In J. Anderson (Ed.), Cognitive skills and their acquisition. Hillsdale, NJ: Lawrence Erlbaum Associates, 1981.
Also in Models of Thought, II - ch. 3.1.

#517

12 The Central Role of Learning in Cognition

Pat Langley and Herbert A. Simon
Department of Psychology
Carnegie-Mellon University

INTRODUCTION

If we compare the past 2 decades of research in cognition, increasingly carried out within an information-processing framework, with the previous 2 decades, mainly Behavioristic and Hullian, we note a remarkable decline in the attention paid to learning. Maze learning experiments with rats, operant conditioning of pigeons, human rote-verbal learning experiments, and experiments in classical conditioning dominate the pages of the Journal of Experimental Psychology during that earlier period. In the later period, the focus on learning is replaced by a concern with performance, from the simplest reaction-time tasks to the most complex tasks of solving problems, recalling information from memory and understanding language.

Within the past 3 to 5 years, two developments have conspired to bring about an accelerating revival of research on learning processes as distinguished from performance processes. First, the study of performance has progressed to the point where we now have a good understanding of the cognitive skills that are required (at both novice and expert levels) in a considerable range of difficult cognitive tasks. This repertory of analyzed performance programs instructs us about the final products of the learning process, the targets at which learning must aim.

Second, new ideas have emerged about the nature and organization of learning processes. Perhaps the most important of these ideas is the *adaptive production system*, first demonstrated by Waterman (1970) in a dissertation describing a poker-playing system that was capable of improving its game.

Is there reason to think that the revival of learning research is more than a passing fad? Why has psychology exhibited such fascination with the phenomena of learning? Historically, there appear to have been two principal reasons for this preoccupation: one theoretical, the other practical. Psychology has only recently lost its close bonds with philosophy. A central question of epistemology is how we come to know the external world. Translated into psychological terms, this becomes a question of how we perceive and how we learn. One has only to go back to John Locke to see the common ancestry of the two disciplines, and it is still quite visible in William James' *Principles*.

On the applied side, education has always been one of the principal domains of practical use for psychological knowledge. Educational institutions, although they are demonstrably successful in imparting skills and knowledge, have had to proceed (as nineteenth century medicine did) by rule of thumb, and without any deep understanding of the learning processes they seek to nurture. If a genuine theory existed of how people learn, it would hold out great promise of marked improvements in educational practices.

The Search for Invariants

The reasons just stated for psychology's interest in learning are as valid today as they ever were. But another reason, as important as these two, can be added. The goal of cognitive psychology is to understand the workings of the human mind. The mind is an adaptive system whose biological function is to enable the organism to behave effectively and, hence, to survive in a complex, changing, and often unpredictable environment. Adaptation takes place on several different time scales. On the shortest time scale, each problem that the environment presents to the organism challenges its adaptive capacity; that is to say, problem solving and other performance skills are the most immediate adaptive mechanisms. On the longest time scale, the biological evolution of the genotype is the adaptive mechanism. Between the rapid processes of problem solving and the slow processes of evolution lie learning processes that gradually bring about improvements in the organism's performance programs on an intermediate time scale.

A difficult problem arises for psychology, because knowledge and strategies are not fixed but are modifiable through learning. Science searches for laws, for invariant regularities of behavior. In the presence of learning mechanisms, knowledge and strategies are not invariants of the mind but are highly malleable. Hence, the description of performance programs, however useful and essential that may be for psychology, will not achieve the goal of describing the human mind in an invariant way. A theory of learning may supply the desired invariants.

John Stuart Mill (1950) put the matter this way:

> In other words, mankind have not one universal character, but there exist universal laws of the formation of character. And since it is by these laws, combined with the facts of each particular case, that the whole of the phenomena of human action and feeling are produced, it is on these that every rational attempt to construct the science of human nature in the concrete and for practical purposes must proceed [p. 319].

The Search for Generality

During the past quarter century, the history of research in human cognition has closely paralleled the trends in artificial intelligence research. This is not surprising, because psychology has adopted information processing as its central paradigm. Most of the chapters presented in this volume make use of computer simulation as a technique for describing learning mechanisms in a fully operational way and for exploring their performance in complex task environments.

During the decade of the mid 50s to mid 60s there was, in artificial intelligence, a striving toward generality—toward programs capable of performing in a wide range of task environments. The General Problem Solver (Newell, Shaw, & Simon, 1960a) and EPAM (Feigenbaum, 1961) were characteristic products of that decade. In these programs, processes like means-ends analysis and recognition by means of discrimination nets were identified as possible invariants of human cognition.

In this same decade, there was considerable activity directed at developing learning programs, also employing very general mechanisms. Most of this kind of learning research was invested with a "clean slate" viewpoint. It preferred to begin with a system processing no initial knowledge and only very general learning mechanisms, hence to rely on learning as the source of all the system's final intelligence. The research on Perceptrons and other self-organizing networks was typical of this point of view (Rosenblatt, 1958). The results of the research were ultimately rather disappointing, and this line of inquiry has tended to die out.

During the decade from the mid 60s to mid 70s, attention in AI turned to professional-level performance in complex task domains. It was discovered, or rediscovered, that expert performance requires knowledge—large amounts of it—as well as wit. Taking a "knowledge-engineering" point of view, researchers built expert systems (e.g., DENDRAL (Feigenbaum, Buchanan, & Lederberg, 1971), INTERNIST (Pople, 1977), MYCIN (Shortliffe, 1976) in a number of task domains, each system limited strictly to a single domain and each resting solidly on a large data base.

Here was the contingent, learning-dependent character of human behavior exhibited in its starkest form. But even in the specialized expert programs, the general mechanisms, the invariants that had been discovered in the first decade, survived, for the large knowledge bases possessed by these programs had to be searched in intelligent fashion. Mechanisms like means-ends analysis and discrimination nets served in the expert programs as principal components of the control structure that guided search through the data bases.

Nevertheless, the proliferation of task-specialized programs during the second decade produced a certain amount of malaise. It seemed to imply that cognitive psychology was almost indistinguishable from an enormous encyclopedia of human knowledge and human skills, and hence was totally contingent on the change and growth of knowledge and its selective allocation to specialized human crania. That the knowledge-based systems required some common core of control structure was only a partial comfort and went only a part of the distance toward providing general, invariant, and parsimonious psychological principles.

Scope of this Chapter

The search for generality and invariance, together with the opportunity provided by the discovery of adaptive production systems, provides the background for the present decade's renewed interest in research on learning, and specifically for the chapters that have been presented in this volume. In a description of the learning process, we look for a new set of invariants to be adjoined to those of the first decade—invariants that will characterize human cognition even when that cognition applies itself to the almost limitless range of specialized tasks that humans are capable of learning to perform.

In this final chapter, it is not our intention to provide a detailed critique of the excellent chapters that have been presented in this volume (nor was that part of our bargain with the chairman), but rather to assess their common conceptual background—the *Zeitgeist* that we have just been describing. In order to do this, we need to say something about the characteristics that a scientific explanation should possess and to make some hypotheses about the multiplicity of learning mechanisms we should expect to find in a system as complex as the human cognitive system. These considerations lead us to ask whether there exist general principles of learning and whether these principles are fully invariant. Our conclusion that they are not leads us, finally, to the topic of learning to learn.

CHARACTERISTICS OF A GOOD EXPLANATION

In addition to stating an invariance, there are at least four additional criteria that a scientific explanation should satisfy: (1) it should explain a variety of phenomena; (2) it should be more basic than the phenomena it explains; (3) it should be simpler than those phenomena; and (4) it should be free of ad hoc components. In the following sections we discuss each of these criteria in turn, along with their implications for theories of learning.

Scope of Explanation

An explanation should summarize and predict data, empirical generalizations, or lower-level laws. The more of these it explains, the better; also, the more diverse the phenomena it explains, the better. A theory that explains only a single empirical law holds no advantage over the law itself.

The classical example of an explanation of broad scope is the law of universal gravitation, which provides a unified explanation for Kepler's laws of the planetary movements, Galileo's law of falling bodies, and the phenomenon of the ocean tides. We would presumably be satisfied with a little less generality than that in a learning theory, but we would be less happy with a system that, while purporting to produce learning about some specific task, itself used knowledge that was obviously specific to the task in question.

For example, we know that the first language a person learns is determined by the language he hears as a child. Thus, a theory of first language acquisition should account for the learning of Finnish and Hopi, as well as for the learning of English. Although an adaptive production system that could learn any Indo-European language would be impressive, we would be even more impressed by one that could learn Finno-Ugric tongues as well.

The proceduralization and composition processes discussed in this volume by Neves and Anderson and by Lewis are good examples of explanations of wide scope. These mechanisms are applicable to any domain in which behavior can be described as a production system (not a small set). Moreover, the composition process alone can account for four diverse phenomena: the speeding up of behavior, acquiring the ability to search in parallel, the loss of intermediate results, and Einstellung. In addition, Anderson has noted (personal communication) that, because composition leads to a reduction in memory load, it can enable behavior that had been impossible earlier because of short-term memory limitations.

Thus, learning mechanisms can be far more general than the knowledge or skills that are learned with their aid in any particular domain. In the chapters of this volume, essentially the same mechanisms show up repeatedly in such domains as geometry, physics, and programming. Learning theories are generally superior to theories of task performance on the criterion of scope of explanation and promise us some of the generality we were in danger of losing with the elaboration of specific individual theories for individual task domains.

Depth of Explanation

A good explanation should be more basic than the phenomena it explains and should not contain important aspects of those phenomena. We regard the law of gravitation as deep, because it derives the orbits of the planets from postulates about how their mutual attractions produce their accelerations—a distinctly nontrivial connection. At the other extreme, explaining a flower's odor by hypothesizing that its atoms have that odor would not be regarded as basic. (It would not even be true, but if true, still not a genuine explanation.) Nor would we accept an explanation of perceptual phenomena that postulated a homunculus inside the head to interpret the information presented by the retina.

A theory of rote memory that simply postulated that stimulus material was stored verbatim in long-term memory would be regarded as providing an exceedingly shallow explanation of learning—if, indeed, we were willing to regard it as an explanation at all. Moreover, such a theory would not predict any of the empirical phenomena of learning other than the simple fact that memory exists, and information can sometimes be transfered to it.

On the other hand, a learning mechanism that took natural language statements as its inputs and produced procedures as its outputs could lay claim to being a relatively deep explanation of the sources of the skills represented by the procedures. (Whether it would be an empirically *valid* explanation of particular phenomena is a separate question.) Although the physics and geometry programs that have been discussed in this volume do not start with natural language, they do convert information from declarative to procedural mode and hence qualify as reasonably deep theories.

One common cause of shallowness in explanations is that too much knowledge (e.g., subject-matter knowledge) is built into them. Fortunately, such built-in knowledge may be detected quite easily by testing a model in different domains. If a theory that purports to explain the learning of language assumes a particular fixed word order, its shallowness is readily exposed by giving it the task of explaining learning in a language that uses a different word order. This suggests that, usually, shallow explanations will fare poorly on the scope criterion; conversely, narrowness of scope is a signal that domain-dependent knowledge may have been built into a system.

More subtle homunculi may hide in the representation of data that the learning theory assumes is available. Some theories, though incorporating learning mechanisms that are apparently general, may produce learning only if a particular representational scheme is used. Because a very large part of human learning, at least in school subjects, begins with oral or written natural-language input, learning theories that do not accept such input have, to that extent, somewhat less depth than we might wish and also need to be tested for scope.

Shallowness of an explanation may be revealed by the fact that it is too facile—that it describes as simple phenomena that can be shown empirically to be complex. Most adaptive production system models learn too quickly because their transformations are too powerful. In this respect, the mechanisms for generalization, discrimination, and strengthening incorporated by Anderson in the ACT system are more satisfactory, for they require interaction with environmental stimuli over long periods to produce an effective performance system.

In comparison with performance systems, all learning systems introduce at least one more step of indirection between environmental stimuli and successful task performance. Most learning systems inject several such intervening steps. Hence, learning theories meet the depth criterion to a greater degree than do performance theories.

Simplicity of Explanation

Although an explanation need not be simpler than every lower-level law it explains, it should be simpler than the collection of these laws taken together. An adaptive production system for explaining some kind of learning that created fewer new productions than the number of productions it contained would not be regarded as a very helpful explanation of the learning.

Simplicity refers to the explanation itself, not the derivation that takes us along the entire path from explanation to empirical phenomena. Hence, an explanation may be simple, yet deep.

Most adaptive production systems score rather well on the criterion of simplicity. One frequently used strategy in such systems has been called "learning by doing" (Anderson, Greeno, Kline, & Neves, this volume; Anzai & Simon, 1979). Weak general methods are used to perform a task (e.g., solve the Tower of Hanoi puzzle, prove a theorem in geometry). The solution then provides information to the mechanisms for creating new productions that enables them to build more powerful solution procedures, which are usually specific to the task domain. Both the weak general problem-solving methods and the adaptive productions are nearly task independent and relatively simple.

Freedom from Ad Hockery

An explanatory mechanism will often consist of a number of (possibly interacting) components. Whereas not all of the components may be essential for explaining any specific phenomenon, the explanation cannot be regarded as parsimonious unless most components are needed for explaining most of the phenomena. Special-case or ad hoc components should be avoided.

Freedom from ad hockery tends to go with scope of explanation. Conceptually at least, none of the learning theories presented in this volume incorporate ad hoc components, for all are, in principle, capable of being extended to task domains beyond those for which they were specifically designed and would use the same mechanisms in those new task domains.

Summary: Criteria of Explanation

Our survey of the criteria that might be used to evaluate theories of human cognition shows that on all scores, except possibly for simplicity, it is usually easier for learning theories to meet these criteria than performance theories. Human learning programs are more likely to be invariant over time then are performance programs, to apply to wide ranges of tasks instead of to special task domains, to be deep, and to be free from ad hoc mechanisms. Hence, there seems to be a sound basis for the current return to learning as a (if not *the*) central preoccupation of cognitive psychology.

LEARNING IN COMPLEX SYSTEMS

Learning is any process that modifies a system so as to improve, more or less irreversibly, its subsequent performance of the same task or of tasks drawn from the same population. For example, if a person solves the Tower of Hanoi problem twice in a row, requiring less time and/or fewer moves the second time, we would attribute the change to learning. We would also say that learning is taking place if solution times decrease as a person solves a sequence of different algebraic equations, or if a person proving the second of two theorems proves it more rapidly than another person who hasn't proved the first.

In understanding a new task, a person stores away information that may show up as learning in a subsequent situation. After the four-disk Tower of Hanoi problem has been understood (even if not solved), it does not take as long to understand the five-disk problem. Here understanding, per se, is not learning, but the process of understanding may result also in learning. Similarly, although problem solving is not learning, it may leave behind a learned residuum (e.g., the problem solution may be remembered and used to help solve another problem).

If a system has many components as the human cognitive system has, there may be many ways in which it can be modified, each constituting a different form of learning. Hence, it is more realistic to speak of a theory of learning *mechanisms* than *a* theory of learning. The human performance systems in which we are interested make use of large knowledge bases and sophisticated strategies, some of which are specific to particular task domains (e.g., strategies for shifting gears in an automobile), whereas others are relatively general (e.g., means–ends analysis and other problem-solving strategies).

Learning may involve modification of any component of the information processing system, including:

1. Additions to or reorganization of its knowledge base: an associative node-link memory organized in schemata.
2. Augmentation of the recognition mechanism, or index, for the knowledge base: a discrimination net (EPAM net).
3. Augmentation of search strategies: organized as production systems.
4. Modification of evaluation functions stored in memory and used to guide search.
5. (Apparent) augmentation of short-term memory capacity by storing new chunks in long-term memory.
6. Augmentation of lexical, syntactic, and semantic knowledge in language-processing systems.
7. Enrichment of the representations of information (ways of organizing information) in memory.

Internally, all these changes can take the form of additions to, or alterations in, data structures or production systems.

The Knowledge Base

The most direct kind of learning is the accumulation of a knowledge base, declarative in form, which might be acquired, for example, by processing input in natural language or in the form of list structures. In the chapters of this volume, several learning processes, including the encoding stage of the learning described by Neves and Anderson and the remarkable use of mnemonics reported by Chase and Ericsson, have special relevance for this aspect of learning.

Efforts at modeling the acquisition of knowledge and the use of knowledge in enhancing subsequent performance go back to the beginnings of artificial intelligence and cognitive simulation. The Logic Theorist (LT: Newell & Simon, 1956), the earliest theorem-proving program that used heuristic search, was able to store in memory the theorems it proved and to use them as premises in proving new theorems.

Storing knowledge in usable form is more than rote memory. It requires conversion of the knowledge to a form of data representation that makes the newly stored knowledge available to the problem-solver's processes. In LT, this was accomplished by storing theorems as list structures (alias node-link structures, or schemata), so that axioms and theorems served as inputs to the search processes and were produced again, in exactly the same form, as outputs. With this kind of homogeneity of memory, search programs have a recursive structure. Search processes operate on expressions to produce new expressions, or chunks are combined into new chunks having the same form and the same capabilities for combination.

Recognition Mechanisms

Studies of the skill of chessmasters, and particularly of their prodigious abilities to recognize familiar patterns of pieces, have demonstrated the important role of recognition abilities in expert performance and the vast amount of learning necessary for acquiring these recognition skills. Models of these learning processes date back to the early years of cognitive simulation. The EPAM program postulated mechanisms for growing discrimination nets, employing sequences of feature tests to recognize patterns presented repeatedly as stimuli. Concept attainment programs were constructed by Winston and others that responded to appropriate stimuli by building up trees of tests enabling them to discriminate among familiar classes of objects.

Studies of the differences in method between experts and novices, of which the volume's chapter by Polson, Atwood, Jeffries, and Turner is an excellent example, almost uniformly attest to the importance of recognition in expert performance. Their most expert subject instantly recognized the solution, known to him from the professional literature, of the key subproblem of the complex software design problem they presented to their subjects.

There was little explicit mention of recognition processes in the learning models reported in this volume, but a deeper analysis of the structure of production systems shows that such processes are invariably embedded in them. The condition side of a production is in fact a set of tests that recognizes a situation as appropriate for the execution of that production. Moreover, if the production system is at all large, it is much too time-consuming to search productions sequentially for their applicability. Instead, there are built into such systems automatic processes for "indexing" the productions—(i.e., constructing a hierarchic organization of the tests, so that they can be searched in "twenty-questions" fashion instead of sequentially).

In research on learning, not much attention has yet been given to the nature of the elementary tests or features that are used to build up recognition nets. Certainly the sources of origins of such features are not addressed in any of the chapters that have been presented here—they are simply taken for granted in the conditions that are incorporated in the productions. For example, in the geometry programs, it is assumed that the simulated subject can recognize a triangle or a line segment. All the systems that have been described possess, at the outset, the ability to discriminate among strings of alphabetic characters. Hence, characterization of the elementary features that can be discriminated and explanation of how these features combine into the tests that appear on the condition sides of productions need to be placed on the agenda for research.

Strategies

The central focus of most of the chapters presented in the volume has been upon the learning of problem-solving strategies for specific task domains. It is here that the machinery of adaptive production systems shows itself to best advantage.

As with the acquisition of knowledge bases and recognition capabilities, the acquisition of strategies has a history that predates the current prominence of production systems. At the heart of the General Problem Solver (GPS) is a so-called "table of connections." This table associates with each of the differences between present situation and goal situation that GPS can detect one or more actions that might be relevant for eliminating or reducing that difference. A number of schemes were proposed, but never completely tested, for acquiring the table of connections through learning (Eavarone & Ernst, 1970; Newell, 1963; Newell, Shaw, & Simon, 1960b). Today, we would recognize the differences and actions in the table of connections as the condition and action parts, respectively, of productions.

But real impetus for learning schemes of this kind came with the explicit introduction of production systems and Waterman's initial design of an adaptive production system for learning to play poker (Waterman, 1970). To this were added various ideas for the source of the information the adaptive productions would employ: Neves' (1978) notion, for example, of learning from worked-out textbook examples and Anzai and Simon's (1979) notion of "learning by doing."

In the present volume, systems for learning strategies are proposed in the chapters by Neves and Anderson, by Anderson, Greeno, Kline, and Neves, by Hayes-Roth, Klahr, and Mostow, and by Rumelhart and Norman. The first two of these chapters use explicitly the machinery of adaptive production systems. The volume's chapters demonstrate a high level of activity and considerable success in developing mechanisms for the acquisition of new strategies.

Evaluation Functions

When problem solving is carried out by heuristic search, evaluation functions of some kind are needed to assess the promise of different branches of the search tree, hence to control the continuation of the search. As early as 1959, Arthur Samuel, in his research on checkers programs, showed that a modifiable evaluation function could be a powerful learning mechanism for improving a system's performance. Samuel used weighted linear functions of elementary features to evaluate positions reached in the look-ahead search. On the basis of its playing success or failure, it could change the weights in these functions or even eliminate some of the less useful features and introduce new ones from a predetermined list. This simple learning procedure enabled his program to progress in a few hours' play from novice status to state-championship level.

None of the chapters that have been presented discusses learning by the modification of evaluation functions. Perhaps one reason for this is that the task domains under study—at least geometry and elementary physics—do not call for extensive heuristic search with the construction of large search trees. However, in the chapter by Anderson, Greeno, Kline, and Neves, processes of generalization, discrimination, and strengthening of associations make use of statistical information about success and failure to tune search. Hence, the statistics that are computed may be viewed as implicit evaluation functions.

Chunking: STM

If, as is generally believed, the capacity of STM is limited to a very small number of chunks—perhaps only three or four—then an enlargement of chunk size can increase the informational content of each chunk and consequently ease the restrictions on processing imposed by the STM limits. But enlargement of chunk size entails an elaboration of the discrimination net that indexes information held in long-term memory, for the larger chunks will be more specialized, hence more numerous. Hence, the learning of discriminations should, as John Anderson observed, serve to relieve somewhat the STM bottleneck.

Two chapters here address the chunk-size question. Newell and Rosenbloom try to show how the observed data on gradual and continuing improvement of skills with practice can be explained as a consequence of chunk acquisition. Chase and Ericsson, in interpreting the performance of their mnemonist, propose that the use of schemata already available in LTM as "placeholders" for newly presented information may make it possible to transfer that information very rapidly to LTM and hence to avoid the congestion of STM.

The work of Shiffrin and Dumais, exploring the effects of a variety of variables on the development of automatism in performance, casts additional light on the chunking phenomena, as well as on the process of composition studied by Lewis in previous works, and in this volume by Neves and Anderson.

Language

None of the chapters in the volume has addressed directly the question of how humans acquire language, and particularly their first language. This topic, a central one for both cognitive and developmental psychology, has also been the focus of earlier research. Some years ago, Uhr built a system that was capable of acquiring a simple syntax, and about a decade ago, Siklóssy (1972) built the ZBIE system, which, when presented with a symbolic representation of a simple scene together with a sentence describing that scene, was able to extract syntactic information about natural language from the pairing. Several other language-learning schemes of a similar kind, and of gradually increasing power, have followed.

In these language-learning schemes, it is assumed that the simulated subject is already able to store perceptual information, for example, to store simple structures involving two or three objects and relations among them. Learning language means (at least) acquiring the ability to map appropriate language strings on these perceptual structures, not simply by rote, but in such a way that new combinations of objects and relations can now be described in words. Thus, having learned "bat hits ball," "boy drinks milk," and "man throws brick," such a system should be able to handle "boy throws ball" when presented with the corresponding percept.

Because it has already been accomplished at a simple level, it is clear that relatively simple formalisms are capable of supporting schemes for learning syntax—that is, for learning to map language structures onto perceptual structures, and vice versa. Developing such schemes of progressively greater power should be high on the agenda for research on learning.

Representations

Another centrally important aspect of learning that has been only lightly represented in this volume is the learning of effective representations for handling problems in different task domains. Two of the chapters did, however, have relevance for this topic. The data obtained by Chase and Ericsson provide clear evidence of the stages of development of the schemata their subject used as part of his mnemonic system. The data provide a rather remarkable picture of the gradual growth of a pattern of list structures in long-term memory. The chapter by Rumelhart and Norman gives us a theoretical view of how new schemata can be obtained by borrowing and modification of analogous structures already in memory.

Not many existing systems model the acquisition of new schemes of representation. Perhaps the clearest example is the UNDERSTAND program (Hayes & Simon, 1974), which, given a natural language description of a simple problem or puzzle, creates a representation of the problem space and operators that enables a GPS to go to work on the problem. Given a description of the Tower of

Hanoi problem, for example, UNDERSTAND creates symbolic entities that denote *pegs* and *disks* and a list structure that describes the arrangement of the disks on the pegs. It then constructs a legal move operator that is capable of modifying the list structure by moving disks from one peg to another.

In the case of the Tower of Hanoi, UNDERSTAND represents a single problem. However, the same mechanism can be used to represent a whole class of problems from a domain. Provided with information about *theorems* and *proof steps*, it could construct a simple set of schemata for representing the domain of geometry.

Novak, in his ISAAC program, showed how a set of schemata stored in LTM could be used to guide a system in its understanding of physics (statics) problems. A program like UNDERSTAND (or, more accurately, a more powerful version of it) could be used to create the sorts of schemata that ISAAC employs.

Hence, we would add to our list of items for the agenda of research on learning the study of how representations can be generated from descriptions of tasks in natural language.

ARE THERE GENERAL PRINCIPLES OF LEARNING?

Whatever part of the information-processing system is adaptively modified by learning, certain basic principles define the conditions under which learning can take place and describe some of the concomitant phenomena. Six of the most prominent and central of these are:

1. *Knowledge of results*. The learning mechanism must be able to detect improvement or degradation of the performance system.[1] In the psychological literature, knowledge of results, a cognitive mechanism, is often confounded with reward, a motivational mechanism. The term *reinforcement* embraces both elements. This volume has been primarily concerned with the cognitive aspects of learning and has ignored the motivational (without denying in any way their great importance in human learning).

2. *Generation of alternatives*. The learning mechanism must be able to try alternative behaviors, hence must be able to generate them in some manner unless they are provided by a teacher. One of the best ways to learn is to make mistakes, but errors can produce learning only if new behaviors are then attempted.

3. *Causal attribution*. Learning in a complex system can be efficient only to the extent that good or bad performance can be attributed accurately to specific

[1]In some concept attainment and pattern recognition tasks, knowledge of results may be supplied by internal criteria of "goodness" of the pattern description, hence may not require feedback from external sources.

components of the system. The system can learn from its mistakes only if it can discover the source of its error.

4. *Hindsight*. Knowledge of results *follows* performance. Hence efficient learning will reexamine and reevaluate past performance in terms of subsequent knowledge of results and causal attribution. (The values of hindsight are known to most A students and to few C students.)

5. *Learning from instruction*. A teacher can provide knowledge of results, information to be stored, a sequence of problems or examples, causal attribution, algorithms, and so on, with large effects on what is learned and how fast.

6. *Automatization*. Continued practice of skills that have been learned will cause continuing improvement in speed and accuracy of performance.

The hallmark of theorizing-by-simulating, well illustrated by the chapters of this volume, is that principles of learning are not stated explicitly in classical fashion but are embedded in programs that actually learn. To see what general principles are exemplified by the programs, we must examine and compare program structures and discover the communalities among them.

Knowledge of results, for example, is implicit in any scheme for learning by doing. In such a scheme, problems are first solved by weak methods often involving much trial and error. Once a correct solution path has been discovered (or provided directly by a teacher), knowledge of this path provides the information that an adaptive production system can use to create new productions that will henceforth permit the same or similar problems to be solved more efficiently. Such uses of knowledge of results are illustrated by the programs described by Neves and Anderson, by Anderson, Greeno, Kline, and Neves, by Hayes-Roth, Klahr, and Mostow, and by Larkin. A minimal knowledge of results (illustrated, for example, by LT) is knowledge of the correct answer; for the answer can be added to memory as a starting point for new searches.

Likewise, all the learning programs described in the volume are capable of *generating alternatives*, primarily in the course of their problem-solving searches. Schemes using analogy, such as the one sketched by Rumelhart and Norman, must have capabilities for modifying schemata that are already available to fit them to new situations.

Brown and de Kleer develop a formal scheme for *causal attribution* that permits complex systems to be analyzed in terms of functional components and their interrelations. In the adaptive production systems, causal attribution is usually implicit. When a new production, $C \rightarrow A$, is constructed by appending an action, A, to the conditions, C (the latter describing the change it produced in some symbolic structure), this conjunction amounts to a tacit assumption that the action was the causal agent for the change.

Hindsight is illustrated by all the programs (e.g., Neves & Anderson, Larkin) that learn by doing or learn from examples. The knowledge of results produced by actually working a problem provides the information necessary for acquiring better methods of solution.

The chapter by Hayes-Roth, Klahr, and Mostow explores some of the possibilities of *learning from instruction*—in this case, learning by taking advice. The advantage of instruction, of course, is that a benign environment (the instructor) can provide precisely the information that will facilitate the acquisition of the next increment of skill. In the adaptive production systems, learning from instruction is implicit in the order in which problems are presented to the system. Although this fact is not explicitly demonstrated in the chapters, any such system will learn rapidly or slowly, depending on that order.

IS LEARNING REALLY INVARIANT?

In an earlier section, we asked whether the learning process is one of the invariant components of the human information-processing system. Nothing we have said about learning demonstrates such invariance, and indeed, there is considerable empirical evidence that learning processes can change over time (Harlow, 1949). These *learning to learn* phenomena challenge our thesis that laws of learning can serve as the central invariant core of psychological theory. To assess how gravely the thesis is damaged, it will be helpful to get a clearer picture of the mechanisms that might underlie learning to learn.

We discuss two possible explanations of learning-to-learn phenomena. As the first possibility already alluded to in our discussions of the knowledge base and of chunking, the accumulation of new concepts in memory may speed up the learning process even when the basic learning mechanisms remain unchanged. As the second possibility, we examine some of the ways in which the learning mechanisms could themselves evolve. In both cases, although we draw, for our examples, upon artificial intelligence programs that are primarily concerned with discovery, we see that these programs have much in common with the theories of learning we have been considering. Discovery—acquiring knowledge by putting questions to Nature—is simply the limiting case of learning—acquiring knowledge from all the sources that are available.

Defining New Concepts

Within the past few years, two artificial intelligence systems have been constructed, AM and BACON, that discover new laws and concepts. The two systems handle quite different classes of problems. AM, given an initial stock of mathematical concepts, strives to invent interesting new concepts and to make conjectures about the relations among concepts. BACON (we are concerned here with the version known as BACON.3), given a body of empirical data, strives to find invariant relations among the variables contained in the data set. In spite of this difference in goals, both systems define new concepts in order to represent information parsimoniously, and both use these newly defined concepts recursively in their subsequent discoveries. Thus, by applying the same discovery heuristics again and again, AM and BACON.3 can generate a potentially endless procession of laws and higher-level concepts.

The AM Program. Three sets of "givens" are provided to AM (Lenat, 1977): a set of criteria for judging how "interesting" a concept or conjecture is, a set of heuristics to guide the search for interesting concepts and conjectures, and a core of basic knowledge about some domain of mathematics. The first two of these three classes of "givens" are part of the structure of AM; the third is particular to each task domain.

A few examples will illustrate their flavor. A concept is interesting if it is closely related to other concepts that have already been found to be interesting. It is interesting if examples that satisfy it can be found, but such examples are not too easily found. A conjecture is interesting if it has strong consequences. The search heuristics suggest generalizing a concept when it is too hard to find examples, but specializing it when it is too easy. Limiting cases of concepts deserve special attention.

There are several hundred such criteria and search heuristics in AM. The system operates by best-first search through the space of concepts and conjectures, using the interest criteria to guide the search and the search heuristics to generate new alternatives.

When AM was provided with the basic ideas of set theory (the concept of set, union and intersection of sets, and so on), it generated, in the course of a couple of hours of computer time, the concepts of the integers, of the operations of addition, subtraction, multiplication, and division on the integers, of factors of an integer, of prime numbers, and many others. It conjectured that any integer can be represented uniquely as a product of powers of primes (the so-called Fundamental Theorem of Arithmetic), and that any even integer can be represented as a sum of two primes (Goldbach's conjecture).

None of this was new to mathematics, but all of it was new to AM—the program had no access to the relevant mathematical literature. Nor was it given any indication of its targets—of what concepts to search for. It operated entirely on the basis of a forward search, guided in the manner that has already been described.

What is of particular interest for our discussion here is that each step AM takes—each new concept or conjecture it reaches—makes still other concepts and conjectures attainable that were not so before. In this sense, AM can be said to "learn to learn." A little later, we will mention a proposed extension to AM that would reduce still further its invariant core in relation to those of its components that are subject to modification.

The BACON.3 Program. The way in which BACON.3 (Langley, 1979) extracts laws from data can be illustrated with an actual example of its work: the (re)discovery of Kepler's Third Law of planetary motion. The law states that the period of revolution of a planet about the sun varies as the 3/2 power of its distance from the sun. In this case, BACON.3 is provided with two independent variables—the *orbiting body* and the *central body*, and two dependent terms—the mean *distance d* between the two objects and the *period p* of revolution. At the outset, the program gathers data about the planets' orbits about the sun: the *central body* is the *sun*, whereas the *orbiting body* takes on values like *Mercury*, *Venus*, *Earth*, and so forth.

BACON.3 examines its data to see whether either of the variables, p or d, varies monotonically (directly or inversely) with the other. In the present case, d increases as p increases. Cued by this regularity, BACON.3 computes d/p to see if it is constant. It is not but does vary monotonically with d, motivating BACON.3 to consider the new ratio d^2/p and finally d^3/p^2.

Because this last term has a constant value whenever the *central body* is the *sun*, BACON.3 formulates an hypothesis to this effect. At this point, the program makes another series of observations in which the *central body* is *Jupiter*, and the orbiting bodies are that planet's satellites. However, instead of retracing its earlier path, BACON.3 simply recalculates the values of the terms d/p, d^2/p, and d^3/p^2. The last of these is again constant, though with a different constant than in the earlier situation.

Thus, the system arrives at an equivalent law for Jupiter much more rapidly than it did for the sun. It can do this because it applies its discovery heuristics to higher-level terms it had defined earlier. Although BACON.3's learning rules did not change, they *appeared* to become more efficient because they already had a useful representation to operate on. Perhaps all learning-to-learn phenomena can be explained through similar recursive mechanisms.

Learning to Learn

An alternative to the previous approach assumes that the learning process actually changes over time. Then the learning-to-learn mechanisms would replace the learning processes as the ultimate cognitive invariant. However, this scheme introduces another layer to the basic architecture, a complexity one would like to avoid. And inevitably, someone would soon suggest that even the learning-to-learn process is not *really* invariant and would propose yet another layer.

Fortunately, there is a third approach that bypasses this infinite regress of learning processes. Assume that the system starts with a few weak but general learning mechanisms, but with an important difference. These heuristics can be used to learn in the standard sense, but they can also be applied to one another. Thus, a heuristic for creating discriminant rules might act upon a generalization heuristic to create a more powerful domain-specific generalization heuristic. In this scheme, the learning heuristics are *identical with* the learning-to-learn heuristics, because they can act upon each other and upon themselves.

This perspective suggests an explanation for the common distinction between conscious and automatic learning. A number of chapters in this volume have explained automatization through a composition process in which condition-action rules are combined. If one assumes that all learning is initially slow and deliberate but that it is possible to learn to learn, then one can imagine the composition process leading to faster and more automatic versions of these learning techniques. Thus, automatic learning occurs when the learning heuristics in use have been sufficiently practiced, whereas conscious learning occurs when less familiar learning rules are used.

Lenat and Harris (1977) have proposed an extension to the AM system that would let it modify its own learning heuristics. AM represented its learning heuristics in one form and the concepts resulting from those heuristics in a different form. The new system would represent both concepts and heuristics in the same format. This would allow heuristics to apply to each other (and to themselves) as easily as to nonheuristic concepts. Such an approach should allow learning to learn to occur without postulating a separate set of mechanisms for that purpose.

A RESEARCH STRATEGY

We have proposed two alternate explanations of learning to learn phenomena, but we have made no claims as to which (if either) is correct. Whatever may be the mechanism for learning to learn, *assuming learning is invariant* is a useful research strategy for the immediate future. If learning is invariant and learning to learn is an artifact resulting from concept formation, then this strategy needs no justification. But even if learning to learn is a reality and the learning mechanisms evolve, they must do so at a much slower rate than do performance procedures. Because the learning-to-learn mechanism must take information about the learning of standard procedures as its data, normal learning must be repeated a large number of times before the learning process can modify itself by observing the outcomes.

Hence, we can expect learning to be invariant in the short run, and laws of learning to be close approximations to the cognitive invariants we are seeking. Because physics has taught us many times that approximate laws may be very useful, and because the laws that appear to be best confirmed are often only approximate, it should cause no great concern to find ourselves in the same plight. Learning theory does indeed have a central role to play in formulating parsimonious, nearly invariant laws of cognition.

ACKNOWLEDGMENTS

This research was supported by Research Grant MH-07722 from the National Institute of Mental Health and Grants SPI-7914852 and IST-7918266 from the National Science Foundation.

REFERENCES

Anzai, Y., & Simon, H. A. The theory of learning by doing. *Psychological Review*. 1979, *86*(2), 124-140.

Eavarone, D. S., & Ernst, G. W. A program that discovers good difference orderings and tables of connections for GPS. *Proceedings of the 1970 IEEE Systems Science and Cybernetics Conference*. N.Y.: IEER, 1970, 226-33.

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

Feigenbaum, E. A., Buchanan, B. G., & Lederberg, J. On generality and problem solving: A case study using the DENDRAL program. *Machine Intelligence 6*. Edinburgh: Edinburgh University Press, 1971.

Harlow, H. F. The formation of learning sets. *Psychological Review*. 1949, *56*, 51-65.

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

Langley, P. Rediscovering physics with BACON.3. *Proceedings of the Sixth International Joint Conference on Artificial Intelligence*. 1979, 505-507.

Lenat, D. B. Automated theory formation in mathematics. *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*. 1977, 833-842.

Lenat, D. B., & Harris, G. Designing a rule system that searches for scientific discoveries. In *Proceedings of the Workshop on Pattern-Directed Inference Systems, SIGART Newsletter*, 1977, 63.

Mill, J. S. A system of logic. Reprinted in E. Nagel (Ed.), *John Stuart Mill's philosophy of scientific method*. New York: Hafner Publishing Co., 1950.

Neves, D. M. A computer program that learns algebraic procedures by examining examples and working problems in a textbook. *Proceedings of the Second National Conference of the Canadian Society for Computational Studies of Intelligence*. 1978, 191-195.

Newell, A. Learning, generality, and problem solving. *Proceedings of the IFIP Congress, 62*. Amsterdam: North-Holland Publishing Company, 1963, 407-12.

Newell, A., Shaw, J. C., & Simon, H. A. Report on a general problem-solving program for a computer. *Information Processing: Proceedings of the International Conference on Information Processing*. Paris: UNESCO, 1960, 256-64. a.

Newell, A., Shaw, J. C., & Simon, H. A. A variety of intelligent learning in a general problem solver. In M. C. Yovits & S. Cameron (Eds.), Self-organizing systems: Proceedings of an interdisciplinary conference. New York: Pergamon Press, 1960. b.

Newell, A., & Simon, H. A. The logic theory machine: A complex information processing system. *IRE Transactions on Information Theory*, 1956, *IT-2*(3), 61-79.

Pople, H. The formation of composite hypotheses in diagnostic problem solving. *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*. 1977, 1030-37.

Rosenblatt, F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 1958, *65*, 386-407.