COMPUTER SIMULATION OF HUMAN THINKING

Allen Newell
Computer Sciences Department
The RAND Corporation
Herbert A. Simon*
Consultant, The RAND Corporation

P-2276

April 20, 1961

To be published in SCIENCE, American Association for the Advancement of Science.

*Professor, Carnegie Institute of Technology

Reproduced by

The RAND Corporation • Santa Monica • California

SUMMARY

The use of computer programs as theories of human thinking and problem solving is illustrated by comparing a sample of human problem-solving behavior with the trace of a computer program, the General Problem Solver, instructed to solve the same problem as the human subject. This use of computers for nonnumerical simulation of symbol-manipulating processes offers a solution to the dilemma that psychology has faced--that the problems of fundamental importance to the field have not always been those that existing research techniques were equipped to handle. Computer simulation promises to provide a powerful tool for constructing and testing theories of complex cognitive behavior.

COMPUTER SIMULATION OF HUMAN THINKING

by Allen Newell and Herbert A. Simon [1]

The path of scientific investigation in any field of knowledge records a response to two opposing pulls. On the one side, a powerful attraction is exerted by "good problems"--questions whose answers would represent fundamental advances in theory or would provide the basis for important applications. On the other side, strong pulls are felt from "good techniques"--tools of observation and analysis that have proved to be incisive and reliable. The fortunate periods in a science are those in which these two pulls do not paralyze inquiry by their opposition, but cooperate to draw research into fruitful channels.

When this happy condition is not substantially satisfied, science is threatened by schism. Some of its investigators will insist on working on important problems with methods that are insufficiently powerful and lack rigor; others will insist on tackling problems that the available tools handle easily, however unimportant those problems may be.

The stress arising from the mismatch of ends and means is seldom completely absent from any science--examples could be provided from contemporary biology, meteorology or mathematics. But it has been blatantly apparent in the science of psychology. This is true even if we leave out of account the tremendously

---

[1] The authors wish to acknowledge their debt to J. C. Shaw, who has been their partner in the research described in this paper.

important practical problems that are posed for the field by its potential applications in the clinic, in education, and in many areas of social policy. In basic research the disparity has been strikingly visible. We can fairly classify most psychological research (and even most research psychologists) by its orientation on this issue. "Gestaltism" is one of the labels applied to question-oriented psychology; "behaviorism" is the commonest label applied to method-oriented psychology. It is no accident that research on human thinking, problem solving, personality, verbal behavior and social phenomena has tended to attract psychologists closest to the "Gestalt" end of the continuum; while research on animal behavior, physiological psychology, rote memory and simple motor skills has been primarily the domain of behaviorists.

It is commonly agreed that the dividing lines between the two points of view have become less clear since World War II. Several reasons might be given for this trend, but a full explanation would have to include the impact of new ideas drawn from "cybernetics" and the rapidly developing communications sciences. Complex electronic devices using feedback mechanisms to secure adaptive behavior have clarified concepts like "goal seeking" and "learning" and showed how such concepts could be made operational. This clarification has encouraged problem-oriented psychologists to give more precise operational meaning to terms that had been vague, and has encouraged technique-oriented psychologists to tackle problems that earlier appeared too complex for their tools.

The developments now taking place in psychology involve much more, however, than just a borrowing of new terms and new metaphors from other sciences. They involve the use of the electronic computer as a tool both for constructing theories and for testing them. Enough has already been learned about this tool and its potentialities to indicate that many of the "good problems" of psychology are now within reach of the "good techniques."

We should like to discuss here one of several important applications of the computer to psychological research--its use as a device for simulating the processes of human thinking. We shall not attempt a review of computer-based research in this one sphere of application, but will present, instead, a specific example drawn from our own work.

The Behavioral Phenomena

Let us begin with a sample of the phenomena we wish to explain. We seat a subject in the laboratory (a college sophomore, member of a ubiquitous species in psychological research). We present him with a problem, which we tell him is a problem in "recoding" symbolic expressions. We present a certain expression:

$$R.(-P \supset Q) \tag{1}$$

and ask him to obtain from it a second expression:

$$(QvP).R \tag{2}$$

by applying to the first expression a succession of rules of transformation drawn from a list which we also put before him.[2]

We ask the subject to announce aloud each rule that he wishes to apply and the expression that would result from its application. The experimenter then writes the new expression on a blackboard. We also ask the subject to talk aloud about what he is doing--"what he is thinking about." We record the entire session on tape.

Here is a sample of the first part of the protocol of a subject working on the problem stated above.[3] (The experimenters remarks to the subject are in italics.)

> I'm looking at the idea of reversing these two things now. *Thinking about reversing what?* The R's... then I'd have a similar group at the beginning, but that seems to be... I could easily leave something like that 'til the end, except than I'll...
>
> *Applying what rule?* Applying... for instance, 2. That would require a sign change.
>
> *Try to keep talking, if you can.* Well... then I look down at rule 3 and that doesn't look any too practical. Now 4 looks interesting. It's got three parts similar to that... and there are dots, so the connective... seems to work easily enough, but there's no switching of order.

---

[2] Readers familiar with symbolic logic will recognize the expressions and the rules, but the subjects were unacquainted with formal logic. They read the first expression, for example as: "(r horseshoe tilde-p) dot (tilde-r horseshoe q)." They made no use of the meanings of the expressions in their usual interpretation, but simply manipulated them as organized collections of symbols. If the reader wishes to follow the analysis in detail, he should adopt the same point of view.

[3] Subject 9; problem α 1.

> I need that P and a Q changed, so... I've got a horseshoe there. That doesn't seem practical any place through here. I'm looking for a way now, to get rid of that horseshoe. Ah... here it is, rule 6.
>
> So I'd apply rule 6 to the second part of what we have up there. Want to do that? Yeah. OK. to line 1 you apply rule 6. Line 2 is R.(PvQ). And now I'd use rule 1. Rule 1 on what part? You can use it with the entire expression or with the right part. I'd use it both places. Well, we'll do them one at a time... which do you want first? Well, do it with P and Q. R.(QvP). Now the entire expression? Yeah. On line 3, rule 1... you'd get (QvP).R. And... that's it. That's it all right, OK... that wasn't too hard.

The research problem, then, is to construct a theory of the processes causing the subject's behavior as he works on the problem, and to test the theory's explanation by comparing the behavior it predicts with the actual behavior of the subject. How can a computer help us solve this problem?

A Nonnumerical Computer Program as a Theory

An electronic digital computer is a device for adding, subtracting, multiplying, and dividing very rapidly. But it is now known to be much more than this. Speed in executing arithmetic operations was achieved by providing the computer with a program (usually stored in the computer memory) to govern the sequence of its operations--but designed to make that sequence conditional on the results of previous operations.

The instructions that make up the computer program, like the data on which it operates, are symbolic expressions. But while the data are normally interpreted as numbers, the instructions are interpreted as sequences of words--as sentences in the imperative mode. When the computer interprets the instruction "ADD A TO B," it produces the same result that a person would produce if he were asked in English to "add the number labelled 'A' to the number labelled 'B'."

We see that a computer is not merely a number-manipulating device; it is a symbol-manipulating device, and the symbols it manipulates may represent numbers, letters, words, or even non-numerical, non-verbal patterns. The computer has quite general capacities for reading symbols or patterns presented by appropriate input devices, storing symbols in memory, copying symbols from one memory location to another, erasing symbols, comparing symbols for identity, detecting specific differences between their patterns, and behaving conditionally on the results of its processes.

Let us return now to our human subject in the laboratory. His behavior, which we wish to explain, consists of a sequence of symbol emissions.[4] We can postulate that the processes going on inside his skin--involving sensory organs, neural tissue, and muscular movements controlled by the neural signals--are also symbol-manipulating processes. That is, patterns in various encodings can be detected, recorded, transmitted, stored

---

[4] This statement does not depend on the "thinking aloud" technique used in these experiments. It would be equally true if the subject had responded to the task in writing, or by pushing buttons. In all cases, his behavior can be interpreted as a sequence of symbol productions--in the last case a sequence of L's and R's, where "L" stands for "left button" and "R" for "right button."

copied, and so on, by the mechanisms of this system. We shall not defend the postulate in detail--its true defense lies in its power to explain the behavior. Nor shall we speculate in detail about the precise neurophysiological mechanisms and processes that correspond to terms like "symbol transmission," "stored symbol," or "copying" and the like.

Instead we shall adopt the tactic, highly successful in other sciences, of allowing several distinct levels of explanation, without for a moment denying that the mechanisms producing the behavior are ultimately reducible to physiological mechanisms and these, in turn to chemical and physical mechanisms. Just as we explain what goes on in the test tube by chemical equations and subsequently explain the chemical equations by means of the mechanisms of quantum physics, so we will attempt to explain what goes on in the course of thinking and problem solving by organizations of symbol-manipulation processes, and put to one side the task of explaining these processes in neurophysiological terms.

This approach to building a theory of complex behavior is depicted in Fig. 1. We shall be concerned with the top half of the figure--with reducing the overt behavior to information processes. If this reduction can be carried out, then a second body of theory will be needed to explain information processes by means of neurological mechanisms. Hopefully, tunneling through our mountain of ignorance from both sides will prove simpler than trying to penetrate the entire distance from one side only.

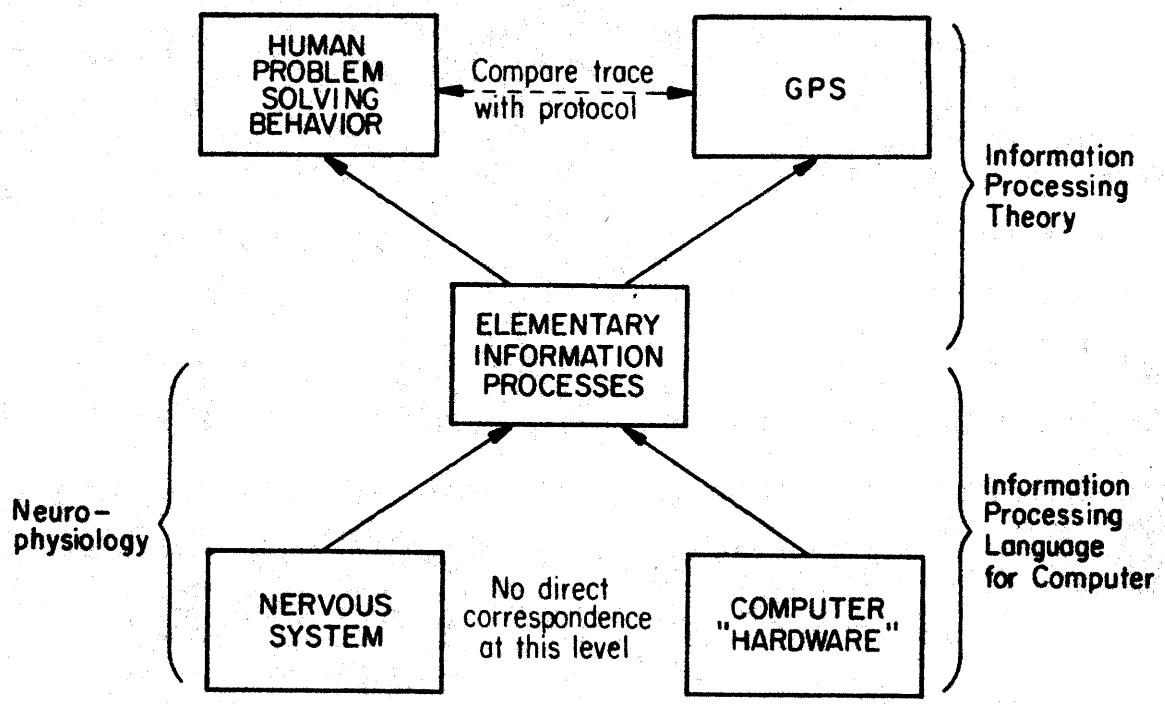


FIG. 1 - LEVELS IN AN INFORMATION PROCESSING THEORY OF HUMAN THINKING

Using Fig. 1, we begin to see how a computer can help with the half of the tunnelling operation that concerns us now. We postulate that the subject's behavior is governed by a program organized from a set of elementary information processes. We encode a set of subprograms (subroutines) for a digital computer, each of which executes a process corresponding to one of these postulated information processes. Then we undertake to write a program, compounded from these subroutines, that will cause the computer to behave in the same way as the subject behaves--to emit substantially the same stream of symbols--when both are given the same problem. If we succeed in devising a program that simulates the subject's behavior rather closely over a significant range of problem-solving situations, then we can regard the program as a theory of the behavior. How highly we will prize the theory depends, as with all theories, on its generality and parsimony--how wide a range of phenomena it explains and how economical of expression it is.

It can be seen that this approach makes no assumption that the "hardware" of computers and brains are similar, beyond the assumptions that both are general-purpose symbol-manipulating devices, and that the computer can be programmed to execute elementary information processes functionally quite like those executed by the brain. When we begin to theorize about the reduction of information processes to hardware, the brain and the computer (at least the computer used in this particular way) part company. (See, again, Fig. 1.) The former calls for a physiologist, the latter for an electrical engineer or physicist.

From a formal standpoint, a computer program used as a theory has the same epistemological status as a set of differential equations or difference equations used as a theory:

Given a set of initial and boundary conditions, the differential equations predict the successive states of the system at subsequent points in time.	Given a set of initial and subsequent environmental inputs, the computer program predicts the successive states of the system (the subject's symbol emissions and the state of his memory) at subsequent points in time.

With this use of the computer we construct "equations" for non-numerical symbol-manipulation phenomena without ever translating the phenomena into numerical form.

The General Problem Solver

Our attempt to explain the problem-solving protocol, excerpted above, and others like it takes the form of a computer program that we call the General Problem Solver (GPS). We can describe its structure.

The program has means for representing internally (i.e., in its memory) symbolic structures corresponding to the problem expressions, the rules for transforming expressions, and new expressions generated by applying the rules. The problem cited above is represented internally in the form of an expression that means "transform (1) into (2)." We shall call the symbolic structures corresponding to the logic expressions, *objects*; the structures corresponding to the problem statement and similar statements, *goals*. GPS attains goals by applying operators to objects, thus transforming them into new objects.

GPS has processes for applying operators to objects. GPS also has processes for comparing pairs of objects. These processes produce (internally) symbols that designate the differences between the objects compared. GPS has processes for generating new goals from given objects, operators, and differences.

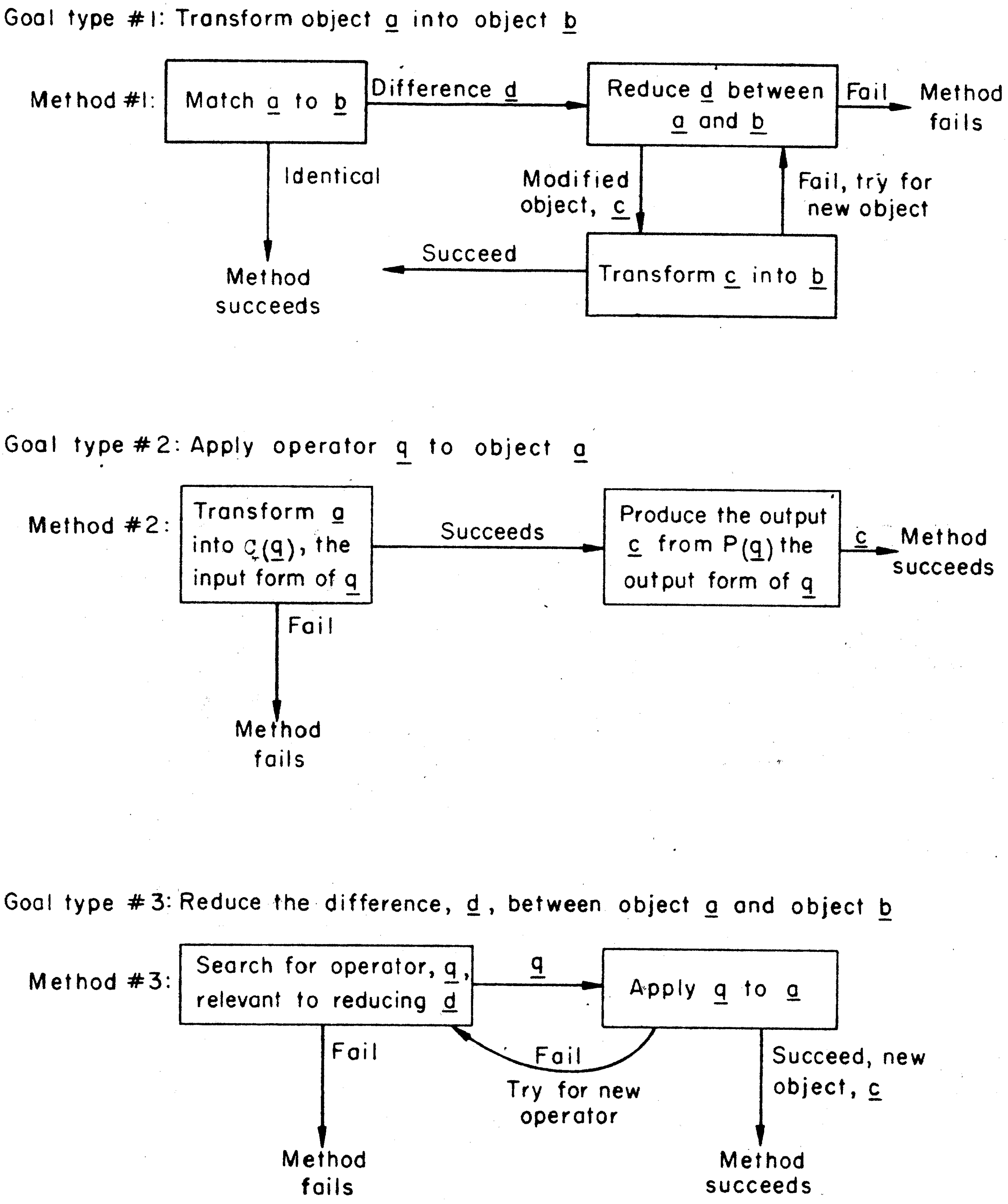
The processes of GPS are organized around three goal types and a small number of methods for attaining goals of these types (see Fig. 2).

1. *Transformation Goals*. These are of the form already illustrated: Transform object *a* into object *b*.

Method 1.[5] Compare *a* with *b* to find a difference, *d*,

---

[5] There is another method, the planning method, for attaining transformation goals. Space limits do not allow us to describe it in detail here. Briefly, it involves replacing the objects by corresponding abstracted objects, say, *a*" and *b*", then transforming *a*" into *b*" by means of the other methods, and using the resulting sequence of operations as a *plan* for transforming *a* into *b*.



Goal type #1: Transform object a into object b

Goal type #2: Apply operator q to object a

Goal type #3: Reduce the difference, d, between object a and object b

Fig. 2 — Methods for means—ends analysis

between them; if there is no difference, the problem is solved. Construct the goal of reducing difference *d* between *a* and *b*. If successful, the result will be a transformation of *a* into a new object, *a'*. Now construct the new goal of transforming *a'* into *b*. Attaining this goal will solve the original problem.

2. *Difference Reduction Goals*. As we have just seen, these are of the form: Reduce difference *d* between objects *a* and *b*.

Method 2. Find an operator, *q*, that is *relevant* to the difference in question (the meaning of relevance will be explained in a moment). Construct the goal of *applying* *q* to *a*. If successful, the result will be a transformation of *a* into a new object, *a'*, which will not have the difference, *d*, with *b*.

3. *Operator Application Goals*. These are of the form: Apply operator *q* to object *a*.

Method 3. Determine whether *a* meets the conditions for application of *q*. If so, apply *q*; if not, determine a difference between *a* and an object to which *q* is applicable. Construct the goal of reducing this difference. If successful, a new object *a'* will be produced, which is a modification of *a*. Now try to apply *q* to *a'*.

Thus, GPS is a program comprised of rather general processes for reasoning about ends (goals) and means (operators). It is general in the sense that the program itself makes no reference to the precise nature of the objects, differences, and operators with which it is dealing. Hence, its problem-solving capacities can be transferred from one kind of task to another by providing it with information about the kinds of objects, differences, and operators that characterize and describe the particular task environment it is to handle. Thus, to solve logic problems, it must be provided with a format for representing logic expressions, tests for the differences that must be recognized between pairs of expressions, and a list of the allowable operators. The rules of the game it is to play must be described to it.[6]

Testing the Theory

The question of the adequacy of GPS as an information processing theory of human problem solving can be asked at several levels of specificity. At the grossest level, we may ask whether GPS does, in fact, solve problems of some of the sorts that humans solve. This it demonstrably does. Hence, its program constitutes a system of mechanisms, constructed from elementary information processes, that is sufficient for solving some problems. It provides an unequivocal demonstration that a mechanism can solve problems by functional reasoning.

---

[6] At present, GPS is also provided with a "table of connections" that lists the operators that are potentially useful (relevant) for reducing each of the recognizable types of differences. We have indicated in another place how GPS could use its own problem-solving processes to construct the table of differences, and might even evolve a suitable set of differences if these were not provided to it in a new task environment (Newell, Shaw, and Simon, 1960).

The general kinds of end-means analysis that GPS uses are also the methods that turn up in the subjects' protocols. We have examined in fair detail some twenty protocols of subjects solving logic problems.[7] Virtually all the behavior in these protocols falls within the general framework of means-end analysis. The three goal types we have described account for three fourths of the subjects' goals, and the additional goal types that appear in the protocols are close relatives to those we have described. The three methods we have outlined represent the vast majority of the methods applied to these problems by the subjects. In addition, the planning method, mentioned in footnote 6, appears in several different forms in the protocols.

Protocols of human problem-solving behavior in a range of tasks--playing chess, solving puzzles, writing computer programs--contain many sequences of behavior that are also quite similar to the means-end analysis of GPS. We may cite, for example, the following excerpt from the thinking-aloud protocol of a chess player:

> Again I notice that one of his pieces is not defended, the Rook, and there must be ways of taking advantage of this. Suppose now, if I push the Pawn up at Bishop 4, if the Bishop retreats I have a Queen check and I can pick up the Rook. If the Bishop takes the Pawn then I can win a piece by simply again bringing either the Queen down with check, or Knight takes Bishop.

---

[7] For further discussion of the comparison of trace with protocols, see Newell and Simon, 1961(a) and 1961(b).

We cannot, of course, on the basis of this kind of evidence, conclude that GPS provides an adequate explanation for all these kinds of problem-solving behavior. Many other mechanisms may be involved besides those that are incorporated in GPS. Only when a program simulates the entire sequence of behaviors--for example, makes the same chess analysis as the human player--do we have any assurance that we have postulated a set of processes sufficient to produce the behavior in question.

These tests are still very general and do not take into account differences among the programs of different subjects. Obviously, not all subjects solve the problems in exactly the same way. The evidence presented thus far suggests that programs of most subjects share the general qualitative features of GPS, but there are variations in detail. We can subject the theory to further tests by seeing what modifications in GPS, if any, will enable us to predict, in detail, the symbolic behavior of a particular subject during some interval of his problem-solving activity.

In Fig. 3 we compare, in parallel columns, the protocol segment we introduced earlier with the output of a variant of GPS set to the task of solving the same problem. In the right-hand half of the figure we reproduce the human protocol; in the left-hand half we reproduce the trace of the program. The language of the subject is much less stylized than the language of the computer. To fit the theory, we must, for example, interpret a sentence like "I'm looking at the idea of reversing

COMPUTER TRACE

```
LO  (QVP).R
L1  R.(-P⊃Q)

GOAL 1  TRANSFORM L1 INTO LO
GO    GOAL 2  CHANGE POSITION IN L1
             GOAL 3  APPLY R1 TO L1  [A.B → B.A]
                    PRODUCES L2 (-P⊃Q).R

      GOAL 4  TRANSFORM L2 INTO LO
             GOAL 5  CHANGE POSITION IN LEFT L2
                    GOAL 6  APPLY R2 TO LEFT L2  [A⊃B → ~B⊃-A]
                           PRODUCES L3  (-Q⊃P).R

             GOAL 7  TRANSFORM L3 INTO LO
                    GOAL 8  CHANGE SIGN LEFT L3
                           NONE FOUND

             GOAL 5
                    GOAL 9  APPLY R3 TO L2  [A.A →  A]
                           REJECT, NOT DESIRABLE
                    GOAL 10  APPLY R4 to L2  [(A.B).C →  A.(B.C)]
                           REJECT, NOT DESIRABLE
                    GOAL 11  APPLY R5 to L2  [A.B →  -(-AV-B)]
                           REJECT, NOT DESIRABLE
                    GOAL 12  APPLY R7 TO L2  [A.(BVC) →  (A.B)VA.C)]
                           REJECT, NOT DESIRABLE
                    GOAL 13  APPLY R8 to L2  [A.B →  A]
                           REJECT, NOT DESIRABLE

             GOAL 5
                    GOAL 14  APPLY R1 TO LEFT L2  [AVB → BVA]
                           GOAL 15  CHANGE CONNECTIVE TO V IN LEFT L2
                                  GOAL 16  APPLY R6 TO LEFT L2 [A⊃B →~AVB]
                                         PRODUCES L4  (PVQ).R

                           GOAL 17  APPLY R1 TO LEFT L4  [AVB → BVA]
                                  PRODUCES L5  (QVP).R

             GOAL 18  TRANSFORM L5 INTO LO
                    IDENTICAL
```

PROTOCOL OF SUBJECT S9

(LO is expression to be obtained)
(L1 is expression given at start)
(Goal 1 is set by the experimenter.)

I'm looking at the idea of reversing these two things now. Thinking about reversing what? The R's... then I'd have a similar group at the beginning, but that seems to be...I could easily leave something like that 'til the end, except then I'll ...

Applying what rule? Applying,.. for instance, 2.

That would require a sign change.

Try to keep talking, if you can. Well... then I look down at rule 3 and that doesn't look any too practical. Now 4 looks interesting. It's got three parts similar to that...and there are dots, so the connective... seems to work easily enough, but there's no switching of order.

I need that P and a Q changed, so... I've got a horseshoe there. That doesn't seem practical any place through here. I'm looking for a way now, to get rid of that horseshoe. Ah... here it is, rule 6.

So I'd apply rule 6 to the second part of what we have up there. Want to do that? Yeah. OK, to line 1 you apply rule 6. Line 2 is R.(PVQ). And now I'd use rule 1. Rule 1 on what part? You can use it with the entire expression or with the right part. I'd use it both places. Well, we'll do them one at a time... which do you want first? Well, do it with P and Q. R.(QVP). Now the entire expression? Yeah. On line 3, rule 1... you'd get (QVP).R. And... that's it. That's it all right, OK.. that wasn't too hard.

FIG. 3

COMPARISON OF GPS TRACE WITH PROTOCOL OF SUBJECT S9 ON PROBLEM α1 these two things now," as equivalent to "Construct the difference reduction goal of eliminating the difference in position of corresponding subparts in objects L1 and L2." To make such a translation is, in practice, not too difficult, and having made it, we can determine in great detail the similarities and differences between the programs of the subject and the computer, respectively.

Let us consider some of the differences visible in the example at hand--differences that represent inadequacies of GPS, in its present form, as an accurate theory of the subject's behavior. Observe that the subject solves the entire problem in his head, and then asks the experimenter to write the actual transformations on the blackboard. GPS, in the version shown here, makes no provision for such a distinction between the internal and external worlds; hence, the trace corresponds only to the subject's covert (but verbalized) problem solving. For example, GPS and the subject both discover in the same sequence the correct rules for transforming the problem expression, but the subject "publicly" applied these rules in the reverse order.

Another difference, characteristic of these data, and such data in general, is that a number of things appear in the trace that have no correspondents in the human protocol--most prominently, the references here in the trace to rules 5, 7, and 8. We cannot tell whether these omissions indicate an error in the theory, or whether the subject noticed the rules in question but failed to mention them aloud.

In contrast to these differences, there is some striking correspondence in detail between computer trace and subject's protocol. First, in noticing differences between pairs of expressions, both GPS and the subject pay most attention to differences in the positions of symbols, next most attention to the presence or absence of "-" signs, and least attention to differences in connectives. This shows up, for example, in the refusal of both to apply Rule 2, after mentioning it, to reorder the expression, because applying the rule involves changing a sign. Second, of the several possible paths to the problem solution, both program and subject chose an application of Rule 6 and two applications of Rule 1.

These samples of success and failure will give the reader some indication of the kind of detailed comparison that can be made between the predictions of theory of computer models of this kind and actual human behavior. Much remains to be learned about how to make such comparisons and how to test their "goodness of fit." The fragmentary evidence we have obtained to date encourages us to think that GPS provides a rather good first approximation to an information processing theory of certain kinds of thinking and problem solving behavior. The processes of "thinking" can no longer be regarded as completely mysterious.

Conclusion

A digital computer is a general purpose symbol manipulating device. By writing appropriate programs for it, it can be made to produce symbolic output than can be compared with the stream of verbalizations of a human being who is thinking aloud while solving problems. The General Problem Solver is a computer program that is capable of simulating, in first approximation, human behavior in a narrow, but significant, problem domain.

The General Problem Solver is not the only existing program of this type. There is a program, the predecessor of GPS, that also discovers proofs for theorems, but only in symbolic logic (Newell and Simon, 1956). There are programs for proving theorems in geometry, (Gelernter and Rochester), for designing electric motors, generators, and transformers, (Goodwin), for writing music, (Hiller and Isaacson), for playing chess, (Newell, Shaw, and Simon, 1958). There are programs that learn--i.e., that modify themselves in various respects on the basis of experience (Samuel; Newell, Shaw, and Simon, 1957). We omit from the list those programs that make primary use of the computer's arithmetic capacities and that are not particularly humanoid even in the general organization of their processes. All of the programs listed, other than GPS, are limited to a single task environment, and none of them seeks to simulate the corresponding human processes in detail. Nevertheless, they all have extremely similar underlying structures, involving selective search for possible solutions based on rules of thumb, or heuristics. This communality provides further support for the basic correctness of the approach illustrated by GPS in the construction of a theory of human thinking.

In our discussion, we have limited ourselves to problem-solving programs. Several recent investigations undertake to simulate other kinds of human cognitive activity that have been studied in the psychological laboratory. Feldman has written a simulation program for partial reinforcement experiments; Feigenbaum, and Feigenbaum and Simon have written a program (EPAM) that simulates subjects' behavior in rote memory experiments; Hunt and Hovland and Laughery and Gregg have written programs that simulate concept forming behavior. There are now a score or more of research psychologists who are constructing and testing information processing theories of cognitive processes, formulating their theories as computer programs, and testing them by comparing the computer simulations with the protocols of human subjects.

Psychology has discovered an important new tool whose power appears to be commensurate with the complexity of the phenomena the science seeks to explain. As our skills in using this new tool develop, we may expect that the paralyzing conflict between the good problems in psychology and the good techniques will be very much ameliorated.

REFERENCES

1. Feigenbaum, E. A., "The Simulation of Verbal Learning Behavior," Proceedings of the 1961 Western Joint Computer Conference, 1961, pp. 121-132.
2. Feldman, J., "Simulation of Behavior in the Binary Choice Experiment," Proceedings of the 1961 Western Joint Computer Conference, 1961, pp. 133-144.
3. Gelernter, H. and N. Rochester, "Intelligent Behavior in Problem-Solving Machines," IBM Journal of Research and Development, Vol. 2, October, 1958, pp. 336-345.
4. Goodwin, G. L., "Digital Computers Tap Out Designs for Large Motors Fast," Power, April, 1958.
5. Hiller, L. A. and L. M. Isaacson, Experimental Music, McGraw-Hill, New York, 1959.
6. Hovland, C. I. and E. B. Hunt, "Programming a Model of Human Concept Formulation," Proceedings of the 1961 Western Joint Computer Conference, 1961.
7. Newell, A., Shaw, J. C., and H. A. Simon, "Empirical Explorations of the Logic Theory Machine," Proceedings of the 1957 Western Joint Computer Conference, 1957, pp. 218-230.
8. Newell, A., Shaw, J. C., and H. A. Simon, "Chess Playing Programs and the Problem of Complexity," IBM Journal of Research and Development, Vol. 2, October, 1958, pp. 320-335.
9. Newell, A., Shaw, J. C., and H. A. Simon, "A General Problem Solving Program for a Computer," Computers and Automation, Vol. 8, July, 1959, pp. 10-17.
10. Newell, A., Shaw, J. C., and H. A. Simon, "A Variety of Intelligent Learning in a General Problem Solver," in M. C. Yovits and S. Cameron, eds., Self-Organizing Systems, Pergamon, New York, 1960, pp. 153-189.
11. Newell, A., and H. A. Simon, "The Logic Theory Machine," IRE Transactions on Information Theory, IT-2; #3, September, 1956, pp. 61-79.
12. Newell, A., and H. A. Simon, "The Simulation of Human Thought," in Current Trends in Psychological Theory, University of Pittsburgh Press, Pittsburgh, 1961(a), pp. 152-179.

13. Newell, A., and H. A. Simon, "GPS, A Program that Simulates Human Problem Solving," to be published in the *Proceedings of the Conference on Learning Automata*, Karlsruhe, April 10-14, 1961(b).

14. Samuel, A. L., "Some Studies in Machine Learning Using the Game of Checkers," *IBM Journal of Research and Development*, Vol. 3, July, 1959, pp. 210-229.