7

Information-Processing Theory of Human Problem Solving

Herbert A. Simon

Carnegie-Mellon University

In the preceding chapter a taxonomy of problem types was proposed, and some hypotheses set forth about the general kinds of skills and knowledge needed to solve problems of each type. The present chapter, adopting the same information-processing point of view as the previous one, undertakes three tasks. The first part of the chapter sets forth the general theory of human problem solving that has emerged from research in the past two decades, especially research that has employed the methods of computer simulation and analysis of thinking-aloud protocols. The second part examines recent and ongoing research aimed at giving an account of the role of perceptual processes in problem solving and a description of the processes for generating problem representations, and research aimed at extending the theory to problem solving in domains that are rich in semantic information and less well structured than those that have been examined in the past. The third part of the chapter discusses some of the methodological issues that must be faced in using the methodologies of simulation and protocol analysis and, in general, to test detailed processing models of human cognitive performance.

Since this chapter and the preceding one both discuss their topics in information-processing terms and since both review essentially the same body of evidence, they reach conclusions that are largely complementary. The present chapter does take, however, a somewhat more sanguine view than the preceding one as to how far we have already progressed in building and testing a coherent theory of human problem solving. The theory discussed in this chapter applies both to Greeno's category of problems of transformation and his category of problems of arrangement. The relation of the theory to problems of inducing structure is discussed in Simon and Lea (1974). For the purposes of the present chapter, problems will be classified according to the definiteness of their structure and the amount of semantic information that must be supplied in order to solve them.

I. A GENERAL THEORY

A human being is confronted with a problem when he has accepted a task but does not know how to carry it out. "Accepting a task" implies having some criterion he can apply to determine when the task has been successfully completed. Problem solving is a nearly ubiquitous human activity; it is doubtful whether anyone spends an hour of his life without doing at least a little of it. The domain of problems ranges from highly structured, puzzle-like tasks often presented to subjects in the psychological laboratory to fuzzy, ill-structured tasks of large magnitude encountered in real life. Solving some problems requires only the information contained in the problem statement—a common characteristic of puzzles. Solving other problems may require drawing upon large stores of information in long-term memory or in external reference sources. Problems presented in the laboratory may take only 15 minutes or less to solve. Some problems presented in real life (for example, certain problems of scientific discovery) may occupy a substantial part of the problem solver's waking time for years.

Information-processing theories have made especially good progress in providing explanations of the processes for solving relatively well-structured, puzzle-like problems of the sorts that have been most commonly studied in the psychological laboratory. The theories describe the behavior as an interaction between an *information-processing system*, the problem solver, and a *task environment*, the latter representing the task as described by the experimenter. In approaching the task, the problem solver represents the situation in terms of a *problem space*, which is his way of viewing the task environment. These three components—information-processing system, task environment, and problem space—establish the framework for the problem-solving behavior (Newell & Simon, 1972, Chapter 14). Specifically:

1. A few, and only a few, gross characteristics of the human information-processing system are invariant over task and problem solver. The information-processing system is an adaptive system, capable of molding its behavior, within wide limits, to the requirements of the task and capable of modifying its behavior substantially over time by learning. Therefore, the basic psychological characteristics of the human information-processing system set broad bounds on possible behavior but do not determine the behavior in detail.

2. These invariant characteristics of the information-processing system are sufficient, however, to determine that it will represent the task environment as a problem space and that the problem solving will take place in a problem space.

3. The structure of the task environment determines the possible structures of the problem space.

4. The structure of the problem space determines the possible programs (strategies) that can be used for problem solving.

These four propositions are *laws of qualitative structure* for human problem solving. We are so accustomed to taking Newton's Laws of Motion as a model of what a theory should look like—or Maxwell's equations, or quantum mechanics—that it is worth reminding ourselves that a large number of important scientific theories do not resemble those in form. Instead,they consist of qualitative statements about the fundamental structure of some set of phenomena (Newell & Simon, 1976). An excellent example is the germ theory of disease, which, as announced by Pasteur, amounted to the following. If you encounter a disease, especially one that spreads rapidly, look for a microorganism. Darwinian evolution is another example, as are the tectonic plate theory of continental drift, the atomic theory of matter, and the cell theory. Sometimes, laws of qualitative structure are later expanded into quantitative theories, sometimes they are not. But at any given moment, they constitute a substantial part of our basic scientific knowledge. The predictions they support are, of course, weaker than the predictions that can be made from more highly quantitative theories, when these are available.

Thus, from a knowledge of the task environment, we can make predictions, but only incomplete ones, about the characteristics of the problem space and from a knowledge of the problem space, incomplete predictions about the problem-solving strategy. In addition, problem space and program must be compatible with the known characteristics of the information-processing system.

A. The Information-Processing System

A few basic characteristics of the human information-processing system shape its problem-solving efforts. Apart from its sensory organs, the system operates almost entirely serially, one process at a time, rather than in parallel fashion. This seriality is reflected in the narrowness of its momentary focus of attention. The elementary processes of the information-processing system are executed in tens or hundreds of milliseconds. The inputs and outputs of these processes are held in a small short-term memory with a capacity of only a few (between, say, four and seven) familiar symbols, or *chunks*. The system has access to an essentially unlimited long-term memory, but the time required to store a new chunk in that memory is of the order of seconds or tens of seconds.

Although many of the details of the system are still in doubt, this general picture of the information-processing system has emerged from psychological experiments of the past 30 years (Norman, 1969). Problem solvers exhibit no behavior that requires simultaneous rapid search of disjoint parts of the problem space. Instead, the behavior takes the form of sequential search, making small successive accretions to the store of information about the problem.

Data for estimating some of the system parameters come from simple laboratory tasks. Rote memory experiments provide evidence that 5 to 10 sec is required to store a chunk in long-term memory. Immediate-recall experiments indicate a short-term memory capacity of perhaps four chunks. Experiments requiring searches down lists or simple arithmetic computations indicate that some 200 msec is needed to transfer symbols into and out of short-term memory. (Some of this evidence is reviewed in Newell & Simon, 1972; Simon, 1974, 1976).

Notice that the limits these parameters place on the behavior of the system are very general. Moreover, except for the capacity limit of short-term memory, they are mostly limits on speed of processing rather than on what processing can be done.

To these processing parameters must be added the organizational characteristics of long-term memory. The classical notion that the human memory is an associative net has been modified into the notion that it may be represented as an organization of list structures (alternatively referred to as node-link structures and by mathematicians, as colored directed graphs). How memory can be modeled with such structures is discussed by Newell and Simon (1972) and Anderson and Bower (1973). The distinction between a classical associative memory and a list structure memory is that the former consists of simple undifferentiated associations between pairs of nodes, while the latter consists of specific, and distinguishable, relations between such pairs. Thus, a memory of the former kind can represent a node only as being associated to another, whereas a list structure memory can represent a node as denoting the color of the object denoted by another, its size, its opposite, a subclass, and so on. List structure memories were anticipated in the *directed associations* of the Würzburg psychologists.

B. Structure of the Task Environment

A puzzle-like problem that has been frequently studied is the missionaries and cannibals problem (Greeno, 1974; Reed, Ernst, & Banerji, 1974; Simon & Reed, 1976; Thomas, 1974). Three missionaries and three cannibals stand on one side of a stream, with a boat capable of carrying just three persons. All six persons are to be transported across the stream, with the condition that at no time, on either side of the stream, may missionaries be outnumbered, even momentarily, by cannibals. Human subjects generally find this a fairly difficult problem, although there are very few alternative legal moves, and it would be easy with paper and pencil to map out the entire problem space and solve the problem directly. As long as subjects confine themselves to legal moves (they usually make only a small percentage of illegal ones), their behavior is highly restricted by the structure of the problem space. Moreover, taking account of the goal—to move people from one bank of the river to the other—we might also predict that most

moves would consist in taking a full boatload (two persons) across and a single person back. The difficulty of the problem is connected with the fact that at one point along the solution path *two* persons must return to the starting side. This move appears inconsistent with the perceived task requirements.

The structure of the problem space constrains behavior in a variety of ways. First, it defines the legal moves. Second, it defines the goal and usually, though implicitly, the direction of movement toward or away from the goal. Third, it interacts with the limits on short-term memory to make some solution paths easier to find than others. If an information-processing system follows a sequence of moves down a blind alley, it must back up to a previous position and search from there in a new direction. But to do this requires some memory of previous positions, which is difficult or impossible to retain for searches of any great size. Hence, when available, methods of search that avoid the necessity of backup will be adopted. If the problem can be factored, for example, and each factor dealt with separately, trying out all combinations of the individual factors may be avoided.

In the well-known cryptarithmetic problem, DONALD + GERALD = ROBERT (Bartlett, 1958; Newell & Simon, 1972), ten distinct digits must be substituted for the ten distinct letters in such a way that the resulting expression is a correct arithmetic sum (526485 + 197485 = 723970). As the problem is usually posed, the hint is given that D = 5. Almost all subjects who solve the problem find the values for the individual letters in a particular sequence: T = 0, E = 9, R = 7, A = 4, L = 8, G = 1, N = 6, B = 3, 0 = 2. The reason is that only if this order is followed can each value be found definitely without considering possible combinations with the values of the other letters. With this order, the solver does not have to remember what alternative values he has assigned to other variables, or to back up if he finds that a combination of assignments leads to a contradiction.

This characteristic of the problem can be determined by examining the structure of the problem itself (not all cryptarithmetic problems have this property); and its strong influence on the search behavior of the information-processing system derives directly from the system's small short-term memory capacity. The empirical fact that solvers do make the assignments in roughly this same order provides us with one important piece of evidence (others can be obtained by analyzing thinking-aloud protocols and eye movements) that the human information-processing system operates as a serial system with limited short-term memory.

C. Problem Spaces

To carry on his problem-solving efforts, the problem solver must represent the task environment in memory in some manner. This representation is his problem space. The problem space—the way a particular subject represents a task in order to work on it—must be distinguished from the task environment—the omniscient observer's way of describing the actual problem. Nevertheless, since the information-processing system is an adaptive system, problem space and task environment will not be unrelated. The simplest problem space for a task, usually called the *basic problem space,* consists of the set of nodes generated by all legal moves.

The relative ease of solving a problem will depend on how successful the solver has been in representing critical features of the task environment in his problem space. Although the problem space and the solver's program are not task-invariant, they constitute the adaptive interface between the invariant features of the information-processing system and the shape of the environment and can be understood by considering the functional requirements that such an interface must satisfy.

Each node in a problem space may be thought of as a possible state of knowledge that the problem solver may attain. A state of knowledge is simply what the problem solver knows about the problem at a particular moment of time, knows in the sense that the information is available to him and can be retrieved in a fraction of a second. After the first few moves in the missionaries and cannibals problem, the subject knows only the current locations of missionaries, cannibals, and boats; the starting situation; and the goal situation. He probably remembers little about the exact situations he has reached before, although after he has worked on the problem for a time, he may begin to store such information in long-term memory. The search for a solution is an odyssey through the problem space, from one knowledge state to another, until the current knowledge state includes the problem solution.

Problem spaces, even those associated with relatively simple task environments, may be enormous. Since there are 9! possible assignments of nine digits to nine letters, the basic DONALD + GERALD space contains a third of a million nodes. The sizes of problem spaces for games like chess or checkers are measured by very large powers of ten. As we have seen, however, the space for a fairly difficult problem like missionaries and cannibals may be very small (only 16 legal positions). Water jug problems (Atwood & Polson, 1976; Luchins, 1942) have problem spaces of about this same size.

Another difficult problem that has a relatively small space of legal moves in is/ the Tower of Hanoi puzzle (Egan, 1973; Gagné & Smith, 1962; Hormann, 1965; Klix, 1971; Simon, 1975). In this problem, there are three pegs, on one of which is a pyramid of wooden disks. The disks are to be moved, one by one, from this peg, and all placed, in the end, on one of the other pegs, with the constraint that a disk may never be placed atop a smaller one. If there are four disks, the problem space comprised of possible arrangements of disks on pegs contains only $3^4 = 81$ nodes, yet the problem is nontrivial for human adults. The five-disk problems, though it admits only 243 arrangements, is very difficult for most people; and the problems with more than five disks are almost unsolvable, until the right problem representation is discovered!

Problems like the Tower of Hanoi and missionaries and cannibals, where the basic problem space is not immense, tell us that the human information-processing system is capable of, or willing to endure, very little trial-and-error search. Problems with immense spaces inform us that the amount of search required to find solutions, making use of representations that capture the structure of the task environment, bears little or no relation to the size of the entire space. An information-processing system need not be concerned with the size of a haystack, if a small part can be identified in which there is sure to be a needle. Effective problem solving involves extracting information about the structure of the task environment and using that information for highly selective heuristic searches for solutions.

D. Information Embedded in Problem Spaces

Problem spaces differ not only in size—a difference we have seen to be usually irrelevant to problem difficulty—but also in the kinds of structure they possess. Structure is simply the antithesis of randomness, providing redundancy and information that can be used to predict the properties of parts of the space not yet visited from the properties of those already searched. This predictability becomes the basis for searching selectively rather than randomly.

The simplest example of information that can be used to solve problems without exhaustive search is the progress test, the test that shows that one is "getting warmer." Most of the principles of selection that problem solvers have been observed to use are based on the "getting warmer" idea. In the cryptarithmetic problem, for example, the number of letters for which definite substitutions have been found is a measure of progress. In the Tower if Hanoi task, the number of disks on the goal peg is a measure of progress. In missionaries and cannibals, the number of persons on the far bank of the river is such a measure. Observations of subjects working on these kinds of problems show that they are generally aware of, and make use of, such criteria of progress. How are the criteria used?

Each knowledge state is a node in the problem space. Having reached a particular node, the problem solver can choose an *operator* from among a set of operators available to him and can apply it to reach a new node. Alternatively, the problem solver can abandon the node he has just reached, select another node from among those previously visited, and proceed from that node. Thus, he must make two kinds of choices: choice of a node from which to proceed and choice of an operator to apply at that node.

We have already noted that becauseof limits of short-term memory, problem # solvers do not, in fact, often backtrack from the current node, for this would require them to keep in mind nodes previously visited. Instead, they tend to focus almost exclusively on proceeding from the current situation, whatever that may be. However, when suitable external memory is provided, as when successive moves are written down on paper, problem solvers may be more willing to back up from an unpromising current situation to a more promising one that was reached earlier. Such branching search is frequently observed, for example, with subjects who are seeking to construct proofs for theorems.

We can think of information as consisting of one or more evaluations (not necessarily numerical, of course) that can be assigned to a node or an operator. The most important kind of evaluation for human problem solvers ranks the operators at each node with respect to their promise as a means of continuing from that node. When we examine what information problem solvers' draw on for their evaluations, we discover several varieties. In the simplest of these, an evaluation may depend only on properties of a single node. Thus, in theorem-proving tasks, we find frequent statements in subjects' protocols to the effect that "it looks like Rule 7 would apply here."

In most problem spaces, however, the choice of an efficient next step cannot be made by absolute evaluation of the sort just mentioned. Instead, it is a function of the problem that is being solved. In theorem proving, for example, what to do next depends on what theorem is to be proved. Hence, an important technique for extracting information to be used in evaluators is to compare the current node with characteristics of the desired state of affairs and to extract *differences* from the comparison. These differences serve as criteria for selecting a relevant operator. Reaching a node that differs less from the goal state than nodes visited previously is progress, and selecting an operator that is relevant to reducing a particular difference between current node and goal is a technique for (possibly) approaching closer to that goal.

The particular heuristic search system that finds differences between current and desired situations, then finds an operator relevant to each difference, and applies the operator to reduce the difference is usually called *means–ends analysis*. Its common occurrence in human problem-solving behavior has been observed and discussed frequently since Duncker (1945; see also Atwood & Polson, 1976; Simon & Reed, 1976; Sydow, 1970). The procedure is captured in concrete information-processing terms by the General Problem Solver (GPS) program, which has now been described several times in the psychological literature.[1] The GPS find-and-reduce-difference heuristic played a central role in the theory of problem solving for a decade beginning with its formulation in 1957, but more extensive data from a wider range of tasks have now shown it to be a special case of the more general information-extracting processes being described here.

[1]Brief descriptions of GPS can be found in Hilgard and Bower (1974) and Hilgard, Atkinson, and Atkinson (1975). For an extensive analysis of GPS, see Ernst and Newell (1969). The relation of GPS to human behavior is discussed in Newell and Simon (1972, Chapter 9).

E. Summary

This, in sum, is a first-order approximation to an account of human problem solving in information-processing terms. A serial information-processing system with limited short-term memory uses the information extractable from the structure of a problem space to evaluate the nodes it reaches and the operators that might be applied at those nodes. Most often, the evaluation involves finding differences between characteristics of the current node and those of the desired node (the goal). The evaluations are used to select a node and an operator for the next step of the search. Operators are usually applied to the current node, but if progress is not being made, the solver may return to a prior node that has been retained in memory, the choice of prior node being determined mostly by short-term memory limits.

This theory of the qualitative structure of problem-solving processes has been shown to account for a substantial part of the human behaviors observed in the half dozen task environments that has been studied intensively. In addition to conventional tests based on experiments and observations, the theory has been supported by strong tests of a novel kind. The theory postulates that problem-solving behavior is produced by a small set of elementary information processes, organized into strategies or programs. It asserts that a system capable of performing these processes can solve problems and produce behavior that closely resembles human behavior in the same problem-solving situations. The sufficiency of these elementary information processes for problem solving has been demonstrated by constructing computer programs that simulate human behavior in considerable detail.

have/

II. EXTENSIONS OF THE THEORY

The theory described in the previous section needs to be altered in several respects to fit the data better and is incomplete in other respects. First, varieties of search that use somewhat different forms of means–ends analysis than have been described thus far must be accommodated. Second, a role must be provided for perceptual processes, and especially recognition processes, in problem-solving. Third, no account has been given of how the problem solver generates the problem space from the description of task environment or from other information that he has available. Fourth, little has been said about tasks that are less well structured than the puzzle-like problems that have been studied in the laboratory. The present section will discuss empirical evidence bearing upon these four topics.

A. Information-Gathering Strategies

Consider a student trying to prove the geometry theorem (new to him) that the base angles of an isoceles triangle are equal. This can be accomplished by proving that the two base angles are corresponding angles of congruent triangles. Appropriate congruent triangles can be constructed by dropping a line from the vertex of the isoceles triangle to its base. This line can be drawn (*a*) perpendicular to the base of the triangle, (*b*) cutting the base at its midpoint, or (*c*) bisecting the vertex angle. (It is the same line in all three cases, but this must be proved and is not assumed in the constructions.) In case *a* the two triangles are congruent because they are right triangles with equal hypotenuses and an equal pair of legs. In case *b* they are congruent because they have three sides equal. In case *c* they are congruent because they have two sides and the included angle equal.

Greeno (1976) has observed that students confronted with this problem established the goal of proving two triangles congruent and carried out one of the constructions. But they did not plan in advance which of the theorems on congruent triangles they would use or which parts of the corresponding triangles they would prove equal. They simply made the construction (one of them), determined what parts were consequently equal, then *recognized* that these equalities matched the hypotheses for one of the theorems on congruent triangles.

The process used by the students might be outlined as follows: to reach G, reach G2, G2, or G3, any one of which leads directly to G. Proceed from the givens of the problem to attain P1, P2, P3 . . . until one of these is recognized as leading directly to G1 or G2 or G3. This process can be described as a form of means–ends analysis, but it has some interesting features. First, some of the problem-solving is done at an abstract planning level, rather than in the concrete problem space of geometry theorems. The subject develops the plan of proving two angles equal by proving they are corresponding angles of congruent triangles. The two triangles are to be proved congruent by proving various (but unspecified) sides and angles to be equal. This plan is readily stated in the language of means and ends, in order to prove two angles equal, prove two triangles congruent; in order to prove two triangles congruent, prove that various parts are equal. Each step in the plan establishes subsidiary problem-solving goals.

Second, while the planning step involves working backward from the final theorem, proving that the two triangles are congruent involves working forward from known premises toward a rather ill-defined goal: the goal of applying *some* theorem about congruent triangles. As Greeno (1976) points out, the subject does not deliberately aim at proving a particular congruence theorem, but *recognizes* when he has established enough premises about equal parts so that one of the available theorems is applicable. The recognition process retrieves the theorem "automatically," just when that stage is reached.

The same aspect of taking action on the basis of local criteria, and without specific reference to the precise problem goal, shows up in many other problem-

solving performances. The simplest behavior of this kind represents working forward from the current situation in the general direction of the goal, making use of some kind of criterion of directionality. Simon and Reed (1976), for example, have shown that subjects' behavior in the missionaries and cannibals problem can be modeled quite well by assuming that they choose moves on the criterion of taking as many persons across the river as possible on each trip, and as few back as possible and that they have a modest capability to avoid repeating moves by holding in short-term memory some recollection of the immediately preceding situation. Ericsson (1975) has observed slightly more elaborate behavior, involving the establishment of subgoals, in the 8's puzzle; but once subgoals were established, his subjects used a similar crude test of directionality to aim their moves at the subgoals.

The relation of these simple procedures to the behavior described by Greeno (1976) becomes clearer if we observe that in theorem-proving and equation-solving tasks (including the cryptarithmetic task), any step that assigns a definite value to a previously unknown variable can be regarded as progress toward the goal (Simon, 1972). In the cryptarithmetic problems, for example, a means for progressing toward an assignment for all the letters is to find a correct assignment for any one of them. In many algebra problems involving more than one variable, if the equations are processed in the right order, they can be solved successively for individual variables, and these values substituted in the remaining equations. Such behavior has been observed, for example, in students solving problems in chemical-engineering thermodynamics (Bhaskar & Simon, 1977). In all of these cases, as in Greeno's case, the problem-solving activity can be described as a search for information—replacing unknowns by known values—rather than a search to reach a particular goal. In such activity, recognition processes play a crucial role (*a*) in determining when enough information is available to establish the value of another variable, and (*b*) when enough values have been established to reach the problem goal.

B. Perception in Problem Solving

The importance of perceptual processes in problem solving was demonstrated early by de Groot (1966) in his studies of the choice of moves in chess. He showed that a grandmaster might discover the correct move in a complex position within 5 sec or less of looking at the position for the first time (but might then spend 15 min verifying the correctness of the move). Continuing research on perception in chess (see, for example, Chase & Simon, 1973, and a discussion by Greeno, Chapter 6 of this volume) has built a substantial body of evidence that an important component of the grandmaster's skill is his ability to recognize a great variety of configurations of pieces in chess positions and to associate with the recognized patterns information about possibly appropriate actions. The chess master's vocabulary of familiar patterns has been estimated to be on the order of

50,000, a number comparable to the nature language vocabulary of a college-educated person.

The organization of a problem solver based on perceptual recognition can be described with the help of the concept of *production* (Newell, 1973; Newell & Simon, 1972). A production is a process with two components: a condition component and an action component. The condition component consists of a set of tests to be applied, for example, to a sensory stimulus. If the stimulus is, for example, a picture of a simple colored geometric shape, as in standard concept-attainment experiments, the condition component of a production might apply the test "red and round." The output of the condition is "true" or "false," as the tests are, or are not, satisfied by the stimulus.

If the condition of a production is satisfied, then the action of of the production is executed; if the condition is not satisfied, nothing is done. Thus, the general paradigm for a production is:

If stimulus is X, then do Y; else exit.

An information-processing system can be constructed wholly of productions, the device being a perfectly general one. Such a system takes on psychological interest when we impose on it some conditions about the nature of the conditions and actions that will be admitted. We consider two classes of productions:

In Class P, the perceptual productions, the conditions are tests on sensory stimuli; the actions transfer symbols from long-term memory to short-term memory.

In Class G, the general productions, the conditions are tests on the contents of short-term memory; the actions are motor acts, changes in short-term memory, or changes (storage or retrieval) in long-term memory.

The perceptual productions, which are the ones of main interest for the present discussion, perform acts of recognition: the condition of each such production is satisfied by some class of stimuli. Recognition of a stimulus as belonging to that class accesses the node in long-term memory where associated information is stored and brings a symbol designating that node into short-term memory.

The Elementary Perceiver and Memorizer (EPAM) (Feigenbaum, 1961; Simon & Feigenbaum, 1964; Gregg & Simon, 1967a) is a system that performs recognitions in this way and also learns to discriminate new stimuli, gradually acquiring an appropriate set of productions in the process. Since the EPAM program has had considerable success in explaining a whole range of empirical phenomena from rote-learning experiments, it provides support for postulating this sort of mechanism in the human recognition process.

To return to the case of chess perception, Simon and Gilmartin (1973) constructed a system, MAPP (Memory And Perceptual Processor), that simulates the chessplayer's perceptual capabilities, growing, in EPAM fashion, a set of patterns it is capable of discriminating and recognizing as a result of exposure to those patterns. The conditions of the productions in this case identify a configuration of pieces on a chessboard. When such a configuration is recognized, a symbol designating it is retrieved from long-term memory and placed in short-term memory. Short-term memory capacity is interpreted as the number of such symbols that can be held simultaneously.

Along with the long-term memory symbol designating a pattern, other information can be stored, for example, a move that would be plausible to consider when that pattern is present on the board. Upon recognition of the pattern by execution of a P production, a G production, detecting the symbol in short-term memory, could retrieve the associated chess move and place *it* in short-term memory. Then a second G production could cause the move to be made on the board. This system could, in fact, play entire games of chess, simply recognizing plausible moves in each position and making them. It would not play good chess, for the first potential move recognized in a position might not be the correct one and, in any event, would not usually be accepted by a player without additional analysis. However, the system would probably be a good representation of the processes used in playing rapid-transit chess, when only a few seconds are allowed for a move. It is well known that grandmasters can play strong games (at expert level) but not grandmaster games under these conditions.

Greeno (1976) has shown how the geometry theorem-proving processes can be simulated by a production system very like the one just described. The conjunction of the premises of a theorem constitutes the condition for a production that retrieves the theorem. When a theorem is retrieved, the instance of it that applies to the specific problem being solved is generated and held in memory.

Simon (1975) has constructed for the Tower of Hanoi problem a whole family of alternative production systems that are capable of solving that problem. This collection of systems demonstrates that quite different information-processing strategies may produce functionally equivalent behaviors. One system that solves a problem may be primarily goal driven: it uses goal and subgoal structures to determine what to do next. Another system may be primarily stimulus driven: it uses visual cues from the current state of the problem apparatus to determine what to do next. A third system may be primarily pattern driven: it uses a stored pattern or rule to calculate each successive move in the solution path. A fourth system may solve the problem simply by rote memory of the sequence of correct moves.

Little empirical research has yet been done to determine, in situations like these where numerous alternative solution strategies are available, which strategies human subjects will use or within what limits their strategies can be determined by problem instructions, past experience, or other experimental manipulations. The potential significance of this line of investigation is demonstrated by the experiments of Katona (1940), who taught subjects alternative strategies for solving the same problem. Subjects taught a rote strategy for solving the problem were less successful in retaining the solution and in transfering the strategy to similar problems than subjects who were taught a strategy based upon perception or analysis of the problem structure.

C. Generation of the Problem Representation

When a subject is presented with a novel problem in the laboratory, he cannot begin to try to solve it until he understands it. As psychological experiments are usually conducted, the subject is introduced to the task through instructions and explanations from the experimenter, followed by an opportunity to practice on some examples. Only then does the gathering of data on his behavior ordinarily begin. Under these circumstances, by the time the actual experiment begins, the subject already understands the task and has generated for himself a problem space within which he can represent it. Within this paradigm, there is no opportunity to discover the process he uses to generate his representation of the problem.

Hayes and Simon (1974, 1976a) have used isomorphs of the Tower of Hanoi problem to study how subjects generate problem representations. Two problems are ismorphic if there is a one-to-one mapping of legal moves of the first problem onto legal moves of the second, such that the starting and goal situations of the first are mapped onto the starting and goal situations of the second. Unlike the original Tower of Hanoi problem, the isomorphs employed by Hayes and Simon do not use an external display, but are described in words. Thinking-aloud protocols covering the entire interval from the time when the subject reads the problem instructions to the time when he is ready to begin work on solving the problem reveal the main features of the subject's behavior while he is generating a representation for a new problem.

This process has been simulated by a computer program called UNDERSTAND. As hypothesized by the program, the understanding process contains two subprocesses: one for interpreting the language of the instructions, the other for constructing the problem space. The process for interpreting language reads the sentences of the problem text and extracts information from them, guided by a set of information-extraction rules. These rules identify the moods of the text sentences, identify noun groups that refer to physical objects and activities, and assign such relations to them as "agent," "instrument," "property," "location," and so on, much in the manner of a case grammar (Fillmore, 1968).

The construction process accepts information, sentence-by-sentence, from the language-interpreting process and builds a representation of the problem space in two parts: a situation description and a set of operators. The description of the situation, based on information extracted from sentences in the indicative mood, represents the problem elements (for example, the pegs and disks in the Tower of Hanoi problem), relations among problem elements (for example, the relation of a disk being on a peg), and the initial and goal states of the problem.

The set of operators, identified from information extracted from conditional statements and sentences in the subjunctive mood, constitutes a production system in which the conditions are represented as states (or aspects of states) of the situation, and the actions are represented as processes for making changes in the situation. A major responsibility of the construction process is to make certain that the representation of the situation is compatible with the representation of the operators, so that the operator processes will perform correctly in changing the situation.

Thus, the model views problem solving by the naive subject as employing two complex processes: an understanding process that generates a problem space from the text of the problem and a solving process that explores the problem space to try to solve the problem. The understanding process is also assumed to proceed in two steps: a language interpretation process followed by a construction process. In the human protocols from which this model was induced, these steps do not occur in invariable sequence. Instead, there is frequent alternation between the understanding process and the solving process and, within the understanding process, between the language interpretation process and the construction process. The solving process appears to exercise overall control in the sense that it begins to run as soon as enough information has been generated about the problem space to permit it to do anything. When it runs out of things to do, it calls the understanding process back to generate more specifications for the problem space. The text of the instructions appears to be interpreted only to the extent that is necessary in order for the solving process to arrive at a problem solution.

The protocols show that the problem representation the subject constructs is determined sensitvely by the precise way in which the problem is stated. As the Tower of Hanoi problem is usually described, disks are associated with pegs, and a legal move consists of moving a disk from one peg to another. An isomorphic problem can be constructed in which pegs are associated with disks, and a legal move consists of changing the peg that is associated with a particular disk. Experiments show (Hayes & Simon, 1976b) that if the problem is described in the instructions in one of these ways, it will almost invariably be represented by the subject in memory in that same way. The experiments also show that the difficulty of solving such problems varies greatly, depending on which of the two representations is used. (In particular, the second representation described above makes the problem about twice as difficult, measured by time required for solution, as the first representation.) From these data we conclude that subjects do not ordinarily search for the most efficient representation for a problem—the representation that will make solving it easiest—but adopt the representation that derives in the most direct and straightforward way from the language of the problem instructions.

The research on the Tower of Hanoi isomorphs treats of problems that are novel to the problem solver. In the case of problems of types that he encountered previously, the understanding process may be determined by that previous experience and may be different for different subjects. Paige and Simon (1966), studying performance in solving algebra word problems, found that some subjects interpreted the problem text almost entirely by syntactic translation from natural language to algebraic equations. Other subjects generated a semantic representation of the situation from the problem text, then used that representation to derive the equations.

Bhaskar and Simon (1977), studying the performance of a skilled subject solving thermodynamics problems, found that the subject used standard formats to express the basic problem conditions in equations, instead of constructing the representation for each problem ab initio.

D. Ill-Structured Problems

The great bulk of the research that has been done on problem solving has made use of task environments whose structure is well defined. It is reasonable to ask to what extent the mechanisms that have been discovered to govern problem solving in well-structured domains are also applicable and used in domains that are more loosely structured. Since there is little evidence as yet to answer this question, my comments on this topic are necessarily somewhat speculative.

There is no precise boundary between problems that may be regarded as well structured and those that are ill structured. Rather, there is a continuum from problems like the Tower of Hanoi or cryptarithmetic puzzles to problems like the task of composing a fugue or designing a house. Among the features that distinguish the second group from the first are these:

1. The criterion that determines whether the goal has been attained is both more complex and less definite.
2. The information needed to solve the problem is not entirely contained in the problem instructions, and indeed, the boundaries of the relevant information are themselves very vague.
3. There is no simple "legal move generator" for finding all of the alternative possibilities at each step.

The earlier discussion of geometry theorem-proving shows that even in well-structured domains, although the final goal may be quite definite, less definite intermediate goals may be pursued along the way. The same phenomenon shows up clearly in the domain of chess. There is no ambiguity in determining whether a player has won the game (has checkmated his opponent), but in choosing moves, the consequences of these moves cannot always be pursued to this final result. Instead, moves must be evaluated by means of sometimes vague and complex criteria that take into account pieces won or lost, positional advantages, and so on. Since problem-solving mechanisms have been demonstrated that operate in the face of these complexities and ambiguities, it appears that no processes beyond the ones we have already mentioned have to be postulated to take care of the first of the three aspects of ill-structuredness listed above.

Reitman (1965), in a study of the protocol of a professional composer writing a fugue, addressed himself to all three aspects. From the protocol evidence, it appeared that over any short interval of time, the composer was dealing with perfectly well-defined subproblems of the total problem. From his long-term memory he repeatedly evoked new information and new generators of alternatives that gradually and continually transformed the problem space in which he was working. Again, the mechanisms appear to be quite similar to those that have been identified in other, more tightly structured, task environments. In particular, recognition of features in the melodic or harmonic fragments he had already created evoked ideas that were associated with those features in long-term memory, much as chess patterns evoke information from the long-term memory of skilled chess players.

Simon (1973) has proposed that, in general, the processes used to solve ill-structured problems are the same as those used to solve well-structured problems. In working on ill-structured problems, however, only a small part of the potentially relevant information stored in long-term memory and in external reference sources plays an active role in the solution processes at any given moment in time. As recognition of particular features in the situation evokes new elements from long-term memory, the solver's problem space undergoes gradual and steady alteration. A production system, he argues, containing a rich repertory of recognition processes and associated with a large store of information in long-term memory, would produce precisely the kind of continually changing problem space that has been observed in protocols of subjects solving such problems.

Some beginning have been made in building and testing theories of the organization of long-term memory (Abelson, 1973; Anderson & Bower, 1973; Quillian, 1968; Rumelhart, Lindsay, & Norman, 1972; Schank, 1972). Most of this work has been done outside the context of problem-solving tasks. Recently, however, Bhaskar and Simon (1977) have undertaken an analysis of the structure of long-term memory used by students solving problems in a college-level course in chemical-engineering thermodynamics. Inventories of the information actually available to persons engaged in skilled performance of professional-level tasks and investigations of how this information is evoked during problem solving will clarify whether the mechanisms and processes that have already been identified in well-structured problem solving are sufficient to account for performance in less structured and information-rich domains.

III. METHODOLOGICAL QUESTIONS

In the introduction, it was noted that the development of an information-processing analysis of problem solving has made use of new experimental and observational methodologies. In this section, these new methodologies are discussed, namely, how the temporal density of observations has been increased and how the modern computer has been used to build and test information-processing theories of the observed behavior.

A. Increasing the Density of Observations

In typical psychological experiments, a stimulus is presented to the subject, he makes a response, and the time required for the response and its correctness become the principal data for analysis. But in a problem-solving situation, 15 min or more may intervene between presentation of stimulus (the problem instructions) and the final response (the answer). The interval between stimulus and response is filled by the subject's information-processing activities, which can form a very long sequence, since the elementary information processes may each occupy no more than a few hundred msec of processing time.

The task of problem-solving research is to identify the organization of processes that enables a subject to solve a problem and that determines how long it takes him and the probability that he will make one or more errors along the way. Since taking measurements of behavior only at the start and the finish seems an unpromising way of learning about the intervening processes, must attention has been given to securing additional observations of the subject's behavior during the course of the problem-solving activity. Two techniques that have been used to increase the density of observations of the information-processing stream are recording thinking-aloud protocols of the problem solver's verbalizations during his activity and recording his eye movements. much

In whatever way the subject is induced to externalize some of his behavior during problem solving—whether with the help of the eye-movement camera or with the use of tape recorder and verbalization—the interpretation of the evidence requires at least a rudimentary theory that connects these behaviors with the problem-solving processes. (This is not an unusual requirement for observation. In physics, for example, the physical theory of the instruments of observation must be understood, at least in first approximation, in order to interpret the observations.) The theory and practice of interpreting verbal protocol data will be discussed in a later section of this chapter.

B. Computer Simulation

The revival of problem solving as a topic for research is closely connected with the discovery that the modern digital computer can be used not only to carry out numerical calculations but to do nonnumerical symbolic information processing as well. This discovery opened the way to programming the computer to simulate human behavior in problem-solving and other cognitive tasks.

Formally speaking, a computer program is a set of difference equations that determines, for each possible state of the computer, what process it will execute

next. If $S(t)$ defines the state of a computer at time t, $I(t)$ its input, and P is its program, then we may describe its behavior by $S(t + 1) = P[S(t), I(t)]$. Difference equations have the same logical structure as differential equations, with the exception that the former treat time as discontinuous, the latter as continuous. Many of the most important theories of physics take the form of systems of differential equations (for example, Newtonian mechanics, Maxwell's equations, wave mechanics). This fact has suggested the idea that computer programs, viewed as difference equations, might provide a powerful language for expressing theories of information-processing systems.

The fundamental hypothesis that motivates the information-processing approach to the study of cognition may be stated thus: The human cognitive system is to be viewed as an information-processing system. The system consists of a set of memories, receptors, and effectors, and processes for acting on them. The memories contain data (information) and programs of information processes. The state of the system at any given moment of time is determined by the data and programs contained in these memories, together with the stimuli that are presented to the receptors.

When a system of differential or difference equations is sufficiently simple—as are some of the equations systems of physics—the equations can be integrated, and invariant properties of the system can be derived that hold generally, not just for special cases. When the system is more complex, the only method that may be available for predicting its behavior is to simulate that behavior for particular circumstances. In general, this is the course that has to be followed when information-processing theories are expressed as computer programs. To explain the behavior of a particular subject, for example, in a specific problem-solving situation, the program is presented with the identical problem.

In those cases where the problems-solving process draws upon semantic information stored in semantic memory, the program that is to simulate the behavior must be provided with that information, or an approximation of it. So, a program that is to simulate the pattern-recognition capabilities of a skilled chess player must be provided with an appropriate vocabulary of familiar patterns, stored in memory in such a way that they will be recognized when presented (Simon & Gilmartin, 1973).

It is sometimes argued that because of the particularistic nature of the programs used for simulation, they cannot be regarded as theories of the cognitive phenomena. But this objection represents a misconception. Simulation programs will vary from one problem-solving environment to another and from one task to another, because different semantic knowledge and processes are used for different problems and by different problem solvers. There can be no invariants in the theory that are not invariants in the behavior to be explained; and the behavior is not invariant, in all particulars, over tasks and over subjects. As was explained earlier, this is a principal reason why the laws of information-processing psychology take the form of laws of qualitative structure. We must seek these qualitative regularities, not in the specific information that a specific subject uses in solving a specific problem, but in the structural invariants that are shared by the programs describing the behavior of the same subject over a range of problem-solving tasks or the behavior of different subjects in a particular task environment.

In spite of this variability in human behavior, it remains the case that computer-programming languages provide a means for stating theories of human information processing with a precision that is not available from ordinary language. The requirement that the programs should, in fact, solve the same problems as the human subjects removes any doubts that a genuine mechanism is being postulated that is adequate to account for the observed behavior and that the real bases for the behavior are not being cloaked in vague nonoperational language. If it is objected that the programs say too much—go beyond the invariants in the behavior—then the theory can be identified with "representative programs," which capture the general mechanisms that have been found, without simulating any particular single person. The "representative program" would play the same role in information-processing psychology as the "representative firm" has played in economic theory, and for the same reason. The programs that have been described in this chapter are mainly representative programs in this sense.

C. Interpretation of Protocol Data

The final methodological question we shall consider is the status of thinking-aloud protocols as a source of data for testing information-processing theories. Three separate issues are involved: (*a*) whether thinking-aloud instructions change the thought process, (*b*) how the information contained in the protocols is related to the underlying thought processes, and (*c*) how thinking-aloud protocols can be compared with the traces of behavior produced by computer-simulation programs.

The first of these questions presents the least difficulty. Even if there are differences between the behavior of a subject solving a problem silently and the same subject solving the equivalent problem while thinking aloud, both performances are examples of human problem solving, and an adequate theory of the one should have as much interest for psychology as a theory of the other. Moreover, the small amount of empirical evidence we possess comparing the two performances suggests that the differences between them are not generally large. The evidence has been reviewed recently by Ericsson (1975). He found that under some circumstances the problem-solving behavior while thinking aloud is somewhat more deliberate and planful, and sometimes a little slower, than the behavior when the subject is not vocalizing. Further, there seems to be some qualitative change in the behavior on tasks where vocalizing leads to recoding of visual stimuli that are not easily describable in words. In tasks where the latter difficulty is not present, however, the weight of evidence indicates a close

similarity of the problem-solving processes in the two conditions. There are no reasons to believe that thinking-aloud instructions cause gross changes in the problem-solving behavior.

There is very little explicit evidence on the relation of the information contained in the thinking-aloud protocols to the underlying thought processes. Most experimenters who have analyzed protocols have assumed that the vocalizations corresponded to a subset of the symbol structures that were temporarily present in short-term memory during the course of the problem-solving process (see Newell & Simon, 1972). That is, only items that pass through short-term memory, but not all of these, will be vocalized. Again, Ericsson (1975) has explored this general hypothesis in some detail. The task he studied (the 8's puzzle) had a substantial perceptual component and, hence, may have been less favorable to relatively complete vocalization than some others. Ericsson found that subjects tended not to vocalize goals that could be realized immediately as often as longer range goals that were reachable through intermediate subgoals. Vocalization decreased as subjects became more proficient in the task, and their responses were "automatic." Ericsson found positive evidence that subjects' goal statements were predictive of their subsequent moves.

In summary, problem-solving behavior during vocalization is genuine problem-solving behavior, hence deserving of study. Moreover, under most circumstances, vocalizing does not greatly alter the behavior. The protocol represents only incompletely the stream of symbols that pass through short-term memory, but there are good reasons to believe that the vocalizations are not an epiphenomenon, but follow closely the actual path of the thinking.

If protocols are accepted as a valid, if highly incomplete, record of the path followed by thought, the problem remains of using them as clues for the underlying processes. The first step in comparing the course of thought with the trace produced by a simulation program is to encode the protocol without destroying its semantic content. A set of process categories is selected to represent the processes that the subject is postulated to be using. Items in the protocol are assigned, clause-by-clause, to these categories. In many cases, a clause can be assigned reliably on the basis of its explicit content. Sometimes, especially when it is elliptical or contains anaphoric references, it has to be interpreted in context. The information retained includes not only the process class, but also its particular instantiation. That is, a statement like, "I'm going to move two missionaries across now," might be encoded, "Move(2M,across)," designating the statement as denoting a move, but also specifying what move it is.

It is not difficult to achieve an acceptable level of reliability in clause-by-clause coding of protocols, and some progress has been made toward automating the process (Waterman & Newell, 1973). The greatest difficulty with encoding as a procedure for treating protocols is its irksomeness as a task for the coder. Hence, automation, or even semiautomation by means of a prompting procedure (Bhaskar & Simon, 1977) is highly desirable.

The next task is to compare protocol with computer trace in order to test the veridicality of the computer-simulation program as a theory of problem solving. As is well known (Gregg & Simon, 1967b), standard statistical tests of hypotheses provide no help in comparing data with models. In particular, the common practice of taking the model as the null hyopthesis for such tests is completely unjustified. On the one hand, this practice leads to the verdict "not rejected" whenever the samples are sufficiently small and the data sufficiently noisy. Hence, the theory embodied in the model is more likely to be accepted with bad data than with good, certainly an undesirable result. On the other hand, this practice leads, when the data are plentiful and good, to rejecting models that explain a large part of the data but are only approximately correct (and it is unreasonable to expect theories to be more than that). Hence, statisticians are unanimous in agreeing that statistical tests are inapplicable to these situations.

The alternative approach is to try to provide some measure of the fraction of the variance in the original data that is accounted for by the model. To this end, the encoded protocol statements may be compared, one by one, with the trace statements. Two kinds of discrepancies between protocol and trace need to be distinguished: errors of omission and errors of commission. For the reasons stated earlier, we must expect that many elements of the program trace, which will always be compulsively complete, will not have corresponding elements in the protocol. Of more consequence are errors of commission, where there is a positive difference between what is predicted by the trace and what actually is found in the protocol.

The extent to which a given degree of fit between protocol and trace should be regarded as supporting the theory will depend also on how parsimoniously the theory is stated. Unfortunately, there is no standard, accepted way to count the number of degrees of freedom in a computer program. Because of the amount of process detail that they make explicit, computer programs appear to have an immense number of degrees of freedom, and it is sometimes thought by the inexperienced that they can be made to fit any behavior path simply by fiddling with them. However, the apparent malleability of programs is largely an illusion. Changing a program to improve the fit to the data in one portion of a protocol will often cause a change in its behavior in other portions, worsening the fit there. But until additional progress has been made toward formally characterizing the parsimony of programs, the evaluation of the goodness of fit between protocol and trace will be a judgmental matter (a not uncommon state of affairs in all of the sciences).

D. Conclusion

A considerable body of tested theory now exists that describes and explains the processes of human problem solving in well-structured task domains having minimal semantic content ("puzzle-like" problems). At the present time, considerable work is going forward that undertakes to extend the theory to broader task domains and, in particular, to explore information-gathering strategies, the interaction between perceptual and cognitive processes in problem solving, and the generation of problem representations. Progress in these directions is bringing various classes of problems, hitherto regarded as ill-structured, within the scope of the theory.

Information-processing approaches to cognition have introduced new methodologies and raised new methodological questions. Computer simulation has been introduced as a major tool for formulating information-processing theories and for testing theories by comparing simulated outputs with longitudinal human data. This has raised novel questions, for which fully satisfactory answers have not yet been found, of how the fit of theory to data should be judged.

Studying human information processes effectively calls for a high temporal density of observations. Several techniques have proved themselves valuable for increasing this density, in particular, recording eye movements, and tape-recording verbal thinking-aloud protocols. Research is beginning to be undertaken on the methodological problems associated with the use of these kinds of empirical data.

REFERENCES

Abelson, R. P. The structure of belief systems. In R. C. Schank & K. Colby (Eds.), *Computer models of thought and language*. San Francisco: Freeman, 1973. Pp. 287–339.

Anderson, J. R., & Bower, G. H. *Human associative memory*. Washington, D.C.: Winston, 1973.

Atwood, M. E., & Polson, P. G. A process model for water jug problems. *Cognitive Psychology*, 1976, **8**, 191–216.

Bartlett, F. C. *Thinking*. New York: Basic Books, 1958.

Bhaskar, R., & Simon, H. A. Problem solving in semantically rich domains: An example from engineering thermodynamics. *Cognitive Science*, 1977, **1**, 193–215.

Chase, W. G., & Simon, H. A. Perception in chess. *Cognitive Psychology*, 1973, **4**, 55–81.

de Groot, A. D. *Thought and choice in chess*. The Hague: Mouton, 1965.

de Groot, A. D. Perception and memory vs. thought: Some old ideas and recent findings. In B. Kleinmuntz (Ed.), *Problem solving: Research, method and theory*. New York: John Wiley & Sons, 1966.

Duncker, K. On problem solving. *Psychological Monographs*, 1945, 58, **5**(Whole No. 270).

Egan, D. E. The structure of experience acquired while learning to solve a class of problems. Unpublished doctoral dissertation, University of Michigan, 1973.

Ericsson, K. A. Instruction to verbalize as a means to study problem solving processes with the 8-puzzle: A preliminary study (No. 458). Stockholm: The University of Stockholm, 1975.

Ernst, G. W., & Newell, A. *GPS: A case study in generality and problem solving*. New York: Academic Press, 1969.

Feigenbaum, E. The simulation of verbal learning behavior. *Proceedings of the Western Joint Computer Conference*, 1961, **19**, 121–132.

Fillmore, C. J. The case for case. In E. Bach & R. T. Harms (Eds.), *Universals in linguistic theory*. New York: Holt, Rinehart, & Winston, 1968. Pp. 1–90.

Gagné, R. M., & Smith, E. C., Jr. A study of the effects of verbalization on problem solving. *Journal of Experimental Psychology*, 1962, **63**, 12–18.

Greeno, J. G. Hobbits and orcs: acquisition of a sequential concept. *Cognitive Psychology*, 1974, **6**, 270–292.

Greeno, J. G. Cognitive objectives of instruction: Theory of knowledge for solving problems and answering questions. In D. Klahr (Ed.), *Cognition and instruction*. Hillsdale, New Jersey: Lawrence Erlbaum Associates, 1976. Pp. 123–159.

Gregg, L. W., & Simon, H. A. An information-processing explanation of one-trial and incremental learning. *Journal of Verbal Learning and Verbal Behavior*, 1967, **6**, 780–787. (a)

Gregg, L. W., & Simon, H. A. Process models and stochastic theories of simple concept formation. *Journal of Mathematical Psychology*, 1967, **4**, 246–276. (b)

Hayes, J. R., & Simon, H. A. Understanding written problem instructions. In L. W. Gregg (Ed.), *Knowledge and cognition*. Potomac, Maryland: Lawrence Erlbaum Associates, 1974. Pp. 167–199.

Hayes, J. R., & Simon, H. A. Understanding complex task instructions. In D. Klahr (Ed.), *Cognition and instruction*. Hillsdale, New Jersey: Lawrence Erlbaum Associates, 1976. Pp. 269–285. (a)

Hayes, J. R., & Simon, H. A. The understanding process: Problem isomorphs. *Cognitive Psychology*, 1976, **8**, 165–190. (b)

Hilgard, E. R., & Bower, G. H. *Theories of learning* (4th ed.). New York: Appleton-Century-Crofts, 1974.

Hilgard, E. R., Atkinson, R. C., & Atkinson, R. L. *Introduction to psychology* (6th ed.). New York: Harcourt, Brace, & World, 1975.

Hormann, A. M. Gaku: An artificial student. *Behavioral Science*, 1965, **10**, 88–107.

Katona, G. *Organizing and memorizing*. New York: Columbia University Press, 1940.

und/ Klix, F. *Information and verhalten*. Berlin: VEB Deutscher Verlag der Wissenschaften, 1971.

Luchins, A. S. Mechanization in problem solving. *Psychological Monographs*, 1942, **54**(6, Whole No. 248).

Newell, A. Production systems: Models of control structures. In W. G. Chase (Ed.), *Visual information processing*. New York: Academic Press, 1973. Pp. 463–526.

Newell, A., & Simon, H. A. *Human problem solving*. Englewood Cliffs, New Jersey: Prentice-Hall, 1972.

of/ Newell, A., & Simon, H. A. Computer science as empirical inquiry: Symbols and search. *Communications in the Association for Computing Machinery*, 1976, **19**, 113–126.

Norman, D. A. *Memory and attention: An introduction to human information processing*. New York: John Wiley & Sons, 1969.

Paige, J. M., & Simon, H. A. Cognitive processes in solving algebra word problems. In B. Kleinmuntz (Ed.), *Problem solving*. New York: John Wiley & Sons, 1966. Pp. 51–119.

Quillian, M. R. Semantic memory. In M. Minsky (Ed.), *Semantic information processing*. Cambridge, Mass.: M.I.T. Press, 1968. Pp. 216–270.

Reed, S. K., Ernst, G. W., & Banerji, R. The role of analogy in transfer between similar problem states. *Cognitive Psychology*, 1974, **6**, 436–450.

Reitman, W. R. *Cognition and Thought*. New York: John Wiley & Sons, 1965.

Rumelhart, D. E., Lindsay, P. H., & Norman, D. A. A process model for long-term memory. In E. Tulving & W. Donaldson (Eds.), *Organization of memory*. New York: Academic Press, 1972. Pp. 198–246.

Schank, R. C. Conceptual dependency: A theory of natural language understanding. *Cognitive Psychology*, 1972, **3**, 552–631.

Simon, H. A. The theory of problem solving. *Proceedings of the IFIP Congress*. Amsterdam: North-Holland Publishing Co., 1972.

Simon, H. A. The structure of ill-structured problems. *Artificial Intelligence*, 1973, **4**, 181–202.

Simon, H. A. How big is a chunk? *Science*, 1974, **183**, 482–488.

Simon, H. A. The functional equivalence of problem solving skills. *Cognitive Psychology*, 1975, **7**, 268–288.

Simon, H. A. The information-storage system called "human memory." In M. R. Rosenzweig & E. L. Bennett (Eds.), *Neural mechanisms of learning and memory*. Cambridge, Mass.: M.I.T. Press, 1976. Pp. 79–96.

Simon, H. A., & Feigenbaum, E. A. An information-processing theory of some effects of similarity, familiarization, and meaningfulness in verbal learning. *Journal of Verbal Learning and Verbal Behavior*, 1964, **3**, 385–396.

Simon, H. A., & Gilmartin, K. A simulation of memory for chess positions. *Cognitive Psychology*, 1973, **5**, 29–46.

Simon, H. A., & Lea, G. Problem solving and rule induction: A unified view. In L. W. Gregg (Ed.), *Knowledge and cognition*. Potomac, Maryland: Lawrence Erlbaum Associates, 1974. Pp. 105–127.

Simon, H. A., & Reed, S. K. Modeling strategy shifts in a problem-solving task. *Cognitive Psychology*, 1976, **8**, 86–97.

Sydow, H. Zur metrischen Erfassung von subjectiven Problemzuständen und zu deren Veränderung im Denkprozess, I and II. *Zeitschrift fur Psychologie*, 1970, **177**:145–198, **178**:1–50.

Thomas, J. C., Jr. An analysis of behavior in the hobbits–orcs problem. *Cognitive Psychology*, 197~~5~~, 4 **6**, 257–269.

Waterman, D. A., & Newell, A. PAS-II: An interactive task-free version of an automatic protocol analysis system. *Proceedings of the Third International Joint Conference on Artificial Intelligence*, 1973. Pp. 431–445.