Gordon S. Novak, Jr.

Artificial Intelligence

Constructing computer programs that exhibit intelligence and symbolic reasoning is an ongoing research effort by University of Texas scientists

Near the end of World War II scientists developed vacuum-tube circuits that could perform arithmetic, and it became possible to produce a calculating machine with about the power of a present-day pocket calculator. Such a machine, of course, would not have fit into a pocket; it would have occupied half a building with the other half of the building needed for air conditioning to cool it. There was a problem, however. If such a large and expensive machine were constructed like an ordinary calculator, with a person pushing the buttons, its speed would be limited to the speed of the button-pusher, and its potential power could not be realized. The great mathematician John von Neumann developed the concept of a stored program, with which the machine would, in effect, push its own buttons in the sequence specified by the program. And the modern computer was born.

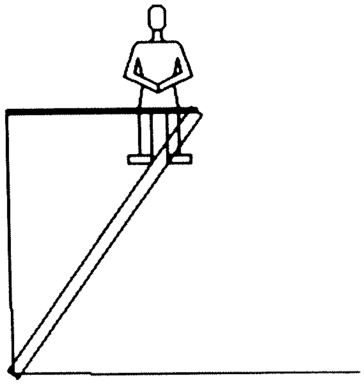
Since that time the computer has become a prodigious calculating engine, capable of performing tens of millions of operations per second. Just as important as the increase in speed has been the rapid drop in the price of computation. Preschool children now own computers that would have been the envy of researchers only a few decades ago. The price of computation has been dropping by a factor of two every two years (compounded, this means a factor of 1,000 every twenty years), and this trend seems likely to continue for at least the next decade or two.

The sudden explosion of fast and cheap computation presents a problem similar to that which faced von Neumann: the benefit mankind can derive from computers is now limited primarily by the difficulty of telling them what we want them to do for us. Our existing programming languages, only a few steps removed from pushing the buttons of a calculator, have kept the production of applications software slow and expensive despite the rapid drop in hardware costs. Computers remain relatively rigid, unable to respond to changes in their applications without extensive reprogramming. It is a dilemma that could be the stuff of Greek mythology: we are given a computational servant of awesome power that will do anything we want—but only if we specify with absolute precision what that is.

Humans, despite their fallibility and slowness in certain kinds of information processing (such as arithmetic), are remarkably robust at dealing with a wide variety of ill-structured problems. As computers become more powerful and cheaper, such robustness in dealing with ill-structured tasks—in a word, intelligence—will become much more important than speed at performing well-structured tasks. Artificial intelligence (commonly abbreviated "AI") is the branch of computer science that seeks to understand intelligent behavior in information-processing terms and to construct computer programs that exhibit intelligence. Areas of artificial intelligence research include perception (machine vision, speech understanding), motor control (robotics), understanding natural languages (such as English), expert reasoning, commonsense reasoning, problem solving, planning, and diagnosis.

Several features distinguish AI programs from ordinary computer programs. One such feature is the kinds of computations performed. Most computer programs are largely concerned with arithmetic operations. Business data processing programs often manipulate strings of alphabetic characters (for example, names and addresses of customers) but rarely are concerned with the *meanings* of these strings. AI programs, in contrast, are largely concerned with *symbolic* reasoning; that is, they deal with symbols that represent objects in the world and relationships among those objects. The reasoning performed by AI programs is often based on syllogistic inferences made from general rules: if Socrates is a man and all men are mortal, then Socrates is mortal. Finally, AI programs often have a more flexible method of control than ordinary programs. Ordinary programs typically do things in a rigidly predetermined sequence. AI programs often are able to change the sequence of what they are doing based on new facts; this allows them to be more flexible when confronted with unexpected situations.

An important class of AI programs is *expert systems*, which are programs that attempt to duplicate human expertise in specialized areas like making medical diagnoses, manipulating mathematical formulas, solving physics problems, or writing computer programs. Ironically, ex-



```
(PROBLEM P8 , NUMBER 19 FROM SCHAUM PAGE 25)
(THE FOOT OF A LADDER RESTS AGAINST A VERTICAL WALL AND ON A HORIZONTAL
FLOOR)(THE TOP OF THE LADDER IS SUPPORTED FROM THE WALL BY A HORIZONTAL
ROPE 30 FT LONG)(THE LADDER IS 50 FT LONG , WEIGHS 100 LB WITH ITS
CENTER OF GRAVITY 20 FT FROM THE FOOT , AND A 150 LB MAN IS 10 FT FROM
THE TOP)(DETERMINE THE TENSION IN THE ROPE)  ANSWER:  120.0 LB
```

FIGURE 1: An example of a problem solved by the ISAAC program. The English sentences shown are the input to the program; the picture is generated by the program from its understanding of the English.

pert reasoning has proved easier to emulate than everyday common sense. The reason for this is that specialist areas tend to be narrow, so that the amount of knowledge required, though large, is manageable. Expert systems derive their power primarily from the knowledge that they embody about the area of expertise, rather than from fast computation per se; for this reason, they are sometimes referred to as *knowledge-based systems.*

As an example of such a system, I shall describe a program called ISAAC (after Isaac Newton) that understands and solves physics problems, stated in English, in the area of rigid-body statics. It has solved more than forty problems, such as the following problem from *Schaum's Outline of College Physics:*

> The foot of a ladder rests against a vertical wall and on a horizontal floor. The top of the ladder is supported from the wall by a horizontal rope 30 ft long. The ladder is 50 ft long, weighs 100 lb with its center of gravity 20 ft from the foot, and a 150 lb man is 10 ft from the top. Determine the tension in the rope.

The ISAAC program reads the problem statement and composes a symbolic model that represents the objects in the problem and their properties and relationships. Next, it makes reasonable assumptions to fill in missing information and builds a geometric model that relates the objects in the problem to a common coordinate system. It decides how to model objects in physical terms and writes equations that express the relationships among objects according to physical laws. The equations are solved algebraically to find the answer to the problem. Finally, the program constructs a picture model and draws a picture of the problem on the screen of the computer terminal. The picture in Figure 1 was generated by the program from the English statement of the problem shown above. How does ISAAC understand and solve such problems?

The first task for the program is to read and understand the problem statement. Each sentence in the problem statement is *parsed*—a process much like the "diagramming" of sentences taught in grammar school, in which the parts of speech and roles of words in the sentence are determined, and a structure is built that ties together the component parts of the sentence and shows the relationships of the parts. The ISAAC program contains a small lexicon that gives the parts of speech and other information for the words that occur in its class of physics problems. The grammar of English used by ISAAC is expressed as small computer programs, each of which knows how to analyze a particular type of phrase (for example, a prepositional phrase). However, a lexicon and grammar alone are not enough to allow English sentences to be understood correctly, and a number of difficulties must be overcome. Many words can have multiple parts of speech and multiple meanings; "foot" can be a part of a person or a unit of length or a location on a ladder. Modifying phrases may potentially modify different parts of a sentence. The sentence "I saw the man on the hill with the telescope" is ambiguous (Who was on the hill? Who had the telescope?). The combinations of these sources of ambiguity allow a computer parser to find literally hundreds of possible interpretations for sentences that humans find perfectly clear. Finally, things that are "obvious" are often omitted from written and spoken language.

To understand a sentence correctly, it is necessary to use pragmatic knowledge of how things are in the real world. Consider the phrase in the example problem, "a 150 lb man is 10 ft from the top." The program must know that it needs to ask the question "from the top of what?" and then it must be able to answer its own question. Since a ladder has been mentioned, and a ladder has a top, it must be the top of the ladder that is meant. "10 ft from the top" must mean a location on the ladder (not some other location that happens to be ten feet from the ladder's top). Finally, the sentence says the man *is* there, but it

Dr. Gordon S. Novak Jr. is director of the Artificial Intelligence Laboratory and an associate professor of computer sciences at The University of Texas at Austin. Before coming to the University, he was a research scientist at the Artificial Intelligence Laboratory of SRI International and spent two years as a visiting faculty member at the Heuristic Programming Project and Computer Science Department at Stanford University. He holds a B.S. degree in electrical engineering and M.A. and Ph.D. degrees in computer sciences from The University of Texas.

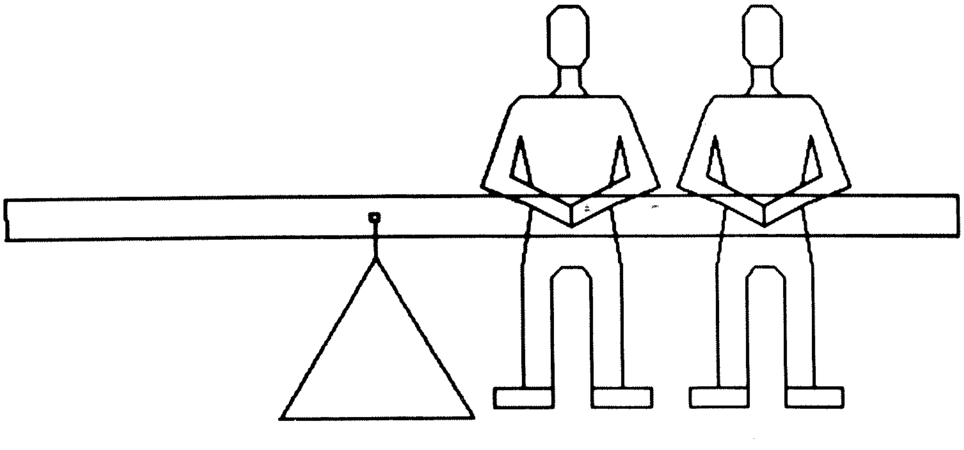
doesn't say what he is doing there; since the ladder is supported in two places and the weight of the man is mentioned, the program infers that the man must be *standing* on the ladder.

As this example illustrates, understanding English is no easy matter. An English sentence does not *contain* the meaning that is to be conveyed but, rather, a sentence is like a computer program that causes the mind of the reader to *construct* the meaning from what the reader already knows. Understanding even ordinary English requires a great deal of "reading between the lines," and that requires much knowledge. Giving a computer program enough knowledge to understand English statements about even a limited area is a large task.

Once the problem statement has been read, ISAAC's memory contains a network of symbols that describes the objects in the problem, their properties, and their relationships. For example, the network represents the facts that the ladder has a weight of 100 pounds, that it is attached to the rope, and that it supports the man. One might wonder how it is possible to tell whether or not a given collection of symbols and relationships is in fact a correct "understanding" of a given set of English statements.

Experimental work with actual problems is very important for artificial intelligence research, because failures often involve knowledge so obvious that nobody ever thought to put it in the program. Figure 2 illustrates a "failure" due to a lack of knowledge of the ISAAC program. The problem as shown is perfectly consistent with the English problem statement, and it is solved correctly. Such "failures" are valuable indicators of reasoning that humans do without conscious effort but that must be explicitly modeled in AI programs.

After the problem statement has been read, inferences are made to fill in facts that are missing. If no weight had been given for the ladder, for example, its weight would have been assumed to be zero. Based on the network of facts, the program decides how to model objects in the problem as "physics objects." (For example, the man on the ladder is modeled as a point mass.) A geometric model is then constructed that relates the objects to a common coordinate system. The Figure 1 example requires that the program solve a right triangle in order to determine the positions of the objects. Equations are written that describe the interactions of the objects according to physical laws and are then solved to find the answer. (For the problem in Figure 1, the tension in the rope is 120 lb.) ISAAC solves equations algebraically, just as people do. By using algebra rather than being restricted to numerical calculations, ISAAC is able to solve problems in which some information is unknown and is represented by a variable. (For example, a problem in which the length of the ladder is unspecified and the program says, in effect, "let the length of the ladder be *l*.") The ability to perform algebraic manipulation is important because an algebraic solution is true for a whole class of problems, rather than for only a single problem.



```
(PROBLEM P25 , NUMBER 3 FROM NILSSON PAGE 78)
(HENRY AND PAUL CARRY A SACK WEIGHING 200.0 NT ON A POLE)(THE POLE IS
5.0 M LONG)(THE WEIGHT IS 2.0 M FROM THE END OF THE POLE)(HENRY IS 2.0 M
 FROM THE WEIGHT AND PAUL IS 1.0 M FROM THE WEIGHT)(WHAT FORCE DOES EACH
 BOY EXERT)
ANSWER:  ( (MINUS FORCE93) , -200.0 ) NT  , ( FORCE93 , 400.0 ) NT
```

FIGURE 2: A misunderstanding caused by a lack of knowledge of the program. The problem as shown in the picture is consistent with the English problem statement, and it is solved correctly. However, people do not usually carry weights on poles in this manner, so this is probably not what the author of the problem intended. Humans are able to understand imprecise English correctly because they can use their knowledge to "patch over" areas of missing or incorrect information.

Finally, the program constructs a picture model for the problem and draws a picture of it on the computer terminal. (The pictures generated by the program for the example problems are shown in the figures.) ISAAC has a set of programs that can draw pictures of individual objects; it also knows how to relate named locations on objects to locations on their pictures. The picture model is constructed by scaling objects to appropriate sizes and "pasting" them together at their points of attachment. The result is a list of the objects that constitute the picture, with the position, size, and orientation of each. Here, again, knowledge is required in order to generate a sensible picture. ISAAC assumes that a person who is supported by some object is supported by his feet unless specified otherwise; thus the man in Figure 1 is shown standing on the ladder. ISAAC knows to draw a man larger than a boy, and to draw a heavier man larger than a lighter man.

Running on a desktop computer, ISAAC reads English at a speed of 5,000 words per minute and solves a complete problem in about five seconds. The significance of this program lies not in the numerical calculations it performs, which are rather trivial, but in its ability to understand and analyze problems that are described informally in English. Expert-systems researchers seek to combine the accumulated knowledge of science and of expert practitioners of science with the computational power of the computer and make both easily available. Giving the computer the knowledge to understand problems stated in the user's language rather than in an arcane programming language will allow its vast computational power to be harnessed and made productive for the many areas of intellectual activity that lie beyond mere numerical calculation.