DOCUMENT RESUME

ED 259 032 TM 850 421

AUTHOR McAllister, Alan
TITLE Problem Solving and Beginning Programming.
PUB DATE Mar 85
NOTE 35p.; Paper presented at the Annual Meeting of the American Educational Research Association (69th, Chicago, IL, March 31-April 4, 1985).
PUB TYPE Speeches/Conference Papers (150) -- Reports - Research/Technical (143)

EDRS PRICE MF01/PC02 Plus Postage.
DESCRIPTORS Academic Achievement; Classroom Research; Cognitive Style; Courseware; *Individual Differences; *Learning Strategies; Primary Education; *Problem Solving; *Programing; Puzzles; Self Concept; Sex Differences; Time Factors (Learning); Young Children
IDENTIFIERS *LOGO Programing Language; *Tower of Hanoi (Puzzle)

ABSTRACT

Based on current models of problem solving within cognitive psychology, this study focused on the spontaneous problem solving strategies used by children as they first learned LOGO computer programming, and on strategy transformations that took place during the problem solving process. The research consisted of a six weeks programming training project using Turtle graphics on Texas Instruments Computers with 19 students in a combined second and third grade classroom. Data were collected from two math tests, teacher ratings of reading and spelling skills, a self concept inventory (the Student's Perception of Ability Scale), records of student time on the computer, and three pencil and paper programming tests. In addition, the programming strategies of eight students were studied closely on the Tower of Hanoi puzzle which has structural similarities to the LOGO language and facilitates similar forms of problem solving. Two main spontaneous problem solving strategies were identified: partitioning and encompassing. These strategies reflected the different ways in which the children represented the task problems (either as extended series or as subunits for unit building), thus managing their memory resources. While unanticipated sex differences in strategies were identified, they were not generalizable due to the research design. (BS)

PROBLEM SOLVING AND BEGINNING PROGRAMMING

Alan McAllister

Psychological Services
The Toronto Board of Education
Toronto, Ontario

Department of Psychology
York University
Downsview, Ontario

February 19, 1985

ACKNOWLEDGEMENTS

The author wishes to acknowledge the following:

The Department of Psychological Services of the Toronto Board of Education for supporting the research project which was the basis for this study;

The staff of Garden Avenue Public School and especially Dr. Forbes Carey, the principal, and Ms. Debra Porter, the teacher of the children studied who provided moral support to the author and was a great help in collecting data;

Dr. James Fair of the Faculty of Education, the University of Toronto, Mr. Tory Hansen of the Institute of Child Study, Ms. Betsy McElvry of the Ontario Institute for Studies in Education, and Mr. Trevor Brown of the Curriculum Department of the Toronto Board of Education, all of whom served on the committee overseeing the research project;

Dr. Marlene Scardamalia of the Psychology Department of York University for her suggestions for the research project and her helpful criticisms of earlier versions of this paper; and

The children of the classroom and especially the four girls and four boys who were selected for intensive study.

INTRODUCTION

This study is concerned with programming as a problem solving activity of young children. Its focus is on the spontaneous or initial problem solving strategies that children use as they first learn LOGO programming and the transformations of these strategies that take place in the course of their problem solving.

The theoretical framework for this study is based on current models of problem solving within cognitive psychology. Newell and Simon's (1972) contributions to the field inspired psychological analyses which initially focused on the mechanics of problem solving, the procedures that are followed in successful problem solving, but more recently attention has been focused on the semantics of problem solving, "how problems are represented and how meaningful relationships within the problem are used in finding and understanding problem situations" (Greeno, 1978).

Since the publication by Bruner and his colleagues of A Study of Thinking (Bruner, Goodnow, & Austin, 1956), cognitive strategies have been recognized as important components of problem solving. These strategies may involve the exercise of some degree of control over such processes as attending, perceiving, encoding, remembering, and thinking (Gagne, 1984). In contrast to such constructs as intelligence or cognitive style, problem solving strategies are changeable (Neches et al, 1978) and learnable (Wood, 1978; Gagne, 1979; and McKenny & Haskins, 1980). They can be highly specific to a task environment or very general and found in a variety of problem solving situations.

An example of a specific strategy would be a child using the learned strategy of regrouping to solve an addition problem. An example of a general

strategy would be when a child uses a self-questioning technique as he does a task to control the pace of his responses and to draw his attention to selected features of the task. Most discussions within the problem solving literature involve strategies falling between these two extremes. Two examples illustrate the nature of these strategies and the kinds of analyses that can be done of them.

Bruner (Bruner, Goodnow, & Austin, 1956) investigated a concept learning task and distinguished two basic strategies-- a focusing strategy, which involves remembering just those attributes which are common to instances that are known as exemplifying the concept, and a scanning strategy, which involves forming one hypothesis at a time and keeping it until it is disconfirmed. The focusing strategy was found to be more efficient in learning than the scanning strategy, and its efficiency was hypothesized as due to the lesser "cognitive strain" or load on memory resources that it involves.

Simon (1976, 1979) investigated the ancient puzzle, the Tower of Hanoi, and distinguished four or five basic strategies by which the puzzle could be solved in the minimum number of moves. Some of these strategies involve an understanding of the logical structure of the puzzle, while others are based on noting certain regularities in the sequences of moves in solving the puzzle or simply remembering past successful solutions. He distinguished the strategies as primarily goal-driven or stimulus-driven strategies and observed that

> ... the strategies make different demands upon short-term memory, require different sets of concepts, and require different perceptual tests to be made for their execution. (1979)

Underlying strategic differences are differences in representations of the problem in memory. Investigations of chess performance (Chase and Simon, 1973a; 1973b) and balance scale problems (Siegler, 1976) suggest that superior

performance in these tasks involves encoding the problems "at the level of organized configurations rather than at the level of individual items" (Siegler, in press), in terms of general principles rather than details, and in terms of larger chunks consisting of familiar subconfigurations (Chase and Simon, 1973a). In chess, for instance, hierarchical organization of chunks seems to lie behind higher levels of skill (Chase and Simon, 1973a).

Programming provides a "semantically rich" (Adelman, 1981) domain for the investigation of strategic differences in problem solving and underlying differences in the representation of problems. Comparisons of expert and novice programmers have shown that there are differences in strategies between them (Schneiderman, 1976; Adelson, 1981; Coombs et al, 1981) and that more efficient forms of representation are used by the experts. Adelson, using a measure of free recall of randomly ordered lines of programming code, found that experts recalled more than novices, that they recalled in larger chunks, and that what they recalled had more "consistent subjective organization" based on the functional principles of their area of expertise (Adelson, 1981). Statistical analysis suggested that experts

> ...chunked individual items into integral wholes that were then organized hierarchically according to procedural similarity. This hierarchical chunking suggests an underlying categorical encoding in which certain features are used as a basis for similiarity and others are ignored. (Adelson, 1981)

The subjects of this study were all novices in programming. The method chosen for this study was to identify individual differences in how a group of children went about problem solving to see how these differences were reflected in their programming; on the basis of these differences, hypotheses can be formed about differences in problem solving strategies and underlying representations of the problem. The discussion proceeds from an analysis of how a classroom of

children learned LOGO programming to a more fine-grained analysis of the problem solving processes of a small group of children. The statistical analyses used here are in the spirit of "teasing-out" information to provide a context for the discussion. The findings are tentative and should be viewed as hypotheses to be investigated in future studies.

THE CLASSROOM CONTEXT

The classroom was a split grades two and three cla . While the children had not had training in programming, they had used a Commodore "PET" computer for games and computer assisted instruction. Since LOGO cannot be implemented on the PET, a Texas Instrument computer, or a "TIC", as the children called it, was used.

The project lasted six weeks and the children were taught the basics of Turtle graphics. Most of the instruction was confined to weekly morning sessions with the whole group of children or the instructor worked with children in small groups of three to five. One-to-one training was given toward the end of the project.

The actual time the children spent on the computer was determined by a computer schedule. The children had fifteen minute blocks of time and they were allowed to take a buddy with them while they used the computer. Signing up for a computer was voluntary and which computer the child chose (the PET or the TIC) was determined by their own wishes and the availability of that computer.

Because the instructor's time was limited, there were limited opportunities for demonstration and, because only one TIC was available for most of the project and the children had to complete their normal classroom curriculum, the amount of experimentation was severely limited. Whatever deficiencies this kind of training may have had, it is probably representative of the kinds of environments in which computer instruction is currently being given.

During the instructional period, a variety of academic measures and measures of academic self-concept were administered to the class as a whole and, at the end of the period, a set of paper and pencil programming tasks was administered.

Eight randomly selected children, four girls and four boys, were also tested individually using the Tower of Hanoi in sessions prior to and after the training and, at the end of the project, were given a special programming performance task. The focus of the study is on this subgroup of children, but the data from the larger group will be used to provide some initial hypotheses about programming and its relationship to academic skills and self-concept and as a context for the discussion of the problem solving strategies of the smaller group.

Results on the classroom measures:

Data were collected on nineteen students using two math tests (one given at the outset of the project, another given at the end of the project), the teacher's rating of reading and spelling skills, a self-concept inventory, a record of time spent on the two computers in the classroom, and three paper and pencil programming tests. The two math tests were based on the written section of the Key Math Diagnostic Arithmetic Test (Connolly et al, 1979) widely used in the Toronto system. The self-concept inventory was the Student's Perception of Ability Scale (Boersma, Chapman, 1977) which has seventy items with six subscales for the child's self-perception of "general ability", "arithmetic", "school satisfaction", "reading and spelling", "penmanship/neatness", and "confidence".

The programming tests were specifically developed for this project. The three programming measures used differed in the nature of their demands on the problem solving skills and programming knowledge of the children. The first measure, "program reading", involved drawing what would appear on the screen with a given program. The second measure, "program writing", involved devising a program that would produce a given drawing. The third measure, "program

creating", was open ended and most like the actual programming that the children had been doing; it involved writing a program of their choice and drawing what it would draw on the screen. Scoring criteria were developed for the programming measures based primarily on a simple pragmatic criteria of whether there was a correspondence between the program and the drawing of the program's effect.

There were two sets of results that highlight how programming was learned in this classroom-- time on the computers and the relationship of programming to academic skills and self-concept. Sex differences were also found on some of the programming measures.

--Time on computers

Use of the (non-LOGO) PET and (LOGO) TIC was found to be significantly correlated with the first math measure and with the teacher's rating of reading and spelling skills ($p<.02$). However, the children tended to prefer one of the computers over the other: TIC time and PET time were negatively correlated ($p<.01$) and TIC time was positively correlated with the programming measures ($p<.05$) while PET time was not (it was negatively correlated with program writing ($p<.05$)). When PET time versus TIC time were compared, PET time was not correlated with the first math measure or the teacher's rating of reading and spelling skills and, in fact, was negatively correlated ($p<.05$) with the second math measure. TIC time, on the other hand, was positively correlated with both math measures and the teacher's rating of reading and spelling skills ($p<.01$). Furthermore, while TIC time was positively correlated with the measure of the children's perception of their general ability ($p<.02$), PET time was negatively correlated with the same measure ($p<.10$).

--Relations between programming and academic skills and attitudes:

The programming measures all positively correlated with one another ($p<.01$), suggesting that the skills tapped by these measures were reasonably homogenous. The total score for the programming measures was positively correlated ($p<.05$ or better) with the total for the self perception inventory and with the subscale measures of general ability, arithmetic, and reading and spelling. There were also positive correlations ($p<.01$) between the programming total score and both math measures and the teacher's rating of reading and spelling skills.

--Sex differences:

Student's t test was used to compare the means between the boys and girls. No significant differences were found on any of the non-computer related tasks. However, although there was no significant difference between the amount of time the boys and the girls used the computers, the boys did significantly better on two of the programming measures, program reading ($p<.01$) and program writing (.10) and on the total score for

programming ($p<.05$). However, there was no significant difference between the boys and the girls on the program creating task which was most like the actual programming that they had done.

The implications of these findings are fairly clear. The PET represented one form of computer use-- the sort of games and computer-assisted instruction with which the children were familiar. The TIC repesented another form of computer use-- a new activity in which the child controlled the computer. When children had a choice of the two activities, the children who felt less able and, in fact, were not as able in math, tended to shy away from the new activity and preferred to spend their time with more familiar computer activities, while those who felt academically more able and were more able in math took on the challenge of the new machine. The correlations between the academic and programming measures indicate that the better a child was academically and the better the child's school-related self-concept the more likely it was that the child learned the basics of programming.

That the more able and more confident children were likely to seek out the challenge of a new activity and learned more about it is not a particularly surprising result. However, the sex differences were unanticipated and mentioned here because they assume importance in the later discussion of the small group.

DATA FROM THE SMALL GROUP

In this section, the rationale for using the Tower of Hanoi to study programming will be developed; the focus will be on the structural similarities between the puzzle and the LOGO language and the similar forms of problem solving that the puzzle and LOGO programming facilitate. The procedures used in administering the puzzle and the programming measure and the children's performances on the two tasks will be analyzed as a preliminary to the development of a framework for describing the strategies of the four boys and four girls who are the focus of this study.

The rationale for the use of the Tower of Hanoi:

One method that has proved successful in studying individual differences in approaching programming has been to present subjects with two tasks, an "indicator" task and a programming "target" task (Coombs et al, 1981). For a task to be a good indicator of its target, it must be well-understood, performance on it must be easily studied, and there must be similarities between the two tasks which make performance on the indicator task a basis for generating hypotheses about strategies and performance on the target task (Coombs et al, 1981). The Tower of Hanoi was chosen because it fulfilled these three requirments for an indicator task for LOGO programming:

(1) It is a well-known task that has been extensively studied in the literature on problem solving (Anzai and Simon, 1979; Karat, 1982; Klahr and Robinson, 1981; Luger, 1976; Luger and Steen. 1981: Nilsson, 1971: Piaget, 1976: Simon, 1976, 1979).

(2) The puzzle is representative of a class of transformation problems

which involve reaching a goal through a sequence of moves. As such and because it is a physical puzzle, it involves a series of observable steps so the decision-making process of the child is accessible for analysis.

(3) It has structural features in common with LOGO and facilitates a similar approach to problem solving.

The Tower of Hanoi is structured as a set of nested subproblems having the property of recursion. There are three pegs and on one peg are arranged a number of disks of increasing size from top to bottom. The task is to transfer all of the disks from the first peg to the third peg in the minimum number of moves under the constraint of two rules: (1) a larger disk cannot be placed on top of a smaller and (2) only one disk can be moved at a time. For each set of disks there is a minimum number of moves according to the formula 2^n-1 with n equal to the number of disks. The problem is recursive in that a problem of n disks can be decomposed into subproblems of the $n-1$ form.

A representation of the structure of the problem can be made in terms of the concept of state representation (Nilsson, 1971; Luger, 1976). A state representation of the four disk problem is shown in Fig. 1. Each circle stands for a possible position or state of the puzzle. The four numbers labelling a state refer to the respective pegs on which the four disks are located. Thus 1 in the first position refers to the state in which the smallest disk is on the first peg, 3 in the second position refers to the state in which the second smallest disk is on the third peg, and so on. A legal move involves a transition from one of the circles representing a state to a neighboring one in the state space. For the four disk problem the minimum number of moves consists of 15 moves (2^4-1) down the right hand side of the triangle from 1111 to 3333. This four disk state-space can be seen to be composed of three three disk subspaces

and nine two disk subspaces (Luger, 1976); the recursive structure is graphically evident in the composition of the whole space out of structurally similar subspaces. It is this structure which facilitates an approach to problem solving which breaks the problem down into subproblems.

```
*************************************
*                FIG. 1             *
*            STATE DIAGRAM          *
*       FOUR DISK TOWER OF HANOI    *
*                     start         *
*                  0   1111         *
*                 0 0   2111        *
*                0   0              *
*         2211  0 0 0 0  3311       *
*              0       0  3321      *
*             0 0     0 0           *
*            0   0   0   0          *
*     3331  0 0 0 0 0 0 0 0  2221   *
*          0               0  2223  *
*         0 0             0 0       *
*        0   0           0   0      *
* 1132  0 0 0 0    3323 0 0 0 0  1123 *
*      0       0       0       0    *
*     0 0     0 0     0 0     0 0   *
*    0   0   0   0   0   0   0   0  *
*   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 *
*2222                            3333*
*                               goal*
*                                   *
*          (adapted from Luger, 1976)*
*************************************
```

The Tower of Hanoi has been used to describe two concepts in LOGO-- the idea of a subprocedure and the idea of recursion (Harvey, 1982). As a procedural language, LOGO is structured in such a way that procedures, which consist of one or more instructions, can be themselves parts of larger procedures; a procedure is recursive if it is a subprocedure of itself (Abelson, 1982a and 1982b; Harvey, 1982). Because of this organization of the language, a problem can be broken down into manageable units and separate procedures can be written for each unit. But this approach of breaking down problems into subproblems and building a

solution from mastered subunits is facilitated at an even more fundamental level in the process of program development.

As in most interpreted computer languages, there are two basic modes of using the language. In LOGO there is the immediate or draw mode which allows the user to issue commands which are immediately executed, and there is the definition or edit mode in which commands are written out as procedures which then can be tried out in immediate mode. The existence of these two modes and the possibility of passing from one to the other provide a basis for dividing a complex problem into more manageable subproblems.

A child can operate totally within the draw mode and have the "Turtle" carry out commands immediately as they are typed into the computer. But if he wants to write a procedure he must go into the edit mode and write out instructions and, if he wants to see the effect of the procedure, he must return to the draw mode and run the procedure. A kind of "dialectical" process analogous to what happens in writing (Scardamalia and Bereiter, 1981) can occur as the programmer uses the results in the draw mode as a basis for modifying the procedure until it achieves his intentions.

In summary, there are a number of similarities between the two tasks which provide a basis for the relationship of indicator task to target task. There are structural characteristics in common and fundamental similarities in the way in which the problems posed in the two tasks can be broken down into subproblems and these elements built into a solution.

Procedures for administration of the tasks:

The children had two sessions with the puzzle, one at the beginning of the training program and one at the end. The procedure was designed to encourage the

children to think in terms of the structure of the problem. In the first session, after successfully completing the two disk problem, the children were given the three disk puzzle and then the four disk puzzle. The three disk puzzle was then readministered. In the second session, after trials with the two disk and three disk problems, they were given the four disk problem. As an aid to their thinking processses (Gagne and Smith, 1962; Brown et al, in press), they were encouraged to verbalize as they did the puzzle and to say what they were doing and why and, at the end of each of the two sessions, they were asked to explain the "secret" of how to solve the puzzle to see if they could abstract a general principle or rule for the solution of the puzzle.

The programming test was given at the end of the project. It was designed to measure the programming proficiency of the children and mimiced closely the kind of interaction the children had had with the computer in the classroom. It was most closely related to the paper-and-pencil program creating task given to the class as a whole.

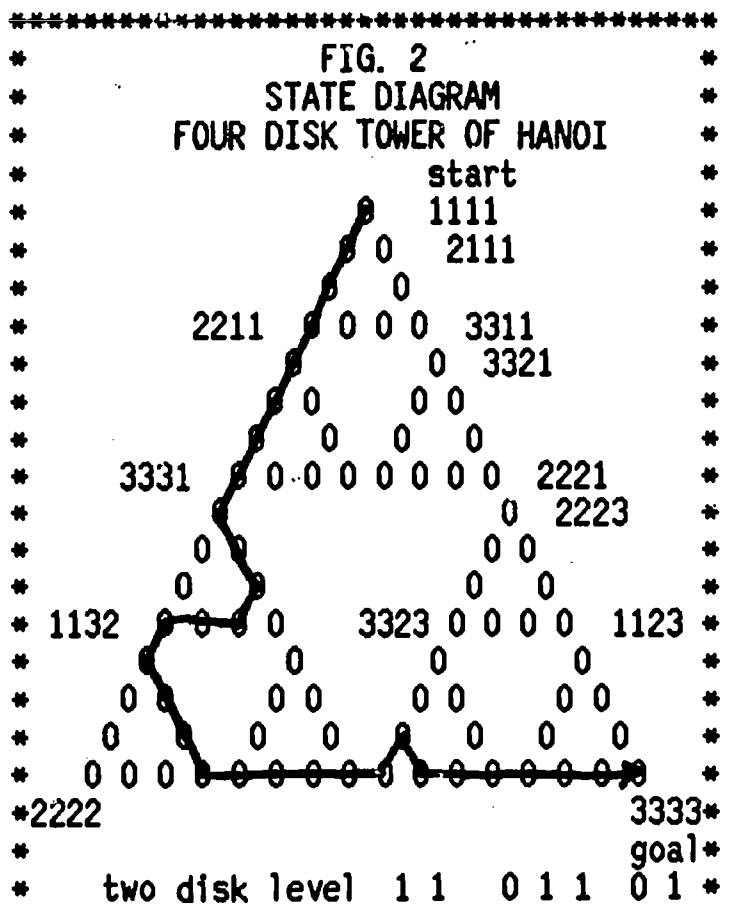
As in the classroom, each child was given a limited time period (ten minutes) in which to create a procedure or as many procedures as he wished. The major differences were that they could not work with other children and the instructor was present to ask questions and to help them over any major hurdles they might encounter, although interventions were kept to a minimum. Since the object was to create a procedure or procedures, whenever a child seemed to have written a procedure and had tried it out in the draw mode, the instructor would ask whether the child wished to write another procedure or modify the existing procedure. Occasionally a child would linger in the draw mode writing a series of commands; if the period of time became prolonged, the instructor would remind the child that the point of the exercise was to create procedures.

Analysis of the two tasks:

The analysis of the two tasks involves basically two steps. The first step involves finding the most significant characteristics of the children's performance on the indicator task. Correlations between the performances of the small group on the indicator task and on the group-administered measures will be used to bring salience to these characteristics. Since the children performed in identifiably different ways in relationship to these characteristics, the next step involves seeing how these differences carried over to the target task with respect to similar characteristics of that task. This procedure provides a sound basis for developing an explanation of the children's approaches to the tasks in terms of different strategies which are common to both tasks.

To quantify the children's performance on the Tower of Hanoi, two scoring systems were used-- one according to the number of moves to solution and the other according to the recursive, subproblem structure of the puzzle. While little of significance was found using the first scoring system, several significant correlations were found using the second system.

This system is based on Luger's concept of "n-ring episodes" in solving the Tower of Hanoi (Luger, 1976). The problem is divided into its subproblems and the path through the problem is considered in terms of whether the problem solver passes through the problem space or subproblem space in the minimum number of moves. Various levels are considered. For instance, for the four disk problem, there is the three disk subproblems level and the two disk subproblems level. The scoring system simply involves determining the percentage of successful (minimal solution) paths at any level. Fig. 2 illustrates possible scorings.



Although all subproblem levels for all trials were analyzed, the discussion will be restricted to the n-1 levels of the second trials of the three and four disk puzzles since this places the focus on the recursive structure of the puzzle and on the trials which were based upon the children's previous experiences with the puzzle:

> For the second trial of the three disk puzzle, the percentage of consistently solved problems at the two disk subproblems level was positively correlated with the program writing test ($p<.10$), the program creating test ($p<.05$), the total for the programming tests ($p<.10$), the first math test ($p<.10$), the second math test ($p<.05$), and the teacher's rating of reading and spelling skills ($p<.10$).
>
> For the second trial of the four disk problem, the percentage of consistently solved problems at the three disk subproblems level was correlated with the program reading test ($p<.01$) and the total for the programming measures ($p<.10$).
>
> Averages were obtained by combining the percentages of consistently

solved subproblems at the n-1 level of the second trials of the three and four disk problems. Correlations were found between these averages and computer time ($p<.10$), the first math test ($p<.10$), program reading ($p<.01$), program writing ($p<.05$) and the total for the programming measures ($p<.01$).

Sex differences were also found. Again focusing on the second trials of the three disk and four disk puzzles, while the girls tended to do somewhat better than the boys (although not significantly) on the first scoring, on the second scoring the boys did significantly better on the four disk problem at the three disk level ($p<.001$). Boys also performed better on the average for the subproblems of the next lower level ($p<.10$). In the second trial of the four disk problem, none of the girls solved the three disk subproblems, even though two of them had solved the three disk problems consistently in two trials and one had solved it in the second trial of the three disk problem. All of the boys solved one of the three disk subproblems in the second trial of the four disk problem, even though only two of them had been able to solve the three disk problem in any of the three trials.

While few significant correlations with the skills and self-perceptions surveyed in the group measures were found on the first scoring system, several correlations were found on the second, suggesting that this scoring system brings out more important characteristics of performance on this task. These results indicate as well that the children used two distinct approaches to solving the puzzle. The boys' higher scores in terms of subproblem solutions suggest they used the recursive structure of the puzzle to a greater extent than the girls; however, the somewhat more efficient performance of the girls in terms of numbers of moves suggests that they too used a distinct approach based on something other than the recursive structure of the puzzle.

Having identified these two distinct approaches with respect to the indicator task, the next step is to see if the children approached the target task in a similar manner.

Three aspects of the children's programming were examined for indications of differences in approach by the two groups: (1) mechanical proficiency in using the computer; (2) the quality and coherence of the procedures they produced; and (3) the "dialectical" use of the computer, that is, the extent to which the

children alternated between the edit and draw modes and modified their procedures on the basis of the feedback they obtained by seeing what effects the procedures had in the draw mode.

Significant differences between the two groups were found only on the third aspect. Proficiency in the mechanics of using the computer and the lack of it and command of coherent form and the lack of it were found in both groups. However, all the boys developed their procedures by continuously going back and forth between the edit mode and draw mode, and they generally used the information they obtained from seeing the procedures carried out for modifying their procedures. The girls, on the other hand, tended to write procedures from start to finish and did very little if any modifications of their procedures and tended not to run their procedures for feedback.

These differences may explain why the boys and girls differed on the two paper-and-pencil programming measures administered to the group. For the program creating test, on which there were no significant sex differences, the child could rely on his or her knowledge of what a particular program would create on the screen. However, for the measures involving reading and writing programs, on which there were differences, the child had to be generally familiar with the relationship between commands and their effects, that is, with the feedback element.

Thus, the boys' approach, which involved more consistent solutions of subproblems within the indicator task, translated into an approach of alternating between edit and draw modes in developing procedures, whereas the somewhat more move-efficient approach of the girls translated into a start-to-finish approach in the target task. What remains to be shown is whether these differences in approach can be explained in terms of strategic differences.

ANALYSIS OF STRATEGIES AND THEIR TRANSFORMATIONS

The analysis of the data from the randomly selected group suggests that the children spontaneously used different means of going about solving the Tower puzzle and doing the programming task and that the children divided into two groups according to the means they used. In this section, the strategies used by the children in both tasks will be identified and the different transformations of these strategies represented in this group of children will be analyzed. The basis for the strategic differences will be explained in terms of different forms of representation of the problem. The relative value of the strategies will be assessed in terms of cognitive strain and the relationship of the strategies to the structure of the puzzle and the LOGO language. An analysis of transformations of these strategies that took place in the course of the children's problem solving in the two tasks will exemplify the descriptive framework developed here.

The strategies:

Three strategies can be distinguished-- two main strategies, the partitioning and the encompassing, and a strategy auxiliary to these two, exploration. These strategies differ from those that Simon distinguished (Simon, 1976, 1979) for the Tower of Hanoi in that these strategies need not result in solutions in the minimum number of moves and, in fact, can be identified only when they do not result in minimum solutions.

The exploratory strategy is an auxiliary strategy because it provides the information base for the other two strategies which, unlike it, presume some prior knowledge of the problem and its structure. Within the Tower of Hanoi, it

is largely through trying out various combinations of moves that the child finds series of moves that work just as it is through trying commands and procedures on the computer that the child makes discoveries in LOGO.

The partitioning strategy was exhibited within the programming task by children who built programs by writing small subunits and, during the constructive process, sought feedback from the computer by running these units. Within the Tower of Hanoi, these children seemed to break the problem into smaller units and solved a higher percentage of subproblems in the mimimum number of moves; this strategy was exhibited by the high scorers according to the second system, primarily the boys. While similar to Simon's goal-recursion strategy, it cannot be taken as a true recursion strategy which would involve the subject having the concept of recursion (Simon, 1976) and rigorously applying it, but is instead a variant which involves partitioning the problem into manageable units and "unit building" (Neches et al, 1978) the "chunks" into larger and larger units until the unit becomes the problem as a whole. The recursive structure of the puzzle plays a role, however, in that the child need only focus on a current subgoal within a subproblem rather than on the solution of the problem as a whole.

The encompassing strategy was exhibited within the programming task by children who wrote programs from start to finish and ran them only when they were completed. Within the Tower of Hanoi, children employing this strategy seemed to focus on achieving the goal by the shortest path without relying on the subproblem structure of the puzzle; the use of this strategy can be identified by low scores on both scoring systems. While not a perfect rote strategy in Simon's sense (Simon, 1976), it resembles it in that the child must try to keep in mind the entire sequence of moves that are necessary to achieve the solution of the

problem as a whole.

As the child tries to solve the Tower of Hanoi two different representations can take form. The solution of the problem can be represented as a hierarchy of ever larger, but structurally similar subseries or as a single extended series. And this is what distinquishes the two forms of representation that are the bases for the two strategies. The partitioning strategy represents the problem in the form of subproblems or subseries while the encompassing strategy represents the problem at the level of the problem as a whole, that is, as an extended series.

When the children came to the programming task, they seemed to have used similar representations of the new problems that were presented to them. The boys tended to partition a problem into easily manageable bits, into small series of instructions about which they could get immediate feedback, and this is why they adopted an interactive or "dialectical", partitioning strategy. The girls dealt with a problem as an unbroken whole and constructed their procedures in the form of an extended series of instructions, allowing for little if any modifications and not requiring or using feedback, and this is why they adopted a start-to-finish, encompassing strategy.

Although the exploratory strategy is auxiliary to the other two strategies, it has a different relationship to each. The exploratory strategy is a natural ally of the partitioning strategy. In doing the Tower of Hanoi using the partitioning strategy, as long as some of the subproblems are solved, discovery can be part of the problem solving process since exploration can take place between consistently solved subproblems. In the programming task, the child using this strategy can write short series of instructions and then check them out on the screen to see how they work, thereby trying out different things, correcting as he goes, and keeping tabs on how the total construction is

proceeding.

The relationship of exploration to the encompassing strategy is quite different. In the Tower of Hanoi, the child using the encompassing strategy does not break the problem down into subproblems but must hold in mind an extended series of moves to the goal. Solving the puzzle requires keeping on track and does not allow for experimentation along the way. In the programming task, the child using this encompassing strategy would have to be able to think through the entire program in order to ensure that the program would do what was intended. With this strategy, while it is possible to experiment and discover, the discovery is only made once the puzzle has been solved or the program has been constructed; the discovery cannot become a part of the process in the way it can with the first strategy. To be utilized the discovery must be "transported" to the next trial of the puzzle or to another programming project.

The partitioning strategy would seem to have definite advantages over the encompassing strategy not only because of its relationhip to exploration. Because it is easier to remember a subseries than it is to remember the whole series, the partitioning strategy is less demanding of memory resources than the encompassing strategy.

For the Tower of Hanoi, as the number of disks increases, the encompassing strategy becomes increasingly problematic at these new levels because the size of the series becomes too burdensome for recall. Moreover, each new level is essentially a different problem and requires additional exploration. With the partitioning strategy, on the other hand, the focus is on subgoals within the problem rather than on the problem as a whole and the basic method can remain the same; there is no need for a search for a new solution series since the solution for each new level is simply built on smaller subunits.

The children who used the interactive, partitioning approach while programming took advantage of the potential within LOGO for breaking a problem down into manageable bits and getting feedback in the course of constructing procedures. By writing short series of instructions and then checking them out on the screen to see how they worked, the programmer can keep tabs on how the total construction is proceeding; the demands on memory are minimal since each chunk of instructions can be tried out and presumably corrected as the construction proceeded. The start-to-finish, encompassing approach, on the other hand, requires that the programmer be able to think through the entire procedure in order to ensure that it will have the intended effect; memory resources become needlessly over-extended.

Thus, the partitioning strategy, because of the form of representation that it involves, is a strategy which more nearly matches the structure of the puzzle and the programming language and effectively uses that structure to conserve memory resources. In these respects it resembles the superior forms of encoding discussed above.

The transformations:

Transformations of strategies in these tasks involve processes of search and information gathering evolving into predominant use of one of the two main strategies in conjunction with further exploration. Changes from the first and second trials of the four disk Tower of Hanoi (see Appendix A) can be used to form hypotheses about these transformations; on the basis of these changes four different strategy transformations can be identified and related to transformations within the programming task:

(1) Exploration decreases and the use of the encompassing strategy

increases (girls):

For the indicator task, while girls 2 and 3 were perhaps the most exploratory girls, based on the number of moves they took in the first trial of the four disk problem, all the girls seemed to have moved from an exploratory to an encompassing strategy with increases in subproblem solutions at the two disk level. For the target task they all pursued an encompassing, start-to-finish strategy. Girls 1 and 2 seemed to have been the most intent on producing complex, coherent forms. Neither was particulary experimental or interactive in her approach. On the other hand, girls 3 and 4 experimented with commands by producing several programs.

(2) Exploration decreases with an increase in the use of the partitioning strategy (boys 2 and 3):

For the indicator task, two of the boys (2 and 3) moved from an exploratory strategy to a partitioning strategy. On the second administration of the four disk puzzle, of the boys these two made the fewest moves and solved the highest percentage of subproblems at the n-1 level, seemingly pursuing a partitioning strategy which was approaching the level of the problem as a whole. While boy 2 seems to have pursued a partitioning strategy to some degree from the outset, boy 3 seems to have progressed to a partitioning strategy from what was initially a more exploratory approach.

For the target task, both these boys were very interactive in their approach, demonstrated a command of coherent form and transferred their discoveries from one program to another. While boy 2 was intent upon producing a complex, coherent form, boy 3 was more exploratory in his approach as suggested by the fact that he had to be reminded to program rather than simply to issue commands in immediate mode.

(3) Exploration decreases but the use of the partitioning strategy remains virtually constant (boy 1):

For the indicator task, boy 1 seems to have pursued a basically exploratory strategy with the use of the partitioning strategy remaining fairly constant for both trials. For the target task, he was experimental and interactive in his approach but had difficulty producing coherent forms even though this was his intent, possibly because of his lack of programming skills. Just as he made the least progress in the indicator task, he made the least progress in learning how to program.

(4) Exploration increases with an increase in the use of the partitioning strategy (boy 4):

While he moved toward a partitioning strategy, boy 4 was basically very exploratory; he was the only child who actually increased the number of moves he made on the second trial of the indicator task. For the target task, he was highly experimental and interactive in his approach, although, unlike boy 1, he had a command of coherent form and of the mechanics of programming.

For the indicator task the transformations were from exploration to an increased role for one of the two main strategies. For certain transformations (1 and 2), exploration would seem to have been used to discover a way to solve the problem much more efficiently according to both the scoring systems. For others (3 and 4), exploration continued to play a significant role. For one (3), there was some decrease in exploration but the use of the partitioning strategy stayed constant; for the other (4), there was an increase in the use of both the exploratory and the partitioning strategies.

Although the children's performance on the target task closely conformed to the transformation types identified according to the indicator task, there were

differences which related to the use of exploration.

While exploration seemed to be on the decrease by the time of the second administration of the four disk Tower of Hanoi, it still played a significant role in the programming task. This no doubt reflected the more complex and unfamiliar nature of the target task. Because the children in each of the groups had achieved different levels of expertise in LOGO, there were children in each group who differed in their ability to produce coherent graphic forms. However, this factor of command of coherent form affected the use of exploration by the two groups in different ways.

Only the girls who were not in command of coherent form (girls 2 and 3) were exploratory in the sense that they produced a variety of programs. However, all of the boys tried out different commands and most of them produced more than one program, even those who were in command of coherent form. In fact, it was the boy who was least in command of coherent form (boy 1) who wrote only one program. Thus for those pursuing the encompassing programming strategy, exploration was resorted to seemingly because they did not have command of coherent form, while exploration was intrinsic to the partitioning strategy.

While, given the very different nature of the two tasks, it would not be possible to account for every aspect of the children's programming in terms of their performance on the Tower of Hanoi, there would seem to be sufficient parallels to assert that each child had an identifiable manner which was common to his approach to both tasks. Underlying these different approaches to the tasks were different ways in which the children represented the problems to themselves and managed their memory resources. Those who pursued the partitioning strategy adopted a strategy very much in tune with the structure of the tasks, while those children who pursued the encompassing strategy adopted a

strategy which tended to impose a form of problem solving onto the structure of the tasks which was very demanding of memory resources. Thus, the more frugal children tended to partition the problem into manageable units, while the more spendthrift tended simply to spread their resources over the entire problem.

CONCLUSION

A descriptive framework has been developed for the analysis of beginning programming by children. Programming was situated in the classroom context as a skill much like many of the other skills taught there; those who succeeded in programming were those who generally succeeded in other areas. From an analysis of data from a small group of children, three strategies have been identified and the children's performances in two task environments have been analyzed in terms of these strategies and their transformations. It has been maintained that what underlies the different strategies are different ways in which the children represented the problems, either as extended series or subunits for unit-building, and, as a result of the forms of representation, managed their memory resources.

The fact that a relationship between performance on the puzzle and LOGO programming was found is not surprising in that the puzzle was chosen for this study precisely because it has many properties in common with LOGO. Both the puzzle and the language are similarly structured; the Tower of Hanoi is a hierarchically nested, recursive puzzle which invites solution by partitioning into subproblems and LOGO is a hierarchically structured language which permits interactive, "dialectical" programming. As been argued, there are distinct advantages in using a partitioning strategy in these problem solving tasks precisely because it matches the structure of the tasks.

What was not anticipated was that there would be differences according to sex. Because the study was not carefully controlled and the groups were not matched for other variables, it would be quite risky to generalize beyond this particular group. Moreover, Maccoby and Jacklin's systematic review of the

literature (Maccoby and Jacklin, 1974) suggests that sex differences along these lines are not likely to exist.

The approach taken here, which involves tracking processes and closely analyzing them, is a promising methodology for expanding knowledge of learning and individual differences, particularly for understanding strategies and the transformations of them that take place in the course of problem solving. It also provides opportunities for the educator to monitor problem solving and to devise teaching strategies to help children build on the strategies they are employing and bridge to alternative and more efficient strategies.

BIBLIOGRAPHY

Abelson, Harold (1982a). Apple LOGO, Peterborough, New Hampshire: BYTE/McGraw-Hill.

..... (1982b). "A Beginner's Guide to LOGO." BYTE, vol. 7(8), August, pp. 88-116.

Adelson, Beth (1981). "Problem solving and the development of abstract categories in programming languages". Memory and Cognition, 1981, vol. 9 (4), 422-433.

Anzai, Yuichiro; Simon, Herbert A. (1979). "The Theory of Learning by Doing". Psychological Review, March vol. 86(21), pp. 124-140.

Boersma, Frederic J. and Chapman, James w. (1979). Student's Perception of Ability Scale. Edmonton, Alberta: University of Alberta.

Brown, A., Bransford, J.D., Ferrara, R.A., and Campione, J.C. (1983). "Learning, remembering, and understanding". In J.H. Flavell and E.M. Markman (eds.), Handbook of Child Psychology, Volume III: Cognitive Development. New York: Wiley.

Bruner, Jerome; Goodnow, J.J.; and Austin, G.A. (1956). A Study of Thinking. New York: John Wiley & Sons.

Chase, William G. and Simon, Herbert A. (1973a). "The Mind's Eye in Chess". In William G. Chase (ed.), Visual Information Processing. New York: Academic Press, pp. 215-281.

..... (1973b). "Perception in Chess". Cognitive Psychology, 4. 55, pp. 55-81.

Coombs, M.J., Gibson, R., and Alty, J.L. (1981) "Acquiring a First Computer Language: A Study of Individual Differences". Computing Skills and the User Interface, ed. M.J. Coombs and S.L. Ally. New York: Academic Press, pp. 289-313.

Connolly, Austin J., Nachtman, William, and Pritchett, E. Milo (1979). Key Math Diagnostic Arithmetic Test. Willowdale, Ontario: Psychan.

Gagne, Robert M. and Smith, Ernest C. (1962). "A study of the effects of verbalization on problem solving". Journal of Experimental Psychology, 63, 12-18.

Gagne, Robert M. (1980). "Learnable Aspects of Problem Solving". Educational Psychologist, vol. 15, No. 2, pp. 85-92.

.....(1984). "Learning Outcomes and Their Effects". American Psychologist, vol. 39, no.4, pp. 377-385.

Greeno, J.G. (1978). "A study of problem solving". In R. Glaser (ed.), Advances in instructional psychology (vol. 1, pp. 13-75). Hillsdale, N.J.: Erlbaum.

Harvey, Brian (1982). "Why LOGO?" BYTE, vol. 7(8), August, pp. 163-195.

Karat, John (1982). "A Model of Problem Solving with Incomplete Constraint Knowledge". Cognitive Psychology, Oct., Vol. 14(4), pp. 538-559.

Klahr, David and Robinson, Mitchell (1981). "Formal Assessment of Problem-Solving and Planning Processes in Preschool Children". Cognitive Psychology, Jan. Vol. 13(1), pp. 113-148.

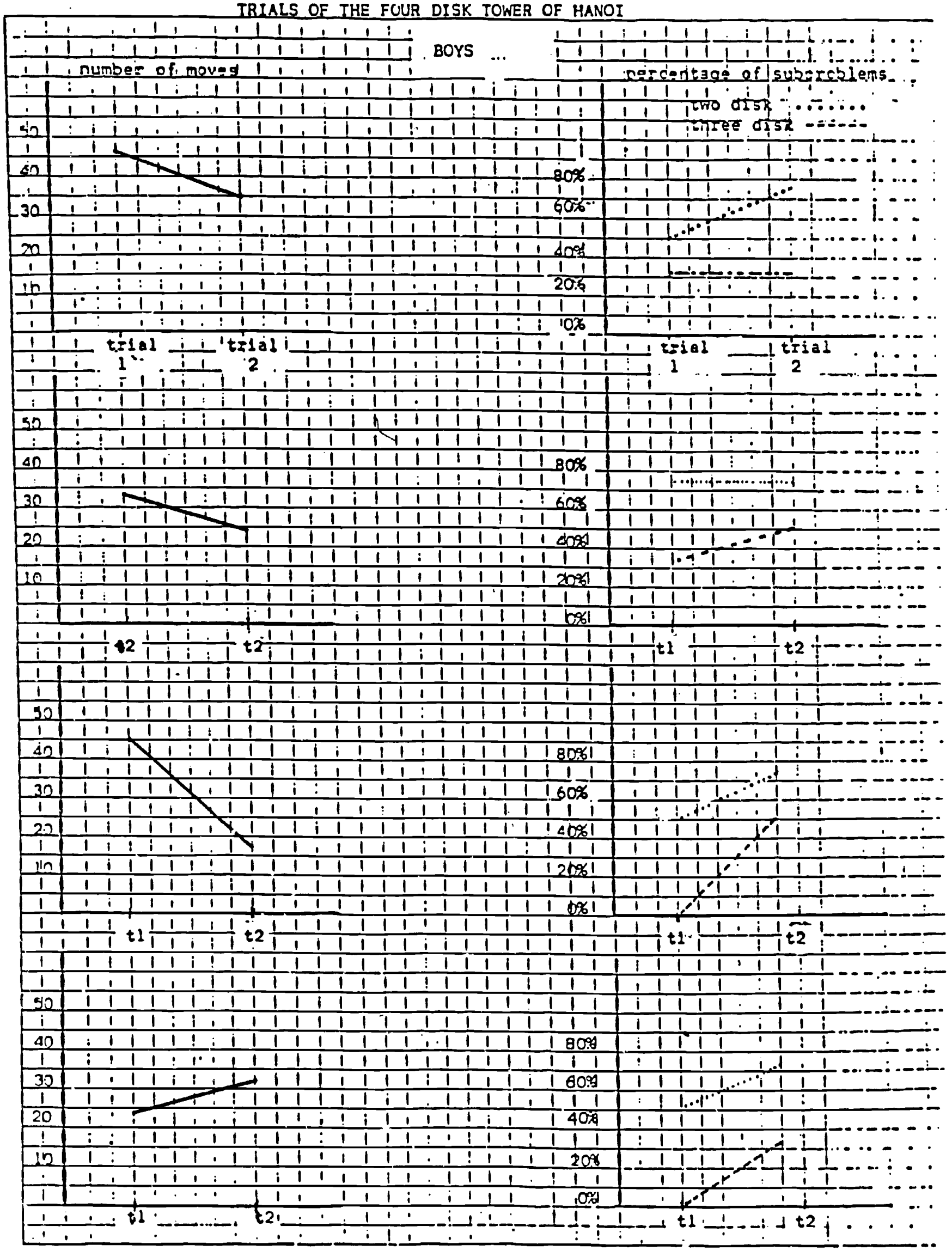
Luger, George F. and Steen, Michael (1981). "Using the State Space to Record the Behavioural Effects of Symmetry in the Tower of Hanoi Problem and an Isomorph." International Journal of Man-Machine Studies. May, Vol 14(4), pp. 449-460.

Luger, George F. (1976). "The Use of the State Space to Record the Behavioural

Effects of Subproblems and Symmetries in the Tower of Hanoi Problem". International Journal of Man-Machine Studies, Jul. Vol 8(4), pp. 411-421.
Maccoby, E.E. and Jacklin, Carol Nagy. The Psychology of Sex Differences. Stanford, Calif.: Stanford University Press, 1974.
McKenny, James and Haskins, Ron (1980). "Cognitive Training and the Development of Problem-Solving Strategies". Exceptional Educational Quarterly, pp. 41-51.
Neches, R. and Hayes, J.R. (1978). "Progress towards a taxonomy of strategy and transformations. In A.M. Lesgold, J.W. Pelligrino, S. Fokkema, and R. Glasser (eds.). Cognitive psychology and instruction. New York: Plenum.
Newell, Alan and Simon, H.A. (1972). Human Problem Solving. Englewood Cliffs, N.J.: Prentice Hall.
Nilsson, Nils J. (1971). Problem-Solving Methods in Artificial Intelligence. New York: McGraw-Hill, 1971.
Papert, S. (1980). Mindstorms. New York: Basic Books.
Piaget, Jean (1976). The Grasp of Consciousness. Cambridge, Massachusetts: Harvard University Press.
Scardamalia, Marlene and Bereiter, Carl (in press), "Development of dialectical processes in composition. in D. Olson, N. Torrance and A. Hildyard (eds.0, Literacy, language and learning: the Nature and Consequence of reading and writing. Cambridge University Press.
Shneiderman, B (1976). "Exploratory experiments in programmer behavior". International Journal of Computer and Information Sciences, 5, 123-354.
Siegler, R.S. (in press). "Encoding and the development of problem solving." In S.S. Chipman, J.W. Segal, and R. Glaser (eds.), Thinking and learning skills: Current research and open questions (vol. 2). Hillsdale, N.J.: Lawrence Erlbaum Associates.
..... (1976). "Three aspects of cognitive development". Cognitive Psychology, 8, pp. 481-520.
Simon, Herbert A. (1976). "Identifying basic abilities underlying intelligent performance of complex tasks". in J. Resnick (ed.), The Nature of Intelligence. New York: Harper and Row, pp. 65-97.
..... (1979). "The Functional Equivalence of Problem Solving Skills." Models of Thought. Yale University Press, New Haven, pp. 230-244.
Wood, David J. (1978). "Problem Solving-- The Nature and Development of Strategies" in Geoffrey Underwood (ed.), Strategies of Information Processing, New York: Academic Press, pp. 329-356.

APPENDIX A

TRIALS OF THE FOUR DISK TOWER OF HANOI



APPENDIX A
TRIALS OF THE FOUR DISK TOWER