A. I. 67

ARTIFICIAL INTELLIGENCE:

THEMES IN THE SECOND DECADE

BY
EDWARD A. FEIGENBAUM

MEMO NO. 67

AUGUST 15, 1968

STANFORD ARTIFICIAL INTELLIGENCE PROJECT
STANFORD UNIVERSITY

D D C

JAN 1 4 1969

43

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)*	2a. REPORT SECURITY CLASSIFICATION
Artificial Intelligence Project	Unclassified
Computer Science Department	2b. GROUP
Stanford University	

3. REPORT TITLE

Artificial Intelligence: Themes in the Second Decade

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

A.I. Memo

5. AUTHOR(S) *(First name, middle initial, last name)*

Edward A. Feigenbaum

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
August 15, 1968	40	76

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
SD-183	AI-67
b. PROJECT NO.	
c.	9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*
d.	

10. DISTRIBUTION STATEMENT

Statement No. 1 - Distribution of this document in unlimited

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY

13. ABSTRACT

In this survey of artificial intelligence research, the substantive focus is
heuristic programming, problem solving and closely associated learning models.
The focus in time is the period 1963-1968. Brief tours are made over a variety
of topics: generality, integrated robots, game playing, theorem proving, semantic
information processing, etc.

One program, which employs the heuristic search paradigm to generate explanatory
hypotheses in the analysis of mass spectra of organic molecules, is described in
some detail. The problem of representation for problem solving systems is discussed.
Various centers of excellence in the artificial intelligence research area is
mentioned. A bibliography of 76 references is given.

DD FORM 1473
NOV 65

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
heuristic programming						
problem solving						

ARTIFICIAL INTELLIGENCE: THEMES IN THE SECOND DECADE

by

Edward A. Feigenbaum

ABSTRACT: In this survey of artificial intelligence research, the
substantive focus is heuristic programming, problem solving,
and closely associated learning models. The focus in time is
the period 1963-1968. Brief tours are made over a variety of
topics: generality, integrated robots, game playing, theorem
proving, semantic information processing, etc.

One program, which employs the heuristic search paradigm to
generate explanatory hypotheses in the analysis of mass
spectra of organic molecules, is described in some detail.
The problem of representation for problem solving systems is
discussed. Various centers of excellence in the artificial
intelligence research area are mentioned. A bibliography of
77 references is given.

ARTIFICIAL INTELLIGENCE: THEMES IN THE SECOND DECADE

by Edward A. Feigenbaum

The purpose of this talk is to survey recent literature in artificial intelligence research, and to delineate and assess trends in the research. For an infant field of research that has been growing as rapidly as this one has, with emphasis on pragmatics and techniques, without benefit of much theoretical underpinning, both the delineation and assessment present problems.

The most memorable scientific talk I ever attended was delivered entirely impromptu to an informal Stanford group by my colleague Professor Joshua Lederberg. The talk ranged over research in what might be called "RNA and DNA information processing". Though his interests range broadly, the ground he covered that day was clearly his own ground-- a territory in which he has few peers.

Like a double helix, his talk had two intertwined strands. One strand carried the basic information on what experiments had been carried out and the empirical findings. The other strand consisted of Lederberg's personal scientific assessment of the quality of individual experiments and the value of the results; of judgments as to the potential fruitfulness of pursuing certain lines of endeavor, and the likely unfruitfulness of pursuing others; of an assessment of core issues needing resolution vs. issues that were merely interesting but peripheral; and of many other threads of an evaluative nature. In general, this strand of the talk consisted of comments covering a broad spectrum, from those for which

1

there was a strong scientific justification ("almost proven") to those based on the subjective and intuitive feelings that long experience in a field is supposed to give ("I have a hunch that ... "). In sum, the listener was left with a mental map of the problem-experiment-theory maze that constituted the current state of this area of molecular biology research, with values for present status and futures associated with the alternate paths through the maze.

This double-stranded approach is the model I have taken for what a survey should attempt. It should be something other than a comprehensive set of pointers into the literature. Careful selection based on sometimes personal criteria of relevance and importance is essential; evaluations based on sometimes subjective criteria of plausibility and potential are useful.

This is a talk, not a book, so I can not survey all the areas that have a rightful place under the umbrella of "artificial intelligence research". My choice of topic headings is not intended as a definition of "artificial intelligence" by implication. Vigorous subareas, with their own scientific "culture" and established publishing patterns, were left to fend for themselves. Thus, for example, the strong subarea that calls itself "pattern recognition research" was not surveyed, nor was the linguistics-translation subarea, bionics, neurophysiological information processing models, and others.

The focus of this talk is heuristic programming, problem solving, and closely associated learning models. Within the beam of this spotlight, I will concentrate on research of the period 1963-68, since I feel that the book Computers and Thought (21) is already an adequate reference work

2

for the 1956-62 period. As a practical measure, I will use the abbreviation "A.I." for "artificial intelligence".

Some Global Characteristics of the A.I. Research Endeavor

Of prime interest is the explosion of problems attacked, projects established, and reports published in the past five years. In spite of this rapid growth, quality has been maintained at a reasonably high level, in my opinion.*

From the very beginning of the A.I. research at Carnegie Tech in 1955-56 (which I regard as The Beginning for all practical purposes), Newell and Simon called their research "Complex Information Processing". They still do, though many projects have been born since as "artific. 1 intelligence projects". In this, Newell and Simon are to be credited with considerable foresight. For A.I. research is becoming ever more enmeshed at its periphery with other areas of computer science research and application that can well be described as "complex information processing". For example, is the research on intelligent question-answering programs still to be regarded as A.I. research, or is it the natural direction for progress in the field called information retrieval research? Is the

---

*Some observers have commented upon a dip in productivity in the period 1960-63, and this appears to be documentable. I believe that this was attributable to: a shift of emphasis at some of the major centers toward technological problems of tool building; a much-needed reassessment of the implications and significance of efforts of the late 1950's; a sub-stantive shift of attention to problems for which a long gestation time was needed (e.g. natural language analysis, integrated robots, represen-tation); and the establishment of academic computer science departments, programs, curricula, etc., which absorbed a significant portion of the energies of the available talent. Each of these was to have its eventual payoff in the productive 1963-68 period.

3

effort to develop a problem solving program to write computer operating systems (24) an A.I. effort or is it research in systems programming? Is a program (22) that forms chemical hypotheses in the analysis of mass spectra of organic molecules a piece of A.I. research, or is it chemistry?

These questions are not as trivial as the obvious "Who cares!" answer would make them seem. There is a general tendency in virtually all lines of scientific endeavor for research disciplines to fragment into special-ities as the early and difficult problems become better understood and as practitioners move into the discipline to make use of the results. "Successes" are then attributed to the specialities, ignoring the con-tributions from the spawning discipline.

In A.I. research, examples of this process at work are numerous. Consider character recognition (i.e. what those "optical readers" do). Much of the early work on the pattern recognition problem focused on character recognition as an interesting initial task. This research, circa 1955, was motivated more by the question, "What are the interesting kinds of behavior that a computer might be made to perform, in contrast to the mundane tasks of the day (such as calculating function tables and payrolls)?" than by the question, "How can we make a machine read characters of the alphabet reliably?" (62) The pursuit of the more general question inspired the early work on problem solving programs. Eventually the line of research turned into the applied art of designing character recognition machines and thereby, for all practical purposes, passed out of the field of A.I. research. List processing provides another example. As a body of techniques it was invented (as far as I can deter-mine) by Newell, Shaw, and Simon for handling the complex problems of

memory allocation and heirarchical (and recursive) control of processing in the Logic Theory program and the earliest version of the General Problem Solver. List processing received additional refinement by another A.I. researcher, McCarthy (LISP). It underwent further change (threaded lists, knotted lists, symmetric lists, etc.) as it made the transition from "something those A.I. researchers are doing" to "software system techniques". By now list processing is an every-day working tool of a number of specialty areas, particularly compiler and operating system implementation.

Every discipline thrives on its successes, particularly in terms of attracting talented individuals and research support funds. There is a danger that the A.I. area, as the residual claimant for the problems not yet solved, the problems not yet well understood, the problems for which failure was the reward for the initial foray, will come to be viewed as the "no win" area of computer science and the home of the "pie-in-the-sky guys". There is a scattering of evidence that such a process is already at work, and I regard this as most unfortunate and undeserved.

Finally, the rapid growth of A.I. research has made the "Invisible College" of the area less viable. There is now a Special Interest Group within the ACM for Artificial Intelligence; a new journal is being prepared; and an international conference is being organized.

The Search for Generality

This is the title of the Newell-Ernst IFIP65 paper, one that deserves rereading (48). Others have since joined the quest. There appear to be two roads: the high road and the low road.

Those who walk the high road seek a generality of the total problem
solving system that will allow a core of problem solving methods that are
not task-specific to discover solutions in a wide variety of problem
domains. Here the problem of the internal representation in terms of
which the core of methods will operate is crucial. If the representation
is made general enough so that each new application does not have to be
tortured to fit into it, can the methods and associated processes be made
general enough to cope with it without a consequent loss of problem
solving power? We lack a good understanding yet of this problem of
generality and representation. A view of existing problem solving programs
would suggest, as common sense would also, that there is a kind of "law
of nature" operating that relates problem solving generality (breadth of
applicability) inversely to power (solution successes, efficiency, etc.),
and power directly to specificity (task-specific information). We do
not now know how to write problem solvers that will accept problems in
a rather general representation at the start but then alter the repre-
sentation systematically toward greater specificity and power as more
problem-specific information becomes available during the problem solving.
An example of this process has been worked out in detail (4), but
mechanization is not in view.

The General Problem Solver traveled the high road alone for nearly
a decade and established the search for generality as a viable research
path. Ernst and Newell's new monograph (17), exploring the successes and
problems encountered in applying GPS to a variety of task environments,
appears to signal the end of the first phase of the GPS trek. More
recently, GPS has acquired a life of its own, independent of its creators.

6

For example, the GPS paradigm has appeared in a slightly more generalized form as a program called FORTRAN Deductive System (54); and considerably transfigured as the Graph Traverser (15, 16). Another GPS variant has just emerged in Sweden (61).

Travelers on the low road seek not a general problem solving system but theorems and generalizations of technique concerning underlying mechanisms that are common to a class of problem solving programs. Perhaps the best example involves the heuristic search paradigm.

As it was a decade ago, the central paradigm of A.I. research is heuristic search. A tree of "tries" (aliases: subproblems, reductions, candidates, solution attempts, alternatives-and-consequences, etc.) is sprouted (or sproutable) by a generator. Solutions (variously defined) exist at particular (unknown) depths along particular (unknown) paths. To find one is a "problem". For any task regarded as nontrivial, the search space is very large. Rules and procedures called heuristics are applied to direct search, to limit search, to constrain the sprouting of the tree, etc.

While some of this tree-searching machinery is entirely task-specific, other parts can be made quite general over the domain of designs employing the heuristic search paradigm. The so-called "alpha-beta" procedure is a classical example (70, 60). Its employment is "obvious" if one is careful and thoughtful about search organization. It was employed as early as 1958 in the Newell-Shaw-Simon chess program, it being so much a part of the underlying machinery that the employers did not consider it worthy of bringing to the attention of others.

7

But what is obvious to some is not obvious to others. Each new program designer, flirting with the heuristic search paradigm, should not be forced to reinvent (or what is worse pass over in ignorance) generally applicable search-facilitating procedures, particularly if the procedures are subtle.

A small, but growing, body of knowledge of this type has emerged. The work of Slagle is noteworthy, particularly his MULTIPLE (70), which is in effect a "system" of such techniques. Other papers of interest are those of Nilsson ("minimum cost paths") (50), Floyd ("nondeterministic algorithms") (23) and Golomb and Baumert ("backtrack programming") (25).

In a sense the travelers on the low road are tool builders, but their tool building is often of an abstract and elegant sort.

Integrated Robots

History will record that in 1968, in three major laboratories for A.I. research, an integrated robot consisted of the following:

     a. a complex receptor (typically a television camera of some sort) sending afferent signals to ...

     b. a computer of considerable power; a large core memory; a variety of programs for analyzing the afferent video signals and making decisions relating to the effectual movement of ...

     c. a mechanical arm-and-hand manipulator or a motor-driven cart.

The intensive effort being invested on the development of computer controlled hand-eye and eye-cart devices is for me the most unexpected

8

occurrence in A.I. research in the 1963-68 period.

Research of this type began effectively with Ernst's thesis on a computer controlled mechanical hand (MH-1) (18). He wrote interesting heuristic programs for solving problems of manual manipulation in a real environment. MH-1 was almost totally "blind", but it did store a symbolic internal representation of the external situation (a "model") in terms of which it did its problem solving. The seminal piece of research for the visual ("eye") processing was the oft-cited thesis of Roberts (58) on the three-dimensional perception of solids from two-dimensional picture input.

The three current robot projects are direct descendents. They are: the Stanford Hand-Eye Project (McCarthy, et. al.), the MIT Hand-Eye Project (Minsky and Papert), and the Stanford Research Institute's Robot Project (Nilsson, Raphael, Rosen, et. al.).

Not much information about these projects has been published. Hence, what follows is to some extent anecdotal.

As one might expect, the design, implementation, and use of the robot hardware presents some difficult, and often expensive, engineering and maintenance problems. If one is to work in this area solving such problems is a necessary prelude but, more often than not, unrewarding because the activity does not address the questions of A.I. research that motivate the project. Why, then, build devices? Why not simulate them and their environments? In fact, the SRI group has done good work in simulating a version of their robot in a simplified environment. (A film of this is available.) So it can be done and the questions raised above are relevant.

9

The answer given is as follows. It is felt by the SRI group that the most unsatisfactory part of their simulation effort was the simulation of the environment. Yet, they say that 90% of the effort of the simulation team went into this part of the simulation. It turned out to be very difficult to reproduce in an internal representation for a computer the necessary richness of environment that would give rise to interesting behavior by the highly adaptive robot. It is easier and cheaper to build a hardware robot to extract what information it needs from the real world than to organize and store a useful model. Crudely put, the SRI group's argument is that the most economic and efficient store of information about the real world is the real world itself.

The task of building an integrated robot is one, I believe, that contains the possibility of studying some problems of major interest in artificial intelligence research, among which are: strategy formation and planning; the problem of representing situations for problem solving processes and subsequent modification of representations as new information becomes available; and visual perceptual processing. Of the three groups, only the SRI group has published papers discussing the more general aspects and goals of this research (59, 57).

Both the MIT and Stanford University groups have worked on programs for controlling a variety of arm-hand manipulators, from the very simple to the very complex, from the anthropomorphic variety to the very non-anthropomorphic. None of the more esoteric manipulators seem to have worked out very well, though there is no published documentation of successes, failures, and reasons.

10

Visual scene analysis programs are important in all of these projects. Most of the programming effort is being invested to build the proper tools and techniques to gain control of this piece of the task. The scene analysis problem is this: the TV image of a scene (digitized) is available to be read into the computer memory; scan and process it as necessary to produce a symbolic description of the objects in the scene and their various interrelationships. Guzman (29) at MIT attacked the scene analysis task in a somewhat abstracted form (no TV camera, a no-noise symbolic simulation of the input scene) with striking success. Guzman's work, incidentally, should be of interest to psychologists working on models of human visual perception processes.

The name of the game however is integrated behavior on a problem. Both the MIT and Stanford University hand-eye-computer aggregates have performed a few types of block-finding and block-stacking behaviors. A paper in the IFIP68 Proceedings describes the Stanford work (53); I have not found a paper describing the MIT block-stacking activity.

So you want to build an integrated robot? Wait! The three lively groups, whose levels of talent and funding are hard to match, have not yet uncovered all the first-level problems. These will be found, reported, and stressed, perhaps within the next two years. The projects are still very much in the tool-building and debugging stage. Whether the integrated robot is a useful and appropriate task for making progress on the general problems of A.I. research remains to be proven.

Theorem Proving

Since Robinson is presenting an invited survey paper on automatic theorem proving at this conference, it would be inappropriate for me to survey this literature here. But perhaps a few comments and a few citations are in order.

Many in the field persist in calling their theorem proving programs deduction programs (thus, the aforementioned FORTRAN Deductive System, DEDUCOM (69), the term "deductive question-answering programs"; Hunt's survey (32) contains numerous instances of this misuse). This is a terminological mistake. If one looks carefully at how these programs find their proofs, much more than deduction in the strict sense is involved. When a theorem proving program applies a rule of inference, say modus ponens, in taking a trial step forward, it is clearly making an elementary deduction. But the search for a proof is not a deduction. In practice, the terminological error has tended to inhibit clear thinking on key problems, for example those involved in the attempt to unify parts of the field (like heuristic search) by a careful examination of the basic mechanisms used in a variety of successful programs. The useage I am vilifying is not harmless because it tends to sort the work of the field into the wrong categories.

I prefer Amarel's term, "problems of derivation type", as correct, clear, and meaningful. I feel the term "discovery processes" is an appropriate and useful one for describing the processes by which the proof of a theorem (or the move from a chess position, or the chemical hypothesis that explains a mass spectrum, etc.) is found.

12

Robinson's resolution method for theorem proving in the predicate calculus has received much attention in the past few years. Unfortunately, this has been accompanied by a sentiment that the resolution method "liberates" one from the guessy, messy chaotic world of heuristic search. Again a false distinction, based on unclear understanding either of resolution or of existing heuristic search proof-finding programs or both, sorts the world along the wrong lines. The resolution method does provide a systematic formal mechanism for guaranteeing the completeness of the space of proof candidates, but it does not by itself deal with the well-known and inevitable problem of the proliferation of problem states (58a). Thus search strategies have been overlaid to bring about effective problem solving using resolution (e.g. "unit preference", "set of support"). The net result is that the processes these programs carry out are much the same as (in some cases, identical to) those carried out by the heuristic search proof-finding programs that are thought to be so different (17). I predict that much more will be heard on this issue in the second decade.

In 1959, McCarthy proposed a class of programs (advice-takers) that would reason about the world in a common-sense way. The "world" was to be given a homogeneous representation in the predicate calculus. Problems which presented themselves would be solved as proofs over the space of expressions. Recertly, Green and Raphael (26) have incorporated the machinery of the resolution method as the proof-finding "engine" in an advice-taker-like question-answering and fact retrieval program. The idea is important (and works), but as just mentioned the necessary heuristic overlay to guide the search effort will have to be provided if the system is to be useful and practical.

13

Game Playing Programs

In the first decade, there were those who wrote chess playing programs because chess provided an interesting and complex task environment in which to study problem solving processes (the capstone of this line of research is a gem of a paper by Newell and Simon (49) examining in great detail an example of human chess play in the light of what we have come to understand about problem solving processes in chess from building chess playing programs). There were others who wrote chess programs because the activity presented such a challenge: chess is THE great centuries-old human intellectual diversion.

Such a group is Greenblatt, et. al. They have written the first program that plays very good (but not yet expert) chess. I have seen only one paper on the Greenblatt program (27). Along with a brief description, it gives some examples of the program's play. Competing against humans under ordinary tournament rules, it is said to have won a Class D tournament in Boston in mid-1967, reportedly beating a Class C player in the process. It is also reported to be much better by now. Apparently, its most celebrated victory was a handy win over Hubert Dreyfus.

Why does it play so well as compared with previous chess programs? I do not know anyone who yet has a convincing answer to this (partially because of the paucity of information about the program). As I view it, the program embodies no fundamentally new ideas about how to organize chess playing programs. It contains much more specific information about chess play than any previous program had. Computer time, top programmer talent, fancy tools of the second decade (CRT displays, interactive access, big core memory) -- the patient has received an order of magnitude more

14

loving care than any other previous patient (all other patients, you will remember, were released in a weak condition; most died). Finally, an excellent "learning loop" is available--through a human at the console. Blunders are analyzed, and quickly fixed with patches and/or new chess knowledge. The system-wide effects of the patch or the new knowledge, if any, can be fairly quickly detected, and revised if causing problems. It is a feasible way to improve (or educate) a program, and a useful one if you are interested in a high level of performance in a specific task rather than in general models for the organization of problem solving. I foresee this technique, albeit in more sophisticated forms, being widely adopted in the second decade.

The first international tournament between chess playing programs was won by a program developed in the Soviet Union at the Institute for Theoretical and Applied Physics in Moscow (by Adelson-Velskii, et. al; no writeup available). The loser was the old MIT program, developed by Kotok (37) and slightly modified by McCarthy's group at Stanford. The play is available for inspection in the SICART Bulletin (56). Neither program played well when compared with the average level of performance of the Greenblatt program.

Samuel's well-publicized checker playing program has undergone extensive revision (60), and now stands near the top of its profession. The major revisions are in the area of the learning routines, and will be discussed later.

Williams (74) has attacked the problem of modeling the generality with which human players approach common board and card games. His program, General Game Playing Program, is given as input a description

15

of the objects used in playing the game; and the rules of the game taken from Hoyle's Rules of Games, transformed straightforwardly into a Hoyle-like input language. The program will then play at least a legal game, for most of the games described in Hoyle.

Machine Learning (Specifically, Internal Mechanisms; not Human-Directed)

The A.I. field still has little grasp of the machine learning problem for problem solvers. For many years, almost the only citation worth making was to Samuel's famed checker playing program and its learning system. (Great interest arose once in a scheme proposed by Newell, Shaw, and Simon for learning in GPS, but the scheme was never realized.) Surprisingly, today we face the same situation.

Samuel's new paper (60) describes a major revision of the position evaluation scheme of the checker player and its attendent learning processes. Evaluation using the linear polynomial function is abandoned in favor of complex nonlinear evaluation processes. The features of positions, which used to be represented as terms in the polynomial, are now grouped according to their (perceived) interdependencies. "Crude" values for the features are used, e.g. 3-,5-, or 7- valued functions. (This is an old idea, whose rationale was given as far back as 1951 by Simon for chess-playing; it is used in the Newell-Shaw-Simon Chess Playing Program.) Vectors of feature values are used to enter "signature tables" at the first level. A table-lookup takes place; the resulting table value is quantized into "crude" states (five-valued) and passed up to a second level of "signature table" aggregation (over a set of first level signature tables). This process is repeated once again at a third level of

16

aggregation, and from this "top" signature table a final score for the evaluation emerges. Samuel shows the considerable performance advantages of this hierarchical scheme over the (already quite successful) linear polynomial evaluation.

Samuel titles one of his sections, "The Heuristic Search for Heuristics", and maintains therein "that the task of making decisions as to the heuristics to be used is also a problem which can only be attacked by heuristic procedures, since it is essentially an even more complicated task than is the playing itself." An interesting case study of the learning of heuristics by a heuristic program is emerging in the dissertation research of Waterman at Stanford (72), nearly complete. The task environment is draw poker. The heuristics are not cast as programs in the traditional mode, but are "brought to the surface" as situation-action rules in a "production" list. Initially the list contains only the null rule: whatever the situation, play randomly. Basically, four things can happen to the table of rules. Situation-action rules can be added. The order of the rules can be altered (since the table is scanned in a top-to-bottom fixed order, this can make a big difference in behavior). A rule can be "generalized" by altering its situation-side so as to ignore certain dimensions of the game situation, thereby causing it to "catch" more situations. Or the situation-side can be "specialized" so as to be more discriminating among game situations and hence "catch" fewer situations. This learning scheme works well in a variety of different training procedures.

17

A Note in Passing: Turning Inward to the Programming Task Itself

In the checker-playing program, Samuel used a "rote memory" learning scheme, in which many checker board positions were stored away along with their scores from previous look-ahead search. If a "memorized" position were encountered in a new analysis, the value was available and would not have to be recomputed. Theorem proving programs use "rote memory" for analogous purposes in building their theorem memories. So do many other programs, e.g. Heuristic DENDRAL, described later.

The issue of store vs. recompute is quite general and classical. One looks forward to the day when the programming system one is using is smart enough to figure out when it should assign function values by table lookup in a rote memory it builds and when by computation (it would make this decision by an analysis of the uses of and performance characteristics of the function as it encounters this information during execution). A first step in this direction has recently been made with the introduction of "memo functions" into the Edinburgh POP-2 language (40). With memo functions, however, the programmer still has decisions to make, and I look forward to a further "Samuelization" of programming language systems for the store vs. recompute decision.

Simon (63) once wrote a problem-solving program (Heuristic Compiler) that solved problems of writing simple programs in IPL-V, given descriptions of how the programs were supposed to alter the state of the IPL-V machine. The program was organized as a simplified version of GPS, and was successful, but it has not been followed up with a more substantial effort. Ph.D. candidates, where are you?

18

Semantic Information Processing and Natural Language

This area of research is of great importance to the A.I. endeavor. Its importance arises not so much from the presumed practical advantages of being able to converse in natural language with a program that "understands", but because the research raises and focuses certain key issues that are quite general. Research of high quality has been done in the past few years. A book (42) covering much of it will be available shortly. Minsky's introduction to the book should be consulted for an extended treatment of the subject, which is impossible here. Parts of a paper by Coles (13) also give a good treatment. Nevertheless, a few comments may be useful.

The research grapples in various ways with the problem of the meaning of ordinary natural language utterances and the computer under-standing of the meaning of these utterances as evidenced by its subsequent linguistic, problem-solving, or question-answering behavior. Meaning is viewed not as something one "puts into" a program (for example, ordinary dictionary entries help very little) but as an emergent from the interplay of syntactic analyzers, models that link to real-world objects and relations, appropriate data structures that link together symbols of the internal-world, a logical calculus and associated discovery processes for solving derivation-type problems. (This list is not necessarily exhaustive.)

Syntactic analysis, which has received massive attention from the computational linguists (with elegant results), is not enough to handle problems of meaning and understanding.* Consider the following example.

_____

*Overemphasis on syntax analysis at the expense of research on semantic processes has hindered the development of the mechanical translation area, I believe.

Bobrow's STUDENT (8) is a problem solver that accepts natural language input (English sentences). High-school-level algebra word problems constitute the domain of discourse. The sentences are simplified and parsed; idioms are transformed. Reference to the real-world is made through a table of global relations. Typically the amount of global information made available to STUDENT has been quite small; hence, STUDENT's understanding of the algebra word problems is largely "syntactic". The appropriate simultaneous algebraic equations are set up and solved for the answer.

STUDENT can be made to solve the following problem: "A board was sawed into two pieces. One piece was two-thirds as long as the whole board and was exceeded in length by the second piece by four feet How long was the board before it was cut?" STUDENT will show that in one sense it understands this problem by issuing the correct solution, that the board length was minus twelve feet. In the psychological experiments done by Paige and Simon in the light of STUDENT (51), some subjects solved the problem just as STUDENT solved it (with a focus on the syntax leading to the correct equation system), but others immediately recognized its physical impossibility and refused to set up the equations. These people were the model builders, who attached the "givens" to the appropriate model of the physical situation, immediately noticing the misfit*. They were exhibiting another level of "understanding of the problem" not available to STUDENT.

_____

*My own informal replications of the experiment at Stanford led me to believe that the effect is independent of the brilliance, or lack of it, of the subject, but a function of whether he is a "visualizer" or a"symbolizer". STUDENT, of course, is a "symbolizer".

The notion of interpretation of natural language in terms of stored models is central to much of the research on semantic information processing. Raphael's Semantic Information Retrieval question-answering system (55) uses a node-link relational model (with a restricted set of relations) to organize its data universe. This model is grown to incorporate new information about objects and their relationships that is extracted from a simple analysis of declarative sentences typed in at the console by the user. Other programs use internal representations of two-dimensional pictures as a model. Coles (13), extending the work of Kirsch, et. al. (33) on the Picture Language Machine, wrote a program that uses a picture (input by the user at a CRT with a light pen) to resolve syntactic ambiguities in English sentences about the picture, and to answer questions about the picture.

Pushing the subject of models and data structures a bit further, consider restructuring a traditional dictionary into a "semantic graph" in the following way. Represent entities as symbols at the nodes, and the various general and special relations between entities as associative links between the nodes. An entity's node is the start of a subgraph which, when traced out, encompasses the various "meanings" of the entity in terms of other entities and relations in the semantic graph. Quillian (53a) has constructed such a graph for a small (but definitely nontrivial) set of dictionary definitions, and a processor for performing a variety of associative and semantic reference tasks. Quillian's program, I believe, is a good foundation for further research on models of human associative memory. This possibility is worth a vigorous pursuit.

21

Simulation of Cognitive Processes

Space limitations here preclude a survey of the work in this interesting

territory at the intersection of computer science and psychology; the

formulation and validation of information processing theories of human

problem solving and learning processes using primarily, but not exclusively,

the techniques of heuristic programming and the methodology of computer

simulation. Fortunately, two thorough reviews (31, 1) have appeared

recently and are recommended. Nevertheless, I can not resist giving my

own personal set of pointers into the literature.

Problem Solving: Newell (47, 46, 43).

> Analysis of human behavior in crypto-arithmetic puzzle solving
> tasks; major methodological advances in analysis of human
> problem solving "think-aloud" protocols and the study of human
> eye movements during problem solving.

Verbal learning and memory: Simon and Feigenbaum (66); Gregg and
Simon (28); Feigenbaum (20); Hintzman (30).

> Further results with Elementary Perceiver and Memorizer (EPAM)
> model; reinterpretation in terms of theory of various levels
> of human memory; extensions by Hintzman to handle additional
> phenomena.

Concept learning and pattern induction: Hunt, Marin, and Stone (32)
reports many experiments with Concept Learning System (CLS).

> Simon and Kotovsky (67); Simon and Sumner (68). Simple, elegant
> program that handles sequence completion tasks from standard
> intelligence test also can infer and extrapolate patterns in
> melodies.

Affect, emotion, beliefs: Tessler, Enea, and Colby (71); Abelson
and Carroll (2). Models of human belief systems and their use
in studying neurotic and "normal" behavior.

> Simon (64). Emotional and motivational concomitants to cognitive
> processes.

Judgment: Kleinmuntz (34, 35).

> Model of the clinician's judgmental process in constructing
> personality profile from subject's answers in Minnesota Multi-
> phasic Personality Inventory; and studies of other clinicians'
> tasks.

The Shape of the Field, 1968: By Examination in Depth of One Program

In attempting to characterize state-of-the-art in a field, a careful
look at one research effort is often revealing. As with an etching, the
lines of the work--lines of ideas, of methods and techniques, of technology
employed--are seen with greater clarity from close up.

I have chosen for the close-up the research work with which I am
personally involved. It is not only more appropriate for me to do this
than to sketch another's work, but also the sketch carries with it an
assured knowledge of detail. Most important, the research lies squarely
in what I consider to be the mainstream of the A.I. research endeavor:
problem solving using the heuristic search paradigm.

Our primary goal was to study processes of hypothesis formation in
a complex task of a scientific nature involving the analysis of empirical
data. The task environment chosen as the medium for this study was the
analysis of the mass spectra of organic molecules: the generation of a
hypothesis to best explain given mass spectral data. This is a relatively
new area of organic chemistry of great interest to physical chemists.
In this sense, the problem is not a "toy" problem; and a program that
solves problems of this type is a useful application of A.I. research
to a problem of importance to science.

We have written a program to infer structural hypotheses from mass
spectral data. The program is called Heuristic DENDRAL. It was developed
at the Stanford University Artificial Intelligence Project by a small
group including Professor Joshua Lederberg of the Stanford Genetics
Department, Dr. Bruce Buchanan, Mrs. Georgia Sutherland, and me, with
the assistance of chemists and mass spectrometrists of the Stanford

23

Chemistry Department. It is an 80,000 word program written in LISP for the PDP-6 computer, and was developed (and is run) interactively under the time-sharing monitor (39, 9, 22).

Heuristic DENDRAL will perform the following two classes of tasks:

1. Given the mass spectrum of an organic molecular sample and the chemical formula of the molecule, the program will produce a short list of molecular "graphs" as hypotheses to explain the given data in the light of the program's models of mass spectrometric processes and stability of organic molecules. The list is rank-ordered from the most satisfactory explanation to the least satisfactory.

2. If no mass spectrum is given, but only a formula, the program will produce a list of all the chemically plausible isomers of the molecule in the light of its model of chemical stability of organic molecules.

The flow diagram of the system is a closed loop consisting of phases of data inspection, hypothesis generation, prediction, and test, corresponding closely to a simple "scientific method" loop.

At the heart of the program is a systematic hypothesis generator. It is based on an algorithm developed by Lederberg called DENDRAL which is capable of generating all of the topologically possible isomers of a chemical formula. The generator is essentially a topologist, knowing nothing about chemistry except for the valences of atoms; but the generating algorithm serves as the guarantor of the completeness of the hypothesis space, in a fashion analogous to the legal move generator in a chess program. Since the generating process is a combinatorial

procedure, it produces for all but the simplest molecules a very large set of structures, almost all of which are chemically implausible though topologically possible. Implicit in its activity is a tree of possible hypothesis candidates. At the top node of the tree all the atoms are found but no structures. At the terminal nodes, only complete structures are found, but no unallocated atoms. Each intermediate node specifies a partially built structure and a residual set of atoms yet to be allocated.

This tree is the implicit problem space for Heuristic DENDRAL. Various heuristic rules and chemical models are employed to control the generation of paths through this space, as follows:

1. A model of the chemical stability of organic molecules based on the presence of certain denied and preferred subgraphs of the chemical graph. It is called the a priori model since it is independent of processes of mass spectrometry.

2. A very crude but efficient theory of the behavior of molecules in a mass spectrometer, called the Zero-order Theory of Mass Spectrometry, used to make a rough initial discarding of whole classes of structures because they are not valid in the light of the data, even to a crude approximation.

3. A set of pattern recognition heuristic rules which allow a preliminary interpretation of the data in terms of the presence of key functional groups, absence of other functional groups, weights of radicals attached to key functional groups, etc. It is called the Preliminary Inference Maker. Its activity allows the Hypothesis Generator to proceed directly to the most plausible subtrees of the space.

25

The output of the Preliminary Inference and Hypothesis Generation processes is a list of molecular structures that are candidate hypotheses for an explanation of the mass spectrum. They are all chemically plausible under the a priori theory and valid explanations of the data under our zero-order theory of mass spectrometry. Typically the list contains a few candidates (but not dozens or hundreds).

Next a confrontation is made between this list of "most likely" hypotheses and the data. For each candidate hypothesis, a detailed prediction is made of its mass spectrum. This is done with a subprogram called the Predictor, a complex theory of mass spectrometry in computer simulation form. The Predictor is not a heuristic program. It is an elaborate but straightforward procedure for deducing consequences of a theory of mass spectrometry extracted by us from chemists and their literature. The spectral prediction for each candidate is matched with the empirical input data by a process called the Evaluation Function. This is a heuristic, hierarchical, non-linear scoring procedure. Some hypothesis candidates are immediately discarded because their predicted spectra fail certain critical confrontations. The remainder are scored, ranked, and printed out in rank order from most to least satisfactory.

For the class of non-ringed organic structures with which we have been working up to the present time, the program's behavior approaches or exceeds the performance of post-doctoral laboratory workers in mass spectrometry for certain classes of organic molecules. These include amino acids, with which for tangential reasons we have done much of our work, and a large variety of simple organic groups that, however, turn out to be considerably more complicated than amino acids from the point of view of mass spectrometry.

26

Heuristic programming provided only the skeleton for the problem
solving processes of Heuristic DENDRAL and the computer techniques to
handle the implementation. The heuristics of chemical plausibility of
structures; of preliminary inference; of evaluation of the predictions;
and also the zero-order and complex theories of mass spectrometry--these
were all extracted from our chemist colleagues by man-machine interaction,
with the process carefully guided by one of our research team. The success
of this mixed discipline for pulling out of the heads of practicing pro-
fessionals the problem solving heuristics they are using has worked far
better than we had any right to expect, and we are now considering further
mechanization of this process.

The Problem of Representation for Problem Solving Systems

A.I. research in the remainder of the second decade will be dominated
by a few key problems of general importance. The problem of representation
for problem solving systems is one of these, and in my view the most
important, though not the most immediately tractable.*

In heuristic problem solving programs, the search for solutions
within a problem space is conducted and controlled by heuristic rules.
The representation that defines the problem space is the problem solver's
"way of looking at" the problem and also specifies the form of solutions.

---

*I have used the term "problem of representation for problem solving
systems" to distinguish this problem from the much more widely
discussed data representation (and data structures) problem. I
believe that we will find eventually that the two sets of questions
have an important intersection, but for the moment it is best to
avoid terminological confusion.

27

Choosing a representation that is right for a problem can improve
spectacularly the efficiency of the solution-finding process. The
choice of problem representation is the job of the human programmer
and is a creative act. Amarel (5) believes that the process of choosing
and shaping appropriate representations for problem solving is the
essence of the behavior in humans that we call "creative". I agree.

Some examples of the impact of choice of representation on problem
solving performance have been discussed in the literature. The classic
is the so-called "tough nut" proposed by McCarthy and discussed by
Newell (45). Mutilate a chess board by removing two corner squares diagonally
opposed; can the mutilated board be covered by dominos? If the standard
piece-board-move game playing representation is employed, an enormous and
almost impossible search would have to be conducted to discover that no
covering solution was possible. But a choice of problem representation
involving the concepts of parity of red-black covering by a domino and
of counting of red and black squares leads immediately to the solution
that no covering is possible because two squares of the same color are
removed in the mutilation.

Another example of much greater complexity has been worked out by
Amarel for the traditional puzzle of transporting missionaries and
cannibals from one bank of a river to the other with a boat under certain
constraints (4). Amarel exhibits a succession of representational shifts
for this problem, from the one usually used to a simple but elegant
matrix-like representation, in terms of which the solution is available
almost immediately by inspection. Amarel has worked out still another
example for theorem proving in the propositional calculus (4a). In fact,

28

as early as 1958, Gelernter in the Geometry Machine used a diagram as an auxiliary problem representation to improve the efficiency of searching the problem-subproblem tree.

Until very recently the problem of representation has been treated in the literature by exploring a few examples in detail. Fortunately, a new paper by Amarel (6), offering a synthesis of his view of problem solving and representation, gives a clear formulation and extended discussion of this difficult area.

Why is it that a shift of problem representation can lead to a spectacular change in problem solving effectiveness? There are many reasons; here are a few. Each problem representation has associated with it a set of specialized methods for manipulating elements of the representation. Shifting to a representation that is "rich" in specialized theory and methods from one that is impoverished in this regard allows the power of the former to be applied to the problem at hand. Similarly, specialized relationships associated with an appropriate representation can be imported into the (often incomplete) statement of a problem thereby supplying missing but crucial augmentations to the problem definition. An example of this has been exhibited by Paige and Simon (51) for alcohol-water mixture problems (the appropriate representation supplies necessary conservation equations). Finally, each representation can have associated with it a data base of descriptions and facts that become available for incorporation into the problem statement or for use in controlling search.

Amarel has discussed the mechanization of the process of shift of representation as a step-by-step process, involving an "evolution" of

29

each representation to a somewhat more powerful one (4, 4a). The evolution is guided by information about the problem (or problem class) that turns up during the problem solving activity within a particular representation of the moment.

The alternative to this step-by-step process is a generator of new representations as trial candidates in a heuristic search for an appropriate representation. Design of such a generator is a formidable task at this early stage in our understanding of the representation problem. The simplest design, however, is to generate the elements of a stored repertoire of previously encountered or potentially useful representations. Such a design was employed in a program by Persson (52) for the problem of choosing the appropriate representation of pattern in a mixture of different sequence extrapolation tasks.

In my view, the use of the concept of analogy between problems is a crucial step in the design of a generator of representations. Candidates for an appropriate problem representation are searched for, discovered, and tried, by a search process that uses analogical reasoning over a store of known representations (and their associated methods, data bases, etc.). Problem solving search using reasoning-by-analogy has received surprisingly little attention in A.I. research, considering the importance of the problem. The work by Evans (19) on a program to solve "intelligence test" problems involving geometrical analogies is the only work I can cite.

Of necessity, these comments on the problem of representation have been sketchy in the extreme. But because of the central importance of this problem, I felt the need to focus attention on it. I believe that the long gestation period for this problem is ending; that the time is

30

ripe for a major push; that there will be important developments on this front in the next five years; and that these will come to be viewed as having the same degree of centrality and importance that now attaches itself to the heuristic search paradigm.

Centers of Excellence

It is conventional to survey research by topic and unconventional to survey it by places and people. Yet I am frequently asked in conversation some form of the question: "In artificial intelligence (particularly heuristic programming), where is the action?" There is no reason for not attempting to answer this question in a public forum.

The reference point in time is mid-1968. The emphasis is on a substantial _quantity_ of _high quality_ research (my assessment).

In the United States, the three major research centers are the A.I. projects at MIT, Carnegie-Mellon University (née Carnegie Tech), and Stanford University. All three receive the major portion of their support from the Advanced Research Projects Agency (Department of Defense). All three train substantial numbers of Ph.D. students in the A.I. area. The MIT and Stanford projects use dedicated PDP-6 computers with big core memories; the Carnegie project uses an IBM 360/67 with a very large extended core memory.

At MIT, the more senior faculty and research principals are Minsky, Papert, and to some extent, Weizenbaum (73); at Carnegie, Newell and Simon; at Stanford, McCarthy, Samuel, Colby, and Feigenbaum. The Stanford group has close ties with the neighboring SRI group (Nilsson and Raphael); similarly the MIT group has close ties with Bobrow's group at Bolt, Beranek, and Newman.

31

Citing some statistics from the Stanford University group, which I obviously know best, there are 75 people (faculty, students, and staff) associated with the A.I. project; about 25 different research projects underway; and a working paper series of 67 papers. I offer these figures to indicate scale of effort at a major center.

Five other centers deserving attention are: Case Western Reserve University (Banerji, Ernst); University of Wisconsin (Travis, Uhr and London); RCA Laboratories at Princeton (Amarel); Heuristics Laboratory, National Institutes of Health (Slagle); and the University of Washington (Hunt).

In Europe, no centers comparable to the major American centers were visible in the first decade. In the past few years, however, a center of the first rank has arisen at the University of Edinburgh, and other centers are emerging in Sweden and the Soviet Union.

At Edinburgh, A.I. research is enshrined in a Department of Machine Intelligence and Perception (how forthrightly can one state one's case?). The principals are Michie, Gregory, Burstall, Doran, and Popplestone. They are supported reasonably generously by the British Government. The research ranges very broadly from the various projects in information processing models of cognition and perception to applications of these models (10, 11) and development of programming languages (12). The latest "score sheet" from the Department gives bibliographic data for 59 "original research contributions" since 1965! This group is responsible for a major series of collected papers, the Machine Intelligence series.

At Uppsala University in Sweden, the Department of Computer Sciences is doing A.I. research (GPS, planning, robot simulations, LISP work).

Principal activist is Sandewall, formerly at Stanford. Psychologists interested in simulation of cognitive processes are participating. The Swedish Natural Science Research Council supports the research.

When I cast my mind's eye as far off as the Soviet Union, the image becomes fuzzy, though I make a determined effort to keep current with Soviet computer science literature (particularly the area of discussion, heuristic programming). The few papers I have seen are motivation-suppliers or clarifications at points of contact with philosophy, psychology, and neurophysiology. However, there are talented groups at various locations that are interested (and perhaps actively working) in the area. These are:

> Institute of Cybernetics, Ukrainian Academy of Sciences, Kiev
>   (Glushkov, Amosov, and coworkers)
>
> Institute of Automation and Remote Control, Moscow
>   (Aizerman's Laboratory)
>
> Moscow State University, Department of Higher Nervous Activity
>   (Napalkov)
>
> Computer Center, Siberian Division, Academy of Sciences of USSR,
>   Novosibirsk   (Yershov, Marchuk and coworkers)

All of these are major centers, interested generally in the problems of A.I. research. Whether the developers of the chess program mentioned earlier, at the Institute of Theoretical and Applied Physics in Moscow, are interested in problems other than chess program development I do not know.

A Russian translation of Computers and Thought, edited by Napalkov and Orfyeev, appeared last year and was an immediate sell-out. A Scientific-Technical Commission on Heuristic Programming has come into existence within the last two years.

33

A computer scientist, writing in _Izvestiya_, claims "solid successes of Soviet heuristic programming" in pattern recognition, chess, and theorem proving, but cites lags in "breadth of the work being done" and "in equipping these projects with computer hardware". (38) It would be useful for the A.I. field to have a survey paper by a Soviet computer scientist on Soviet work in heuristic programming.

References
    The list of references given below represents a
personal list relevant to the particular purposes of
this survey article. It is not intended as a compre-
hensive bibliography of the field. A comprehensive
bibliography of the literature through 1962 is given
in (21). No such bibliography yet exists for the
period 1963-68. The Special Interest Group for
Artificial Intelligence of the ACM is attempting
currently to build one.

1. R. Abelson, "Computer Simulation of Social Behavior",
        in: G. Lindzey (ed.) Handbook of Social Psychology,
        2nd edition (New York, Addison-Wesley, forthcoming
        Aug. 1968).
2. R. Abelson and J. Carroll, "Computer Simulation of
        Individual Belief Systems", American Behavioral
        Scientist (Vol. 8, 1965).
3. S. Amarel, "On Representations and Modeling in Problem
        Solving and on Future Directions for Intelligent
        Systems", (Scientific Report, RCA Laboratories,
        Princeton, New Jersey, June 1968).
4. S. Amarel, "On Representations of Problems of Reasoning
        about Actions", in: D. Michie (ed.) Machine
        Intelligence 3 (Edinburgh University Press,
        1968).
4a. S. Amarel, "An Approach to Heuristic Problem Solving
        and Theorem Proving in the Propositional Calculus",
        in: J. F. Hart and S. Takasu (eds.), Systems and
        Computer Science, (University of Toronto Press, 1967).
5. S. Amarel, "On the Mechanization of Creative Processes",
        IEEE Spectrum, (April 1966).
6. S. Amarel, "On the Representation of Problems and Goal-
        Directed Procedures for Computers", Working Paper
        (Computer Theory Research Group, RCA Laboratories,
        Princeton, New Jersey). To be published in: Proceedings
        of the First Annual Symposium of the American Society
        for Cybernetics (forthcoming).
7. S. Amarel, "Problem Solving Procedures for Efficient
        Syntactic Analysis", (Scientific Report, RCA
        Laboratories, Princeton, New Jersey, May 1968).
8. D.G. Bobrow, "A Question Answering System for High
        School Algebra Word Problems", in: Proceedings
        of the FJCC, 25 (1964).
9. B. Buchanan and G. Sutherland, "Heuristic Dendral:
        A Program for Generating Explanatory Hypotheses
        in Organic Chemistry", Artificial Intelligence
        Working Paper No. 62 (Computer Science Department,
        Stanford University), to be published in C. Michie
        et. al. (eds.), Machine Intelligence 4 (in pre-
        paration).

35

10. R. M. Burstall, "A Heuristic Method for a Job Scheduling Problem", Operations Research Quarterly 17 (1966).

11. R. M. Burstall, "Computer Design for Electricity Supply Networks by Heuristic Method", Computer Journal 9 (1966) 3.

12. R. M. Burstall and R. J. Popplestone, "POP-2 Reference Manual", In: E. Dale and D. Michie (eds.) Machine Intelligence 2 (Edinburgh, Oliver and Boyd, 1968), also in POP-2 Papers (Edinburgh, Oliver and Boyd, 1968).

13. L.S. Coles, "Syntax Directed Interpretation of Natural Language", Ph.D. dissertation (Computer Science Department, Carnegie-Mellon University, 1967).

14. J.D. Darlington, "Machine Methods for Proving Logical Arguments Expressed in English", Mechanical Translation 8 (1965) 3 & 4.

15. J. Doran, "New Developments of the Graph Traverser", In: E. Dale and D. Michie (eds.), Machine Intelligence 2 (Oliver and Boyd, Edinburgh, 1968).

16. J.E. Doran and D. Michie, "Experiments with the Graph Traverser Program", In: Proceedings of the Royal Society A, 294 (1966).

17. G. Ernst and A. Newell, "GPS: A Case Study in Generality and Problem Solving", ACM Monograph Series (in press).

18. H.A. Ernst, "MH-1. A Computer-operated Mechanical Hand", Ph.D. dissertation (Massachusetts Institute of Technology, 1961), presented at the WJCC, May 1962.

19. T.G. Evans, "A Heuristic Program to Solve Geometric Analogy Problems", In: Proceedings of the SJCC 25 (1964).

20. E.A. Feigenbaum, "Information Processing and Memory", In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Vol. 4, Biology and Problems of Health (University of California Press, 1967).

21. E. Feigenbaum and J. Feldman (eds.) Computers and Thought, (McGraw-Hill Book Company, 1963).

22. E.A. Feigenbaum, J. Lederberg and B. Buchanan, "Heuristic Dendral", In: Proceedings of the Hawaii International Conference on System Sciences, University of Hawaii and IEEE (University of Hawaii Press, 1968).

23. R.W. Floyd, "Non-Deterministic Algorithms", Journal of the Association for Computing Machinery 14 (Oct. 1967) 4.

24. P. Freeman, Ph.D. dissertation research (Carnegie-Mellon University, 1968).

25. S.W. Golomb and L.D. Baumert, "Backtrack Programming", Journal of the Association for Computing Machinery 12 (Oct. 1965) 4.

26. C.C. Green and B. Raphael, "Research on Intelligent Question Answering System", Scientific Report AFCRL-67-0370, (Artificial Intelligence Project, Stanford Research Institute, Menlo Park, Calif., May 1967).

27. G. Greenblatt, D. Eastlake and S. Crocker, "The Greenblatt Chess Program", in: Proceedings of the Fall Joint Computer Conference (Anaheim, Calif. 1967).

28. L.W. Gregg and H.A. Simon, "An Information-Processing Explanation of One-Trial and Incremental Learning", Journal of Verbal Learning and Verbal Behavior 6 (Oct. 1967) 5.

29. A. Guzman, "Some Aspects of Pattern Recognition by Computer", MIT Masters thesis, Report No. MAC-TR-37 (Project MAC, MIT, Feb. 1967).

30. D.L. Hintzman, "Exploration for a Discrimination Net Model for Paired Associate Learning", Journal of Mathematical Psychology 5 (Feb. 1968) 1.

31. E. Hunt, "Computer Simulation: Artificial Intelligence Studies and Their Relevance to Psychology", Annual Review of Psychology, 1968.

32. E. Hunt, J. Marin and F. Stone, "Experiments in Induction", (Academic Press, 1966).

33. R. A. Kirsch, "Computer Interpretation of English Text and Picture Patterns", IEEE Transaction on Electronic Computers, Vol. EC-13, (Aug. 1964).

34. B. Kleinmuntz, "Profile Analysis Revisited: A Heuristic Approach", Journal of Counseling Psychology 10 (1963).

35. B. Kleinmuntz, "The Processing of Clinical Information by Man and Machine", in: B. Kleinmuntz (ed.) Formal Representation of Human Judgment (Wiley, 1968).

36. M. Kochen, "On the Representation of Limited Information by Means of Pictures, Trees, and English-like Sentences", Working Paper, Computer Theory Research Group (RCA Laboratories, Princeton, New Jersey, 1965).

37. A. Kotok, "A Chess Playing Program for the IBM 7090", unpublished B.S. thesis (MIT, 1962).

38. A. Kronrod, "The Computer Becomes 'More Intelligent'", in: Izvestiya (March 15, 1967). Translated in: W. Holland (ed.) Soviet Cybernetics: Recent News Items, No. 3, (The RAND Corporation, Santa Monica, Calif., 1967).

39. J. Lederberg and E. Feigenbaum, "Mechanization of Inductive Inference in Organic Chemistry", in: B. Kleinmuntz (ed.) Formal Representation of Human Judgment (New York, John Wiley, 1968).

40. D. Michie, "Memo Functions and Machine Learning", Nature 218 (1968).

41. M. L. Minsky, "Artificial Intelligence", in: Information (Freeman, 1966).

42. M. L. Minsky (ed.) Semantic Information Processing (MIT Press, 1968, forthcoming).

43. A. Newell, "Eye Movements and Problem Solving", Computer Science Research Review (Carnegie-Mellon University, Pittsburgh, Pennsylvania, Dec. 1967).

44. A. Newell, "Heuristic Programming: Ill Structured Problems", Working Paper (Computer Science Department, Carnegie-Mellon University, Nov. 1967), to appear in: Progress in Operations Research (in press).

45. A. Newell, "Limitations of the Current Stock of Ideas about
     Problem Solving", In: A. Kent and O.E. Taulbee (eds.)
     Electronic Information Handling (Spartan, 1965).
46. A. Newell, "On the Analysis of Human Problem Solving
     Protocols", Working Paper (Computer Science Department,
     Carnegie-Mellon University, July 1967).
47. A. Newell, "Studies in Problem Solving: Subject 3 on the
     Crypt-Arithmetic Task Donald + Gerald = Robert",
     Working Paper (Computer Science Department,
     Carnegie-Mellon University, July 1967).
48. A. Newell and G. Ernst, "The Search for Generality", In:
     Proceedings of the IFIP65 Congress (New York, Spartan,
     1965).
49. A. Newell and H.A. Simon, "An Example of Human Chess Play
     in the Light of Chess Playing Programs", Working Paper
     (Computer Science Department, Carnegie-Mellon University,
     August 1964), published in: J. Schade and
     H. Wiener (eds.), Progress in Neurocybernetics,
     (Elsevier, Amsterdam, 1965).
50. N. J. Nilsson, "A New Method for Searching Problem-Solving
     and Game Playing Trees", Working Paper (Artificial
     Intelligence Group, Stanford Research Institute, Menlo
     Park, Calif., Nov. 1967). Paper presented at the
     IFIP68 Congress, Edinburgh.
51. J. Paige and H.A. Simon, "Cognitive Processes in Solving
     Algebra Word Problems", In B. Kleinmuntz (ed.) Problem
     Solving: Research, Method and Theory (Proceedings of the
     First Carnegie Symposium on Cognition, Wiley, New York, 1966).
52. S. Persson, "Some Sequence Extrapolating Programs: A Study of
     Representation and Modeling in Inquiring Systems",
     Report CS50 (Computer Science Department, Stanford
     University, Sept. 1966).
53. K. K. Pingle, J. A. Singer and W. M. Wichman, "Computer
     Control of a Mechanical Arm through Visual Input",
     paper presented at IFIP68 Congress, Edinburgh.
53a. J. R. Quillian, "Word Concepts: A Theory and Simulation of
     Some Basic Semantic Capabilities", Behavioral Science 12,
     1967.
54. J. R. Quinlan and E. B. Hunt, "The Fortran Deductive
     System", Technical Report 68-1-01 (Department of
     Psychology, University of Washington, Seattle
     Washington, Jan. 1968).
55. B. Raphael, "A Computer Program which 'Understands'",
     In: Proceedings of the FJCC 25 (1964).
56. B. Raphael (ed.), ACM SICART Bulletin (June 1967) p. 11.
57. B. Raphael, "Programming a Robot", Working Paper (Artificial
     Intelligence Project, Stanford Research Institute,
     Menlo Park, Calif.) Paper presented at the IFIP68
     Congress, Edinburgh.
58. L. G. Roberts, "Machine Perception of Three Dimensional
     Solids", In: Optical and Electro-optical Processing
     of Information (MIT Press, Cambridge, Mass., 1965).

58a. J. A. Robinson, "Heuristic and Complete Processes in the Mechanization of Theorem Proving", In: J. F. Hart and S. Takasu (eds.), Systems and Computer Science, (University of Toronto Press, 1967).

59. C. A. Rosen and N. J. Nilsson, "An Intelligent Automaton", Working Paper (Artificial Intelligence Group, Stanford Research Institute, Menlo Park, Calif.). Paper presented at the 1967 IEEE International Convention, New York.

60. A. Samuel, "Studies in Machine Learning Using the Game of Checkers, 2, - Recent Progress", IBM Journal (Nov. 1967). To be reprinted in Annual Review of Automatic Programming, M. Halpern (ed.) (Pergamon Press), in press.

61. E. Sandewall, "The GPS Expressed in LISP-A", Working Paper (Department of Computer Sciences, Uppsala, Oct. 1967).

62. O. G. Selfridge, "Pattern Recognition and Modern Computers", In: Proceedings of the 1955 Western Joint Computer Conference (1955).

63. H. A. Simon, "Experiments with a Heuristic Compiler", Journal of the Association for Computing Machinery 10 (Oct. 1963) 4.

64. H. A. Simon, "Motivational and Emotional Controls of Cognition", Psychological Review, (1967).

65. H. A. Simon, "Scientific Discovery and the Psychology of Problem Solving", In: R. G. Colodny (ed.), Mind and Cosmos (University of Pittsburgh Press, 1966).

66. H. A. Simon and E. A. Feigenbaum, "An Information-Processing Theory of Some Effects of Similarity, Familiarization, and Meaningfulness in Verbal Learning", Journal of Verbal Learning and Verbal Behavior 3 (Oct. 1964).

67. H. A. Simon and K. Kotovsky, "Human Acquisition of Concepts for Sequential Patterns", Psychological Review 70 (1963).

68. H. A. Simon and R. K. Sumner, "Pattern in Music", In: B. Kleinmuntz (ed.) Formal Representation of Human Judgment (Wiley, 1968).

69. J. Slagle, "Experiments with a Deductive Question Answering Program", Communications of the ACM 8 (1965) 12.

70. J. R. Slagle and P. Bursky, "Experiments with a Multi-Purpose, Theorem Proving Heuristic Program", Journal of the Association for Computing Machinery 15 (Jan. 1968) 1.

71. L. Tessler, H. Enea and K. M. Colby, "A Directed Graph Representation for Computer Simulation of Belief Systems", Mathematical Biosciences 2 (1968).

72. D. Waterman, "Machine Learning of Heuristics", Ph.D. dissertation (Artificial Intelligence Project, Stanford University, 1968, forthcoming).

73. J. Weizenbaum, "Contextual Understanding by Computers", In: P.A. Kolers and M. Eden (eds.) Recognizing Patterns: Studies in Living and Automatic Systems (MIT Press, 1968).

74. P. G. Williams, "Some Studies in Game Playing with a Digital Computer", Ph.D. dissertation (Computer Science Department, Carnegie-Mellon University, July 1965).