In H. Billing (Ed.),
Lernende automaten. Munchen:
R. Oldenbourg, 1961.

#191

GPS, A Program that Simulates Human Thought

By A. Newell, Santa Monica (Calif.) and H. A. Simon, Pittsburgh (Pa.)

Working from the "protocol" recording the behaviour of a testperson solving a logic problem, a computer program called GPS (for General Problem Solver) is developped, which leads to a psychological theory of human problem solving. It is shown, how data giving the same results as derived from the protocol yields to an analysis in terms of a program characterized by a recursive structure of goals and subgoals.

Von dem „Protokoll" ausgehend, das das Verhalten einer Versuchsperson beim Lösen eines logischen Problems angibt, wird ein Rechnerprogramm, genannt „GPS" (General Problem Solver), entwickelt, das zu einer psychologischen Theorie des Problemlösens beim Menschen führt. Es wird gezeigt, wie sich aus den Daten, die das gleiche Ergebnis zeigen, wie es aus dem Protokoll abzulesen ist, sich eine Analyse ergibt, die als Programm vorliegt und durch eine rekursive Struktur von Haupt- und Nebenzielsetzungen gekennzeichnet ist.

1. An Introduction

This paper is concerned with the psychology of human thinking. It sets forth a theory to explain how some humans try to solve some simple formal problems. The research from which the theory emerged[1] is intimately related to the field of information processing and the construction of intelligent automata, and the theory is expressed in the form of a computer program. The rapid technical advances in the art of programming digital computers to do sophisticated tasks have made such a theory feasible.

It is often argued that a careful line must be drawn between the attempt to *accomplish* with machines the same tasks that humans perform, and the attempt to *simulate* the processes humans actually use to accomplish these tasks. The program discussed in the paper, GPS (General Problem Solver), maximally confuses the two approaches — with mutual benefit. GPS has previously been described as an attempt to build a problem-solving program [1, 2], and in our own research it remains a major vehicle for exploring the area of artificial intelligence. Simultaneously, variants of GPS provide simulations of human

[1] We would like to express our indebtedness to J. C. Shaw, who has been our colleague in most of our research into complex information processes, including the GPS program which forms the basis of this paper.

behavior [8]. It is this latter aspect — the use of GPS as a theory of human problem-solving — that we want to focus on exclusively here, with special attention to the relation between the theory and the data.

As a context for the discussion that is to follow, let us make some brief comments on some history of psychology. At the beginning of this century the prevailing thesis in psychology was Associationsim. It was an atomistic doctrine, which postulated a theory of hard little elements, either sensations or ideas, that became hooked or associated together without modification. It was a mechanistic doctrine, with simple fixed laws of continuity in time and space to account for the formation of new associations. Those were its assumptions. Behavior proceeded by the stream of associations: Each association produced its successors, and acquiied new attachments with the sensations arriving from the environment.

In the first decade of the century a reaction developed to this doctrine through the work of the *Würzburg* school. Rejecting the notion of a completely self-determining stream of associations, it introduced the task (*Aufgabe*) as a necessary factor in describing the process of thinking. The task gave direction to thought. A note-worthy innovation of the *Würzburg* school was the use of systematic introspection to shed light on the thinking process and the contents of consciousness. The result was a blend of mechanics and phenomenalism, which gave rise in turn to two divergent antitheses, Behaviorism and the *Gestalt* movement.

The behavioristic reaction insisted that introspection was a highly unstable, subjective procedure, whose futility was amply demonstrated in the controversy on imageless thought. Behaviorism reformulated the task of psychology as one of explaining the response of organisms as a function of the stimuli impinging upon them and measuring both objectively. However, Behaviorism accepted, and indeed reinforced, the mechanistic assumption that the connections between stimulus and response were formed and maintained as simple, determinate functions of the environment.

The *Gestalt* reaction took an opposite turn. It rejected the mechanistic nature of the associationist doctrine but maintained the value of phenomenal observation. In many ways it continued the *Würzburg* school's insistence that thinking was more than association — thinking has direction given to it by the task or by the set of the subject. *Gestalt* psychology elaborated this doctrine in genuinely new ways in terms of holistic principles of organization.

Today psychology lives in a state of relatively stable tension between the poles of Behaviorism and *Gestalt* psychology. All of us have internalized the major lessons of both: We treat sceptically the subjective elements in our experiments and agree that all notions must eventually be made operational by means of behavioral measures. We also recognize that a human being is a tremendously complex, organized system, and that the simple schemes of modern behavioristic psychology seem hardly to reflect this at all.

2. An Experimental Situation

In this context, then, consider the following situation. A human subject, a student in engineering in an American college, sits in front of a blackboard on which are written the following expressions:

$$(R \supset -P) . (-R \supset Q) \quad | \quad -(-Q . P) .$$

Objects are formed by building up expressions from letters (P, Q, R, ...) and connectives . (dot), v (wedge), ⊃ (horseshoe), and – (tilde). Examples are P, –Q, P v Q, –(R ⊃ S) . –P; ––P is equivalent to P throughout.

Twelve rules exist for transforming expressions (where A, B, and C may be any expressions or subexpressions):

R 1. $A . B \rightarrow B . A$
$A \vee B \rightarrow B \vee A$

R 2. $A \supset B \rightarrow -B \supset -A$

R 3. $A . A \leftrightarrow A$
$A \vee A \leftrightarrow A$

R 4. $A . (B . C) \leftrightarrow (A . B) . C$
$A \vee (B \vee C) \leftrightarrow (A \vee B) \vee C$

R 5. $A \vee B \leftrightarrow -(-A . -B)$

R 6. $A \supset B \leftrightarrow -A \vee B$

R 7. $A . (B \vee C) \leftrightarrow (A . B) \vee (A . C)$
$A \vee (B . C) \leftrightarrow (A \vee B) . (A \vee C)$

R 8. $A . B \rightarrow A$
$A . B \rightarrow B$ Applies to main expression only.

R 9. $A \rightarrow A \vee X$ Applies to main expression only.

R 10. $\left.\begin{matrix} A \\ B \end{matrix}\right\} \rightarrow A . B$ A and B are two main expressions.

R 11. $\left.\begin{matrix} A \\ A \supset B \end{matrix}\right\} \rightarrow B$ A and A ⊃ B are two main expressions.

R 12. $\left.\begin{matrix} A \supset B \\ B \supset C \end{matrix}\right\} \rightarrow A \supset C$ A⊃B and B⊃C are two main expressions.

Example, showing subject's entire course of solution on problem:

1. $(R \supset -P) . (-R \supset Q)$	$-(-Q . P)$
2. $(-R \vee -P) . (R \vee Q)$	Rule *6* applied to left and right of 1.
8. $(-R \vee -P) . (-R \supset Q)$	Rule *6* applied to left of 1.
4. $R \supset -P$	Rule *8* applied to 1.
5. $-R \vee -P$	Rule *6* applied to 4.
6. $-R \supset Q$	Rule *8* applied to 1.
7. $R \vee Q$	Rule *6* applied to 6.
8. $(-R \vee -P) . (R \vee Q)$	Rule *10* applied to 5. and 7.
9. $P \supset -R$	Rule *2* applied to 4.
10. $-Q \supset R$	Rule *2* applied to 6.
11. $P \supset Q$	Rule *12* applied to 6. and 9.
12. $-P \vee Q$	Rule *6* applied to 11.
18. $-(P . -Q)$	Rule *5* applied to 12.
14. $-(-Q . P)$	Rule *1* applied to 18. QED.

Figure 1. The Task of Symbolic Logic

This is a problem in elementary symbolic logic, but the student does not know it [4]. He does know that he has twelve rules for manipulating expressions containing letters connected by "dots" (.), "wedges" (v), "horseshoes" (⊃), and "tildes" (—), which stand respectively for "and", "or", "implies", and "not". These rules, given in Fig. 1, show that expressions of certain forms (at the tails of the arrows) can be transformed into expressions of somewhat different form (at the heads of the arrows). Double arrows indicate transformations can take place in either direction. The subject has practiced applying the rules, but he has previously done only one other problem like this. The experimenter has instructed him that his problem is to obtain the expression in the upper right corner from the expression in the upper left corner using the twelve rules. At any time the subject can request the experimenter to apply one of the rules to an expression that is already on the blackboard. If the transformation is legal, the experimenter writes down the new expression in the left-hand column, with the name of the rule in the right-hand column beside it. The subject's actual course of solution is shown beneath the rules in Fig. 1.

The subject was also asked to talk aloud as he worked; his comments were recorded and then transcribed into a "protocol", — i.e., a verbatim record of all that he or the experimenter said during the experiment. The initial section of this subject's protocol is reproduced in Fig. 2.

"Well, looking at the left hand side of the equation, first we want to eliminate one of the sides by using rule *8*. It appears too complicated to work with first. Now — no, — no, I can't do that because I will be eliminating either the Q or the P in that total expression. I won't do that at first. Now I'm looking for a way to get rid of the horseshoe inside the two brackets that appear on the left and right sides of the equation. And I don't see it. Yeh, if you apply rule *6* to both sides of the equation, from there I'm going to see if I can apply rule *7*."

Experimenter writes: 2 nd. $(-R \vee -P) \cdot (R \vee Q)$

"I can almost apply rule *7*, but one R needs a tilde. So I'll have to look for another rule. I'm going to see if I can change that R to a tilde R. As a matter of fact, I should have used rule *6* on only the left hand side of the equation. So use rule *6*, but only on the left-hand side."

Experimenter writes: 3 rd. $(-R \vee -P) \cdot (-R \supset Q)$

"Now I'll apply rule *7* as it is expressed. Both — excuse me, excuse me, it can't be done because of the horseshoe. So — now I'm looking — scanning the rules here for a second, and seeing if I can change the R to —R in the second equation, but I don't see any way of doing it." (Sigh.) "I'm just sort of lost for a second."

Figure 2. Subject's Protocol on First Part of Problem

3. The Problem of Explanation

It is now proposed that the protocol of Fig. 2 constitutes data about human behavior that are to be explained by a psychological theory. But what are we to make of this? Are we back to the introspections of the *Würzburgers*? And how are we to extract information from the behavior of a single subject when we have not defined the operational measures we wish to consider?

There is little difficulty in viewing this situation through behavioristic eyes. The verbal utterances of the subject are as much behavior as would be his arm movements or galvanic skin responses. The subject was not introspecting; he was simply emitting a continuous stream of verbal behavior while solving the problem. Our task is to find a model of the human problem solver that explains the salient features of this stream of behavior. This stream contains not only the subject's extemporaneous comments, but also his commands to the experimenter, which determine whether he solves the problem or not.

Although this way of viewing the behavior answers the questions stated above, it raises some of its own. How is one to deal with such variable behavior? Isn't language behavior considered among the most complex human behavior? How does one make reliable inferences from a single sample of data on a single subject?

The answers to these questions rest upon the recent, striking advances that have been made in computers, computer programming and artificial intelligence. We have learned that a computer is a general manipulator of symbols — not just a manipulator of numbers. Basically, a computer is a transformer of patterns. By suitable devices, most notably its addressing logic, these patterns can be given all the essential characteristics of linguistic symbols. They can be copied and formed into expressions. We have known this abstractly since TURING's work in the mid-thirties, but it is only recently that computers have become powerful enough to let us actually explore the capabilities of complex symbol manipulating systems.

For our purpose here, the most important branch of these explorations is the attempt to construct programs that solve tasks requiring intelligence. Considerable success has already been attained [5, 6, 7, 8, 9, 10, 11]. These accomplishments form a body of ideas and techniques that allow a new approach to the building of psychological theories. (Much of the work on artificial intelligence, especially our own, has been partly motivated by concern for psychology; hence, the resulting rapprochement is not entirely coincidental).

We may then conceive of an intelligent program that manipulates symbols in the same way that our subject does — by taking as inputs the symbolic logic expressions, and producing as outputs a sequence of rule applications that coincides with the subject's. If we observed this program in operation, it would be considering various rules and evaluating various expressions, the same sorts of things we see expressed in the protocol of the subject. If the fit of such a program were close enough to the overt behavior of our human subject — i.e., to the protocol — then it would constitute a good theory of the subject's problem solving.

Conceptually the matter is perfectly straightforward. A program prescribes in abstract terms (expressed in some programming language) how a set of symbols in a memory is to be transformed through time. It is completely analogous to a set of difference equations that prescribes the transformation of a set of numbers through time. Given enough information about an individual, a program could be written that would describe the symbolic behavior of that individual. Each individual would be described by a different program, and those aspects of human problem solving that are not idiosyncratic would emerge as the common structure and content of the programs of many individuals.

But is it possible to write programs that do the kinds of manipulation that humans do? Given a specific protocol, such as the one of Fig. 2, is it possible to induct the program of the subject? How well does a program fit the data? The remainder of the paper will be devoted to answering some of these questions by means of the single example already presented. We will consider only how GPS behaves on the first part of the problem, and we will compare it in detail with the subject's behavior as revealed in the protocol. This will shed considerable light on how far we can consider programs as theories of human problem solving.

4. The GPS Program

We will only briefly recapitulate the GPS program, since our description will add little to what has already been published [1, 2]. GPS deals with a task environment consisting of *objects* which can be transformed by various *operators*; it detects *differences* between objects; and it organizes the information about the task environment into *goals*. Each goal is a collection of information that defines what constitutes goal attainment, makes available the various kinds of information relevant to attaining the goal, and relates the information to other goals. There are three types of goals:

(*1*) Transform object A into object B,

(*2*) Reduce difference D between object A and object B,

(*3*) Apply operator Q to object A.

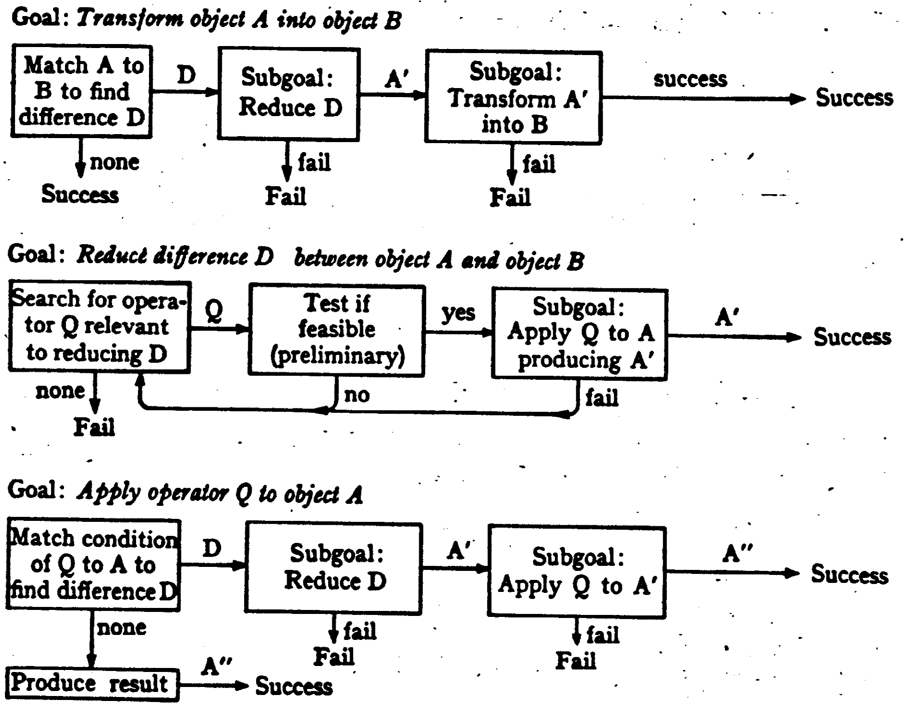
For the task of symbolic logic, the objects are logic expressions; the operators are the twelve rules (actually the specific variants of them); and the differences are expressions like "change connective" or "add a term". Thus the objects and operators are given by the task; whereas the differences are something GPS brings to the problem. They represent the ways of relating operators to their respective effects upon objects.

Basically, the GPS program is a way of achieving a goal by setting up subgoals whose attainment leads to the attainment of the initial goal. GPS has various schemes, called methods, for doing this. Three crucial methods are presented in Fig. 8, one method associated with each goal type. Thus, to transform an object A into an object B, the objects are first matched — put into correspondence and compared element by element. If the match reveals a difference, D, between the two objects, then a subgoal is set up to reduce this difference. If this subgoal is attained, a new object, A′, is produced which (hopefully) no longer has the difference D when compared with object B. Then a new subgoal is created to transform A′ into B. If the transformation succeeds, the entire goal has been attained in two steps: from A to A′ and from A′ to B.

If the goal is to reduce the difference between two objects, the first step is to find an operator that is relevant to this difference. Relevance here means that the operator affects objects with respect to the difference. Operationally, relevance can be determined by applying the matching process already used to the input and output forms of the operators, due account being taken of variables. The results can be summarized in a table of connections, as shown in Fig. 8, which lists for each difference the operators that are relevant to it. This table also lists the differences that GPS recognizes. (This set is somewhat different from the one given in [1]; it corresponds to the program we will deal with in this paper.) If a relevant operator, Q, is found, it is subjected to a preliminary test of feasibility, one version of which is given in Fig. 8. If the operator passes this test, a subgoal is set up to apply the operator to the object. If the operator is successfully applied, a new object, A′, is produced which is a modification of the original one in the direction of reducing the difference. (Of course, other modifications may also have occurred which nullify the usefulness of the new object.)

If the goal is to apply an operator, the first step is to see if the conditions of the operator are satisfied. The preliminary test above by no means guarantees this. If the conditions are satisfied, then the output A″, can be generated. If the conditions are not satisfied, then some difference, D, has been detected and a subgoal is created to reduce this difference, just as with the transform goal. Similarly, if a modified object. A′, is obtained, a new subgoal is formed to try to apply the operator to this new object.

These methods form a recursive system that generates a tree of subgoals in attempting to attain a given goal. For every new difficulty that is encountered a new subgoal is created to overcome this difficulty. GPS has a number of tests it applies to keep the expansion of this goal tree from proceeding in unprofitable directions. The most important of these is a test which is applied to new subgoals differences. GPS contains an ordering of the differences, so that some


Goal: *Transform object A into object B*
Match A to B to find difference D
D
Subgoal: Reduce D
A′
Subgoal: Transform A′ into B
success
Success
none
Success
fail
Fail
fail
Fail
Goal: *Reduce difference D between object A and object B*
Search for operator Q relevant to reducing D
Q
Test if feasible (preliminary)
yes
Subgoal: Apply Q to A producing A′
A′
Success
none
Fail
no
fail
Goal: *Apply operator Q to object A*
Match condition of Q to A to find difference D
D
Subgoal: Reduce D
A′
Subgoal: Apply Q to A′
A″
Success
none
fail
Fail
fail
Fail
Produce result
A″
Success

For the logic task of the text:

Feasibility test (preliminary):

Is the main connective the same? (E. g., A · B → B fails against P v Q)
Is the operator too big? (E. g., (A v B) · (A v C) → A v (B · C) fails against P · Q)
Is the operator too easy? (E. g., A → A · A · A applies to anything)
Are the side conditions satisfied? (E. g., *R8* applies only to main expressions)

Table of connections

	R1	*R2*	*R3*	*R4*	*R5*	*R6*	*R7*	*R8*	*R9*	*R10*	*R11*	*R12*
Add terms			●				●		●	●	●	●
Delete terms			●				●	●			●	●
Change connective					●	●	●					
Change sign					●							
Change lower sign		●			●	●						
Change grouping							●					
Change position	●	●										

● means some variant of the rule is relevant. GPS will pick the appropriate variant.

Figure 3. Methods for GPS

differences are considered easier than others. This ordering is given by the table of connections in Fig. 8, which lists the most difficult differences first. GPS will not try a subgoal if it is harder than one of its supergoals. It will also not try a goal if it follows an easier goal. That is, GPS insists on working on the hard differences first and expects to find easier ones as it goes along. The other tests that GPS applies involve external limits (e. g., a limit on the total depth of a goal tree it will tolerate), and whether new objects or goals are identical to ones already generated.

5. GPS on the Problem

The description we have just given is adequate to verify the reasonableness, although not the detail, of a trace of GPS's behavior on a specific problem. (In particular we have not described how the two-lines rules, R 10 through R 12, are handled, since they do not enter into the protocol we are examining.) In Fig. 4, we give the trace on the initial part of problem D 1. Indentation is used to indicate the relation of a subgoal to a goal. Although the methods are not shown, they can clearly be inferred from the goals that occur.

The initial problem is to transform L 1 into L 0. Matching L 1 to L 0 reveals that there are R's in L 1 and no R's in L 0. This difference leads to the formulation of a reduce goal, which for readability has been given its functional name, *Delete*. The attempt to reach this goal leads to a search for rules which finds rule 8. Since there are two forms of rule 8, both of which are admissible, GPS chooses the first. (Variants of rules are not indicated, but can be inferred easily from the trace.) Since rule 8 is applicable, a new object, L 2, is produced. Following the method for transform goals, at the next step a new goal has been generated: to transform L 2 into L 0. This in turn leads to another reduce goal: to restore a Q to L 2. But this goal is rejected by the evaluation, since adding a term is more difficult than deleting a term. GPS then returns to goal 2 and seeks another rule which will delete terms. This time it finds the other form of rule 8 and goes through a similar excursion, ending with the rejection of goal 8 altogether.

Returning again to goal 2 to find another rule for deleting terms, GPS obtains rule 7. It selects the variant (A v B) . (A v C) → A v (B . C), since only this one both decrease terms and has a dot as its main connective. Rule 7 is not immediately applicable; GPS first discovers that there is a difference of connective in the left subexpression, and then that there is one in the right subexpression. In both cases it finds and applies rule 6 to change the connective from horseshoe to wedge, obtaining successively L 4 and L 5. But the new expression reveals a difference in sign, which leads again to rule 6 – that is, to the same rule as before, but perceived as accomplishing a different function. Rule 6 produces L 6, which happens to be identical with L 4 although GPS does not notice the identity here. This leads, in goal 19, to the difference in connective being redetected;

```
L0 -(-Q.P)
L1 (R⊃-P). (-R⊃Q)

GOAL 1 TRANSFORM L1 INTO L0
     GOAL 2 DELETE R FROM L1
          GOAL 3 APPLY R8 TO L1
              PRODUCES L2 R⊃-P

   GOAL 4 TRANSFORM L2 INTO L0
        GOAL 5 ADD Q TO L2
            REJECT

   GOAL 2
        GOAL 6 APPLY R8 TO L1
            PRODUCES L3 -R⊃Q

   GOAL 7 TRANSFORM L3 INTO L0
        GOAL 8 ADD P TO L3
            REJECT

   GOAL 2
        GOAL 9 APPYL R7 to L1
            GOAL 10 CHANGE CONNECTIVE TO V IN LEFT L1
                 GOAL 11 APPLY R6 to LEFT L1
                      PRODUCES L4 (-RV-P). (-R⊃Q)

            GOAL 12 APPLY R7 to L4
                 GOAL 13 CHANGE CONNECTIVE TO V IN RIGHT L4
                      GOAL 14 APPLY R6 TO RIGHT L4
                          PRODUCES L5 (-RV-P). (RvQ)

                 GOAL 15 APPLY R7 TO L5
                      GOAL 16 CHANGE SIGN OF LEFT RIGHT L5
                          GOAL 17 APPLY R6 TO RIGHT L5
                               PRODUCES L6 (-RV-P). (-R⊃Q)

                      GOAL 18 APPLY R7 TO L6
                          GOAL 19 CHANGE CONNECTIVE TO V
                                  IN RIGHT L6
                               REJECT
                      GOAL 16
                          NOTHING MORE
                 GOAL 13
                      NOTHING MORE
            GOAL 10
                 NOTHING MORE
```

Figure 4. Trace of GPS on First Part of Problem

whereupon the goal is finally rejected as representing no progress over goal 18. Further attempts to find alternative ways to change signs or connectives fail to yield anything. This ends the episode.

6. Comparison of the GPS Trace with the Protocol

We now have a highly detailed trace of what GPS did. What can we find in the subject's protocol that either confirms or refutes the assertion that this program is a detailed model of the symbol manipulations the subject is carrying out? What sort of correspondence can we expect? The program does not provide us with an English language output that can be put into one-to-one correspondence with the words of the subject. We have not even given GPS a goal to "do the task and talk at the same time", which would be a necessary reformulation if we were to attempt a correspondence in such detail. On the other hand, the trace, backed up by our knowledge of how it was generated, does provide a complete record of all the task content that was considered by GPS, and the order in which it was taken up. Hence, we should expect to find every feature of the protocol that concerns the task mirrored in an essential way in the program trace. The converse is not true, since many things concerning the task surely occurred without the subject's commenting on them (or even being aware of them). Thus, our test of correspondence is one-sided but exacting.
Let us start with the first sentence of the subject's protocol:

"Well, looking at the left-hand side of the equation, first we want to eliminate one of the sides by using rule 8."

We see here a desire to decrease L1 or eliminate something from it, and the selection of rule 8 as the means to do this. This stands in direct correspondence with goals 1, 2, and 8 of the trace. Let us skip to the third and fourth sentences:

"Now — no, — no, I can't do that because I will be eliminating either the Q or the P in that total expression. I won't do that at first."

We see here a direct expression of the covert application of rule 8, the subsequent comparison of the resulting expression with L0, and the rejection of this course of action because it deletes a letter that is required in the final expression. It would be hard to find a set of words that expressed these ideas more clearly. Conversely, if the mechanism of the program (or something essentially similar to it) were not operating, it would be hard to explain why the subject uttered the remarks that he did.

One discrepancyis quite clear. The subject handled both forms of rule 8 together, at least as far as his comment is concerned. GPS, on the other hand, took a separate cycle of consideration for each form. Possibly the subject followed the program covertly and simply reported the two results together. However, we would feel that the fit was better if GPS had proceeded something as follows:

```
GOAL 2 DELETE R FROM L1
    GOAL 3 APPLY R8 TO L1
        PRODUCES L2 $R \supset -P$ OR $-R \supset Q$
GOAL 4 TRANSFORM L 2 INTO L0
    GOAL 5 ADD Q TO $R \supset -P$ OR ADD P TO $-R \supset Q$
        REJECT
```

We will consider further evidence on this point later.

Let us return to the second sentence, which we skipped over:

"It appears too complicated to work with first."

Nothing in the program is in simple correspondence with this statement, though it is easy to imagine some possible explanations. For example, this could merely be an expression of the matching — of the fact that L1 is such a big expression that the subject cannot absorb all its detail. There is not enough data locally to determine what part of the trace should correspond to this statement, so the sentence must stand as an unexplained element of the subject's behavior.

Now let us consider the next few sentences of the protocol:

"Now I'm looking for a way to get rid of the horseshoe inside the two brackets that appear on the left and right side of the equation. and I don't see it. Yeh, if you apply rule 6 to both sides of the equation, from there I'm going to see if I can apply rule 7."

This is in direct correspondence with goals 9 through 14 of the trace. The comment at the end makes it clear that applying rule 7 is the main concern and that changing connectives is required in order to accomplish this. Further, the protocol shows clearly that rule 6 was selected as the means. All three rule selections provide some confirmation that preliminary test for feasibility was made by the subject — as by GPS — in the reduce goal method. If there was not selection on the main connective, why wasn't rule 5 selected instead of rule 6? Or why wasn't the $(A . B) \vee (A . C) \rightarrow A \cdot (B \vee C)$ form of rule 7 selected? However, there is a discrepancy between trace and protocol, for the subject handles both applications of rule 6 simultaneously (and apparently was also handling the two differences simultaneously); whereas GPS handles them sequentially. This is similar to the discrepancy noted earlier in handling rule 8. Since we now have two examples of parallel processing, it is likely that there is a real difference on this score. Again, we would feel better if GPS proceeded somewhat as follows:

```
GOAL 9 APPLY R7 TO L1
    GOAL 10 CHANGE CONNECTIVE TO v IN LEFT L1
                                    AND RIGHT L1
        GOAL 11 APPLY R6 TO LEFT L1 AND RIGHT L1
            PRODUCES L5 $(-R \vee -P) . (R \vee Q)$
```

A common feature of both these discrepancies is that forming the compound expressions does not complicate the methods in any essential way. Thus, in the case involving rule 8, the two results stem from the same input form, and require only the single match. In the case involving rule 7, a single search was made for a rule and the rule applied to both parts simultaneously, just, as if only a single unit was involved.

There are two aspects in which the protocol provides information that the program is not equipped to explain. First, the subject handled the application of rule 8 covertly commanded the experimenter to make the applications of rule 6 on the board. The version of GPS used here did not make any distinction between internal and external actions. To this extent it fails to be an adequate model. The overt-covert distinction has consequences that run throughout a problem, since expressions on the blackboard have very different memory characteristics from expressions generated only in the head. Second, this version of GPS does not simulate the search process sufficiently well to provide a correspondent to *"And I don't see it. Yeh, ..."*. This requires providing a facsimile of the rule sheet, and distinguishing search on the sheet from searches in the memory. The next few sentences read:

"I can almost apply rule 7, but one R needs a tilde. So I'll have to look for another rule. I'm going to see if I can change that R to a tilde R."

Again the trace and the protocol agree on the difference that is seen. They also agree that this difference was not attended to earlier, even though it was present. Some fine structure of the data also agrees with the trace. The right-hand R is taken as having the difference (R to $-$R) rather than the left-hand one, although either is possible. This preference arises in the program (and presumably in the subject) from the language habit of working from left to right. It is not without consequences, however, since it determines whether the subject goes to work on the left side or the right side of the expression; hence, it can affect the entire course of events for quite a while. Similarly, in the rule 8 episode the subject apparently worked from left to right and from top to bottom in order to arrive at "Q or P" rather than "P or Q". This may seem like concern with excessively detailed features of the protocol, yet those details support the contention that what is going on inside the human system is quite akin to the symbol manipulations going inside GPS. The next portion of the protocol is:

"As a matter of fact, I should have used rule 6 on only the left-hand side of the equation. So use 6, but only on the left-hand side."

Here we have a strong departure from the GPS trace, although, curiously enough, the trace and the protocol end up at the same spot, $(-R \vee -P).(-R \supset Q)$. Both the subject and GPS found rule 6 as the appropriate one to change signs. At this point GPS simply applied the rule to the current expression; whereas the subject went back and corrected the previous application. Nothing exists in the program that corresponds to this. The most direct explanation is that the application of rule 6 in the inverse direction is perceived by the subject as undoing the previous application of rule 6. After following out this line of reasoning, he then takes the simpler (and less foolish-appearing) alternative, which is to correct the original action.

The final segment of the protocol reads:

"Now I'll apply rule 7 as it is expressed. Both — excuse me, excuse me, it can't be done because of the horseshoe. So — now I'm looking — scanning the rules here for a second, and seeing if I can change the R to −R in the second equation, but I don't see any way of doing it (Sigh). I'm just sort of lost for a second."

The trace and the protocol are again in good agreement. This is one of the few self-correcting errors we have encountered. The protocol records the futile search for additional operators to affect the differences of sign and connective, always with negative results. The final comment of mild despair can be interpreted as reflecting the impact of several successive failures.

7. Summary of the Fit of the Trace to the Protocol

Let us take stock of the agreements and disagreements between the trace and the protocol. The program provides a complete explanation of the subject's task behavior with five exceptions of varying degrees of seriousness.

There are two aspects in which GPS is unprepared to simulate the subject's behavior: in distinguishing between the internal and external worlds, and in an adequate representation of the spaces in which the search for rules takes place. Both of these are generalized deficiencies that can be remedied. I twill remain to be seen how well GPS can then explain data about these aspects of behaviour.

The subject handles certain sets of items in parallel by using compound expressions; whereas GPS handles all items one at a time. In the example examined here, no striking differences in problem solving occur as a result, but larger discrepancies could arise under other conditions. It is fairly clear how GPS could be extended to incorporate this feature.

There are two cases in which nothing corresponds in the program to some clear task-oriented behavior in the protocol. One of these, the early comment about "complication", seems to be mostly a case of insufficient information. The program is making numerous comparisons and evaluations which could give rise to comments of the type in question. Thus this error does not seem too serious. The other case, involving the „*should have* . . ." passage, does seem serious. It clearly implies a mechanism (maybe a whole set of them) that is not in GPS. Adding the mechanism required to handle this one passage could significantly increase the total capabilities of the program. For example, there might be no reasonable way to accomplish this except to provide GPS with a little continuous hindsight about its past actions.

An additional general caution must be suggested. The quantity of data is not large considering the size and complexity of the program. This implies that there are many degrees of freedom available to fit the program to the data. More important, we have no good way to assess how many relevant degrees of freedom a program possesses, and thus to know how easy it is to fit alternative programs. All we do know is that numerous minor modifications could certainly be made, but that not one has proposed any major alternative theories that provide anything like a comparably detailed explanation of human problem solving data.

It would help if we knew something of how idiosyncratic the program was. We have discussed it here only in relation to one sample of data for one subject. We know enough about subjects on logic problems to assert that the same mechanisms show up repeatedly, but we cannot discuss these data here in detail. In addition, several recent investigations more generally support the concept of information processing theories of human thinking [12, 13, 14, 15, 16].

8. Conclusion

We have been concerned in this paper with showing that the techniques that have emerged for constructing sophisticated problem-solving programs also provide us with new, strong tools for constructing theories of human thinking. They allow us to merge the rigor and objectivity associated with Behaviorism with the wealth of data and complex behavior associated with the *Gestalt* movement. To this end their key feature is not that they provide a general framework for understanding problem-solving behavior (although they do that too), but that they finally reveal with great clarity that the free behavior of a reasonably intelligent human can be understood as the product of a complex but finite and determinate set of laws. Although we know this only for small fragments of behavior, the depth of the explanation is striking.

References

[1] Newell, A., Shaw, J. C., and Simon, H. A.: Report on a General Problem Solving Program. Proceedings of the International Conference on Information Processing. UNESCO, June 1959.

[2] Newell, A., Shaw, J. C., and Simon, H. A.: A Variety of Intelligent Learning in a General Problem Solver. In Yovits and Cameron (eds.) Self-Organizing Systems. Pergamon 1960.

[3] Newell, A., and Simon, H. A.: The Simulation of Human Thought. (Current Trends in Psychology) University of Pittsburgh Press 1961.

[4] Moore, O. K., and Anderson, S. B.: Modern Logic and Tasks for Experiments on Problem Solving. Journal of Psychology, 38 (1954), pp. 151—160.

[5] Gelernter, H.: Realization of a Geometry Theorem Proving Machine. Proceedings of the International Conference on Information Processing, UNESCO, June 1959.

[6] Kilburn, T., Grimsdale, R. L., and Sumner, F. H.: Experiments in Machine Learning and Thinking. Proceedings of the International Conference on Information Processing. UNESCO, June 1959.

[7] Minsky, M.: Steps Toward Artificial Intelligence. Proceedings of the Institute of Radio Engineers. January 1961.

[8] Newell, A., Shaw, J. C., and Simon, H. A.: Empirical Explorations in the Logic Theory Machine. Proceedings of the 1957 Western Joint Computer Conference. February 1957.

[9] Newell, A., Shaw, J. C., and Simon, H. A.: Chess Playing Programs and the Problem of Complexity. IBM Journal of Research and Development, 2, 4 (1958).

[10] Samuel, A. L.: Some Studies in Machine Learning, Using the Game of Checkers. IBM Journal of Research and Development, 3, 3 (1959) — Nachdruck mit Nachbemerkung des Autors auf Seite 155 dieses Buches.

[11] Tonge, F.: An Assembly Line Balancing Procedure. Management Science, 7, 1 (1960).

[12] Bruner, J. S., Goodnow, J. J., and Austin, C. A.: A Study of Thinking. Wiley 1956.

[13] Feigenbaum, E.: The Simulation of Verbal Learning Behavior, Proceedings of the 1961 Western Joint Computer Conference. May 1961.

[14] Feldman, J.: Simulation of Behavior in the Binary Choice Experiment. Proceedings of the 1961 Western Joint Computer Conference. May 1961.

[15] Hovland, C.I., and Hunt, E.B.: Computer Simulation of Concept Attainment. Behavioral Science, 5, 3 (1960).

[16] Miller, G.A., Galanter, E., and Pribram, K. H.: Plans and the Structure of Behavior. Holt 1960.