

February 1970

A SURVEY OF THE LITERATURE ON  
PROBLEM-SOLVING METHODS IN ARTIFICIAL INTELLIGENCE

by

Nils J. Nilsson

Artificial Intelligence Group

Technical Note 21

SRI Project 7024

This research was supported by the Office of Naval Research,  
Washington, D.C., under Contract N0014-68-C-266.

## ABSTRACT

Problem-solving methods using some sort of heuristically guided search process have been the subject of much research in Artificial Intelligence. This paper groups these problem-solving methods under three major headings: the State-Space Approach, the Problem-Reduction Approach, and the Formal-Logic Approach. Using this framework, a survey is presented of most of the important articles and books dealing with problem solving, game playing, and automatic theorem proving. The bibliography contains over 150 references.

## I INTRODUCTION

### A. Overviews, Surveys, and Source Materials

Many human activities, such as solving puzzles, playing games, doing mathematics, and even driving an automobile, are said to demand "intelligence." If computers could perform such tasks as these, then these computers (together with their programs) presumably would possess some degree of "artificial intelligence."

The question of whether or not machines can (or ever will be able to) "think" still provokes lively debate, even among those who concede that man is a machine. Turing (1950)<sup>\*</sup> disposes of many of the standard arguments against intelligent machines. To decide whether or not a machine is intelligent, he proposes what has come to be called the "Turing Test."

Selfridge and Kelly (1962) debate the magnitude of the practical problems in creating intelligent machines after agreeing that there are no known theoretical barriers. Hubert Dreyfus (1965) thinks that digital computers are inherently incapable of such "necessities" of intelligence as "fringe consciousness" and "perspicuous grouping." His arguments are systematically refuted by Papert (1968).

Attempts to "organize" the field of Artificial Intelligence have never been wholly successful. The subtopics of search, pattern recognition,

---

\*References are listed alphabetically at the end of this survey paper.

learning, problem solving, and induction have been suggested by Minsky (1961a) in an important survey article. This breakdown is still useful, even though its logical validity is challenged by the plethora of statements in the literature that have the form: "The problem of X is basically a problem of Y," where X and Y could be any pair of subtopics. More recently Minsky (1968) has written another thoughtful article on the foundations of Artificial Intelligence and concludes that a major problem is that of acquiring, maintaining, and accessing a large knowledge base. In this survey we shall be concerned almost exclusively with the literature on problem solving.

Artificial Intelligence is a much-surveyed field, and there are many bibliographies. An early annotated bibliography was compiled by Minsky (1961b). Later surveys by Feigenbaum (1963) and Solomonoff (1966) include many additional references. A recent survey by Feigenbaum (1968) lists more articles and also speculates about the future of the field.

A volume that is often referenced because it contains many of the early papers is entitled Computers and Thought, edited by Feigenbaum and Feldman. D. Michie and others have edited a series of books called Machine Intelligence. These contain papers delivered at the Machine Intelligence Workshops held annually in Edinburgh. Another important book is called Semantic Information Processing edited by Minsky; it contains complete versions of several Ph.D. dissertations dealing with language processing and "understanding."

Artificial Intelligence is a journal devoted exclusively to A. I.; it will begin publishing in 1970. Also, the Journal of the Association for Computing Machinery occasionally publishes articles on Artificial Intelligence subjects.

In the United States, Artificial Intelligence activities are coordinated through a Special Interest Group on Artificial Intelligence (SIGART) of the Association for Computing Machinery (ACM). SIGART publishes a newsletter that occasionally contains reference material not published elsewhere. The Artificial Intelligence and Simulation of Behavior (AISB) group of the British Computer Society publishes a European newsletter.

For completeness, we shall occasionally reference unpublished memoranda and reports in this survey. The authors of such material will sometimes provide copies upon request.

Problem-solving programs demand a heavy diet of puzzles and games on which to sharpen and refine techniques. Some good general books of puzzles are those of Martin Gardner (1959, 1961) who edits a puzzle column in The Scientific American. Also see the books of puzzles by Dudeney (1958, 1967) who was a famous British puzzle inventor.

#### B. Approaches to Artificial Intelligence

In attempting to build intelligent machines, one naturally asks "What is the secret of animal intelligence?" People have had a variety of adventures pursuing this question but no one has yet found the "secret."

Rosenblatt (1962) suggested brain models called perceptrons; these were networks of "artificial neurons" based on the neuron models of McCulloch and Pitts (1943). The study of perceptrons stimulated early pattern-recognition research and led to some elegant mathematical results on computational geometry by Minsky and Papert (1969). The complex processes of "intelligence," however, were beyond the power of these simple perceptron models.

Another biologically based strategy was the rather grandiose attempt to simulate evolution itself. Since evolution produced intelligent man in two billion years or so, let us simulate the processes of evolution at high speed in the computer. Fogel, et al. (1966) describe experiments involving the production of many generations of finite-state machines using the strategies of mutation and selective survival. Although such a technique may be capable of condensing the first few million years of evolution to a few days of computer time, it seems that the important middle and later stages of evolution involve structures already so complex (though not yet "intelligent") that their evolution cannot be speeded up by computer simulation. Thus the "artificial evolution" approach did not succeed in producing adequately complex machines either.

Another way to learn about intelligence from animals is to study their behavior, particularly the problem-solving behavior of man. Travis (1963, 1967) discusses the role of introspection in the design of problem solvers.

Newell, Shaw, and Simon (1959) describe a "General Problem Solver" that is supposed to attack problems in much the same way that humans do.

A rich source of ideas about how humans attack problems is found in Polya (1957). Polya quotes the Greek mathematician Pappus who, around 300 A.D., gave an excellent description of the "problem-reduction approach:"

"In analysis, we start from what is required, we take it for granted, and we draw consequences from it, and consequences from the consequences, till we reach a point that we can use as a starting point in synthesis. For in analysis we assume what is required to be done as already done (what is sought as already found, what we have to prove as true). We inquire from what antecedent the desired result could be derived; then we inquire again what could be the antecedent of that antecedent, and so on, until passing from antecedent to antecedent, we come eventually upon something already known or admittedly true. This procedure we call analysis, or solution backwards, or regressive reasoning."\*

In the problem-solving methods based on analysis of human behavior, we find that trial-and-error search at some level plays a key role.

---

\* From p. 142 of Polya (1957).

Campbell (1960) calls the unguided search process a "blind-variation-and-selective-survival process." He concludes:

- (1) A blind-variation-and-selective-survival process is fundamental to all inductive achievements, to all genuine increases in knowledge, to all increases in fit of system to environment.
- (2) The processes that shortcut the full blind-variation-and-selective-survival process are in themselves inductive achievements containing wisdom about the environment achieved originally by a blind-variation-and-selective-survival process.
- (3) In addition, such substitute processes contain in their own operation a blind-variation-and-selective-survival process at some level.

We agree with Campbell about the ultimate primacy of search.

The real trick in designing an efficient automatic problem solver is to search at the highest level permitted by the available information about the problem and about how it might be solved. In this survey we are primarily concerned with the literature on efficient search techniques and how they are used in automatic problem-solving systems.

### C. Problem Solving Methods

There have been only a few attempts to study problem-solving processes abstractly in order to catalog the different methods and



to deduce general properties of these methods. In this survey we identify three major approaches to automatic problem solving, although our division would probably not meet with universal approval among researchers in this field. We shall give these approaches the following names:

- (1) The state-space approach,
- (2) The problem-reduction approach, and
- (3) The formal-logic approach.

A slightly different taxonomy of problem-solving methods is suggested by Newell (1969). Our classification scheme has been much influenced by a series of difficult but worthwhile papers by Amarel (1965, 1967, 1969). A paper by Sandewall (1969) formalizes some of the same problem-solving ideas treated in the present survey. A book by Banerji (1969) contains a highly formal treatment of problem solving and game playing.

In the state-space approach, the problem to be solved is identified with the problem of finding a path in a graph. The nodes of the graph represent "states" in a problem space. The initial node represents the beginning state of the problem, e.g., the starting configuration of a puzzle, an integral expression to be integrated, or the set of axioms from which a theorem is to be proved. The problem is solved when a sequence of state transformations is found that change the initial state into a goal state. Thus, each possible state transformation is represented by an arc connecting two nodes in the graph. Arcs represent, for example, the allowable moves

in a puzzle, algebraic manipulations in integration problems, or valid inferences in a theorem-proving problem. A problem solution is a path of arcs that terminates in a goal node. The goal node corresponds to a problem state satisfying the solution criteria, for example the desired end configuration of a puzzle, an integral in an integral table, or a set of logical expressions containing the theorem to be proved. The problem is "solved" by searching for a solution path in the state-space graph.

The state-space approach to problem solving gets its name from the use of "state spaces" for similar purposes in control theory. It is also used extensively in operations research. Some of the state space search methods to be cited later are identical to those called branch-and-bound methods in operations research. Lawler and Wood (1966) give a survey of branch-and-bound methods and their applications.

The problem-reduction approach involves techniques for reducing problems into sets of subproblems and these into sub-subproblems and so on until ultimately all of the resulting problems are trivial. Sometimes the reduction is such that the original problem is solved only by solving all of the subproblems; alternatively, there are reductions in which the original problem can be solved by solving any one of the subproblems. Search methods are also used in the problem-reduction approach, since one never knows at the outset which problem reductions will ultimately lead to solutions. Special graph structures, called AND/OR graphs, have been useful for representing problem-reduction solution methods.

Our predilection to distinguish between state-space and problem-reduction methods derives from the different search strategies used by each method. The distinction is precisely the same as that made by Amarel (1967) between "production-type" and "reduction-type" methods. Slagle (1963a) also finds the problem-reduction formulation useful in describing his program for symbolic integration. Although the General Problem Solver (GPS) of Newell and his co-workers can be described in terms of state-space methods, it is our opinion that its operation can be understood more clearly if it is described as a problem-reduction type problem solver. GPS is thoroughly discussed in a book by Ernst and Newell (1969).

The formal-logic approach to problem-solving systems stems from the "advice-taker" memoranda of McCarthy (1958, 1963). The advice taker was to be a system that used formal methods to deduce the solutions to problems from a large set of axioms representing the problem-solver's knowledge base. It could be given "advice" merely by adding new axioms. Some early work related to this idea was undertaken by Black (1964). We shall mention some of the more recent work in this field later in the survey.

Some excellent ideas on solving large combinatorial problems have been suggested by Shen Lin (1965, 1968). Although these ideas do not appear to fit neatly into any of our three categories of problem-solving methods, they deserve the attention of any serious student in the field.

#### D. Applications of Problem-Solving Programs

It is fair to ask whether or not any of these methods that work so well on simple problems such as puzzles and games have been usefully employed on "real" problems. State-space methods have been employed in the solution of operations research problems such as the well-known traveling salesman problem. The "best" exact solution technique so far is a state-space method proposed in a Ph.D. dissertation by Shapiro (1966) and discussed by Bellmore and Nemhauser (1968). Although the traveling salesman problem may appear as frivolous as do puzzles and games, it is a model for problems of economic importance in scheduling and production design.

Other applications of the state-space method have been made in the control of remote manipulators by Whitney (1969) and in sequential decoding by Jelinek (1969). Problem-reduction methods have been employed in a system that performs symbolic integration (Slagle 1963a) and in a system that analyzes mass spectrograph data (Buchanan, Sutherland, and Feigenbaum 1969).

## II THE STATE-SPACE APPROACH

### A. Formulating Problems for the State-Space Approach

#### 1. Elements of a State-Space Representation

There are three major elements of a state-space representation: the state descriptions, the operators that change a state description into another one, and the goal test. These have long been recognized as basic in automatic problem solving. These notions are discussed in the book on the General Problem Solving (GPS) program by Ernst and Newell (1969). A more abstract treatment of the problem of description of states and operators may be found in Amarel (1967, 1969).

The basic vocabulary of graph theory (arcs, nodes, paths, etc.,) is often used to describe problem-solving processes. Classic books on graph theory are those of Berge (1962) and Ore (1962). Ore (1963) has also written a popular, elementary book illustrating applications of graph theory to various combinatorial problems.

A state-space representation can also be given in terms of a "nondeterministic" program. The phrase "nondeterministic algorithm" was proposed by Floyd (1967a). In these algorithms, Floyd allowed the use of a "choice" function to simplify the description of exhaustive search strategies. Later Manna (1970) described a class of programs that allowed nondeterministic assignment statements (similar to the choice function) as well as nondeterministic branching operations. He described techniques

for proving the "correctness" of programs containing these new elements. (In an earlier paper, Manna (1969) considered the problem of proving the correctness of ordinary, nondeterministic programs.) Fikes (1970) describes a complete problem-solving system in which problems are represented in a procedural language that allows the use of nondeterministic choice functions.

## 2. Typical Problem Areas in Which State-Space Methods Have Been Applied

Many of the specialized problem-solving methods from a variety of fields can be viewed as applications of the state-space approach. The traveling salesman problem of Operations Research, syntax-analysis problems, distribution or flow problems, and control-theory problems have all been attacked by state-space methods.

For a review of methods for solving the traveling salesman problem, see Bellmore and Nemhauser (1968). Problems of syntax analysis are common in language processing. Feldman and Gries (1968) give a thorough survey of syntax-analysis techniques as used by translating systems for the formal languages of computation. Amarel (1965) discusses syntax analysis from the point of view of automatic problem solving and proposes a problem-reduction procedure. For a highly readable treatment of the use of symbol strings and production systems as models of computation see the final chapters in a book on computation by Minsky (1967). Many problems involving distribution,

flow, and queuing can be solved by state-space methods. A discussion of these can be found in a book by Ford and Fulkerson (1962). Control theory is a large and specialized field with a variety of problem-solution methods. A book by De Russo, Roy, and Close (1965) is a good introduction to modern control theory.

### 3. The Problem of Finding a "Good" Representation

The problem of finding "good" representations for problems has been treated by only a few researchers, notably by Amarel. Amarel (1968) is a classic paper on the subject; it takes the reader through a series of progressively better representations for the "missionaries and cannibals" problem.

A powerful representational technique involves the use of variables in state descriptions. Later versions of GPS allowed the use of "object schemas" with variables as did Fikes' (1970) problem-solving system. A paper by Newell (1965) examines some possible approaches (and their limitations) toward making progress on "the representation problem."

## B. State-Space Search Techniques

### 1. Shortest-Path Algorithms

Efficient methods for finding the shortest (or least costly) path between two nodes in a graph are of great interest in a variety of disciplines. Efficient breadth-first search methods have been described by Dijkstra (1959) and by Moore (1959). Also, the dynamic programming

algorithms of Bellman are essentially breadth-first search methods. For a thorough discussion of dynamic programming see the book by Bellman and Dreyfus (1962). A depth-first search procedure, often called "backtrack programming" in computer science, is described by Golomb and Baumert (1965). Stuart Dreyfus (1969) presents a detailed survey of some of these and other graph-searching methods.

## 2. Heuristic-Search Techniques

The use of special heuristic information to increase search efficiency has been studied both in Artificial Intelligence and in Operations Research. This information is often incorporated in an "evaluation function." The evaluation function is used to help direct the search for a path to a goal node as discussed by Doran and Michie (1966), for example. A general theory of the use of evaluation functions to guide search was presented in a paper by Hart, Nilsson, and Raphael (1968).

In the "branch-and-bound" methods of Operations Research we also see the use of evaluation functions to guide search. For a description of these, see the survey article by Lawler and Wood (1966). The branch-and-bound technique proposed by Shapiro (1966) for the traveling salesman problem can also be interpreted as a direct application of state-space search methods using an evaluation function.

For further theoretical analyses of heuristic graph searching see Pohl (1969, 1970). Pohl (1969) considers also the problem of searching



outward from both start and goal nodes. Particularly interesting here is his thorough discussion of the more complex termination criterion needed for "bidirectional search."

Doran and Michie (1966) proposed a performance measure called penetrance for judging the efficiency of a given search. Slagle and Dixon (1969) propose another measure, which they call the "depth ratio."

### III THE PROBLEM REDUCTION APPROACH

#### A. Formulating Problems for the Problem-Reduction Approach

##### 1. Some Example Problem-Reduction Formulations

Some of the very earliest work in Artificial Intelligence used the problem-reduction technique (also called "reasoning backward") to solve problems. Newell, Shaw, and Simon (1957) programmed a Logic Theory Machine (LT) that proved theorems in the propositional calculus by working backwards from the theorem to be proved. In the LT program, use was made of a tree of subproblems to keep track of alternative chains of reasoning.

Another early program in which problem-reduction methods and subproblem trees were used was the Symbolic Integration (SAINT) program of Slagle (1961, 1963a). Slagle first called these trees AND/OR trees; in a later paper, Slagle and Bursky (1968) used the term Proposition Tree. A much more elaborate integration system (called SIN) was later programmed by Moses (1967). Moses' system embodied so many special criteria for applying the various operators that most integrations are carried out with little or no search. One might speculate that most of the search effort that might have been needed to perform integrations was carried out once and for all by Moses himself in designing the program. The results of this design search were the special rules about which operators to apply in all cases.

A geometry theorem-proving system using problem-reduction methods was programmed by Gelernter, et al. (1959, 1960). This program was never

really completed; nevertheless, several of its features and proposed features represented important and original innovations. The General Problem Solver (GPS) of Newell and his co-workers (see Ernst and Newell (1969)) can also be viewed as an application of the problem-reduction method. Ernst (1969) inquired about conditions under which GPS is guaranteed to find a solution.

Amarel (1967) used special graph structures similar to AND/OR trees in discussing problem-reduction methods. Manna (1970) gives several examples of representing an implicit AND/OR tree by nondeterministic programs.

## 2. Games and the Problem-Reduction Approach

Slagle and his co-workers have stressed the essential similarities between AND/OR trees and game trees (Slagle, 1970, Slagle and Bursky, 1968). This similarity is particularly apparent in simple games that can be searched to termination or in the "end games" of more complex games such as chess. Indeed, one often thinks of chess end-game puzzles as problems of proof rather than as games.

## B. Problem-Reduction Search Techniques

### 1. Development of AND/OR Graph-Searching Techniques

The search strategies of many of the early programs that generated subproblem trees employed only rather simple node-ordering methods. Early versions of GPS used a depth-first strategy and a means for measuring problem difficulty; backtracking occurred when a successor problem was

judged more difficult than any of its ancestors. Slagle's SAINT program used depth of function nesting as a measure of problem difficulty and generally worked on the easiest problems first.

Slagle and his co-workers experimented with many search strategies for game trees and AND/OR trees during the 1960s. The most complex of these involved a "dynamic ordering" of the nodes. In their problem-solving system called MULTIPLE (MULTIpurpose Program that LEarns), Slagle and Bursky (1968) incorporated a general strategy for searching AND/OR trees. This strategy uses the notion of the "probability that a proposition is true" and then defines a "merit function" over the open nodes of the tree. That node having greatest effect on the probability that the original proposition is true is said to have the largest "merit"; the system always attempted to solve that subproblem corresponding to the node with the largest merit.

Amarel (1967) proposed an "attention control" strategy for ordering the nodes in an AND/OR tree. This strategy attempted to find a "minimal cost" solution. The present author (Nilsson, 1968) also suggested a cost-minimizing strategy and later realized that it was essentially identical to Amarel's. In Nilsson (1968) a proof is given that the strategy does indeed find minimal-cost solutions; this proof is based on a similar one for state-space graphs by Hart, Nilsson, and Raphael (1968). The Amarel-Nilsson strategy differs little from the dynamic-ordering method of Slagle and Dixon.

## 2. Development of Game Tree Searching Techniques

Claude Shannon (1950) discussed some of the problems inherent in programming a machine to play a complex game such as chess. He suggested a minimax-search procedure to be used in conjunction with a static-evaluation function. Newell, Shaw, and Simon (1958) used several of these ideas in constructing an early chess-playing program; they also give an excellent discussion of this area of research. Additional discussion on chess-playing programs, together with a "Five-Year Plan" for automatic chess, can be found in an article by Good (1968), which also lists several references.

Later, Samuel (1959) described a checker-playing program that incorporated polynomial-evaluation functions, minimax-search methods, and various "learning" strategies for improving play. Samuel's program plays an excellent checker game and beats all but the very best players. It continues to be one of the outstanding examples of the application of Artificial Intelligence techniques. Later work on this program is described in Samuel (1967). One of the features of more recent versions of Samuel's program is a dynamic-ordering search procedure somewhat similar to that of Slagle and Dixon (1969).

The  $\alpha$ - $\beta$  procedure is an elaboration of the minimaxing technique that prevents search effort from being wasted on futile paths. It was "discovered" independently by many workers. It is first described by

Newell, Shaw, and Simon (1958)\* and was the subject of much investigation by McCarthy and his students at MIT (Edwards and Hart, 1963). Few clear expositions of the method and its properties exist. Samuel's second checkers paper (Samuel, 1967) contains a good description, as does the paper by Slagle and Dixon (1969). Some results on the search efficiency of the  $\alpha$ - $\beta$  procedure were first stated by Edwards and Hart (1963) based on a theorem that they attribute to Michael Levin. Later Slagle and Dixon (1969) give what they consider to be the first published proof of this theorem. Slagle and Dixon discuss several variations of the  $\alpha$ - $\beta$  procedure culminating with one employing dynamic ordering. The performance of these various strategies is compared using the ancient game of Kalah.

Papers by Slagle (1963b) and Slagle and Dixon (1970) describe an "M and N Tree-Searching Program" that adds (or subtracts) a bonus when backing up minimax scores. This bonus acknowledges the value of having more than one good alternative move and thus results in some improvement in play.

### 3. Some Representative Game-Playing Programs

#### Chess

Kister, et al. (1957) describe the earliest chess system programmed on a computer (MANIAC I at Los Alamos). It used a reduced board (6 x 6) and played rather poorly.

---

\* Samuel stated (personal communication) that his early checkers program also used the  $\alpha$ - $\beta$  procedure but that at the time he thought its use too straightforward to merit discussion in the paper.

Bernstein, et al. (1958) describe a chess system programmed at IBM. It also played rather poorly, but on a full (8 x 8) board.

Newell, Shaw, and Simon (1958) present another early chess program under development at Carnegie.

Kotok (1962) wrote an early MIT program later taken to Stanford by John McCarthy and modified slightly. This one achieved the level of mediocre play.

Adelson-Velskii, et al.; (no paper available) wrote a program at the Institute for Theoretical and Applied Physics in Moscow. This program beat the Kotok-McCarthy program in a tournament. [see SIGART Newsletter, No. 4, p. 11 (June 1967)].

Greenblatt, et al. (1967) describe an MIT program now called Mac Hack. Its level of play can be described as "middle amateur." It is an honorary member of the Massachusetts Chess Society and has been given a Class C rating. For some example games see the following SIGART Newsletters: No. 6, p. 8 (October 1967) [here, the computer beat H. Dreyfus who earlier doubted that a machine could beat even an amateur player]; No. 9, pp. 9-10 (April 1968) No. 15, pp. 8-10 (April 1969) and No. 16, pp. 9-11 (June 1969).

#### Checkers

Samuel (1959, 1967) continues to improve a program that plays excellent checkers but cannot quite beat world champions.

### Kalah

Russell (1964) wrote an early Kalah program.

Slagle and Dixon (1969) describe experiments using the game of Kalah.

Slagle and Dixon (1970) discuss more experiments using Kalah to test the "M & N" procedure.

(The Kalah programs are probably unbeatable by human players)

### Go

Zobrist (1969) has written a program to play this ancient and difficult game. It plays rather poorly by human standards and does no tree searching. For an example of how Zobrist's program performs, see SIGART Newsletter No. 18, pp. 20-22, October 1969.



#### IV THE FORMAL LOGIC APPROACH

##### A. Theorem Proving in the Predicate Calculus

The formal-logic approach to automatic problem solving requires the ability to make deductions in a formal system such as mathematical logic. Two excellent textbooks on logic are those of Mendelson (1964) and Robbin (1969). For the specialist there is the classic by Alonzo Church (1956).

The resolution principle of J. A. Robinson (1965a) has permitted the development of some highly efficient automatic theorem-proving systems for the first-order predicate calculus. The resolution principle is based on the "semantic-tableaux" proof procedure of Herbrand (1930). Herbrand's procedure could (in principle) be implemented by the construction of "semantic trees" as discussed by Kowalski and Hayes (1969) and J. A. Robinson (1968b). The straightforward use of this construction would be grossly inefficient, however. The use of the resolution-inference rule permits a substantial increase in efficiency. The completeness and soundness of resolution can be easily justified in terms of the semantic-tree construction, while at the same time the use of resolution eliminates the need to construct semantic trees.

Resolution is a rule of inference that combines modus ponens, substitution, and other syllogisms. In order to apply resolution, well-formed formulas are first converted into quantifier-free, conjunctive-normal form following a procedure outlined by Davis and Putnam (1960).

A clearly written, concise review of the resolution principle with proofs of its soundness and completeness can be found in a paper by Luckham (1967). J. A. Robinson's initial paper (Robinson, 1965a) also contains proofs of the soundness and completeness of resolution. J. A. Robinson (1970) has also written an excellent survey entitled "The Present State of Mechanical Theorem Proving."

Complex, general problem solvers based on the formal-logic approach will probably also need facilities for higher-order logics and special mechanisms to handle the equality relation. For a discussion of the application of higher-order logics to problem solving, see McCarthy and Hayes (1969). Robbin (1969) contains a section on second-order logic, and some papers by J. A. Robinson (1968a, 1969) discuss the general problem of proof procedures for higher-order logics.

It is still not clear how the equality relation (and other standard, ubiquitous relations) ought to be "built in" to automatic theorem provers. One scheme for building the equality relation into a resolution theorem prover is discussed by Robinson and Wos (1969).

## B. Formulating Problems for the Formal-Logic Approach

### 1. Development of Formal-Logic Methods

We have already mentioned that work on techniques for solving problems using formal-logic methods was stimulated by the "advice taker" memoranda of McCarthy (1958, 1963). Work toward implementing such a system

was undertaken by Black (1964). Cordell Green was the first to develop a formal-logic problem solver using a complete inference system (resolution) for first-order logic. Much of this work is described in his dissertation (Green, 1969a) and in two papers (Green, 1969b, 1969c).

Professor McCarthy continued his investigations on the requirements for a general, formal problem-solving system. He was particularly concerned with the necessity for including higher (than first) - order logic features in order to formalize such concepts as situations, future operators, actions, strategies, results of strategies, and knowledge. An excellent discussion of these ideas is contained in the paper by McCarthy and Hayes (1969).

## 2. Question-Answering Systems

One of the tasks that could be given to a formal problem-solving system is that of "question answering." This task involves a sophisticated type of information retrieval in which the answers to queries require that logical deductions be made from various facts stored in the data base. The design of question-answering systems also raises the question of how to translate between a natural language, such as English, and a formal language, used by the deductive system, such as the predicate calculus.

An early, general question-answering system was developed by Raphael (1964a, 1964b). Raphael concentrated on the deductive and

associative mechanisms needed and largely ignored the issue of natural-language translation. On the other hand, Bobrow (1964a, 1964b) developed a system for solving simple algebra problems stated in English. His system could translate these into the appropriate equations to be solved. Another general question-answering system called DEDUCOM (without English-logic translation abilities) was developed by Slagle (1965). Green and Raphael (1968) collaborated in the development of a question-answering system using resolution and first-order logic. Coles (1968) has developed a limited English-logic translation program that has been added to this question-answering system.

Two good surveys of work on natural-language question-answering systems have been written by Simmons (1965, 1970).

### 3. Example Applications of the Formal-Logic Method

Green (1969a) discusses several example formulations for problem solving using the formal-logic method. One of these is the problem of automatic program writing. The problem of writing computer programs is related to that of proving them correct. There is a substantial body of work on this latter subject including papers by McCarthy (1962), Floyd (1967b), and Manna (1969). London (1969) gives a good survey of work on proving the correctness of programs. A somewhat different (but still resolution-based) procedure for automatic program writing is described by Waldinger and Lee (1969).

Applications of theorem-proving techniques to automatic problem solving are also discussed by Green (1969a,b,c). In these, Green makes use of a special "state variable" that is added to each predicate.

One of the most obvious applications of automatic theorem provers is proving mathematical theorems. This application has been pursued by Robinson and Wos (1969) and by Guard, et al. (1969). In particular, Guard's program (with some human aid) has succeeded in finding the first proof for a conjecture in modular lattice theory.

### C. Formal-Logic Search Methods

An excellent discussion of the problem of developing heuristically effective search strategies for resolution theorem provers while maintaining logical completeness is contained in a paper by J. A. Robinson (1967). G. A. Robinson, et al. (1964) give a very clear exposition of several strategies with many examples.

There are two basic families of strategies for improving the efficiency of resolution theorem provers. One, which might be called refinement strategies, attempts to shorten the search by eliminating all but certain types of inferences. The other, which might be called ordering strategies, attempts to shorten the search by performing those resolutions first that are most likely to lead to a proof.

The ancestry-filtered (AF) form strategy is the simplest of a group of related refinement strategies. It restricts the allowable resolutions

to those that could lead to a proof tree having a particularly simple form. The AF-form strategy is discussed by Luckham (1969).

Several elaborations of the AF-form strategy are possible. Some of these combine the AF-form strategy with a strategy proposed by Andrews (1968) involving merges. Among the papers proving the completeness of AF form with merging are Kieburz and Luckham (1970), Yates, Raphael, and Hart (1970), and Anderson and Bledsoe (1970). The first of these contains other results about the properties of AF-form proofs, while the last two use rather novel methods of proving completeness that are of independent interest. Anderson and Bledsoe use their method to establish the completeness of other strategies as well. A paper by Loveland (1968) establishes the completeness of another restriction on the AF-form strategy.

Other refinement strategies use the idea of a model to restrict resolutions. Of these we might mention the  $P_1$ -deduction strategy of J. A. Robinson (1965b) and its generalization by Slagle (1967). An easier-understood, special case of Slagle's result has been proposed by Luckham (1969). Meltzer (1966, 1968) provides some additional results about  $P_1$ -type deductions.

The well known set-of-support strategy (Wos, et al., 1965), can be viewed as a special case of either the AF-form strategy or of the model strategies.

An inference-ordering strategy called the unit preference strategy was proposed and justified in a paper by Wos, et al. (1964). It and the set-of-support strategy have been incorporated as basic parts of a number of automatic theorem provers.

Most of the search strategies developed to date involve "syntactic" rather than "semantic" rules (that is, search restrictions and orderings were based on the form of clauses and possible deductions rather than on their meaning). Semantic guidance could be provided in a number of ways, but there have not yet been many attempts in this direction. One might ask whether it would be possible to use an "evaluation function" over pairs of clauses that are candidates to be resolved. Presumably this evaluation function could be influenced by the available semantic information as well as by the forms of the candidate clauses. Some theoretical results about the properties of strategies using evaluation functions are contained in a paper and dissertation by Kowalski (1970a, 1970b). Kowalski's results on searching "inference-graph" structures parallel those of Hart, et al. (1968) for state-space graph structures.

Several automatic theorem-proving programs have now been written. As examples we might mention those of Robinson and Wos (1969), Allen and Luckham (1970), Guard, et al. (1969), and the Green-Raphael-Yates system recently modified by Garvey and Kling (1969).

## REFERENCES

- Allen, J., and D. Luckham (1970),  
"An Interactive Theorem-Proving Program," in Machine Intelligence,  
Vol. 5, (to appear).
- Anderson, R., and W. Bledsoe (1970),  
"A Linear Format for Resolution with Merging and a New Technique for  
Establishing Completeness," (to appear in J. ACM).
- Andrews, P. (1968),  
"Resolution with Merging," J. ACM, Vol. 15, No. 3, pp. 367-381 (July  
1968). [Also see corrections in J. ACM, Vol. 15, No. 4, p. 720  
(October 1968)].
- Amarel, S. (1965),  
"Problem Solving Procedures for Efficient Syntactic Analysis" ACM  
20th National Conference (1965) [Also printed as Scientific Report  
No. 1, AFOSR Contract No. AF49(638)-1184 (1968)].
- Amarel, S. (1967),  
"An Approach to Heuristic Problem-Solving and Theorem Proving in the  
Propositional Calculus," Systems and Computer Science, J. Hart and  
S. Takasu, eds. (University of Toronto Press, 1967).
- Amarel, S. (1968),  
"On Representations of Problems of Reasoning About Actions," in  
Machine Intelligence, Vol. 3, D. Michie, ed., pp. 131-171 (American  
Elsevier Publishing Company, Inc., New York, New York 1968).
- Amarel, S. (1969),  
"On the Representation of Problems and Goal Directed Procedures for  
Computers," Comm. Amer. Soc. Cybernetics, Vol. 1, No. 2 (1969).
- Banerji, R. (1969),  
Theory of Problem Solving: An Approach to Artificial Intelligence  
(American Elsevier Publishing Co., Inc., New York, New York 1969).
- Bellman, R., and S. Dreyfus (1962),  
Applied Dynamic Programming (Princeton University Press, Princeton,  
New Jersey 1962).



- Bellmore, M., and G. Nemhauser (1968),  
 "The Traveling Salesman Problem: A Survey," Operations Research,  
 Vol. 16, No. 3, pp. 538-558 (May-June 1968).
- Berge, C. (1962),  
The Theory of Graphs and its Applications, (Translated by Alison  
 Doig) (John Wiley & Sons, Inc., New York, New York, 1962) [First  
 published in France by Dunod, Paris (1958)].
- Bernstein, A., et al. (1958),  
 "A Chess-Playing Program for the IBM 704 Computer," Proc. Western  
 Joint Computer Conference, pp. 157-159 (1958).
- Black, F. (1964),  
 "A Deductive Question-Answering System," Ph.D. Dissertation, Harvard  
 University, Cambridge, Massachusetts (June 1964). [Reprinted in  
Semantic Information Processing, pp. 354-402, M. Minsky, ed.  
 (MIT Press, Cambridge, Massachusetts (1968).]
- Bobrow, D. (1964a),  
 "Natural Language Input for a Computer Problem-Solving System," Ph.D.  
 Dissertation, MIT, Cambridge, Massachusetts (September 1964).  
 [Reprinted in Semantic Information Processing, M. Minsky, ed.  
 (MIT Press, Cambridge, Massachusetts (1968).]
- Bobrow, D. (1964b),  
 "A Question-Answering System for High-School Algebra Word Problems,"  
 in Proc. AFIPS Fall Joint Computer Conference, pp. 591-614 (1964).
- Buchanan, B., G. Sutherland, and E. Feigenbaum (1969),  
 "Heuristic Dendral: A Program for Generating Explanatory Hypotheses  
 in Organic Chemistry," in Machine Intelligence, Vol 4, B. Meltzer  
 and D. Michie, eds., pp. 209-254 (American Elsevier Publishing  
 Company, Inc., New York, New York, 1969).
- Collins, N., and D. Michie, eds. (1967)  
Machine Intelligence, Vol. 1 (American Elsevier Publishing Company,  
 Inc., New York, New York, 1967).
- Campbell, D. (1960),  
 "Blind Variation and Selective Survival as a General Strategy in  
 Knowledge-Processes," in Self-Organizing Systems, M. Yovits and S.  
 Cameron, eds., pp. 205-231 (Pergamon Press, New York, New York, 1960).

- Church, A. (1956),  
Introduction to Mathematical Logic (Princeton University Press,  
 Princeton, New Jersey, 1956).
- Coles, L.S. (1968),  
 "An On-Line Question-Answering System with Natural Language and  
 Pictorial Input," Proc. ACM 23rd Nat'l Conf., (Brandon Systems  
 Press, Princeton, New Jersey, pp. 157-167, 1968).
- Dale, E., and D. Michie, eds. (1968),  
Machine Intelligence, Vol. 2 (American Elsevier Publishing Company,  
 Inc., New York, New York, 1968).
- Davis, M., and H. Putnam (1960),  
 "A Computing Procedure for Quantification Theory," J. ACM, Vol. 7,  
 No. 3, pp. 201-215 (July 1960).
- De Russo, P., R. Roy, and C. Close (1965),  
State Variables for Engineers (John Wiley and Sons, Inc., New York,  
 New York, 1965).
- Dijkstra, E. (1959),  
 "A Note on Two Problems in Connection with Graphs," Numerische  
 Mathematik, Vol. 1, pp. 269-271 (1959),
- Doran, J., and D. Michie (1966),  
 "Experiments with the Graph Traverser Program," Proc. Royal Soc. A,  
 Vol. 294, pp. 235-259 (1966).
- Dreyfus, H. (1965),  
 "Alchemy and Artificial Intelligence," Rand Corporation Paper P3244,  
 Rand Corporation, Santa Monica, California, December 1965, AD-625 719.
- Dreyfus, S. (1969),  
 "An Appraisal of Some Shortest Path Algorithms," Operations Research,  
 Vol. 17, No. 3, pp. 395-412 (May-June 1969).
- Dudeney, H. (1958),  
The Canterbury Puzzles (Dover Publications, Inc., New York, New York,  
 1968). [Originally published in 1907.]
- Dudeney, H. (1967),  
536 Puzzles and Curious Problems, Martin Gardner, ed. (Charles Scribner's  
 Sons, New York, New York, 1967). [A collection from two of Dudeney's  
 books: Modern Puzzles (1926) and Puzzles and Curious Problems (1931)].

- Edwards, D., and T. Hart (1963),  
"The  $\alpha$ - $\beta$  Heuristic," Massachusetts Institute of Technology Artificial Intelligence Memo No. 30 (Revised) (October 28, 1963) [Originally printed as "The Tree Prune (TP) Algorithm," (December 4, 1961)].
- Ernst, G. (1969)  
"Sufficient Conditions for the Success of GPS," J. ACM, Vol. 16, No. 4, pp. 517-533 (October 1969).
- Ernst, G. and A. Newell (1969),  
GPS: A Case Study in Generality and Problem Solving ACM Monograph Series (Academic Press, New York, New York, 1969).
- Feigenbaum, E. (1963),  
"Artificial Intelligence Research," IEEE Trans. Info. Theory, Vol. IT-9, No. 4, pp. 248-261 (October 1963).
- Feigenbaum, E. (1968),  
"Artificial Intelligence: Themes in the Second Decade," Proc. IFIPS Congress, Edinburgh (August 1968). [Also printed as Stanford University Artificial Intelligence Project Memo No. 67 (August 15, 1968).]
- Feigenbaum, E., and J. Feldman, eds. (1963),  
Computers and Thought (McGraw-Hill Book Company, New York, New York, 1963).
- Feldman, J., and D. Gries (1968),  
"Translator Writing Systems," Comm. ACM, Vol. 11, No. 2, pp. 77-113 (February 1968).
- Fikes, R. (1970),  
"Ref-Arf: A System for Solving Problems Stated as Procedures," Artificial Intelligence, Vol. 1, No. 1 (1970).
- Floyd, R. (1967a)  
"Nondeterministic Algorithms," J. ACM, Vol. 14, No. 4, pp. 636-644, (October 1967).
- Floyd, R. (1967b)  
"Assigning Meanings to Programs," Proc. Symp. Appl. Math., Amer. Math. Soc., Vol. 19, pp. 19-32 (1967).

- Fogel, L., A. Owens, and M. Walsh (1966),  
Artificial Intelligence Through Simulated Evolution (John Wiley  
and Sons, Inc., New York, New York, 1966).
- Ford, L., Jr., and D. Fulkerson (1962),  
Flows in Networks (Princeton University Press, Princeton, New Jersey,  
1962).
- Gardner, M. (1959),  
The Scientific American Book of Mathematical Puzzles and Diversions  
(Simon and Schuster, New York, New York, 1959).
- Gardner, M. (1961),  
The Second Scientific American Book of Mathematical Puzzles and  
Diversions (Simon and Schuster, New York, New York, 1961).
- Garvey, T., and R. Kling (1969),  
"User's Guide to QA3.5 Question-Answering System," Stanford Research  
Institute Artificial Intelligence Group Technical Note 15, Stanford  
Research Institute, Menlo Park, California (December 1969).
- Gelernter, H. (1959)  
"Realization of a Geometry Theorem-Proving Machine," Proc. Int'l.  
Conf. Info. Proc., pp. 273-282 (UNESCO House, Paris, 1959). [Re-  
printed in Computers and Thought, E. Feigenbaum and J. Feldman, eds.,  
pp. 134-152 (McGraw-Hill Book Co., New York, New York, 1963).]
- Gelernter, H., J. Hansen, and D. Loveland (1960),  
"Empirical Explorations of the Geometry Theorem Proving Machine,"  
Proc. Western Joint Computer Conf., Vol. 17, pp. 143-147 (1960).  
[Reprinted in Computers and Thought, E. Feigenbaum and J. Feldman,  
eds., pp. 153-167 (McGraw-Hill Book Company, New York, New York
- Golomb, L., and L. Baumert (1965),  
"Backtrack Programming," J. ACM, Vol. 12, No. 4, pp. 516-524  
(October 1965).
- Good, I. (1968),  
"A Five-Year Plan for Automatic Chess," in Machine Intelligence,  
Vol. 2, E. Dale and D. Michie, eds., pp. 89-118 (American Elsevier  
Publishing Co., Inc., New York, New York, 1968).

- Green, C. (1969a),  
"The Application of Theorem-Proving to Question-Answering Systems,"  
Ph.D. dissertation, Electrical Engineering Department, Stanford  
University, Stanford, California (1969). [Also printed as Stanford  
Artificial Intelligence Project Memo AI-96 (June 1969).]
- Green, C. (1969b),  
"Theorem-Proving by Resolution as a Basis for Question-Answering  
Systems," in Machine Intelligence, Vol. 4, B. Meltzer and D. Michie,  
eds., pp. 183-205 (American Elsevier Publishing Company, Inc.,  
New York, New York, 1969).
- Green, C. (1969c),  
"Application of Theorem-Proving to Problem Solving," Proc. Int'l.  
Joint Conf. Artificial Intelligence, Washington, D.C. (May 1969).
- Green, C., and B. Raphael (1968),  
"The Use of Theorem-Proving Techniques in Question-Answering Systems,"  
Proc. 23rd Nat'l. Conf. ACM, pp. 169-181 (Brandon Systems Press,  
Princeton, New Jersey, 1968).
- Greenblatt, R., et al. (1967),  
"The Greenblatt Chess Program," Proc. AFIPS Fall Joint Computer  
Conference, Anaheim, California, pp. 801-810 (1967).
- Guard, J., et al. (1969),  
"Semi-Automated Mathematics," J. ACM, Vol. 16, No. 1, pp. 49-62,  
(January 1969).
- Hart, P., N. Nilsson, and B. Raphael (1968),  
"A Formal Basis for the Heuristic Determination of Minimum Cost  
Paths," IEEE Trans. Systems Sciences and Cybernetics, Vol. SSC-4,  
No. 2, pp. 100-107 (July 1968).
- Herbrand, J. (1930),  
"Recherches sur la théorie de la démonstration," Travaux de la Société  
des Sciences et des Lettres de Varsovie, Classe III Sciences  
Mathématiques et Physiques, No. 33 (1930).
- Jelinek, F. (1969),  
"Fast Sequential Decoding Algorithm Using a Stack," IBM J. Res. and  
Dev., Vol. 13, No. 6, pp. 675-685 (November 1969).

- Kieburztz, R., and D. Luckham (1970),  
"Compatibility of Refinements of the Resolution Principle," (to appear).
- Kister, J., et al. (1957),  
"Experiments in Chess," J. ACM, Vol. 4, No. 2, pp. 174-177  
(April 1957).
- Kotok, A. (1962),  
"A Chess Playing Program for the IBM 7090," unpublished B.S. Thesis,  
MIT, Cambridge, Massachusetts (1962).
- Kowalski, R. (1970a),  
"Search Strategies for Theorem-Proving" in Machine Intelligence,  
Vol. 5 (to appear).
- Kowalski, R. (1970b),  
"Studies in the Completeness and Efficiency of Theorem-Proving by  
Resolution," Ph.D. Thesis, University of Edinburgh (to appear).
- Kowalski, R., and P. Hayes (1969),  
"Semantic Trees in Automatic Theorem-Proving", in Machine  
Intelligence, Vol. 4, B. Meltzer and D. Michie, eds., pp. 87-101  
(American Elsevier Publishing Company, Inc., New York, New York,  
1969).
- Lawler, E., and D. Wood (1966),  
"Branch and Bound Methods: A Survey," Operations Research, Vol. 14,  
No. 4, pp. 699-719 (July-August 1966).
- Lin, Shen (1965),  
"Computer Solutions of the Traveling Salesman Problem,"  
Bell Sys. Tech. J., Vol. 44, No. 10 (December 1965).
- Lin, Shen (1968),  
"Heuristic Techniques for Solving Large Combinatorial Problems on a  
Computer," Formal Systems and Non-Numerical Problem-Solving by Computer,  
Case Western Reserve University, Cleveland, Ohio (November 1968).
- London, R. (1969),  
"Bibliography on Proving the Correctness of Computer Programs," Tech.  
Report No. 64, Computer Sciences Department, University of Wisconsin,  
Milwaukee, Wisconsin (June 1969).

- Loveland, D. (1968),  
 "A Linear Format for Resolution," Carnegie-Mellon University Computer Science Department Report (December 1968). [To appear in Proc. IRIA 1968 Symposium on Automatic Deduction.]
- Luckham, D. (1967),  
 "The Resolution Principle in Theorem-Proving," in Machine Intelligence, Vol. 1, N. Collins and D. Michie, eds., pp. 47-61 (American Elsevier Publishing Company, Inc., New York, New York, 1967).
- Luckham, D. (1969),  
 "Refinement Theorems in Resolution Theory," Stanford Artificial Intelligence Project Memo AI-81 (March 24, 1969). [Also to appear in Proc. IRIA 1968 Symposium on Automatic Deduction.]
- Manna, Z. (1969),  
 "The Correctness of Programs," J. Comp. Sys. Sci., Vol. 3 (May 1969).
- Manna, Z. (1970),  
 "The Correctness of Non-Deterministic Programs," Artificial Intelligence, Vol. 1, No. 1 (1970).
- McCarthy, J. (1958),  
 "Programs with Common Sense," in Mechanization of Thought Processes, Vol. 1, pp. 77-84, Proc. Symposium, National Physical Laboratory, London (November 24-27, 1958). [Reprinted in Semantic Information Processing, M. Minsky, ed., pp. 403-410 (MIT Press, Cambridge, Massachusetts, 1968).]
- McCarthy, J. (1962),  
 "Towards a Mathematical Science of Computation," Proc. IFIP Congress 62 (North Holland Publishing Company, Amsterdam, 1962).
- McCarthy, J., and P. Hayes (1969),  
 "Some Philosophical Problems from the Standpoint of Artificial Intelligence," in Machine Intelligence, Vol. 4, B. Meltzer and D. Michie, eds., pp. 463-502 (American Elsevier Publishing Company, Inc., New York, New York, 1969).
- McCarthy, J. (1963),  
 "Situations, Actions and Causal Laws," Stanford University Artificial Intelligence Project Memo No. 2 (1963). [Reprinted in Semantic Information Processing, M. Minsky, ed., pp. 410-418 (MIT Press, Cambridge, Massachusetts, 1968).]

- McCulloch, W. S., and W. Pitts (1943),  
"A Logical Calculus of the Ideas Immanent in Neural Nets," Bull. Math. Biophys., Vol. 5, pp. 115-137 (1943).
- Meltzer, B. (1966),  
"Theorem-Proving for Computers: Some Results on Resolution and Renaming," Comp. J., Vol. 8, pp. 341-343 (1966).
- Meltzer, B. (1968),  
"Some Notes on Resolution Strategies," in Machine Intelligence, Vol. 3, D. Michie, ed., pp. 71-75 (American Elsevier Publishing Co., Inc., New York, New York, 1968).
- Meltzer, B. and D. Michie, eds. (1969)  
Machine Intelligence, Vol. 4 (American Elsevier Publishing Company, Inc., New York, New York, 1969).
- Mendelson, E. (1964),  
Introduction to Mathematical Logic (D. Van Nostrand Company, Inc., Princeton, New Jersey, 1964).
- Michie, D., ed. (1968),  
Machine Intelligence, Vol. 3 (American Elsevier Publishing Company, Inc., New York, New York, 1968).
- Michie, D., ed. (1970),  
Machine Intelligence, Vol. 5 (American Elsevier Publishing Company, Inc., New York, New York, 1970).
- Minsky, M. (1961a),  
"Steps Toward Artificial Intelligence," Proc. IRE, Vol. 49, pp. 8-30 (January 1961). [Reprinted in Computers and Thought, E. Feigenbaum and J. Feldman, eds., pp. 406-450 (McGraw-Hill Book Co., New York, New York, 1963).]
- Minsky, M. (1961b),  
"A Selected Descriptor - Indexed Bibliography to the Literature on Artificial Intelligence," IRE Trans. Human Factors in Electronics, HFE-2, pp. 39-55 (March 1961). [A revision is reprinted in Computers and Thought, E. Feigenbaum and J. Feldman, eds., pp. 453-523 (McGraw-Hill Book Company, New York, New York, 1963).]



- Minsky, M. (1967),  
Computation: Finite and Infinite Machines (Prentice Hall, Inc.,  
 Englewood Cliffs, New Jersey, 1967).
- Minsky, M., and S. Papert (1969),  
Perceptrons: An Introduction to Computational Geometry (The MIT  
 Press, Cambridge, Massachusetts, 1969).
- Minsky, M., ed. (1968),  
Semantic Information Processing (The MIT Press, Cambridge,  
 Massachusetts, 1968).
- Moore, E. (1959),  
 "The Shortest Path Through a Maze," Proc. Int'l Symp. Theory of  
 Switching, Part II, April 2-5, 1957, in The Annals of the Computation  
 Laboratory of Harvard University, Vol. 30 (Harvard University Press,  
 Cambridge, Massachusetts, 1959).
- Moses, J. (1967),  
Symbolic Integration, Thesis, Massachusetts Institute of Technology,  
 Project MAC, Report MAC-TR-47 (December 1967).
- Newell A. (1965),  
 "Limitations of the Current Stock of Ideas About Problem Solving,"  
 in Electronic Information Handling, A. Kent and O. Taulbee, eds.  
 (Spartan Books, Washington D.C., 1965).
- Newell, A. (1969),  
 "Heuristic Programming: Ill-Structured Problems," Progress in  
 Operations Research, Vol. 3, J. Aronofsky, ed. (John Wiley and  
 Sons, Inc., New York, New York, 1969).
- Newell, A., J. Shaw, and H. Simon (1957),  
 "Empirical Explorations of the Logic Theory Machine," Proc. Western  
 Joint Computer Conference, pp. 218-239 (1957). [Reprinted in Computers  
 and Thought, E. Feigenbaum and J. Feldman, eds., pp. 109-133 (McGraw-  
 Hill Book Company, New York, New York, 1963).]
- Newell, A., J. Shaw, and H. Simon (1958),  
 "Chess Playing Programs and the Problem of Complexity," IBM J.  
 Research and Development, Vol. 2, pp. 320-335 (October 1958).  
 [Reprinted in Computers and Thought, E. Feigenbaum and J. Feldman,  
 eds., pp. 39-70 (McGraw-Hill Book Company, New York, New York, 1963).]

- Newell, A., J. Shaw, and H. Simon (1959),  
"Report on a General Problem-Solving Program," Proc. Int'l Conf. on Information Processing, pp. 256-264 (UNESCO House, Paris, 1969).
- Nilsson, N. (1968),  
"Searching Problem-Solving and Game-Playing Trees for Minimal Cost Solutions," Proc. IFIP Congress, Edinburgh (1968).
- Ore, O. (1962),  
Theory of Graphs, American Mathematical Society Colloquium Publication, Vol. 38, Providence, Rhode Island (1962).
- Ore, O. (1963),  
Graphs and Their Uses (Random House, New York, New York, 1963).
- Papert, S. (1968),  
"The Artificial Intelligence of Hubert L. Dreyfus. A Budget of Fallacies," Artificial Intelligence Memo No. 54, Massachusetts Institute of Technology, Cambridge, Massachusetts (January 1968).
- Pohl, I. (1969),  
"Bi-Directional and Heuristic Search in Path Problems," Ph.D. Dissertation, Computer Science Department, Stanford University, Stanford, California (1969). [Also printed as Stanford Linear Accelerator Report 104 (May 1969).]
- Pohl, I. (1970),  
"First Results on the Effect of Error in Heuristic Search," Machine Intelligence, Vol. 5 (to appear).
- Polya, G. (1957),  
How to Solve It, 2nd Edition (Doubleday and Company, Inc., Garden City, New York, 1957).
- Raphael, B. (1964a),  
"SIR: A Computer Program for Semantic Information Retrieval," Ph.D. Dissertation, Massachusetts Institute of Technology, Cambridge, Massachusetts (June 1964). [Reprinted in Semantic Information Processing, M. Minsky, ed. (MIT Press, Cambridge, Massachusetts, 1968).]

- Raphael, B. (1964b),  
"A Computer Program which 'Understands'," in Proc. AFIPS Fall Joint  
Computer Conference, pp. 577-589 (1964).
- Robbin, J. (1969),  
Mathematical Logic: A First Course (W. A. Benjamin, Inc., New York,  
New York, 1969).
- Robinson, J. A. (1965a),  
"A Machine-Oriented Logic Based on the Resolution Principle,"  
J. ACM, Vol. 12, No. 1, pp. 23-41 (January 1965).
- Robinson, J. A. (1965b),  
"Automatic Deduction with Hyper-Resolution," Int'l J. Comp. Math.,  
Vol. 1, pp. 227-234 (1965).
- Robinson, J. A. (1967),  
"Heuristic and Complete Processes in the Mechanization of Theorem-  
Proving," in Systems and Computer Science, J. T. Hart and S. Takasu,  
eds., pp. 116-124 (University of Toronto Press, Toronto, 1967).
- Robinson, J. A. (1968a),  
"New Directions in Mechanical Theorem Proving," Proc. IFIP Congress,  
Edinburgh (1968).
- Robinson, J. A. (1968b),  
"The Generalized Resolution Principle," in Machine Intelligence,  
Vol. 3, D. Michie, ed., pp. 77-93 (American Elsevier Publishing  
Company, Inc., New York, New York, 1968).
- Robinson, J. A. (1969),  
"Mechanizing Higher Order Logic," in Machine Intelligence, Vol. 4,  
B. Meltzer and D. Michie, eds. (American Elsevier Publishing Company,  
Inc., New York, New York, 1969).
- Robinson, J. A. (1970),  
"The Present State of Mechanical Theorem Proving," Formal Systems and  
Non-Numerical Problem Solving by Computer, M. Mesarovic and R. Banerji,  
eds. (to appear).
- Robinson, G. A., L. T. Wos, and D. F. Carson (1964),  
"Some Theorem-Proving Strategies and Their Implementation," Argonne  
Nat'l Laboratories Technical Memorandum No. 72 (1964).

- Robinson, G. and L. Wos (1969),  
"Paramodulation and Theorem-Proving in First-Order Theories with  
Equality," in Machine Intelligence, Vol. 4, B. Meltzer and D.  
Michie, eds., pp. 135-150 (American Elsevier Publishing Company,  
Inc., New York, New York, 1969).
- Rosenblatt, F. (1962),  
Principles of Neurodynamics (Spartan Books, New York, New York,  
1962).
- Russell, R. (1964),  
"Kalah--The Game and The Program" Artificial Intelligence Project  
Memo No. 22, Stanford University, Stanford, California (3 September  
1964).
- Samuel, A. (1959),  
"Some Studies in Machine Learning Using the Game of Checkers," IBM  
J., Vol. 3, pp. 211-229 (1959). [Reprinted in Computers and Thought  
E. Feigenbaum and J. Feldman, eds., pp. 71-105 (McGraw-Hill Book  
Company, New York, New York, 1963).]
- Samuel, A. (1967),  
"Some Studies in Machine Learning Using the Game of Checkers II.  
Recent Progress," IBM J. Research and Development, Vol. 11, No. 6,  
pp. 601-617 (November 1967).
- Sandewall, E. (1969),  
"Concepts and Methods for Heuristic Search," Proc. Int'l Joint  
Conference on Artificial Intelligence, Washington D.C. (May 1969).
- Selfridge, O., and J. Kelly, Jr. (1962),  
"Sophistication in Computers: a Disagreement," IRE Trans. Info.  
Theory (February 1962).
- Shannon, C. (1950),  
"Programming a Digital Computer for Playing Chess," Philosophy  
Magazine, Vol. 41, pp. 356-375 (March 1950). [Reprinted in The  
World of Mathematics, Vol. 4, Newman, ed. (Simon and Schuster, New  
York, New York, 1954).]
- Shapiro, D. (1966),  
"Algorithms for the Solution of the Optimal Cost Traveling Salesman  
Problem," Sc.D. Thesis, Washington University, St. Louis,  
Missouri (1966).

- Simmons, R. (1965),  
"Answering English Questions by Computer: A Survey," Comm. ACM,  
Vol. 8, pp. 53-70 (January 1965).
- Simmons, R. (1970),  
"Natural Language Question Answering Systems: 1969," Comm. ACM,  
Vol. 13, No. 1, pp. 15-30 (January 1970).
- Slagle, J. (1961),  
"A Computer Program for Solving Problems in Freshman Calculus (SAINT),"  
Doctoral Dissertation, Massachusetts Institute of Technology, Cambridge,  
Massachusetts (1961). [Also printed as Lincoln Laboratory Report  
5G-0001 (10 May 1961).]
- Slagle, J. (1963a),  
"A Heuristic Program that Solves Symbolic Integration Problems in  
Freshman Calculus," J. ACM, Vol. 10, No. 4, pp. 507-520 (October  
1963). [Reprinted in Computers and Thought, E. Feigenbaum and J.  
Feldman, eds., pp. 191-203 (McGraw-Hill Book Company, New York, New  
York, 1963).]
- Slagle, J. (1963b),  
"Game Trees, M and N Minimizing, and the M and N Alpha-Beta Procedure,"  
Artificial Intelligence Group Report No. 3, UCRL-4671, University of  
California Lawrence Radiation Laboratory, Livermore, California  
(November 1963).
- Slagle, J. (1965),  
"Experiments with a Deductive-Question Answering Program," Comm.  
ACM, Vol. 8, pp. 792-798 (December 1965).
- Slagle, J. (1967),  
"Automatic Theorem-Proving with Renamable and Semantic Resolution,"  
J. ACM, Vol. 14, No. 4, pp. 687-697 (October 1967).
- Slagle, J. (1970),  
"Heuristic Search Programs," in Formal Systems and Non-Numerical  
Problem Solving by Computer, M. Mesarovic and R. Banerji, eds.  
(to appear).
- Slagle, J., and P. Bursky (1968),  
"Experiments with a Multipurpose, Theorem-Proving Heuristic Program,"  
J. ACM, Vol. 15, No. 1, pp. 85-99 (January 1968).

- Slagle, J., and J. Dixon (1969),  
 "Experiments with Some Programs that Search Game Trees," J. ACM  
 Vol. 16, No. 2, pp. 189-207 (April 1969).
- Slagle, J., and J. Dixon (1970),  
 "Experiments with the M and N Tree Searching Program," (to appear).
- Solomonoff, R. (1966),  
 "Some Recent Work in Artificial Intelligence," Proc. IEEE, Vol. 54,  
 No. 112 (December 1966).
- Travis, L. (1963),  
 "The Value of Introspection to the Designer of Mechanical Problem-  
 Solvers," Behav. Science, Vol. 8, No. 3, pp. 227-233 (July 1963).
- Travis, L. (1967),  
 "Psychology and Bionics: Many Old Problems and a Few New Machines,"  
 Conference Record 1967 Winter Convention on Aerospace and Electronic  
 Systems (WINCON), Vol. 6, IEEE Publication No. 10-C-42 (February 1967).
- Turing, A. M. (1950),  
 "Computing Machinery and Intelligence," Mind, Vol. 59, pp. 433-460  
 (October 1950). [Reprinted in Computers and Thought, E. Feigenbaum  
 and J. Feldman, eds., pp. 11-35 (McGraw-Hill Book Company, New York,  
 New York, 1963).]
- Waldinger, R., and R. Lee (1969),  
 "PROW: A Step Toward Automatic Program-Writing," Proc. Int'l  
Joint Conf. Artificial Intelligence, Washington D.C. (May 1969).
- Whitney, D. (1969),  
 "State Space Models of Remote Manipulation Tasks," in Proc.  
Int'l Joint Conf. on Artificial Intelligence, pp. 495-507 (1969).
- Wos, L., D. Carson, and G. Robinson (1964),  
 "The Unit Preference Strategy in Theorem Proving," Proc. AFIPS  
1964 Fall Joint Computer Conf., Vol. 26, pp. 616-621 (1964).
- Wos, L., G. Robinson, and D. Carson (1965),  
 "Efficiency and Completeness of the Set of Support Strategy in  
 Theorem-Proving," J. ACM, Vol. 12, No. 4, pp. 536-541 (October 1965).

Yates, R., B. Raphael, and T. Hart (1970),  
"Resolution Graphs," SRI Artificial Intelligence Memo, Stanford  
Research Institute, Menlo Park, California (January 1970).

Zobrist, A. (1969),  
"A Model of Visual Organization for the Game of Go," in Proc. AFIPS  
Spring Joint Computer Conf., pp. 103-112 (1969).