are several improvements that could be made upon the current simulator. A model of communication is needed that can interpret natural language utterances and can account for metacommunication and emotional aspects. Also, a complex model is needed for the process of perception; the simulator must be able to determine how an agent perceives what he sees.

There are other aspects of this thesis that might be improved on. There may be better rules for combining opinions that could be developed, or a richer theory of opinions and justifications may be developed that can be implemented practically.

A better solution to the persistence problem is needed that can account for states that do not persist over long periods of time (e.g.,

a person's age, the weather). Also, it would be useful for the system to permit agents to learn new rules.

Another goal is to develop a planner in the counterfactual engine that could be integrated into the simulation with the belief reasoning system. This would allow agents to make lengthy plans in advance rather than merely react to stimuli. Although agents can currently reason about actions, they should also be able to reason about causality.

In addition, the system would be improved by the inclusion of a goal recognition system. This in itself is a complex problem, but we have already implemented some elementary goal reasoning that uses beliefs about the world and about another agent's goals to predict additional goals of the agent. Thus, there are many areas in which work could be done to extend the system.

We have developed a basic, functional system on which to improve and build. Although many needed improvements (such as planning and goal recognition) will require a great deal of further research, the final system achieved by including these improvements would be a powerful, autonomous, and flexible plan recognition system.

15. Literature Cited

- [Anderson and Belnap, 1975] Anderson, Alan and Nuel Belnap Jr. Entailment: The Logic of Relevance and Necessity. Princeton, New Jersey: Princeton University Press, 1975.
- [Barwise, 1989] Barwise, Jon. The Situation in Logic. Stanford University: Center for the Study of Language and Information, 1989]
- [Barwise and Perry, 1983] Barwise, Jon and John Perry. Situations and Attitudes. Cambridge, Massachusetts: MIT Press, 1983.
- [Bernays-Schonfinkel, 1928] Bernays, P. and M. Schonfinkel. "Zum Entscheidungsproblem der mathematischen Logik," M.A. 99, 1928.
- [Bowyer, 1982] Bowyer, J. Barton. Cheating. New York: St. Martin's Press, 1982.
- [Bradley and Swartz, 1979] Bradley, R. and N. Swartz. Possible Worlds: an Introduction to Logic and its Philosophy. Indianapolis, Indiana: Hackett Publishing Company, 1979.
- [Cohen, 1985] Cohen, Paul R. Heuristic Reasoning About Uncertainty. London: Pitman Publishing Limited, 1985.
- [Dubois and Prade, 1988] Dubois, Didier and Henri Prade. "Representation and combination of uncertainty with belief functions and possibility measures" *Computation Intelligence*, Vol. 4, Number 3, 1988.
- [Fikes and Nilsson, 1971] Fikes, R.E. and N.J. Nilsson "STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving," in Artificial Intelligence. Volume 2 Number 3-4, 1971.
- [Frege, 1892] Frege, G. "On Concept and Object," in Translations from the Philosophical Writings of Gottlob Frege edited by Peter Geach and Max Black. Oxford: Basil Blackwell, 1952.
- [Frege, 1950] Frege, G. The Foundations of Arithmetic: A logico-mathematical inquiry into the concept of number. Oxford: Basil Blackwell, 1950.
- [Gardenfors, 1988] Gardenfors, Peter. Knowledge in Flux. Cambridge, Massachusetts: MIT Press, 1988.
- [Gat, 1987] Gat, Erann. Reasoning About Knowledge Using Extensional Logics. Master's Thesis, Department of Computer Science, Virginia Tech, 1987.
- [Genesereth and Nilsson, 1987] Genesereth, M. and N. Nilsson.

- Logical Foundations of Artificial Intelligence. Palo Alto, California: Morgan Kaufmann Publishers, Inc., 1987.
- [Haack, 1978] Haack, Susan. Philosophy of Logics. Cambridge, England: Cambridge University Press, 1978.
- [Harman, 1986] Harman, G. Change in View: Principles of Reasoning. Cambridge, Massachusetts: M.I.T. Press, 1986.
- [Hayes, 1973] Hayes, Patrick J. "The Frame Problem and Related Problems in Artificial Intelligence" in Artificial and Human Thinking, edited by Alick Elithorn and David Jones, San Francisco, California: Jossey-Bass Inc., Publishers, 1973.
- [Hirst, 1989] Hirst, Graeme. "Ontological assumptions in knowledge representation," Department of Computer Science, M5S 1A4, University of Toronto, November, 1989.
- [Hughes and Cresswell, 1968] Hughes, G. E. and M. J. Cresswell. An Introduction to Modal Logic. London: Methuen and Co. LTD, 1968.
- [Johnson-Laird, 1983] Johnson-Laird, P. Mental Models. Cambridge, Massachusetts: Harvard University Press, 1983.
- [Khambati, 1989] Khambati, Khozema. "WIFII: A Hypothetical Time Reasoning System," User's manual, Department of Computer Science, Virginia Tech, 1989.
- [Konolige, 1985] Konolige, Kurt. "Belief and Incompleteness" in Formal Theories of the Commonsense World edited by Jerry Hobbs and Robert C. Moore. Norwood, New Jersey: Ablex Publishing Corporation, 1985.
- [Kowalski, 1979] Kowalski, R. Logic for Problem Solving. New York: North Holland, 1979
- [Lakoff, 1988] Lakoff, George. Women, Fire, and Dangerous Things: What Categories Reveal About the Mind. Chicago: University of Chicago Press, 1988.
- [Landman, 1986] Landman, Fred. Towards a Theory of Information. Dordrecht, The Netherlands: Foris Publications Holland, 1986.
- [Langholm, 1988] Langholm, Tore. Partiality, Truth and Persistence. Stanford, California: Center for the Study of Language and Information, Stanford University, 1988.
- [Lewis, 1986a] Lewis, David. On the Plurality of Worlds. New York: Basil Blackwell Inc., 1986.

- [Lewis, 1986b] Lewis, David. Philosophical Papers, Vol. II. Oxford: Oxford University Press, 1986.
- [Maida and Shapiro, 1985] Maida, Anthony and Stuart Shapiro. "Intensional Concepts in Propositional Semantic Networks"
- [Martins and Shapiro, 1988] Martins, J. and S. Shapiro. "A Model for Belief Revision," Artificial Intelligence Volume 35 Number 1, May 1988.
- [McCarthy, 1962] McCarthy, J. et. al. LISP 1.5 Programmer's Manual. Cambridge, Massachusetts: The M.I.T. Press, 1962.
- [McCarthy, 1985a] McCarthy, John. "First Order Theories of Individual Concepts" in Readings in Knowledge Representation edited by Ronald Brachman and Hector Levasque. Los Altos, California: Morgan Kaufmann Publishers, Inc., 1985.
- [McCarthy, 1985b] McCarthy, John. "Epistemological Problems of Artificial Intelligence" in Readings in Knowledge Representation edited by Ronald Brachman and Hector Levasque. Los Altos, California: Morgan Kaufmann Publishers, Inc., 1985.
- [McDermott, 1982] McDermott, D. "Non-monotonic logic II: non-nontonic modal theories" in JACM 29 (1).
- [Moore, 1985] Moore, Robert C. "A Formal Theory of Knowledge and Action" in Formal Theories of the Commonsense World edited by Jerry Hobbs and Robert C. Moore. Norwood, New Jersey: Ablex Publishing Corporation, 1985.
- [Meinong, 1904] Meinong, Alexius. "Uber Gegenstanstheorie" in Untersuchungen zur Gegenstands theorie und Psychologie edited by Alexius Meinong. Barth, 1904.
- [Monk, 1976] Monk, J. Donald. Mathematical Logic. New York: Springer-Verlag, 1976.
- [Montague, 1973] Montague, R. "The proper treatment of quantification in ordinary English" in Approaches to Natural Language: Proceedings of the 1970 Stanford Workshop on Grammar and Semantics. Dordrecht: Riedel, 1973.
- [Nutter, 1983] Nutter, J.T. "Default Reasoning in AI Systems." SUNY at Buffalo Computer Science Technical Report 204, 1983.
- [Olsen, 1987] Olsen, Kenneth. An Essay on Facts. Stanford, California: Center for the Study of Language and Information, Stanford University, 1987.
- [Pendergraft, 1988] Pendergraft, James O. Planning with

- Hypothetical Reasoning. Master's Thesis, Department of Computer Science, Virginia Tech, 1988.
- [Prior, 1966] Prior, A. N. "Postulates for tense-logic" in American Philosophical Quarterly, vol. 3, 1966
- [Reiter, 1977] Reiter, Raymond. "On Closed World Data Bases" in Logic and Data Bases edited by Herve Gallaire and Jack Minker. New York: Plenum Press, 1977.
- [Rescher, 1964] Rescher, Nicholas. Hypothetical Reasoning. Amsterdam, Holland: North-Holland 1964.
- [Rescher, 1976] Rescher, Nicholas. Plausible Reasoning. Amsterdam, The Netherlands: Van Gorcum, Assen, 1976.
- [Rescher and Urquhart, 1971] Rescher, Nicholas and Alasdair Urquhart. Temporal Logic. Berlin, West Germany: Springer-Verlag, 1971.
- [Ross and Anderson, 1982] Ross, L. and C.A. Anderson. "Shortcomings in the attribution process: On the origins and maintenance of erroneous social assessments," in Judgement under Uncertainty: Heuristics and Biases. edited by D. Kahneman, P. Slovic, and A. Tversky. Cambridge: Cambridge University Press, 1982.
- [Simon and Rescher, 1966] Simon, H. A. and N. Rescher. "Cause and Counterfactual" in Philosophy of Science, 33, Bruges, Belgium, 1966.
- [Turner, 1984] Turner, Raymond. Logics for Artificial Intelligence. West Sussex, England: Ellis Horwood Limited, 1984.
- [Vasek, 1986] Vasek, Marie. "Lying as a Skill: The Development of Deception in Children" in Deception perspectives on human and non-human deceit edited by Robert W. Mitchell and Nicholas S. Thompson. Albany, New York: State University of New York Press, 1986.
- [Venkateshwaran, 1990] Venkateshwaran, R. A Temporal Analysis of Natural Language Narrative Text. Thesis, Department of Computer Science, Virginia Tech, 1990.
- [Virkar, forthcoming] Virkar, R. Forthcoming Doctoral Dissertation, Department of Computer Science, Virginia Tech.
- [Whitehead, 1988] Whitehead, J. Douglas H. Fault Diagnosis Based on Causal Reasoning. Master's Thesis, Department of Computer Science, Virginia Tech, 1988.

VITA

Brian Klock was born in Bethesda, MD on June 5, 1967 and was raised in Rockville, MD. He received a B.S. in electrical engineering in 1988 and an M.S. in electrical engineering in 1990 from Virginia Tech. While at Virginia Tech, he was a member of the student chapters of IEEE and ISHM. He was also a member of Tau Beta Pi, Eta Kappa Nu, and Gamma Beta Phi honor societies. He currently resides in Gaithersburg, MD and is employed at the U. S. Patent and Trademark Office. He is also enrolled in the evening division at the University of Maryland School of Law.

A handwritten signature in black ink, appearing to read "Brian Klock". The signature is fluid and cursive, with a long horizontal stroke extending to the right.