Problem Solving in Schools and Beyond: Transitions from the Naive to the Neophyte to the Master

Joseph M. Scandura
University of Pennsylvania

The naive learner's knowledge may be characterized in terms of rules and higher-order rules that are not adequate individually for solving desired problems, but which by proper combination can be made so. The neophyte's knowledge is likened to knowing explicit rules for solving particular classes of problems. The master's, by way of contrast, corresponds to knowing rules consisting of more efficient procedures which operate on relatively complex structures. Processes of transition from naive to neophyte and from neophyte to master are shown to be explicable in terms of interactions among appropriate higher and lower-order rules governed by a single universal control mechanism. Incidental comparisons are made to other cognitive theories.

Why is it that some people can solve problems that others cannot? And, how is it that initially naive learners acquire new competencies — and gradually come to acquire the sort of mastery we normally associate with the expert?

To see what is involved, let us first consider the game of chess. Specifically, consider the problem first from the standpoint of the naive learner, then from the standpoint of the neophyte — one who has already learned the rules of the game and can play passably by analyzing one's move very carefully and acting accordingly. Finally, we consider the chess master.

Depending on one's age, education, and experience, the naive learner enters into the experience with varying degrees of generalized knowledge which allow one to understand spoken or written directions; generalized strategies and techniques that might carry over from analogous games like checkers that the learner already knows how to play, including even some informal awareness of chess itself. On facing the task of learning to play chess, the learner must rely exclusively on the general learning capabilities, strategies, and perhaps the general knowledge of board games already learned. The newcomer must first learn the permissible moves for each of the pieces, how to anticipate an opponent's moves, and some generalized chess-specific strategies for beginning games and ultimately for producing checkmates.

The neophyte, by way of contrast, has acquired most of these skills, and with effort can play a passable game by carefully and systematically analyzing moves. One characteristic of play at this stage is the relatively long period of time between moves required to "check into all possibilities," each of which may be analyzed to a depth of two or three moves in advance.

To be sure, the chess master may also take considerable time before deciding on a next move, especially when playing with other chess masters. Even here, however, the chess master seems able to discard most moves almost automatically, concentrating on and exploring in detail only those few possibilities that make sense (see de Groot, 1966). This automatic, nearly instantaneous assessment of chess positions and possibilities becomes readily apparent in games the master plays with the neophyte. While the serious neophyte might spend minutes, even tens of minutes, analyzing a given move, the chess master might be seen to react almost instantaneously, often producing a checkmate within the first few moves.

What is involved in these situations? What occurs in the case of chess, I would propose, is not really unlike what happens in almost any other area where one must learn how to solve problems.
Indeed, a number of theories have been proposed to explain how people solve problems (e.g., Chase & Simon, 1973; Kleinmuntz, 1966; Newell & Simon, 1972; Polya, 1962; Wertheimer, 1945). As is the case with all theories, they differ in comprehensiveness, cohesiveness, precision, parsimony, and in the extent to which they are operational. Structural learning theories (i.e., cognitive theories which share certain structural and methodological characteristics — see Scandura, 1971a, 1973, 1978, 1980) seem best to fulfill these various requirements. In structural learning theories it is assumed that what a person does in a given problem-solving situation depends on two things: (a) knowledge the person has and (b) certain general capacities which apply universally in all problem solving. These are elaborated below.

Two kinds of knowledge fall in the category of prior or available knowledge: (a) knowledge of procedures or algorithms and (b) knowledge of structures or propositions. Procedures or algorithms might be termed prescriptive knowledge in that they provide step-by-step prescriptions for telling the student exactly what to do and when to do it. The subtraction algorithm of ordinary arithmetic is one such procedure or algorithm. Do not be misled into thinking, however, that algorithmic knowledge is restricted to such things as subtracting numbers. Algorithms exist for solving any number of other kinds of tasks as well. Indeed, as we shall see, algorithms play an important role in all problem solving. Though space prevents delving into this research here, we have developed algorithms for proving complex theorems in mathematics (Scandura, 1977); for constructing complex geometric figures using straight edge and compass (Scandura, Durnin & Wulfeck, 1974); for use by young children in solving Piagetian tasks (Scandura & Scandura, 1980); and so forth (e.g., see Scandura 1977, 1978, 1980).

Unlike procedures, structures represent the static aspects of knowledge. Structures are more holistic. For example, the knowledge that the chess master might bring to bear to rapidly assess particular board configurations is more naturally represented in terms of structures.

The basic theoretical construct in structural learning theories is that of the rule (e.g., Scandura, 1970). Rules may be characterized as triples, consisting of a domain, or class of structures to which the rule applies; a range, or set of anticipated results of applying the rule; and a restricted type of procedure or algorithm which operates on the structures in the domain (and generates structures in the range).

In effect, the rule is a theoretical construct that includes both procedural and structural knowledge. The domain of a rule constitutes a class of structures or a structural type and is analogous to what other theorists have called a "frame" (e.g., Minsky, 1975). As we shall see later, the rule is a very flexible construct and allows one to characterize the knowledge available both to the neophyte and to the master.

My own research suggests that problem-solving behavior can be explained in a surprisingly simply way: Thus, in addition to learner-specific knowledge, it appears sufficient to assume only two problem-solving universals.

The first of these universals involves a control mechanism which determines how learners use the knowledge they have available, and the second deals with the capacity of the human processor or the amount of information the human can keep in mind at any one time.

The control mechanism, in concert with specific problems, provides a means of determining which of a person's available rules to use and when to use them. Roughly speaking, the control mechanism operates in the following manner: When confronted with a problem, the problem solver first checks each rule in his processor to see if the given of the problem is in the domain of the rule and that the range of the rule contains the problem goal. If exactly one rule satisfies these conditions, it is applied. If not, the problem solver checks the range of each rule to see if it contains a rule /' such that the problem given is in the domain of /' and that the range of /' contains the problem goal. If exactly one higher-order rule satisfies these conditions, it is applied — thereby generating a solution rule that does satisfy the original problem conditions.

In general, when no solution rules are immediately applicable, or when there is more than one, the search moves to the next higher level to check for higher-order rules which have the potential to generate a rule that applies. Once a rule has been selected and applied, the output of such application (possibly a newly generated rule) is added to the set of available rules. Control then reverts to the next lower level where the search continues, this time to a rule set that contains the new rule.
Although it would take us too far afield to go into details here, the restrictions placed on the procedures of rules correspond precisely to the hypothesized control mechanism. Specifically, rule procedures are not allowed to generate new subprocedures and later "call" (use) them. Where needed to explain (or predict) behavior, the control mechanism takes over this "calling" function.¹

Since both rules and the control mechanism are theoretical constructs one might wonder why separate them (especially since the distinction is arbitrary from a formal/mathematical point of view). The main reason one must distinguish them in instructionally relevant theories is because some constructs (i.e., lower and higher-order rules) may have to be learned whereas others never do. Our research shows that the hypothesized control mechanism is uniformly available to all school-age learners.

In the first (Scandura 1971a, 1974) of a long series of experiments (e.g., Scandura 1973, 1977) 24 young children were trained on simple rules of the form A → B and composite rules of the form A → B → C (where A, B, C, etc. were objects to be traded like paper clips, pencils, erasers, etc.). Half of the children also were trained on a higher-order rule, which if applied to pairs of rules of the form (A → B, B → C), generated corresponding composite rules of the form A → B → C.

Then, all of the children were trained on new A → B', B' → C' rules which they had never seen before. The children who had been trained on the higher-order, as well as the new rules, ALL succeeded on the crucial A → C test problem. Those who were not trained on the higher-order rule uniformly failed.

These results and many others like them indicate two things:

1. Even young children come "wired in" with something equivalent to the postulated goal-switching control mechanism. If, for example, the A → B', B' → C and higher-order rules had been programmed into a computer and then it was presented with the problem (A' → C') of trading C' objects for A' objects — nothing would happen!! The computer would not know what to do with the information unless it had been preequipped with an adequate control mechanism. By way of contrast, having failed to find a solution rule available, the higher-order rules children apparently searched for and found an available higher-order rule appropriate to the situation. They applied it and generated a needed A → B' → C' rule. Control then returned to the original level and the search yielded this newly derived rule. Subsequent application solved the A → C problem.

2. The fact that the non-higher-order rules children failed indicates that people are not necessarily able to compose (put together) arbitrary pairs of rules of the form A → B, B → C. Consequently, assuming a control mechanism that allows for arbitrary composition of rules, as is common in many cognitive theories, is too strong an assumption. Such a mechanism would not be universally available to all people and, hence, would be of limited value in explaining human cognition.

Processing capacity is the second hypothesized cognitive universal (Scandura, 1971a, 1973; Voorhies & Scandura, 1977). In structural learning as in most cognitive theories learners are assumed able to process a fixed number of entities (or "chunks") at any one time. In contrast to most cognitive theories (e.g., Newell & Simon, 1972), however, working memory in structural learning theories is assumed to hold not only data, the stuff on which rules operate, but rules (processes) themselves (e.g., Scandura, 1971a, 1973). In more recent years, other theorists appear to be moving in a similar direction (e.g., Anderson, 1976; Case, Note 1).

This difference has a number of important implications pertaining to a variety of memory phenomena, only some of which have been investigated empirically. Unfortunately, most of these implications are rather subtle and going into them here would detract from our main purposes. The interested reader is referred to Scandura (1973, Chapter 10) and Scandura and Brainerd (1978, pp. 155-166) for further discussion.

To see what is meant by processing capacity, imagine the following task. If I were to read you a list of numbers and ask you to repeat them back to me, the list 2, 4, 5 would present no problems. Yet on a first attempt, the list 7, 3, 9, 1, 5, 7, 4, 6, 8, 9, 5, 7, 9, 5 would probably be impossible to repeat. What is the difference? Clearly, one can only keep so many things in mind at one time.

Suppose, however, that you were to practice these numbers over and over. You could certainly memorize the digits so that they could

¹ The control mechanism effectively eliminates the need for "recursion" in procedures of rules while retaining the power and efficiency recursion provides.
be repeated on cue. Once this has been accomplished, it is no longer necessary to remember all of the individual digits as digits. In order to think about a previously memorized list, you would need only remember what list it was, perhaps by assigning it some label to distinguish it from other lists. And, anytime you want a particular digit in that list, you for first call on that list as a unit, as an integrated structure or what Miller (1956) called a ‘‘chunk’’.2

The above discussion has important implications for problem solving. In particular, having already learned a rule (so that it is available) does not necessarily guarantee that the rule may be carried out effectively or efficiently. For example, applying the rule may impose a heavy load on the user’s processor, thereby requiring the student to resort to pencil and paper or other memory aids. Hence, once a new rule has been learned, the emphasis changes from acquisition to level of skill. In structural learning theories this means acquiring more efficient rules.

The question is, ‘‘How does the learner learn these more and more efficient rules?’’ Oddly enough, in structural learning theories this process of automatization takes place exactly as in all other learning. In particular, the learner uses higher order rules to eliminate unnecessary or redundant steps, in the process generating more comprehensive rule domains (containing “chunks” or structures capable of holding increasing amounts of data), and/or learns new, more efficient rules for dealing with particular problem subtypes (e.g., Scandura, 1973, p. 104).

For example, in the case of ordinary addition, most people eventually learn short cuts, such as those for adding numbers that end in zero or five. Over time, we also memorize special facts, such as 75 + 50 = 125. Higher-order automatization rules have the effect of incorporating these special facts into new, more efficient shortcut rules for performing complex additions.

In these cases what tends to happen is that the procedures associated with rules tend to become simpler and/or more efficient. The structures on which the rules operate correspondingly become more comprehensive. For example, in chess, sequences of play, each step of which is viewed distinctly by the neophyte, gradually come for the chess master to be viewed as holistic structures.

To summarize, the naive learner’s knowledge consists of rules and higher-order rules that are not in themselves adequate for solving desired problems, but which by proper combination can be made so. The neophyte’s knowledge can be likened to knowing explicit rules for solving particular problems. The master’s, by way of contrast, corresponds to knowing more efficient rules — which operate on more complex structures. Acquiring these more efficient rules, of course, requires prior mastery of the requisite neophyte rules and needed higher-order automatization rules.

Clearly, the examples I have described are a long way from what is involved in chess. But exactly these same principles apply there as well.

In order to avoid any misunderstandings, let me emphasize that what I have done so far is to merely sketch some of the major ideas involved in understanding and explaining how people solve problems. It is intended primarily as a basis for the second part of this article. The educator needs to know how one might facilitate the learner’s mastery of problem-solving skills; specifically, how one can facilitate transfer from the naive student to the neophyte to the master.

A Prototypic Instructional Strategy Based On A Structural Learning Theory

Instructional implications of structural learning theories are perhaps best seen by example. Let us consider column subtraction problems as a simple prototype.

In order to utilize structural learning principles in designing instruction, the essential first step is to identify: (a) the educational goals — what the learner is to be able to do after instruction and (b) prototypic cognitive processes or rules — what the learner must learn if he is to successfully perform tasks associated with the educational goals.

In the case of subtraction, for example, let us assume, given column subtraction problems, that our educational goal is to find the differences. By a prototypic rule, or cognitive process in this case, I refer essentially to what the learner must master in order to subtract numbers.

2Chase and Simon (1973) have developed a technique for isolating and studying such structures with more complex content, like chess.
"The Structural Learning Theory" provides a general method of analysis, called Structural Analysis, by which the rules to be learned can be derived from suitably operationalized educational goals. While there are many details still to be completely objectified, the method is relatively systematic and has been applied successfully in analyzing a wide variety of content.

The first step in structural analysis involves selecting a representative sample of problems associated with the goal in question. Representative problems may be defined as problems that the "teacher" feels best represents the goal — in the sense that being able to solve them would almost certainly imply being able to solve the others. In the case of simple subtraction, this might include problems like:

\[
\begin{array}{ccc}
879 & 432 & 402 \\
-325 & -129 & -129 \\
??? & ??? & ??? \\
\end{array}
\]

The second step in structural analysis involves identifying rules which make it possible to solve each of the selected problems. Identifying such rules involves several identifiable substeps:

1. Assumptions must be made regarding the minimal encoding and decoding capabilities of the students in the target population. In the case of second graders, for example, the teacher/analyst would normally assume that all students are able to distinguish "the minus sign", the individual digits 0, 1, ..., 9, the columns, the rows, and that all are able to write the individual digits in desired locations. The present illustration builds on this assumption. Consequently, the remainder of the analysis will be inadequate just to the extent that these assumptions are in error for students in any given target population.

2. The analyst must decide the scope of each of the representative problems. This scope effectively defines the domain of the rule associated with the prototype. The problem

\[
\begin{array}{c}
432 \\
-129 \\
??? \\
\end{array}
\]

for example, might be held prototypic of the entire class of column subtraction problems, namely those formed by varying the individual digits 0, 1, ..., 9 and/or the number of columns. Indeed, in the present case, each of the selected representative problems is prototypic of this same domain. Consequently, in the present case, it is reasonable to assume that there is only one domain, the domain of column subtraction problems.

3. Next, the analyst must identify the steps (operations and decisions) involved in solving each of the representative problems. These operations and decisions must be sufficiently simple that using them refers only to abilities that are assumed available to all students in the target population (i.e., encoding/decoding capabilities). The operations and decision conditions also must be atomic in the sense that, for each student in the target population, the ability to correctly use one once is indicative of uniform success, and conversely for failure.

The flow diagram in Figure 1 depicts the procedural portion of a rule based on equal additions. In this rule, it is implicitly assumed that each operation acts only on digits, rows and columns — consequently, the previously referred to need to assume certain minimal encoding/decoding abilities. The operations and decision conditions of this procedure (e.g., top number greater than bottom number) constitute additional assumptions concerning atomicity. According to structural learning principles, only to the extent that these assumptions are met will this subtraction rule provide a useful and operationally precise basis for designing efficient and effective instructional strategies.

More could be said about the actual processes by which such rules are constructed but intensive work in this direction is currently underway and going more deeply into it here would detract from our main concerns. The essential thing to emphasize is that the use of structural learning theories for purposes of designing instruction necessarily begins with a rule-based analysis of the subject matter (broadly conceived) in question. In this sense structural analysis is quite analogous to traditional task analysis (Gagné, 1962). Notice, however, that nothing has been said about a taxonomy of subject matter content. All content from a structural learning perspective must be analyzed. Some types, of course, such as our subtraction example, are easier to analyze than others.

Once an analysis has been completed, designing an effective instructional strategy follows directly and precisely from the theory.
Specifically, once an analysis has been completed, one knows (a) what the student is to be able to do once he has achieved the educational objective (e.g., solve arbitrary column subtraction problems) and (b) what the student must learn in order to be able to do that (i.e., the equal additions rule).

Given this information, the first thing one must do in designing an effective instructional strategy is to determine what each student already knows, specifically, that part of what the student knows which is directly relevant to what one wants the student to learn. The process by which this is accomplished has been detailed in the literature (e.g., Scandura, 1971a, 1973, 1977; Durnin & Scandura, 1973) and I will not consider it here. It is sufficient for present purposes to observe that solving particular subtraction problems involves following one and only path through the subtraction rule. In effect, there is a unique class of problems associated with each path through the rule. (Note: There are a finite number of paths associated with any given rule.)

A basic principle in structural learning theories is that rules must be represented in terms of operations and decisions that are atomic; they are either totally available or unavailable to any given learner in the target population (i.e., the population the analyst had in mind in performing the structural analysis). The existence of such a representation can always be guaranteed (e.g., Scandura, 1970, 1973; Suppes, 1976).

In effect, success or failure on any one problem associated with a class of problems

\[\begin{align*}
\text{START} \\
\rightarrow \quad \text{Go to rightmost column} \\
\quad \rightarrow \quad \text{Is top no.} \geq \text{bottom no.?} \\
\quad \rightarrow \quad \text{YES} \\
\quad \rightarrow \quad \text{Subtract bottom no. from top no. using facts for top no.} \leq 9 \\
\quad \rightarrow \quad \text{YES} \\
\quad \rightarrow \quad \text{Is there only 1 column to left with 1 at top?} \\
\quad \rightarrow \quad \text{NO} \\
\quad \rightarrow \quad \text{NO} \\
\rightarrow \quad \text{Add 10 to top and subtract, go to next column and add 1 to bottom} \\
\rightarrow \quad \text{STOP} \\
\rightarrow \quad \text{Any more columns?} \\
\quad \rightarrow \quad \text{NO} \\
\end{align*}\]

*Figure 1. Equal additions algorithm for subtraction.*
provides complete information as to the availability to the student of the corresponding path. For example, the problem

\[
\begin{array}{c}
879 \\
-325 \\
\hline
??? \\
\end{array}
\]

is solved by following the path in Figure 1 defined by operation one, then two, then three, and back to two and three, then two and three again, before stopping.

By testing on a small, finite set of problems, it is possible to identify precisely and unambiguously which parts of the subtraction rule any given individual knows and which parts the student does not know. Such testing, in effect, defines the student's entering level. In this regard, more can be said about such things as testing in situations where more than one rule is involved and about increasing efficiency via sequential testing (e.g., Scandura, 1971, 1973, 1977) but this is not necessary for present purposes.

Prescribing instruction, then, follows directly from what the student knows. All one needs to do is to identify the missing portions of the desired subtraction rule and to present them to the student. The theory is neutral on whether this information should be presented, say, in an expository or a discovery manner. Thus, for example, deciding on the appropriate method of presentation depends on secondary objectives that the teacher may (or may not) have in mind (e.g., to help students learn how to detect regularities). The important part insofar as being able to perform subtraction is concerned is simply to be able to perform according to the rule.

As an illustration, suppose a student's knowledge may be represented by the flow diagram shown earlier, minus only the loop involving operation five (add 10 to the top and subtract, go to the next column and add 1 to the bottom). In this case, the instructor would need only to make sure that the student knows, at the appropriate points, how to add 10 to the top number in the column and how to add 1 to the bottom of the next column. Where the student knows less, of course, one would start with the simpler prototypes (partial rules representing what the student knows) and gradually "elaborate", or add increasing detail until the student has mastered the entire rule.

To summarize, I must emphasize that this illustration of prescriptive aspects of the structural learning theories constitutes only a simple prototype. It "epitomizes" the instructional aspects of the theory. The theoretical system itself provides a far more generalized basis for instructional prescription — which in principle, may be used with any subject matter (or educational goal) that might be of interest.

What I have described so far corresponds to aiding the learner in his passage from the naive to the neophyte. Unlike the naive learner, who enters the problem solving situation with only prerequisite skills and knowledge, the neophyte knows, at least in a textbook sense, how to deal with the problems at hand. In the present case, one who has learned the subtraction algorithm would know how to solve subtraction problems perfectly well. Indeed, in principle, given any subtraction problem, the neophyte could generate the appropriate difference.

One might wonder, therefore, whether learning stops here. A little further thought, however, will convince us that there are indeed differences among people who know how to subtract numbers. Thus, for example, whereas the neophyte goes through every step of the algorithm systematically and in detail, others introduce shortcuts of varying degrees of sophistication. It is unlikely, for example, to find an experienced financial analyst performing preliminary calculations in longhand. Indeed, in this regard, I am not simply referring to the availability of calculators, but to the skilled analyst's quick, insightful estimations that allow him or her to evaluate far more quickly than the neophyte the profitability of particular courses of action.

I would propose that as in chess, and other areas of complex problem-solving, these holistic skills are acquired gradually as a result of introducing shortcuts into the problem-solving process. Thus, rather than applying the subtraction algorithm as such to the problem of subtracting 350 from 500, most adults would simply 'know' that the answer is 150. With additional skill and practice, many people learn to consider such problems as wholes. In addition to developing effective shortcuts for solving the problems, they may even learn the exact differences attached to particular pairs of numbers (such as 500 and 350). Certainly, the problems I have described in the case of arithmetic are not of the same order of complexity as those in chess. I would propose, nonetheless, that they are of the same genre.
Before showing more precisely how one might pass from the neophyte stage to that of the master, let me make several points.

1. It is a mathematical fact that anything that can be represented as a procedure can also be represented as a holistic structure. Consider, for example, the case of the young child learning to identify triangles as a particular type of geometric figure. The child may first learn to distinguish triangles by counting the vertices, getting three, and determining whether the lines connecting the vertices are straight. For the neophyte, all of these steps are taken sequentially and systematically in order. It does not take long, of course, for the child to learn to eliminate most of these steps, henceforth immediately seeing triangles as wholes.

2. It is the procedure in the rule that takes the time. Following a procedure, cognitively or otherwise, involves taking a series of steps, each of which requires some finite time to carry out. The domain of a rule, by way of contrast, is tested holistically, all at once. Thus, for example, whereas the neophyte can tell almost immediately which problems are subtraction problems and which are not, it takes him some finite time to actually generate a solution to particular subtraction problems.

3. The reason the master typically takes less time in solving a problem is that all other things being equal, the rules of knowledge he uses involve relatively simple procedures. Much of the master's knowledge is absorbed into relatively complex domains (of rules) which tell him when to apply this or that procedure. When a class of tasks has been "fully" mastered, the procedural part of the rule frequently amounts to little more than relating a given type of situation to the appropriate response — for example, where the skilled physician assesses a complex set of symptoms and immediately proposes a diagnosis, or when the chess master quickly decides on his next play.

Figure 2. One column addition algorithm.
Figure 3. Two column addition algorithm.

4. The time-tested way of passing from neophyte to master is that of practice. Clearly, by gaining experience and going over greater and greater varieties of problem solving situations, the neophyte gradually learns more and more efficient ways of dealing with those situations.

In structural learning theories, as I mentioned above, mastery is achieved in the same way as all other learning — by the application of higher order rules which generate new, more efficient rules.

To illustrate how learners achieve mastery, consider Figures 2 and 3. The procedure of figure 2 provides for addition one column at a time. For example, the sum of

\[
\begin{align*}
4322 \\
+ 2540 \\
\hline
???
\end{align*}
\]

would be determined by the simple path consisting of going to the first column (operation 1), adding the digits 2 and 0 (operation 2), writing 2 below the line (operation 3), continuing on to the tens column (operation 4) and repeating — adding 2 and 4 (operation 2), and so on.

By way of contrast, the procedure of Figure 3 provides for adding two columns at a time. This is accomplished by operating on more
complex substructures (i.e., pairs of columns) at each step. This time, the first two columns, are added simultaneously, and then the third and fourth.

The two procedures are obviously very similar. While the individual steps are different (e.g., adding pairs of two digit numbers), and overall "flow of control" is identical in each. Indeed, the main difference is that the structures associated with (the rule of) Figure 3 are more complex than those associated with Figure 2.

The relevant question here is how a person knowing the rule of Figure 2 might come to know the more efficient rule of Figure 3. Or, put differently, where do higher-order rules come in? In this case, in operating on the rule of Figure 2, the higher order rule substitutes new operations and decisions while maintaining the overall flow among corresponding components. For example, the basic addition facts of Figure 2 are replaced in Figure 3 by facts to 99 + 99. Correspondingly, the structures on which these "facts" operate are changed from pairs of one-digit numbers to pairs of two digit numbers. In effect, the number of algorithmic steps necessary to solve any given addition problem is reduced precisely because the eliminated steps have been absorbed into the more complex structures on which the procedure operates.

In presenting this example, I make no claims that master arithmeticians necessarily add by pairs of columns (although some undoubtedly use variants), nor do they learn such rules via any one particular higher-order rule. My point is simply that they do learn some such rule and that they learn it by means of some higher-order rule.

Moreover, once learned, a higher-order substitution rule of the sort sketched could be used to learn more efficient variants of other computational rules. It takes little imagination, for example, to conceive of an equal additions (or more familiar borrowing) rule in subtraction that operates simultaneously on pairs of columns.

Suppose we were to utilize such algorithms and analyse more directly in our instruction. Initially, these algorithms might provide a systematic basis for passage from the naive to the neophyte stage.

But even after the underlying rules have been mastered, could not related analyses provide more precise prescriptions as to the kinds of hints and specific information (e.g., replace one column with two) that the neophyte might utilize in his passage from process to structure? In our own research, we have found this to be not only possible, but highly feasible. In-study problem solving (e.g., Scandura, 1977), and more recently the growth of logical thinking during childhood (Scandura & Scandura, 1980) for example, we have been able to systematically facilitate children's passage from one stage to the next — that is, from naive to neophyte, and from there to the master.

In conclusion, I would like to call your attention to certain theoretical points which have been detailed in the literature (e.g., Scandura, 1977, 1980) elsewhere but which are only implicit in the above discussion. Notice first that individual knowledge is measured in relativistic terms. What the learner knows about subtraction is measured relative to what it is we want him eventually to learn, namely, the entire subtraction algorithm. (In the more naturalistic setting, as opposed to one involving education, what must be learned is replaced by a prototypic characterization of what the idealized learner knows. For example, one might characterize, as we (Scandura & Scandura, 1980) have done in our analysis of Piagetian conservation, the idealized knowledge available to children performing at various developmental levels.)

Second, measuring knowledge in relativistic terms makes it possible not only to operationalize individual knowledge, but to do so in a generalizable way. In particular, this approach avoids the dilemma faced by the normative experimental methods of traditional cognitive psychology (e.g., Scandura, 1979), on the one hand, and individualistic computer simulation methods (e.g., Newell & Simon, 1972), on the other. Thus, for example, normative data obtained in the traditional experimental paradigm refers to average behavior and may have little to do with how individuals perform on particular problems. Conversely, whereas simulation methods do deal with individual behavior, they do so in a nongeneralizable way. A different theory may be needed to model the behavior of each individual (e.g., see Scandura, 1977, 1980).

Third, I would like to reemphasize the paradigm we introduced some years ago for studying complex human behavior (e.g., Scandura, 1964, 1970, 1971a). The first step in this research methodology involves structural analysis. Once the domain of interest has been established, the next step is to identify
prototypic or idealized ways in which target populations of students may be expected to deal with that domain. Only after this step has been completed can one progress to the next step, that is, the assessment of individual knowledge (e.g., Scandura, 1971a, 1977). This assessment, then, plus the universals of cognitive behavior, provide the basis upon which we may predict and/or control (and not just explain) student learning.

In effect, in structural learning theories, content, individual differences, and cognitive universals are all treated within a single, unitary system. I have argued at length in this regard that not just any cognitive theory will serve instructional needs (e.g., Scandura, 1980). Close interrelationships exist among the constructs and assumptions used to characterize what is to be learned, the learner's cognitive processes, and the measurement of individual knowledge. These interrelationships place severe constraints on the form of any viable theory — a form that conforms to the class of content-specific structural learning theories. Indeed, there is strong reason to believe that any cognitive-based and operational instructional theory that deals with individual behavior in a generalizable way will necessarily be a structural learning theory (e.g., Scandura, 1980).

As I have argued elsewhere (e.g., Scandura, 1977, 1980), these same comments apply to methodology as well. Indeed, "the paradigm shift we have heard so much about in cognitive psychology in recent years is at least partly a myth. True, in our theorizing we have begun to ask how and why — and not only what a person will do. Rather than developing methodologies to fit the problem, however, we have all too frequently taken the easy but less profitable route of applying methodologies developed for other purposes" (Scandura, 1980, p. 389). Fortunately for the future, an increasing number of studies and content-specific cognitive theories have adopted, either explicitly or implicitly, structural learning methodologies, and/or belong to the class of structural learning theories.

Reference Note


References


Miller, G.A. The magic number seven, plus or minus two: Some limits in our capacity for processing information. Psychological Review, 1956, 63, 81-97.


