CS 129

RESEARCH
IN THE
COMPUTER SCIENCE DEPARTMENT
STANFORD UNIVERSITY

APRIL 21, 1969

RESEARCH

in the

COMPUTER SCIENCE DEPARTMENT

STANFORD UNIVERSITY

summarized by

Professor W. F. Miller

April 21, 1969

RESEARCH IN THE COMPUTER SCIENCE DEPARTMENT

STANFORD UNIVERSITY

April 21, 1969

Summarized by

W.F. Miller

The research program of the Computer Science Department can perhaps be best summarized in terms of its research projects. The chart on the following page lists the projects and the participation by faculty and students. Two observations should be made to complete the picture. Within the Artificial Intelligence Project, the Stanford Computation Center, the SLAC Computation Group, and the INFO project, there are a large number of highly competent professional computer scientists who add greatly to the total capability of the campus. Also, there are a number of projects in other schools or departments which are making significant contributions to computer science. These, too, add to the total computer environment.

RESEARCH PROJECTS OF THE COMPUTER SCIENCE DEPARTMENT

Project	Numerical Analysis	Artificial Intelligence	Programming Models and the Control of Computing Systems	Graphic Processing Analysis, Languages, Data Struture	SLAC	Computer Simulation of Belief Systems	H-P 2116 Control Computer	Mathematical Programming Language	Operations Research**	Parallel Operations in Digital Networks and Systems***	Programming Languages	Analysis of Algorithms
Supporting Agency(s)	ONR NSF	ARPA	AEC	NSF	AEC	NIH	Hewlett-Packard	NSF	AEC ARO NSF ONR	Joint Services	Proposed	Proposed
Principal Investigator(s)	G.E. Forsythe J.G. Herriot G.H. Golub	J. McCarthy E. Feigenbaum A. Samuel	W.F. Miller	W.F. Miller	W.F. Miller	K.M. Colby	H. Stone	G.R. Dantzig	G.R. Dantzig	E. McClusky	R. Floyd D. Knuth	R. Floyd D. Knuth
Other Investigator(s) (Faculty) and (Res. Assoc.)		J. Feldman R. Floyd Z. Manna D. Reddy N. Nilsson L. Earnest J. Beauchamp B. Buchanan M. Hueckel D. Luckham G. Sutherland	N. Nilsson H. Stone		D.J. Gries J.R. Ehrman C. Riedl G.A. Robinson	J. Feldman T. Callahan R. Schank		D. Gries A. Manne (Op. Res)	R.W. Cottle A.S. Manne (Op. Res.) R. Wilson (Business)		Z. Manna	Z. Manna
Number of Graduate Research Assistants*												
Computer Science Dept.	4	12	3	1	8	3	1	1	1	1		
Supported from Other Dept.		5			1			1	5	2		
Fellows and outside Support* Working on Project		3			3							

* Some graduate students are teaching assistants, and some are not yet associated with a research project.

** Professor Dantzig holds a joint appointment in Computer Science and Operations Research; these research projects are administratively in the latter department.

*** Professor McCluskey holds a joint appointment in Computer Science and Electrical Engineering; this project administratively in the latter department.

TABLE OF CONTENTS

Section	Page
Numerical Analysis Research	1
Artificial Intelligence	8
Programming Models and the Control of Computing Systems	54
Graphic Processing: Analysis, Languages, Data Structures	56
SLAC Computation Group	62
Computer Simulation of Belief Systems	69
The Hewlett-Packard Control Computer	74
Mathematical Programming Language and Operations Research	75
Parallel Operations in Digital Networks and Systems	79
Research in Programming Languages	80
Research in the Analysis of Algorithms	82

NUMERICAL ANALYSIS RESEARCH

The major computer science research projects under way in numerical analysis are supported by the Office of Naval Research and the National Science Foundation. Professor George Forsythe is the Principal Investigator of the ONR project, which has been supported at Stanford for 10 years. Professor Forsythe and Professor John G. Her-riot are Principal , Investigators of the project supported by NSF. Professor Gene Golub is also associated with the NSF project. Some of Professor George Dantzig's work is closely related, but is separately described under "OPERATIONS RESEARCH" below. There is a good prospect of new support for research in numerical analysis.

Both existing projects provide modest funds to bring visiting faculty to Stanford for periods of up to one year.

Statements of the purpose of the research and a listing of the research areas of the two projects follow.

Office of Naval Research

Purpose of project: The project has the purpose of conducting research to increase the effectiveness with which automatic digital computers are used to solve contemporary scientific and technological problems of a mathematical nature. This includes the invention, criticism, and particularly the mathematical analysis of algorithms for numerical computation. In the future it will deal in growing amounts also with algorithms for symbolic computation that enters mathematical problems. It will also include a study of on-line man-machine interaction in **the solution** of mathematical problems.

Research areas: Research projects are selected from the following areas:

1. Analysis and computation methods for large, sparse matrices
2. Minimizing functions of many variables
3. Conversational mode of solving problems
4. Maintenance of scientific program libraries
5. Improving, certifying and recording algorithms
6. Round-off analysis

National Science Foundation

Purpose of Project: The project has purposes that differ very little from those of the Office of Naval Research project. The problem areas. now being investigated include the solution of linear and nonlinear systems of algebraic equations and partial differential equations, and linear and non-linear least-squares problems.

Research Areas: Research projects are mainly selected from the following areas:

1. Analysis of large sparse matrices
2. Algorithms for least squares problems, including linear least squares problems with inequality constraints, or with a quadratic constraint, perturbation theory, exponential fitting, interactive data fitting, and algorithms for large matrices
3. Solution of partial differential equations
4. Minimizing functions of many variables
5. Conversational mode of solving problems

6. Maintenance of scientific program libraries

7. Publication and evaluation of algorithms.

Numerical Analysis with Emphasis on Least Square Problems

Purpose of Project: One of the problems that frequently arises in a computation laboratory is that of fitting data by least squares. These problems arise from a number of disciplines in a variety of contexts, and often have special forms. For example, experimental physicists fit bubble-chamber data with helices; radiologists fit data with the sum of exponentials.

In recent years, there has been a great deal of activity on studying both linear and nonlinear least-squares problems. Golub and his colleagues have made a number of contributions to the literature in the study of linear problems. A number of algorithms have been developed which solve the linear case in a very accurate manner. It is proposed that research be extended into the area of non-linear least squares problems to develop accurate algorithms for solving such problems, and also for linear least squares problems with special constraints.

Computation Center

A few of the faculty members in numerical analysis receive part of their support from the Stanford Computation Center. This is in return for consulting and leadership in the Center's scientific programming library. For-years the Center has had good compilers and languages that were dialects of Algol 60, and our library had been built around Algol. With the departure of the IBM 7090 in September 1967 and the Burroughs B5500 in December 1967, the Center now has only System/360 computers. This has left a real gap in the scientific programming library. Under OS/360 we have several Fortrans, a rigid Algol 60, a very slow PL/1 compiler, and an extended Algol dialect called Algol W. Only Fortran and PL/1 provide for direct access input/output to disks, the storage of precompiled programs, and linkage with machine-language subroutines. Our scientific programs include those furnished by IBM, about fifty in Fortran or OS/Assembler language prepared by John Welsch, John Ehrman, and others at the SLAC facility, and a number now being collected.

The situation is roughly this: many of the world's best small algorithms are published in Algol 60; the most versatile compiler language for System/360 is Fortran; and IBM states that PL/1 is the language of the future. The problem, then, is what language should a scientific programming library now deal with. This is truly a perplexing question, and there, is surely no clear answer.

Despite the over-all uncertainty, we are continuing to develop useful algorithms. G.H. Golub and his students have been particularly active in this, and are generally using Algol W as the development language, while publishing in Algol 60. Some students under G.E. Forsythe are collecting Fortran programs from Argonne National Laboratories and other sources, putting them on disk files of the 360/67 accessible from the Wylbur console system, making indexes, and creating some publicity about their existence. To a limited extent, these students are offering to consult for scientific computer users at Stanford who have special problems. It requires a good deal of effort to convince people to take the trouble to seek advice, and then one cannot always help them!

I. RECENT PUBLICATIONS

(a) Technical Reports

Technical Report No.	Title	Author	Date
CS 81	CALCULATION OF GAUSS QUADRATURE RULES	G. H. Golub J. H. Welsch	11-3-67
CS 83	ITERATIVE REFINEMENTS OF LINEAR LEAST SQUARES SOLUTIONS BY HOUSEHOLDER TRANSFORMATIONS	A. Björck G. Golub	1-19-68
CS 88	RELAXATION METHODS FOR CONVEX PROBLEMS	S. Schechter	2-16-68
CS 96	INTERVAL ARITHMETIC DETERMINANT EVALUATION AND ITS USE IN TESTING FOR A CHEBYSHEV SYSTEM	L.B. Smith	4-26-68
cs 102	INTEGER PROGRAMMING OVER A CONE	A. Pnueli	7-12-68
CS 104	A NUMERICAL INVESTIGATION OF THE SIMPLEX METHOD	R. Bartels	7-68
CS 105	EPSILON CALCULUS	P. Richman	8-68
CS 107	A THREE-STAGE VARIABLE-SHIFT ITERATION FOR POLYNOMIAL ZEROS AND ITS RELATION TO GENERALIZED RAYLEIGH ITERATION	M. Jenkins J. Traub	8-68
CS 124	MATRIX DECOMPOSITIONS AND STATISTICAL CALCULATION	G.H. Golub	3-69
CS 128	THE METHOD OF ODD/EVEN REDUCTION AND FACTORIZATION WITH APPLICATION TO POISSON'S EQUATION	B.L. Buzbee G.H. Golub C.W. Nielson	4-69

Also published, but not directly related to numerical analysis was:

CS 93	COMPUTER SCIENCE AND EDUCATION Invited address to International Federation of Information Processing 68 (Edinburgh).	G.E. Forsythe	3-27-68

(b) Printed Publications

Richard H. Bartels and Gene H. Golub, "Stable numerical methods for obtaining the Chebyshev solution to an overdetermined system of equations," Communications of Assoc. Computing Mach., vol. 11 (1968), pp.401-406.

A. Björck and G. Golub, 'Iterative refinement of linear least square solutions by Householder transformation," BIT, vol. 7 (1967), pp. 322-337.

George E. Forsythe, 'On the asymptotic directions of the s-dimensional optimum gradient method," Numerische Math., vol. 11 (1968), pp. 57-76.

Gene H. Golub, "Least squares, singular values, and matrix approximation," Aplikace Matematiky, vol. 13 (1968), pp. 44-51.

David Gries, 'Characterizations of certain classes of norms,' Numerische Math., vol. 10 (1967), pp. 30-41.

W.C. Mitchell, 'An evaluation of Golomb's constant,' Math. Computation, vol. 22 (1968), pp. 411-415.

G. Polya, "Graeffe's method for eigenvalues," Numerische Math., vol. 11 (1968), pp. 315-319.

Samuel Schechter, 'Relaxation methods for convex problems,' SIAM J. Numer. Anal. vol. 5 (1968), pp. 601-612.

J.H. Wilkinson, 'Almost diagonal matrices with multiple or close eigenvalues," Linear Algebra and Its Applications, vol. 1 (1968), pp 1-12.

George E. Forsythe, "Computer Science and Education," Proceedings of International Federation for Information Processing, Congress 1968, Edinburgh, Invited Addresses, pp. 92-106.

Richard H. Bartels and Gene H. Golub, Algorithm 328 Chebyshev Solution to an Overdetermined Linear System, Comm. ACM 11 (June 1968), pp. 428-430.

S. Bergman, J.G. Herriot and T.G. Kurtz, On Numerical Calculation of Transonic Flow Patterns, Math of Comp. vol. 22 (1968), pp. 13-27.

Richard H. Bartels and Gene H. Golub, Algorithm 350, Simplex Method Procedure Employing LV Decomposition, Comm. ACM to appear May 1969.

J.M. Varah, 'The calculation of the eigenvectors of a general complex matrix by inverse iteration,' Math. Computation, vol. 22 (1968) pp. 785-791.

J.M. Varah, 'Rigorous machine bounds for the eigensystem of a general complex matrix", Math Computation, vol. 22 (1968), pp. 793-801.

Richard H. Bartels and Gene H. Golub, "The simplex method of linear programming using LU decomposition," Comm. Assoc. Comput. Mach., to appear in May 1969.

Lyle B. Smith, 'Interval arithmetic determinant evaluation and its use in testing for a Chebyshev system," Comm. Assoc. Comput. Mach., vol. 12 (1969), pp. 89-93.

(c) Editorial Work

George E. Forsythe served as an associate editor of Numerische Mathematik, and wrote occasional reviews for Computing Reviews.

Gene H. Golub served as an associate editor for Numerische Mathematik and the Mathematics of Computation.

Several persons helped J.G. Herriot with refereeing contributions to the Algorithms section of the Communications of the Association for Computing Machinery.

Introduction (John McCarthy)

The work of the Stanford Artificial Intelligence Project is charted in Figure 1. Besides artificial intelligence work proper, it includes work in mathematical theory of computation and in computing with symbolic expressions.

In this introduction we shall make a few remarks about the outline as a whole, then comment on the status of the subdivisions and our plans for future work in them, More detailed descriptions and plans will be given later.

This is not the place for a full discussion of the present state of research in artificial intelligence. Instead, we shall make a few rather flat statements about the general situation into which our work fits.

1. Artificial intelligence is the experimental and theoretical study of perceptual and intellectual processes using computers. Its ultimate goal is to understand these processes well enough to make a computer perceive, understand and act in ways now only possible for humans.

2. The information for-this study comes in part from observation of human behavior, including self-observation, but mainly from experiments with programs designed to solve problems chosen to require the intellectual processes under study.

3. This understanding is at present in a very preliminary state. No one can say how long it will take to reach the understanding required to duplicate human intellectual performance because there are

STANFORD A.I. PROJECT

ARTIFICIAL INTELLIGENCE	MATH THEORY OF COMPUTATION	SYMBOLIC COMPUTATION
	properties of algorithms, automatic deduction	REDUCE

EPISTEMOLOGY	HEURISTICS	INTERACTION WITH PHYSICAL WORLD	MAN-MACHINE INTERACTION
Advice Taker, higher mental functions	DENDRAL machine learning	Hand-Eye, computer-controlled vehicle	DENDRAL speech recognition, language research, higher mental functions

FIGURE 1. TASK AREAS OF THE STANFORD ARTIFICIAL INTELLIGENCE PROJECT

fundamental discoveries yet to be made.

4. Nevertheless, progress in identifying and duplicating intellectual mechanisms is being made and the range of problems that computers can be made to solve is increasing.

5. Many blind alleys have been and are being followed and many mistakes are being made. Nevertheless, a body of fundamental knowledge is accumulating.

6. An important limitation up to the present has been the lack of well-trained and well-motivated scientists, working on artificial intelligence. This situation is improving.

7. Although full solution of the artificial intelligence problem is unpredictably far off, the understanding so far achieved has important potential practical applications. The development of these applications is worth undertaking to pay back society for the support given so far, to confirm the utility of the ideas on which they are based, and to increase the confidence of the government in supporting the research.

We have divided our work in artificial intelligence into four categories: epistemology, heuristics, interaction with the physical world, and man-machine interaction.

The identification of the epistemological and heuristic parts of the artificial intelligence problems as separate entities which can be studied separately to a substantial extent is a result of work of the last few years, mainly by McCarthy. The epistemological part is to choose a suitable representation for situations and the rules that describe how situations change. This description must be general enough to cover all problem solving situations, and even more important, it must be able to express all likely states of knowledge of the situation and the rules by which it changes spontaneously or by the actions of the problem solver.

The present state of the epistemological problem is described in (McCarthy and Hayes 1969). At present, work on the epistemological problem is mainly theoretical and is carried on by McCarthy.

The heuristic part of the artifical intelligence problem is to devise methods that, starting from a suitable representation of the information, will express explicitly the actions that have to be performed, e.g., find the right chess move or the right next step in a proof. Work in heuristics has been carried on in many places for a number of years. Here, Samuel is leading work applying learning . methods and Luckham is leading work in theorem proving by computer. This work is expected to continue more or less at its present pace.

Much of our work involves man-machine interaction in one way or another, but the most concentrated attack on this is through Feigenbaum's and Lederberg's Heuristic Dendral Project which involves getting chemists to express their rules of reasonableness for the structure of organic compounds in a way that can be used by the computer program.

The largest area of work of the Stanford Artificial Intelligence Project, both in terms of people and money, has been computer interaction with the physical world.

Frankly, this work has gone slower than was anticipated when the project started and has been more expensive than first anticipated, There are two reasons for this. First, it was more difficult, expensive, and time-consuming than expected to get a time-shared facility capable of both real-time and ordinary time-sharing. This task is still not complete, but all except for some reliability problems appear to be solved. (Basically, the problem is that the PDP-6 computer memories have not proved reliable at the required data rates on the memory bus. These data rates are within the specifications of the machine, but are greater than the data rates at which the machine was originally de-bugged.)

The second problem is that we have been slow in developing a group of good computer scientists whose main interest is the vision problem. Our own students have been slow to face the equipment and other difficulties involved in getting started in this area. However, the situation is improving, and we are beginning to get a group of people at an adequate scientific level whose main scientific interests are in this area.

Another project is the computer controlled vehicle. Our work-in this area comes from the idea that while full solution of the artificial intelligence problem is a long way off, some practical applications can be developed based on present knowledge. We hope that within five years we can bring a computer controlled automobile to the pseudo-practical level. By this we mean that a computer controlled car will drive from our lab to Bayshore Freeway paying proper attention to the road, other cars, pedestrians, animals and traffic signals but with low reliability. At this point it should be possible to get others to take over the development.

Besides artificial intelligence our group has worked in two other areas: mathematical theory of computation and symbolic computation. In the former area considerable results have been obtained by McCarthy and his students and more recently by Manna and Pnueli. The Stanford work in this area has been further strengthened by the addition of Robert Floyd and Donald Knuth to the faculty. This is resulting in a large bulge in the costs in this area to the ARPA project, but Floyd and Knuth expect to receive separate support for their work in the near future.

Symbolic computation work has been mainly carried out here by Anthony Hearn of the physics department. With the aid of the ARPA project in interactive computer support and research assistance, he has developed a package called REDUCE for general purpose manipulation (especially simplification) of algebraic expressions. This package is getting widespread use by physicists and engineers.

REFERENCE

J. McCarthy and P. Hayes, "Some Philosophical Problems from the Standpoint of Artificial Intelligence", D. Michie (Ed.), Machine Intelligence 5, University of Edinburgh Press, 1969.

Hand-Eye (Jerome Feldman)

In its early development, Artificial Intelligience Research was concentrated on very limited problem areas. Over the past few y ars, the Stanford Artificial Intelligience group has been attempting to expand the range of study to problems which are much closer to reality. The keystone of this effort has been the attempt to produce a coordinated hand-eye system capable of performing a variety of complex manipulation tasks. Many potential applications of such a system have been described in previous proposals and this aspect of the project needs no further emphasis.

The overall goal of the hand-eye project is to design and implement a system which exhibits interesting perceptual-motor behavior. An important subgoal is that the problems that arise in the design of system components be solved in ways which are sufficiently general to be scientifically interesting. Thus, for example, we have put considerable effort into understanding depth perception although the special environment we are using allows for ad hoc solutions. The possible applications of our work and its relevance to the study of animal behavior are not primary areas of interest.

Our first hand-eye system used many ad hoc solutions and was mainly concerned with the problems of combining the minimum necessary hardware and software components. This primitive, but complete system for block-stacking under visual control was completed in May, 1967 and has been described elsewhere [25]. The functional diagram of Figure 1 provides a sufficient description for our purposes. Our most recent

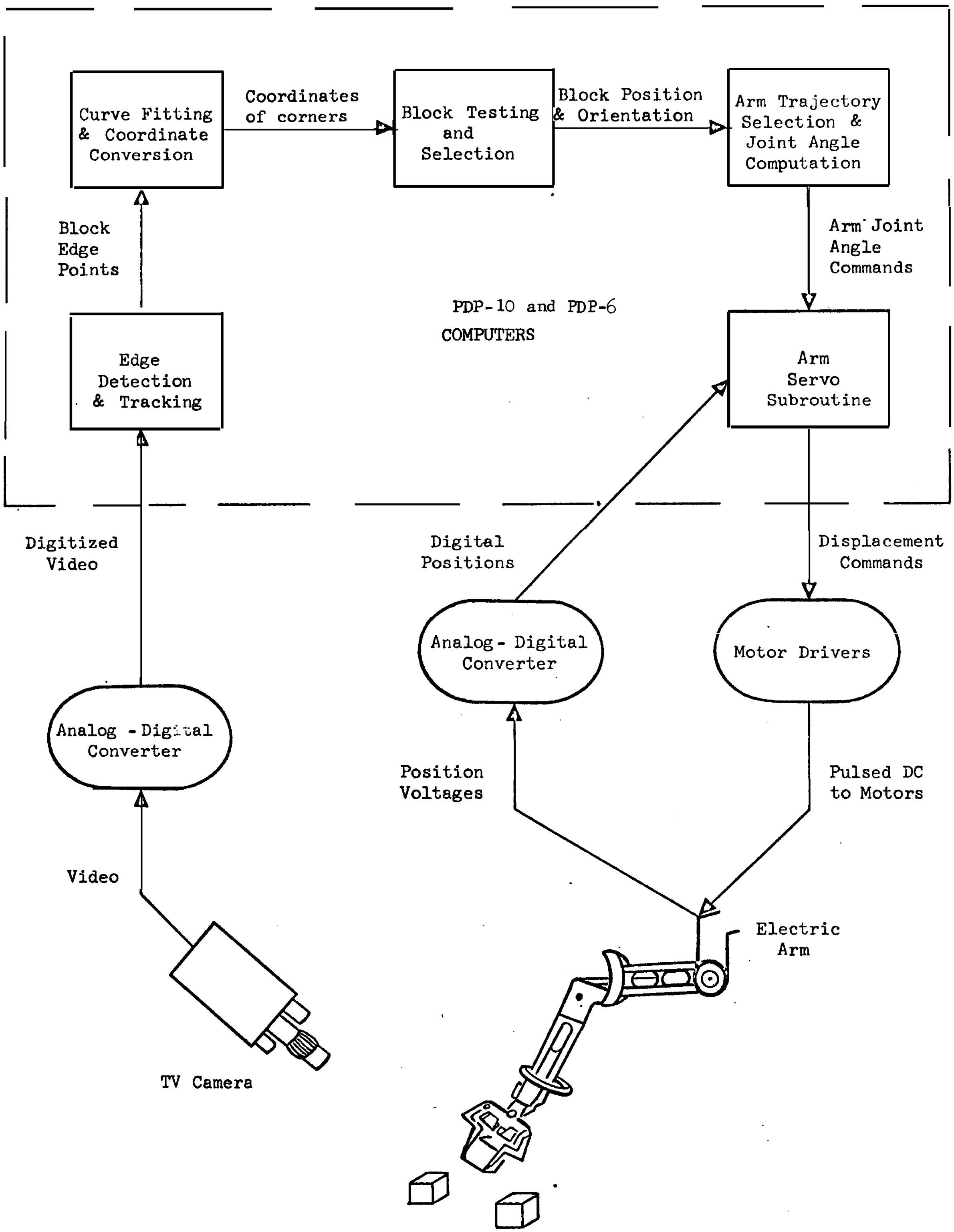


Figure 1. The Initial Block-Stacking System

work has involved the redesign of the system configuration and more careful study of each of the component programs.

Our attempt to develop an integrated hand-eye system has forced us to confront several AI problems which had received little previous attention. The two causes underlying the new problems are the complexity of the desired behavior and the innate perversity of inanimate objects.

Pattern recognition, problem solving, modeling, etc. which have been studied in idealized contexts take on new aspects in the hand-eye system. The most striking result to date is that traditional approaches to these problems have not proved adequate. We are not yet in a position to make definitive statements on what is needed, but a common understanding of the issues is arising among workers in the field.

The main principle which has emerged from the Stanford work is the dependence of everything on everything. For example, one might use entirely different perceptual strategies with a random access (image dissector) camera than with a scanning (vidicon) device. This inseparability contributes to the high entrance cost of hand-eye research; there is, as yet, no way to experiment with a part of the program without detailed knowledge of the other parts. Much of our effort has gone towards reconciling this mutual interdependence of programs with the inherent independence of programmers. The problem is exacerbated at a university by the need of graduate students to produce clearly separable contributions to the project.

These facts, plus the availability of systems-oriented students, encouraged us to undertake a rather ambitious system-programming project including a submonitor, a high-level language, and a new data structure. The goal of this project is to produce a hand-eye laboratory in which it will be relatively easy to experiment with new ideas in perception, modeling, problem-solving and control. This laboratory will also, hopefully, provide a testing ground for many related artificial intelligence projects.

The hand-eye laboratory will have to **accomodate** programs whose total size is several times the-size of core memory. Further, as we will show below ., the order in which these programs are executed cannot be determined in advance. These programs must be able to communicate with each other and with a common global model which represents the system's knowledge of the world. Since many operations require moving physical devices (like the arm and camera) which entail long delays, we would like to allow parallel execution of hand-eye subprograms. All of these requirements can be met by the addition of one basic feature, the pseudo-teletype, to the PDP-10 time-sharing monitor, A pseudo-teletype is simply a buffer set up by one job which acts as the control console of another job. Subprograms are each set-up as a separate job; all active, jobs will be automatically time-shared by the main monitor, The submonitor is responsible for handling messages, some interrupts and changes to the global model and will also be able to record its actions as an aid to debugging the system.

The language and data-structure designs are closely tied to the submonitor and to each other. The language is an extension of our ALGOL Compiler [33] along the lines of the associative language, LEAP [6].

The central concept of LEAP and the underlying data structure is the association: **attribute·object** = value. The use of associations for world-modeling is described in detail in [19]. An important new concept of this version of LEAP is the use of local and global associative structures. Every atomic object (item) is either local or global; the associative structure local to a subprogram may contain associations including global items, but not vice-versa. Any attempt to alter the global associative structure is trapped to the submonitor which determines when the alteration should be allowed. The language will contain primitives for local and global associations, message handling and interrupt processing.

Work on the hand-eye problem, proper continues in parallel with the system development. Much of this work has been directed toward the development of a flexible set of vision programs, the subject of the next section. To provideasense ofdirection and to bound our aspirations, we proposed a class of tasks which we hope to have the hand-eye perform. The main task is the building of fairly complex constructions (castles) out of simple blocks. The blocks were restricted to being plane-bounded and convex. The castle might be explicitly described by a setofassociations relating its sub-parts orwemight simply be given one or more views of it. Even this task is too difficult for the system to solve in general, but it has provided a useful context for the development of various routines.

Building a castle out of children's blocks is a problem in which there is no technical literature. Shapiro [30] has concerned himself with the development of optimal strategies for doing this with our mechanical hand which can only place a block with 1/4 inch accuracy. The first problems attacked were the development of heuristics for stability analysis and for generating the proper sequence of actions assuming ideal placement. Subsequent work will remove this restriction and attempt to develop strategies which compensate for observed imperfections in the performance of the mechanical manipulator. One of the most interesting aspects of this task is the various levels of feedback which can be used in the building process. In some cases, one need only know that a block is still in place and tactile feedback is sufficient. If the situation is more critical one might visually determine the placement error and alter the remainder of the strategy accordingly. Finally, there is the possibility of adjusting the block, under visual control, until the error is sufficiently small [36]. An important part of,the castle building problem has been solved by Pieper [21] in his development of an obstacle avoidance program for the arm.

The use of visual feedback in block stacking presents a rather different problem than those normally discussed in picture processing. The vision routine has the job of determining the accuracy with which some block was placed. The total scene may be very complicated and it would be absurd to perform a complete scene analysis. Furthermore, the properties of the blocks to be examined may be known in great detail and the vision routine would be able to take advantage of this fact. One of our major efforts has been directed toward solving these problems of context-sensitive visual perception. The overall system designed to do this is quite complex and is the subject of the next section.

The Organization of a Visual Perception System

Perception, and most particularly visual perception, is a complex process requiring a system which is sensitive to all the various levels of detail of the environment. Furthermore, since the available data is potentially overwhelming (consider the number of different viewpoints) the system must have both the mechanisms and appropriate strategies to select what data are worthy of its attention and what level of detail is best suited to the current perceptual goal.

We will concentrate on these two aspects of visual perception -- levels of detail and strategies for attention. Data from a scene may be structured to varying degrees. At the lowest level lie the intensity and color of the light at a particular point in the visual field; at a higher level are those objects in the visual scene which we dignify by the use of nouns; at a still higher level one notices interrelationships and relative motion between objects. At the highest level one is aware of the total situation -- as "Danger., Collision imminent." Each of these levels of perception is necessary and we must integrate all of them. Ordinarily, we are conscious only of our perceptions of objects and situations, but the fact that we can learn to draw-indicates that lower level details are perceived and can be made accessible to consciousness.

It is curious that we must learn to draw -- as if the lower levels of visual patterns are coalesced into objects at a preconscious level. This notion gives rise to a simplified theory of perception held by many workers in perception and pattern recognition. The theory is embodied in a strategy of perception which places attention first at the lowest level of detail and then extracts successively higher levels until the organization of the entire scene is understood. Thus by processing intensity and color distributions one obtains texture, edges, and corners. From this information regions are extracted and these in turn are associated into bodies. Then the bodies are identified as objects and their various interrelationships are derived. Thus:

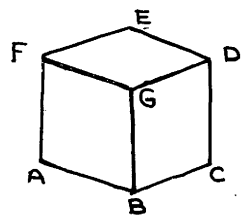
points → lines → regions → bodies → objects → scene

Essentially, all the early work on visual perception, including our own, proceeded along these lines. To some extent, the beautiful work of Guzman [11] on finding the distinct bodies in a perfect line drawing had an undesirable effect on the field. Guzman's program was so successful that it sent people on a quest for the perfect line drawing program. Although we have had considerable success [13, 24] at generating line-drawings, it has become apparent that the strict bottom-to-top processing sequence is not optimal, We will present some general discussion on the organization of visions systems and then describe our proposed approach.

The model of visions which we find useful involves routines at various levels cooperating in an attempt to understand a scene, There is a large body of psychological evidence [9] indicating the dependence of perception upon global information and upon preconceived ideas. Many of the well known optical illusions fall in this class. One can also show that there are simple scenes which are ambiguous in the absence of global information, but are easily resolved in context.

A most striking case of this is the ground plane assumption [27], which has become a cornerstone of all robot perceptual systems. From a monocular image it is impossible, in general to calculate the distance of an object from the camera. If, however, the object is lying on a known plane (one whose transformation to image coordinates is available) then the depth of the object's base vertices is known. This particular piece of global information has been implicity used for depth information, but has many other uses.

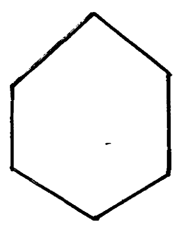
Consider the following line drawing.



If one knew that this object were lying on the plane determined by ABC which is known, then one would know the projection of each point in the image onto the ABC plane. Each point e.g. F must be on the line determined by its projection onto the ABC plane and the lens center. If the line AF is perpendicular to the plane we then know the length of AF.

Further, we can often determine whether or not AF is perpendicular to the plane from the information available. The lens center, point A and the projection of point F determine a plane, which contains the line AF, If this plane is perpendicular to ABC then the line AF is also, for objects which are at all regular [26]. If one knew the length of AF, BG and CD are computable and assuming F, G, D and E are in a plane is sufficient to determine E. More technically, the assumptions we have made allow only one degree of freedom in the choice of plane-bounded convex objects which could yield this image. Thus, the ground plane hypothesis plus some global regularity conditions allow for the complete description of an object from a single monocular view. Of course, these conditions may not hold, but Falk has some encouraging preliminary results in object recognition using these kinds of techniques.

A somewhat more basic problem arises in the consideration of the following image:



which might have come from, among other things:

The interior edges might very well be less distinct and be missed by the program which first tried to form a line drawing. At some higher perceptual level, a program could detect the ambiguity and attempt to find the interior edges. With the contextual information available, the system could then use highly specialized tests to determine the presence of an edge. Further, since the area involved is relatively small, it might also be reasonable to apply very sensitive general operations which are too costly *to* use on an entire scene. In both cases we see how our system organization facilitates a perceptual strategy involving selective attention. A vision system which worked strictly bottom-to-top would have no notion of attention. There would be a standard line finding operation, followed by an attempt to fit inter sections, etc. These are inherent limitations [28] in any such system balancing noise sensitivity with ability to perceive detail. The flexible organization discussed here allows for the use of different hardware and software components in different contexts and has much greater potential.

Those readers unfamiliar with the field will probably feel that we have set up an elaborate straw man. Cognoscenti will recognize the man as real enough, but will look for a way to make our grand design operational. The remainder of this section will be devoted to a discussion of how we are attempting to do this.

The goal, once again, is to produce a flexible visual perception system capable of selective attention and of integrating information from all levels of perception. An obvious prerequisite for such a system is a monitor, language, and data structure capable of its support. Our proposed design was described in Section 1.

A second necessary ingredient of any system is a large set of flexible basic vision routines. Among the necessary functions are: reading raw data, changing the camera position and parameters, edge finding, corner fitting, region finding, analysis into distinct bodies, identification of particular objects, and complete scene analysis. Work is proposed in all these areas but we will be content to describe briefly some of the work which seems to be most interesting.

One important aspect of the general vision system is accommodation, the adaptation of the input mechanisms to the visual environment. Selective attention can then be implemented in the vision hardware by choosing accommodative strategies which reflect current perceptual goals. For example, the camera could be sensitized to a specific color characteristic of a desired object (via a color filter). This effects a gross reduction in the volume of information which must be input and subsequently searched to determine its relevance.

The camera parameters currently under computer control are the pan and tilt angles, focus, magnification and digitization level. There are two hard problems in accommodation which arise from the need for a common world model. When the camera is panned, it gets a new view. The images of objects in this new view must be placed in correspondence with the old images of the same objects. An even more difficult problem is to compute accurately the perspective transformation [27] applicable in the new situation. Sobel [32] is developing techniques for these problems, relying heavily on the literature of photogrammetry.

A major area of interest has been the development of low-level edge and line finders. The visual system of the original system was little more than a good edge follower plus a routine--which used the ground plane assumption and the existence of only cubes to locate objects. There have been extensive analytical and practical studies of various spatial filtering and edge finding techniques [13, 34]. More recently, we have begun to look at feature verifiers which will use global information and a prediction to help identify a feature.

There are also programs which do fairly well at corner finding, region extraction, etc. These are fairly flexible and might be incorporated into a vision system organized as we have suggested. The real problem is to develop routines for these tasks which are sensitive to possible errors and ambiguities and know when to ask for help. A related issue is the language for communicating between vision programs at various levels. We have just begun to seriously confront these issues.

We are currently completing an interactive version of our grandiose vision scheme. Grape is extending his programs to allow for user intervention at several stages in the scene analysis process. As intermediate stages of analysis are displayed, the user will be able to interrupt and add information to the system. Using this system and some hard thought, we hope to come up with a reasonable first cut at the multi-level vision system. The process of refining this system and adding to its basic capabilities will be a long cumulative process.

The vision system described here will also be useful in connection with the mobile cart project. Other aspects of the Stanford Artificial Intelligience effort will also interact with the hand-eye project. The work on language studies, data structures and problem solving have immediate applications. From the speech work, we are developing ideas on how to handle properties, like texture, which have no obvious Canonical form. These interactions, plus the inherent interest of the problem, make the hand-eye work a central part of our effort.

REFERENCES

1. Becker, J., "The Modeling of Simple Analogic and Inductive Processes in a Semantic Memory System:, A.I. Memo forthcoming. Stanford University, Stanford, California, January 1969.

2. Buchanan, B., and G. Sutherland, "Heuristic Dendral: A Program for Generating Hypotheses in Organic Chemistry" in D. Michie, ed. Machine Intelligence 4, American Elsevier, 1969.

3. Colby, K., and Enea, H., "heuristic Methods for Computer Understanding of Natural Language in Context-Restricted On-Line Dialogues," Mathematical Biosciences 1, 1-25 (1967).

4. Earnest, L.D. (1967). 'On Choosing an Eye for a Computer'. AI Memo No. 51, Stanford University, Stanford, California.

5. Ernst, H.A (1961). 'MH-1 a Computer Operated Mechanical Hand'. Doctoral Thesis, M.I.T., Cambridge, Massachusetts.

6. Feldman, J., and Rovner, P.D., "An Algol-based Associative Language", AI Memo No. 66, Stanford University, Stanford, California, to appear in Comm. ACM.

7. Feldman, J., "First Thoughts on Grammatical Inference", AI Memo No. 55, Stanford University, Stanford, California.

8. Floyd, R., "Non-deterministic Algorithms:, J.ACM 14, (October 1967).

9. Gibson, J., The Senses Considered as Perceptual Systems, Boston, Houghton-Mifflin, 1966.

10. Green, C., "Theorem Proving by Resolution as a Basis for Question Answering Systems:, Machine Intelligence 4, American Elsevier, 1969.

11. Guzman,-A., "Decomposition of a Visual Scene into Three-dimensional Bodies", Proj. FJCC, 1968, P. 291-304.

12. Guzman, A., (1967). 'Some Aspects of Pattern Recognition by Computer'. MAC-TR-37, Project MAC, M.I.T., Cambridge,'Massachusetts.

13. Hueckel, M., "Locating Edges in Pictures", forthcoming AI Memo, Stanford University, Stanford, California.

14. Kaplan, D.M., "Regular Expressions and the Equivalence of Programs", AI Memo No. 63, Stanford University, Stanford, California.

15. Manna, Z., "Properties of Programs and the First Order Predicate Calculus", J.ACM, 1969.

16. McCarthy, J., and Hayes, P., "Some Philosophical Problems from the Standpoint of Artificial Intelligence", AI Memo No. 73, Stanford University, Stanford, California, November 1968 (to appear in Machine Intelligence 4, American Elsevier, 1969).

17. Miller, W .F ., and Shaw, A., "A Picture Calculus" in Emerging Concept in Graphics, University of Illinois Press, 1968.

18. Nilsson, N., "The Stanford Research Institute Robto Project", Proc. 1st Inst. Cong. on Art. Int.

19. Paul, R., Falk, G., Feldman, J., "The Computer Representation of Simply Described Scenes", Proc. Illinois Graphics Conference, April, 1969.

20. Paul, R., "A Representation of a Tower of Plano-convex Objects", Artificial Intelligence Operating Note No. 41, Stanford University, Stanford, California, August 1968.

21. Pieper, D., "The Kinematics of Manipulators under Computer Control" AI Memo No. 73, Stanford University, Stanford, California, October 1968, (Ph.D. Dissertation of Mechanical Engineering).

22. Pingle, K., "Hand-eye Library File", Artificial Intelligence Operating Note No. 35, Stanford University, Stanford, California. August 1968.

23. Pingle, K., "A List Processing Language for Picture Processing". Artificial Intelligence Operating Note No. 33, Stanford University, Stanford, California.

24. Pingle, K., "Visual Perception by a Computer", Proc. Summer School on Automatic Interpretation and Classification of Images, Pisa, Italy, August 1968.

25. Pingle K., Singer, J.A., and Wichman, W.M. (1968). 'Computer Control of Mechanical Arm Through Visual Input'. Proc. IFIP Conference, Edinburgh, 1968.

26. Pohl, I. "Search Processes in Graphs", Stanford University Dissertation, forthcoming.

27. Roberts, L.G., (1963). 'Machine Perception of Three-Dimensional Solids'. Optical and Electra-Optical Processing of Information, MIT Press, Cambridge, Massachusetts.

28. Samuel, A., "Studies in Machine Learning Using the Game of Checkers", IBM Journal, November 1967.

29. Schank, R., and Tesler, L., "A Conceptual Parser for Natural Language ", AI Memo No. 76, Stanford University, Stanford, California, January 1969.

30. Shapiro, G., "Advanced Hand-Eye Manipulating", Internal Memo, Stanford AI Project, Stanford University, Stanford, California,

31. Scheinman, V., "Considerations in the Design of a Mechanical Arm" Internal Memo, Stanford AI Project, Stanford University, Stanford, California.

32. Sobel, I., forthcoming thesis on "Visual Accommodation in Macine Perception".

33. Swinehart, b., "Golgol III Reference Manual", Artificial Intelligence Operating Note No. 48, Stanford University, Stanford, California.

34. Tenenbaum, J., "An Integrated Visual Processing System", forthcoming.

35. Waterman, D., "Machine Learning of Heuristics", AI Memo No. 74, Ph.D. Dissertation, Stanford University, Stanford, California.

36. Wichman, W.H., "Use of Optical Feedback in the Computer Control of an Arm", Engineers Thesis, Stanford University, Stanford, California, August 1967.

37. Thompson, M., "Manual of Photogrammetry", 3rd Edition, 1966.

38. McCarthy, Earnest, Reddy, and Vicens. 'A Computer with Hands, Eyes , and Ears,' Proc of FJCC '68 (1968).

Machine Learning (Arthur Samuel)

Enough experience has now been gained in the use of signature tables as a machine learning technique [ ] to justify a serious attempt to apply this technique to problems other than checkers. The work on machine learning is therefore currently being expanded both in scope and in intensity.

Work on the game of checkers is still continuing as this 'game provides a restricted field in which meaningful experiments can be formulated without the complicated effects of a less well structured environment. The playing portion of the checker program has been completely rewritten for the PDP/10 and is now thought to be reasonably bug free. The program is a good deal more complicated than was the version on the 7094 and it trades memory space for an increase in playing speed to more than compensate for the difference in machine speeds. Even without benefit of a satisfactory learning program it plays quite well. The le-arning portion of the program is currently being extensively modified and is not completely debugged.

Work is also continuing on the game of Go which offers another formal system but one in which the look-ahead procedure (so useful in checkers and chess) cannot usually be applied. This work is continuing at a slow pace with but one graduate student involved. The difficulties of the game and lack of knowledgeable players makes it hard to recruit people for this work. Go does, however, present a quite different environment from checkers and chess and is therefore well worth study.

The first serious non-game application being made of these machine learning principles is in the field of speech recognition. Messrs. Samuel and Reddy are collaborating in this work with the help of several people. Learning techniques are to be applied initially to the phoneme identification portion of the problem with their use in segmentation and in word recognition to be undertaken later.

REFERENCE

1. A. Samuel, "Studies in Machine Learning Using the Game of Checkers", IBM Journal, Nov. 1967.

Automatic Deduction (David **Luckham**)

A theorem-proving program based on the Resolution Principle [1] for First Order Logic has been implemented using the PDP-10 LISP system. This program incorporates all of the efficiency strategies for proof search that are the subject of references [2,3,4,5,6], and has been used to compare the relative merits of various combinations of these strategies. In addition, it also has a special replacement rule for equality, [7]. The program is capable of dealing with all of the basic theorems of elementary algebra (Theory of Groups, Rings, Fields, Boolean Algebras and Modular Lattice Theory) and number theory that have been proved by programs discussed in the above references, and also with simple questions that arise in information retrieval, question answering and automatic program writing [8]. A rudimentary interactive system has been incorporated into the program to allow the user to question and make suggestions to the program in the course of a proof search.

It is proposed to continue research in this area along both theoretical. and practical lines towards the goal of developing this program into a practicable basis for automatic mathematics systems (proof checking and proposition testing), program writing, information retrieval, and other projects in Artificial Intelligence. Specific lines of research will include the following: (1) the further development of the interactive feature, (2) the addition of a proof-tree analyser for extracting solutions to existential statements from proofs of them, (this is a basic part of the program writing and information retrieval applications), (3) the extension of the efficiency strategies mentioned above to a proof system incorporating rules of inference for the basic relations of equality, associativity, commutativity, etc., (4) the use of the algebraic symplification system, Reduce [9] in some of the phases of the proof search procedure, (5) the incorporation of decision procedures for certain classes of problem, (6) the extension of the basic logical system to a many-sorted calculus, and possibly the introduction of modal logic and w-order logic.

At the present time, this is an area of intense endeavor and, using the facilities available here, we propose to test each theoretical possibility on practical problems.

REFERENCES

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

2. Luckham, D., "Some Tree-Paring Strategies for Theorem-Proving", Machine Intelligence 3, D. Michie (ed), Edinburgh University Press, pp. 95-112, 1968.

3. Luckham, D., "Refinement Theorems in Resolution Theory", A.I. Memo 81, Stanford University, Stanford,California, (March 24 , 1969).

4. Wos, L., D. Carson, G. Robinson, "The Unit Preference Strategy in Theorem Proving", AFIPS Conf. Proc. 26, Washington, D.C., Spartan Books, pp. 615-621, 1964.

5. Wos, L., et. al., "Efficiency and Completeness of the Set of Support Strategy in Theorem Proving", JACM, Vol. 12, No. 4, pp. 536-541, October 1965.

6. Wos, L., et. al., "The Concept of Demodulation in Theorem-Proving", JACM, Vol. 14, No. 4, pp. 698-704

7. Robinson, G., and L. Wos, "Paramodulation and Theorem-Proving in First-Order Theories with Equality", Machine Intelligence IV, D. Michie (Ed), (to appear - 1969).

8. Green C., "Theorem-Proving by Resolution as a Basis for Question-Answering Systems", Machine Intelligence 4, D. Michie (Ed), (to appear 1969).

9. Hearn, A.E., "Reduce Users Manual", A.I. Memo, No. 50, April 1968.

DENDRAL PROJECT (Edward Feigenbaum)

The Heuristic DENDRAL Project is an effort to explore the processes of problem solving, hypothesis formation, and theory formation in scientific work. The particular task chosen for the exploration is: the determination of the molecular structure of organic molecules from mass spectral data.

In the interpretation of mass spectra of organic molecules, a chemist seeks a hypothesis about molecular structure that will serve to explain given empirical data produced by an instrument called a mass spectrometer. Heuristric DENDRAL is a LISP program that performs in this task environment. The overall structure of the program is a "scientific method" cycle, consisting of phases of preliminary inference making, systematic hypothesis generation, prediction from hypotheses, and test against empirical data with evaluation of goodness-of-fit.

The problem solving behavior of the program is remarkably good for the limited subset of organic molecules with which we have worked, in a task that is not a "toy" problem but a "real life" problem of considerable interest to science. For these molecules, the program's ability -is at least equal to, and usually better than, that of trained post-doctoralmass spectrum analysts.-

The complete state of the program as of October 1968 is documented in (4). Its status is an application of artificial intelligence research to theoretical chemistry is represented by the acceptance of two project papers in the Journal of the American Chemical Society (1,2).

Its status as a model of human judgmental process is represented by (6). Its potential contribution to the philosophy of science is given in (5). Its contribution to symbolic computation is represented by (3).

Chemists, bringing their own post-doctoral fellowships, will be working with the project in the next period.

Work to be Undertaken During the Period of the Proposal:

1. Application of A.I. to Chemical Inference

As an application to chemistry, Heuristic DENDRAL is open-ended (at least in relation to mass spectrum analysis and some closely allied problems). Increments of new chemical knowledge, formalized by the man-machine technique we have been using, produce beneficial increments in the problem solving capability of the program. This type of work -- of broadening and deepening the application to chemistry -- will continue, with the assistance of our chemist affiliates and post-doctoral fellows in chemistry. This will include greater concentration on ringed structures; adding many more organic "functional groups" to the analysis; bringing to bear additional chemical data (particularly infrared, ultra-violet, and N.M.R. spectra); improving the theory of molecular fragmentation used in the Predictor; and improving the goodness-of-fit evaluation criteria. We will also attempt to make the program available to the rest of the scientific community in LISP on the IBM 360.

2. The Problem of Representation for Problem Solving Systems

The problem of representation has been much discussed as a key problem of artificial intelligence research (7). We will be studying this problem in a number of different ways in connection with the various representations of the theory of fragmentation of organic molecules that are currently used by the Heuristic DENDRAL program. We will be attempting to write programs that will automatically alter the representation of chemical knowledge, from forms ill-adapted to a particular type of problem solving in the system (but perhaps more convenient for the input of knowledge by chemists) to forms more efficient in the problem solving. Initial experiments will involve programs that will move knowledge about fragmentation processes from the Predictor's form to the Preliminary Inference Maker's form; and programs that will re-represent the Predictor itself in a more convenient flexible form.

3. The Problem of Knowledge Assimilation

This problem is closely allied to the problem of representation. The Heuristic DENDRAL program has its knowledge (its "model") of mass spectrometric and other chemical processes represented in at least four different forms, each particularly well-suited to the performance of a specific type of inference process. But how can the system check the consistency of these various forms of the same knowledge? When a change is made to one of the representations, involving the input of new chemical knowledge at that point, can it be guaranteed that the proper changes will be made in the other representations to preserve consistency in the system's knowledge of chemistry? We have a way of attacking this problem that ~~involves some~~ of the experience to ~~be gained~~ in (2) ~~above,~~ and we plan ~~to devote~~ considerable effort in this period to working this out.

4. The Problem of Man-Machine Communication for Transfer of Knowledge

In the DENDRAL system, man-machine interaction is used not merely to debug and run the program but also as the primary vehicle for the communication of new knowledge about chemistry to the program. The problem of how to do this effectively is initmately bound up with (3) and (2) above. Chemists do not necessarily (or even usually) have their knowledge about the chemical world represented in the fashion most convenient for our various programs to use. Thus, automatic re-representation of knowledge from the "ground state" forms used by our chemists to more "finely tuned" forms for use by the program is essential.

The work of (3) above will allow us to experiment with more intelligent man-machine communication. In addition, to facilitate the transfer to knowledge via the man-machine interaction, we plan to develop a "problem-oriented" language for chemistry, oriented toward the problem of expressing chemical knowledge to the system and translating this knowledge to the appropriate internal representation.

5. <u>The Problem of Efficient Design</u>

How knowledge of the chemical world is employed by DENDRAL at the various points in its "scientific method" cycle effects markedly the efficiency of the problem solving process. Thus, some important questions of systems design-are raised, which must be carefully studied and understood -- a task that we propose to do. Since there is no epistemological theory which can serve as a guide, this understanding must be gained experimentally. For example, under what general conditions does it "pay" to apply knowledge in the DENDRAL hypothesis generation stage rather than in the hypothesis validation stage (and vice-versa)? Can, indeed, any general answers be given to system design questions of this sort?

Summary and Generalization:

A problem solving program has been written which exhibits a high level of performance on a complex scientific inference task, Using this performance system as a springboard, a series of additional program-writing,endeavors, addressing themselves to what we believe are key problems in artificial intelligence research, are proposed. These involve the problems of: representation; knowledge assimilation; man-'machine communication for transfer of knowledge; and efficient systems design. Though these problems are to be studied concretely in the context of chemical inference, they are clearly of considerable general interest. All of them, for example, are important in the development of canputer programs for the intelligent "management" of large data bases. The key questions being asked involve how to organize and 'manage" knowledge: how to do this most efficiently to control search, and how to get programs to do this automatically, in an effective manner, with minimal human intervention.

Project Publications (and publications in press):

1. J. Lederberg, G.L. Sutherland, B.G. Buchanan, E.A. Feigenbaum, A.V. Robertson, A.M. Duffield, and C. Djerassi, "Applications of Artificial Intelligence for Chemical Inference I. The Number of Possible Organic Compounds: Acyclic Structures Containing C,H,O and N". Journal of the American Chemical Society (in press).

2. A.M. Duffield, A.V. Robertson, C. Dherassi, G.G. Buchanan, G.L. Sutherland, E.A. Feigenbaum, and J. Lederberg. "Application of Artificial Intelligence for Chemical Inference II. Interpretation of Low Resolution Mass Spectra of Ketones". Journal of the American Chemical Society (in press)

3. Sutherland, G.G., Heuristic DENDRAL: "A Family of LIPS Programs". To appear in D. Bobrow (ed.), LISP Applications. Also, Stanford A.I. Project Memo No. 80.

4. Buchanan, B.G., G.L. Sutherland, and E.A. Feigenbaum. "Heuristic DENDRAL: A Program for Generating Explanatory Hypotheses in Organic Chemistry. "In D. Michie (Ed.), Machine Intelligence IV, University of Edinburgh Press, 1969 (in press). (Also Stanford A.I. Project Working Paper No. 62).

5. Churchman, C.W. and B.G. Buchanan. "On the Design of Inductive Systems: Some Philogophical Problems". British Journal for the Philosophy of Science, to appear Autumn 1969 (in press).

6. Lederberg, J. and E.A. Feigenbaum. "Mechanization of Inductive Inference in Organic Chemistry". In B. Kleinmuntz (Ed.) Formal Representations for Human Judgment, Wiley, 1968. (Also, Stanford A.I. Project Working Paper No. 54).

7. Feigenbaum, E.A., "Artificial Intelligence: Themes in the Second Decade". **Proceedings of the IFIP68 International Congress**, Edinbergh, August 1968 (in press). (Also, Stanford A.I. Project Working Paper No. 67).

Reports, unpublished or in preparation:

A. Feigenbaum, E.A., B.G. Buchanan, and G.L. Sutherland. Repbrt for the Case Western Reserve University Conference on Formal Systems and Non-Numeric Processing, November 1968 (in preparation).

B-W Lederberg, J, Report on DENDRAL Project for Accounts of Chemical Research (in preparation).

c. Lederberg, J. DENDRAL 64 - A system for computer construction, enumeration and notation or organic molecules as tree structures and cyclic graphs,

c.1. Part I, notational algorithm for tree structures, NASA CR-57029 and STAR N65-13158

c.2. Part II, Topology of cyclic graphs, NASA CR-68898 and STAR N66-14074

C.3. Part III, A general outline of the DENDRAL system, Systematics of organic molecules, graph topology and Hamilton circuits, NASA CR-68899 and STAR N66-14075

D. Sutherland, G.L. A Computer Program for Generating and Filtering Chemical Structures. Stanford A.I. Memo No. 49.

Computer-Controlled Vehicles (Lester Earnest)

We consider the computer control of cars and other vehicles to be an important problem that is ripe for research. This section outlines an approach to the problem, some reasons for following this approach rather than others, and the specific research that we plan to pursue.

There are a number of new transportation systems in various 'stages of planning and implementation. Some, such as the Bay Area Rapid Transit, seek to induce the commuter to give up his car in favor of a mass transit system that offers greater speed between terminals, but less convenience. Such systems may be considered plausible only where population densities are very high and even there, they are only partially successful. The problem is that these systems do not provide as good an approximation to timely point-to-point service as do cars, for most people. People are willing to tolerate substantial expense and parking nuisance in order to have service from their door.

One approach to an improved transportation system would be the employment of car-like vehicles together with external control or guidance devices, such as buried cables. Such systems could offer the existing conveniences of automobiles plus the advantages of automatic control. Not only could you be chauffeured to your destination but the car could be sent to a parking lot when you got there. Alternatively, the car could be dispatched to pick up things, given suitable loading facilities. Since automatic control systems can operate with substantially faster response times than humans, it should be possible to operate automatic cars at high speed with very close spacing so as to effectively double or triple the capacities of highways.

A disadvantage of buried cables or other external control components is the necessity of establishing signalling standards and making large investments before the new system can begin to operate. An attractive alternative would be to make each vehicle essentially autonomous through the use of an on-board computer with optical and 'other sensors to do steering and navigation, It is clear that if the technological problems in this approach can be solved, the economic problems will be manageable. Small computers already cost less than expensive cars and in the near future will be even less expensive.

We propose to explore the possibility of constructing autonomous vehicles. The key problem in developing such vehicles is the perception of steering and danger clues in real time. The approach that we plan to pursue involves two intitial steps: 1) Experimentation with a small electrically-powered cart tied to our existing computer facility by two-way radio links and 2) construction of a vehicle carrying a computer for (closely monitored) operation on actual streets and highways.

The electric cart is now in operation in our laboratory and on the road adjacent. It contains a television camera that transmits a picture to the computer, which in turn controls the steering, speed and camera orientation of the cart. The system has been programmed to follow a white line on the ground and successfully does so under good seeing conditions.

Our next immediate goal is to navigate the road around the laboratory using as visual clues the edge of the road and the rather obscure dotted line in the center. Following this, we shall attempt to detect and avoid obstacles of various kinds, both fixed and moving.

If this program is successful, we shall have a muchclearer understanding of the computer capacity and performance required to undertake a full-scale experiment. For practical reasons it will probably be desirable to use a small truck with automatic transmission.

Mathematical Theory of Computation (Zohar Manna)

The research in Mathematical Theory of Computation was concerned last year mainly with the problem of formalizing properties (termination, correctness and equivalence) of algorithms. The algorithms that have been investigated are: (a) abstract programs [1-4], (b) programs [5], and (c) recursively defined functions [6,7].

In the future we shall try to extend such formalisms to a wider class of algorithms (such as Algol programs and algorithms with parallel computations). Increased effort on these problems should lead us not only to new theoretical results, but closer to automatically debugging programs and writing programs.

REFERENCES

1. D.M. Kaplan, "A Formal Theory Concerning the Equivalence of Algorithms", A.I. Memo 59, Stanford University, May 6, 1968.

2. D.M. Kaplan, "The Formal Theoretic Analysis of Strong Equivalence for Elemental Programs", A.I. Memo 60, Stanford University, June 1968.

3. D.M. Kaplan, "Regular Expressions and the Equivalence of Programs", A.I. Memo 63, Stanford University, July, 1968.

4. Z. Manna-, "Properties of Programs and the First Order Predicate Calculus", JACM (April 1969).

5. Z. Manna, "Formalization of Properties of Programs", A.I. Memo 64, Stanford University, July, 1968. (to appear in the Journal of System and Computer Sciences, May 1969).

6. Z. Manna and A. Pnueli, "The Validity Problem of the 91-Function", A.I. Memo 68, Stanford University, August, 1968.

7. Z. Manna and A. Pnueli, "Formalization of Properties of Recursively Defined Functions", A.I. Memo 82, Stanford University, March 12, 1969

Perception Project (Raj Reddy)

Much of the work, to date, on visual perception by machines has centered around plane-bounded man-made objects. Unfortunately, less than 5 percent of naturally occurring objects in a visual scene are plane-bounded. Most objects do not have well defined edges. This raises the question of whether many of the presently used edge and 'object recognition techniques are likely to be useful and if not, what other perception techniques should we be looking at?

In the perception project we have been mainly looking at people in a scene. An overall look at the scene with special purpose people-detection operators determines the areas of interest. A profile recognizer determines some gross parameter about the person. Using automatic pan, tilt, and zoom we obtain a close-up view of the faces for further analysis.

Traditional edge detectors and region analyzers yield a great deal of spurious information. Therefore, we are using goal directed feature extractions which are not critically dependent on edge detection. The results obtained so far are sketchy and inconclusive leading us to be neither optimistic nor pessimistic at this point. Mike Kelly, a Ph.D. student in our department is primarily working on this problem.

Future Plans

Besides, people recognition, it seems desirable to consider the following problems which are not directly related or are not being considered by the hand-eye project. Several people have suggested the use of color, texture, and environmental constraints in visual perception by machine. However, there have been no explicit suggestions as to how to obtain and use these features. During the next year we hope to solve the following specific problems concerning color, texture and shadows of a scene.

Using three color filters obtains digitized images consisting of (toy) houses, cars, trees and rocks. Develop three-dimensional operators which permit picture segmentation using a similarity or homogenity criterion. The operator should be extendable to n parameters instead of three at each point. This would permit extension of the program to use depth, light intensity, and texture features.

Two dimensional Fourier transform of the whole scene has been suggested for obtaining texture parameters. Besides being computationally expensive, this cannot conveniently be used in conjunction with color, light intensity, and depth paramenters. What is needed is the concept of local texture, i.e. texture parameters at a given point, so that one may decide whether that point belongs to one object or another in the scene.

Local texture at any point can be determined by Fourier transform of a small neighborhood around the point (perhaps convolved with a Bell-shaped or a Cone-shaped weighting function). This would still be computationally expensive. An alternative is to count the number of local maxima and minima (at a certain distance apart) in successive neighborhoods around the point. We hope to develop a local texture determination program and use it with 3-D operators to separate regions and objects.

Papert has proposed that one can get rid of shadows by combining them with an adjacent region at a later stage in Region Analysis. We believe environmental constraints such as the approximate position of the lights are essential to perform such region combining.

Language Research (Jerome Feldman)

There are a number of problems relating to the automatic processing of natural language which are continuing to be of interest. This is divided into three major sub-areas: associative data structures, models of cognitive structures and grammatical inference.

We have been studying the problems of associative memory in conventional computers for several years. The most recent development is the ability to have several independent, parallel programs all sharing the same associative structure. We hope to study the problems of controlling access to a global structure in the context of hand-eye tasks. Another important topic to be studied is the addition of deductive inference capabilities to the associative retrieval mechanisms.

One of the most interesting and difficult problems in artificial intelligence is the modeling of human cognitive structures. We have developed such a model [ ] and are studying several problems in language processing and understanding with this model. The model is unique in that it uses the notion of consequence (temporal, causal, etc.) as a central element. We are developing theories of analogy, generalization over instances and the relation between perception and understanding using this model.

Work on grammatical inference continues to be fruitful. The theoretical work on decidability is complete [ ] and we are looking at questions of optimal learning and teaching strategies. Many of these results will be directly converted into program heuristics. We are also studying the extension of these techniques to other problems of generalization.

REFERENCES

1. Feldman, J., and Rovner, P., "An Algol-Based Associative Language", A.I. Memo 66, Stanford University, August 1968.

2. Feldman, J.A., "First Thoughts on Grammatical Inference", A.I. Memo 55, Stanford University, August 11, 1967

3. Feldman, J.A., J. Gips, J. Horning and S. Rider, "Grammatical Inference and Complexity", Computer Science Report 125, Stanford University, Stanford, California.

4. Becker, J.D., "The Modeling of Simple Analogic and Inductive Processes in a Semantic Memory System", A.I. Memo 77, Stanford University, January 20, 1969.

Speech Recognition Project (Raj Reddy)

At present the PDP-10 System can recognize limited sets of words or phrases (as many as 500 to 1000) or a few simple constructs made out of these. The time for recognition is around 2 to 15 seconds (depending on the size of the vocabulary) per word with an accuracy of 95 to 98 percent. The system has been used for voice control of ,a computer-controlled mechanical hand. Some relevant references are Reddy and Vicens (1968), McCarthy, Earnest, Reddy, and Vicens (1968), Reddy and Neely (1969).

18 Month Plan

We will attempt to solve the following problems during the next 18 months.

1. To recognize several speakers, at present, we have to train the system with all their voices. To recognize the speech of a random speaker without explicitly training the system with each word or phrase he is likely to utter requires some form of speaker identification and normalization. Previous attempts at Speaker Recognition have been disappointing. Cur approach will involve having the speaker speak a kernel set of sentences initially and then every time he begins to use the machine he would identify himself by saying "I am John McCarthy" or some such. The system will then modify the recognition algorithm to correspond to his "acoustic signature".

2. Language Design for Man-Machine Voice Communication. A language for speaking to machines should not be unwieldy and ambiguous like: English, nor should it be unnatural like a programming

language. We believe a Lisp-like quasi-functional notation might be more convenient for some applications, i.e., it is more desirable to say "add Alpha to Beta" rather than "Beta Equals Alpha Plus Beta Semicolon". The structure of spoken machine languages will vary from task to task. We hope to have a system in which the user will specify the language as a BNF grammar or some such. The system will then analyze the grammar for possible phonemic, word boundary, and syntactic ambiguities and suggest possible modifications. The system will also provide a skeleton recognizer which-the user can use to train the system to recognize his language for one or several speakers.

3. We hope to build a better phoneme recognizer using Signature-Table learning techniques (see Samuel).

4. We hope to have at least a partially working conversational computer system using speech input and output. Speech output from the computer will use some form of time-domain compressed speech. We expect that the user and the computer will be able to carry on a plausible (and perhaps irrelevant) conversation in a question-answering situation.

Five Year Plan

In the next five years we expect to have a processor of the speed of a PDP/10 servicing 3 to 6 terminals with speech input and output capability. Such a system would be used as:

1. An extra motor process as to hands and feet (possible applications: Aeronautics, Space, and Interactive graphis),

2. A fast response, and fast data rate motor process (when a real time control of devices by a computer is performed under

human supervision).

3. Natural mode of communication between man and machine. Some possibilities are day-to-day calculations (voice controlled desk calculator), information retrieval and question-answering systems, Dictation (of programs, letters, etc.), Medical Diagnosis, and Psychotherapy.

If, as it seems likely, a computer 5 to 10 times the speed of the PDP/10 becomes available for about the same cost as a PDP/10 in the next 5 to 10 years, then we believe it will be possible to extend the above system to service 30 to 50 terminals with voice input and output capability. Such a system should then be competitive with conventional input-output equipment.

REFERENCES

1. Reddy and Vicens (1968), "A Procedure for Segmentation of Connected Speech", J. Audio Engg. Soc., October, 1968.

2. McCarthy, Earnest, Reddy and Vicens (1968), "A Computer with Hands, Eyes, and Ears", Proc. of FJCC, 1968.

3. Reddy and Neely, "Contextual Analysis of Phonemes of English", Stanford AI Memo 79, Stanford University, Stanford, Calif.

PROGRAMMING MODELS AND THE
CONTROL OF COMPUTING SYSTEMS

This project is directed toward the development of theoretical models of programs and the utilization of these models to develop both the theory and practice of the control of computing systems. In our present research we have developed a graph program model and a programming language for expressing programs within that model. We have 1. investigated the logical properties of this model, 2. formalized the programming language, and 3. demonstrated the capability of the model for representing a large class of programs. We plan next to 1. develop a higher level language for translation into the graph program language, 2. investigate the resource allocation in executive control of computations on these graphs, and 3. investigate the feasibility of building hardware to implement the graph models.

The computation graph in various forms is the central idea in many current studies of programming. Most of these studies deal with interpretative models which are used to discuss quantitative-properties of programs such as determinacy, termination conditions, boundedness or queues, etc. or to calculate quantative properties such as path length -(computation time) and its variants, effectiveness of parallel processing, etc. Our work is concerned with interpretative models and their utilization in programming, controlling computations, and hardware/software implementations.

Additional work is being done on memory organization and memory hierarchies. Particular attention is being given to organization of data on secondary storage and the control mechanisms between primary and secondary storage. Also languages for the control of secondary storage are under development. Data management and large data base problems are perhaps the most critical problems in current computing systems. Before we shall be able to realize many of the benefits of the third generation, solutions to these problems must be found.

GRAPHIC PROCESSING: ANALYSIS, LANGUAGES, DATA STRUCTURES

Within the Computer Science Department structure, Professor William Miller, who also directs the Computing Group of the Stanford Linear Accelerator Center, has a project with National Science Foundation support for research in graphics.

Summary of Previous Work

The work accomplished to date includes four major areas:

1. Recognition (Publications 1, 2, 3, 4, 5, 8, 9, 10, 11, 12)

A Picture Description Language (PDL) was developed for both the recognition and generation of pictures. A Picture Calculus was developed which included PDL and various formal properties; within this formal system, pictures can be proven -equivalent or weakly equivalent. In addition, several formal properties relating to the power of PDL were developed.

SPDL (a subset of PDL) was implemented and successfully recognized and analyzed particle physics photographs (spark chamber photographs) with a significant reduction in the bookkeeping present in other recognition systems. This version of SPDL was goal directed by an input grammar and, hence, was not specifically intended for particle physics photographs; in fact, it may be used for the analysis of a wide class of photographs.

2. Generation (Publications 6, 7, 13)

An interactive version of PDL was implemented. This interactive version was used in studying the power and limitations of PDL for creating pictures. It illustrated the limitations predicted by Shaw (Pub. 9) and, at the same time, suggested various improvements which form part of the proposed research plan.

3. Data Structures (Publications 17, 18)

List and Plex structures were investigated in detail including representation, definition, garbage collection and implementation. Also, the dependence of efficient Plex processing on storage management was studied.

4. Primitive Recognizers (Publication 15)

Further studies of curvaturepoint recognizers and generators used in recognizing two-dimensional patterns was completed.

Facilities for Graphics Research

The machines available for graphics research at the Stanford Linear Accelerator Center are:

1. IBM 360/91
   a. IBM 2260 CRT displays
   b. IBM 2250 Graphic Terminals
2. GIF (Graphic Interpretation Facility)

   An IDIIOM (Information Displays, Inc., Input-Output Machine) system consisting of a CRT graphic terminal and a Varian 620/I computer. Hardware features of the display include four character sizes, four intensity levels, blinking, character rotation and programmable line structures. The input devices are a function keyboard, a lightpen and a character keyboard. The IDIIOM is also linked to the IBM 360/91.
3. Film Digitizers

Research Plan

The plan for research under this proposal is:

1. Continued development of the Picture Calculus in the areas of:

   a. Geometrical Relationships and Constraints

      This will add the flexibility of conditional recognition depending upon the geometrical relationships between various primitives or sub-pictures.

   b. Probabilistic Recognition

      Probabilistic recognition and arbitrary semantics at execution time will permit a recognition system to be adjusted for optimum performance in analysis accuracy. It may also permit an automatic parameter adjustment of a recognition system (i.e., learning).

   c. Composition Features

      Some interesting composition features are positioning requirements, scaling constraints of a particular display device and windowing (or zooming) for examination of the detail of a picture.

2. Study and Development of Graphic Meta-Systems

   A graphic meta-system is a system for describing and implementing graphical languages or graphical systems. Intellectually, it will contribute to a better understanding of graphical languages; practically, it will allow a class of graphical languages to be implemented faster and more economically.

   The goals of a graphic meta-system are: (a) to find description languages that represent a large class of graphical languages,

(b) to find machine representations of these descriptions, and

(c) to find useful implementations of the meta-system for the construction of these described graphical languages.

An initial example of a graphic meta-system has been investigated and described (Pub. 14). This system is currently being implemented and will permit an economical experimentation within a class of graphical languages of which PDL is an example. This system will also provide device independence for this class of graphical languages.

Publications: June 21, 1967 to June 30, 1969

1. W. F. Miller and Alan C. Shaw, "A Picture Calculus" Stanford Linear Accelerator Center Computation Group, GSG-40, June 21,1967.

2. Alan C. Shaw, 'A Picture Calculus - Further Definitions and Some Basic Theorems", Stanford Linear Accelerator Center Computation Group, GSG-46, June 1967.

3. R. Conny, W. Fridh, "An Attempt to Put Parts of Roberts' (Machine Perception of Three-Dimensional Solids) into the Framework of the Picture Calculus", Stanford Linear Accelerator Center Computation Group, CGTM-28, October 1967.

4. W. F. Miller and Alan C. Shaw, "Search Procedures in the Picture Calculus", Stanford Linear Accelerator Center Computation Group, GSG-51, October 1967.

5. W. F. Miller and Alan C. Shaw, "A Picture Calculus"', Stanford Linear Accelerator Center, October 1967 (Presented at the conference on "Emerging Concepts in Computer Graphics", University of Illinois, Urbana, Illinois, November 1967).

6. James E. George, "Picture Generation Using the Picture Calculus", Stanford Linear Accelerator Center Computation Group, GSG-50, December 1967.

7. J. E. George and W. F. Miller, "String Descriptions of Data for Display", Stanford Linear Accelerator Center, (Presented at the 9th Annual Symposium of the Society for Information Display, May 1968).

8. Ingrid Carlbom, "Algorithms for Transforming PDL-expressions into Standard Form and into a Primitive Connection Matrix", Stanford Linear Accelerator Center Computation Group, CGTM-38, February 1968.

9. Alan C. Shaw, "The Formal Description and Parsing of Pictures" , Stanford University, Computer Science Department, CS-94, April 5,1968.

10. Alan C. Shaw, "A Formal Picture Description Scheme as a Basis for Picture Processing Systems", Stanford Linear Accelerator Center, April 1968, (Submitted to Information and Control).

11. W. F. Miller and A. C. Shaw, "Linguistic Methods in Picture Processing -- A Survey", Stanford Linear Accelerator Center, August 1968 (Presented at the 1968 Fall Joint Computer Conference, December 1968).

12. Erik Sandewall, "Use of Ambiguity Logic in the Picture Description Language", Stanford Linear Accelerator Center Computation Group, GSG-52, December 1968.

13. James E. George, "CALGEN -- An Interactive Picture Calculus Generation System", Stanford University, Computer Science Department, CS-114, December 1968. .

14. James E. George, "The System Specification of GIAF: A Linear String Graphical Language Facility", Stanford Linear Accelerator Center Computation Group, GSG-61, February 1, 1969.

15. C. T. Zahn, "A Formal Description for Two-Dimensional Patterns'*, Stanford Linear Accelerator Center, March 1969, (To be presented at 'the International Joint Conference on Artificial Intelligence, May 1969).

16. W. F. Miller, "Hardware and Software for Effective Man-Machine Interaction in the Laboratory", Stanford Linear Accelerator Center, April 1969.

17. W. J. Hansen, "Compact List Representation: Definition, Garbage Collection, System Implementation'; accepted for publication in the Communications of the Association for Computing Machinery.

18. W. J. Hansen, "The Impact of Storage Management on Plex Processing Language Implementation", Computer Science Department Technical Report (press).

SLAC COMPUTATION GROUP

There are three major computer science research projects under way in the SLAC Computation Group. In addition, there are a number of practical development projects and one- or two-man projects. The three principal research projects are the Compiler-Compiler Project under the supervision of Professor David Gries, the Control of Computing Systems Project under the direction of Professor W. F. Miller, and the Pattern Recognition and Graphics Project under the direction of Professor W. F. Miller. A brief description of each of these three projects is given below.

1. The Compiler-Compiler Project. This term denotes the language (actually the set of languages) in which one writes a compiler. Most compilers are written in the assembly language of some machine. That is, the syntax and semantics are expressed in the assembly language. The assembly language program representing the compiler is then assembled, producing the machine language version of the compiler. Thus, the assembler should be called a "compiler-assembler", since it is used to assemble compilers. One does not call it a "compiler-assembler" because the assembly language is so close to the machine language that the two are considered to be equivalent, or the same. Unfortunately, such a compiler is very unwieldly; it takes a long time to write, it is not readable by most people, it is not a very natural description of the source language in question, and consequently is hard to change. Also, the compiler has then been written for one particular machine only.

What we want to do then, is to design and implement a language for writing compilers , just as ALGOL and FORTRAN have been designed and implemented for programming scientific algorithms. We want a compiler-compiler and not a compiler-assembler.

Our compiler writing system will be based mainly on Feldman's FSL system.

2. The Control of Computing Systems Project is concerned with models for computing systems, description languages for describing the control functions of computing systems, higher languages for programming executive control systems, and implementation techniques. The goals of the project are to develop an understanding of the complexity of real-time computing systems, to develop some techniques for guaranteeing the integrity and determinacy of competing sequential processes and to develop the facilities for generating and changing the systems built under the model developed.

There are two models being explored-in considerable depth. The first is the Dijkstra, hierarchy of control model. This model has been particularly designed to guarantee determinancy and avoid the circular-wait problem. It does not address itself to the time urgency aspect of real-time problem. The model is being investigated to see how real-time problems can be brought under its scope. The second model being explored is the computation graph model of Karp and Miller. A programming language has been developed which permits the display of the full parallelism present in a given algorithm. The emphasis here is that the programmer has the facility to display this parallelism. We do not automatically find it from some other representation of the algorithm. This model is directed toward the resource allocation problem and determinacy for parallel competing processes. The language as it now stands permits us to express a large number of important procedures necessary for programming a variety of problems. The local control has been specified and the logical soundness of the system has been proved.

We are also concerned with meta-languages for real-time data processing and executive control systems. We have practical work under way on the development of a multi-level data acquisition and data analysis system employing an IBM 1800 and the IBM 360/91. We also have a resource allocation control model and meta-language under development for implementing such practical real-time systems.

3. Pattern Recognition and Graphics Project is concerned with the development of picture-processing equipment, and graphic control systems. Rather sophisticated film digitizers and their control systems have been developed and reported in the literature. One of our main and most successful efforts is concerned with the development of picture grammars and the utilization of these grammars to direct the search of the field of pictures. This approach is called Picture Parsing. A rather efficient, and at the same time versatile, Picture Parsing Program has been written and applied to the problem of recognition of data boxes and events in particle physics pictures. This work has opened up a large number of other problems and opportunities for development in picture recognition. We intend to continue to explore the possibility of more sophisticated grammar systems for picture recognition.

We have developed a large number of heavily used graphic applications programs, such as visual histogramming, beam optics design, curve fitting, and text editing.

COMPUTATION GROUP TECHNICAL MEMOS

CGTM NO.

1. Pattern Recognition Techniques in Film Data Analysis, W. F. Miller, April 1965.

2. Computation and Control in Complex Experiments, W. F. Miller, May 1965.

3. Utility Routine Requirements for a General Time-Sharing System, W. F. Miller, R. T. Braden.

4. TRACK, V. Whitis, October 1965.

5. LAYOUT, S. Howry, October 1965.

6. Dynamic Syntax Directed Compilers, E. Burfine, November 1965.

7. The SLAC Program Library, J. Welsch, December 1965.

8. Determination of Linear and Quadratic Non-Homogeneity Terms from Magnetic Measurements of the 3^{o} Switchyard Bending Magnets, S. Howry, February 1966.

9. An Outline of the Basic Theory of Formal Languages, E. Burfine, June 1966.

10. Beam Switchyard Control Computer System Language, S. Howry, August 1966.

11. Simple-Minded Elementary List Language, J. Ehrman, December 1966.

12. (Adams) SRS, A Syntax Retrieval System, D. Adams, January 1967.

13. (Crason) Utilization of Available Memory on IBM 360 Computers with FORTRAN Programs, D. Carson, March 1967.

14. Concerning the Real-Time Computing Issue: Centralization vs. Decentralization, R. Braden, May 1967.

15. ALGOL to FORTRAN Translator, Wayne Wilner, June 1967.

16. Box Syntax -- A 2-Dimensional Metalanguage, Dan Ross, June 1967.

17. System/360 Assembler Language Programming, John Ehrman, August 1967.

18. A Multiple-Precision Floating-Point Arithmetic Package for System/360, John Ehrman, August 1967.

19. A Study of Arithmetic Type Conversion on the IBM System/360, John Welsch, July 1967.

20. A Study of the Round Operation on the IBM System/360, John Wlesch, August 1967.

CGTM
NO.

21. A Study of Swap Instructions for the IBM System/360, John Welsch, July 1967.

22. A Comparison of Code Densities on the 7090 and S/360 Computers, Hilma Mortell, August 1967.

23. Translation from B5500 Extended Algol to OS/360 Algol, Ira Pohl, September 1967.

24. Goodness-of-Fit of Curves, G. S. Watson, August 1967.

25. A Subset of GPAK, a Graphics Package for the 2250, Mary Anne Fisherkeller, October 1967.

26. VRL-1 Computer, Victor Lesser, October 1967.

27. PDP-8 I/O and Interrupt Programming System for Spark Chamber Data Logging Programs, Ed Mueller, October 1967.

28. An Attempt to Put Parts of Roberts' "Machine Perception of 3-Dimensional Solids" into Framework of Picture Calculus, Conny Fridh, October 1967.

29. The SLAC Central Computer, R. Braden and W. F. Miller, October 1967.

30. Algorithms for Performing Multiplication and Division with "Logical" Operands, John Ehrman, November 1967.

31. Implementation on PDP 1 of Subset of Picture Calculus, Yves Noyelle, November 1967.

32. Syntax of an Interactive Language, William E. Riddle, November 1967.

33. The SPIRES Scope Demonstration System, J. George, November 1967.

34. SARPSIS: Syntax Analyzer, Recognizer, Parser and Semantic Interpretation -System, J. George, November 1967.

35. Garbage Collection in Hash-Addressing Processors, J. Ehrman, January 1968.

36. Handling Arithmetic Error Conditions on the IBM System/360, J. Welsch January 1968.

37. A Multi-Level Computer Organization Designed to Separate Data-Accessing from the Computation, V. Lesser, January 1968.

38. Algorithms for Transforming PDL expressions into Standard Form and into a Primitive Connection Matrix, I. Carlbom, January 1968.

39. B5500 Extended Algol to Algol W Translation, E. Satterthwaite, February 1968.

40. Implementing a Graph Model of Parallel Computation, J. Levy, February 1968.

CGTM
NO.

41. Hidden Goodies in Fortran H, Model 1, Robinson, Ehrman, March 1968.

42. SMEAR 2 - A Pre-Processor for OS 360 JCL, R. Denham, March 1968.

43. Notes on Construction of Subsystems within Operating System/360, E. Satterthwaite, March 1968.

44. Scope Routine for TORTOS Simulator, W. Wilner, April 1968.

45. A Computation Model with Data-Sequenced Control, D. Adams, May 1968.

46. PL/1 → FORTRAN via INVOKE, C. Zahn, May 1968.

47. Some Notes on Branching Efficiency in Fortran, J. Ehrman, June 1968.

48. Some Examples of Optimizable Fortran Programs, J. Ehrman, June 1968.

49. Computers and Privacy: The Present and the Future, L. Hoffman, June 1968.

50. Conceivable Extensions to System/360 Assembler Language, J. Ehrman, June 1968.

51. Fortran H. Goodies Revisited, J. Ehrman, July 1968.

52. Comments on the 1800 Hardware, R. Russell, August 1968.

53. Dependence of Equality Axioms in Elementary Group Theory, G. Robinson, September 1967.

54. Report of the Share-Guide Assembler Project, J. Ehrman, August 1968.

55. An Algorithm for Finding Bridges -- and Its Extension, Ira Pohl, Sept. 1968.

56. Revised Report of the SHARE-GUIDE Assembler Project, J. Ehrman, January 1969.

57. Desk Calculator on Beam Switchyard Computer, S. Howry, January 1969.

58. PEG - On-Line Data Fitting Program, L. Smith, January 1969.

59. A Study of Attribute Notations in the IBM OS/360 Assembler Language, J. Ehrman, January 1969.

60. Formularies -- Program Controlled Privacy in Large Data Bases, Lance Hoffman, February 1969.

61. A Comparison of On-Line Computers, B. Russell, J. George, S. Levine, February 1969.

62. 1800/360 Link, R. Russell, March 1969.

63. Interrupt Systems, R. Russell, April 1969.

CGTM
NO.

64. SLAC Spiral Reader, M. Hu, March 1969.

65. Read-Write Store from a Read-Only Store, W. F. Miller, April 1969.

66. Q-type Address Constants, Dummy External Symbols, and Pseudo-Registers, J. Ehrman, April 1969.

100-106 Compiler Implementation System, D. Gries, November 1967 - March 1968.

COMPUTER SIMULATION OF BELIEF SYSTEMS

Kenneth M. Colby, M.D., who is a Senior Research Associate in the Computer Science Department, terminated his private practice of psychiatry to devote full time to investigations in this area of computer simulation. The National Institutes of Health sponsor two projects under Dr. Colby's direction. One of these is a Research Career Award and the other is a research project which continues the investigations in which his group has been engaged for the past seven years.

Research Plan

A. Introduction and Specific Aims:

The clinical problems of psychopathology and psychotherapy require further investigation since so little is known about their essential processes. Some of this ignorance stems from a lack at a basic science level of dependable knowledge regarding higher mental processes such as cognition and effect. The research of the project attempts to approach both the clinical and basic science problems from the viewpoint of information-processing models and computer simulation techniques. This viewpoint is exemplified by current work in the fields of cognitive theory, attitude change, belief systems, computer simulation and artificial intelligence.

The rationale of our approach to these clinical problems lies in a conceptualization of them as information-processing problems involving higher mental functions. Computer concepts and techniques are appropriate to this level of conceptualization. Their success in other sciences would lead one to expect they might be of aid in the areas of psychopathology and psychotherapy.

The specific aims of this project relate to a long-term goal of developing more satisfactory explicit theories and models of psychopathological processes. The models can then be experimented with in ways which cannot be carried out on actual patients. Knowledge gained in this manner can then be applied to clinical situations.

B. Methods of Procesure:

We have now gained considerable experience with methods for writing programs of two types. The first type of program represents a computer model of an individual person's belief system. We have constructed two versions of a model of an actual patient in psychotherapy and we are currently writing programs which simulate the belief systems of two normal individuals. We feel we need more experience in simulating normal belief systems before attempting further work on pathological belief systems. The second type of program represents an interviewing program which attempts to conduct an on-line dialogue intended to collect data regarding an individual's interpersonal relations. We have written two such interviewing programs and at present we are collaborating with two psychiatric residents in writing a program which can conduct a diagnostic psychiatric interview.

A computer model of a 'belief system consists of a large data-base and procedures for processing the information it contains. The data-base consists of concepts and beliefs organized in a structure which represents an individual's conceptualization of himself and other persons of importance to him in his life space. This data is collected from each individual informant by interviews. Validation of the model is also carried out in interviews in which the informant is asked to confirm or disconfirm the outcome of experiments on the particular model which represents his belief system. Because of the well-known effects of human interviewer bias, the process of data-collection and validation should ideally be carried out by on-line man-machine dialogues and this is a major reason for our attempt to write interviewing programs. However, the difficulties in machine utilization of natural language remain great and until this problem is reduced we must use human interviewers.

We have learned that the major defect in both belief system models and interviewing programs lies in the lack of a satisfactory inference system. Our current attempt is to formulate or implement a more powerful inference system which can be used to generate both new beliefs in a model and appropriate questions in an interview.

We have written one type of therapeutic interactive program which is designed to aid language development in nonspeaking autistic children. We have used it for the past year on fifteen children with considerable success (86% linguistic improvements). We intend to continue using this program and to instruct professionals in psychiatry and speech therapy in how to write, operate and improve such therapy programs for specific conditions.

c. Significance of this Research:

This research has significance for the psychiatry, behavioral and computer sciences.

Psychiatry lacks satisfactory classifications and explanations of psychopathology. We feel these problems should be conceptualized in terms of pathological belief systems. Data collection in psychiatry is performed by humans whose interactive effects are believed to account for a large percentage of the unreliability in psychiatric diagnosis. Diagnostic interviewing should ideally be conducted by computer programs. Finally, the process and mechanisms of psychotherapy are not well understood. Since experimentation on computer models is more feasible and controllable than experimentation on patients, this approach may contribute to our understanding of psychotherapy as an information-processing problem.

It is estimated that 90% of the data collected in the behavioral sciences is collected through interviews. Again, a great deal of the variance should be reduced by having consistent programs conduct interviews. Also, this research has significance for cognitive theory, attitude change and social psychology.

Computer science is concerned with problems of man-machine dialogue in natural language, with optimal memory organization and with the search problem in large data-structures. This research bears on these problems as well as on a crucial problem in artificial intelligence, i.e., inductive inference by intelligent machines.

D. Collaboration:

We are collaborating with two psychiatric centers for distrubed children and the Department of Speech Therapy at Stanford. We are also collaborating with residents in the Department of Psychiatry and with graduate students in computer science, psychology, education and electrical engineering.

Recent Publications

1. Colby, K. M. Experimental treatment of neurotic computer programs. Archives of General Psychiatry, 10, 220-227 (1964).

2. Colby, K. M. and Gilbert, J. P. Programming a computer model of neurosis. Journal of Mathematical Psychology. 1, 405-417 (1964).

3. Colby, K. M. Computer simulation of neurotic processes. In Computers in Biomedical Research, Vol. I, Stacey, R. W. and Waxman, B. Eds., Academic Press, New York (1965).

4. Colby, K. M., Watt, J. and Gilbert, J. P. A computer method of psychotherapy. Journal of Nervous and Mental Disease, 142, 148-152 (1966).

5. Colby, K. M. and Enea, H. Heuristic-methods for computer understanding in context-restricted on-line dialogues. Mathematical Biosciences, Vol. I, 1-25 (1967).

6. Colby, K. M. Computer simulation of change in personal belief systems. Behavioral Science, 12, 248-253 (1967).

7. Colby, K. M. A programmable theory of cognition and affect in individual personal belief systems. In Theories of Cognitive Consistency, Abelson, R., Aranson, E., McGuire, W., Newcomb, T., Tannebaum, P. Eds., Rand-McNally, New York (1967). In Press.

8. Colby, K. M. and Enea, H. Inductive inference by intelligent machines. Scientia (1967). In Press.

9. Tesler, L., Enea, H. and Colby, K. M. A directed graph representation for computer simulation of belief systems. Mathematical Biosciences, 1 (1967). In Press.

10. Colby, K. M. Computer-aided language development in nonspeaking autistic children. Technical Report, No. CS 85 (1967), Stanford Department of Computer Science.

11. Enea, H. MLISP. Technical Report, CS 92 (1968), Stanford Department of Computer Science.

12. Colby, K. M., Tesler, L., Enea, H., Search Experiments with the Data Base of Human Belief Structure, (paper to be delivered at International Joint Conference on Artificial Intelligence, Washington, D. C., May '69).

The Hewlett-Packard Control Computer

This project is concerned with the development of the Hewlett-Packard Control Computer for teaching and research in computer systems programming. It is used as a principal tool in teaching Computer Science 139 to enable students to become familiar with aspects of programming such as input-output, and interrupts that are not accessible to them in large batch processed computers. It is the only hands-on machine available to the students in the department. Students have occasionally used the computer as the vehicle for independent research projects and computer laboratory projects covering areas such as meta-compilers, assemblers, interactive desk-calculators, and computer graphics.

HSS:mjr

MATHEMATICAL PROGRAMMING LANGUAGE (MPL)

OPERATIONS RESEARCH

1. A new programming language called MPL has been under development, with Professor George B. Dantzig as Principal Investigator. The language is essentially standard matrix and set notation, and is the one used by most researchers in mathematical programming to express their algorithms prior to coding for the computers.

Research on definition of the language began in the Spring of 1967. Professor William Miller helped in the overall organization of the project. Professor David Gries has provided the computer science expertise to the group, laying out the detailed design of the language. The language has been stated in Backus Normal Form. Gries' Compiler-Compiler will be used to develop a machine language compiler for MPL. Chris Witzgall and Rudy Bayer of Boeing and Paul Davis of Union Carbide have contributed to MPL's definition. The graduate student research committee on the project currently are Michael McGrath, Steven Maier and Julie Herman. McGrath has recently completed a PL/1 translator for MPL that translates a code written in MPL into PL/1.

It-is expected that MPL will speed up the coding of methods for solving large-scale linear programs. Algol and Fortran have been too unreadable to permit easy writing and correction of complex programs.

2. Professor Dantzig was co-director and co-editor (with A. F. Veinott, Jr.) of the American Mathematical Society 1967 Summer Seminar on Applied Mathematics. The Society has recently published two volumes entitled

'Mathematics of the Decision Sciences' presented at the Seminar, including a Series of Lectures on Computer Science by William Miller and one by A. Taub.

3. Professor Dantzig is the principal investigator of the following projects:

Mathematical Models in Operations Research
and Computer Science
Sponsored by the Office of Naval Research

Research in Mathematical Biology
Sponsored by the Office of Naval Research
and National Institutes of Health

Optimization Theory
Sponsored by the National Science Foundation

Stochastic Mathematical Programs
Sponsored by the Atomic Energy Commission

Time Dependent Mathematical Programs
Sponsored by the Army Research Office

Mathematical Programming Language [MPL]
Sponsored by the National Science Foundation

Professors Richard W. Cottle and Alan S. Manne of the Operations Research Department are associated with Professor George Dantzig in the Mathematical Methods in Operations Research and Computer Science project. Professor Robert Wilson of the Graduate School of Business is associated with him on the Stochastic Mathematical Programs research project. Professor David Gries is associated with him in the recently approved Mathematical Programming Language project.

RESEARCH REPORTS

CS 119	Mathematical Programming Language	Bayer/Bigelow/Dantzig/Gries/McGrath/Pinsky/Schuck/Witzgall
CS 126	Complementary Spanning Trees	Dantzig
OR 66-1	Finding a Cycle in a Graph with Minimum Cost to Time Ratio with Applications to a Ship Routing Problem	Dantzig/Blattner/Rao
OR 66-2	All Shortest Routes from a Fixed Origin in a Graph	Dantzig/Blattner/Rao
OR 66-3	All Shortest Routes in a Graph	Dantzig
OR 67-1	On Positive Principal Minors	Dantzig
OR 67-2	Complementary Pivot Theory of Mathematical Programming	Cottle/Dantzig
OR 67-3	Approximate Algorithm for the Fixed Charge Capacitated Site Location	Jandy
OR 67-5	Duality in Discrete Programming	Balas
OR 67-6	Quadratic Forms Semi-Definite Over Convex Cones	Habetler/Cottle/Lemke
OR 67-7	Integral Extreme Points	Dantzig/Veinott
OR 67-8	Large-Scale Linear Programming	Dantzig
OR 68-1	The Chemical Equilibrium Problem	Bigelow
OR 68-2	The Hospital Admission Problem	Dantzig
OR 68-3	Selecting Different Dropping Variables in the Simplex Algorithm	Gordon
OR 68-4	Equivalent Integer Programs I: Basic Theory	Bradley
OR 68-5	A Dynamic Approach to the Development of Water Resources	Buras
OR 68-6	Equivalent Integer Programs II: The Special Problem	Bradley

OR 68-8	Domains of Positivity in R^n	Koecher/translated by Cottle
OR 68-9	A Generalization of the Linear Complementary Problem	Dantzig/Cottle
OR 69-1	Sparse Matrix Techniques in Two Mathematical Programming Codes	Dantzig/Harvey/McKnight/Smith
OR 69-3		Hopkins
OR 69-4		Eaves
CS 132	A Combinatorial m-step Conjecture	Dantzig

PARALLEL OPERATIONS IN DIGITAL NETWORKS AND SYSTEMS

Professor McCluskey holds a joint appointment in Electrical Engineering and Computer Science. His research activity is conducted organizationally in the Digital Systems Laboratory of the Stanford Electronics Laboratories; but he utilizes both Computer Science and Electrical Engineering graduate student resources.

In the research under his direction studies are being carried out to provide a unified theory of parallel operations implemented either by logic networks or programs. One objective is to try to weaken some of the restrictions imposed in order to guarantee "speed-independent" or "deterministic" performance.

RESEARCH IN PROGRAMMING LANGUAGES

In order to help find long-range solutions to the "software problem", Professors Robert Floyd and Donald Knuth propose to direct a series of investigations of important aspects of compiler writing and closely related subjects. A tentative outline of the contemplated research considers:

The major unknown factor in contemporary compiler design is the lack of reliable data about what users do with programming languages; it would be beneficial to know how frequently various constructions are used, in order to be able to make a rational choice between different compiler methods. It is felt that it should be possible to measure many of the critical parameters with reasonable accuracy. It should be possible to estimate, for example, how much time is spent in various kinds of lexical and syntactical analysis, as well as to determine how often various optimizable constructions occur(statistically and dynamically). This investigation would involve theoretical as well as empirical considerations.

While statistics like these are being gathered, it is also likely that some general aids, by means of which the users of a language can gather similar statistics about the running of their own programs, can be developed; moreover, these aids can be extended to some new kinds of debugging tools that may have a significant impact on the debugging process.

Typical areas of research which the investors plan to pursue are:

a) Development of theories relevant to code generation

b) Development of new algorithms for global optimization

c) Design of languages to facilitate real-time compiler writing (this includes "structured assembly programs" such as Wirth's PL360)

d) Design of two levels of intermediate languages: one approximately at a Polish notation level, one approximately at an "infinite register machine" level, both suitably extensible that they can adapt to a variety of source languages, and suitably clean that they can support research of classes a) and b) above.

e) Design and development of experimental programming systems

RESEARCH IN THE ANALYSIS OF ALGORITHMS

Professors Floyd and Knuth also consider "analysis of algorithms' a field of study directed to an understanding of the behavior of particular algorithms. Two kinds of problems are usually investigated.

A. Quantitative analysis of an algorithm. In this case the goal is usually to determine the running time and/or memory space requirements of a given algorithm. The determination of running time can be done in an essentially machine-independent manner by expressing the algorithm in some machine-independent language (not necessarily a formal language) and counting the number of times each step is executed.

B. Determination of "optimal" algorithms. In this case the goal is usually to find the "best possible" algorithm in a given class of algorithms. We set up some definition of "best possible" which reflects, as realistically as possible, the pertinent characteristics of the hardware which is to be associated with the algorithm.

The following list of research topics typifies the investigations into algorithmic analysis that are contemplated:

1) Study of the solution of special types of recurrence relations, both in closed form and from an asymptotic viewpoint.

2) Type A analyses of important algorithms for storage allocation, arithmetic, sorting, information retrieval, language analysis, scheduling, etc., for the many cases where published analyses are incomplete.

3) Further work on Type B analysis for several unsolved problems.

4) Extensions to programming languages and compilers intended to facilitate making empirical analyses of algorithms; and to facilitate making use of theoretical analyses by computing the running time when approximate parameters are supplied.

5) Development of a "verifying compiler' which facilitates the construction of rigorous proofs that an algorithm is correct. (A proof of validity may be considered as the first step in the analysis of an algorithm.)