Essays of an Information Scientist, Vol:6, p.404-412, 1983 Current Contents, #49, p.5-13, December 5, 1983

Current Comments®

Artificial Intelligence: Using Computers to Think about Thinking. Part 1. Representing Knowledge

Number 49 December 5, 1983

In 1950, Alan M. Turing, the late deputy director of the University of Manchester's Computing Laboratory in England, proposed a novel test to determine whether a machine was capable of thinking. In this test, an interrogator has a teletype conversation with a man and a woman, both of whom must try to convince the interrogator that they are the woman. At some point unknown to the interrogator, the man is replaced by a machine. If the interrogator is fooled as often by the machine as by the man, that machine can be said to have displayed intelligent behavior.[1]

Some 30 years after Turing proposed this test, many aspects of human behavior have been simulated by a computer. Programs have been designed to play checkers[2] and chess,[3] prove mathematical theorems,[4,5] and even mimic the behavior of a paranoid human being.[6] Despite the success of these and many other programs, none of the researchers investigating what's been variously called "applied epistemology" or "artificial intelligence" (AI) would claim this means the "thinking machine" has arrived. Instead, they would agree that these programs have contributed important information about human behavior, and how computers can simulate it.

The first part of this two-part essay will review some of the theories AI researchers have developed to explain human "information processing." The second part of the essay will cover some applications of AI research. These include programs used in robotics, programs that communicate with computer users in natural languages such as English, and "expert systems" which help chemists, physicians, and others perform decision-making tasks. The "pioneer" expert system, DENDRAL, will be discussed in some detail.[7,8]

AI grew out of the convergence of ideas in several different fields, and the availability of new technologies. According to Avrom Barr and Edward A. Feigenbaum, Stanford University, California, the single most important factor contributing to the birth of the field was the invention of the computer.[9] They point out that human beings have always drawn analogies between mechanical devices and their own behavior. Computers, with their memories and information-processing abilities, naturally invited analogies with the human brain.

Shortly after digital computers became available, computer scientists began creating programs that, they hoped, would perform tasks generally considered to require intelligence. Their earliest efforts were directed at programming computers to solve puzzles, play games such as chess, backgammon, and checkers, solve mathematical theorems, and translate text from one language to another. The early computer programs performed these tasks, but not very well. For example, chess programs were successful at following the step-by-step instructions for moving chessmen. But computers couldn't independently generate the strategies needed to win. In those days, computer scientists were not ready to program the *reasoning* required to make appropriate moves. Indeed, computer scientists couldn't easily explain how they chose a strategy when playing chess. Barr and Feigenbaum note, "The realization that the detailed steps of almost all intelligent human activity were unknown marked the beginning of Artificial Intelligence as a separate part of computer science."[9] (p. 6)

Today, AI represents a blend of cognitive psychology and computer science. Computer scientists base their programs on many of the theories developed to explain human cognition and learning. For example, one of the most basic AI methods for incorporating knowledge in a computer program draws upon theories on how humans acquire new knowledge. Jean Piaget, the late director of the University of Geneva's International Center for Epistemology, Switzerland, suggested that human knowledge takes the form of simple models of the complex world we experience.[10] The process of learning involves noting where new data fail to meet our existing models, or expectations, and adapting these models to incorporate the new information. Simply put, existing knowledge is modified and enhanced by new information. As we gain more and more information, our models become more and more sophisticated. Most AI programs now operate by associating, or relating, larger and larger pieces of information with one another until these combined pieces of information have some meaning.[11]

Although simulating human thinking has been the goal of all AI programs, the AI community is presently split into two camps over how these pieces of information should be related to one another by the computer before it can reason. One camp, led by Marvin Minsky, Massachusetts Institute of Technology (MIT), Cambridge, Massachusetts, believes researchers should develop programs that include the type of inconsistent or "fuzzy" information humans generally use. The other camp, led by John McCarthy, Stanford University, believes computers should be designed to reason according to the precise language of mathematical logic.[12]

Most of the AI researchers working on programs that mimic human thinking are building upon the work of linguists and other scholars who, in the 1950s, were trying to develop machines that could translate text. Most early efforts failed because these machines did little more than look up words in a bilingual computer dictionary and rearrange the terms according to rules of syntax. The machines could not deal with the fact that words and phrases can have multiple meanings. As noted in a previous essay, the machine's failure to deal with this ambiguity in language led to a number of amusing translations.[13] One of the earliest English-Russian translating machines converted "Out of sight, out of mind" into "invisible idiot."[14] More importantly, people like Yehoshua Bar-Hillel, Hebrew University, Jerusalem, Israel, predicted that complete machine translation would never be possible.[15]

In the 1960s, Noam Chomsky, MIT, developed an influential theory that helped to explain how people deal with these multiple meanings of words. He proposed that we are born with a mental structure for language which enables us to interpret the sentences we hear.[16] This structure makes it possible to repeat something we've heard some time back, without having to remember the explicit words. AI researchers have built upon this concept of a mental structure in their attempts to represent knowledge in computer programs. Programs designed to mimic human thinking associate new information with concepts or facts already programmed into the computer. The methods used in AI to store and manipulate data so they can be called up and associated in the way we think are called "knowledge representations."[17]

Some of the earliest knowledge representations were designed for natural language (NL) programs which permitted one to communicate with the computer in English, or another native language. Since language is the most tangible reflection of human thought, AI workers reasoned that simulating language would be an important step toward simulating thought. And, as they learned through their failure with early translating machines, simulating language understanding requires sentence comprehension, which requires a great deal of knowledge. Steven K. Roberts, a microprocessor-systems consultant in Dublin, Ohio, explains: "Language has to be considered as only one part of a much more complex communication process, one in which the knowledge and state of mind of the participants are as much responsible for the interpretation of verbal utterances as are the words."[18]

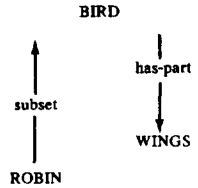
One of the earliest NL programs, called ELIZA, mimicked the responses of a psychiatrist. Developed in 1966 by Joseph Weizenbaum, MIT, each response of the computer is called up from a set of stored sentences associated with words spoken by the patient. For example, whenever the word "mother" is spoken, the computer replies with a stock sentence, such as, "Tell me more about your mother."[19]

ELIZA's human-like responses lured many people into reciting their personal problems to a computer. But the computer actually had no understanding of language—it merely reacted to key words. Most subsequent attempts at language understanding, therefore, focused on programs that could parse, or pick apart a sentence to determine what each word and, ultimately, the sentence meant. In AI programs, a parser uses grammatical rules as well as other sources of knowledge to determine the role each word plays in the sentence, and how these words relate to one another, to mean something.[9]

In the early 1970s, two basic approaches to knowledge representations in NL programs were emerging in the AI community—procedural and declarative representations. In the procedural representation, Barr and Feigenbaum explain, "Knowledge about the world is contained in procedures—small programs that know how to do specific things, how to proceed in a well-specified situation."[9] (p. 155) The procedural representation contains information about the problem to be solved, as well as information about which procedure contains the solution. For example, a parser for an NL-understanding program would know that a noun phrase contains articles, adjectives, and nouns. When confronted with a noun phrase, it would call up routines that process articles, adjectives, and nouns.[9]

In a declarative representation scheme, the direct, explicit relationships between words are programmed into the computer. One of the most influential declarative schemes is the semantic network,[20] proposed as a model for human associative memory by M. Ross Quillian, formerly of Bolt, Beranek & Newman, Inc., Cambridge, Massachusetts. A similar knowledge representation had been developed at IBM some years earlier, according to Manfred Kochen, University of Michigan, Ann Arbor.[21] A semantic network consists of nodes, which are objects, concepts, and events, and of links, or symbolic references between these nodes which represent their relationships. In the network shown in Figure 1, for example, "bird," "wings," and "robin" are concepts. Their relationships are specified as "has-part" and "subset." When given the sentence, "Robins are birds," the computer would search the linkages in this network and

Figure 1: Network segment from a semantic network knowledge representation. Words in uppercase are nodes, or concepts. Words in lowercase are links, or symbolic references between the concepts. This network segment would be interpreted by the computer as, "Since robins are birds, and birds have wings, then robins have wings."



Source: **Barr A & Feigenbaum E A**, eds. *The handbook of artificial intelligence*. Los Altos, CA: William Kaufmann, 1981. Vol. I. p. 157. (Reprinted with permission of the publisher.)

conclude: "Since robins are birds, and birds have wings, then robins have wings."[9] (p. 157)

Subsequent semantic networks[22] built upon Quillian's model, but used different schemes for describing the primitive, or basic, attributes of the information in the data base. In his work on NL programs that could paraphrase a sentence, Roger C. Schank, Yale University, New Haven, Connecticut, developed programs based on what he called "primitives of conceptual dependency."[23] These primitives include the set of actions in which there is a transfer of mental information (telling, hearing, writing); the set in which there's a transfer of possession (giving, taking, buying, selling); and the set involving sensing (watching, smelling, listening). These primitives are used to diagram the semantic structure of a sentence. With these programs, diagrams representing a sentence, and a paraphrase of that sentence, are similar. Moreover, incomplete sentences entered into the program organize themselves in such a way that blank spaces are left for expected items. For example, given the sentence fragment, "John gave," the program expects an object and a recipient of that object.[24]

Schank's work on primitives of conceptual dependency was part of a long-term trend toward viewing language understanding, and ultimately thinking, as a process "in which words serve as cues for retrieving expectations from memory."[24] Schank and Minsky are now working on representation schemes that operate by using the expectations an individual might have when confronted with certain stereotyped situations.

With Minsky's frame, or framework, representation,[25] each frame is dominated by a central concept—such as a bird, person, or event—and includes a set of slots describing attributes of that concept. A bird frame, for example, might include slots for such attributes as egg laying, flying, wings, and singing. Each attribute in the frame is labeled optional or not optional for each of the stereotyped situations created in the program. This makes it possible to determine which attributes are most important for a given situation. For example, in a biological context, flying and singing might be optional to the bird frame. Egg laying and wings would not.[12]

In the knowledge representation developed by Schank and Robert P. Abelson, Yale University, the machine is taught about the real world through frames. It also learns about the world through "scripts" that summarize common human experiences.[26] The stereotyped situations described in these scripts enable the machine to fill in missing information—make assumptions about a statement based on the context in which that statement is made. Schank's restaurant script, for example, describes the events that typically take place from the time a person walks in and is seated until the time he or she pays the bill and leaves. Using the contextual information provided in this script, the computer can resolve ambiguities in language used to describe a visit to a restaurant. From the statement, "I ate a steak," the computer would know that the steak had to be ordered from the menu and paid for. Furthermore, as Schank explains in a *Fortune* article on

AI, the computer would know that "ordering a steak is different from ordering a subordinate or ordering your affairs."[11]

Schank and Minsky's representation schemes are based on explicitly psychological models of human thought. These scientists, and many of their colleagues, believe that if a large enough volume of information can be manipulated by frames or scripts, computerized reasoning may, ultimately, be possible.

But McCarthy and Nils Nilsson, SRI International, Menlo Park, California, disagree. They believe that psychologically based systems tend to be "somewhat fuzzy and mushy"[12]—that is, lacking in precision. Instead, they propose that AI systems be based on the well-formulated languages of mathematical logic, even if these languages aren't "psychologically real."[27]

At their most basic level, AI programs based on logic employ propositions or statements known to be true or false to derive other, more complicated statements that must also be true or false. For example, if it's true that all birds have wings, and that Joe is a bird, you can infer that Joe has wings. McCarthy, however, has found that commonsense reasoning isn't always as clear-cut as mathematical logic. Invariably, exceptions can be found for most of the statements proved true or false. His answer is a new form of logic that he calls "circumscription."[28] This is a form of non-monotonic reasoning which provides for these exceptions. In monotonic reasoning, a computer (or person) might accept the statement, "Birds can fly." But this statement would be false if you were talking about dead birds, or ostriches and penguins. Using circumscription, however, the statement would be put forth as, "Birds can fly unless something prevents them." Then the computer could approach the problem by reasoning, "If Joe is a bird and Joe is not a member of the set 'prevented from flying' then Joe can fly."[12]

Although the knowledge representations described organize knowledge in very different ways, all rely on the same basic set of operating principles. In general, an AI system will include a "knowledge base" which consists of such knowledge structures as semantic networks or predicate calculus expressions. The knowledge base is manipulated by a set of "operators," or rules for using the information in the knowledge base. In chess, for example, these operators are the rules for moving chessmen. The third component of the AI system is the "control strategy" for deciding which operators, or rules, are needed to solve the problem at hand. The object in an AI program is to solve a problem by applying the appropriate sequence of operators to the knowledge base to reach a solution. Each time an operator is applied, the configuration of knowledge in the knowledge base changes. In chess, the goal is to find the right sequence of moves to reach checkmate. Each time an operator (chess rule) is applied, a new board configuration is created.[9]

In AI, each new configuration of knowledge is called a state. The process of solving a problem is generally diagramed as a "search tree," with each node in this branching structure representing a new state, which can be the solution or a step toward the solution. The computer searches along each branch until it reaches a node that represents the correct solution.[9]

The major difficulty in searching a complicated search tree—solving a complex problem—is that examining all alternatives involves an unreasonable amount of time on even the fastest computer. A chess program that looks ahead 52 moves would have to analyze 10^{80} possible courses of action.[11] This problem, called the "combinatorial explosion," plagues all problem-solvers, human or mechanical. Most current knowledge representations, therefore, incorporate information or procedures that limit the search for solutions to a search for the best possible alternatives.

In the 1950s, Allen Newell and J.C. Shaw, formerly of the RAND Corporation, Santa Monica, California, and Herbert A. Simon, Carnegie-Mellon University, Pittsburgh, Pennsylvania, developed an important technique for searching. They equipped their early programs with heuristics, or "good guess" knowledge, to help computers focus on the most promising course of action.[29,30] Heuristics are based on research into the strategies humans use for solving problems. An example of one such heuristic, used for an expert system that helps physicians diagnose infectious diseases, is shown in Figure 2. Computers must, of course, be programmed to use these rules. But once built into the program, they are extremely effective at interpreting new data in a meaningful fashion. Most AI programs now incorporate these heuristics to some extent. A convenient representation for heuristics is called the production rule. Production rules are used in many of the expert systems I'll discuss in the second part of this essay.

Figure 2: Heuristic, or production rule, for the expert system MYCIN.

IF	1) *the infection is primary-bacteremia,* and 2) the site of the culture is one of the sterile sites, and 3) the suspected portal of entry of the organism is the gastrointestinal tract,
THEN	there is suggestive evidence (.7) that the identity of the organism is bacteroides.

Source: **Barr A & Feigenbaum E A**, eds. *The handbook of artificial intelligence*. Los Altos, CA: William Kaufmann, 1981. Vol. II. p. 187. (Reprinted with permission of the publisher.)

Researchers have made great strides in modeling human cognition since the term artificial intelligence was coined by McCarthy in 1956.[31] But most members of the AI community believe that until machines can perform commonsense reasoning, they cannot be called intelligent. And common sense seems to be based on a great deal of implicit knowledge—too much, thus far, for current computers to store and for current programs to manipulate. For example, for a computer to understand the sentence, "I'm going to the store," it must know what a store is, and realize that I'll need money or a credit card to buy something. Giving a program a great deal of commonsense knowledge is a major goal of AI research.

Even if reasoning programs are created, though, AI researchers believe that true thinking machines must await the development of an entirely different type of computer. Current serial, or so-called von Neumann, computers consist of thousands to billions of individual "memory cells," each of which feeds information to a single central processing unit (CPU). The CPU then performs one operation at a time. The human brain, by contrast, performs many different operations at the same time. Many AI investigators believe that a computer is needed now that can perform thousands of operations in parallel, and relate these operations to one another in the same manner humans associate different ideas.[32]

Researchers at MIT; Lawrence Livermore Laboratory, Berkeley, California; and University of Manchester are already using computers with thousands of CPUs that operate in parallel.[32] Other such computers are being built at the University of Maryland, College Park; Columbia University, New York; and New York University, New York.[32] And the "Fifth Generation" project funded by the Japanese government is directed toward the development of special AI hardware and programs.[33] But few programs currently exist that can fully use the "parallelism" of these computers.[32] Very large scale integrated circuitry (VLSI) technology also offers some promise for the development of parallel processing computers. With this technology, circuits that act as processors, memories, and input-output circuits may be etched on a single silicon chip the size of a fingernail. But VLSI is still in a developmental stage.[32]

This brief survey of the literature on AI clearly cannot cover every concern of investigators in this wide-ranging field. For example, no matter how much information a computer contains, unless it can make new associations and expand its own capabilities, it will be limited by a programmer's knowledge and time. So a number of schemes have been developed for teaching computers to learn. These include rote learning, trial and error, and adaptation.[2,21,34-36] Also essential to the growth of the field have been the new list-processing computer languages which make AI programs possible. Although the first such language was IPL, developed by Newell, Shaw, and Simon in 1957,[37] the vast majority of AI researchers currently use LISP, developed by McCarthy in 1958.[38] Unlike such languages as FORTRAN and BASIC, which were designed for "number crunching" or pure calculation, LISP is designed for manipulating symbols. LISP also differs from these other languages in that programs can be changed fairly easily because memory can be easily added to or deleted from any data structure. Perhaps the most important difference between LISP and other computer languages is that "each site in a computer's memory where a symbol is stored also contains directions to other sites where associated symbols reside."[39] For example, the word "elephant" might have pointers to "gray," "has a trunk," and "four-legged." These pointers make it possible for one concept to call up another or, in anthropomorphic terms, for the machine to make associations.[40]

The computer scientists and cognitive psychologists who make up the majority of AI researchers have been joined in their search for an explanation of intelligence by linguists, philosophers, cyberneticists, pattern recognition researchers, and others. Not surprisingly, therefore, papers on AI can be found in such journals as *Behavioral and Brain Sciences*, *Cognition and Brain Theory*, *Transactions on Pattern Recognition and Machine Translation*, *International Journal of Man-Machine Studies*, *Communications of the Association for Computing Machinery*, and *Human Systems Management*. Popular magazines, such as *Robotics Age* and *OMNI*, also carry occasional articles on AI, and *AI Magazine* carries semitechnical reports. The key journals in the field, according to Barr and Feigenbaum,[9] are *Artificial Intelligence*, *Cognitive Science*, and *American Journal of Computational Linguistics*. *Artificial Intelligence* and *Cognitive Science* are covered in *Social Sciences Citation Index*® (*SSCI*®), and *Current Contents*®/*Social & Behavioral Sciences*. *Artificial Intelligence* is also covered in *Science Citation Index*® (*SCI*®); *SCI, Abridged Edition*; and *ISI/CompuMath*®. The field well illustrates the futility of trying to separate "social" from natural or physical sci-

Table 1: *ISI/CompuMath*® research fronts on artificial intelligence. A=research front number. B=research front name. C=number of core papers in the research front. D=number of citing papers in the research front.

A	B	C	D
80-0191	Retrieval processes, computational linguistics, and language processing	9	69
80-0724	Automatic, resolution, and nonresolution theorem proving	3	25
80-0726	Computer-aided diagnosis and clinical judgment	2	18
80-0739	Nonrecursive grammars, natural languages, and inductive inferences of formal languages	6	60
80-1033	Computer assistance for structure elucidation of organic compounds	3	46
80-1155	Cognition, psychological epistemology, and experiments in artificial intelligence	2	20
80-1963	Knowledge-engineering and computer-aided medical decision-making	2	26

ences. That is the problem that underlies the creation of a unified index to science.

By the standards we are used to in the life sciences, AI is not a large field. Even now its total literature is relatively small. For example, the paper in which Turing presents his famous test has "only" been cited about 130 times in *SCI* and *SSCI* since these indexes were started. At such a low annual rate, it is not surprising that AI papers do not yet turn up in the larger research fronts obtained initially from *SCI*. However, once we "extracted" the fields of computer science and math to create our *ISI/CompuMath* data base, research fronts in AI did surface. Research fronts are specialty areas identified when a group of current papers cites one or more core papers for that topic. Eight such fronts are listed in Table 1. Two on this list, "Knowledge-engineering and computer-aided medical decision-making" and "Computer-aided diagnosis and clinical judgment," are concerned with expert systems that help physicians make decisions and diagnoses. The research front entitled "Computer assistance for structure elucidation of organic compounds" also focuses on an expert system, only this is one used by chemists. These expert systems will be discussed in some detail in Part 2 of this essay. We will also provide a list of those core papers and books not already mentioned.

The second part will also discuss a variety of other AI applications, including certain robotic and NL systems. However, since expert systems have become such an important spin-off of AI research, and because they demonstrate how the concepts developed through basic AI research have been applied, a great deal of attention will be devoted to them. In Part 2 we will also discuss some of the institutions involved in AI research and explain how ISI® is using AI to solve some of our bibliographic verification and classification problems. The whole problem of automatic classification and indexing is invariably an AI problem. But with day-to-day practical problems of delivering usable indexed information, it is not possible to wait while AI matures.

* * * * *

My thanks to Joan Lipinsky Cochran and Amy Stone for their help in the preparation of this essay.

REFERENCES

1. **Turing A M.** Computing machinery and intelligence. *Mind* 59:433-60, 1950.
2. **Samuel A L.** Some studies in machine learning using the game of checkers. II. Recent progress. *IBM J. Res. Develop.* 11:601-17, 1967.
3. **Shannon C E.** Programming a computer for playing chess. *Phil. Mag.* 41(Ser. 7):256-75, 1950.
4. **Gelernter H, Hansen J R & Loveland D W.** Empirical explorations of the geometry-theorem proving machine. (Feigenbaum E A & Feldman J, eds.) *Computers and thought.* New York: McGraw-Hill, 1963. p. 153-63.
5. **Fikes R E & Nilsson N J.** STRIPS: a new approach to the application of theorem proving to problem solving. *Artif. Intell.* 2:189-208, 1971.
6. **Colby K M, Weber S & Hilf F D.** Artificial paranoia. *Artif. Intell.* 2:1-25, 1971.
7. **Lederberg J & Feigenbaum E A.** Mechanization of inductive inference in organic chemistry. (Kleinmuntz B, ed.) *Formal representation of human judgment.* New York: Wiley, 1968. p. 187-218.
8. **Lindsay R K, Buchanan B G, Feigenbaum E A & Lederberg J.** *Applications of artificial intelligence for chemical inference: the DENDRAL project.* New York: McGraw-Hill, 1980. 256 p.
9. **Barr A & Feigenbaum E A**, eds. *The handbook of artificial intelligence.* Los Altos, CA: William Kaufmann, 1981. Vol. I. 409 p.
10. **Inhelder B & Piaget J.** *The growth of logical thinking.* New York: Basic, 1958. 356 p.
11. **Alexander T.** Teaching computers the art of reason. *Fortune* 105(10):82-92, 1982.
12. **Kolata G.** How can computers get common sense? *Science* 217:1237-8, 1982.

13. **Garfield E.** Can machines be scientific translators? *Essays of an information scientist.* Philadelphia: ISI Press, 1981. Vol. 4. p. 574-8. (Reprinted from: *Current Contents* (33):5-9, 18 August 1980.)
14. **Kouwenhoven J A.** The trouble with translation. *Harper's* 225(1347):38-44, 1962.
15. **Bar-Hillel Y.** The present status of automatic translation of languages. (Alt F L, ed.) *Advances in computers.* New York: Academic Press, 1960. Vol. 1. p. 91-163.
16. **Chomsky N.** *Language and mind.* New York: Harcourt Brace Jovanovich, 1972. 194 p.
17. **Amarel S.** On representations of problems of reasoning about actions. (Michie D, ed.) *Machine intelligence 3.* New York: American Elsevier, 1968. p. 131-71.
18. **Roberts S K.** Artificial intelligence. *BYTE* 6(9):164-78, 1981.
19. **Weizenbaum J.** ELIZA—a computer program for the study of natural language communication between man and machine. *Commun. ACM* 9:36-45, 1966.
20. **Quillian M R.** Semantic memory. (Minsky M, ed.) *Semantic information processing.* Cambridge, MA: MIT Press, 1968. p. 216-70.
21. **Kochen M.** Adaptive mechanisms in digital "concept" processing. *Discrete adaptive processes. Joint Automatic Control Conference,* 29 June 1962, New York. New York: American Institute of Electrical Engineers, 1962. p. 50-8.
22. **Findler N V,** ed. *Associative networks: representation and use of knowledge by computers.* New York: Academic Press, 1979. 462 p.
23. **Schank R C.** Conceptual dependency: a theory of natural language understanding. *Cog. Psychol.* 3:552-631, 1972.
24. **Waltz D L.** Artificial intelligence. *Sci. Amer.* 247(4):118-33, 1982.
25. **Minsky M.** A framework for representing knowledge. (Winston P H, ed.) *The psychology of computer vision.* New York: McGraw-Hill, 1975. p. 211-77.
26. **Schank R C & Abelson R P.** Scripts, plans and knowledge. *Advance papers of the Fourth International Joint Conference on Artificial Intelligence,* 3-8 September 1975, Tbilisi, Georgia, USSR. Cambridge, MA: Artificial Intelligence Laboratory, MIT, 1975. Vol. 1. p. 151-7.
27. **Hilts P J.** Interview: John McCarthy. *OMNI* 5(7):101-6; 122, 1983.
28. **McCarthy J.** Circumscription—a form of non-monotonic reasoning. *Artif. Intell.* 13:27-39, 1980.
29. **Newell A & Simon H A.** The logic theory machine: a complex information processing system. *IEEE Trans. Inform. Theory* IT-2(3):61-79, 1956.
30. **Newell A, Shaw J C & Simon H A.** Report on a general problem-solving program. *Information processing. Proceedings of the International Conference on Information Processing,* 15-20 June 1959, Paris, France. Paris: UNESCO, 1960. p. 256-64.
31. **McCorduck P.** *Machines who think.* San Francisco: W.H. Freeman, 1979. 375 p.
32. **Herman R.** Computers on the road to intelligence. *New Sci.* 95:358-61, 1982.
33. **Feigenbaum E A & McCorduck P.** *The fifth generation.* Reading, MA: Addison-Wesley, 1983. 275 p.
34. **Sussman G J.** *A computer model of skill acquisition.* New York: American Elsevier, 1975. 133 p.
35. **Winston P H.** Learning structural descriptions from examples. (Winston P H, ed.) *The psychology of computer vision.* New York: McGraw-Hill, 1975. p. 157-209.
36. **Winograd T.** *Understanding natural language.* New York: Academic Press, 1972. 195 p.
37. **Newell A & Shaw J C.** Programming the logic theory machine. *Proceedings of the Western Joint Computer Conference,* 26-28 February 1957, Los Angeles, CA. New York: Institute of Radio Engineers, 1957. p. 230-40.
38. **McCarthy J.** History of LISP. *ACM SIGPLAN Notices* 13:217-23, 1978.
39. **Alexander T.** Computers on the road to self-improvement. *Fortune* 105(12):148-60, 1982.
40. **Bernstein J.** Profiles: Marvin Minsky. *New Yorker* 57(43):50-126, 1981.