# Chapter 1

# Automating Common Sense

> Most of their remarks were the sort it would not be easy to disagree with: "What I always say is, when a chap's hungry, he likes some victuals," or "Getting dark now; always does at night," or even, "Ah, you've come over the water. Powerful wet stuff, ain't it?"
>
> — C.S. Lewis, *The Voyage of the Dawn Treader*

In order for an intelligent creature to act sensibly in the real world, it must know about that world and be able to use its knowledge effectively. The common knowledge about the world that is possessed by every schoolchild and the methods for making obvious inferences from this knowledge are called common sense. Commonsense knowledge and commonsense reasoning are involved in most types of intelligent activities, such as using natural language, planning, learning, high-level vision, and expert-level reasoning. How to endow a computer program with common sense has been recognized as one of the central problems of artificial intelligence since the inception of the field [McCarthy, 1959].

It is a very difficult problem. Common sense involves many subtle modes of reasoning and a vast body of knowledge with complex interactions. Consider the following quotation from *The Tale of Benjamin Bunny,* by Beatrix Potter:

> Peter did not eat anything; he said he would like to go home. Presently, he dropped half the onions.

Except that Peter is a rabbit, there is nothing subtle or strange here, and the passage is easily understood by five-year-old children. Yet the three clauses involve, implicitly or explicitly, concepts of quantity, space, time, physics, goals, plans, needs, and communication. To understand the passage, an intelligent system needs relevant knowledge from each of these domains, and the ability to connect this knowledge to the story. To design such a system, we must create theories of all the commonsense domains involved, determine what kind of knowledge in each domain is likely to be useful, determine how it can be effectively used, and find a way of implementing all this in a working computer program.

Since common sense consists (by definition) of knowledge and reasoning methods that are utterly obvious to us, we often overlook its astonishing scope and power. The variety of domains involved in a commonsense understanding of the world is not much less than the variety involved in all of human knowledge about the world; that is, most domains of human knowledge have some basis in a commonsense understanding. Likewise the kinds of reasoning involved in common sense include, in simple form, most if not all of the kinds of reasoning which are consciously usable by human intelligence. In short, most of what we know and most of the conscious thinking we do has its roots in common sense. Thus, a complete theory of common sense would contain the fundamental kernel of a complete theory of human knowledge and intelligence.

The purpose of this introductory chapter is to give a general feeling for the field of AI commonsense reasoning and introduce some basic ideas and terminology. We will discuss the kinds of issues which a theory of commonsense reasoning must address, or avoid; the general structure of these kinds of theories; the most important methodological debates in the field; and the relationship between theories of commonsense reasoning and other fields of study. Chapters 2 and 3 will lay a more detailed groundwork by describing the various logical theories we will use in this book.

The remaining chapters will each deal with one particular commonsense domain: quantities, time, space, physics, cognition, purposive actions, and interpersonal relations. Human common sense, of course, encompasses many other areas as well; there are commonsense theories of biology, of terrestrial and astronomical phenomena, of morality, of aesthetics, and so on. We have omitted these areas here simply because they have not been extensively studied in AI. This neglect is rather a pity; the restricted commonsense domains studied in AI sometimes give the feeling that the world consists of four blocks, five electronic devices, three people, an elephant, and a penguin.

This book entirely omits any discussion of the development of common sense: what aspects of common sense are innate, what aspects are learned, how they are learned, and to what extent they depend on the culture and the individual. Though these issues are ultimately of the highest importance in AI and cognitive psychology, at present we know very little about them. This theory will require a combination of major advances in AI, developmental psychology, and comparative anthropology. Therefore, this book makes no attempt to arrive at theories that are innate or culture-independent. The theories presented here are almost entirely the products of people born in the twentieth century and educated in Western culture generally and in science and math in particular; and they reflect this origin. Some aspects of these theories would certainly have appeared quite unnatural in many times and places; for example, the sharp division made between physical and psychological phenomena and the omission of the divine or spiritual as a central category.

## 1.1 Knowledge Bases

Commonsense reasoning in AI programs can be viewed as largely the performance of *inference* on a body of *object-level* information. Object-level information is information that describes the subject matter of the reasoning, as opposed to *control-level* information, that describes the internal state of the reasoning process itself.[1] Inference is the process of deriving new information from old information. Programs that do this kind of reasoning are generally designed as *knowledge-based systems.* A knowledge based system is a program consisting of two parts:

- The *knowledge base* — a data structure that encodes a body of object-level information.

[1] This distinction breaks down in a program that can introspect on its own reasoning process.

- The *knowledge base manager* — a collection of procedures that perform inferences on the information on the knowledge base, possibly modifying the knowledge base in the process.

In general, any object-level information that cannot be foreseen by the programmer, particularly problem-specific information, must be represented in the knowledge base. Information that can be foreseen by the programmer, such as fixed rules governing the subject domain, may be represented in the knowledge base, called a *declarative* representation, or it may be incorporated into the inference procedures, called a *procedural* representation. Typically, declarative representations have the advantage that the information may be more easily changed; that the information may be used in more different ways; and that the program may reason directly about its use of the information. Procedural representations typically have the advantage of efficiency.

All other modules of the program use the knowledge base manager to access or change this information in the knowledge base. A knowledge base manager generally includes procedures for *assimilation,* adding some new information to the knowledge base, and drawing appropriate conclusions; and *query answering,* providing requested information on the basis of the information in the knowledge base. Some knowledge bases also provide facilities for *deletion,* the removing of information from the knowledge base, and the withdrawal of all conclusions drawn from it.

Various different modes of inference are used in common sense. An inference is *deductive* if it is logically sound; the inferred information is necessarily true if the starting information is true. An inference is *abductive* if it offers a plausible explanation of some particular fact. For instance, if you see that there is no mail in your mailbox, you may infer that your spouse has come home; this is an abduction, but not a deduction, since other explanations are possible. An inference is *inductive* if it infers a general rule as an explanation or characterization of some number of particular instances. For example, if you come to a new town, and three or four times you see a white firetruck, you may infer the rule, “In this town, firetrucks are white.”

There are substantial advantages to using deductive inferences, when possible. First, no other type of inference is fully understood. Second, the fact that deductive inference preserves truth greatly simplifies the design of knowledge bases. If a knowledge base uses only deductive inference, and all the information that is assimilated is true, then all the conclusions it draws will be true. If a knowledge base uses non-deductive inference, then it runs the risk of making false conclusions even if all the given information is true. We must therefore consider the problem of what to do when the inference is discovered to be false. Unfortunately, much of commonsense reasoning is inescapably non-deductive by nature.

## 1.2 Methodology

The focus of this book is on the development of declarative representations for the knowledge used in commonsense inferences. The basic approach used here, as in much of the research in automating commonsense reasoning, is to take a number of examples of commonsense inference in a commonsense domain, generally deductive inference; to identify the general domain knowledge and the particular problem specification used in the inference; to develop a formal language in which this knowledge can be expressed; and to define the primitives of the language as carefully and precisely as possible. Only occasionally is there any discussion here of the algorithms, the control structures, or the organization of data that would be used in an actual reasoning system.[2] The reader may therefore wonder how any of this contributes to the building of actual AI programs. We justify our approach by discussing that this kind of analysis is an important first step in designing AI programs, and how it fits into an overall design methodology.

Our approach rest on three general methodological assumptions:

1. In designing a reasoning system, it is generally worthwhile to characterize the object-level information that the system will use, and the inferences that it will perform, independently of determining how the system should perform these inferences.

2. A number of fundamental commonsense domains arise so frequently in designing reasoning systems that it is worthwhile for the research community to study the basic structure of commonsense inference in these domains independently of any particular application. Results from this study will be useful across a large range of applications. The more domain information can be represented declaratively, the easier it will be to design systems that can handle this information flexibly.

3. Certain characteristics of the knowledge used in commonsense inference make it particularly important that the language used to express this knowledge be very carefully and tightly defined. In particular: (a) Commonsense inferences tend to use a great variety of partial and incomplete knowledge. Representations for partial knowledge whose meaning is left at all open to the intuitive understanding of the programmer are very liable to be ambiguous or unclear in important respects. (b) The very familiarity of commonsense concepts and their close connection to basic natural language forms renders them liable to ambiguity. Unless particular care is taken, it is very easy for a symbol or formula to be interpreted in one way in one part of a program, and in a way that is subtly but importantly different in a different part.

We may clarify these points by contrasting commonsense reasoning with other areas of computer science. In standard algorithmic or numerical analysis, the domains of discourse, such as graph theory or differential analysis, are very restricted and mathematical. The domains have generally been given precise definitions and characterized; the researcher can focus on finding effective algorithms. In database system design, the details of finding effective primitives for a particular application are generally considered to be the problem of the user, and the theory gives him little guidance. The design of the database system is concerned with providing an architecture for the manipulation of information, not with giving a language for its expression. Moreover, database systems typically deal with only limited forms of partial information, such as missing tuples, or null values. (Some recent work in database theory has been drawing close to AI issues.)

Based on these assumptions, we suggest that a commonsense reasoning system be designed as a knowledge base, recording information and performing inference on object-level information about some *microworld*. A microworld is a restricted, idealized model of the world containing only those relations and entities of interest in the particular reasoning system being designed. The analysis of the microworld takes place on three levels:

1. The definition of the microworld: its entities, its relations, and the rules that govern them.

[2] There is almost no discussion of domain-independent techniques; see the bibliography for references. A number of domain-specific algorithms are described.

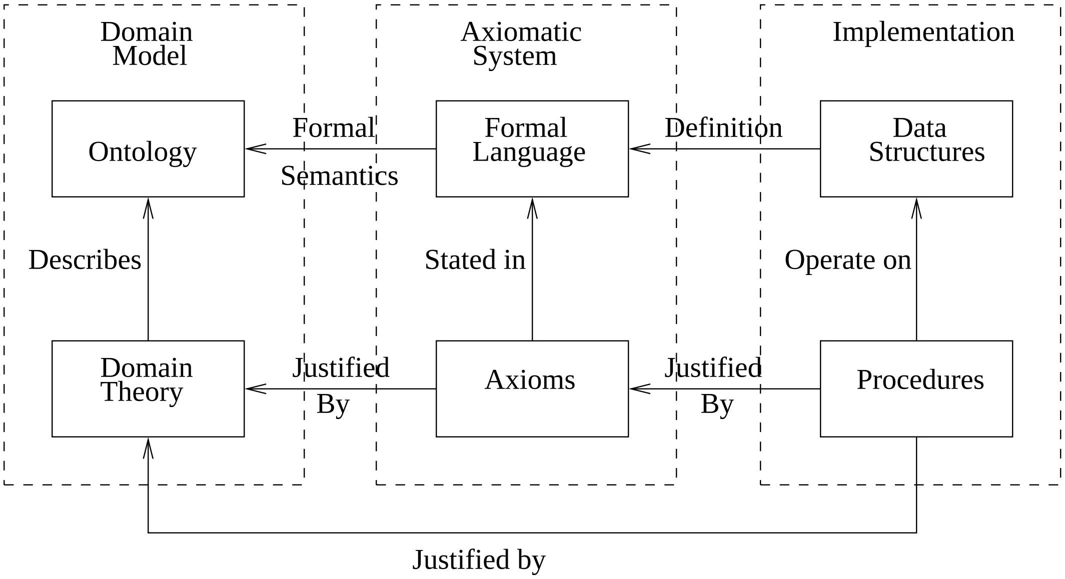

Figure 1.1: Theory structure

This definition is called the *ontology* of the domain.

2. The analysis of the types of information that the system must record and the types of inferences it must perform. The result of this analysis is generally a *formal language*, in which facts of the microworld can be expressed, and axioms or rules of inference characterizing inferences that are reasonable within the microworld.

3. An *implementation* of the information in mutable data structures and of the inferences in procedures and pre-defined data structures.

The analysis also specifies connections between these levels. The formal language is connected to the ontology by a *semantics*, which specifies how sentences in the language are interpreted as facts about the microworld. The implementation is connected to the language by formal definitions given for the data structures, and by arguments establishing that the inference procedures carry out legitimate inferences. (Figure 1.1).

For a simple example, suppose that we are constructing a knowledge base recording family relationships between individual people. That is, we want to be able to input a collection of facts about individuals, like “Fleur is Soames’ daughter,” “Winifred is Soames’ sister,” “James is Winifred’s father”, and then make queries like, “Who is Fleur’s aunt?” or “How is Fleur related to James?”

Our ontology for the domain can be very simple: the only entities we need consider are people, which are *atomic* (not decomposable), and people have various relations defined on them. In particular, people have a sex, male or female; there is a binary parent-child relation; and there is a binary husband-wife relationship.

We can construct a simple first-order language[3] for this domain using four predicate symbols: “male($X$)”, “female($X$)”, “parent($X, Y$)”, and “married($X, Y$)”, together with constant symbols

[3] We assume that the reader is familiar with first-order logic. For a brief review, see section 2.3.

for individual people. The semantics for this language is established by relating the symbols to the corresponding relations of the microworld.

We can now study the power of this language by studying the kinds of facts that can be expressed in it. Any fact asserting a specific relation between two people can be expressed. For instance "Winifred is Soames' sister," can be expressed in the form

$$\text{female(winifred)} \wedge \exists_X \text{ parent}(X,\text{winifred}) \wedge \text{parent}(X,\text{soames})$$

"James is Fleur's grandfather," can be expressed in the form,

$$\text{male(james)} \wedge \exists_X \text{ parent}(\text{james},X) \wedge \text{parent}(X,\text{fleur})$$

However, there are partial specifications of relations that cannot be expressed in this language. For example, there is no way to express the fact, "James is an ancestor of Fleur," or the fact, "The relation between Soames and Fleur is the same as the relation between James and Winifred." To express the first fact, we need to add a new predicate "ancestor$(X, Y)$". To express the second fact, we need to augment our language with the ability to treat relations as objects in themselves. For instance, we could add the function "relation_of$(X, Y)$" as a function to our language. We then can express the above fact in the form, "relation_of(soames,fleur) = relation_of(james,winifred)."

This treating of relations or properties as objects is known as *reifying*. In order to be able to give a clear semantics to reifications and to the symbols like "relation_of" that describe them, it is necessary to identify the reified entities with some well defined entity in the microworld. One common technique for achieving this (due to Frege) is to identify properties with the set of objects that have the property, and to identify relations with the set of object tuples satisfying the relation. For example, the property of maleness could be identified with the set of male people: Maleness = {James, Soames, Nicholas, Timothy ...}. The relation of aunthood could be identified with the pair of all aunts with their niece or nephew: Aunthood = { < Winifred, Fleur >, < Ann, Soames >, < Julia, Soames > ...}. We can then define the predicate "holds$(R, X, Y)$", meaning that relation $R$ holds on persons $X$ and $Y$, as true just if $< X, Y > \in R$. The function "relation_of$(X, Y)$" then maps the two persons $X$ and $Y$ to the relation that holds between them. (If $X$ and $Y$ are related in two different ways, relation_of$(X, Y)$ denotes their most direct relation, according to some particular criterion of directness.)

We can also use the formal language to express general facts about the microworld, and we can interpret certain commonsense inferences in the domain as inferences from these facts. For example, consider the old riddle:

Brothers and sisters have I none,<br>
But that man's father is my father's son.

One can frame the drawing of the conclusion, "That man is my son," as an inference from the riddle and plausible axioms, as shown in Table 1.1.

One simple knowledge structure for this domain is a labelled graph, with nodes denoting individual people, labelled with their name and sex, and with arcs representing parenthood and marriage

Definition D.1: A brother is a different male with the same parent.

$\forall_{X,Y}$ brother$(X, Y) \Leftrightarrow [\ X \neq Y \wedge$ male$(X) \wedge \exists_Z$ parent$(Z, X) \wedge$ parent$(Z, Y)]$

Definition D.2: A sister is a different female with the same parent.

$\forall_{X,Y}$ sister$(X, Y) \Leftrightarrow [\ X \neq Y \wedge$ female$(X) \wedge \exists_Z$ parent$(Z, X) \wedge$ parent$(Z, Y)]$

Definition D.3: A father is a male parent.

$\forall_{X,Y}$ father$(X, Y) \Leftrightarrow [$ parent$(X, Y) \wedge$ male$(X)]$

Definition D.4: A son is a male child.

$\forall_{X,Y}$ son$(X, Y) \Leftrightarrow$ parent$(Y, X) \wedge$ male$(X)$

Axiom A.1: Everyone is either male or female.

$\forall_X$ male$(X) \vee$ female$(X)$

Fact F.1: Brothers and sisters have I none.

$\neg(\exists_Z$ brother$(Z,$me$) \vee$ sister$(Z,$me$))$

Fact F.2: That man's father is my father's son.

$\exists_{U,V}$ father$(U,$that$) \wedge$ father$(V,$me$) \wedge$ son$(U, V)$

Fact F.3. That man is male.

male(that)

Prove: That man is my son.

son(that,me)

Step S.1: Someone with the same parent is either a brother, a sister, or self.

$\forall_{X,Y,Z}$ [ parent$(X, Y) \wedge$ parent$(X, Z)$ ] $\Rightarrow$ [ brother$(Y, Z) \vee$ sister$(Y, Z) \vee Y = Z$ ]

Proof: D.1, D.2, A.1

Step S.2: If a parent of mine has a son, then that is me.

$\forall_{X,Y}$ [ parent$(X,$me$) \wedge$ son$(Y, X)$ ] $\Rightarrow Y =$me

Proof: D.4, S.1, F.1

Step S.3: I am that man's parent.

parent(me,that).

Proof: D.3, S.2, F.2

Step S.4: That man is my son.

son(that,me).

Proof: S.3, D.4, F.3

Table 1.1: Proof in the Family Microworld

(Figure 1.2). It is easy to devise graph search procedures to answer queries such as "Who are Soames' aunts?" or "What is the relation between Winifred and Fleur?" and to verify that these procedures are correct in terms of the definition of the data structure and the properties of these relations. Such an implementation has the advantage of simplicity and ease of use, but it is limited in its capacity to express partial information. For instance, it is not possible in this implementation to record the fact, "Sarah is either married to Martin or to Jimmy." By contrast, the fact is easily expressed in our formal language.

$$\text{married(sarah,martin)} \vee \text{married(sarah,jimmy)}$$

If the application of the reasoning program demand the manipulation of facts such as this, therefore, some more powerful data structure must be used. One possibility would be a direct representation of the above formal sentence as a string of symbols.

One potential source of ambiguity in using implementation like the labelled graph of family relations is the question of completeness. Can one assume that the non-existence of an arc means that no such relation holds or not? In the family graph, for example, if there is no "married" arc attached to a person's node, does that mean that the person is definitely unmarried? or that he/she is definitely not married to any of the other people in the graph, but may be married to someone not in the graph? or does it not offer any such constraint? Similarly, if there is an arc indicating that Fleur is the child of Soames, but no other parenthood arc from Soames, does that mean that Fleur is definitely Soames' only child? Certainly, one cannot assume that the graph shows everyone's parents; otherwise the graph would have to be infinite. Depending on the application, one may choose to decide this question either way, but it is important to be clear which choice is made, and to interpret the inferences made by the system in light of the chosen interpretation. Under any choice, the expressive power of the graph is limited in one respect or another. If the graph must show every child of each individual, then there is no way to express a fact like, "Fleur has children," if the number of children is not known. If the graph need not show every child, then there is no way to express a fact like, "Fleur has no children."

There are definitely costs to giving precise definitions to the representations used in AI systems. First, doing so can be a lot of hard work, as this book will amply illustrate. If the inferences to be performed have limited scope, then it may be easier to code them up intuitively, and rely on the empirical success of the program as justification. Second, using a precisely defined representation may make it harder to express vague concepts (section 1.2) or to establish a correspondence between the representation and natural language text (section 1.6). Third, the use of precise definitions for concepts relies on a distinction between analytic knowledge (knowledge that is true by definition) and synthetic knowledge (knowledge that is true by virtue of the external world), a distinction that has been philosophically in ill-repute for the last forty years or so [Quine, 53].

Finally, it is not clear how to use precisely defined representations in the context of a program that learns concepts incrementally. For example, a friend of mine was under the impression, when she was a child, that ponies were young horses; she learned, in the eighth grade, that they are a small breed of horse. A computer program that did this would use a symbol PONY to represent the concept of ponies; start with the definition "PONY is a young horse;" and change to the definition "PONY is a breed of small horse." But, in our view of representation, it is hard to understand why

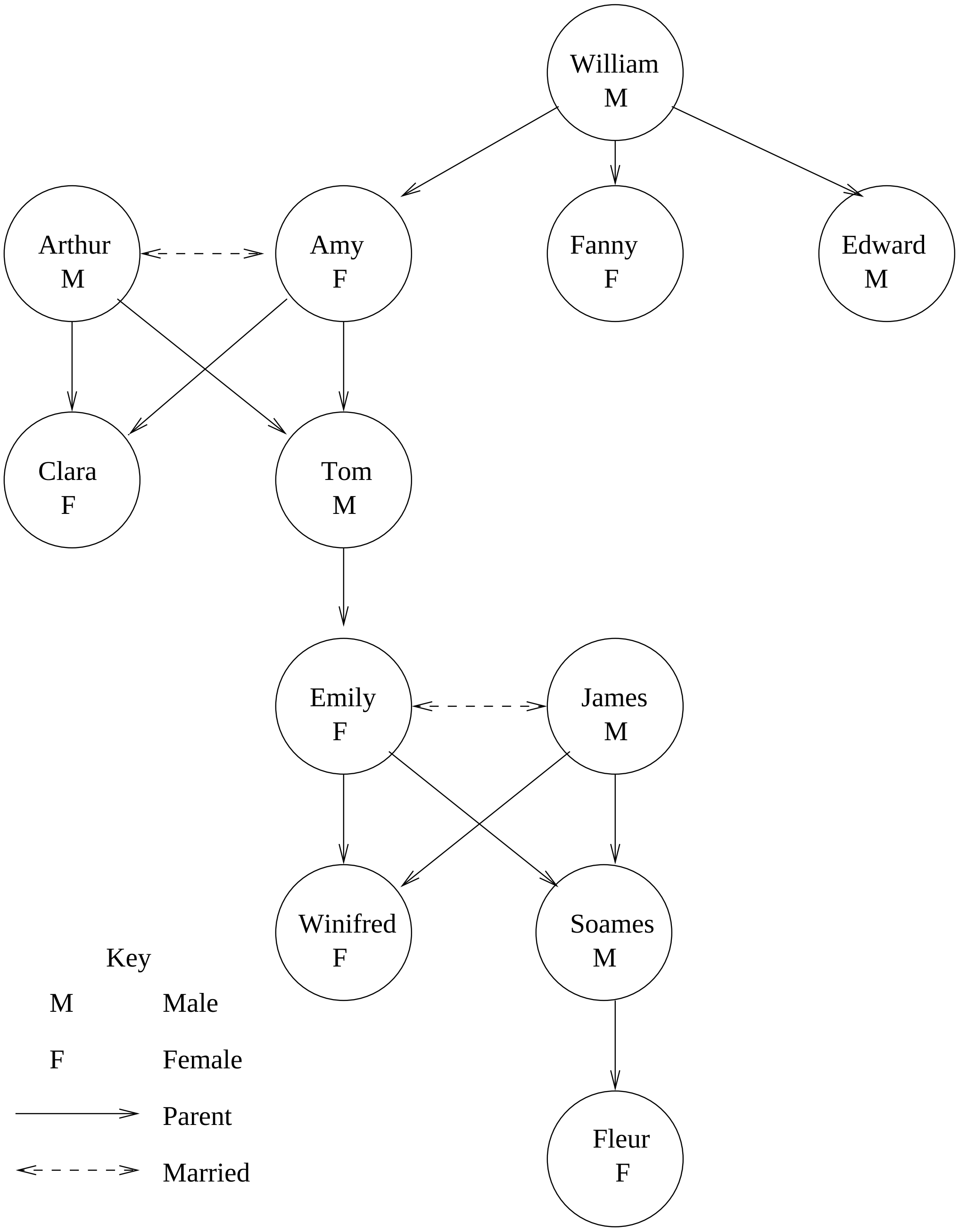

Figure 1.2: Family tree

this is progress. Is it the symbol 'PONY' that has changed its meaning? If so, the change seems perfectly arbitrary, since 'PONY' is an internal symbol that could just as well be 'G0011'. In what respect is it "better" for 'PONY' to represent ponies than young horses? Is it the concept of PONY that has changed its meaning? To say so would seem to introduce a peculiar distinction between a concept, on the one hand, and the meaning of the concept, on the other.[4] All this is not to say that the program is not behaving reasonably; just that it is difficult to explain its behavior under the view we have taken of concepts.

## 1.3 Implementation

Since the bottom line in AI research is the construction of intelligent programs, it is reasonable to ask how formal domain theories like those we discuss here will help us in the actual programming. This question of the relation of theory to practice is difficult in many fields; it is particularly difficult in AI, where basic techniques and methodology are still very much in a state of development. We can, however, point out a number of possible connections:

1. The axioms of the domain may be used directly in symbolic form as input to a theorem prover[5] or other domain-independent inference engine. The inferences made by the inference engine from the axioms would then constitute part or all of the reasoning performed by the AI program. The achievement of such a direct link between a logical theory and a working program is one of the central ideals of logic programming. At present, however, the techniques of domain-independent inference are too weak to work efficiently on theories as complex as those we will consider.

2. The theory can be used to verify the correctness of an AI program. The theoretical justification of a domain-specific AI program must rest on a precise theory of that domain. "Theoretical justification" is, of course, a broad category, [DeMillo, Lipton, and Perlis, 79], which may range from a fully symbolic proof, based on the semantics of the programming language, to a argument, based on an understanding of the program and the domain.

At some future date, it may be possible to automate the verification or even the construction of AI programs based on formal specifications of the domain and of the purpose of the program.

3. The formal characterization of a domain may allow the application of known mathematical results to solve particular problems, or to determine their difficulty.

4. The knowledge structures used in the program may be based on the formal theory. As discussed in section 1.4, the familiarity of commonsense domains makes it particularly easy to abuse notation in AI programs and to use a single symbol in different ways that are not quite mutually consistent. A formal definition of the meaning of the symbols is very helpful in avoiding this kind

[4] Another possible suggestion is that the definition of the symbol PONY is, from first to last, "Any creature called a 'pony' by the English-speaking community," and that the program goes from the belief of the (synthetic) fact that ponies, so defined, are young horses to the belief that they are a small breed of horse. Aside from the fact that this make concepts subordinate to language, this solution can be objected to on two grounds: (A) This will do for an individual, but how do we explain what happens when the community finds that it has been mistaken about ponies? (B) This definition breaks down if we discover we have made a mistake about the word. If we discover that the word is actually "ponay" then we are left believing nothing.

[5] A theorem prover is a domain-independent program which performs deductive inference from information expressed as a set of formulas in some logical notation. Resolution theorem provers and Prolog interpreters are two types of theorem provers.

of bug. For this purpose, the key part of the formal theory is the definition of the microworld and the semantics of the symbols. The construction of an axiomatic system is less important.

5. Logical analysis of the domain often serves to uncover peculiarities or anomalies that a programmer may at first overlook in writing code. In particular, a logical analysis is a good way to determine what information is needed to support a given conclusion, and what kinds of assumptions are reasonable and necessary in reasoning about given domain. For example, in designing a system that will reason about knowledge and belief, it is generally wise to determine what kinds of problems of referential opacity (section 2.5) will arise in the particular application, and to decide how they can be dealt with, at an early stage.

6. After an AI program has been written, a *post hoc* logical analysis may aid in debugging, understanding, presenting, and extending it. The logical analysis may reveal some flaw in the program's reasoning or some inconsistency in the assumptions that it makes about the microworld that could lead to a bug. It may reveal some gap in the program's reasoning that could easily be filled in a natural extension. It can help separate central features of the program from peripheral features and control issues from object-level issues. It can help clarify the scope and limits of the program by explicitly defining the microworld and the class of inferences involved.

In this book we look in detail at a number of existing programs: MERCATOR (section 6.2.3) and TOUR (6.2.4), which maintain cognitive maps; NEWTON (6.2.6), ENVISION (7.1), and QP (7.2), which make physical predications; and TWEAK (9.1.1) and RAP (9.4.1), which are planners. We also look at a number of knowledge structures that have been developed and applied primarily in the context of programs rather than in the abstract, including qualitative differential equations (4.9), occupancy arrays (6.2.1), constructive solid geometry (6.2.2), and configuration spaces (6.2.5).

## 1.4 The Role of Natural Language

People have developed many flexible, broad, powerful schemes for expressing and communicating commonsense facts; namely, the natural languages. How AI theories of commonsense reasoning should use natural language concepts is an important question, but a difficult and highly disputed one.

There is widespread agreement that AI programs should not use full natural language text as a knowledge representation language. Natural language cannot be easily manipulated algorithmically. It is full of ambiguities. Its meaning is context-dependent. Its syntax is extremely complex, and strongly dependent on semantics. Its connectives (prepositions, articles, conjunctions) are remarkably vague and unsystematic. There are few powerful rules of inference on natural language strings. Moreover, natural language is notoriously ineffective at transmitting certain types of information, notably spatial information. (A picture is worth a thousand words.)

Thus, AI representations should be very much systematized as compared to natural language. The question is, how radical should this systematization be? At the one extreme are those theorists who form their representations by starting with natural language text, simplifying syntax and connectives, and resolving ambiguities. At the other extreme are those who construct their representations like a mathematical notation, analyzing the knowledge necessary for a problem in terms as abstract as possible.

There are several practical and theoretical arguments in favor of basing representation on natural language. The first of these is simply one of ease. We describe commonsense situations and we assert commonsense rules in natural language. It is relatively easy to "transliterate" these descriptions and rules into a fixed syntax, and it is hard to develop a language that is better or more precise, particularly in domains where no precise scientific language is available. In a related way, knowledge engineers who are trying to encode expert knowledge generally extract the basic domain primitives from verbal and written protocols. There is not much else one can do.

Second, a major function of commonsense inference systems is to interact with natural language systems. Obviously, the automatic translation of natural language to an internal representation is easier if the internal representation is close to the natural language. More importantly, we must be able to represent anything sensible that appears in the text. If a natural language term is vague or ill-defined, then any representation of its content will likewise be vague and ill-defined. (See [Hobbs, 1985c].)

Third, several important philosophical schools argue in various ways (not wholly compatible) that beliefs and mental states are inextricably tied up with natural language. It has been argued that any representation scheme ultimately draws its meaning from natural language [Wittgenstein, 1958]; or that beliefs can be reasonably ascribed only to creatures that use language [Davidson, 1975]; or that the particular forms of a native natural language influence the basic structure of thought [Whorf, 1956]; or that, since natural language terms are not precisely definable, the representation of natural language text must be in terms of primitives very close to the text [Fodor, 1975]. If any of these philosophical arguments is correct, then it is inherently futile to look for a language-independent system of representation. (Many of the philosophers who make these arguments would claim that any attempt at artificial intelligence is inherently futile.)

Despite these arguments, my opinion is that the study of AI representations should not be focussed on the representation of natural language text. I feel that natural language text is a poor starting point for thinking about representation; that natural language is too vague and too ill-defined, even when rationalized, to allow systematic forms of inference; and that the tailoring of representations to natural language text can lead to an emphasis on extremely subtle issues of modality, reference, and context, while avoiding "nuts and bolts" issues such as spatial and physical reasoning.

## 1.5 The Role of Logic

No issue has been more debated in knowledge representation than the proper role of logic. Extreme logicists maintain that almost all intelligent functions can be characterized as deductions from some set of axioms, and can be carried out by a sufficiently well-crafted general-purpose theorem prover. Extreme anti-logicists maintain logic is wholly irrelevant, either as a notation, or as a theory of inference. There are also many possible intermediate positions. It is not possible here to present the various sides of this debate in any detail. The bibliography gives references to some of the most important papers on the issue. Here, we will just point out some of the central issues. Our own position should be clear from our discussion of methodology in section 1.4.

The debate over logic deals with the role of "theorem provers" in programs and the role of formalisms (formal ontologies, languages, and semantics) in AI theories. The ultimate questions to resolve are, is an AI program likely to be largely a theorem prover, or to contain a theorem prover as a small component, or not to contain anything that looks anything like a theorem prover? and, to what degree should the representations in an AI program be given formal definitions? The following issues have been particularly important in the debate:

- How much intelligent reasoning can be characterized or approximated as deduction?
- Can formal theories of inference be extended to cover non-deductive inference?
- Does intelligent reasoning require the full power of logical inference?
- It is possible to do arbitrary computation using a theorem prover. Is this relevant?
- Are formal representations useful even if inferences are not formally characterized?
- Is it useful to characterize the inferences carried out in an AI program independently of the procedures to carry them out? Is it possible to separate domain knowledge from control knowledge?
- Must AI deal with vague concepts? If so, does that make logic irrelevant?

The arguments over the role of logic in commonsense reasoning are orthogonal to the arguments over the role of language. AI researchers have taken all combinations of positions. For instance, Hobbs [1985c] uses natural language primitives in a predicate calculus syntax. Montague semanticists [Montague, 1974] have combined logic and natural language constructs in an even stronger form, by interpreting natural language as a formal logic. Much of the work on natural language processing has used natural language terms for representation, but has not attempted to give precise definition, for either the terms or the syntax of their representational scheme. (Wilks [1976] explicitly claims that it would be an mistake to attempt precise definitions.) Much of the work in spatial, temporal, and physical reasoning has used a precise, logical language and ignored the relevant natural language constructs. Work on representing narrative, particularly that of Schank and his students [1975, 1977], has used a system of abstract primitives which are language-independent, but which are defined informally rather than logically; this work uses neither natural language nor logic as a basis. Connectionist systems uses no symbolic representations at all.

The choice of methodology often depends on domain. Work in complex, poorly structured domains, like many expert systems domains, tends to use natural language terms, since they are available, and to avoid formal definitions, since they are hard to formulate. Work in simple, highly structured domains, such as simple physics and temporal reasoning, tends to avoid natural language, which is ambiguous and convoluted, and to use logical definitions, since precision is relatively easy to attain.

To some extent, these issues reflect similar issues in the philosophy of language and epistemology. The view that theories should be formal and language-independent derives ultimately from the example of mathematics, and was argued by Bertrand Russell [1940] and related philosophers

([Carnap, 1967], [Ayer, 1946]). The view that theories should be informal and based on natural language is related to the "ordinary language" philosophy of Austin [1961] and Wittgenstein [1958]. The view that natural language can be associated with a logical structure is particularly associated with Richard Montague (1974).

## 1.6 Incomplete and Uncertain Knowledge

There is a great gap to be bridged between the rigidity of a computer program and the flexibility of human reasoning. One aspect of this flexibility is the broad scope of human common sense discussed at the beginning of the chapter. Another aspect is the human ability to deal with partial and uncertain information. Very little of the knowledge that you use in daily life is precise or complete, and very little is certain. You know that the water in the tea-kettle will come to a boil soon after you turn the knob on the stove, but you do not know how soon it will boil, how hot the flame is, or how much liquid is in the kettle. This is incompleteness. Moreover, you cannot be entirely sure that the water will boil at all; the stove may be broken, or someone may come and turn it off, or the kettle may be empty. This is uncertainty. Similar considerations apply in all commonsense domains. You may have good reason to believe that your niece will be pleased if you send her flowers for her birthday. But you may not know whether she would prefer roses or irises (incompleteness) and it is possible that she hates either flowers, or her birthday, or you.

The use of partial information is critically important for several reasons. First, there are many cases where exact information is expensive or impossible to attain. Vision systems can deliver exact information only at great cost, or in very restricted environments. An intelligent creature that makes use of a practical vision system must therefore be able to cope with partial information as to the positions, shapes, and types of objects it sees. Complete information about the psychology of another person cannot be obtained at any cost; in fact, it is not clear that such a concept makes any sense. Again, decisions must be based on incomplete information. Second, even when the information is available, it may be too complicated to use. If you drive a car off a cliff, what happens? Well, if exact specifications of the car have been obtained from the manufacturer, and the cliff has been carefully surveyed, it may be possible to calculate the result very precisely. But the calculation will be long and difficult, and quite pointless if all you want to know is that you should not go over the cliff as a shortcut to the bottom. Third, it is often useful to do generic reasoning about whole classes of similar, but not quite identical cases. Suppose that you are putting up a chicken-wire fence to keep the chickens out of the asparagus, and you want to be sure that none of your chickens can get through it. Even if you happen to know the exact proportions of each chicken, you do not want to do a separate calculation for each one, establishing that it cannot go through your fence. You want to be able to determine that no chicken can go through the fence, by reasoning about the general characteristics of chickens.

The use of uncertain information is necessary when some important datum derives from some unreliable source. This can be an imperfect sensor, or an unreliable informant, or an uncertain rule. Uncertain rules, in turn, must often be used to supplement incomplete knowledge with plausible, but uncertain, guesses. If we see a car coming down the street, we infer the presence of a driver. This inference is not certain, but it is generally correct, and it is necessary to plan sensible action.

The representation of incomplete knowledge in various domains is one of the focal issues of this book. The central technique is to find predicates that express useful partial information. Disjunction and existential quantification also serve to express partial information. The representation and use of uncertain information is discussed in chapter 3. Elsewhere in this book, we largely avoid the use of uncertain information.

## 1.7 Vagueness

In some respects, the concepts of commonsense knowledge are *vague,* in a sense that goes beyond uncertainty and incompleteness. Many categories of common sense have no well-marked boundary lines; there are clear examples and clear non-examples, but in between lies an uncertain region which we cannot categorize, even in principle. For example, the concepts of the various species would seem to be necessary in a commonsense understanding of the natural world. Yet, as we know, since species evolve one from another, there cannot be any clear line dividing them; by traveling down and up the evolutionary chain, we can turn a dog into a cat in nearly continuous steps. There would be no way to decide at what precise point the ancestors of dogs cease to be dogs, as one goes back in time, even if we knew everything about these proto-dogs. In the same way, there is no way to define the exact line, to the inch, that marks the boundary of the Rocky Mountains; there is no maximum numbers of hairs permitted to a man who is bald; there is no telling, to the second, when a person ceases to be a child. All these terms — “dog”, “Rocky Mountains”, “bald”, “child”, — are vague.

From a theoretical point of view, this vagueness is extremely difficult to deal with, and no really satisfactory solutions have been proposed. The difficulties are made vivid by an ancient paradox called the “Sorites” (meaning heap). If you have a heap of sand, and you take away one single grain of sand, you will obviously still have a heap of sand. But therefore, by induction, you can take away all the sand, and still have a heap, which is absurd. Like the other terms above, “heap” is vague; there is no specific minimum number of grains that a heap can contain.

Often, we can treat vagueness of this kind in the same way as incompleteness. Though inadequate theoretically, this works in many practical cases. Suppose that “bald” did refer to some specific number of hairs on the head, only we did not know which number. We know that a man with twenty thousand hairs on his head is not bald, and that a man with three hairs on his head is bald, but somewhere in between we are doubtful. Under this analysis, the statement “Sam is bald” is no more problematic than “Sam has fewer hairs than John.” In either case, sometimes we will be sure that it is true, sometimes we will be sure that it is false, sometimes we will be uncertain. But despite this uncertainty, “bald” is a precise concept just as “the number of hairs on John’s head” is a precise concept (at some specific moment), despite the fact that we do not know this number. Under this account the basic step of the Sorites paradox applied to baldness, “If a man is not bald, then losing one hair will not make him bald,” becomes merely a plausible, not a necessary inference, just as, “If a man has more hairs than John then he will still have more hairs after losing one,” is true in most, but not all, cases. It seems very likely that this account will prove inadequate for some purposes, since it is unquestionably wrong — “bald” is *not* a precise, but undetermined concept; it is a vague concept — but it is not clear where it breaks down. In any case, this analysis is all we need for the purposes of this book. (Another technique for dealing with vague concepts is fuzzy logic [Zadeh, 87].)

## 1.8 Indexicals

Particular problems arise with indexicals, words like “I”, “here”, “now”, and “this”. The meaning of these words and the truth of sentences containing these words depends entirely on the circumstances of their use. Formal theories have therefore in general avoided these concepts by replacing sentences like “I am hungry now,” with *eternal* sentences such as “Ernie Davis is hungry at 3:30, April 12, 1987.”[6] However, this is not quite adequate for AI systems. An AI system must at some level be able to distinguish between itself and other systems, and between the present moment and other moments. If the robot comes to believe “At time $T$ a truck is bearing down on $P$,” then what it must do and how hard it must think depends critically on whether $T$ is the present moment, and on whether $P$ is itself.

The approach we advocate for a theoretical analysis is that inference should be done in terms of eternal sentences. The robot’s knowledge of its own identity and of the present moment should be programmed into the interfaces between the sensors and the knowledge base, the interfaces between the knowledge base and the effectors, and the control structure that chooses focuses for inference. For example, if the visual sensors of George, the robot, detect a truck bearing down on it, the visual interpretation system, consulting George’s internal clock,[7] will enter the sentence, “A truck is bearing down on George at 1:37, March 5” into the knowledge base. The inference control structure will note that the time is the present and George is himself, and that therefore this should be given high priority for inference. The inference engine will then deduce the conclusions, “George is in danger at 1:37, March 5,” “George should flee at 1:37, March 5,” and “George’s wheels should roll rapidly backwards at 1:37 March 5.” At this point, the effector control, finding a statement about what George’s effectors should do at the present moment, will set the wheels rolling. (This is discussed further in section 5.13.)

Of course, this is a very much idealized model, and in practice one would want to short-circuit a lot of this. Reflexes in humans, for example, amount to a direct short circuit from the sensors to the effectors with no involvement of the knowledge base. A lot of the low level planning for converting active current plans into effector actions may be carried out without any explicit representation either of the time or of the robot, since these inferences are only needed for planning the robot’s own immediate motions, and are not entered in a permanent knowledge base.

## 1.9 Commonsense Reasoning in Artificial Intelligence

As we have argued above, all parts of artificial intelligence — vision, natural language, robotics, planning, learning, expert systems — must, sooner or later, make use of commonsense knowledge and commonsense reasoning techniques. Indeed, commonsense reasoning is so closely bound up with

[6] An alternative approach is to construct a theory of meaning which builds in circumstance [Barwise and Perry, 82].

[7] The robot need not have a chronometer that gives it times like “1:37, March 5.” All it needs is the ability to give names, such as “t1045” to times, and to mark these extralogically as the current moment.

these other subfields that it is by no means clear that it forms an independent domain of inquiry. Certainly we do not make any strong distinction in common parlance: one might well say of a person who consistently adopted semantically strange readings of natural language sentences, or who could not use context to interpret what he saw, or who could not plan his way out of a paper bag, or who did not learn from experience, that he did not have any common sense.

In fact, it seems clear that there is no sharp demarcation line between common sense and these other AI domains, either theoretical or psychological.[8] Nonetheless, for the purposes of research, it does seem possible and useful to delimit a certain area of study. We are studying "consensus reality" [Lenat, 1988], knowledge and reasoning available to the overwhelming majority of people in our culture past early childhood; thus, not expert knowledge. Once this knowledge is acquired at a young age, it is fairly stable; thus, we can more or less ignore learning. This knowledge is held by people of many different countries; thus, it does not include knowledge of any particular language, though it would include knowledge about languages generally, such as that people communicate in language. The use of sensors and effectors at the low-level involves a wholly different class of computational techniques; thus we can ignore low-level vision and robotics, though probably not high-level vision and robotics. We are studying types of reasoning that seem easy, not those requiring substantial conscious effort; thus, we can ignore sophisticated planning, though not elementary planning.

## 1.10 Philosophy

> Warm yourself by the fire of the wise, but beware lest you burn yourself with their coals, for their bite is the bite of a jackal, and their sting is the sting of a scorpion, and their hiss is the hiss of a serpent, and all their words are burning coals.
> — Mishnah Avot, 2.15

The most important external influence on AI theories of commonsense reasoning has been twentieth-century analytical philosophy. Most of our basic analytical tools, particularly formal logics, and much of our analysis of specific domains, particularly time, action, and mind, were developed by philosophers and mathematical logicians. If an AI researcher wants to develop a representation for a commonsense domain, he should certainly take the time to check out what the philosophers have said.

However, he must be prepared to be disappointed. He is likely to find that the philosophers and logicians have nothing to say about the issues that seem key to him, and have focussed instead on issues and examples that to him seem trivial, far-fetched or irrelevant. Simply, AI and philosophy have substantially different objectives and methodologies; they work, so to speak, on different wavelengths.

A central difference between the two fields is that AI looks for useful answers to practical problems

[8] Another possibility could be that there is no general purpose commonsense reasoning; each separate task domain — vision, natural language processing, robotics, and so on — uses its own separate body of commonsense knowledge and its own inference procedures. We would argue that the basic knowledge of the external world required in all these tasks overlaps sufficiently to make it worthwhile to study the manipulation of this knowledge independently of the particular tasks involved.

while philosophy looks for ultimate answers to fundamental problems. Accordingly, AI theories must be detailed and specific to the level of implementing systems that solve particular problems. However, they need not stand up under all conceivable criticisms; they need merely be adequate for the problems to which they are addressed. In contrast, philosophical theories are often quite vague, but they are expected to be absolutely valid without exception or counter-example. For this reason, much work in AI, such as describing the expected sequence of events in a restaurant, seems entirely trivial to philosophers, while much work in philosophy seems to AI people to be either uselessly vague or focussed on weird counterexamples.

Consider, for example, the problems associated with a "natural kind" term such as "cat". The major philosophical problem associated with such words is to determine what we mean when we use such a word. What is presupposed by the use of "cat"? Under what, if any, circumstances, would we be justified in deciding that "cat" is, in fact, a meaningless word, and that therefore all our previous beliefs about cats were likewise meaningless? Conversely, if we found out that cats were quite a different kind of entity than we had previously believed, how would our previous uses of the words have obtained this meaning? Such questions lead naturally to the construction of hypothetical queries like, "If, on some unexplored island, we find a race of creatures that are exactly like our cats, but have a wholly different evolutionary heritage, would they be cats?"[9]

From the AI point of view, the primary problem is, what kinds of commonsense information do we have about cats? The overwhelming bulk of this information (probably) relates to the superficial, sensory properties of cats: what they look, sound, feel, and smell like. Then there are much smaller bodies of information regarding the internal structure, behavior, and origin of cats. Which of these properties, if any, constitute the defining characteristics of a cat is very rarely of practical importance. Our problem is to represent and manipulate all of this very mundane and obvious information about cats. The philosophers have not addressed such issues. The philosophical issues, though possibly of great ultimate importance in formulating theories of concept learning, are not of immediate concern to AI research.

## 1.11 Mathematics and Commonsense Reasoning

The other major external influence on formal theories of commonsense reasoning is mathematics. In the theories of quantities and geometrical space, mathematical theories are so rich that the problem for us is one of selecting, and to a small degree amplifying, the available tools, rather than originating new theories. The fundamental theories of logic and set theories also owe as much to mathematics as to philosophy. In other commonsense domains, particularly domains involving mental activity, there is very little math that bears on our problems.

There are, however, two important difference between the mathematical mind-set and the mind-set of research in commonsense reasoning. First, mathematics is driven by abstraction; it looks for common abstract structures that underly seemingly very different phenomena. It is one of the glories of mathematics that phenomena as different as small-arc pendulums, masses on springs, LRC circuits,

[9] Note that the question is not "Would we then (after finding them) call them cats?", which could be a question about how we change language in response to discoveries. The question is, "When we now say 'cat', do we mean to include such hypothetical creatures?" The need to make this fine distinction is itself a significant contribution of philosophical study.

and time-varying populations can all be characterized by the same kinds of differential equations. In commonsense theories, by contrast, while it is important that structural similarities exist between representations and different domains, so that analogies can be constructed and applied, it is equally important that the concrete, individual aspects be kept in sight. If two very different kinds of things look the same in your theory, then you have abstracted some important issues away. Things which are commonsensically different should be described by a different kind of commonsense theory. (This argument is contested in [Hobbs, 1987].)

Another mathematical goal which is not, generally, carried over to AI is that of axiomatic parsimony. Mathematicians like to find the smallest, weakest set of independent axioms which will give the desired results. Partly, this is a matter of mathematical aesthetics; partly, a desire to make theories as general as possible; partly, a desire to make the axioms as self-evidently true and consistent as possible. None of this matters in AI. Knowledge bases in commonsense domains will, in any case, have so many axioms (many of them just stating contingent facts like "John loves Mary") that nothing will make them aesthetically appealing, or self-evidently true. Generality hardly matters, since AI systems are resolutely focussed on the specific.

## 1.12 References

**General:** As of the date of writing, this is, as far as I know, the only book-length general survey of commonsense reasoning in AI. [Davis, 1987a] surveys the field in a short article. [Charniak and McDermott, 1985] gives a good introduction to the area in chapters 1, 6, and 7. [Genesereth and Nilsson, 1987] has several relevant chapters on logic, plausible reasoning, knowledge and belief, and planning. Several important collections of papers in the area have been published. [Hobbs and Moore, 1985] is a collection of state-of-the-art research papers in various commonsense domains. The introductory chapter, by Hobbs, is an excellent overview of current research in the area. [Brachman and Levesque, 1985] is a collection of classic papers on knowledge representation and commonsense reasoning. These papers tend to be at a more abstract and general level than this book. [Brachman, Levesque, and Reiter, 1989] contains the proceedings of a recent conference on knowledge representation.

The necessity for commonsense reasoning in AI was first put forward by John McCarthy in "Programs with Common Sense," in 1959. This paper, and its successor, "Situations, Actions, and Causal Laws," [McCarthy, 1963] are still very much worth reading. [Lifschitz and McCarthy, 1989] is a collection of papers by McCarthy on commonsense reasoning and artificial intelligence.

The CYC program [Lenat et. al., 1986] is an ambitious project to encode the contents of an encyclopedia in a computer system for intelligent data retrieval. The results of this project are not yet available for evaluation.

**Vagueness:** The only theory that claims to deal with vagueness adequately is Zadeh's [1963, 1987] fuzzy logic.

**Methodology:** Many papers have discussed the role of logic in AI, and the issue is still hotly debated. Some of the most significant papers are [McCarthy, 1959], [McCarthy, 1963], [McCarthy and Hayes, 69], [Minsky, 1975], [Hayes, 1977], [Hayes, 1978], [McDermott, 1978a], [Newell, 1981], [Moore, 1982], and [McDermott, 1987a]. The use of natural language constructs in representational systems has been much less discussed; [Wilks, 1976], [McDermott, 1976] [Hobbs, 1985a], and [Hobbs, 1987] are relevant. The methodology advocated in section 1.4 here is distilled from [Hayes, 1977], [Hayes, 1978] and [McDermott, 1978a].

One methodological issue not discussed here is the use of multiple microworlds in an AI program. This is discussed from a theoretical standpoint in [Hobbs, 1985b] and [Addanki et. al., 1989] and from an implementational standpoint in [de Kleer, 1986].

**Philosophy:** Discussions of philosophy in the AI literature, and vice versa, mostly discuss philosophy's view of AI as a theory of mind, rather than AI's view of philosophy as providing analyses of commonsense concepts. Particular successful adaptations of philosophical ideas in AI will be found throughout the text and will not be enumerated here. The philosophical discussions of "natural kinds" mentioned may be found in [Putnam, 1962] and [Kripke, 1972] among other places. Early attempts at a logical analysis of statements about external reality are to be found in the works of Bertrand Russell [1903], A.J. Ayer [1946], and Rudolf Carnap [1967], among others. It may be noted that the "commonsense school" of philosophy was concerned with the defense of common sense rather than its analysis, and so contributed little if anything relevant to our enterprise. Attempts to find common ground between philosophical and AI research have led to some peculiar discussions; see, for example, [Pylyshyn, 1987].

**Architectures:** In this book, we omit any discussion of domain-independent architectures for reasoning. We list here a few references on particularly well-known theories. [Reichgelt, in preparation] is a general survey of the area. [Charniak et. al., 1988] is a textbook that covers a large range of reasoning architectures implemented in LISP. Logic programming is discussed in [Kowalski, 1979] and [Wos et. al., 1984]. Semantic networks, and their relation to logical representations, are discussed in [Woods, 1975], [Schubert, 1978], and [Brachman, 1985]. Frame-based programming systems are discussed in [Bobrow and Winograd, 1977].

**Other:** [Geertz, 1983] is an interesting discussion of common sense from an anthropological perspective. The major study of the developmental psychology of common sense is the work of Piaget; see, for example, [Piaget, 1951].