CHAPTER 1
Introduction

ARTIFICIAL INTELLIGENCE (AI) is the study of intelligent behavior. Its ultimate goal is a theory of intelligence that accounts for the behavior of naturally occurring intelligent entities and that guides the creation of artificial entities capable of intelligent behavior. Thus, AI is both a branch of science and a branch of engineering.

As *engineering*, AI is concerned with the concepts, theory, and practice of building intelligent machines. Examples of machines already within the reach of AI include *expert systems* that give advice about specialized subjects (such as medicine, mineral exploration, and finance), question-answering systems for answering queries posed in restricted but large subsets of English and other natural languages, and theorem-proving systems for verifying that computer programs and digital hardware meet stated specifications. Ahead lie more flexible and capable robots, computers that can converse naturally with people, and machines capable of performing much of the world's "knowledge work."

As *science*, AI is developing concepts and vocabulary to help us to understand intelligent behavior in people and in other animals. Although there are necessary and important contributions to this same scientific goal by psychologists and by neuroscientists, we agree with the statement made by the sixteenth-century Italian philosopher Vico: *Certum quod factum* (one is certain of only what one builds). Aerodynamics, for

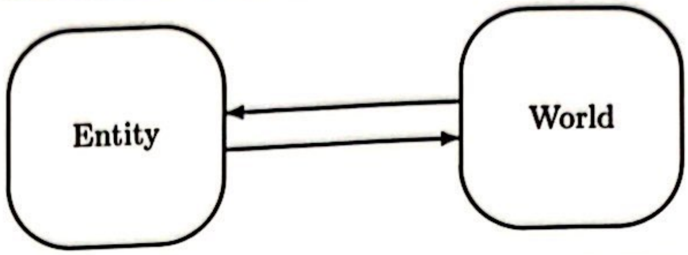

Figure 1.1 Entity and environment.

example, matured as it did because of its concern for flying machines; then it also helped us to explain and understand flight in animals. Thus, notwithstanding its engineering orientation, an ultimate goal of AI is a comprehensive theory of intelligence as it occurs in animals as well as in machines.

Note that in talking about the behavior of an intelligent entity in its environment, we have implicitly divided the world into two parts. We have placed an envelope around the entity, separating it from its environment, and we have chosen to focus on the transactions across that envelope. (See Figure 1.1.) Of course, a theory of intelligence must not only describe these transactions but must also give a clear picture of the structure of the entity responsible for those transactions. An important concept in this regard is that of *knowledge*. Intelligent entities seem to anticipate their environments and the consequences of their actions. They act as if they know, in some sense, what the results would be. We can account for this anticipatory behavior by assuming that intelligent entities themselves possess knowledge of their environments.

What more can we say about such knowledge? What forms can it take? What are its limits? How do entities use knowledge? How is knowledge acquired? Unfortunately, we cannot say much to answer these questions insofar as they pertain to natural, biological organisms. Even though we are beginning to learn how neurons process simple signals, our understanding of how animal brains—which are composed of neurons—represent and process knowledge about the world is regretfully deficient.

The situation is rather different when we turn our attention to artifacts, such as computer systems, capable of rudimentary intelligent behavior. Although we have not yet built machines approaching human levels of intelligence, nevertheless we can talk about how such machines can be said to possess knowledge. Because we design and build these machines, we ought to be able to decide what it means for them to *know* about their environments.

There are two major ways we can think about a machine having knowledge about its world. Although our ideas about the distinction between these two points of view are still being clarified, it seems that, in some of our machines, the knowledge is implicit; in other machines, it is represented explicitly.

We would be inclined to say, for example, that the mathematical knowledge built into a computer program for inverting matrices is implicit knowledge, "stored," as it is, in the sequence of operations performed by the program. Knowledge represented in this way is manifest in the actual running or execution of the matrix-inverting program. It would be difficult to extract it from the text of the computer code itself for other uses. Computer scientists have come to call knowledge represented in this way *procedural knowledge*, because it is inextricably contained in the very procedures that use it.

On the other hand, consider a tabular database of salary data. In this case, we would be inclined to say that the knowledge is explicit. Programs designed to represent their knowledge explicitly have turned out to be more versatile in performing the complex tasks that we usually think of as requiring intelligence. Particularly useful explicit representations of knowledge are those that can be interpreted as making declarative statements. We call knowledge represented in this way *declarative knowledge* because it is contained in declarations about the world. Such statements typically are stored in symbol structures that are accessed by the procedures that use this knowledge.

There are several reasons to prefer declaratively represented knowledge when designing intelligent machines. One advantage is that such knowledge can be changed more easily. To make a small change to a machine's declarative knowledge, usually we need to change just a few statements. Even small adjustments to procedural knowledge, on the other hand, may require extensive changes to the program. Knowledge represented declaratively can be used for several different purposes, even purposes not explicitly anticipated at the time the knowledge is assembled. The knowledge base itself does not have to be repeated for each application, nor does it have to be specifically designed for each application. Declarative knowledge often can be extended, beyond that explicitly represented, by *reasoning* processes that derive additional knowledge. Finally, declarative knowledge can be accessed by *introspective* programs, so that a machine can answer questions (for itself or for others) about what it knows. A price is paid for these advantages, however. Using declarative knowledge usually is more costly and slower than is directly applying procedural knowledge. We give up efficiency to gain flexibility.

It is tempting to speculate about the roles of these two kinds of knowledge in biological organisms. Many insects and other not-very-brainy creatures seem so well attuned to their environments that it is difficult to avoid saying that they have a great deal of knowledge about their worlds. A spider, for example, must use quite a bit of knowledge about materials and structures in spinning a web. Once we understand such creatures better, it seems likely that we will conclude that the knowledge they have evolved about their special niches is procedural. On the other hand, when a human mechanical engineer is consciously thinking about a new bridge design, it seems likely that he refers to declaratively represented knowledge about materials and structures. Admittedly, humans often (perhaps even usually) use procedural knowledge also. The tennis knowledge *used* by a champion player seems procedural, whereas that *taught* by an excellent teacher seems declarative. Perhaps when the distinctions between declarative and procedural knowledge are more clearly understood by computer scientists, they will indeed help biologists and psychologists characterize the knowledge of animals.

In any case, intelligent machines will need both procedural and declarative knowledge. Thus, it is difficult to see how we can study them properly without involving *all* of computer science. The most flexible kinds of intelligence, however, seem to depend strongly on declarative knowledge, and AI has concerned itself more and more with that subject. Our emphasis on declarative knowledge in this book should not be taken to imply that we think procedural knowledge unimportant. For example, when declarative knowledge is used over and over again for the same specific purpose, it would be advisable to compile it into a procedure tailored for that purpose. Nevertheless, the study of representing and using declarative knowledge is such a large and important subject in itself that it deserves book-length treatment.

The book is divided roughly into four parts. In the first five chapters, we present the main features of what is commonly called the *logicist* approach to AI. We begin by describing *conceptualizations* of the subject matter about which we want our intelligent systems to have knowledge. Then we present the syntax and semantics of the *first-order predicate calculus*, a declarative language in which we can write sentences about these conceptualizations. We then formalize the process of inference. Finally, we discuss a simple but powerful inference procedure called *resolution*, and we show how it can be used in reasoning systems.

In the next three chapters, we broaden the logical approach in various ways to deal with several inadequacies of strict logical deduction. First, we describe methods that allow *nonmonotonic reasoning*; i.e., reasoning in which tentative conclusions can be derived. Next, we discuss extensions that permit systems to learn new facts. Then, we show how to represent and reason with knowledge that is not certain.

In the next two chapters, we expand our language and its semantics by introducing new constructs, called *modal operators*, that facilitate representing and reasoning about knowledge about what other agents know or believe. Then, we show how the whole process of writing predicate-calculus sentences to capture conceptualizations can be turned in on itself at the *metalevel* to permit sentences about sentences and about reasoning processes.

In the final three chapters, we concern ourselves with agents that can perceive and act in the world. We first discuss the representation of knowledge about states and actions. Then, we show how this knowledge can be used to derive plans to achieve goals. Finally, we present a framework that allows us to relate sensory knowledge and inferred knowledge and that allows us to say how this knowledge affects an intelligent agent's choice of actions.

1.1 Bibliographical and Historical Remarks

The quest to build machines that think like people has a long tradition. Gardner [Gardner 1982] attributed to Leibniz the dream of "a universal algebra by which all knowledge, including moral and metaphysical truths, can some day be brought within a single deductive system." Frege, one of the founders of modern symbolic logic, proposed a notational system for mechanical reasoning [Frege 1879]. When digital computers were first being developed in the 1940s and 1950s, several researchers wrote programs that could perform elementary reasoning tasks, such as proving mathematical theorems, answering simple questions, and playing board games such as chess and checkers. In 1956, several of these researchers attended a workshop on AI at Dartmouth College, organized by McCarthy (who, incidentally, suggested the name *Artificial Intelligence* for the field) [McCorduck 1979]. (McCorduck's book is an interesting, nontechnical history of early AI work and workers.) Many of the important first papers about AI are contained in the collection *Computers and Thought* [Feigenbaum 1963].

From AI's very beginnings, people have pursued many approaches to the discipline. One, based on building parallel machines that could *learn* to recognize patterns, occupied many AI researchers during the 1960s and continues as one strand of what has come to be called *connectionism.* See [Nilsson 1965] for an example of some of the early work using this approach, and [Rumelhart 1986] for a collection of connectionist papers.

The computational manipulation of arbitrary symbolic structures (as opposed to operations on numbers) is at the heart of much work in AI. The idea that symbol manipulation is a sufficient process for explaining intelligence was forcefully stated in the *physical symbol system hypothesis* of Newell and Simon [Newell 1976]. The need for manipulating symbols led to the development of special computer languages. LISP, invented by McCarthy in the late 1950s [McCarthy 1960], continues to be the most popular of these languages. PROLOG [Colmerauer 1973, Warren 1977], stemming from ideas proposed by Green [Green 1969a], Hayes [Hayes 1973b], and Kowalski [Kowalski 1974, Kowalski 1979a] is rapidly gaining adherents. Much of the work in AI still is characterized mainly by the use of sophisticated symbol manipulation to perform complex reasoning tasks.

One articulation of the symbol-manipulating approach uses *production systems*, a term that has been used rather loosely in AI. Production systems derive from a computational formalism proposed by Post [Post 1943] based on string-replacement rules. The closely related idea of a *Markov algorithm* [Markov 1954, Galler 1970] involves imposing an order on the replacement rules and using this order to decide which applicable rule to apply next. Newell and Simon [Newell 1972, Newell 1973] used string-modifying production rules, with a simple control strategy, to model certain types of human problem-solving behavior. An earlier textbook by Nilsson [Nilsson 1980] used production systems as an organizing theme. More recently, the OPS family of symbol-manipulating computer-programming languages has been based on production rules [Forgy 1981, Brownston 1985]. Work on SOAR by Laird, Newell, and Rosenbloom [Laird 1987] and on *blackboard systems* by a variety of researchers [Erman 1982, Hayes-Roth 1985] can be regarded as following the production system approach.

Another important aspect of AI is *heuristic search*. Search methods are described as a control strategy for production systems in [Nilsson 1980]. Pearl's book [Pearl 1984] gave a thorough mathematical treatment of heuristic search, and his review article summarized the subject [Pearl 1987]. Lenat's work [Lenat 1982, Lenat 1983a, Lenat 1983b] on the nature of heuristics resulted in systems that exploit general heuristic properties in specific problems.

The view taken toward AI in this book follows the theme hinted at by Leibniz and Frege and then substantially elaborated and developed into specific proposals by McCarthy [McCarthy 1958 (the *advice taker* paper), McCarthy 1963]. It is based on two related ideas. First, the knowledge needed by intelligent programs can be expressed as declarative sentences in a form that is more or less independent of the uses to which that knowledge might later be put. Second, the reasoning performed by intelligent programs involves logical operations on these sentences. Good accounts of the importance of logic in AI, for representation and for reasoning, have been written by Hayes [Hayes 1977], Israel [Israel 1983], Moore [Moore 1982, Moore 1986], and Levesque [Levesque 1986].

Several people, however, have argued that logic has severe limitations as a foundation for AI. McDermott's article contained several cogent criticisms of logic [McDermott 1987a], whereas Simon emphasized the role of search in AI [Simon 1983]. Many AI researchers have stressed the importance of specialized procedures and of procedural (as opposed to declarative) representations of knowledge (see, for example, [Winograd 1975, Winograd 1980]). Minsky has claimed that intelligence in humans is the result of the interaction of a very large and complex assembly of loosely connected subunits operating much like a *society* but within a single individual [Minsky 1986].

Notwithstanding the various criticisms of logic, there does seem to be an emerging consensus among researchers that logical tools are important, at the very least, for helping us to analyze and understand AI systems. Newell [Newell 1982] made that point in his article about the *knowledge level.* The work of Rosenschein and Kaelbling on *situated automata* is a good example of an approach to AI that acknowledges the analytic utility of logic while pursuing an alternative implementational strategy [Rosenschein 1986]. The assertion that predicate calculus and logical operations can also usefully serve directly in the implementation of AI systems as a representation language and as reasoning processes, respectively, is a much stronger claim.

Several thinkers have claimed that none of the techniques currently being explored will ever achieve true, human-level intelligence. Prominent among these are the Dreyfuses, who argued that symbol manipulation operations are not the foundation of intelligence [Dreyfus 1972, Dreyfus 1981, Dreyfus 1986] (although their suggestions about what might be needed seem compatible with the claims of the connectionists). Winograd and Flores argued, mainly, that whatever mechanistic processes are involved in thinking, they are probably too complicated to be fully expressed in artificial machines designed and built by human engineers [Winograd 1986]. Searle attempted to distinguish between *real* thought and mere *simulations* of thought by rule-like computations [Searle 1980]. He also seemed to claim that computer-like machines built of silicon, for example, will not do, although machines built according to different principles out of protein might. Taking a somewhat different tack, Weizenbaum argued that, even if we could build intelligent machines to perform many human functions, it might be unethical to do so [Weizenbaum 1976].

There are several other good AI textbooks. Most of them differ from this one in that they do not emphasize logic as much as we do, and they describe applications of AI such as natural-language processing, expert systems, and vision. The books by Charniak and McDermott, Winston, and Rich are three such texts [Charniak 1984, Winston 1977, Rich 1983]. The book by Boden [Boden 1977] treats some of the philosophical issues related to AI. In addition to these books, the reader might also refer to encyclopedic collections of short articles about key ideas in AI [Shapiro 1987, Barr 1982, Cohen 1982].

Many important articles describing AI research appear in the journal *Artificial Intelligence.* In addition, there are several other relevant journals, including the *Journal of Automated Reasoning*, *Machine Learning*, and *Cognitive Science.* Several articles are reprinted in special collections. The American Association for Artificial Intelligence and other national organizations hold annual conferences with published proceedings [AAAI 1980]. The International Joint Council on Artificial Intelligence holds biannual conferences with published proceedings [IJCAI 1969]. Technical notes and memoranda published by the several university and industrial laboratories performing research in AI are available in microfilm from Scientific DataLink (a division of Comtex Scientific Corporation) in New York.

For an interesting summary of the opinions of several AI reseachers about the status of the field during the mid-1980s see [Bobrow 1985]. Trappl's book contains articles about the social implications of AI [Trappl 1986].

Exercises

1. *Structure and behavior.* It is common in discussing the design of artifacts to distinguish between the structure of a device (i.e., its parts and their interconnections) and its behavior (i.e., its external effects).

   a. Give a brief description of a thermostat. Describe both its external behavior and its internal structure. Explain how its structure achieves its behavior.

   b. Is it possible to determine the purpose of an artifact unambiguously, given its behavior? Provide examples to justify your answer.

   c. In his paper "Ascribing Mental Qualities to Machines," John McCarthy[McCarthy 1979b] suggests that it is convenient to talk about artifacts (such as thermostats and computers) as having mental qualities (such as beliefs and desires). For example, according to McCarthy, a thermostat *believes* it is too hot, too cold, or just right, and *desires* that it be just right. Try to adopt McCarthy's viewpoint and indicate the beliefs and desires you think an alarm clock possesses.

2. *Missionaries and cannibals.* Three missionaries and three cannibals seek to cross a river. A boat is available that can hold two people and can be navigated by any combination of missionaries and cannibals involving one or two people. If at any time the missionaries on either bank of the river or en route on the river are outnumbered by cannibals, the cannibals will indulge their anthropophagic tendencies and do away with the missionaries.

   a. Find the simplest schedule of crossings that will permit all the missionaries and cannibals to cross the river safely.

   b. State at least three facts about the world you used in solving the problem. For example, you had to know that a person can be in only one place at a time.

   c. Describe the steps that you took to solve the problem. For each step, record the facts or assumptions you used and the conclusions you drew. The purpose of this part of the problem is to get you to think about the process of solving a problem, not just to arrive at the final solution. Do just enough to get a feel for this distinction.

CHAPTER 2
Declarative Knowledge

AS WE HAVE ALREADY ARGUED, intelligent behavior depends on the knowledge an entity has about its environment. Much of this knowledge is descriptive and can be expressed in *declarative* form. The goal of this chapter is to elucidate the issues involved in formally expressing declarative knowledge.

Our approach to formalizing knowledge is much the same as that of scientists who describe the physical world; in fact, our language is similar to that used to state results in mathematics and the natural sciences. The difference is that in this book we are concerned with the issues of formalizing knowledge, rather than with discovering the knowledge to be formalized.

2.1 Conceptualization

The formalization of knowledge in declarative form begins with a *conceptualization.* This includes the objects presumed or hypothesized to exist in the world and their interrelationships.

The notion of an *object* used here is quite broad. Objects can be concrete (e.g., this book, Confucius, the sun) or abstract (e.g., the number 2, the set of all integers, the concept of justice). Objects can be primitive or composite (e.g., a circuit that consists of many subcircuits). Objects can even be fictional (e.g., a unicorn, Sherlock Holmes, Miss Right). In short, an object can be anything about which we want to say something.

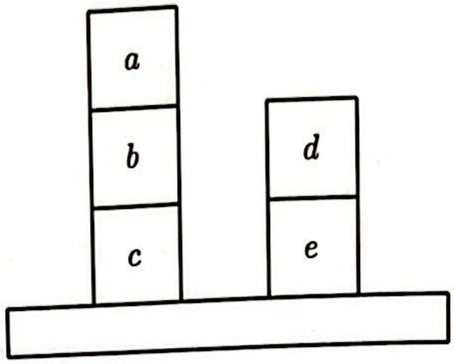

Figure 2.1 A scene from the Blocks World.

Not all knowledge-representation tasks require that we consider all the objects in the world; in some cases, only those objects in a particular set are relevant. For example, number theorists usually are concerned with the properties of numbers and usually are not concerned with physical objects such as resistors and transistors. Electrical engineers usually are concerned with resistors and transistors and usually are not concerned with buildings and bridges. The set of objects about which knowledge is being expressed is often called a *universe of discourse.*

As an example, consider the Blocks World scene in Figure 2.1. Most people looking at this figure interpret it as a configuration of toy blocks. Some people conceptualize the table on which the blocks are resting as an object as well; but, for simplicity, we ignore it here.

The universe of discourse corresponding to this conceptualization is the set consisting of the five blocks in the scene.

$$\{a, b, c, d, e\}$$

Although in this example there are finitely many elements in our universe of discourse, this need not always be the case. It is common in mathematics, for example, to consider the set of all integers, or the set of all real numbers, or the set of all n-tuples of real numbers, as universes with infinitely many elements.

A *function* is one kind of interrelationship among the objects in a universe of discourse. Although we can define many functions for a given set of objects, in conceptualizing a portion of the world we usually emphasize some functions and ignore others. The set of functions emphasized in a conceptualization is called the *functional basis set.*

For example, in thinking about the Blocks World, it would make sense to conceptualize the partial function *hat* that maps a block into the block on top of it, if any such block exists. The tuples corresponding to this partial function are as follows:

$$\{\langle b, a\rangle, \langle c, b\rangle, \langle e, d\rangle\}$$

When concentrating on spatial relationships, we would probably ignore functions that do not have any spatial significance, such as the *rotate* function that maps blocks into blocks according to the alphabetic order of their labels.

$$\{\langle a, b\rangle, \langle b, c\rangle, \langle c, d\rangle, \langle d, e\rangle, \langle e, a\rangle\}$$

A *relation* is the second kind of interrelationship among objects in a universe of discourse. As we do with functions, in conceptualizing a portion of the world, we emphasize some relations and ignore others. The set of relations in a conceptualization is called the *relational basis set.*

In a spatial conceptualization of the Blocks World, there are numerous meaningful relations. For example, it makes sense to think about the *on* relation that holds between two blocks if and only if one is immediately above the other. For the scene in Figure 2.1, *on* is defined by the following set of tuples.

$$\{\langle a, b\rangle, \langle b, c\rangle, \langle d, e\rangle\}$$

We might also think about the *above* relation that holds between two blocks if and only if one is anywhere above the other.

$$\{\langle a, b\rangle, \langle b, c\rangle, \langle a, c\rangle, \langle d, e\rangle\}$$

The *clear* relation holds of a block if and only if there is no block on top of it. For the scene in Figure 2.1, this relation has the following elements.

$$\{a, d\}$$

The *table* relation holds of a block if and only if that block is resting on the table.

$$\{c, e\}$$

The generality of relations can be determined by comparing their elements. Thus, the *on* relation is less general than the *above* relation since, when viewed as a set of tuples, it is a subset of the *above* relation. Of course, some relations are empty (e.g., the *unsupported* relation), whereas others consist of all n-tuples over the universe of discourse (e.g., the *block* relation).

It is worthwhile to note that, for a finite universe of discourse, there is an upper bound on the number of possible n-ary relations. In particular, for a universe of discourse of size b, there are b^n distinct n-tuples. Every n-ary relation is a subset of these b^n tuples. Therefore, an n-ary relation must be one of at most $2^{(b^n)}$ possible sets.

Formally, a *conceptualization* is a triple consisting of a universe of discourse, a functional basis set for that universe of discourse, and a relational basis set. For example, the following triple is one conceptualization of the world in Figure 2.1.

$$\langle \{a, b, c, d, e\}, \{hat\}, \{on, above, clear, table\} \rangle$$

Note that, although we have have written the *names* of objects, functions, and relations here, the conceptualization consists of the objects, functions, and relations themselves.

No matter how we choose to conceptualize the world, it is important to realize that there are other conceptualizations as well. Furthermore, there need not be any correspondence between the objects, functions, and relations in one conceptualization and the objects, functions, and relations in another.

In some cases, changing one's conceptualization of the world can make it impossible to express certain kinds of knowledge. A famous example of this is the controversy in the field of physics between the view of light as a wave phenomenon and the view of light in terms of particles. Each conceptualization allowed physicists to explain different aspects of the behavior of light, but neither alone sufficed. Not until the two views were merged in modern quantum physics were the discrepancies resolved.

In other cases, changing one's conceptualization can make it more difficult to express knowledge, without necessarily making it impossible. A good example of this, once again in the field of physics, is changing one's frame of reference. Given Aristotle's geocentric view of the universe, astronomers had great difficulty explaining the motions of the moon and other planets. The data were explained (with epicycles, etc.) in the Aristotelian conceptualization, although the explanation was extremely cumbersome. The switch to a heliocentric view quickly led to a more perspicuous theory.

This raises the question of what makes one conceptualization more appropriate than another for knowledge formalization. Currently, there is no comprehensive answer to this question. However, there are a few issues that are especially noteworthy.

One such issue is the *grain size* of the objects associated with a conceptualization. Choosing too small a grain can make knowledge formalization prohibitively tedious. Choosing too large a grain can make it impossible. As an example of the former problem, consider a conceptualization of the scene in Figure 2.1 in which the objects in the universe of discourse are the atoms composing the blocks in the picture. Each block is composed of enormously many atoms, so the universe of discourse is extremely large. Although it is in principle possible to describe the scene at this level of detail, it is senseless if we are interested in only the vertical relationship of the blocks made up of those atoms. Of course, for a chemist interested in the composition of blocks, the atomic view of the scene might be more appropriate. For this purpose, our conceptualization in terms of blocks has too large a grain.

Finally, there is the issue of *reification* of functions and relations as objects in the universe of discourse. The advantage of this is that it allows us to consider properties of properties. As an example, consider a Blocks World conceptualization in which there are five blocks, no functions, and three unary relations, each corresponding to a different color. This conceptualization allows us to consider the colors of blocks but not the properties of those colors.

$$\langle\{a, b, c, d, e\}, \{\}, \{red, white, blue\}\rangle$$

We can remedy this deficiency by *reifying* various color relations as objects in their own right and by adding a partial function—such as *color*—to relate blocks to colors. Because the colors are objects in the universe of discourse, we can then add relations that characterize them; e.g., *nice*.

$$\langle\{a, b, c, d, e, red, white, blue\}, \{color\}, \{nice\}\rangle$$

Note that, in this discussion, no attention has been paid to the question of whether the objects in one's conceptualization of the world really exist. We have adopted neither the standpoint of *realism*, which posits that the objects in one's conceptualization really exist, nor that of *nominalism*, which holds that one's concepts have no necessary external existence. Conceptualizations are our inventions, and their justification is based solely on their utility. This lack of commitment indicates the essential ontological promiscuity of AI: Any conceptualization of the world is accommodated, and we seek those that are useful for our purposes.

2.2 Predicate Calculus

Given a conceptualization of the world, we can begin to formalize knowledge as sentences in a language appropriate to that conceptualization. In this section, we define a formal language called *predicate calculus.*

All the sentences in predicate calculus are strings of characters arranged according to precise rules of grammar. For example, we can express the fact that block a is above block b by taking a relation symbol such as `Above` and object symbols `A` and `B` and combining them with appropriate parentheses and commas, as follows.

```
Above(A,B)
```

One source of expressiveness in predicate calculus is the availability of logical operators that allow us to form complex sentences from simple ones without specifying the truth or falsity of the constituent sentences. For example, the following sentence using the operator `∨` states that either