Reprint Series
20 June 1980, Volume 208, pp. 1335-1342

SCIENCE

#503

Expert and Novice Performance in Solving Physics Problems

Jill Larkin, John McDermott,
Dorothea P. Simon, Herbert A. Simon

Experts solve complex problems considerably faster and more accurately than novices do. Those differences are commonplaces of everyday experience, yet only recently have we begun to understand what the expert does differently from the novice to account for this superiority.

The magic of words is such that, when we are unable to explain a phenomenon, we sometimes find a name for it—as Molière's physician "explained" the effects of opium by its dormitive property. So, we "explain" superior problem-solving skill by calling it "talent," "intuition," "judgment," and "imagination." Behind such words, however, there usually lies a reality we must discover if we are to understand expert performance.

One label often applied to persons skillful in solving physics and engineering problems is "physical intuition." A person with good physical intuition can often solve difficult problems rapidly and without much conscious deliberation about a plan of attack. It just "occurs to him (or her)" that applying the principle of conservation of momentum will cause the answer to fall out, or that a term in kinetic energy can be ignored because it will be small in comparison with other terms in an equation. But admitting the reality of physical intuition is simply the prelude to demanding an explanation for it. How does it operate, and how can it be acquired?

In this article, we undertake to describe what is known about human ex-

The first, third, and fourth authors are members of the Psychology Department, and the second and fourth authors of the Computer Science Department, Carnegie-Mellon University.

0036-8075/80/0620-1335$02.00/0

pert performance in domains such as physics. Since some of the evidence on which our characterization is based comes from research on other tasks—notably chess playing and solving word problems in algebra—we will have something to say about these too. Expertness probably has much the same foundations wherever encountered. As in genetics, we learn much about all organisms by studying a few intensively. Chess, algebra, and physics are serving as the *Drosophila*, *Neurospora*, and *Escherichia coli* of research on human cognitive skills.

Summary. Although a sizable body of knowledge is prerequisite to expert skill, that knowledge must be indexed by large numbers of patterns that, on recognition, guide the expert in a fraction of a second to relevant parts of the knowledge store. The knowledge forms complex schemata that can guide a problem's interpretation and solution and that constitute a large part of what we call physical intuition.

During the past decade, substantial progress has been made in exploring and explaining the human information processes that underlie expert performance. A central problem in this research is to obtain a temporal density in the records of the problem-solving behavior commensurate with the speed of the underlying cognitive process. A major data-gathering technique has been to record verbal accounts by experts and novices as they think aloud during the solution of problems, and to analyze them for similarities and differences. In some studies, videotapes or records of eye movements have been substituted for these thinking-aloud protocols.

At best, however, we obtain in these ways observations every second or half second, whereas the critical human information processes that we must understand appear to be only a few tens or hundreds of milliseconds long. Considerable induction is therefore required to extract and test explanations of the process with data of these kinds. It is nonetheless possible to develop and test theories, in the form of programs for computers, that simulate important aspects of human performance and provide insight into information processing during problem solving.

Our account is based primarily on data from thinking-aloud protocols and computer simulations developed from them. Although we will not emphasize the technical details of the computer programs, their existence and their ability to simulate human behavior demonstrate that our explanations are operational and do not depend on vague, mentalistic concepts. We have shown that a system possessing only the mechanisms incorporated in the programs can account for the main features of the (expert or novice) problem-solving behavior we have observed. Thus, although the theories may commend themselves to common sense as plausible, their primary claim to acceptance rests on their solid foundation in data and computer simulation.

We shall consider a number of components of the expert's skill—perceptual knowledge, recognition capabilities, and the way in which information is represented in long-term memory. Then we will compare and contrast the knowledge and skills of experts and novices in elementary college physics. We will conclude with a brief discussion of processes for acquiring expert skills.

Perceptual Knowledge and the Expert

The most obvious difference between expert and novice is that the expert knows a great many things the novice does not know and can rapidly evoke the particular items relevant to the problem at hand. Two important tasks are to assess the quantity of the expert's knowledge and to determine the form in which it is held in long-term memory. To draw an analogy between the expert's knowledge and the contents of an encyclopedia or other reference book, we must be concerned not only with the book's contents but also with the access routes to those contents, that is, its index. These topics have been studied most intensively in chess playing.

Much chess research, which began as a study of the apparently extraordinary visual imagery of strong chess players, has become research on the chess expert's knowledge and the way in which this knowledge enables a rapid and accurate response. This capacity is usually called "intuition," just as the physicist's rapid response to questions in physics is called physical intuition.

The phenomena of expert chess perception and intuition are illustrated by a simple experiment (*1*). The subject is shown a position from an actual chess game with about 25 pieces on the board for 5 to 10 seconds, and is then required to reproduce the position from memory. A master or grand master can perform this task with about 90 percent accuracy; a weaker player will do well to replace five or six pieces correctly on the board. Next, the experiment is repeated with 25 pieces placed at random on the board instead of in an arrangement from a game. The expert's performance now falls to the level of the novice. The experiment demonstrates that these perceptual skills stem from no innate general superiority of memory, or capacity to visualize, for the superiority is limited strictly to the expert's area of competence—only typical situations are retained.

The principal explanation for these memory phenomena is the "chunking" of familiar stimuli (*2*). (A chunk is any stimulus that has become familiar from previous repeated exposure and hence is recognizable as a single unit.) Brief exposure of a stimulus allows no time to fixate it in long-term memory; it must be retained in short-term memory. But short-term memory has a capacity of only about four to six items, or chunks. For a chess novice, each piece or perhaps pair of pieces on a chessboard is a distinct chunk; hence the locations of only about half a dozen pieces can be held in short-term memory during the reconstruction of the board. For a master, familiar configurations of two to five or six pieces are recognized as distinct chunks, and at least four of these chunks can be retained in short-term memory during reconstruction. A random board has few familiar configurations, hence the master is reduced to trying to remember it piece by piece, like the novice.

If this is the mechanism that permits the master to perform the memory task, the statistics of his performance give us a measure of the amount of perceptual knowledge, measured in chunks or familiar patterns, held in long-term memory. Estimates arrived at by several routes indicate that a grand master or master can recognize perhaps 50,000 such patterns—roughly the number of words and idioms in the vocabulary of a college-educated person (*3*).

This large set of perceptual patterns serves as an index, or access route, not only to the expert's factual knowledge but also to his or her information about actions and strategies. Thus, recognition of a pattern often evokes from memory stored information about actions and strategies that may be appropriate in contexts in which the pattern is present. A chess master recognizing that one of the files on the board is open—free of pieces—realizes immediately that one of his rooks might be moved to the foot of the file. A feature of the board, noted

consciously or unconsciously, produces in a fraction of a second the intuition that a certain action may be appropriate.

The indexed memory, according to this hypothesis, is organized as a large set of productions, each production consisting of a condition and an action (*4*, *5*). Whenever the stimulus to which a person is attending satisfies the conditions of one of his productions (that is, contains a recognizable pattern), the action is immediately evoked (and possibly executed). The actions are stored in memory, and the conditions are the index by means of which memory is accessed.

The condition-action pairs forming productions are the more sophisticated counterparts of the stimulus-response pairs of classical behaviorist psychology. Production systems have been constructed that simulate the chess recognition phenomena just described, and the models of physics problem solving considered in this paper are also production systems.

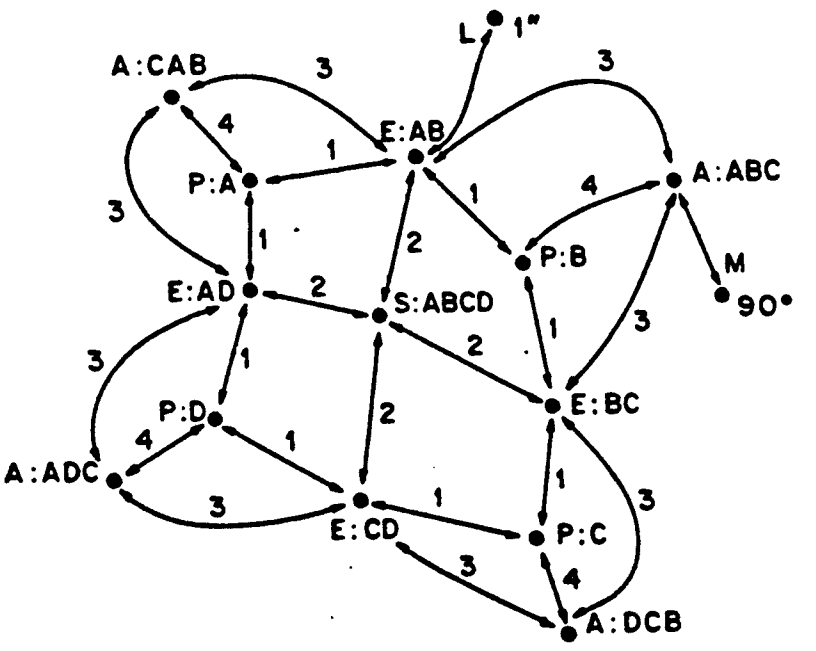

Fig. 1. Node-link representation of the square, *ABCD*. Nodes represent corners (*P*), edges (*E*), angles (*A*), and the surface (*S*). Links connect corners with edges (1), edges with the surface (2), angles with edges (3), and angles with corners (4). Descriptors can be linked to nodes, as shown for the length (*L*) of edge *AB*, and the magnitude (*M*) of angle *ABC*.

Representation

Chess research suggests a general recognition mechanism to explain the intuitions of experts in all fields. However, it leaves open many other questions about expert skills, including how information (in addition to the condition-action units) is represented in long-term memory. The "action" triggered by a production may simply be the recovery from memory of knowledge that includes internal representations of information from the outside world. That is, the action may be "Recall *X*," where *X* is a memory structure representing external information. How can we characterize these memory structures?

Since it is not easy to describe memory representations literally, we often resort to metaphors that may be misleading. For example, virtually everyone can form what we call a "mental picture" of a square. But this does not mean that somewhere in the brain there is a two-dimensional region describable in terms of points, edges, and surfaces, isomorphic to the square. It is hard to imagine a mechanism for manipulating (for example, rotating) or making inferences about such a physical structure in the brain.

More likely is the proposal, supported by gradually accumulating evidence, that human memory consists of a complex organization of nodes connected by links, called a "list structure" (*4*).

In a list structure, objects and components of objects correspond to nodes, and relations between objects correspond to links. As an example, Fig. 1 shows how a square might be represented in memory by (i) a set of nodes denoting points, edges, and a surface and (ii) links relating edges to their endpoints, the surface to its boundary edges, and so on. Descriptive information such as the lengths of edges and the magnitudes of angles can also be incorporated in the node-link structure. Although it has not been shown conclusively that human long-term memory can be represented formally by such node-link structures, a great deal of evidence (*6*) points in this direction and almost all computer simulations of cognition use list structures together with productions that can act on these list structures as their fundamental means for representing memory. These formalisms capture the associative properties of long-term memory.

Before we show how this kind of structure can be used to represent physics problems, it is necessary to discuss the way the natural language statement of a problem can be transformed into such an internal representation. A critical component in the skill of solving physics problems is the ability to translate verbal statements into the language of mathematics, that is, into equations. This skill is first acquired with algebra problems such as

> A board was sawed into two pieces. One piece was one-third as long as the whole board. It was exceeded in length by the second piece by 4 feet. How long was the board before it was cut?

It is easy to see how the translation might proceed. A variable name (x) is assigned to "length of the board." The first piece mentioned then becomes $x/3$ and the second piece ($x/3 + 4$), whereupon the problem states that $x/3 + (x/3 + 4) = x$. In 1968, STUDENT, an early computer program capable of carrying out such translations, was written (*7*). The program was mainly syntactic; it analyzed the grammatical structure of the verbal problem, supplied arbitrary names for quantities mentioned in noun phrases, and translated certain verb forms into algebraic operators and relations such as + and =. STUDENT's semantic knowledge (knowledge of the meaning of the words) was extremely limited, extending mainly to the vocabulary of algebra. To deal with the problem of the board, it did not have to know what a board was, nor sawing.

Human students sometimes, but by no means always, behave as STUDENT does (*8*). One way in which evidence has been gathered is with the help of problems like the example given above but with a slight modification. Suppose we change the second sentence of the problem to "One piece was two-thirds as long as the whole board." STUDENT will have no trouble with the new problem, translating it, by the same processes as were used previously, into $2x/3 + (2x/3 + 4) = x$.

Some human students do exactly what STUDENT does, but others write instead: $2x/3 + (2x/3 - 4) = x$. Careless readers, we may say. Clearly the problem states that the second piece exceeds the first by 4 feet. But a third group of human students, when presented with the problem ask, "Isn't there a contradiction?" Of course they are using the term "contradiction" loosely. There is nothing contradictory about either of the equations written above. But if we solve the first equation, we find that the board is -12 feet long, whereas if we solve the second, which is an incorrect rendering of the problem statement, we find that it is $+12$ feet long. The contradiction, then, for the students who noticed it, was a conflict between the literal interpretation of the problem statement and their knowledge that boards (in the physical world) do not have negative lengths.

Thus some students processed the problem syntactically, writing down equations inconsistent with any semantic knowledge of how sawing boards works. Others used this semantic knowledge, either to write reasonable (although syntactically incorrect) equations, or to note a contradiction between the problem syntax and their own semantic knowledge.

If the semantics of physical objects is important to understanding algebra word problems, it is even more crucial to understanding problems in mechanics. Figure 2 (generated on a cathode-ray tube by a computer program) illustrates a typical statics problem in a college physics text. The components out of which this problem is constructed are, in their most concrete form, objects (ladders, floors) having specified properties (weight, rigidity) and relations (the ladder stands on the floor and its foot presses against the wall). Thus, at the level of abstraction appropriate to the algebraic translation of the problem, the ladder is simply a lever, and the point of contact between ladder and wall is a point of equilibrium for certain forces.

Novak has constructed a simulation program, ISAAC (*9*, *10*), that interprets statics problems written in English. It generates representations of the problem in computer memory, derives appropriate equations from the representations, and solves the problems. ISAAC provides both a model of the processes human subjects use to solve the problems and a theory of how physical representations enter into these processes.

ISAAC can analyze natural language in a way analogous to, but more sophisticated than, the English language processing components of STUDENT. But the critical component of ISAAC is a set of schemata, stored in its long-term memory, that describes archetypal levers, fulcrums, ropes, frictionless surfaces, and the like. These schemata constitute ISAAC's semantic knowledge of the workings of idealized physical objects. Each schema is a list structure containing lists of descriptors that characterize it. Thus, the schema for "lever" refers to such properties as its length and such components as its fulcrum.

When ISAAC recognizes an object mentioned in the problem text as an instance of one of its schemata (it can recognize, for instance, that a ladder is a lever), it constructs a copy of the schema, associating with it the specific properties of the object in the problem, such as the length of the ladder and the angle it makes with the wall. It then generates nodes to link the several objects at their points of contact and thus gradually builds up in memory a representation of the problem situation—a node-link structure that is more or less isomorphic with a diagram one might draw of the situation. In fact, ISAAC's problem representation contains sufficient information that a simple computer program can display on the screen of a cathode-ray tube a picture of the problem (Fig. 2). The pictorial part of that figure was in fact generated by ISAAC from the problem text displayed below it.

Thus ISAAC does not translate syntactically from the natural language problem text directly into algebraic equations, but uses its semantic knowledge to construct a physical representation—a node-link abstract diagram of the problem—to guide its generation of equations. The creation of the physical representation identifies the points at which forces must be equilibrated and provides a scheme of connections so that the forces themselves can be traced and identified. It is hard to see how these inferences could be made without the help of the representation or something equivalent to it.

ISAAC gives us a very specific notion, both concrete and formal, of what the expert's internal representation of physics problems may be like and of the schemata that provide the source of his physical intuition. Observations of experts solving problems in kinetics (*11*) begin to provide us with empirical evidence for the reality of representational schemata of this general kind and with a basis for modeling them in greater detail.

Strategies in a Simple Domain: Kinematics

Empirical evidence for some of the differences between expert and novice strategies can be found in problem-solving accounts from simple problem situations in physics (*12*). An early chapter of a typical elementary physics textbook is devoted to kinematics—specifically, motion under uniform acceleration or deceleration. Measured in terms of the number of new concepts introduced or the number of equations, the content of such a chapter is not large. There are about 11 formulas, some of which are redundant, that express the relations among the various quantities involved [for example, $v = v_0 + aT$, $\bar{v} = \frac{1}{2}(v_0 + v)$ and $S = \bar{v}T$, where S is distance, $\bar{v}$ is average velocity, T is time, v is terminal velocity, v_0 is velocity at the origin, and a is acceleration].

Mastery of the entire chapter requires the student to learn only about ten "things"—concepts and laws. If this chapter is typical of the whole text, then a 1-year physics course calls for the mastery of about 300 "things." Again, if this course is typical of high school or college courses, a student carrying four courses might be expected, during a school year, to learn 1000 to 2000 "things." Learning at this rate, a student might acquire, over a decade or so, the 50,000 perceptual chunks that the chess master is thought to acquire over a comparable interval.

What is involved in expertness in solving kinematics problems? Several differences in solution process were revealed by a comparison of expert and novice solutions to typical textbook kinematics problems. The expert had strong mathematical skills and extensive experience in solving problems in mechanics; the novice had fair skill in algebra but had only recently studied the kinematics chapter and was doing problems such as the following for the first time.

> A bullet leaves the muzzle of a gun at a speed of 400 m/sec. The length of the gun barrel is 0.5 m. Assuming that the bullet is uniformly accelerated, what is the average speed within the barrel?

The most obvious difference between the subjects was that the expert solved the problems in less than one-quarter of the time required by the novice and with fewer errors.

A second difference, verified from their worksheets and the thinking-aloud protocols they produced, was that the novice solved most of the problems by working backward from the unknown problem solution to the given quantities, while the expert usually worked forward from the givens to the desired quantities. This was surprising, since working backward is usually thought to be a more sophisticated strategy than working forward. But experts work forward only on easy problems, where experience assures them that, without any particular planning, solving all possible equations will lead them quickly to a full understanding of the situation, including finding the particular quantity they are asked to solve for. They thus solve the problem by accumulating knowledge about the quantities that were initially unknown. Novices, having little experience with kinematics, seem to require goals and subgoals to direct their search. The management of goals and subgoals—deciding periodically what to do next—may occupy considerable time and place a substantial burden on limited short-term memory.

The solution paths followed by the expert and the novice can be simulated by

production systems representing the working-forward and working-backward strategies, respectively. In these production systems, the action part of each production is one of the kinematic equations; the condition part is a list of variables in the equation. The two production systems differ in only one important respect. In the expert system, the rule of action is: If you know the values of all the independent variables in any equation (condition), try to solve for the dependent variable (action). In the novice system the rule of action is: If the dependent variable in an equation is the desired quantity (condition), try to solve the equation (action); if the values of some of the independent variables are not known, create a goal to find the values of these variables.

A third difference between expert and novice is that the latter, in her accounts, mentioned each equation she was about to use, then substituted into it the values of the independent variables. The expert usually mentioned aloud only the numerical result of the substitution, not the original literal equation. This may be merely a difference in verbalization: The expert, working much more rapidly than the novice, simply did not have time to verbalize everything. Another interpretation seems more likely, however. The expert apparently had stored directly (perhaps as a production) an entire procedure for obtaining a desired value from related known values; he then applied this procedure and stated only the result he obtained. The novice, in contrast, had stored the knowledge that particular equations can be used to obtain values of certain variables. Hence, the verbalized result of her recognition was the equation itself. Substituting values for variables and solving the equation were treated as distinct, separately verbalized steps.

At each step in the path, the novice had to ask herself, "What do I do next?" She obtained the answer partly by looking at the equations she had written and determining what information had to be developed to fit them. In addition, unachieved goals held in short-term memory helped identify relevant actions.

Although our knowledge of the control structure of the novice's program is incomplete, there is ample evidence that frequent tests had to be performed to determine the sequence of actions and that the testing process consumed considerable time. The expert, on the other hand, had "automated" many sequences, so that they could be carried out without the need for recurrent testing. For him identifying the right equation, substituting the values of the independent variables in it, and solving for the dependent variable were carried through to the end as a single step, without reference to goals or cueing from symbols written on the worksheet.

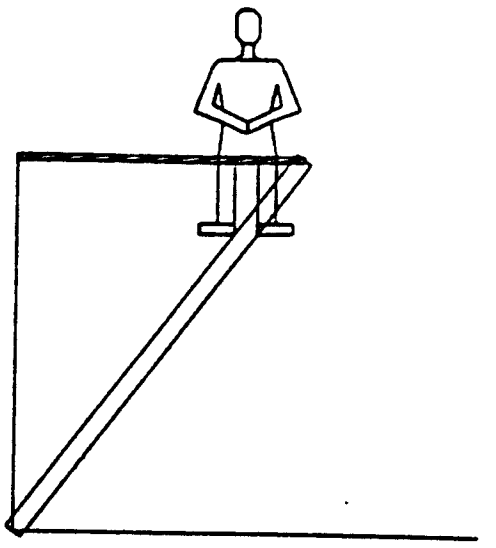

Fig. 2. Computer-drawn problem statement from (*9*).

Computer science provides us with one clue to the cause of this difference between the step-by-step procedure of the novice and the longer leaps of the expert. Computer instructions may either be executed interpretively or be compiled. In interpretive execution, a control system, the interpreter, orders the execution of each successive step, testing information in memory to determine what step is called for. Compiling the program eliminates many of these tests, welding whole sequences of program steps together into segments. Typically, a compiled program is executed about ten times as fast as the corresponding interpreted program.

In at least one instance (*13*), compiling has been simulated for a human cognitive process. A program called UNDERSTAND interprets instructions for puzzlelike problems written in English (much as STUDENT interprets algebra problems and ISAAC interprets lever problems). UNDERSTAND builds up representations of the problems in memory and generates programs to make legal moves. The legal moves must be interpreted appropriately to match them to the problem representation. Having constructed these components of the problem-solving system, UNDERSTAND is capable of compiling the legal move operators, gaining a factor of about eight in their speed of execution.

The notion of compiling provides us with a hypothesis for why the expert can carry out a sequence of problem-solving steps in kinematics four times as fast as the novice can carry out almost the same steps. Although the production systems that simulate the two subjects are similar, we hypothesize that the novice executes the productions interpretively, while the expert executes them in compiled form.

A fourth possible difference between expert and novice, may be indicated by some sketchy evidence from the thinking-aloud protocols of the simple kinematics problems. The novice often seemed to use a process of direct syntactic translation, much like that used by STUDENT. The expert, on the other hand, seemed to generate some sort of physical representation, in which accelerations produce velocities and velocities produce distances traveled. The kinematics problems are so simple for the expert, however, that we found only rudimentary traces of his semantic processes and his physical representation in the protocols. To obtain more information about physical representations, we must turn to somewhat more complex problems.

Expert Performance in Dynamics

McDermott and Larkin (*14*) and Larkin (*15*) have built simulation programs for the behavior of a single expert solving problems in kinetics. These programs bypass the natural language translation step [which has been studied and simulated by other investigators (*7–9*, *16*)] and use as input an encoding of the picture of the problem to be solved. The simulations stipulate the main stages of solution. (i) If the problem statement is not accompanied by a picture, the expert will sketch one. (ii) Selecting tentatively a set of principles to use, the expert will construct an abstract problem representation containing physical entities (such as forces and energies) relevant to those principles. In sufficiently complex problems, this representation is often written explicitly (for example, the commonly used "free-body" diagram). (iii) Finally, the expert will rerepresent the problem as a set of equations.

The simulation program begins with step (i) already completed (the problem is presented to the system schematically) and carries out steps (ii) and (iii). Consider the problem shown in Fig. 3 (*17*). Using node-link structures, the problem is represented in the McDermott-Larkin program by schemata for block B, block A,

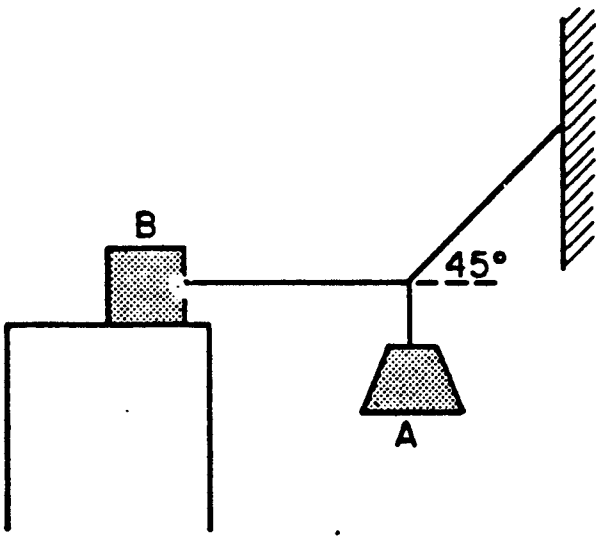

Fig. 3. A simple mechanics problem (*17*). Block *B* weighs 160 pounds. The coefficient of static friction between the block and the table is 0.25. Find the maximum weight of block *A* for which the system will be in equilibrium.

the junction point of the strings, the anchor for the diagonal string, the contact between block B and the table, and the contacts corresponding to the three strings. Two of these structures are shown in Fig. 4.

In problems of the sort shown here, skilled physicists generally seem to use a different representation of the problem for each of the three main stages (the labeled sketch, the sketch containing physical entities, and the equations). Pictorial representation is convenient. Much of the difficulty with problems in mechanics lies in understanding the spatial relations among the objects. Moreover, the pictures drawn can be highly stylized, abstracting away irrelevant information in the verbal problem statement.

The expert also spends time constructing the still more abstract representation, for at least three reasons: (i) to quickly determine whether the qualitative approach to the problem is appropriate, (ii) to identify the forces and energies at work and to represent them in a uniform way for all parts of the problem, and (iii) by decoupling the discovery of the forces or energies from the generation of equations, to reduce the amount of information that must be attended to at any one time.

The problem-solving system that operates on these representations is a production system. Each production encapsulates some small part of the expert's knowledge, and the condition side of each is constructed to evoke the knowledge at just those times when it will be relevant to the problem. With a few exceptions, one of the condition elements for each production is the description of a goal, so that the production will be executed only when that goal is active. The set of productions that share the same goal condition constitute a method. The system at present has about 40 methods, ranging in size from 3 or 4 productions to 20 or 30. Organizing the system's knowledge in this way confines its attention to just that part of its knowledge associated with the goals it has not yet achieved. The hierarchical arrangement of the methods allows knowledge relevant to many different methods to be associated with a very general method, toward the top of the hierarchy.

The recognize-act cycle characteristic of production systems makes the system continually responsive to the current state of the problem. Although methods impose a structure on production memory and some orderly sequence of attention from general to specific goals, individual productions within a method are not evoked in any fixed sequence; the sequence is determined by the knowledge in short-term memory that indicates what is currently of interest. We hypothesize that this is also the way the expert's attention and sequence of goals is controlled.

The simulation programs for describing expert performance and contrasting expert with novice performance in physics cast light on what the expert needs to know and on how knowledge is represented in memory, but they leave many unanswered questions. Only narrow domains within physics have thus far been explored. We have mentioned kinematics and several kinds of simple problems in mechanics. Some work has been done on chemical engineering thermodynamics (*18*) and some on electrical circuits (*19*), revealing problem-solving processes not unlike those we described.

Name:	Block B
Type:	Object
Subtype:	Block
Mass:	160
Motion	None
Contacts:	
Right:	1 String
Down:	2 Table

Name:	Contact 2
Type:	Contact
Subtype:	Surface
Objects:	B, Table
Quality:	Rough
Static friction:	0.25

Fig. 4. Some of the node-link structures describing the problem situation of Fig. 3.

Recent Developments

1) Interaction between the problem solver and external memory aids—the use of paper and pencil. The capacity of short-term memory (working memory) constitutes one of the most severe constraints on human problem solving. The capacity of short-term memory (probably about four familiar items, or "chunks") limits the ability of a chess-player to reproduce a board position he has seen for a few seconds. The outputs and inputs to all conscious mental activities have to pass through short-term memory and be held there for a brief interval. It is not clear whether the current task goals also have to be held in this same memory or whether there is some additional capacity for them elsewhere; but making the most generous allowance for control information, human problem solvers are almost certainly not able to retain more than twice that—about eight chunks—unless they take time (perhaps about 8 seconds per chunk) to fixate the information in long-term memory.

Paper and pencil provide an unlimited extension of the problem solver's working memory capacity, but at the cost of writing down the information (which can be done more rapidly than it can be memorized) and of gaining access to it when it is needed. To model human problem solving, it is necessary to distinguish between internal and external memory and to provide a specific role and specific processes for the latter.

Larkin (*20*) has constructed a simulation program for solving physics problems that models external as well as internal memory. The basic idea is that many of memory's productions (perceptual productions) are activated by noticing or recognizing pieces of information or relations that are recorded on a worksheet. To allow ready access to information, two mechanisms organize the worksheet. (i) Closely related information (for example, known and desired quantities, information about one subsystem) is written together. (ii) Information is generally added sequentially, so that recent information (likely to be relevant to current goals) is near the bottom of the paper.

New information (a new equation, a new value for a variable) is written on paper. The recognition on the worksheet of something familiar, or relevant to a current goal, evokes new action, which may, in turn, produce new information. In this respect the problem-solver is "stimulus driven"—there is a constantly repeated cycle of interaction between thought processes and the gradually developing body of information on the worksheet.

2) Representations of situations that change through time. Simple dynamics problems (such as pulley problems, in

which the disequilibrium of forces produces a constant acceleration) can be handled with static descriptions of the situation at a typical moment and without any special concern for changes with time. With problems more complicated, this simplification may not hold. Consider the following kinematics problem:

> At the moment car A is starting from rest and accelerating at 4 m/sec², car B passes it, moving at a constant speed of 28 m/sec. How long will it take car A to catch up with car B?

Skillful subjects construct some representation of this problem which allows them to make inferences easily about the relations of times and distances. The simulation program uses a node-link scheme to produce a "mental picture" of the cars.

The representational scheme organizes all events around time instants that bound time intervals, resulting in the following (partial) representation for the problem.

Position [(car A) (instant 1)] start
Position [(car B) (instant 1)] start
Position [(car A) (instant 2)] pass
Position [(car B) (instant 2)] pass

The simulation program contains productions that recognize that if two objects (cars) have the same relative positions at instant 1 and again at instant 2, they have traveled the same distance during the interval, T, between 1 and 2. It then infers that, for the first car, S is related to T by $S = \frac{1}{2}aT^2$, and for the second car by $S = \bar{v}T$, and can now solve these two simultaneous equations (since a and $\bar{v}$ are known) for S and T. The importance of these equations is recognized in the same general manner as in the simple production systems for kinematics described earlier, again with different criteria for evocation being used by the expert and the novice.

The representation scheme in the new program is sufficiently comprehensive and flexible to handle virtually all the types of mechanics problems we have mentioned. Moreover, it is also able to handle standard problems about chemical engineering thermodynamics. Consider, for example, the following problem:

> Nitrogen flows along a constant-area duct. It enters at 40F and 200 psi. It leaves at atmospheric pressure and at a temperature of −21°F. Assuming that the flow rate is 100 lb/min, determine how much heat will be transfered to the surroundings.

Here, the state of the gas at entry to the process is associated with instant 1 and the state at exit with instant 2; the processes taking place in the duct are associated with the interval T. The equation for conservation of energy associates the difference between input and output states with the processes occurring between them. The equations of state for various substances are stored in memory in association with those substances so that they can be evoked when appropriate.

3) Learning to be expert. As we begin to gain a picture of the expert's knowledge, our curiosity is aroused as to how expertness might be produced.

A beginning physics student listens to lectures, studies a textbook, and works problems. One avenue to an understanding of learning processes would be to write programs that would allow a computer to read textbooks (assuming these to be roughly interchangeable with lectures) and work problems and thus reach some level of skill and knowledge in physics.

Some hypotheses are necessary to account for the shape the learning program is to take. The most promising candidates are adaptive production systems (APS's). An APS can grow by creating new productions and incorporating them in itself so that they will be evoked when appropriate. Such systems have now been tested for a number of simple tasks (*21*, *22*).

The idea underlying APS's is that effective learning involves more than (and differs from) memorizing materials presented in texts and lectures. Each production in a production system has a condition and an action. The action controls what is to be done by the production, whereas the condition defines when the production is to be evoked and executed.

Every textbook chapter on kinematics presents and explains the basic kinematic equations, but few if any explain how the student is to judge when a particular equation is appropriate for a particular problem. Thus, the textbook typically presents explicitly the material for the action parts of the productions the student must acquire, but does not present the cues of the condition parts. Nor is this asymmetry peculiar to the subject-matter of kinematics; it is a general characteristic of textbooks. (It is less characteristic of "how-to" books on athletic and motor skills and of modern chess books.) Careful study of the consequences of the asymmetry for learning could significantly improve textbook construction and instruction. Eylon (*23*) has shown, with problem-solving making use of Archimedes' principle, that explicit instruction about the occasions for using procedures was distinctly advantageous. Landa (*24*) has analyzed the conditions for applying procedures in subjects like grammar and high-school mathematics.

An APS learns by generating new productions. If (as is usually the case) such productions are not suggested directly by the instructional environment, the system must be able to build them from available information. Two strategies for doing this may be called "learning by example" and "learning by doing."

Textbooks commonly display examples of solved problems. These examples carry through the solution step by step, often stating the justification for each step. The following is a typical example in algebra.

$3x + 4 = x - 12$	
$2x + 4 = -12$	(subtract x from both sides)
$2x = -16$	(subtract 4 from both sides)
$x = -8$	(divide both sides by 2)

The student has already been told that equal quantities may be added to or subtracted from both sides of an equation and that both sides may be multiplied or divided by the same quantity. Hence, the student comes to this example with the set of actions needed for new productions. But the textbook is silent on why particular actions are taken in a particular order.

Neves (*22*) has constructed an APS that learns from examples such as this one. It inspects each pair of successive steps in the derivation to determine what change has been made that reduces the distance between the initial step of the pair and the final expression. The first action, for example, gets rid of the unwanted literal term on the right side of the equation; the second gets rid of the unwanted numerical term on the left side. Neves's APS is capable of forming a production like, "If there is a literal term on the right side of the equation, subtract that term from both sides of the equation." Thus, it generates an appropriate condition to associate with the action.

It can be seen that three productions, one for each of the three steps in the example, will, when generalized to replace specific coefficients with variables, constitute a fairly general algorithm for solving linear algebraic equations in one unknown (not quite, for additional productions are needed to carry out the simplification steps left implicit in this example). Thus, Neves's APS is able to acquire skills in subjects like algebra by working through a few examples.

A system that can learn from examples can learn by doing if it is supplied with one additional capability. Suppose that the system has a simple problem-solving component that enables it to solve (some) problems by trial-and-error search. Trial-and-error search is, of course, inefficient for solving problems, but if the problems are simple enough, it sometimes succeeds. Once a problem has been solved, the system has acquired a new example, which its learning-by-example component can then use to bootstrap itself. Having strengthened its problem-solving capabilities through the new productions it has acquired, the system can now solve some slightly more difficult problems, and use these again as examples from which to learn.

The learning-by-doing system accomplishes its learning by hindsight. Perhaps demonstrating this program to students who abandon their homework problems as soon as they have found the answers would persuade them that it is only after one has solved a problem that one can learn most effectively how one should have solved it.

These explorations with adaptive production systems represent only beginnings, but they suggest the general kinds of paths a learning system must follow in order to acquire the sorts of programs used by experts in solving problems.

Conclusion

We have reviewed some recent findings about human problem-solving processes and especially about the sources of expert skills. In every domain that has been explored, considerable knowledge has been found to be an essential prerequisite to expert skill. The expert is not merely an unindexed compendium of facts, however. Instead, large numbers of patterns serve as an index to guide the expert in a fraction of a second to relevant parts of the knowledge store. This knowledge includes sets of rich schemata that can guide a problem's interpretation and solution and add crucial pieces of information. This capacity to use pattern-indexed schemata is probably a large part of what we call physical intuition.

Since we are now able to build production systems that simulate this rapid processing, intuition need no longer be regarded as mysterious or inexplicable. It is scarcely more mysterious that a skilled physicist can recover a particular equation from memory than that we can find a word in the dictionary in a few moments when we want to check its definition. Indexed node-link structures seem suited for both tasks.

Our growing understanding of an expert's knowledge and the kinds of processes an expert uses when solving problems enables us to begin to explore the learning processes needed to acquire suitable knowledge and problem-solving processes. We have no reason to suppose, however, that one day people will be able to become painlessly and instantly expert. The extent of the knowledge an expert must be able to call upon is demonstrably large, and everything we know today about human learning processes suggests that, even at their most efficient, those processes must be long exercised. Although we have a reasonable basis for hope that we may find ways to make learning processes more efficient, we should not expect to produce the miracle of effortless learning.

References and Notes

1. W. G. Chase and H. A. Simon, *Cognit. Psychol.* 4, 55 (1973); in *Visual Information Processing*, W. G. Chase, Ed. (Academic Press, New York, 1973), p. 215; H. A. Simon and W. G. Chase, *Am. Sci.* 61, 394 (1973); A. de Groot, *Thought and Choice in Chess* (Mouton, New York, 1978).
2. H. A. Simon, *Science* 183, 482 (1974).
3. _____ and M. Barenfeld, *Psychol. Rev.* 76, 473 (1969).
4. A. Newell and H. A. Simon, *Human Problem Solving* (Prentice-Hall, Englewood Cliffs, N.J., 1972).
5. A. Newell, in *Visual Information Processing*, W. G. Chase, Ed. (Academic Press, New York, 1973), p. 463.
6. J. R. Anderson and G. H. Bower, *Human Associative Memory* (Wiley, New York, 1973).
7. D. G. Bobrow, in *Semantic Information Processing*, M. Minsky, Ed. (MIT Press, Cambridge, Mass., 1968), p. 135.
8. J. M. Paige and H. A. Simon, in *Problem Solving*, B. Kleinmuntz, Ed. (Wiley, New York, 1966), p. 51.
9. G. S. Novak, Jr., *Tech. Rep. NL-30* (Department of Computer Sciences, University of Texas, Austin, 1976).
10. _____, in *Proceedings of the 5th International Joint Conference on Artificial Intelligence* (IJCAI-77, Cambridge, Mass., 1977), pp. 286–291.
11. J. H. Larkin, *Skilled Problem Solving in Experts* (Technical Report, Group in Science and Mathematics Education, University of California, Berkeley, 1977).
12. D. P. Simon and H. A. Simon, in *Children's Thinking: What Develops?*, R. S. Siegler, Ed. (Erlbaum, Hillsdale, N.J., 1978), p. 325.
13. J. R. Hayes and H. A. Simon, in *Knowledge and Cognition*, L. W. Gregg, Ed. (Erlbaum, Potomac, Md., 1974), p. 167.
14. J. McDermott and J. H. Larkin, in *Proceedings of the 2nd Conference of the Canadian Society for Computational Studies of Intelligence* (Univ. of Toronto Press, Toronto, 1978), pp. 156–164.
15. J. H. Larkin, *J. Struct. Learn.*, in press.
16. T. Winograd, *Understanding Natural Language* (Academic Press, New York, 1972).
17. D. Halliday and R. Resnick, *Physics* (Wiley, New York, 1966).
18. R. Bhaskar and H. A. Simon, *Cognit. Sci.*, 1, 193 (1977).
19. J. S. Brown and R. R. Burton, in *Representation and Understanding*, D. G. Bobrow and A. Collins, Eds. (Academic Press, New York, 1975), p. 311.
20. J. H. Larkin, "Models of competence in solving physics problems" (Complex Information Processing working paper No. 408, Department of Psychology, Carnegie-Mellon University, Pittsburgh, 1979).
21. D. A. Waterman, *Artif. Intell.*, 1, 121 (1970); J. Anderson, *Language, Memory and Thought* (Erlbaum, Hillsdale, N.J., 1976). Y. Anzai and H. A. Simon, *Psychol. Rev.* 86, 124 (1979).
22. D. Neves, in *Proceedings of the 2nd Conference of the Canadian Society for Computational Studies of Intelligence* (Univ. of Toronto Press, Toronto, 1978), pp. 191–195.
23. B. Eylon, thesis, University of California, Berkeley (1979).
24. L. N. Landa, *Algorithm in Learning and Instruction* (Educational Technology, Englewood Cliffs, N.J., 1974); *Instructional Regulation and Control: Cybernetics, Algorithmization, and Heuristics in Education* (Educational Technology, Englewood Cliffs, N.J., 1976).

25. Supported by grant SED78-21986 from the National Institute of Education and the National Science Foundation, and by contract F44620-73-C-0074 with the Advanced Research Projects Agency of the Office of the Secretary of Defense, which is monitored by the Air Force Office of Scientific Research.