the student, even if the separate concepts are, he will be unable to operate. Approaches to mathematics teaching that are largely content based will attempt to develop it logically and to develop all parts of the *subject* in an ordered fashion — and this is admirable. However, more important than the schema lying within the subject (in Popper's world 3) are those in the mind of the student (world 2) (Popper 1973). We need to check with the students whether they find that the information conveyed fits what they have in their minds. An interesting experiment is to ask a number of people whether "minus times a minus is a plus" fits comfortably into their minds. When the student believes, rightly or wrongly, that the idea does fit, then and only then should you move on. It is the "emotional acceptability" of what we are told or read that is the measure of whether we can advance.

Most teachers check out whether their students understand, and by this they are addressing the cognitive. It is necessary to ask whether they accept — and *that* is affective. Once the strings of symbols are attached comfortably to those patterns we already have in our minds, we are secure.

Finally we should mention one counter-indication to what we have said, and point again to one question discussed earlier but not resolved. In the discussion on (ξ,η) we did assume that the schema of two coordinate axes, and the plotting of points was known and comfortable. Why did the use of unfamiliar symbols induce discomfort? Perhaps it is felt that they must convey something more, something mysterious — else why were such letters used? But the reason is not clear.

As for why q is so "strange" — perhaps someone can help?

References

Buxton L. G. (1980). *Do You Panic About Maths?* Heineman Educational, London.

Popper K. R. (1973). "Indeterminism is not enough." *Encounter,* May 1973.

Skemp R. R. (1980). *Intelligence, Learning, and Action.* Wiley, Chichester.

Mathematical Language and Problem Solving

Gerald A. Goldin

Problem solving in mathematics may require different kinds of language: the verbal or mathematical language in which the problem itself is posed, the notational language of problem representations available to the solver, and planning language for heuristic reasoning and formulation of strategies. This paper explores some relationships among these languages, with examples of ways they can influence problem-solving processes.

I Introduction

Problem solving in mathematics refers to situations in which some items of information are given or available, and one or more goals are described. The problem solver is expected to attain the goal(s) through logical or mathematical procedures. Sometimes the term "problem solving" is restricted to the case in which the solver has no routine algorithm available for this purpose. Mathematics educators have become increasingly interested in studying problem solving and improving its teaching (Polya 1962 and 1965; Harvey & Romberg 1980; Krulik 1980; Lester 1980).

Kilpatrick (1978) proposed to organize the independent variables of problem-solving research into three main categories — subject variables, task variables, and situation variables — for the purpose of understanding how problem-solving outcomes depend on variables in each category. A collaborative study of task variables was conducted by a number of researchers (Goldin & McClintock 1979). In this work the characteristics of problem tasks were considered under the following headings: syntax variables, describing the grammar and syntax of the problem statement; content and context variables, describing the semantics of the problem statement; structure variables, describing mathematical aspects of a problem representation; and heuristic behavior variables, describing heuristic processes associated with or intrinsic to specific problems. Task variables were taken to be independent of the individual problem solver, and defined instead with respect to a population of solvers. They are subject in principle to control by the researcher or the teacher.

Visible Language, XVI 3 (Summer 1982), pp. 221-238.
Author's address: Department of Math Science, Northern Illinois University, DeKalb, IL 60115.
0022-2224/82/007-0221$02.00/0

Let us now distinguish among various kinds of language which can be employed during problem solving: the verbal language of the problem, notational languages, and planning language.

A Verbal Language

The verbal language in which the problem itself is posed may be a natural language such as ordinary English, and may include technical terms from mathematical English. Task syntax variables are descriptive of this language. Barnett (1979) reviewed a large number of studies on syntax variables, organizing them into the following categories: variables describing problem length; variables describing grammatical complexity; formats (verbal or symbolic) of numbers or other mathematical expressions; variables descriptive of the question sentence; and the sequence of information in the problem statement. Linear regression studies have indicated that variables of length and grammatical complexity, defined in various ways, do affect the difficulty of verbal problems in arithmetic (Loftus 1970; Beardslee & Jerman 1973), but have provided little insight into how this occurs.

The problem statement is often descriptive of a "real-life" situation which can be pictured or visualized. Content and context variables, reviewed by Webb (1979), describe the semantics of the problem statement. The term "content" refers to mathematical meanings, and the term "context" to nonmathematical meanings, insofar as this distinction can be maintained. Sometimes a problem posed in words may be accompanied by a picture or diagram; then we regard this picture as part of the problem content or context.

B Notational Languages

Notational languages available for problem solving, unlike ordinary language, are highly structured formal systems. They may have strict semantical rules for writing well-formed expressions, and a well-defined set of allowed transformations from one expression to another. Examples include the notations for our system of numeration, for arithmetic operations, for fractions, decimals and percents, for algebra, trigonometry and calculus, for set theory and symbolic logic, and diagrams picturing allowed constructions in Euclidean geometry. Evidently a great deal of the teaching of mathematics is devoted to communicating the rules for working within such languages.

Once a problem has been translated into a notational language, purely formal manipulation of symbols according to the rules of procedure is usually sufficient to arrive at a solution. Nevertheless, the symbol-manipulation may continue to be motivated by visualization of the "real-life" situation that the notation now describes.

The concept of a problem state-space has been employed to describe the mathematical structure of problems, as well as to map the behavior paths of subjects (Goldin 1979). As defined by Nilsson (1971), a state-space for a problem is a set of distinguishable problem configurations, called states, together with permitted steps from one state to another, called moves. A particular state is designated as the initial state, and a set of goal states is distinguished by the conditions of the problem. When a problem can be translated into a standard notational language, the mathematical sentence or diagram which is the most direct translation becomes the initial state. A notational language thus provides a standard representational framework in which the state-spaces of many problems can be embedded. Sometimes for non-standard problems, the solver is in effect presented with a new notational language, with simply stated rules of procedure, and the object is to "learn" the language in proceeding from the initial symbol-configuration to the goal. A problem state-space is thus a notational language in miniature.

C Planning Language

Finally we have the language available to the problem solver for heuristic planning or formulation of strategies. This is the language in which the solver establishes subgoals, organizes trial-and-error search, seeks analogous problems, or engages in the many other forms of planning described by Polya. Thus it is a language *about* problem solving as well as a language *for* problem solving. It appears that children and adults engage in heuristic planning to a considerably greater extent than they can describe explicitly. One of the goals of protocol analysis in studying problem-solving behavior is to describe from an information processing standpoint the planning which occurs, based on a transcript of a subject's "thinking aloud" statements. It would be valuable to systematize such language so that it could be used in the teaching of problem solving.

Figure 1 shows the various levels of language available for problem solving. The perspective of this paper is to treat all of the levels of language as "existing" apart from the individual problem solver, defining them in relation to a population of problem solvers sharing a common "mathematical language."

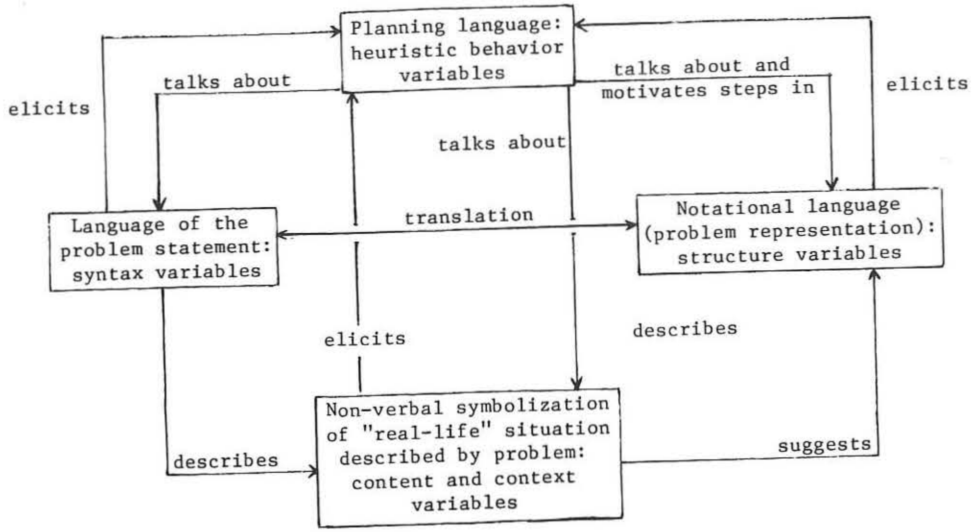


Figure 1. Relationships among levels of language available for problem solving.

II Examples

In this section we describe two examples which illustrate the concepts introduced above. In addition they illustrate the important point that *small* changes in the statement of a problem can result in *large* changes in problem-solving processes, even when the mathematical structure of the problem is held fixed.

A Plants and Flowerpots, Cats and Dogs

The following problems were used with elementary, junior high, and senior high school students (Caldwell & Goldin 1979; Goldin & Caldwell 1979):

1 Alan bought an equal number of plants and flowerpots. Each plant cost three dollars and each flowerpot cost five dollars, so that he spent 48 dollars in all. How many plants did Alan buy?

2 Jane has an equal number of dogs and cats. If she had twice as many dogs and four times as many cats, she would have 42 pets in all. How many dogs does Jane have?

The two problems were originally intended to be parallel, except that the first problem is stated factually and the second has a hypothetical component. It turned out that Problem *1* was less difficult than Problem *2* for every school population studied, but not necessarily for the reason expected.

The languages of the problem statements have values for syntax variables which are quite close. Both problems have three sentences; Problem *1* has 33 words and Problem *2* has 34 (excluding articles). Both problems contain three items of numerical information, with the first two (small) numbers written in words and the third (larger) number written as a numeral. The grammatical complexity of the two problems is comparable, as measured by a "syntactic complexity coefficient" developed by Botel, Dawkins, and Granowsky (1973). The question sentences occur at the end of both problems, and are of exactly parallel length and grammatical construction. The two problems differ in syntax in the factual/hypothetical variable. There are other minor syntactic differences as well; the second problem, for example, uses the pronoun "she" twice, while the first uses the pronoun "he" but once.

The notational language of algebra provides a standard representation for each of these problems (unlikely to be available, of course, to the students in elementary or lower junior high school grades). With the obvious choices of letters for unknowns, Problem *1* translates to: $P = F$, $3P + 5F = 48$; while Problem *2* translates to: $D = C$, $2D + 4C = 42$. These two systems of equations can be solved in exactly the same manner, and in exactly the same number of steps, to yield $P = 6$ (for the first problem) and $D = 7$ (for the second). We therefore say that in this representation, the two problems have the same structure. An alternate notational language, often used by younger children, involves the use of "guess and check" procedures. For example, the child may first make a "guess" as to the number of plants, and compute the total cost. If this is too low, a new "guess" is made. Schematically, we have something like this: "If 1 plant, 1 flowerpot, $3 + 5 = 8$, too low; if 2 plants, 2 flowerpots, $3 \times 2 = 6$, $5 \times 2 = 10$, $6 + 10 = 16$, still too low; . . . . " until the trial "6 plants" occurs. This procedure can also be used to find the number of dogs in the second problem. Some children are able to carry out these procedures aloud, without the use of written notation at all. Whether written or oral, it is convenient to think of the procedure as occurring in a formal language containing, for example, "trial" statements and "comparison" statements acting on a domain of whole numbers (the "search space"). Such procedures have been examined by Harik (1979).

The planning which takes place when these problems are solved is often silent. The algebra student may say, "First I will write down some equations, then I will solve them," and the grade school student may comment, "Let's try some numbers." Along the way, additional

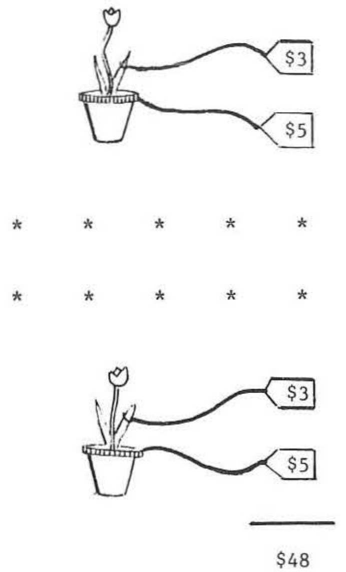


Figure 2a. One way to visualize the plants and flowerpots.

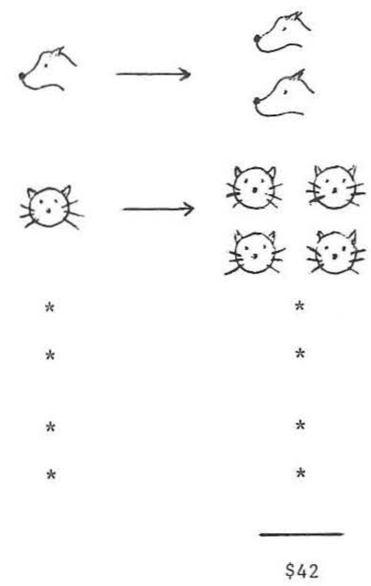


Figure 2b. One way to visualize the cats and dogs.

planning may occur aloud; for example, "Skip some numbers." Most often the observer is left to infer the nature of the planning which occurred, through analysis of the solver's verbal protocol.

Turning to the process of translation from the problem statements to algebraic notation, we note that these problems contain "key words"— words which very frequently translate to particular mathematical operations. For example, the phrases "Each ... cost" and "times as many" translate to multiplication (×), while "in all" translates to addition (+). Since such terms occur nearly in parallel in the two problems, students who translate directly from the problem statement to notational language (as in Figure 1) should arrive at parallel systems of equations.

On the other hand, the real-life situations described by the two problem statements are quite different. Figure 2 depicts one way in which these may be visualized. This difference allows the following method of solution for Problem *1*, which is not available for Problem *2*. In Problem *1* the picture suggests: "Each plant cost $3 and each flowerpot cost $5, so that the pair cost $8. Since Alan spent $48, he bought 48 ÷ 8 = 6 plants." The analogous line of reasoning for Problem *2* is extremely awkward to phrase or to visualize, even though the problems are of corresponding mathematical structure. For this reason, the original intent of creating problems which were parallel except for the factual/hypothetical variable was not entirely achieved. Referring again to Figure 1, the language of the problem statement described a "real-life" situation which in turn suggested a notation (3 + 5 = 8, 48 ÷ 8 = 6) different from that obtained by direct translation, and in this case more efficient.

B A Checkerboard and Paper Clips

This well-known problem provides a second example for discussion:

> *3* Consider an 8 × 8 arrangement of squares, from which diagonally opposite corner squares have been removed (Figure 3). A paper clip may be placed so as to cover two squares adjacent horizontally or vertically, as in the illustration. Can all the squares be covered by paper clips without overlap? If so, how; if not, why not?

The problem statement describes a concrete apparatus which itself can serve as a notation for making moves. Often solvers proceed to experiment by placing paper clips, until after several trials they acknowledge their inability to achieve the goal. During this stage of problem solving, little overt planning may occur. Atwood, Masson, and Polson (1980) discuss a model for problems which are similar to this one in that successor states are generated from an initial state by application of a single rule of procedure. Their basic assumption is that subjects do not plan, but use only information from the current problem state and those which immediately follow to make each move. In a study of "water jug" problems, they found their model to account adequately for subjects' behavior. It may well be the fact that a notation is provided by the problem itself which encourages subjects, at least initially, to restrict themselves to mechanical moves within the notation.

In Problem *3*, however, planning is necessary if the solver is to proceed beyond the observation that the trials do not succeed. More

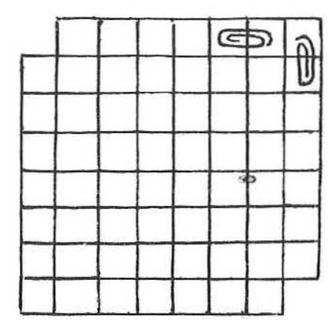


Figure 3.
Diagram for Problem *3*.

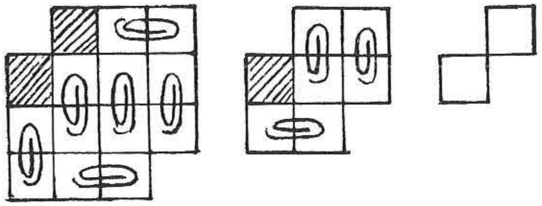


Figure 4.
Trying to solve
a simpler problem.

sophisticated or "educated" problem solvers might even engage in planning from the start. For example, the heuristic advice, "Try to solve a simpler related problem," may lead to examination of the 2 × 2 case (clearly impossible), the 3 × 3 case (impossible since there are an odd number of squares), and the 4 × 4 case, which is quite similar to the given problem but allows much more rapid exploration (Figure 4). Trials on the 4 × 4 case may lead to the observations that diagonally attached squares often remain uncovered after a trial, and that the same squares seem to remain in a variety of trials.

One way to achieve insight into this problem is to *improve* the notation by coloring those squares which remain after various trials. The decision to do this requires the ability to think or talk *about* the language being used to represent problem states; i.e., to think on the level of planning language. The pattern of colored squares which results is that of an ordinary checkerboard. Now it can be observed that a paper clip always covers a colored square and a white square. Since in the initial 8 × 8 problem there were 32 colored squares and only 30 white squares, and they are being reduced in equal numbers, the squares cannot all be covered by paper clips — there will always be two colored squares left over.

Possibly Problem *3* would be less difficult if its statement referred to "an 8 × 8 checkerboard" instead of "an 8 × 8 arrangement of squares," or if one set of squares were shaded in the diagram. The original notation was less effective because essential information was not *visually* apparent in the representation of a state (although it could have been obtained of course by counting). Again a small change in the problem statement, which does not affect the problem structure, suggests a substantial change of notation which in turn facilitates the problem solution.

III Efficient Notational Language and the Structure of Problem Representations

This section first looks at examples of efficient and inefficient notation in standard representational frameworks. Then we examine how, in

non-standard representations, the choice of symbolism can illuminate or conceal important structural features such as problem symmetry, or affect the complexity of each move.

A Standard Languages of Mathematics

Much of the progress of mathematics across history is attributable to the development of improved systems of numeration and modern algebraic notation. Arithmetic problems which would have posed formidable challenges in ancient Greece or Rome can be solved by today's school-children. The process of experimentation and notational change is an ongoing one today in algebra and analysis. From the perspective of problem solving, an effective notation should have certain characteristics, among which are the following: (1) Symbol-configurations should be reasonably concise, with information most likely to be important made visible rather than suppressed. The number of steps needed to move from one configuration of symbols to another should be small. (2) To the extent that concepts are parallel mathematically, they should be represented in parallel syntactically. Two examples will illustrate these points.

When the "new mathematics" was introduced in the 1950's and 1960's, precision of meaning in notation was sometimes emphasized at the expense of problem-solving effectiveness. The "raised minus sign" was introduced to denote negative numbers (additive inverses), and $^{-}3$ was called "negative three," not "minus three." "Minus" was reserved for the operation of subtraction, with "$8 - 6$" defined as "$8 + {}^{-}6$." Operations such as addition, subtraction, multiplication, and division were treated strictly as binary operations (acting on two numbers at a time), and each step had to be justified with reference to the appropriate structural property of the number system (associative property for addition, commutative property for multiplication, etc.). A consequence of rigid adherence to these rules might be the following sequence of steps in algebra: $3X + 7 = 19$ [given], $(3X + 7) + {}^{-}7 = 19 + {}^{-}7$ [addition of the same number to equals yields equals], $(3X + 7) + {}^{-}7 = 19 - 7$ [definition of subtraction], $(3X + 7) + {}^{-}7 = 12$ [renaming], $3X + (7 + {}^{-}7) = 12$ [associative property for addition], $3X + 0 = 12$ [additive inverse], $3X = 12$ [additive identity], $(1/3)(3X) = (1/3)12$ [multiplication of equals by the same number yields equals], $((1/3)3)X = (1/3)12$ [associative property for multiplication], $((1/3)3)X = 4$ [renaming], $1X = 4$ [multiplicative inverse], $X = 4$ [multiplicative identity].

Obviously the purpose of an exercise such as the above is to develop a sophisticated awareness of the use of axioms, and not to facilitate efficient problem solving. The efficient problem solver would write $3X + 7 = 19, 3X = 19 - 7 = 12, X = 12 \div 3 = 4$. Unfortunately many teachers and textbooks stressed the precision of the axiomatic notation at the expense of facility with the usual notation, and basic computational and problem-solving skills suffered. The axiomatic language in this case requires more steps, and is less concise.

An example of notational improvement is taken from the APL computer language (Iverson 1966 and 1969). It is common to write max $\{a,b\}$ to denote the larger of two real numbers a and b, and a^b to represent a taken to the bth power. In APL these and many other operations are assigned special symbols, and treated as binary functions. Thus, $3 \lceil 7$ denotes the larger of 3 and 7, having the value 7; $2 \uparrow 5$ stands for 2 to the 5th power, and has the value 32. Borrowing just these symbols from APL and incorporating them into ordinary arithmetic, we see that their place in the syntax becomes the same as that of $+$, $-$, $\times$, and $\div$. Structural properties for $+$ and $\times$, such as the associative and commutative properties, can now be tested for $\lceil$ and $\uparrow$ ($\lceil$ is commutative and associative, $\uparrow$ is neither). The distributive property for multiplication across addition, which states (left distributive property) that $a \times (b + c) = (a \times b) + (a \times c)$ for all real numbers a, b, and c, can be generalized and tested for various pairs of operations: for example, $a + (b \lceil c) = (a + b) \lceil (a + c)$.

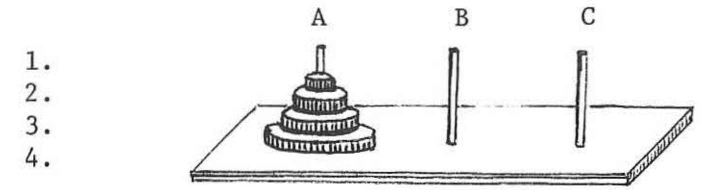
Thus the principle of using syntactically parallel notation to represent mathematically parallel concepts allows greater insight at the elementary level into the meaning of structural properties of binary functions. APL contains many other notational innovations which have potential application to the teaching of mathematics (Peelle 1974 and 1979).

B Non-Standard Problem Representations

Sometimes a standard representation is not available—either the problem itself poses a novel symbol-configuration together with rules of procedure, as in the "checkerboard problem" above, or the solver is expected to construct a new representation for the problem. State-spaces for such problems have been used to define task structure variables, to characterize "relatedness" between problem representations, and to record the behavior paths taken by subjects (Goldin 1979). Two problems are said to be isomorphic when the states, legal moves, and solution paths of one can be placed in one-to-one correspondence with the states, legal moves, and solution paths of the other. A problem has *symmetry* if it is isomorphic to itself in more than one way.

We shall consider the example of the Tower of Hanoi problem and

its isomorphs, which have been studied by several authors (Simon & Hayes 1976; Hayes & Simon 1977; Luger 1979; Luger & Steen 1981):

> 4 Four concentric rings (labeled 1, 2, 3, 4 respectively) are placed in order of size, the smallest at the top, on the first of three pegs (labeled A, B, C), as in the diagram:
>
> 
>
>
> The object of the problem is to transfer all of the rings from peg A to peg C in a minimum number of moves. Only one ring may be moved at a time, and no larger ring may be placed above a smaller one on any peg.

The complete state-space for this problem is shown in Figure 5. Each state is labeled with four letters, referring to the respective pegs on which the four rings are located. From the network of states the problem symmetry is apparent—the roles of pegs A, B, or C can be exchanged without changing the structure of the problem. In particular, state BBBB is conjugate to the goal state CCCC, but is not itself a goal. The state-space displays forward-backward symmetry in that if CCCC is taken as the initial state and AAAA as the goal, the problem structure is unchanged.

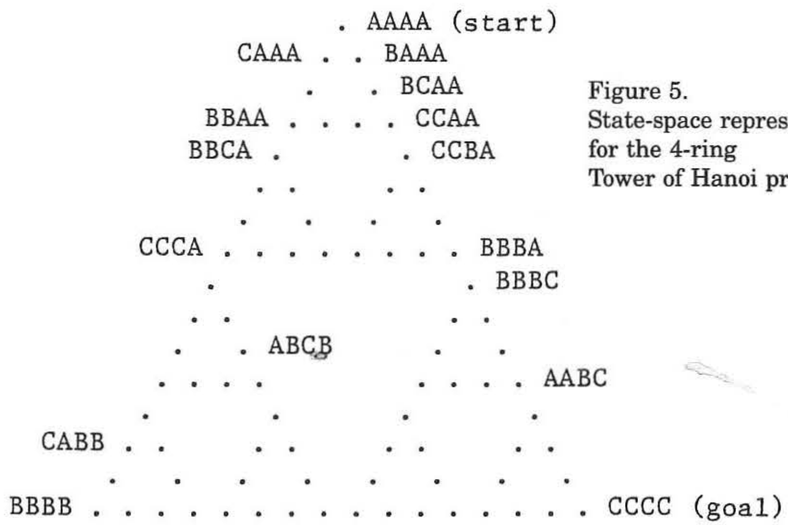


Figure 5.
State-space representation for the 4-ring Tower of Hanoi problem.

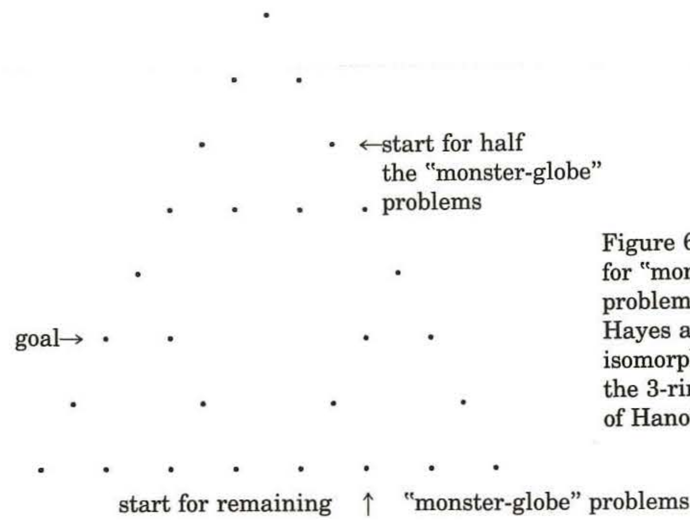


Figure 6. State-space for "monster-globe" problems of Hayes and Simon, isomorphic to the 3-ring Tower of Hanoi problem.

In the above version of the problem, studied by Luger, the pegs and the board present the solver with a notation for keeping track of moves, and solvers proceed by means of successive trials. This notation is extremely efficient for determining the availability of legal moves, but it does not preserve information about the history of moves which have occurred. The symmetry that is present is overt — "there to be noticed" — in the external representation. During the course of problem solving, some solvers who have started on a path heading "towards" state BBBB in the state-space in Figure 5 come to recognize the symmetry, and are able to correct to the symmetrically conjugate path leading to the goal. Additional discussion of symmetry as a task variable, and of overt vs. hidden symmetry, may be found in the references (Goldin 1979; Luger 1979; Goldin & McClintock 1980; Luger & Steen 1981).

Hayes and Simon employed isomorphs of the 3-ring Tower of Hanoi problem in order to study the effects of changing the problem statement on the notations adopted by subjects. The tasks were eight different "monster-globe" problems, stated in complicated language, all of which (when represented most efficiently) had state-spaces as in Figure 6. The tasks differed from each other in two ways: In Transfer problems a monster or globe was moved, while in Change problems a monster or globe changed size. Secondly, in Agent problems the monsters moved or changed the globes, while in Patient problems they moved or changed themselves. Some of the problems also differed from the others in the description of the initial state.

Since no external notational language was presented to subjects beyond the problem statement, it was necessary for them to devise their own. Three main types of notation were "invented" under these conditions, called "operator-sequence" notation, "state-matrix" notation, and "labeled-diagram" notation. These notations preserved the history of moves, but were of varying efficiency for testing the legality of moves, and not nearly so efficient as the rings-and-pegs apparatus. The types of notation remained relatively constant in frequency across problem variables. However, Transfer problems and Change problems elicited different notations within the broader categories of operator-sequence and state-matrix notations, and Agent and Patient problems elicited some differences within the operator-sequence category. For example in the operator-sequence category, indirect naming of objects was used more frequently with Change problems than with Transfer problems. In the state-matrix category Transfer problems resulted in symbols being moved from column to column, while Change problems resulted in symbols being altered within each column. Hayes and Simon postulate how the notational differences might have been caused by the differences in the problem statements.

It was observed that Change problems required greater times to solution. Change problem notations required more steps to test the problem conditions in selecting legal moves than did Transfer problem notations. This study is convincing in demonstrating how the choice of notation may affect problem difficulty through increasing the complexity of move selection.

In a discussion of the well-known "missionary-cannibal" problem, the author has suggested that the extra steps needed to test moves for legality may be described by enlarging the formal state-space to include additional "testing" moves (Goldin 1979, p. 135). In the present case this would result in a more complex state-space in which the states of Figure 6 are embedded. Such an embedding of one state-space into another is an example of one kind of state-space *homomorphism*. In general, homomorphisms may be used to describe the different kinds of relatedness which can exist between alternate problem notations.

The preceding examples lead to the following observations about efficient notational language: (1) Features of problem states which have to do with nearness to solution (as we saw in Problem 3) should be visible in the notation. (2) Problem symmetry should be overt rather than hidden wherever possible. (3) Problem representations are more efficient when the information needed to move from one symbol-configuration to the next is visually apparent in the notation, or requires few steps to obtain from each state.

IV Planning Language

This section focuses very briefly on the specialized domain of ordinary English devoted to heuristic planning. It is plausible that just as notational language can be efficient or inefficient for problem solving, so can planning language. Explicit attention to language at the planning level would then be necessary before we can teach problem solving in the same way that we now teach students mathematical notation. It may be valuable to introduce planning symbolism in order to make more visible the steps in the planning process.

Polya (1945) proposed to organize heuristic processes into four main stages: understanding the problem, devising a plan, carrying out the plan, and looking back. Much of his subsequent work was devoted to elaborating on the processes contributing to each stage. Wickelgren (1974) sought to improve problem-solving planning by introducing more technical language from artificial intelligence research—for example, he discusses "hill-climbing" which is a metaphor for state-space search algorithms with evaluation functions used in mechanical problem-solving programs. Schoenfeld (1979) devised a more elaborate stage model for organizing heuristic processes, reproduced in Figure 7.

An earlier version of Schoenfeld's model formed the basis for an extraordinarily detailed process-sequence coding scheme developed by Lucas et al. (1979), in which over fifty different symbols are used to represent process and outcome categories observed during problem solving. More recently, the author worked with Carpenter, Kulm, Schaaf, and Smith toward grouping these into a more manageable system for recording the processes used by junior high school students (Kulm, et al. 1981). This system is still undergoing revision, but in order to convey its flavor a partial dictionary is given in Figure 8. Next to each process code, the language level to which this code refers, or the translation process to which it refers, has been indicated. Thus a correspondence can be drawn between the observed processes in problem solving, and the kinds of language depicted in Figure 1.

The domain of planning language about which we can say the most, based on examples in this paper, is that which governs or talks about notational language. Silver, Branca, and Adams (1980) have examined the role of "metacognition" in problem solving. In fact, planning language as described in the present paper is a "meta-language" with respect to formal problem-solving notations. It includes the following kinds of steps: adoption of a notational language; choice of a goal or subgoal state within a notational language; modifying notational language to describe simpler problems; modifying notational language to reduce the complexity of moves; modifying notational language to

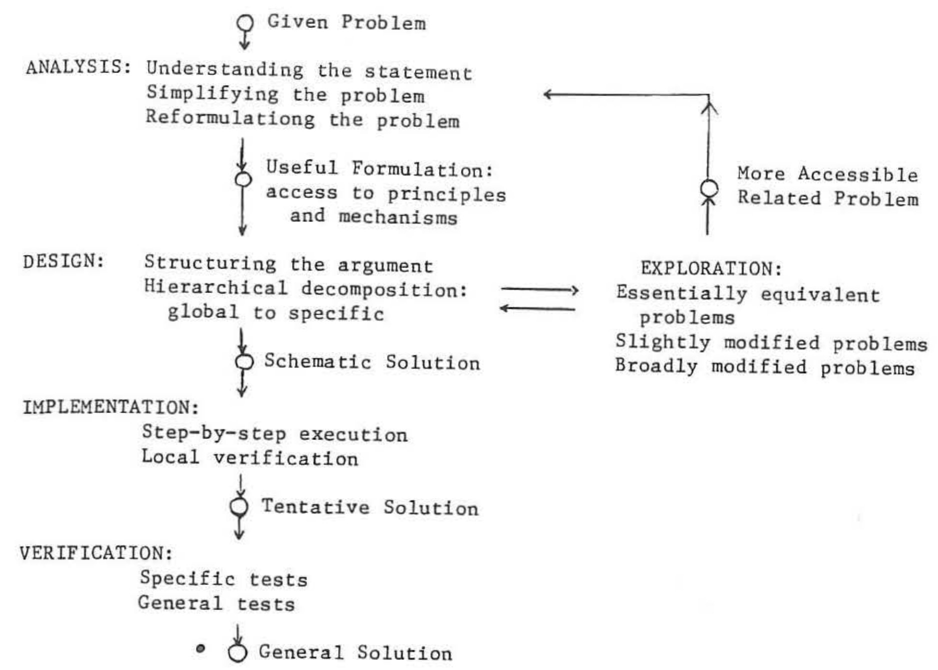


Figure 7. Schoenfeld's schematic overview of problem-solving stages (abridged from Schoenfeld 1979).

make symmetry more overt; and modifying notational language to make more visible features of problem states describing nearness to solution. In our discussion we have seen examples of how such notational modifications could greatly assist the problem solver. The inefficient or naive planner is unable or unwilling to take such steps. It therefore seems reasonable to conjecture that explicit introduction of planning language into problem-solving instruction, including practice in talking *about* problem notations and evaluating their effectiveness, could substantially improve higher problem-solving skills.

Kulm, G., Campbell, P. F., Frank, M., Talsma, G., & Smith, P. (1981). Analysis and synthesis of mathematics problem solving processes. NCTM 59th Annual Meeting Presentation.
Lester, F. (1980). "Research on mathematical problem solving." In: R. J. Shumway (Ed.), *Research in Mathematics Education.* Reston, Va.: NCTM.
Loftus, E. J. (1970). An analysis of the structural variables that determine problem-solving difficulty on a computer-based teletype. Tech. Report 162, Inst. for Mathematical Studies in the Social Sciences, Stanford University.
Lucas, J. F., Branca, N., Goldberg, D., Kantowski, M. G., Kellogg, H., & Smith, J. P. (1979). "A process-sequence coding system for behavioral analysis of mathematical problem solving. In: G. A. Goldin & C. E. McClintock (Eds.), *op. cit.*
Luger, G. F. (1979). "State-space representation of problem-solving behavior. In: G. A. Goldin & C. E. McClintock (Eds.), *op. cit.*
Luger, G. F., & Steen, M. (1981). "Using the state-space to record the behavioural effects of symmetry in the Tower of Hanoi problem and an isomorph." *Int. J. Man-Machine Studies, 14* (in press).
Nilssen, N. J. (1971). *Problem Solving Methods in Artificial Intelligence.* New York: McGraw-Hill.
Peelle, H. A. (1974). "Computer glass boxes: teaching children concepts with A Programming Language." *Educational Technology, 14,* 9-16.
Peelle, H. A. (1979). "Teaching mathematics via APL (A Programming Language)." *The Mathematics Teacher, 72,* 97-116.
Polya, G. (1957). *How To Solve It.* New York: Doubleday (second edition).
Polya, G. (1962, 1965). *Mathematical Discovery, 1 & 2. On Understanding, Learning, and Teaching Problem Solving.* New York: Wiley.
Schoenfeld, A. H. (1979). "Heuristic behavior variables in instruction." In: G. A. Goldin & C. E. McClintock, *op. cit.*
Silver, E. A., Branca, N. A., & Adams, V. M. (1980). "Metacognition: the missing link in problem solving?" In: *Proceedings of the Fourth International Conference for the Psychology of Mathematics Education,* R. Karplus (Ed.). Berkeley, Cal.: Lawrence Hall of Science.
Simon, H. A., & Hayes, J. R. (1976). "Understanding complex task instructions." In: D. Klahr (Ed.), *Cognition and Instruction.* Hillsdale, N. J.: Erlbaum.
Webb, N. (1979). "Content and context variables in problem tasks." In: G. A. Goldin & C. E. McClintock, *op. cit.*
Wickelgren, W. (1974). *How to Solve Problems: Elements of a Theory of Problems and Problem Solving.* San Francisco: W. H. Freeman.

Symbols, Icons, and Mathematical Understanding

William Higginson

Extracts are taken from the biographies of Hobbes, Rousseau, Darwin, and Russell which refer to their mathematical education. The common feature of an attraction toward geometry and an aversion to elementary algebra is noted. These experiences are analysed using theoretical positions promulgated by Davis, Hersh, Skemp, and Bruner. The central thesis is that these men probably have had difficulty learning elementary algebra because they had failed to develop a strong image or iconic representation of the concepts involved. This thesis is developed in relation to "squaring a binomial," the concept which troubled both Rousseau and Russell.

> *Mathematics is often considered a difficult and mysterious science, because of the numerous symbols which it employs.*
> *A.N. Whitehead*

Much of the power of mathematics stems from the potency of its symbols. There is, however, a price to be paid for this potency. The symbols which serve as highly effective tools for some are the most formidable of barriers for others. In the following pages a thesis is outlined which attempts to account for some of the difficulties which learners meet when studying mathematics. The method of approach is largely biographical; the essence of the argument: that we have paid too little attention to the role of images in mathematical understanding.

The unique cluster of insights, associations, and emotions which characterizes every encounter of individual with idea is never easy to capture. One of the few sources to which we can turn in such a quest is biographical literature. The examination of this literature for accounts of man meeting mathematics reveals some interesting commonalities in the experiences of a number of people. For our purposes we consider four distinguished thinkers; Thomas Hobbes (1588-1679), Jean-Jacques Rousseau (1712-1778), Charles Darwin (1809-1882), and Bertrand Russell (1872-1970).

One of the most striking features of John Aubrey's marvelous collection of short biographies, *Brief Lives,* is the picture it gives of the impact of the release of the mathematical sciences from the Greek and

Visible Language, XVI 3 (Summer 1982), pp. 239-248.
Author's address: Faculty of Education, Queen's University, Kingston, Ontario K7L 31V6, Canada.
0022-2224/82/007-0239$02.00/0