# STANFORD COMPUTER SCIENCE DEPARTMENT
# RESEARCH REPORT

by

R. Davis and M. Wright
(revised January 1976 by CSD staff)

STAN-CS-76-405
JANUARY 1976

COMPUTER SCIENCE DEPARTMENT
School of Humanities and Sciences
STANFORD UNIVERSITY

S T A N F O R D   C O M P U T E R   S C I E N C E   D E P A R T M E N T

RESEARCH REPORT

January 1976

# TABLE OF CONTENTS

# P R E F A C E

This collection of reports is divided into two sections. The first contains the research summaries for individual faculty members and research associates in the Computer Science Department. Two professors from Electrical Engineering are included as "Affiliated Faculty" because their interests are closely related to those of the Department.

The second section gives an overview of the activities of research groups in the Department. ('Group" here is taken to imply many different things, including people related by various degrees of intellectual interests, physical proximity, or funding considerations. We have tried to describe any group whose scope of interest is greater than that of one person. The list of recent publications for each is not intended to be comprehensive, but rather to give a feeling for the range of topics considered.

This collection of reports has been assembled to provide a reasonably comprehensive review of research activities in the Department. We hope that it will be widely useful -- in particular, students in the Department may find it helpful in discovering interesting projects and possible thesis topics. We expect also that it will be of interest to many other people, both within and outride the Department.

Opinions herein expressed are generally the responsibility of the individual researchers.

Randall Davis
Margaret Wright

Revised January, 1976
Computer Science Department Staff

# SECTION 1

FOREST BASKETT, III
Assistant Professor of Computer Science
and Electrical Engineering

analysis of operating systems
measurement and modeling of
computing systems

Professor Baskett's interests are in the general areas of operating systems . His research has included work on the design of practical operating systems, as well as analysis of scheduling methods, computing system **compþnents,** and memory organizations. He has worked on several kinds of modeling, e.g. of computing systems, program behavior, and file systems.

His future research will be directed toward further design and analysis of computing systems and architectures. The general aim is to determine features that are useful and feasible in systems, and to analyze the effects of these features on system performance.- Techniques are being developed for analysis and comparison of different architectures, configurations, and components , given varying assumptions about the computing environment. Mathematical models, typically stochastic, are used as the bases for analysis and for defining and evaluating the measures to be used for comparison. The hope is to provide soundly based methods that will allow definition and implementation of 'optimized' computing systems. The research in these areas combines theory and practice -- choosing appropriate parameters and measurements for models, and experiments to decide whether the models have validity in representing realized systems.

Professor Baskett sees this research as valuable in the design of newly doveloping operating systems and in the improvement of existing computing systems. Scientific methods for achieving stated goals under given constraints in the design of computing and operating systems should provide the basis for the quantitative engineering of such systems.

THOMAS **BINFORD**
Research Associate, AI Lab

machine vision
perception and automation
representations for use in perception

Dr. **Binford** leads research in computer vision and manipulation. He emphasizes the underlying mathematical representations which support the semantics for **programming** languages for manipulation and industrial assembly (AL) and vision (Vision Language). The work is at two levels, implementation of programming systems, and theory. The applications objectives of funded research are: advanced productivity technology and **image** understanding for photointerpretation and cartography. There is also a small program in visual control of a cart, directed toward planetary exploration. Current research areas **are:**

implementation of a language for industrial assembly (AL);
depth perception using stereo and motion parallax;
motion tracking of moving objects;
visual control of assembly operations;
vision language;
arm control using touch and force sensing;
planning systems for assembly (automatic programming; very high level language).

Students have made major contributions in perception and recognition of complex objects and in texture description. There is much to be done in the problems described above, and in high level vision systems, description procedures for complex objects, representations of space for efficient **planning** of collision-free trajectories for manipulators.

BRUCE G. BUCHANAN
Research Computer Scientist

artificial intelligence
scientific inference
theory formation
biomedical applications

Dr. Buchanan is interested in artificial intelligence, particularly in the general problems of scientific inference, theory formation, and knowledge acquisition by computer. His work with the DENDRAL project has been aimed at elucidating techniques for generating plausible explanations of empirical data. The model of scientific inference in the DENDRAL program is heuristic search through a space of possible explanations of analytic data in organic chemistry.

Theory formation is carried out in the context of the Meta-DENDRAL program, which finds regularities in large sets of data and general principles which will account for them. A heuristic search model is used for this program also, but the search space is a space of possible general principles. Empirical theory formation is one method of imparting knowledge of a scientific discipline to a performance program. The task domain is a subset of organic chemistry.

His interest in biomedical applications of artificial intelligence has led to involvement with an interdisciplinary project at the Medical School, MYCIN, whose goal is to provide computerized therapy consultation. An Interesting aspect of this work is its emphasis on a system with both the capacity to offer competent therapeutic advice and the ability to incorporate into the data base new knowledge learned through interaction with users.

Dr. Buchanan has also done some unusual work on computer-aided legal reasoning which investigated the possibilities of automated case analysis. Rather than simple information retrieval, the work surveyed the problem of providing an analysis of each case according to the legal principles under consideration.

VINTON CERF
Assistant Professor of Electrical Engineering

operating systems
computer networks

Professor Cerf's current research is in the general field of computer networks and computer communication systems. He has worked previously on numerous aspects of computing systems: the design and implementation of computer system measurement; a study of parallel computer modeling; analysis of multiprocessor systems; network measurement; and interactive graphics.

He is now working primarily on three problems related to computer communication system protocols. The first problem is to define for interconnected packet switching networks a *set* of protocols and message formats to permit maximum throughput while preserving sequencing, and allowing duplicate detection, flow control, alternate routing, and encryption status reporting. This research is intended to be applied to new ARPA Network protocols, and the creation of generally accepted packet switching standards. Another problem is the development of a graph-theoretic approach to understanding the effects of topology on throughput, delay, and congestion in networks. Finally, some of his work deals with the economics of networks, involving primarily **charging** strategies, regulation issues, and interfacing standards.

Professor Cerf believes that computer networks are an area of increasing importance as they become progressively more widespread. The need for computer communication has grown explosively; computer networks offer a new means for improved communication and sharing of results.

**VACLAV** CHVATAL
Assistant Professor of Computer Science
and Operations Research

combinatorics
graph theory
linear programming

Professor Chvatal works in combinatorics and, in particular, in graph theory; his interest is biased towards extremal problems **("Erdos-type** combinatorics") and design of efficient combinatorial algorithms rather than enumeration. He is also interested in linear programming and its applications in combinatorics. He likes to work with students and combinatorics seems to be just right for that: indeed, its problems are easy to state and, in order to solve them, one has to be just smart rather than well-versed in his background. (Anyhow, there often is no background to speak of.) At present, Professor Chvatal keeps himself-busy working on a textbook in graph theory; he sincerely hopes to complete it in a finite time.

GEORGE B. DANTZIG
Professor of Operations Research and
Computer Science (Criley Chair of Transportation)

Professor Dantzig is interested in the optimization of large-scale systems . In particular, in the modelling and optimization of large-scale energy systems. He is also interested in combinatorial mathematics and mathematical programming.

LESTER D. EARNEST
Research Scientist, SAIL

display-oriented timesharing
documentation languages

The design and use of display terminals and timesharing systems to enhance individual productivity is a central interest of Les Earnest. This includes both the development of good interactive equipment (audio-visual displays, keyboard, etc.) and system software that permits efficient control of multiple processes. Display terminals can (but usually don't) help keep contexts straight as the individual switches among a number of concurrent tasks.

He is also interested in the development of advanced documentation languages. In the next decade or so, the use of computer facilities for creating, modifying, and disseminating documentation is expected to increase greatly. The problem of efficiently creating and representing high quality text, drawings, and photographs in a way that will permit reproduction on a wide variety of display and printing devices is far from solved. Several "first cut" documentation languages have been developed at the Artificial Intelligence Laboratory.

EDWARD A. FEIGENBAUM
Professor of Computer Science

heuristic programming, DENDRAL
information processing psychology

The primary focus of Professor Feigenbaum's work in the past several years has been the Heuristic Programming Project (primarily the DENDRAL program) whose high-level goal is an understanding of the process of scientific theory formation. The vehicle currently used is data from chemical mass spectra, and the work proceeds in two parallel efforts -- first, the development of information-processing models of hypothesis induction from specific, individual spectra,-and second, theory induction from large collections of spectra. The first of these is embodied in the performance program called Heuristic DENDRAL. The development of automatic theory formation has seen realization in the **Meta-DENDRAL** program which attempts to construct sets of rules about molecular fragmentation by generalization from large sets of data.

Professor Feigenbaum describes the long-term goals of the Heuristic Programming Project as: the achievement of power in AI problem-solving programs using various types of knowledge obtained from experts; constructing homogeneous heuristic program structures in terms of which this knowledge can function; and constructing programs for extracting such knowledge and embodying it in formalized theories.

ROBERT W. FLOYD
Professor of Computer Science

complexity theory
analysis algorithms

Professor Floyd has worked in many areas of computer science: programming language design, sorting algorithms, language and compiler theory, formal linguistics, and mechanical theorem proving. His present research involves primarily the area of analysis of algorithms, in particular the theory of minimal algorithms. Computing problems can often be simply characterized and solved by simple algorithms; but known algorithms may require more time (or space, or other scarce resource) than the best possible algorithm to solve the problem. His research, after postulating models of computation which are reasonably accurate descriptions of present and foreseeable computing equipment, attempts (1) to prove absolute lower bounds on the expected time, or the maximum time, required for any algorithm to solve a given problem, and (2) to construct very efficient algorithms for given problems, thereby providing upper bounds on necessary expected or maximum time. The difference between upper and lower bounds can suggest where to look for improvements in one or the other, possibly suggesting revision of the model of computation, followed by renewed attempts on goals (1) and (2). The final result ideally should be the design of an actual computer program which is probably the minimal time program.

Several problems that have been considered from this viewpoint are computing quantiles (elements of an ordered set at a fixed rank; e.g., medians) in minimum expected time and minimum maximum time, sorting with two tapes in **minimax** time, transposing boolean matrices, permuting records on disc or drum storage, and adding in minimum time. The ultimate aim of this work is to provide a calculus of optimal algorithms, which would allow one to systematically assemble the best possible algorithm for a broad range of practical problems on a broad range of computing equipment.

Professor Floyd is also working on the design and implementation of a programming language (Chiron) primarily for student use, to be suitable for teaching structured programming systematically to novices, and to be nearly universal in its capabilities. His experience with current languages and methods for teaching introductory programming suggests that technical details tend to overwhelm questions of systematic design. Chiron represents an attempt to provide a programming environment in which, to the largest extent possible, one designs a program by designing the process which the program carries out.

JOHN T. GILL, III
Assistant Professor of Electrical Engineering

computational complexity
probabilistic computation

Professor Gill's work is in the area of computational complexity and information theory, with emphasis on axiomatic and concrete complexity theory. He has recently worked on three main topics. The first is probabilistic computation (the study of a "coin-tossing computer"); the goal of this work is to decide whether certain kinds of computation can be carried out faster on a "coin-tossing" machine than on other kinds of computers.

Another topic is the study of efficient ways of making fair decisions based on the outputs of a biased information source. An example is to analyze successive tosses of a biased coin to obtain a "fair guess." The criterion of efficiency means in this example to determine as many fair decisions as possible from the smallest number of tosses.

A third research area involves variable-length source codes -- for example, in transmission of messages, where frequently occurring strings of symbols can be transmitted in compressed form. Professor Gill is interested in the mathematical and combinatorial properties of these codes, whose formal structure corresponds to that of binary trees.

His future work will be in the area of probabilistic computation mentioned above, which has many interesting applications. He also plans to consider independence and parallelism in computation. A formal theory is to be developed to define what it means for two processes to be independent, and to analyze the implications of independence in parallel computing.

GENE H. GOLUB
**Professor** of Computer Science

matrix computation
structured linear systems
least-squares and eigenvalues

Professor Golub's work has the unifying theme of matrix computation, with the aim of devising and analyzing algorithms for solving numerical problems that arise in a variety of applications. His research is oriented toward development of methods based on mathematical theory, with an emphasis on practical considerations of computing.

He is currently working on several different problems in numerical analysis. One of these is the fast direct solution of systems of linear equations arising from elliptic partial differential equations; these techniqdes are particularly useful for solving Poisson's equations in a rectangle, and are very widely used. Once a "black-box" is developed to solve this subproblem, similar problems can then be solved by iteration and mathematical manipulation. Another area of interest is the solution of various kinds of least-squares problems, many of which arise in data analysis. He feels that a very useful technique for the nonlinear case involves the use of stable linear techniques; a particular application is to the analysis of ozone in the Los Angeles air in order to analyze the effect of pollution devices, and seasonal parameters. Calculation of eigenvalues is a further area being studied. Different matrix structures are being considered; some recent uses of this work have been in analysis of ocean tides and nuclear reactor fuel control. Finally, Professor Golub is working on methods for updating stable matrix **factorizations,** which are increasingly used in mathematical **optimiza**tfon programs and control theory.

Professor Golub feels that numerical analysis questions are at the heart of many different problems in the physical sciences, social sciences and statistics. A strong numerical analysis technology exists which can satisfactorily be used, and it is hoped his future research will continue to have wide applications.

C. CORDELL GREEN
Assistant Professor of Computer Science

automatic programming
production automation
biomedical applications

Professor Green's primary interest has been in automatic programming. His group is studying the problems of finding new methods of specifying programs, codification of programming knowledge, and implementation of working program-writing systems. The main emphasis is on codification of the considerable body of list-processing programming knowledge. An example is an analysis of all information necessary for a machine to synthesize any of a single class of sort programs. Another interesting feature of the research is some emphasis on 'human' methods of program specification, such as example input-output pairs, generic examples, annotated traces, and natural language descriptions. Prototype systems have been developed that can write short programs (fewer than seven lines), including sort, merge, reverse, list flattening, etc. A recent project is the PUNS program understanding system involving several students. The one-year goal of the project is the automatic synthesis of a six-page concept formation program from an interactive specification consisting primarily of natural language.

To a letter extent, he is involved with the production automation project at the AI Lab, where the effort is to design a very specific automatic programmer that writes plans for the assembly of small engines by the Hand-Eye system. The current goal is the generation of a detailed plan for the assembly of a chainsaw, given the kind of high-level task description now used for assembly-line workers.

Professor Green is part of a group working on a biomedical application, namely the design and implementation of a program to give antimicrobial therapy consultation in a clinical setting. A prototype system has been developed and tested, and work has started on the problem of automatic knowledge acquisition from experts. A complete set of therapy rules for a particular class of infections may include several hundred rules. The acquisition, manipulation, and interaction of these rules present interesting challenges in a real-world problem situation.

His interest in automatic programming is sparked by the feeling that it is the "most critical problem in AI," and is a prerequisite to the development of intelligent systems. Such systems require changing representations, changing strategies, and doing concept formation, all of which involve reprogramming. And in order to reprogram very complex systems, humans will require machine assistance. He finds the involvement in the antimicrobial therapy system interesting as a study in knowledge acquisition "where it can be done" -- in a real problem from a real domain.

JOHN G. **HERRIOT**
Professor of Computer Science

spline functions
partial differential equations

Professor Herriot's interests are in the field of numerical analysis, with emphasis on implementation of algorithms for computation. He has previously worked on elliptic partial differential equations, especially on the "method of particular solutions," where a linear combination of particular solutions is computed which satisfies in some sense the given boundary conditions. These methods are useful in fluid flow and other physical problems.

In more recent work he has studied how to compute interpolating natural spline functions most efficiently and accurately. Although the theory has been known for some time, he has dealt with the specific problem of combining efficient coding and maximum numerical accuracy. He has written procedures to solve the general case, as well as procedures for the cubic case and the quintic case which take advantage of simplifications to speed up the calculations tremendously. Various special kinds of splines are also being considered in this same context.

He is now making a survey of available algorithms for computing spline functions for use in solving interpolation problems, smoothing problems and other similar problems utilizing spline functions. Comparisons will be made with a view to determining the most efficient and most accurate algorithms to use.

He also plans to work in the future on applications of spline functions to partial differential equations, such as the finite element method.

Professor **Herriot** feels that this work is important because there is often a gap between the mathematical theory of a problem and the actual computation of a numerical solution. Implementation of an algorithm for the computer involves a combination of mathematical and programming knowledge. Since the ultimate goal is to provide a computed solution, the numerical analyst must be concerned with programs as well as theory.

DONALD E. **KNUTH**
Professor of Computer Science

analysis of algorithms
combinatorics and discrete mathematics

The broad area of Professor **Knuth's** work is analysis of algorithms, i.e., making quantitative statements about how good algorithms are. This research has two "flavors." The first is analyzing various aspects of particular algorithms -- for example, determining how fast they are and how much space they require, and comparing several algorithms for solving the same problem to decide which is best in terms of given criteria. The second is developing the tools of discrete and combinatorial mathematics required to answer such questions about computer methods. A more general problem is to prove that a method is optimum in the class of all possible ways to solve a problem. This approach leads to questions about definition and study of the basic complexity of algorithms.

The emphasis is on the computer methods that are the most fundamental and application-independent. The motivation for this research is to obtain more understanding of known methods, and to learn the fine points of their behavior under varying conditions. Such detailed study is at the foundation and center of computer science, not on the periphery; it is like "fertilizing the soil and establishing law and order rather than pushing back the frontiers."

Professor Knuth feels that there is an unlimited set of problems to be solved in this field, with no end in the foreseeable future. His continuing work on his series of books, which summarize and bring together related results, is aimed to provide a thorough foundation in the essentials of computer science.

JOSHUA LEDERBERG
Professor of Genetics
Professor of Computer Science (by courtesy)

artificial intelligence
machine-aided inference in experimental science

Professor Lederberg's research interests are rooted in his experimental work in molecular biology but for many years he has been investigating ways in which computer science could be used to support the work of the laboratory investigator in the cognitive domain. His concrete efforts in this direction relate to the DENDRAL research project and to the SUMEX computer facility.

At the present time his main focus of investigative interest in computer science is how to represent formal and informal knowledge in the field of molecular genetics, so as to facilitate machine induction of new hypothetical principles to be tested in the laboratory. Specifically, the repertoire of procedures available in the laboratory will be abstracted so as to be included in the library of legal moves or transformations of input structures. The problem space of molecular genetics will be abstracted so as to be represented by a set of hypothetical structures. The general principles of heuristic tree pruning and systematic generation that have been exemplified in the DENDRAL work on organic molecular structure will be applied in this new domain in a fashion to suggest the most efficient experimental procedures to test a given structural hypothesis. The next stage of the program is the translation of real problems as they arise from day to day in the laboratory into structural representations, and then eventually to translate the statements in a form similar to which they appear in the existing scientific literature into a machinable format. It is not proposed to do this final translation by purely machine methods at the present time but to afford a maximum facility for a man-machine interface towards this end.

Connected problems that have arisen in the framework of this study relate to some topological problems in systematic graph theory.

The SUMEX project is an effort to facilitate human to human interaction in the field of computer science by proving a network for common access to a unified computer (software and hardware) facility to a national community of cooperating investigators.

DAVID C. **LUCKHAM**
Research Associate, AI Lab

theorem-proving
program verification
automatic program construction

Dr. **Luckham** has been working for the past few years on theorem-proving, program verification, and automatic program construction; three corresponding systems, intended for on-line interactive use, have been developed.

The theorem-prover has been used in many successful experiments testing its usefulness in proving theorems in different areas of mathematics, mainly algebra and geometry. Some steps have been taken to develop its application to information retrieval; question-answering facilities have been programmed and tested on small data bases, but nothing further has been done yet. The prover is also being used as part of a system for verifying the correctness of programs written in PASCAL.

The PASCAL verifier is based on a verification condition generator and incorporates special procedures for algebraic simplification and problem reduction in addition to the theorem prover. A number of programs, including standard algorithms for sorting linear arrays, have been verified. Programs that manipulate pointers and records, including a standard garbage collection -algorithm have very recently been verified using this system. Logarithmic time bounds on some sorting and searching algorithms have also been verified. Work in this area is very much "in progress." Methods for applying the verifier to debugging are being developed. Currently the system is being extended to deal with programs containing parallel processes and data types such as history sequences.

The automatic programming system is an interactive system that writes programs to satisfy given input-output specifications. The system requires as input a programming environment consisting, roughly speaking, of primitive functions and procedures, rules of composition and logical facts. It will take account of certain kinds of advice from the user in solving a problem. If successful, the system will output the solution in the form of a program in a language somewhat similar to a subset of Algol containing assignments, function calls, conditional branches, while loops, and non-recursive procedure calls. This system has been used to generate programs for robot control and automation problems, for every-day planning, and for computing arithmetical functions.

**JOHN** MCCARTHY
Professor of Computer Science

artificial intelligence
mathematical theory of computation
timesharing

Professor McCarthy has devoted most of his recent research efforts to representation theory and the mathematical theory of computation. His work in representation theory -- how facts about a particular situation are represented in the computer -- is oriented toward formalisms from **mathematical** logic, in particular predicate calculus and set theory. Research in this area includes the development of an interactive proof-checking and theorem-proving system (FOL, First Order Logic) by Richard Weyrauch. The mathematical theory of computation is concerned with proving correctness and assertions about programs, and operates-by taking a statement in a formal langauge about a program's purpose as a theorem to be proved.

Current timesharing projects include the development of services for home terminals (the news service based on the Associated Press editor is a **start)** and a system for a uniform access to data bases stored in many different computers.

The common problem of the non-exportability of programs has led him to consider the need for a standardized programming language with standardized facilities for interaction with people, files, computer networks, etc. While this work has not yet reached the status of a full-fledged research project, . it would involve discovering a framework for and then implementing the primitive operations performed by programs so that they were possible on any major operating system.

Professor McCarthy sees the problem of artificial intelligence as the "grand-daddy of all science problems," in the sense that if a good solution can be found, the possibility exists for programs capable of bootstrapping themselves to high levels of intelligence. The work is still in an exploratory stage.

He has also supervised Ph.D. theses in chess programs, speech recognition, proving compilers correct, and other areas of artificial intelligence, mathematical theory of computation, and timesharing.

Current research projects include: (1) verification of parts of an actual working operating system; (2) developing a HUNCH language for expressing intuitions about how proofs of conjectures (i.e., expected theorems) are likely to be found (a very primitive HUNCH language already exists); (3) programming new deduction rules and special strategies for particular problem areas such as the verification of programs containing parallel processes; and (4) development of an automatic programming system as part of a PASCAL compiler.

continue to become bigger, and hence require more safeguards; and with electronic components becoming cheaper, the user can afford and ask for **increasingly** sophisticated reliability equipment and mechanisms.

EDWARD J. MCCLUSKEY
Professor of Computer Science
and Electrical Engineering

digftal systems
reliability

Professor McCluskey's current work is in the general area of computer reliability, or fault-tolerant computing, which involves the study of methods to detect, analyze, and correct the errors that may occur in a computing system. The field of reliability includes a broad spectrum of topics and applications, ranging from specific and practical to general and theoretical. Professor McCluskey is working with several different projects, all investigating varying aspects of computer reliability. One particular effort is a study with NASA of a computer system installed in an airplane. This kind of implementation provides a chance to try out theoretical results in a practical environment.

A major emphasis of research in reliability is the design of general techniques for improving reliability, as well as methods for evaluating and comparing the resulting techniques. For example, an early model for reliability was von **Neumann's** proposal of triplicated systems, where the non-matching value of the three is considered to be at fault. This model has been generalized to that of triplicated systems with spares, and to more complex configurations; several new techniques for improving reliability are being studied in these more general settings. Another project along these lines involves the actual design of a self-diagnosing mini-computer with the capability of detecting internal errors and reporting their presence and location to the outside world. The general problem of fault testing and diagnosis in arbitrary logic networks is also being studied; methods of network testing include manipulation of gate-equivalent algebraic expressions and probabilistic test generation.

Professor McCluskey believes that reliability is an important research area because most topics have aspects that involve reliability in one form or another, so that many different problems can be studied with respect to a unified approach. Furthermore, reliability in computing is becoming more and more significant for several reasons: computer systems are being used in an increasing number of places where their reliability is vital; computer systems

JOSEPH OLIGER
Assistant Professor of Computer Science

ordinary differential equations
partial differential equations

Professor Oliger's main interest is computational methods for time dependent problems whose solutions are governed by ordinary and partial differential equations. His present work deals primarily with problems in continuum mechanics with applications in geophysical fluid dynamics (numerical weather prediction, oceanographic calculations and geophysical prospecting) and aerodynamics. Some of this work deals with initial boundary-value problems associated with limited area forecasts and the study of small scale phenomena in bounded regions. The solutions of these problems are governed by hyperbolic and parabolic partial differential equations. There are still several critical mathematical questions which need to be answered for these problems before computational methods can be placed on a firm footing so this work is a combination of applied mathematics and numerical analysis. This work also deals with the efficiency of such methods. This requires the development of useful parametrizations of these problems.

Professor Oliger believes that efficient, useful methods for computational problems in mathematical physics requires interdisciplinary effort from the applications area, applied mathematics, numerical analysis and other areas of computer science such as data structures and languages. It is often necessary that the problem be reformulated on several levels in order that the final product be successful.

ROBERT E. **TARJAN**
Assistant Professor of Computer Science

analysis of algorithms
graph theory

The aim of Professor **Tarjan's** research has been to develop the most efficient algorithms for a variety of problems, to develop general techniques and data structures to use in building algorithms, and to study the inherent complexity of problems for simple but realistic computation models.

Ongoing research includes a study of the axiomatic basis of linear algebra techniques, the application of such techniques in non-numeric problems such as arise in operations research and global program **optimization,** and the efficient application of the techniques. Other projects include a study of minimum spanning tree algorithms, of optimum alphabetic search tree algorithms, and of graphical search techniques.

**TERRY** WINOGRAD
Assistant Professor of Computer Science and Linguistics

natural language
representation of knowledge

Professor Winograd's recent work has been in computer understanding of natural language, which he sees as a key to understanding the much broader problem of human cognition. "I'm interested in how people think, and natural language is a good way to get at that." He sees representation of knowledge as a critical subproblem, and is investigating the correspondence between a set of pieces of knowledge and its incarnation in a program for natural language. One approach to this adopts an automatic programming-like view of the human knowledge assimilation process, in suggesting the presence of an internal parser and compiler which transform bits of knowledge into internal subroutines available for use. His approach includes the belief that such automatic programming should be based on a strong foundation of knowledge about **programming** rather than on transformations suggested by input/output pairs. He also feels that learning by example is useful for knowledge acquisition, and that much may be gained by investigating the structures **nec**essary for its implementation.

His interest in AI is aroused by its implications about the character of human intelligence, and he believes that the computational metaphor will make an important difference in many fields. It is, he feels, a new and exciting way of looking at many classic problems in varied fields.

# SECTION 2

## RESEARCH IN COMBINATORICS AND ANALYSIS OF ALGORITHMS

Professors Knuth, **Chvátal,** and **Tarjan,** together with graduate students and mathematicians who are invited to visit Stanford for extended periods, are actively pursuing research in combinatorial mathematics. This subject, which links mathematics with operations research, artificial intelligence, electrical engineering, and computer science, is presently flourishing in many parts of the world, and the group at Stanford has been holding productive seminars. Problems relevant to the analysis of algorithms are given special consideration.

Recent publications include:

Knuth, D. E. and R. W. Floyd, "Notes on Avoiding 'Go To' Statements," Inf. **Proc.** Letters 1 **(1971),** 23-31. (Also STAN-CS-70-148)

Knuth, D. E., "Optimum Binary Search Trees," **ACTA** Information 1, **(1971), 14-**25. (Also STAN-CS-70-149)

Knuth, D. E., "Examples of Formal Semantics," STAN-CS-70-169.

Floyd, R. W. and D. E. Knuth, "The Bose-Nelson Sorting Problem," **STAN-CS-70-**177.

Knuth, D. E., "An Empirical Study of **Fortran** Programs," Software, vol. 1, **(1971), 105-133.** (Also STAN-CS-70-186)

Hopcroft, J., "An N Log N Algorithm For Minimizing States in a Finite Automation," STAN-CS-70-190.

Hopcroft, J., "An N Log N Algorithm For Isomorphism of Planar Triply Connected Graphs," STAN-CS-70-192.

Knuth, D. E., "The Art of Computer Programming - Errata et Addenda," **STAN-CS-70-194.**

Hopcroft, J. and R. **Tarjan,** "Planarity Testing in V Log V Steps: Extended Abstracts," STAN-CS-71-201.

Rnuth, D. E., "Mathematical Analysis of Algorithms," **Proc.** Int. Math. Congress, Nice, (1970); **Proc.** IFIP Congress, (1971). (Also STAN-CS-71-206.

Graham, R. L., D. E. Rnuth, and T. S. Motzkin, "Complements and Transitive Closures," **Discrete** Mathematics, 1972. (Also STAN-CS-71-214)

Debruijn, N. G., D. E. Rnuth and S. 0. Rice, "The Average Height of Plane Trees," Graph Theory and Comput., 1972. (Also STAN-CS-71-218)

**Tarjan,** R., "An Efficient Planarity Algorithm," STAN-CS-71-244.

**Fredman,** M. and D. E. Rnuth, "Recurrence Relations Based on **Minimiation,"** STAN-CS-71-248.

Forsythe, G. E., "Von Neumann's Comparison Method for Random Sampling from the Normal and Other Distributions," STAN-CS-72-254.

**Chvátal,** V., "Edmonds Polyhedra and Weakly Hamiltonian Graphs," Math. Programming 5, **(1973), 29-40.**

Wirth, N., "On Pascal, Code Generation, and the CDC 6000 Computer," **STAN-CS-**72-257.

Brown, H., "Some Basic Machine Algorithms for Integral Order Computations," STAN-CS-72-258.

Crane, C., "Linear Lists and Priority Queues as Balanced Binary Trees," **STAN-**cs-72-259.

Pratt, V., "Shellsort and Sorting Networks," STAN-CS-72-260.

Klarner, D. and R. Rivest, "A Procedure for Improving the Upper Bound for the Number of **N-Ominoes,"** STAN-CS-72-263.

**Chvátal,** V., "Degrees and Matchings," STAN-CS-72-268.

Klarner, D. and R. Rado, "Arithmetic Properties of Certain Recursively Defined Sets," STAN-CS-72-269.

**Chvátal,** V. and J. Sichler, "Chromatic Automorphisms of Graphs," J. Comb. Theory 14, **(1973), 209-215.**

Klarner, D. and R. Rado, "Linear Combinations of Sets of Consecutive Integers," STAN-CS-72-274.

Klarner, D., "Sets Generated by Iteration of a Linear Operation," STAN-CS-72-275.

Chandra, A. K., "On the Solution of Moser's Problem in 4 Dimensions, and Related Issues," STAN-CS-72-286.

**Chvátal,** V., D. A. Klarnerand D. E. Knuth, "Selected Combinatorial Research Problems," STAN-CS-72-292.

**Fredman,** M. J., "Growth Properties of a Class of Recursively Defined Functions," STAN-CS-72-296.

Klarner, D. A. and R. Rivest, "Asymptotic Bounds for the Number of Convex **N-Ominoes,"** STAN-CS-72-327.

Klarner, D. A., "A Finite Basis Theorem Revisited," STAN-CS-73-338.

Levine, D. R., "Computer Based Analytic Grading for German Grammar Instruction," STAN-cs-73-343.

Klarner, D. A., "The Number of **SDR's** in Certain Regular Systems," STAN-CS-73-354.

Amble, D. and D. E. Knuth, "Ordered Hash Tables," The Computer Journal, 1974, 135-142. (Also STAN-CS-73-367)

Knuth, D. E., "A Review of 'Structured Programming'," STAN-CS-73-371.

Knuth, D. E. "Sorting and Searching - Errata and Addenda,' STAN-CS-73-392.

**Hoare,** C. A. R., 'Parallel Programming: An Axiomatic Approach," STAN-CS-73-394.

**Hoare,** C. A. R., 'Recursive Data Structures,' STAN-CS-74-400.

**Hoare,** C. A. R., 'Hints on Programming Language Design," STAN-CS-74-403.

Knuth, D., and F. Stevenson, 'Optimum Measurement Points for Program Frequency Counts," BIT, 13 **(1973),** 313-322.

Knuth, D., "The Asymptotic Number of Geometries," Journal of Combinatorial Theory, 1974.

Knuth, D., "Wheels Within Wheels,' Discrete Mathematics, 1974.

Knuth, D., "Permutations with Nonnegative Partial Sums," Discrete Mathematics. 5, 367-371, 1973.

Knuth, D., "The **Triel:** A New Solution," Journal of Recreational Mathematics, 6, 1-7, 1973.

**Knuth,** D., " Structured Programming with 'Go To' Statements,' Computing Surveys, 6, 1974. (Also STAN-CS-74-416)

Morris, J. H., V. R. Pratt and D. E. Knuth, "Fast Pattern Matching in Strings," STAN-cs-74-440.

Knuth, D., 'Estimating the Efficiency of Backtrack Programs,' Math. of Computation, vol. 29, 1975, 121-136. (Also STAN-CS-74-442)

Knuth, D., "Random **Matroids,"** Discrete Mathematics, vol. 12, 1975, 341-358. (Also STAN-cs-74-455)

Porter, T. and I. Simon, 'Random Insertion into a Priority Queue Structure,' STAN-CS-74-460)

Krogdahl, S., 'A Combinatorial Base for Some Optimal Matriod Intersection Algorithms," STAN-CS-74-468.

**Trabb-Pardo,** L., "Stable Sorting and Merging with Optimal Space and Time Bounds," STAN-CS-74-470.

**Chvátal,** V. and D. **Sankoff,** 'Longest Common Subsequences of Two Random Sequences,' **STAN-CS-75-477.**

Knuth, D., "Notes on Generalized Dedekind Sums," STAN-CS-75-480.

Erdos, P., "On Packing Squares," STAN-CS-75-483.

Graham, R. L. and E. Szemeredi, "On Subgraph Number Independence in Trees," STAN-cs-75-484.

Erdos, P. and E. Szemeredi, "On Multiplicative Representations of Integers," STAN-G-75-485.

Bollobas, B., P. Erdos, and E. Szemeredi, 'On Complete Subgraphs of R-Chromatic Graphs," STAN-CS-75-488.

Szemeredi, E., "Regular Partitions of Graphs," STAN-CS-75-489.

Gosper, R. W., "Numerical Experiments with the Spectral Test," STAN-CS-75-490.

Sedgewick, R., "Quicksort," STAN-CS-75-492.

Kurki-Suonio, R., "Describing Automata in Terms of Languages Associated with Their Peripheral Devices," STAN-CS-75-493.

Krogdahl, S., "The Dependence Graph for Bases in Matroids,' STAN-CS-75-495.

Graham R. L. and L. Lovasz, 'Distance Matrices of Trees," STAN-CS-75-497.

Kurki-Suonio, R., "Towards Better Definitions of Programming Languages," STAN-CS-75-500.

Szemeredi, E., "On Sparse Graphs with Dense Long Paths," STAN-CS-75-504.

Chvital, V., "Some Linear Programming Aspects of Combinatorics,' STAN-CS-75-505.

Eve, J., "On Computing the Transitive Closure of a Relation," STAN-CS-75-508.

Overton, M. and A. Proskurowski, 'Finding the Maximal Incidence Matrix of a Large Graph," STAN-CS-75-509.

Yao, A. and D. E. Knuth, "Analysis of the Subtractive Algorithm for Greatest Common Divisors," STAN-CS-75-510.

Dubost, P. and J. M. Trousse, "Software Implementation of a New Method of Combinatorial Hashing," STAN-CS-75-511.

Tarjan, R., "Applications of Path Compression on Balanced Trees," STAN-CS-75-512.

Chvátal, V. and C. Thomassen, 'Distances in Orientation of Graphs," STAN-CS-75-517.

Chvátal, V. and P. L. Hammer, "Aggregation of Inequalities in Integer Programming, STAN-CS-75-518.

Tarjan, R., "Graph Theory and Gaussian Elimination," STAN-(X-75-526.

DIGITAL SYSTEMS LABORATORY

The Digital Systems Laboratory (DSL) is a collection of faculty and graduate students from Computer Science and Electrical Engineering who are interested in digital systems and computer organization; it is part of the Stanford Electronics Laboratories. Faculty members include Professors Edward J. McCluskey (Director), Michael J. Flynn, Vinton Cerf, and John Wakerly. Among the areas of active research at DSL are reliability, computer networks, microprogramming and emulation, computer architecture, and computer performance. There are several special-interest groups which meet regularly at seminars devoted to these particular areas.

Recent publications include:


Cerf, V. G. et. al., Trivalent Generalized Moore Networks on Sixteen Nodes, Utilitas Mathematica, vol. 6, 1974, 259-283.

Cerf, V. G. et. al., "A Lower Bound on Average Path Length in Regular Graphs," Networks, vol. 4, no. 4, 1974, 335-342.

Cerf, V. G. et. al., "Topological Design Considerations in Computer-Communication Networks," Computer Communication Networks, (R. L. Grimsdale and F. F. Kuo, eds.), Academic Book Services, Holland, Netherlands, 1974.

Cerf, V. G. and R. Kahn, "A Protocol for Packet Network Intercommunication," IEEE Trans. on Communications, vol. COM-22, no. 5, May 1974, 637-648.

Flynn, M. J., "Interpretation, Microprogramming and the Control of a Computer," Introduction to Computer Architecture, (H. S. Stone, ed.), Science Research Associates, Inc., 1975, 432-472.

Flynn, M. J., "Microprogramming: Another Look at Computer Control," Proc. IEEE, accepted for publication.

Flynn, M. J., "Microprogramming Concepts," Software Systems Principles: A Survey, (P. Freeman, ed.), Science Research Associates, Inc., 1974, 70-101.

Gschwind, H. W. and E. J. McCluskey, Design of Digital Computers, 2nd edition, Springer-Verlag Publishers, Inc.3 New York, 1975.

McCluskey, E. J., "Micros, Minis, and Networks," Calcolo, (Italy), accepted for publication.

Parker, K. P. and E. J. McCluskey, "Probabilisitc Treatment of General Combinational Networks," IEEE Trans. on Computers, vol. C-24, no. 6, June 1975, 668-670.

Parker, K. P. and E. J. McCluskey, "Analysis of Logic Circuits with Faults Using Input Signal Probabilities," IEEE Trans. on Computers, vol. C-24, no. 3, May 1975, 573957%.

Wakerly, J. F., "Comments on 'Asynchronous Sequential Machines Designed for Fault Detection'," IEEE Trans. on Computers, vol. C-24, no. 7, July 1975, 759-760.

Wakerly, J. F., "Transient Failures in Triple Modular Redundancy Systems with Sequential Modules," IEEE Trans. on Computers, vol. C-24, no. 5, May 1975, 570-572.

Wakerly, J. F., "Partial, Self-Checking Circuits and Their Use in Performing Logical Operations," IEEE Trans. on Computers, vol. C-23, no. 7, July 1974, 658-666.

# HEURISTIC PROGRAMMING PROJECT

Principal Investigators: Edward Peigenbaum, Joshua Lederberg, Carl Djerassi
Associate Investigator: Bruce Buchanan
Research Associates: Dennis Smith, Ray Carhart, Harold Brown
Visiting Researcher: Carroll Johnson
Post-Doctoral Fellow: Tomas Varkony (Chemistry)

The Heuristic Programming Project is an interdisciplinary group working on information processing models of scientific reasoning. A performance program, termed Heuristic DENDRAL, reasons from empirical data to explanatory hypotheses. Its task domain is mass spectrometry, one discipline within analytic chemistry, in which the empirical data are analytic data from a mass spectrometer, and the explanatory hypotheses are the graph structures of organic molecules. The program is written as a heuristic search program through the space of molecular graphs. Knowledge of chemistry in general, and mass spectrometry in particular, is used to constrain the search.

Recent advances in the project include the enhancement of the capabilities of the performance program to analyze estrogens and estrogen mixtures, as well as the development of an algorithm for generating cyclic graph structures. Work is also currently going on in the application of the program to biomedical problems. In particular, it is hoped that the mass spectra analysis can be coupled to information available from gas chromatography, and this powerful combination used in such problems as analysis of the chemical components of urine in premature infants.

A theory-formation program, termed Meta-DENDRAL, reasons from collections of empirical observations to general rules (a theory), also in the domain of mass spectrometry. Theory formation of this sort is seen as one important way for large knowledge-based systems to acquire the expertise that they need for high performance. In this case, the rules that are inferred are used by the analysis program described above.

Robert Engelmore, in collaboration with chemists at UCSD, is investigating new ways to elucidate the three-dimensional structure of protein molecules, relying primarily on inferential reasoning rather than extensive empirical data. Working with 3-D intensity data produced by X-ray diffraction, they hypothesize partial structures and use Patterson functions for verification. The approach is still being tested in molecules of known structure, but will soon be tried on those whose structure is as yet unknown.

Collaboration with the MYCIN Project in the Medical School has resulted in a-knowledge-based consultant for infectious disease diagnosis and therapy. The program is a goal-directed search program using a knowledge base in the form of production rules. It can reason under uncertainty, use global strategy rules, explain its reasoning processes, and acquire new rules.

Recent publications include:

Brown, H., L. Masinter, and L. Hjelmeland, "Constructive Graph Labeling Using Double Cosets, to appear in Discrete Mathematics.

Buchanan, B. G., "Applications of Artificial Intelligence to Scientific Reasoning." Proceedings of Second USA-Japan Computer Conference, August, 1975.

Buchanan, B. G., "Scientific Theory Formation by Computer." Proceedings of NATO Advanced Study Institute on Computer Oriented Learning Processes, 1974, **Bonas,** France.

Buchanan, B. G. and E. A. Feigenbaum, and N. S. Sridharan, "Heuristic Theory Formation: Data Interpretation and Rule Formation," in Machine Intelligence 7, Edinburgh University Press (1972).

Buchanan, B. G. and N. S. Sridharan, "Rule Formation on Non-Homogeneous Classes of Objects," in Proceedings of the Third International Joint Conference on Artificial Intelligence, Stanford, California, August (1973).

**Carhart,** R. E., D. H. Smith, H. Brown, and C. Djerassi, "Applications of Artificial Intelligence for Chemical Inference. XVII. An Approach to Computer-Assisted Elucidation of Molecular Structure." Journal of the American Chemical Society, submitted for publication.

**Carhart,** R. E., D. H. Smith, H. Brown, and N. S. Sridharan, "Applications of Artificial Intelligence for Chemical Inference. XVI. Computer Generation of Vertex Graphs and Ring Systems." Journal of Chemical Information and Computer Science (formerly Journal of Chemical Documentation), 15 **(1975), 124.**

Davis, R., B. G. Buchanan, and E. H. Shortliffe, "Production Rules as a Representation for a Knowledge-Based Consultation System.," Available as Technical Report AIM-266, Stanford Artificial Intelligence Laboratory, Stanford, California. Also submitted for publication in Artificial Intelligence, October 1975.

Davis, R. and J. King, "Overview of Production Systems," Machine Representation of Knowledge, Proceedings of the NATO **ASI** Conference, July, 1975.

Dromey, R. G., B. G. Buchanan, J. Lederberg, and C. Djerassi, "Applications of Artificial Intelligence for Chemical Inference. XIV. A General Method for Predicting Molecular Ions in Mass Spectra." Journal of Organic Chemistry, 40 **(1975),** 770.

Feigenbaum, E. A., "Computer Applications: Introductory Remarks," in Proceedings of Federation of American Societies for Experimental **Biology,** vol. 33, no. 12 (Dec. 1974) **2331-2332.**

Michie, D. and B. G. Buchanan, Current Status of the Heuristic **DENDRAL** Program for Applying Artificial Intelligence to the Interpretation of Mass Spectra, August 1973.

Shortliffe, E. H., S. G. **Axline,** B. G. Buchanan, R. Davis, and S. N. Cohen, "A Computer-Based Approach to the Promotion of Rational Clinical Use of **Antimicrobials** (invited paper). Proceedings of the International Symposium on Clinical Pharmacy and **Clinical Pharmacology,** Boston, Massachusetts, **18** September 1975.

Shortliffe, E. H. and B. G. Buchanan, "A Model of Inexact Reasoning in Medicine," Math. Biosci. **23,** 351-379 (1975).

Shortliffe, E. H., R. Davis, S. G. **Axline,** B. G. Buchanan, C. C. Green, and S. N. Cohen, "Computer-Based Consultations in Clinical Therapeutics: Explanation and Rule-Acquisition Capabilities of the MYCIN System. **Comput.** Biomed. Res. 8, 303-320 (1975)

Shortliffe, E. H., F. S. Rhame, S. G. **Axline,** S. N. Cohen, B. G. Buchanan, R. Davis, A. C. Scott, R. Chavez-Pardo, and W. J. van Melle, **"MYCIN:** A Computer Program Providing Antimicrobial Therapy Recommendations **(abstract** only). Presented at the 28th Annual Meeting, Western Society for Clinical Research, **Carmel,** California, 6 February 1975. Clin. Res. **23,** 107a (1975). Reproduced in Clinical Medicine p. 34, August 1975.

Smith, D. H., B. G. Buchanan, R. S. Engelmore, H. Aldercreutz, and C. Djerassi, "Applications of Artificial Intelligence for Chemical,Inference IX. Analysis of Mixtures Without Prior Separation as Illustrated for **Estrogens,"** to appear in Journal of the American Chemical Society.

Smith, D. H., B. G. Buchanan, W. C. White, E. A. Feigenbaum, C. Djerassi, and J. Lederberg, "Applications of Artificial Intelligence for Chemical Inference X. **Intsum.** A Data Interpretation Program as Applied to the Collected Mass Spectra of Estrogenic Steroids, to appear in Tetrahedron.

NUMERICAL ANALYSIS GROUP

The numerical analysis group at Stanford includes Professors Gene H. Golub, John G. Herriot and J. Oliger as the permanent faculty members, and about ten graduate students; their interests are supplemented by those of visiting scholars who are at Stanford for various parts of each year. There is great activity with respect to visitors during the summer term. The research in numerical analysis involves two closely related aspects: development of mathematically based theory to solve particular problems; and implementation of appropriate computer algorithms, with emphasis on programming considerations such as coding efficiency, numerical accuracy, generality of application, data structures, and machine independence.

A fairly broad library of programs to solve numerical problems is informally maintained by the numerical analysis group, in cooperation with the Stanford Center for Information Processing. Connections are also maintained with the program library efforts of the NATS project at Argonne, Illinois, and the Nottingham Algorithms Group in England.

Recent publications include:

Björck, A. and G. H. Golub, "Eigenproblems for Matrices Associated with Periodic Boundary Conditions," Stanford Computer Science Report CS-75-486 (1975).

Concus, P. and G. H. Golub, "Use of Fast Direct Methods for the Efficient Numerical Solution of Nonseparable Elliptic Equations," SIAM J. Num. Anal. **10**, no. 6, 1103-1120, (1973).

Cottle, R. W., G. H. Golub and R. Sacher, "On the Solution of Large, Structured Linear Complementarity Problems: III," Stanford Computer Science Report CS-74-439 (1974).

Gill, P., G. H. Golub, W. Murray and M. Saunders, "Methods for Modifying Matrix Factorizations," Math. Comp. 28, no. 126, 505-535 (1974).

Golub, G. H., L. Jenning and W. Yang, "Waves in Periodically Structured Media," J. Comp. Phy. **17**, no. 4, 349-357 (1975).

Golub, G. H. and V. Pereyra, "The Differentiation of Pseudo-Inverses and Nonlinear Least Squares Whose Variables Separate," SIAM J. Num. Anal. **10**, no. 2, 413-432 (1973).

Golub, G. H. and E. Seneta, "Computation of the Stationary Distribution of an Infinite Stochastic Matrix of Special Form," Bull. Austral. Math. Soc. 10, 255-261 (1974).

Golub, G. H. and J. H. Wilkinson, "Ill-conditioned Eigensystems and the Computation of the Jordan Canonical Form," Stanford Computer Science Report CS-75-478 (1975).

**Herriot,** J. G. and C. Reinsch, Algorithm **472,** Procedures for Natural Spline Interpolation, Comm. ACM **16,** no. **12,763-768** (1973).

**Herriot,** J. G. and C. Reinsch, **Algol60,** Procedures for the Calculation of Interpolating Natural Quintic Spline Functions, Stanford Computer Science Report CS 402, January 1974. (Submitted to ACM Trans. on Mathematical Software.)

Kaufman, L., "The LZ Algorithm to Solve the Generalized Eigenvalue Problem," Stanford Computer Science Report CS **363** (1973).

Oliger, J., "Fourth Order Difference Methods for the Initial-Boundary Value Problem for Hyperbolic Equations," Math. Comp. **28,** 15-25 (1974).

Oliger, J., "Hybrid Difference Methods for the Intitial-Boundary Value Problem for Hyperbolic Equations," Stanford Computer Science Report **CS-75-481.**

# ARTIFICIAL INTELLIGENCE AND SAIL

Director: John McCarthy
Associate Director: Les Earnest

Artificial intelligence is the name given to the study of intellectual processes and how computers can be made to carry them out. Most workers in the field believe that it will be possible to program computers to carry out any intellectual process now done by humans. However, almost all agree that we are not very close to this goal and that some fundamental discoveries must be made first. Therefore, work in AI includes trying to analyze intelligent behavior into more basic data structures and processes, experiments to determine if processes proposed to solve some class of problems really work, and attempts to apply what we have found so far to practical problems.

The idea of intelligent machines is-very old in fiction, but present work dates from the time stored program electronic computers became available starting in 1949. Any behavior that can be carried out by any mechanical device can be done by a computer, and getting a particular behavior is just a matter of writing a program unless the behavior requires special input and output equipment. It is perhaps reasonable to date AI from A.M. Turing's 1950 paper [1]. Newell, Shaw and Simon started their group in 1954 and the M.I.T. Artificial Intelligence Laboratory was started by McCarthy and Minsky in 1958.

## Board Games

Early work in AI included programs to play games like chess and checkers and kalah and go. The success of these programs was related to the extent that human play of these games makes use of mechanisms we didn't understand well enough to program. If the game requires only well understood mechanisms, computers play better than humans. Kalah is such a game. The best rating obtained in tournament play by a chess program so far is around 1700 which is a good amateur level. The chess programmers hope to do better.

## Formal Reasoning

Another early problem do-main was theorem proving in logic. This is important for two reasons. First, it provides another area in which our accomplishments in artificial intelligence can be compared with human intelligence. Again the results obtained depend on what intellectual mechanisms the theorem proving requires, but in general the results have not been as good as with game playing. This is partly because the mathematical logical systems available were designed for proving metatheorems about rather than for proving theorems in.

The second reason why theorem proving is important is that logical languages can be used to express what we wish to tell the computer about the world, and we can try to make it reason from this what it should do to solve the problems we give it. It turns out to be quite difficult to express what humans know about the world in the present logical languages or in any other way. Some of what we know is readily expressed in natural language, but much basic information about causality and what may happen when an action is taken is not ever explicitly stated in human speech. This gives rise to the representation problem of determining what is known in general about the world and how to express it in a form that can be used by the computer to solve problems. As this is written - summer 1973 - the representation problem seems to be the key problem in artificial intelligence.

## Publications

The results of current research in artificial intelligence is published in the journal Artificial Intelligence, and in more general computer science publications such as those of the ACM and the British Computer Society. The ACM has a special interest group on artificial intelligence called SIGART which publishes a newsletter. Every two years there is an international conference on artificial intelligence which publishes a proceedings. The fourth and most recent was held in the U.S.S.R. at Tbilisi in September 1975, and the proceedings are available [2].

Some current books and papers are listed the the end of this writeup [3, 4, 5, 6]. The Stanford Artificial Intelligence Laboratory has a series of research reports (A.I. Memos) which are included in the reports put out by the Computer Science Department. One can get on the list to receive announcements of reports by writing to them [7]. A ten-year summary of SAIL research, including abstracts of all reports, is available [8], as is a summary of more recent work [9]. We also have some 16mm films for loan [10].

## Stanford Artificial Intelligence Laboratory

SAIL was started in 1963 and moved to its present location at 1600 Arastradero Road, Palo Alto, in 1966. Since the beginning, our work has been mostly supported by the Advanced Research Projects Agency of the Defense Department, but we have also received support from the National Science Foundation, the National Aeronautics and Space Agency, the National Institutes of Health, and private foundations.

Our research goals include all areas of artificial intelligence, mathematical theory of computation (the problem of specifying properties of computer programs and proving that the programs meet their specifications), studies in natural language, and studies in time-sharing and other aspects of computer systems . However, the main work of the laboratory since its inception has included the following:

1. Computer vision. Images are obtained from a television camera and are processed to describe the scene in ways appropriate to the purpose of the program. These purposes include manipulation, driving a vehicle, and simply checking our understanding of perceptual mechanisms.

2. Manipulation. Programs are developed to assemble objects out of parts. For example, one system that we developed assembles the water pump from a Model T Ford.

3. Driving a vehicle. Experiments are continuing on a computer-controlled electric cart equipped with a television camera that attempts to navigate both outdoors and with the building on the basis of visual information.

4. Theorem proving. Programs using J. Alan Robinson's resolution method of proving theorems in first order logic are used to prove theorems in mathematics, to prove properties of computer programs, and to generate computer programs having prescribed properties.

5. Mathematical theory of computation. Methods for proving properties of programs are developed. Programs for checking proofs in first order logic and in a special logic of computable functions have been developed.

6. Game playing. Some work in checkers, chess and go has been done, but we are not active in this field at present.

7. Speech recognition. Also inactive currently.

8. Design automation. Programs have been written to allow a computer designer to put logic diagrams in the computer and diagrams for printed circuit boards. An integrated system checks consistency of the various diagrams, permits changes to be made easily, and produces output for the automatic manufacture of printed circuit boards and for automatic wire wrapping machines.

9. Natural language understanding. Programs are being developed to "understand" and act upon information presented in the form of text in English and other natural languages.

10. Time-sharing systems. There is also development of time-sharing techniques especially for display-oriented systems.

11. Affiliated projects. There have been two independent projects affiliated with SAIL in the areas of psychiatric applications of computers (recently moved to UCLA) and computer music. The latter group has developed computer simulations of acoustic spaces in which sound sources can be placed at various locations, using quadraphonic techniques.

## Computer Facilities

The computer facilities of the laboratory currently comprise PDP-10 and PDP-6 processors, 256K words of core, a swapping disk, a disk system for file storage (capacity of 6.5 x $10^9$ bits), 60 raster-type display terminals, and 6 vestor-type displays. The system includes standard peripherals, including a plotter, a Xerox Graphics Printer, A-D and D-A converters, a connection the ARPA network, and a few external phone lines.

There is also a PDP-11/45 system for controlling real-time devices, with 200K words of MOS and core memories and an SPS-41 processor. Connected to the system are television cameras, mechanical arms, and (via radio links) a computer-controlled cart with a TV camera.

## Staff

The personnel of the laboratory include faculty and students of the Computer Science Department and a few other departments at Stanford University, the professional staff, and some research associates. Potential graduate students wishing to specialize in artificial intelligence and do research in the laboratory should usually apply for admission to the Computer Science Department at Stanford University. Potential research associates should apply directly to the laboratory.

## References


**[1]** Turing, A. M., Computing Machinery and Intelligence, Mind. 1950. Reprinted in J. R. Newman (ed.), The World of Mathematics, Vol. 4, p. 2099, Simon and Shuster, New York, (1956).

**[2]** Advanced Papers of the Fourth International Joint Conference on Artificial Intelligence, September 1975. Available from Artificial Intelligence Laboratory, 545 Technology Square, Room 817, Cambridge, Massachusetts 02139, USA.

**[3]** Newell, Allen and Simon, Herbert A., Human Problem Solving, Prentice-Hall, Englewood Cliffs, New Jersey (1972).

**[4]** Nilsson, Nils, Problem Solving Methods in Artificial Intelligence, McGraw-Hill, Inc. (1971).

**[5]** Nilsson, Nils, Artificial Intelligence in Proceedings IFIP Congress 1974, Stockholm, Sweden, (1974) and also Artificial Intelligence Center Technical Note #89, Stanford Research Institute, Menlo Park, California, (1974).

**[6]** Slagle, James R., Artificial Intelligence: The Heuristic-Programming Approach, McGraw-Hill, (1971).

**[7]** To obtain abstracts of recent Computer Science Reports, including AI Memos, write to:

Documentation Services
Computer Science Department
Stanford University
Stanford, California 94305

**[8]** Earnest, Lester, (ed.), The First Ten Years of Artificial Intelligence Research at Stanford, **Stanford AI Lab Memo AIM-228,** July 1973. Available in **microfiche** only from Documentation Services (address in letterhead), or in hard copy or microfiche as accession No. **ADA776233/9WC** from:

National Technical Information Service
Springfield
Virginia 22161

**[9]** Earnest, Lester, (ed.) Recent Research in Artificial Intelligence, Heuristic Programming, and Network **Protocols,** Stanford AI Lab Memo AIM-**252,** July 1974. **Available** in microfiche from AI Lab (address in letter-head) or in hard copy or microfiche from NTIS (address just above) as accession No. ADAO02246.

**[10]** To request a list of films on AI Lab research, write to:

Documentation Services
Stanford Artificial Intelligence Lab.
Stanford University
Stanford, California 94305