STANFORD RESEARCH INSTITUTE
MENLO PARK, CALIFORNIA

RESEARCH IN ADVANCED FORMAL THEOREM-PROVING TECHNIQUES

By

Cordell Green
Robert Yates
Bertram Raphael
Charles Rosen

Artificial Intelligence Group
Information Science Laboratory

June 1969

I INTRODUCTION

Stanford Research Institute is now engaged in a continuing project* to investigate and develop techniques in artificial intelligence and to apply them to the control of a mobile automaton. Major topics being studied under that project include problem solving, visual pattern recognition, computer representations, question answering, and system integration. The problem-solving research thus far has been based largely upon theorem-proving methods in the first-order predicate calculus. Continued work is planned under the cited project on improving the efficiency of existing theorem-proving techniques and increasing their usefulness in automaton problem-solving situations.

As a by-product of this research, some ideas have recently been developed for a completely new formal mechanical theorem-proving system. This system would be based upon higher order logic and would emphasize the role of semantic information and of flexible control strategies. These ideas are beyond the scope of the main lines of research under the automaton project at this stage. However, a successful implementation of the proposed system would be of tremendous value in several research fields including automaton studies. Therefore we are here proposing a major independent effort in the development of these new theorem-proving ideas.

II OBJECTIVE

This proposal describes a program of research in the development and application of advanced formal theorem-proving techniques. The objective of the proposed work is to design and implement a computer program with general, powerful, and extremely flexible capabilities for both logical inference and data management. The proposed system would have several important potential uses, which can be explored as part of the work:

(1) As a basis for "question-answering systems"--large fact storage and inferential retrieval systems;

(2) As a vehicle for research in automatic problem solving in general, with special emphasis on the problem of automatically writing, verifying, and debugging computer programs;

(3) As a problem-solving framework for use by the mobile automaton system being developed at SRI;

---

*Contract F30602-C-0056, Advanced Research Projects Agency, Department of Defense, and Rome Air Development Center.

(4) As an automated or semiautomated mathematical theorem-proving system.

III BACKGROUND

A. Problem Solving

The development of general methods applicable to the solution of many problems is one of the major goals of current artificial intelligence research. Many research projects that have been classified as problem solving, question answering, inferential information storage and retrieval, theorem proving, program writing, and complex information processing, all have certain basic underlying similarities. We shall refer to any such problem that raises the common central questions of representation, search, and answer generation as a <u>problem-solving</u> task.

Several approaches to the development of general problem-solving systems have been suggested and implemented. Three of the most promising candidates are:

(1) The GPS approach of Newell.[1*] This approach is based upon a pattern-matching capability that identifies differences between objects, followed by the use of transformation operators that potentially can reduce those differences. Although pattern matching is a useful feature, the choice of operators is usually determined by a table and is thus rather inflexible.

(2) The formal language approach. Fikes,[2] Pople,[3] and others have designed formal languages for describing problem-solving processes. Fikes' system includes a search algorithm for evaluating procedures expressed in his language. Such systems, while providing considerable control, require complex encoding (by the user) of information in the usual problem statement.

(3) The theorem-proving approach of Green.[4,5] This approach uses information generated during certain formal proofs of theorems in first-order predicate calculus to construct solutions to problems. Although extremely general, this approach has been somewhat disappointing in performance because of the low efficiency of current mechanical theorem-proving algorithms and the difficulty of introducing strategic control principles into these algorithms.

Each of these problem-solving approaches, and perhaps several others, appears to have both advantages and disadvantages. They are all currently

*References are listed at the end of this proposal.

being studied and compared at SRI (under the automaton project previously cited). This present proposal is for a drastic modification of the theorem-proving approach mentioned above. By designing a new, more powerful mechanical theorem-proving system with integral pattern matching and control language capabilities, we hope to establish a new level of automated problem-solving ability.

B. Mechanical Theorem Proving

The main line of recent theorem-proving research may be traced from Robinson's landmark paper[6] that introduced a rule of inference called the resolution principle. Since then, several research groups have added theorems and heuristics that result in improved performance by resolution-based theorem-proving programs for first-order logic. A recent paper[7] describes how one such program aided in the discovery of a new mathematically significant theorem in lattice theory.

Unfortunately, virtually all existing theorem-proving programs suffer from certain inherent limitations. The major limitations lie in the following areas:

(1) Semantics. Current theorem-proving systems are essentially pure syntactic procedures. In complex problems, however, significant gains in performance can be made only by focusing attention on the semantics of the problem domain, and by invoking strategies that are highly problem-oriented. No mechanism for doing this is presently available.

(2) Strategies. The major fault of current theorem provers is their low efficiency--even simple proofs invariably entail a tremendous amount of computation that is irrelevant to the final solution. Efficiency ultimately depends upon the particular strategies and heuristics employed to search the space of possible proofs. New, more efficient strategies are frequently being discovered or proposed. Unfortunately, however, the stragegy used by a particular theorem-proving program is generally "frozen" into its code so that newer strategies are extremely difficult, if not impossible, to use.

(3) Level of Logic. No existing program operates in a domain of logic significantly beyond the first-order predicate calculus. However, many interesting problems have a natural, compact formulation in higher order logic. (These include problems of writing computer programs, and of describing the strategies of the theorem prover itself.) Although such formulations can always be reduced to first-order logic, the reduction is awkward, tedious, and usually obscures the significance of the statements.

The work described below is aimed at overcoming the above limitations.

IV PROPOSED WORK

During the first year we propose to design and implement a preliminary version of a new theorem-proving program. This version will have several features not present in other theorem provers and will provide a basis for future continued expansion and evolution of the system. In the course of system design, particular attention will be given to the following research areas:

(1) Data structures and data manipulation. This includes the representation of objects, sets, and expressions, and the specification of operations including pattern matching, object construction and transformation, and data scanning.

(2) Strategy specification and control. This refers to methods for specifying, modifying, and monitoring the search and decision procedures used in the system.

(3) Higher order logic and semantics. We shall try to discover efficient methods for extending mechanical theorem-proving techniques to ω-order predicate calculus, and for embedding semantic description and expression evaluation into the basic logical formalism.

In addition, we propose to explore potential applications for the new theorem-proving system while it is being developed. One major application to be studied under this proposal is the area of computer program writing. (Application of this system to robot problem solving will be pursued, when appropriate, under the separate support of the automaton project previously cited.)

V METHOD OF APPROACH

During the past four years we have developed a series of question-answering and problem-solving systems based upon formal theorem-proving methods. As a result of this experience, we now have

(1) A working question-answering program called QA3, which consists of a theorem prover based upon Robinson's resolution rule for the first-order predicate calculus and associated search strategy and data management routines, and

(2) A list of features that have been found desirable but difficult or impossible to implement in the present system in any natural way.

The QA3 program provides a benchmark for comparison of future systems. Our initial goal will be to construct a system that matches the basic capabilities of QA3, but is considerably easier to modify and extend.

The list of features represents design goals for the proposed system. Moreover, these features appear to be highly interrelated. Experience has shown that all of these aspects should be present in order to achieve the desired power in the resulting system. In each case, our goal will be to achieve the desired capability in as general and flexible a manner as possible. These desired capabilities are discussed below.

A. Pattern Matching and Transformations

Pattern matching and transformation operations consist of recognizing and naming certain substructures within an expression and reforming these subparts into new expressions. The system will make several fundamental uses of pattern-matching capabilities. First, all strategy operations, evaluation, and state transformations need access to the properties of expressions; pattern matching offers a convenient technique for extracting those properties which are dependent upon the syntactic structure of the expression. Secondly, some form of pattern matching is necessary in the task of comparing expressions for difference techniques. Finally, most inference rules require a given set of expressions to satisfy some pattern to infer a new expression transformed from the given set.

B. Representation Changes

These operations will facilitate the use of various representations of information. For example, suppose that during a complex problem-solving task three different subprocessors require access to a certain piece of data. Each subprocessor might require the data to be represented in a different form. The three different forms might be functions, predicates, and sets. The representation operators will be able to transform the data into the alternate forms. This facility will also allow experimentation with alternate representations of information and with problems that require finding new representations.

C. Ordered Set Operations

The purpose of these operations is to manipulate ordered sets, indexed sets, implicit sets, computed sets, etc.

There should be a set manipulation language describing set operations. The set operations should include:

- Enumerate a set
- Enumerate a set according to a given ordering relation
- Find elements of a set having specified properties
- Store a set according to a given ordering relation
- Reorder a set and store or enumerate according to the new relation
- Ordered set union, intersection, relative complements
- Add and delete elements of sets
- Create indices for sets
- Find the extension (closure) of a set under specified operators

- Find the extension of a set under specified operators, but grow the extension according to a specified ordering relation on the set
- Create partitions and covering subsets of sets.

Information storage and retrieval of data relevant for inference is an important aspect of a theorem-proving system. Ordered set operations provide a basis for efficient storage and retrieval.

D. Higher Order Logic

The system will be embedded in the formal framework of ω-order predicate calculus. This higher order system will provide great generality and expressive power. The syntax and semantics of any desired portion of the system itself can be expressed in its own language. A recent paper by Robinson[8] develops a foundation for extending resolution theorem provers to this domain.

E. Expression Evaluation

The evaluation of an expression is the process of discovering that the expression denotes a particular object, or has a particular object as its value. The concept of the value of an expression is central to higher order logic since the semantics of the language as well as all proof procedures are defined in terms of the interpretations (values). Consequently, most of the problem-solving system is devoted to finding the values of expressions. In the case where the expression is computable, that is, expressible in terms of computable constructs, then finding the value reduces to the ordinary process of computation or code evaluation.

One major difference between the proposed system and the usual "theorem-proving" systems is illustrated by the importance to the new system of expression evaluation, an essentially semantic operation. Methods called "theorem proving" usually denote a procedure that uses syntactic rules of inference to establish theorems. The significance of such methods is that, in an adequate logical inference system such as resolution, the set of theorems coincides with the set of semantically valid statements. However, in the proposed work we are really only interested in the semantic validity of a statement, and theorem proving is merely a tool that allows us to determine validity. In the higher order system that we are considering, we will directly use semantic methods for determining validity. Such methods promise to be more effective. However, until it is clear that semantic methods will completely dominate syntactic methods, we will of course also carry along a complete inference system.

F. Semantic Description

By semantic description we mean the ability to specify completely the semantics of all expressions in the system. Thus, for example, not only will the system have statements in a strategy language which specify procedures, but it will have the ability to describe and deal with the effect of carrying out a strategy. Another example of this notion is self-description of the program's semantics. Another example is a description of an internal representation of data which represents, say, a set stored sequentially according to a given ordering relation. The expressive power of higher order logic lies in its ability to make statements about other statements, sets of statements, or any functional portion of a statement.

Semantic descriptions will be stated in the language of higher order logic, and evaluation procedures will be used to find denotations.

G. Inference

Inference, along with strategies and representations, constitutes one of the major aspects of theorem proving. The new system will possess at least the inference capabilities of earlier resolution-based systems such as QA3, although it is expected that gradually semantic methods will dominate the syntactic inference methods. Such is already the case for propositional calculus.

We anticipate that the unification process that is central to the resolution inference principle will be extended, in theory and practice, to higher order logic, with special facilities for equality (expressions denoting the same object).

Inference operations will utilize set operations, pattern-match-and-transform operations, and higher order logic operations.

H. Strategy Operations

The purpose of the strategy operations is to create a flexible system capable of change and self-description. The function of a strategy is to specify (schedule) the next operation(s) to be executed or attempted during a process.

There will be a strategy language in which strategies can be described. The system will include an interpreter that can execute strategies specified in this language. The language will be capable of concisely expressing complex and interesting strategies. The strategy operations that will be included will be selected on the basis of their utility in describing strategies and their semantics being expressible, rather than by whether they are easily or most efficiently implementable.

I. Monitoring and Control Operations

Monitoring and control operations deal with the flow and processing of information within the system and with respect to the external world. These operations consist of task scheduling, priorities, interrupts, interactiveness, and input-output. The function of the monitor would be as an overriding control unit which relegates the tasks to be performed, accepts and acknowledges interrupts, and acts as the I-O controller.

(Information requests and transmission subprocesses would interface through the monitor.) Since many problem-solving tasks have independent status, it is important to be able to monitor the performances of the individual tasks to determine which subproblems are explored first, and to interrupt control when other subgoals seem more promising.

VI PRESENT STATUS

The QA3 system is an operational program written in the LISP 1.5 language and running on the SDS 940 computer at SRI. It has been used for experimental studies in theorem proving, question answering, robot problem solving, information retrieval, and reasoning by analogy.

Preliminary work on the new system proposed here is already in progress. Candidate forms for the strategy, monitoring, set and pattern-match and transformation operations have been proposed and are being evaluated. As our next step we plan to implement a version of QA3 in the new formalism in order to test these choices.

The SDS 940 computer will be replaced by a PDP-10 from Digital Equipment Corp. by the end of 1969. The large core memory, more powerful instruction set, and better LISP implementation on the PDP-10 is expected to result in a vast improvement in the efficiency of our experimental programming work. Although the new computer will be a dedicated facility purchased by the U. S. Government for the automaton project, we are confident that permission can be obtained to use the system for the work proposed here.

VII PERSONNEL

The following key personnel are expected to participate in the proposed research.

C. CORDELL GREEN, RESEARCH MATHEMATICIAN
INFORMATION SCIENCE LABORATORY
INFORMATION SCIENCE AND ENGINEERING DIVISION

Specialized professional competence

- Question-answering systems
- Problem-solving systems
- Mathematical logic
- Automatic theorem proving

Representative research assignments at SRI (joined 1966)

- Design and implementation of three question-answering and theorem-proving computer programs
- Finding first-order logic representation of English language facts, questions, and answers
- Extension of resolution-type automatic mathematical proof procedures to constructive proof procedures for question-answering applications
- Design of experimental problem-solving system for SRI robot
- Investigation of automatic program writing, verifying, and debugging techniques

Other professional experience

- Researcher, Synnoetics Systems (Dr. Louis Fein), Los Altos, Calif., 1965
- Engineer, Texas Instruments, Houston, Texas, 1963-64

Academic background

- B. S. in electrical engineering (1964), Rice University
- M. S. in electrical engineering (1965), Stanford University
- Ph.D. in electrical engineering (1969), Stanford University

Publications

- "The Use of Theorem-Proving Techniques in Question-Answering Systems," coauthored with B. Raphael, Proc. 23rd Natl. Conf. ACM (Brandon/Systems Press, Inc., Princeton, N. J., 1968)
- "Theorem Proving by Resolution as the Basis for Question-Answering Systems," Machine Intelligence 4, Michie, ed. (Edinburgh University Press, Edinburgh, Scotland, 1969)
- "Application of Theorem Proving to Problem Solving," Proc. Int'l. Joint Conf. on Artificial Intelligence, Washington, D.C. (1969)

Professional associations and honors

- Association for Computing Machinery
- Institute of Electrical and Electronics Engineers
- Tau Beta Pi
- Sigma Tau

NILS J. NILSSON, SENIOR RESEARCH ENGINEER
INFORMATION SCIENCE LABORATORY
INFORMATION SCIENCE AND ENGINEERING DIVISION

Specialized professional competence

- Artificial intelligence
- Systems theory
- Pattern recognition

Representative research assignments at SRI (joined 1961)

- Studies in the theory of pattern recognition
- Feature detection studies
- Studies in heuristic search procedures
- Automatic theorem-proving studies
- Planning, promotion, and direction of robot systems research
- Head, Artificial Intelligence Group (1963-67)

Other professional experience

- Taught courses in pattern recognition at Stanford University and at University of California, Berkeley, 1962-63
- Acting Associate Professor at Stanford University, Computer Science Department (one-half time, 1968-69)

Academic background

- M.S. in electrical engineering (communication theory)(1956), Stanford University
- Ph.D. in electrical engineering (communication theory)(1958), S.U.

Publications

- Twelve articles on pattern recognition and artificial intelligence
- Learning Machines (McGraw-Hill, 1965)

Professional associations

- Institute of Electrical and Electronics Engineers
- Association for Computing Machinery
- Tau Beta Pi
- Sigma Xi

BERTRAM RAPHAEL, SENIOR RESEARCH MATHEMATICIAN
INFORMATION SCIENCE LABORATORY
INFORMATION SCIENCE AND ENGINEERING DIVISION

Specialized professional competence

- Question-answering systems
- Heuristic problem solving
- Symbol manipulation techniques
- Theorem-proving methods

Representative research assignments at SRI (joined 1965)

- Development of data structures and deductive techniques for on-line question-answering systems
- Studies of problem-solving activity in a simulated robot
- Direction of system design for an experimental "intelligent" automaton
- Survey of computer languages for symbolic and algebraic manipulation

Other professional experience

- Lecturer, Electrical Engineering and Computer Science, University of California at Berkeley; Lecturer, Computer Science, Stanford University; and Instructor, summer course at University of California at Los Angeles
- Consultant, Computer Science Department, RAND Corp., Santa Monica, California
- Assistant Research Scientist, University of California at Berkeley
- Part-time research staff, Bolt, Beranek and Newman, Inc., Cambridge, Massachusetts

Academic background

- B.S. in physics (1957), Rensselaer Polytechnic Institute
- M.S. in applied mathematics (1959), Brown University
- Ph.D. in mathematics (1964), Massachusetts Institute of Technology

Publications

- More than a dozen papers in technical journals and in the proceedings of national and international computer conferences

Professional associations

- Association for Computing Machinery (National Lecturer, 1967-68; founding editor, Newsletter of the group on artificial intelligence)
- Association for Computational Linguistics
- Sigma Xi

CHARLES A. ROSEN, MANAGER, **ARTIFICIAL INTELLIGENCE GROUP**
INFORMATION SCIENCE LABORATORY
INFORMATION SCIENCE AND ENGINEERING DIVISION

Specialized professional competence

- Artificial intelligence
- Pattern recognition
- Solid-state devices (especially piezoelectric)
- Electron physics

Representative research assignments at SRI (joined 1957)

- Developed the Electron Physics Group and the Artificial Intelligence Group, as group manager
- Development of mobile automaton system
- Pattern-recognition and learning-machine studies and applications
- Development of microelectronic devices and systems

Other professional experience

- Assistant Head, Transistor Circuit Group; Head, Dielectric Devices Group; Consulting Engineer, Dielectrics and Magnetics, General Electric Company
- Manager of Radio Department and Spot Weld Department, Fairchild Aircraft, Canada
- Technical investigations for radio and instruments, British Air Commission
- Co-owner, Electrolabs Reg'd., Montreal, Canada, Alarm Intercom Systems
- Lecturer, Stanford University, piezoelectric and ferroelectric devices

Academic background

- B.E.E. (1940), Cooper Union Institute of Technology
- M.Eng. in communications (1950), McGill University
- Ph.D. in electrical engineering (minor in solid-state physics) (1956), Syracuse University

Publications and patents

- Coauthor of Principles of Transistor Circuits, R. F. Shea, editor (John Wiley and Sons, Inc., 1953)
- Coauthor of Solid State Dielectric and Magnetic Devices, H. Katz, editor (John Wiley and Sons, Inc., 1959)
- Author or coauthor of several papers in the fields of piezoelectric devices, learning machines, pattern recognition
- Six patents relating to solid-state devices

Professional associations

- Senior Member of the Institute of Electrical and Electronics Engineers
- Member of the American Physical Society
- Member of the Scientific Research Society of America

RICHARD J. WALDINGER, RESEARCH MATHEMATICIAN
INFORMATION SCIENCE LABORATORY
INFORMATION SCIENCE AND ENGINEERING DIVISION

Specialized professional competence

- Artificial intelligence
- Automata theory
- Logic
- Recursive function theory
- Automatic theorem proving

Other professional experience

- Research assistant, teaching assistant, project scientist, Carnegie-Mellon University, Pittsburgh, Pennsylvania

- Mathematician, Heuristics Laboratory, National Institute of Health, Bethesda, Maryland

Academic background

- A.B. in mathematics (1964), Columbia College, New York, N. Y.

- Ph.D. in computer science (1969), Carnegie-Mellon University, Pittsburgh, Pennsylvania

Publications

- "PROW: A Step toward Automatic Program Writing," Proc. International Joint Conference on Artificial Intelligence, Washington, D. C. (1969)

ROBERT A. YATES, SYSTEMS PROGRAMMER
INFORMATION SCIENCE LABORATORY
INFORMATION SCIENCE AND ENGINEERING DIVISION

Specialized professional competence

- Mathematics
- Programming languages
- Compilers
- Diffraction

Representative research assignments at SRI (joined 1967)

- Design and development of a question-answering computer system based on first-order predicate calculus

Other professional experience

- Member of technical staff, Bell Telephone Laboratories, Holmdel, New Jersey; work on design and implementation of SNOBOL4 programming language
- Programmer, Johns Hopkins University, Physics Department
- Programmer, Politecnico, Mexico City; design and implementation of LISP system and compiler

Academic background

- B.A. in mathematics (1965), Johns Hopkins University
- M.A. in mathematics (1967), Stanford University

Professional associations

- Phi Beta Kappa

REFERENCES

1. G. Ernst and A. Newell, "Generality and GPS," Technical Report, Carnegie Institute of Technology, Pittsburgh, Pa. (January 1967).

2. R. E. Fikes, "Stating Problems as Procedures to a General Problem Solving Program," Proc. Fourth Systems Symposium, Case Western Reserve University, Cleveland, Ohio, November 1968 (to be published).

3. H. E. Pople, Jr., "A Goal-Oriented Language for the Computer," Ph.D. Thesis, Graduate School of Industrial Administration, Carnegie Mellon University, Pittsburgh, Pa. (1969).

4. C. Green, "Theorem Proving by Resolution as a Basis for Question-Answering Systems," Machine Intelligence 4, Michie and Meltzer, eds. (Edinburgh University Press, Edinburgh, Scotland, 1969).

5. C. Green, "Application of Theorem Proving to Problem Solving," Proc. International Joint Conference on Artificial Intelligence, Washington, D.C. (May 1969).

6. J. A. Robinson, "A Machine-Oriented Logic Based on the Resolution Principle," J. ACM, Vol. 12, No. 1, pp. 23-41 (January 1965).

7. J. R. Guard, F. C. Oglesby, J. H. Bennett, and L. G. Settle, "Semi-Automated Mathematics," J.ACM, Vol. 16, No. 1, pp. 49-62 (January 1969).

8. J. A. Robinson, "Mechanizing Higher Order Logic," Machine Intelligence 4, Michie and Meltzer, eds. (Edinburgh University Press, Scotland, 1969).