DRAFT
9-15-59
AN:hs

Project: Complex Information Processing (RPN-4133)

This is a joint research project carried on by RAND and the Carnegie Institute of Technology since fall, 1954. It now includes people from the University of Pittsburgh and the Bell Telephone Laboratories. Relationships are not formalized; work is carried on by informal coordination of the participants.

Personnel:

From RAND:

- E. A. Feigenbaum, consultant, graduate student at Carnegie Institute of Technology
- H. Kanter
- A. Newell
- J. C. Shaw
- H. A. Simon, consultant, Professor of Administration, Carnegie Institute of Technology
- F. Tonge

Elsewhere:

- J. Feldman, now Assistant professor of Business Administration, University of California, Berkeley.
- L. Gregg, Associate professor of Psychology, Carnegie Institute of Technology
- K. Laughery, graduate student at Carnegie Institute of Technology
- R. Lindsay, graduate student at Carnegie Institute of Technology
- G. Mealy, member of the research staff, Bell Laboratories
- W. Reitman, Assistant professor of Psychology and Administration, Carnegie Institute of Technology
- N. Saber, graduate student at the University of Pittsburgh.

Purposes and Objectives:

We are engaged in continuing, fundamental research into the information processes that underlie human intellectual, adaptive, and creative abilities. The problem is to discover the information processes required for such tasks as playing good chess, proving difficult mathematical theorems, discovering new scientific laws, understanding English prose, making military command decisions, framing a foreign policy, holding intelligent conversation, or writing novels. The variety of tasks indicates that we wish to understand the full array of intelligent processes, the particular task being a tactical issue. Our aim is to understand both the psychology of human intelligent behavior, and the varieties of effective problem-solving processes, independent of whether humans use them or not. Our bias is towards understanding human behavior, as having the more far-reaching and positive consequences.

The scientific justification for the project lies in the emergence of a new technique: the synthesis of intelligent systems on digital computers. The development of the digital computer and the maturation of the programming art have provided the possibility that intelligence, thinking, and problem solving can be studied in complete and explicit detail. The techniques of programming computers to behave intelligently are just emerging, and we are concerned with both developing the technique and using it.

The effort of the project divides rather naturally into five parts:

<u>Psychology of mental processes:</u> construction of theories of human higher mental processes by writing programs that effectively simulate the behavior of the thinking adult. This work includes coordinated experimental studies.

<u>Heuristic programs:</u> construction of computer programs to accomplish tasks that require intelligence, including the development of theories of problem solving.

<u>Information processing languages:</u> construction of languages for programming computers that are adapted to the complex processes required for intelligent behavior.

<u>Computer design:</u> design of computers that will more efficiently execute the information processing languages we are constructing.

<u>Education:</u> communication to the various professional groups affected by the research of the project, and the training of graduate students and others to do research in the area. This activity is not research; however, it is a significant fraction of our effort.

Significant Results:

The following very general statements seem justified, as describing the content of the detailed research of this project. Some qualifications should be kept in mind. First, the work of a number of others has contributed to the acceptance of these statements, although it is legitimate to ascribe much of the concrete evidence to this project. Second, these statements are still controversial. Third, the tasks that have been studied are highly formal — e.g., symbolic logic, chess — and the levels of intelligence attained are not great compared with the peak achievements of man, although they are within the range that average college students consider difficult without prior study. Finally, all these statements really open up to deeper enquiry what they purport to state with finality.

— Computers can be programmed to solve tasks requiring intelligence.

- Computers are general symbol-manipulating mechanisms, and not just number-manipulating mechanisms. The original picture of a computer as a numerical calculator is an historical accident due to the purposes for which computers were built, and to better understanding of of arithmetic processes in comparison to other symbolic processes.

- Programs can be constructed which simulate closely the behavior of humans solving problems. Such a program is a theory of how a human thinks in the same sense that a system of differential equations is a theory how a physical system behaves.

- Problem solving consists of sophisticated trial and error, utilizing many bits of partial information (called heuristics) to organize the search for solutions and to select fruitful paths for exploration. Although the range of information that can aid in the discovery of problem solutions is largely unexplored, there appears to be nothing unique, special, or mysterious about the problem-solving process either of humans or machines.

- If computers are to solve difficult problems they must be selective and sophisticated, rather than just fast. This is true despite the fact that most, if not all, problems can be solved in a simple way if sufficient processing is done. The amount of processing to be done in hard problems exceeds by all bounds any computer capacities attainable. In chess, for example, one can find the best move simply by examining all continuations. These number about 10^{120}, which is practically infinite. It is so large a number that to be able to do in a mocrosecond what one could previously do in a year (a speed factor of 10^{13}) provides no appreciable step toward feasibility.

Summary and Present Status of Research:

Heuristic programs

LT, the Logic Theorist (Newell, Shaw, Simon, 6,7): In 1956 we succeeded in getting a computer program — the Logic Theorist on JOHNNIAC — to prove theorems in the sentential calculus of Whitehead and Russell, a form of elementary symbolic logic. This program, with variations, attempted about 200 problems under a variety of conditions, with fair success. The Logic Theorist is a true problem-solving program, using some of the same techniques as an intelligent human would in the same situation. Probably it is the first program deserving the appellation intelligent. It provided conclusive demonstration that mechanisms could perform intelligently, laying to rest the proposition that computers are simply very fast but very stupid clerks. We are no longer working with the program, and are preparing a final summary of its behavior.

<u>The Chess Player</u> (Newell, Shaw, Simon, Kanter, 18): The chess program is built in modular fashion, with separate sets of sub-programs, called goals, corresponding to various features of the chess situation. Two goals, material balance and control of center, have been working for some time. Our experience with the program is quite favorable — it explores and evaluates much as we would like it to, with only an occasional flurry of "brute force" exploration when its heuristics fail to provide guidance. The program is very partial with only two goals, of course, and it handily loses to everyone who plays it (although it lasted 20 odd moves against a weak player).

We are just beginning to run the program with the third goal, development, which will enrich the play considerably. We expect the opening play to be quite appropriate, but the complexity of the program makes prediction risky. It will remain a fairly passive — and hence easy to beat — program until we add a goal that attempts to make threats such as forks and discovered attacks.

The program is slow (25 minutes per move at the moment), very complex, and too large for the JOHNNIAC's memory, so the profit from working with the program is marginal. We also know now (see GPS below) how to build much more flexible programs. However, it is sufficiently close to playing reasonable chess, that we are loathe to give it up quite yet.

<u>GPS, the General Problem Solver.</u> (Newell, Shaw, Simon, Kanter, 20.24): This program is also our main psychological effort. The program currently engages in two ~~kx~~ major kinds of problem-solving behavior: means-ends analysis, where it determines which of the operations that it can perform will lead in the direction of the desired result; and planning, where it simplifies the problem, solves the simplified problem, and then uses the solution as a guide in solving the original problem. GPS was originally designed to solve problems in symbolic logic, but is built so that different subject matters can be "plugged in."

We have GPS running on the means-ends part of logic; and have hand simulated its behavior on trigonometric identities and integration of simple functions with the aid of a table of integrals. We have also investigated, (24) whether various kinds of learning by GPS could be so phrased that they could be problems on which GPS could work. This effort is aimed at exploring the relation between learning and problem solving. We succeeded, via a very crude hand simulation of GPS, in demonstrating that GPS could learn a significant part of its own program when faced with a new task environment.

GPS is coded in IPL-IV for JOHNNIAC, and we are abandoning the attempt to continue with it on JOHNNIAC with its small memory and unreliable drum. We expect to recode GPS in IPL-V for the 704.

<u>Assembly Line Balancing</u> (Tonge, 14): This program (Fred Tonge's Ph.D. thesis at CIT) works on the following problem: to assign jobs to a series of work stations along a production line (assembling washing machines, say), given the time for each job and constraints on which jobs must precede~~d~~ other jobs. The objective is to set up a minimum number of work stations, each with about the same work

load, that will achieve a specified production rate. The constraints are sufficiently obnoxious to make the problem so far intractable to standard mathematical programming techniques. This heuristic program engages in a sophisticated search for good solutions: it first describes the constraints in terms that allow it to substitute more simplified and approximate problems for the original; it then works back and forth among the approximate problems until it can take into account all of the conditions of the total problem. The program is coded in IPL-IV and is running on JOHNNIAC. This program represents our only excursion so far into solving an applied management science problem.

Psychology of Mental Processes:

LT, the Logic Theorist (Newell, Shaw, Simon, 10): LT was constructed from the introspections of its designers to prove theorems. Nevertheless, in qualitative outline it shows many of the features of human problem solving.

GPS, the General Problem Solver (Newell, Shaw, Simon, 16,23): This program was developed, as a sequel to LT, to simulate the behavior of college students working on problems in symbolic logic. The basis for it is about 40 hours of protocols of human behavior: subjects talking aloud as much as possible while they worked. Our procedure is to take the protocol for a specific session and see if we can construct a variant of GPS that will explain as much as possible of the behavior evidenced in the protocol. This means adding new methods, adjusting priorities, etc., until the hand simulated trace of the program (a step-by-step statement of what it did) can be laid along side the protocol and correspondences found all the way along. Our version of GPS on JOHNNIAC has not gotten far enough along to let us simulate on the computer yet.

We are at the stage of explaining a bit of behavior — that is, seeing how good a fit we can obtain between a program and the given behavior. It is also necessary to try predicting with programs: Having fitted a program to one segment of behavior to see how well it simulates a part of behavior not yet examined. We have tried this informally once, with quite encouraging results. The program was fitted to a half hour of continuous behavior. It was then used to predict the first few minutes of the behavior on the next problem the subject tried (24 hours later). The program produced much of the details of exploration, but failed to predict the major choice of direction.

EPAM, Elementary Perceiver and Memorizer (Feigenbaum, Simon, 18,26): The problem is to explain the established psychological facts about how a person memorizes simple items — e.g., nonsense syllables — and makes elementary discriminations — e.g., differentiates letters of an unfamiliar alphabet. This area is of major concern to psychology: many facts are known, and a number of theories have been advanced to explain them. The aim here is to see how far considerations of information processing can go to explain these facts. A program has been written that can make new discriminations and form associations between them — the two major processes in a memorization experiment. It is coded in IPL-V, and has been running on the 704 this summer. The program agrees qualitatively with the major experimental findings on human behavior. This project is Ed Feigenbaum's Ph.D. thesis at CIT; it is completed and Ed should get his degree in the immediate future.

One reason for investigating this area is the ease with which memorization and discrimination are carried out by computers. Thus, it becomes a problem to understand why humans have so much difficulty. It is possible that the differences between humans and computers are assignable to basic differences in "hardware." More likely we are making a miscomparison: the computer preprogrammed for a fixed set of categories versus the human learning his own categories. Construction of programs that operate in the same situations and with the flexibility as humans should shed some light on this.

Binary Choice (Feldman, Newell, 22,25): The problem is to write programs for how humans predict sequences as a function of their experience with the sequence. A large number of protocols have been obtained of humans guessing a sequence of 200 binary digits and giving reasons for each guess. Programs have been defined that simulate the behavior in three of the protocols. These programs are highly similar, although differing in detail to fit the unique behavior of the individual subjects. They all generate hypotheses that explain the past behavior of the sequence, and use these to predict the next digit. The class of hypotheses consists of simple patterns, such as runs of a digit or simple alteration, plus a few notions like "the experimenter is trying to throw me off." Also, there is a general tendency toward "negative recency" — i.e., no state of affairs will continue indefinitely, so that eventually successful hypotheses will be discarded. The program for one of the subjects is substantially running in IPL-V on the 704; all the early work was done by hand simulation, since IPL-V was not yet ready. This project is Julian Feldman's Ph.D. thesis at CIT. He has obtained his degree, and is now at University of California in Berkeley, where he is continuing research on this problem.

Chess (Newell, Simon): We have protocols on humans analyzing chess situations, and have drawn some preliminary diagrams of the exploration humans conduct on the chess board.

Concept Attainment (Gregg, Laughery, Simon): The problem is to write programs for humans in the following task: the subject faces a black box with four switches (16 possible configurations). The experimenter has determined the sequence of configurations according to some pattern — e.g., the binary representations of the numbers 0 through 15, where 1 means "switch set left", and 0 means "switch set right", The subject tries switch settings until he gets each configuration, and must gradually induct the pattern that the experimenter has determined. The task bears many similarities to Julian Feldman's binary choice problem. Considerable experimental work has been done, and a program is being coded in IPL-V.

Information Processing Languages

IPL-I to IPL-IV (Newell, Shaw, 6,8,11,19): We have developed a sequence of languages for programming complex processes on digital computers, that are simultaneously the necessary tools for the programming we have done, and fundamental research into processing languages and how to mechanize them. These languages are concerned with the general manipulation of symbols, and not just arithmetic. In fact we have temporarily sacrificed the computer's built-in-facility to do arithmetic in order to focus on the processes required for general intellectual activity. These languages use the list as a unit of information rather than the isolated word (or number) as in current hardware. Thus operations like "delete this item from this list" and "join these two lists together" become the fundamental building blocks for programs, rather than "add these two numbers" as in current computers. We are using IPL-IV on JOHNNIAC; (see 19) chess, GPS, and the assembly line balancing program are coded in it. LT was coded in IPL-II (see 8); and IPL-I and IPL-III (see 6 and 11) were intermediate versions, never used for running programs.

IPL-V (Newell, Feigenbaum, Mealy, Saber, Tonge, 13,19): We are producing an information processing language, IPL-V, for the IBM 650, IBM 704, and IBM 709. Our aim in producing IPL-V is not to make a better language than our others, but to provide a well documented and polished language on "public" computers that our scientific colleagues can use. Work started well over a year ago with a language for the 650 (see 13). The 704 version, coded by George Mealy of Bell Laboratories and Ed Feigenbaum, is now working; although we are still in the processes of extracting small bugs and getting agreement with the capabilities stated in the manual. The 650 version, coded by Nick Saber of the University of Pittsburgh, is still being debugged. The 709 version is really an independent project by Bert Green, a psychologist at Lincoln Laboratory; and is apparently working. We are not yet ready to distribute the language, but will be using it intensively over the next few months. A final version of the manual will be available when we are fully satisfied that the language meets specifications. Some of the raison d'etre for IPL-V has departed with the appearance of two other list languages for the 704, one by Herb Gelernter of IBM (FLPL) and one by John McCarthy of MIT (LISP).

<u>Computer Design</u> (Shaw, Newell, Simon, Ellis, 15): We spent some time during the spring of 1958 analyzing how a computer should be organized for a language similar to IPL-IV (which we called IPL-VI). We are using the resulting paper (15) to obtain some information of the patent situation on computer programs and designs. Tom Ellis has been working on this.

<u>Education:</u>

We continue to put a major effort into education and communication. The work on IPL-V can be viewed primarily in this light, although we are now planning to use it ourselves. The graduate program at Carnegie Institute of Technology has already produced three Ph.D.'s (almost): Ed Feigenbaum, Julian Feldman, and Fred Tonge. We feel this is a significant contribution to scientific talent working in the general area. At least two others who were with us for a while at Carnegie have continued to work in the area: Carl Hensley, who is working on information retrieval problems at IBM; and David Lefkovitz, who is working on heuristic programs at the University of Pennsylvania. At Carnegie we have Bob Lindsay still working on his thesis,and several others at an earlier stage of development. Much of the interest of the students focuses on management science problems, such as scheduling, rather than on theories of human thinking.

Last year we gave an SSRC training institute at RAND on Simulation of Cognitive Processes. There is some evidence of tangible fruit. A student of Carl Hovland at Yale, named Earl Hunt, has a program for simulating a human concept attainment experiment, which he is coding in IPL-V. Bert Green, the psychologist who is coding IPL-V for the 709, was a participant. Bob Hamblin, also a participant, has received an SSRC grant to work on some small group experiments with an attempt to simulate their behavior with a program. Several others still have aspirations to work in this area — but are awaiting our tardy ~~delivery~~ delivery of a language.

During the last six months we had the following opportunities to talk before professional audiences: at the McKinsey Conference in Chicago ________________, Feb. 13, (Simon, 21); at the Cleveland Chapter of the ACM, February 10 (Newell); six lectures on simulation of cognitive processes at Columbia University, February 18 to 27 (Simon); as one of the lectures on Current Trends in Psychology at the University of Pittsburgh, March 12 (Simon, 23); at the Eastern Psychological Association meeting, Atlantic City, April 4, (Simon); at Lincoln Laboratory, April 6 (Newell); before the Carnegie Physical Society, April 13 and 27 (Newell); Western Psychological Association meeting, San Diego, April 17 (Tonge); at the Interdisciplinary Conference on Self Organizing Systems in Chicago, May 5 (Newell, 24); and at the International Conference ~~x~~ on Information Processing in Paris, June 19 (Newell, 20).

<u>Future Plans</u>

IPL-IV on JOHNNIAC has about outlived its usefulness; it will be retired soon, along with the problems coded in it. Assembly Line Balancing will be completed in the next few months; Chess within the next six months; and we have already decided to shift GPS to IPL-V on the 704 with 32K of fast storage. This shift will allow us to spend more time on IPL-VII, since it is apparent that we need better tools. Concurrently with work on ~~this~~ this next language we will consider various hardware possibilities for realizing it efficiently. (Any language can be realized interpretively on current computers, but may make very inefficient use of them.)

We will continue with our efforts to simulate human behavior, since the results have been very encouraging. Progress will be slow, since it takes very large amounts of effort to produce simulating programs, which again reveals that our tools are inadequate.

~~The~~

A book generally summarizing the research to date is finally underway with a first draft in being, and due for radical surgery.

<u>Project Effort</u>

The project is continuing in nature in nature, with Herschel Kanter, Allen Newell, and Cliff Shaw continuing at their present level. Fred Tonge will cease to spend very much effort on this project when he finishes the work on the line balancing problem. As consultants, Herb Simon will continue at about 3 or 4 days a month. Ed Feigenbaum will be in Europe on a Fulbright next year, so that his services will be minimal, although he will spend some time exploring European efforts in this general area.

Our JOHNNIAC time rose to around 50 hours a month, while we were trying to run GPS, Chess, and Line Balancing simultaneously. It should fall off considerably as these various efforts taper off. On the other hand, we will begin using more 704 time with IPL-V. We will not use large amounts of time until GPS gets going, and the timing on this is somewhat uncertain. (We may also be using the Westinghouse 704 in Pittsburgh for part of the IPL-V work, although the extent is still uncertain.)

Relationship to other RAND Projects:

With IPL-V available, we hope that other projects in the department can experiment a bit with these ideas. Fred Tonge has continued to work on the SACCS problem,

(Paul Armer or someone write this)

References

1. H. A. Simon, A Behavioral Model of Rational Choice, Quarterly Journal of Economics, February 1955.

2. A. Newell, The Chess Machine, P-620, Proceedings of The Western Joint Computer Conference, March 1955.

3. H. A. Simon, Rational Behavior and the Difficulty of the Environment, Psychological Review, March 1956.

4. H. A. Simon, and A. Newell, Models: Their Uses and Limitations, White (ed.), The State of the Social Sciences, University of Chicago, 1956.

5. A. Newell and H. A. Simon, Current Developments in Complex Information Processing, P-850, May 1956.

6. A. Newell, J. C. Shaw, and H. A. Simon, The Logic Theory Machine, P-868, IRE Transactions on information Theory, vol. IT-2,3 September 1956.

7. A. Newell, J. C. Shaw, and H. A. Simon, Empirical Explorations of the Logic Theory Machine, P-951, Proceedings of the Western Joint Computer Conference, February 1957.

8. A. Newell and J. C. Shaw, Programming the Logic Theory Machine, P-954, Proceedings of the Western Joint Computer Conference, February 1957.

9. A. Newell, J. C. Shaw, and H. A. Simon, Problem Solving in Humans and Computers, P-987, Carnegie Technical, March 1957.

10. A. Newell, J. C. Shaw and H. A. Simon, Elements of a Theory of Human Problem Solving, P-971 Psychological Review, March 1958.

11. A. Newell, Lectures on Heuristic Programs, in Applications of Logic to Advanced Digital Computer Programming, University of Michigan Summer Session Notes, August, 1957.

12. H. A. Simon and A. Newell, Heuristic Problem Solving: The Next Advance in Operations Research, Operations Research, February 1958.

13. C. Hensley, A. Newell, and F. Tonge, 650 Information Processing Language, C.I.P. Working Paper #9, January 1958, dittoed.

14. F. Tonge, Development of an Heuristic Program for an Assembly Line Balancing Problem, C.I.P., Working Paper #11, April 1958, dittoed.

15. J. C. Shaw, A. Newell, H. A. Simon, and T. O. Ellis, A Command Structure for Complex Information Processing, P-1277, Proceedings of the 1958 Western Joint Computer Conference, May 1958.

16. A. Newell, J. C. Shaw, and H. A. Simon, The Processes of Creative Thinking, P-1320, May 1958, to be published.

17. E. Feigenbaum and H. A. Simon, A Theory of the Serial Position Effect, C.I.P. Working Paper #14, May 1958, dittoed.

18. A. Newell, J. C. Shaw and H. A. Simon, Chess Playing Programs and the Problem of Complexity, P-1319, IBM Journal of Research and Development, 2,4, October 1958.

19. A. Newell, et al, Manual for IPL-IV and IPL-V, C.I.P. Working Paper #16, December, 1958.

20. A. Newell, J. C. Shaw, and H. A. Simon, Report on a General Problem-Solving Program, P-1584, January 1959.

21. H. A. Simon and A. Newell, What have Computers to do with Management, P-1708, May 1959, to be published.

22. J. Feldman, An Analysis of Predictive Behavior in a Two-Choice Situation, Unpublished Ph.D. dissertation, June 1959, Carnegie Institute of Technology.

23. A. Newell, H. A. Simon, The Simulation of Human Thought, P-1738, June 1959, to be published.

24. A. Newell, J. C. Shaw, and H. A. Simon, A Variety of Intelligent Learning in a General Problem Solver, P-1742, June 1959, to be published.

25. J. Feldman, A Note on the Negative Recency Hypothesis in the Prediction of a Series of Binary Symbols, 1959, Amer. J. Psychol., forthcoming.

26. E. Feigenbaum, An Information Processing Theory of Verbal ~~LEARNING~~ Learning, unpublished Ph.D. dissertation, 1959, Carnegie Institute of Technology.