DOCUMENT RESUME

ED 349 953 IR 015 689

AUTHOR	Ippel, Martin J.
TITLE	Discovery Learning of Addition Strategies for Two-Digit Numbers in a Computer-Based Microworld.
PUB DATE	Apr 92
NOTE	17p.; Paper presented at the Annual Conference of the American Educational Research Association (San Francisco, April 20-24, 1992). This study was partly made possible by a grant from the Iowa Measurement Research Foundation.
PUB TYPE	Reports - Research/Technical (143) -- Speeches/Conference Papers (150)
EDRS PRICE	MF01/PC01 Plus Postage.
DESCRIPTORS	Arithmetic; *Cognitive Processes; *Computer Assisted Instruction; *Discovery Learning; *Elementary School Mathematics; Foreign Countries; Grade 2; *Instructional Design; Microcomputers; Multivariate Analysis; Primary Education; Thinking Skills
IDENTIFIERS	Dienes Multibase Arithmetic Blocks; Message Design; *Microworlds; Netherlands; Task Environments

ABSTRACT

This paper presents a framework for understanding conditions of discovery learning in computer-based microworlds. It begins with a short discussion of problems related to a traditional type of microworld--i.e., learning tools for mathematics--using the Dienes Multibase Arithmetic blocks as an example. In this discussion, hypotheses are developed about characteristics of task environments that seem to be necessary for discovery learning. The design of a microworld in which students are challenged to acquire algorithms for mental addition of two-digit numbers is then proposed, in a discussion that advances two general principles for the design of computational microworlds in which procedural skills can be acquired by discovery learning and that identifies two conditions that can be expected to foster problem solving behavior. The project in which the 10-square microworld was designed is then described, together with a first experiment conducted to determine whether second grade students would be able to acquire the target algorithms without the help of a teacher; the problem solving methods they utilized in reaching the goal are also described. It is noted that a prototype of the 10-square microworld implemented for a Macintosh SE/30 computer system was used in the experiment, which consisted of a sequence of five microworlds with different sets of constraints and a sixth microworld without constraints. Analyses of the study data indicate that continuing experience with the task turned the process of solving addition problems into a routine action, and that the 10-square microworld was effective in narrowing down the number of student choices. The desirability of further study of the transfer from the 10-square model to mental problem solving is indicated. Eight figures are provided. (Contains 16 references.) (ALF)

DISCOVERY LEARNING OF ADDITION STRATEGIES FOR TWO-DIGIT NUMBERS IN A COMPUTER-BASED MICROWORLD

Martin J. Ippel

Center for the Study of Education and Instruction
Leiden University

The Netherlands

Paper to be presented
at the
A.E.R.A. Annual Conference
April 20 - 24 1992
San Francisco

This study was partly made possible by a grant
from the
Iowa Measurement Research Foundation
The University of Iowa

INTRODUCTION

For most students it takes only one or two years of regular training in elementary arithmetic to be able to retrieve the basic addition facts from memory. Basic addition facts refer to, for example, problems of the form $m + n = s$, with the whole numbers m and n less than 10. There is some evidence that the memory representation of these facts is partly based upon the perceived magnitudes of numbers (Resnick, 1989; Beem, 1990; Beem & Ippel, 1990).

To manipulate larger numbers, say numbers greater than 20, humans utilize a so-called base-10 system, a conceptual system that defines the meaning of numbers in terms of their interrelationships, rather than the quantities they refer to. The teaching of mathematical structures, like the base-10 system, poses an intriguing problem to education since these structures are not obvious in children's everyday environments. One way to solve this instructional problem is the design of teaching materials that embody these structures and make them more concrete (e.g. the Dienes Multibase Arithmetic Blocks).

Work in this line by math educators such as Dienes tries to challenge children's capacity for inquiry and invention (Resnick & Ford, 1981). As it turns out, though, teaching tools for mathematics like Dienes blocks do not provide for a stand-alone learning environment. A teacher is needed to introduce and maintain the rules according to which the blocks system can be manipulated.

In general, research on discovery learning in the 1960s casted doubt as to whether this mode of instruction could produce efficious learning outcomes. However, the introduction of computers at schools during the 1980s has renewed the quest for the design of instructional environments in which students can learn by independent exploration. The LOGO Turtle world is one prototypical instance of such an environment - a microworld for exploring spatial and mathematical concepts (Papert, 1980; Abelson & diSessa, 1980). As an environment for exploratory learning the LOGO Turtle world has not met with universal success. In fact, it can be concluded that the introduction of the new technology did not come with a new understanding of what conditions empower students to learn by independent exploration.

In this paper we present a framework for understanding conditions of discovery learning in computer-based microworlds. We will start with a short discussion of problems related to a traditional type of microworld - learning tools for mathematics. The Dienes Multibase Arithmetic blocks will be used as an example. In this discussion hypotheses will be developed about characteristics of task environments that seem to be necessary for discovery learning. Subsequently, we will discuss the design of a microworld in which students are challenged to acquire algorithms for mental addition of two-digit numbers. Finally, an experiment will be reported that was conducted to examine the utility of the proposed framework.

TRADITIONAL TEACHING TOOLS FOR MATHEMATICS

The Dienes Multibase Arithmetic Blocks

Groen (1984) defines a microworld as a structure, i.e. a set of states and transformations between states, with certain additional properties. The most important of which are: (1) there should exist mappings to other structures that are representations of concrete actions in the real world, (2) a transformation can be undone to go back to the previous state. If we accept this prescription, it can be argued that certain traditional (i.e. non computer-based) teaching tools for mathematics can be viewed as microworlds. As an example of such a tool the so-called Dienes Multibase Arithmetic blocks will be discussed. The Dienes blocks system is a concrete model designed to provide meaning to base systems, and the associated place-value system. In such models the properties of the conceptual system are made concrete through mapping on a physical medium the possible states of the system and the operations to transform one state into another. In this blocks system the abstract notions $(10)^0$ to $(10)^3$ are made concrete by using wooden blocks of different forms and sizes. Figure 1 shows one state of the Dienes block system. Changing the state, i.e. addition or subtraction, can be realized by adding, removing, and/or changing blocks according to certain rules.

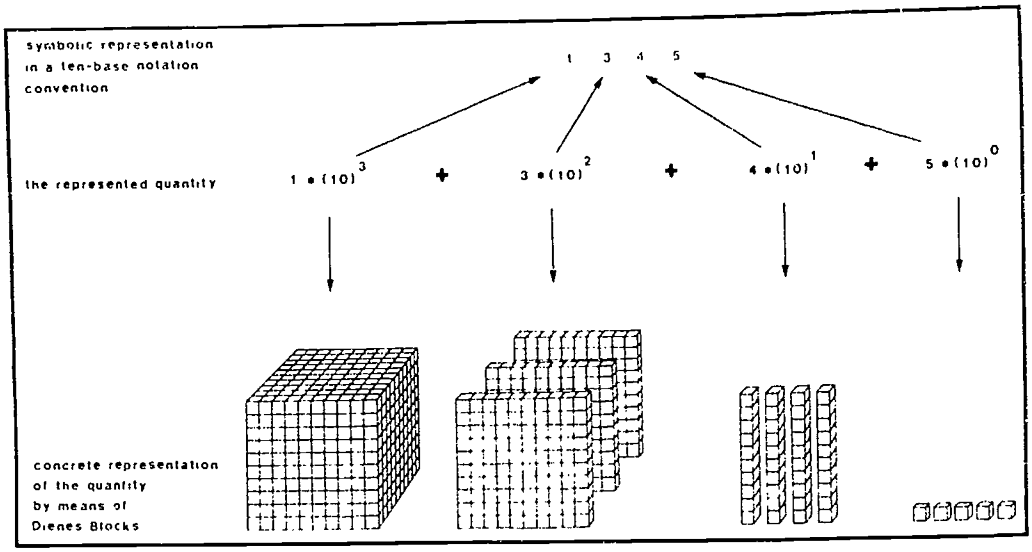


Figure 1. A system state of the Dienes Multibase Arithmetic Blocks

The question that I would like to investigate is why can the properties of a teaching tool for mathematics not be discovered without the help of a teacher?

Weak representations

I would like to start this investigation by pointing at a shortcoming that seems to be fairly common to many instructional tools, and is very sailiently present in the Dienes blocks system.

The Dienes blocks system is designed as a model of the base-10 system (and of several other related base systems). This implies a one-to-one mapping between elements of the model and the system being modeled. However, it can be easily shown that this condition of isomorphism is violated. This is caused by the fact that the Dienes blocks system does not provide for a canonical representation of the numbers. For example, the number <1345> represented correctly in Figure 1, actually can be represented in several other (but syntactically incorrect) ways, such as using 13 flats to represent the first two digits. To define the admissible states of the Dienes blocks system, rules have to be imposed related to:

a. The maximum number of units, longs, flats, and cubes that may be used to encode a number;

b. the mutual spatial relationships among units, longs, flats, and cubes;

In fact, the Dienes blocks system can be in many more different states than the states that correctly represent the states of the system being modeled.

TASK ENVIRONMENT AND PROBLEM SPACE

To get a better idea of what these rules do to the Dienes blocks system, I now will introduce an important conceptual distinction derived from cognitive science, i.e. the distinction between task environment and problem space (Newell & Simon, 1972; Newell, 1980; Simon & Kaplan, 1990).

a. Task environment. Newell and Simon (1972) define a task environment as an environment coupled with a goal, or problem, or task. The Dienes blocks system consists of sets of blocks of various forms and sizes. These forms and sizes relate in a systematic fashion to the quantities they represent (see Figure 1). As such the Dienes blocks system provides for an environment with which young children would love to play. For example, they can use the blocks to build houses, castles, or bridges. The environment allows for all these outcomes. Once this environment is coupled to a particular goal, say, to be used for building a house, or to compute the sum of e.g. 234 + 457, certain rules apply which are not part of the environment, but are imposed by the goal. These rules constrain the way in which a child is allowed to manipulate the blocks, and differ according to the goal to be obtained. It is through these rules that a neutral environment turns into a task environment. Also, these rules divide the total set of possible states and operations of the system into a class af admissible states and operations, and a class of non-admissible states and operations.

b. Problem Space. Cognitive science regards humans as symbol manipulating systems. Cognition is supposed to imply operations on a symbolic representation of the task environment, rather than on the task environment itself. A currently popular paradigm for characterizing the internal representation of the task environment is the state space or problem space. A problem space is the internal representation of a task environment consisting of a set of admissible symbol structures or problem states, including an initial state and a goal state, as well as a set of admissible operators to transfer one state into another. For example, a base-10 representation of the numbers <1> through <100> would require one hunderd different symbol structures, i.e. numbers, and a set of operators, which are essentially based on the difference-1, and difference-10 relations between the numbers.

From this it can be concluded that the Dienes blocks system constitutes a task environment that can generate a much larger set of possible configurations and operations than can be considered admissible configurations and operations within a problem space for addition and subtraction. Instruction is needed to specify whether a specific block configuration can be considered as an admissible configuration. Therefore, an important function of instruction is - to introduce and maintain of the rules that constrain the number of possible states and operators.

DESIGN PRINCIPLES

I now would like to propose a few general principles for the design of computational microworlds in which procedural skills can be acquired by discovery learning. These principles are based on Newell's (1981) problem space hypothesis, which asserts that skilled, routine behavior is organized within a problem space by accumulation of search control knowledge. As for self-discovery environments Newell's hypothesis seems to imply the following principles for the design of a task environment or learning environment:

Principle 1:
A task environment should allow for no more states and operators than are admissible in a problem space in which a particular skill can be developed.

This principle directly follows from the distinction between real world task environments and problem spaces. Information technology has great potential for implementation of this principle.

If no teacher is supposed to be present, the student has only one way to acquire knowledge about the task environment: using problem solving. This idea is incorporated in the following principle:

Principle 2:
A task environment should allow students to acquire the correct structure of required task behavior by just applying weak search heuristics or general problem solving techniques.

The choice for a problem space representation of a task implies a commitment to a search modeling of cognitive processes. Problem solving often relies on partial knowledge of the task domain, or 'rules of thumb' which constrain the search space and the numbers of candidate states at any state in the space. Methods that rely on partial knowledge about a particular task domain to solve problems are called heuristic search techniques. Although these methods have shown up first in AI investigations, they seem to provide for a natural description of the human problem solving behavior as well. There is a growing evidence that even young children utilize such problem solving methods (e.g. Byrnes & Spitz, 1979; Borys, Spitz, & Dorans, 1982; Klahr, 1985).

Thusfar, we were able to identify two conditions that can be expected to foster problem solving behavior

Condition 1: Provide for a mental model.

A condition that might increase the efficiency of the problem solving process involves the transparancy of the task environment. In general this requires task environments to allow the student to utilize knowledge she has gained through previous experience to support the partial knowledge of the problem domain. An already familiar situation often serves as a model to interpret the new task environment.

Condition 2: New knowledge can be acquired by making previously transparant solutions problematic.

Here we propose the hypothesis that the transformation of more initial forms of procedural knowledge into more advanced forms should be triggered by introducing constraints on earlier established behavior patterns (see Holland et al., 1986).

THE TEN-SQUARE MICROWORLD

To illustrate the application of the proposed principles for the design of a microworld, I will discuss a project in which we designed a microworld for the acquisition of algorithms for mental addition of two-digit numbers by discovery learning.

Goal of the environment. Figure 2 shows the surface structure of the microworld that is being designed. It is a so-called ten-square, a spatial arrangement of 10 * 10 positions each representing a number from <1> on top left to <100> at the bottom right. This ten-square has a pawn on it, an object with which the student can move to different positions. Each of these positions corresponds with a number. In fact, this ten-square microworld comprises an ordered set of related data structures, each corresponding with a particular problem space corresponding with increasingly more efficient versions of the target algorithm. The microworld intends to challenge children to discover step-by-step an highly efficient ('expert') algorithm for mental addition with two-digit numbers.

Mental addition and subtraction problems of the form xx ± yy = .., and xx ± .. = zz induce a heavy load on children's cognitive resources. It is therefore important that they learn to use calculation methods that provide short ways to a solution.

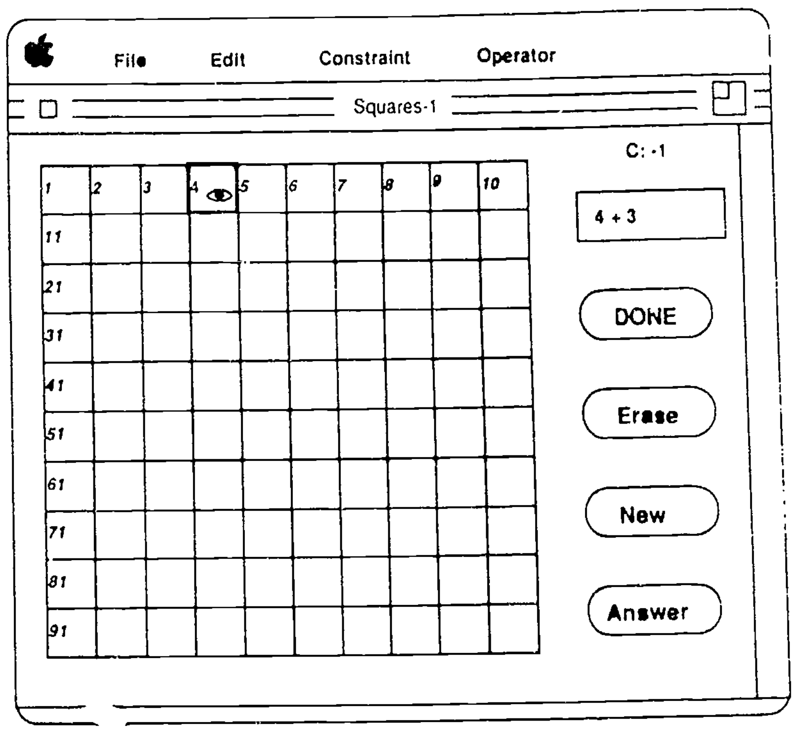


FIGURE: A screen as generated by the system when the user just starts the microworld

Figure 2. The computer implementation of the ten-square

The algorithms to be learned share the feature that in addition or subtraction of two numbers the first term is retained as a whole, whereas the second term is decomposed into tens and units. For example, 53 + 22 = (53 + 20) + 2; An advanced version of the algorithm involves a more flexible use of the properties of the base-10 system. For example, **not:** 57 + 18 = (57 + 10) + 8; **but:** 57 + 18 = (57 + 20) - 2. The ultimate goal of the environment is to induce an adaptive switching between the algorithms such that the shorter way to a solution will be followed.

Principle 1: a constrained microworld

Conceptual constraints. The ten-square microworld provides a spatial representation of the base-10 system for the numbers one to one hundred. Its structure emphasizes some fundamental relationships between numbers by means of spatial adjacency: horizontal adjacent refers to difference-1 relations, vertical adjacency refers to difference-10 relations. A correct addition (or subtraction) procedure utilizes these relations. Accordingly, the student should move the pawn only in horizontal or vertical directions. Violation of this principle, for example by a diagonal move, would lead to an incorrect procedure. A central idea in the design of this world is to implement the properties of the base-10 system as (conceptual) constraints on the moves of the pawn. The task of the student is to identify the constraints that are imposed on the movements of the pawn, and to search for solution paths that satisfy these constraints. It is expected that this will turn the learning of a calculation procedure into a problem solving task.

Procedural constraints. Notice that merely using these basic characteristics of the base-10 system, i.e. the difference-1 and difference-10 relations will lead to a step-by-step solution, very unlike the efficient two-step strategy of an 'expert'. In other words, the 'expert' strategy satisfies more constraints, which are not directly implied by the base-10 system. Identification of the latter type of constraints is of equal importance for the construction of the microworld as is the former type of constraint.[1]

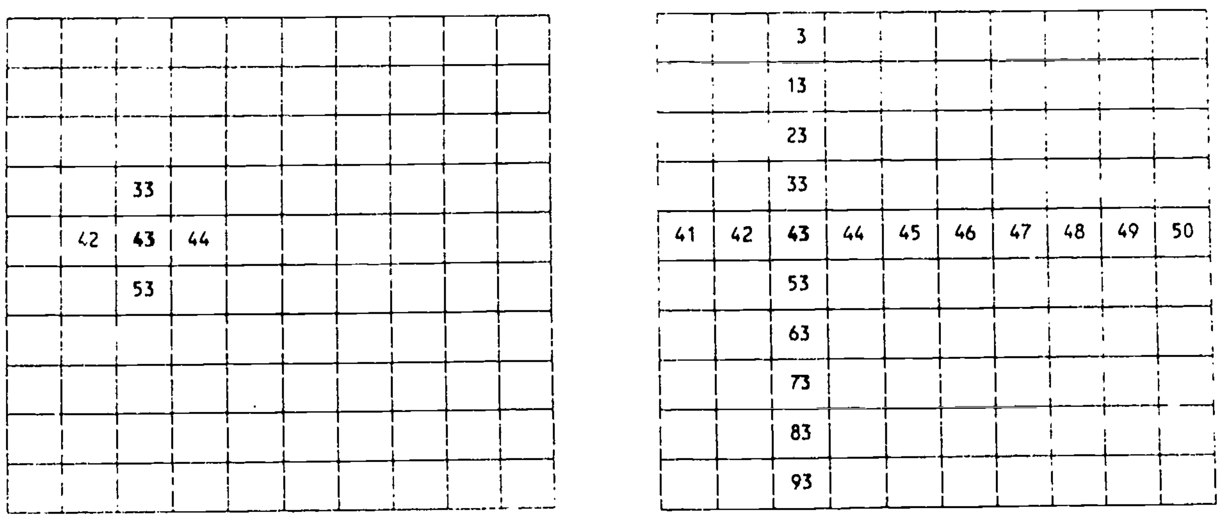


Figure 3. The immediate successor states of position <43> in two different problem spaces: (a) the initial microworld, i.e. the immediate successor realtions based on spatial adjacancy; (b) in a more advanced world, i.e. every position in a particular row or column.

Principle 2: trigger problem solving

The ten-square provides students with a spatial metaphor of the base-10 system for the numbers <1> through <100>. Most people - including children - reason rather fluently about motions through space. The strength of the ten-square as a metaphor is that it provides students with a spatial logic for choosing the pawn's next position. In this sense the ten-square is supposed to function as a mental model which makes the task environment more transparant to the student. It provides students with a general sense of direction as to where the goal state can be found.

---

1 In order to identify the constraints for this ten-square microworld a cognitive simulation modeling technique has been used (Ippel & Meulemans, 1990; Ippel, Kemmere, & Meulemans, 1989).

Also, the task of searching for a path to reach this goal state may activate previously acquired knowledge about path finding in other spatial situations. It is the type of information that can be utilized in very simple weak search methods, like hill-climbing (e.g. Rich, 1983). The hill-climbing method explores a set of actions that are possible at a given point. then chooses the best, based on some numerical evaluation function. Following the selection of the best alternative, the method recurs, once again trying out new alternatives and selecting the best. In short, hill climbing is a search heuristic that comes down to 'do whatever seems the best at the time'.

However, hill-climbing based on application of the spatial metaphor leads only to step-by-step problem solving behavior similar to that of a novice. In order to acquire the 'expert' two step procedure, the student should come to invent operators other than just <move-1>, or <move-10>, resp. horizontally adjacent, or vertically adjacent. Figure 3 shows what the student has to invent. In the terminology of the search paradigm the student has to learn that not only the spatial adjacent positions are immediate successor states (i.e. can be reached by application of one single operator), but every position in the same row or column. Card, Moran, & Newell (1981) mention the construction of new operators (which effectively reduce the problem space to be searched) as one of the mechanisms by which problem solving evolves into cognitive skill.

In this learning environment the invention of new operators is triggered by creating an impasse in a routine solution process. This is done by increasingly constraining on the movements of the pawn. As a result a previously transparent solution becomes a problem again, which is expected to evoked new problem solving behavior. This process of increasingly imposing constraints on the solution path is implemented as a series of related microworlds, each corresponding with a problem space in which the solution path is further constrained. Thus the microworlds have identical interfaces, which represent the numbers of <1> through <100>, but they differ with respect to the set of immediate successor states, given a particular problem (see Figure 4).

A FIRST EXPERIMENT WITH THE TEN-SQUARE MICROWORLD

Now some results of a first experiment with this microworld will be reported. In this study we were primarily interested in two questions. First, will students indeed be able to acquire the target algorithms without the help of a teacher. Second, what problem solving methods do students utilize to find a path to the goal state.

Subjects

Twenty second grade students participated in the experiment (age mean = 8 years and 5 months, s.d. = 6 months). The students had not yet been exposed to the usual columnwise addition and subtraction as practiced in U.S. elementary schools.

Method

In this experiment a prototype of the ten-square microworld was used. It is implemented for a Macintosh SE/30 computer system using Think C as programming language. It consists of a sequence of five microworlds, each comprising different sets of constraints. A sixth microworld did not have any constraints implemented.

All students first received two subsequent tests in which they were asked to locate numbers either in a particular quadrant of the ten-square (global test), or to point at the precise location of a number (detailed test). The students were required to use a mouse to point at these positions. Subsequently, all subjects were asked to solve some simple sums on the ten-square. For example, 5 + 3 =.. , or: 7 + 6 = ..;

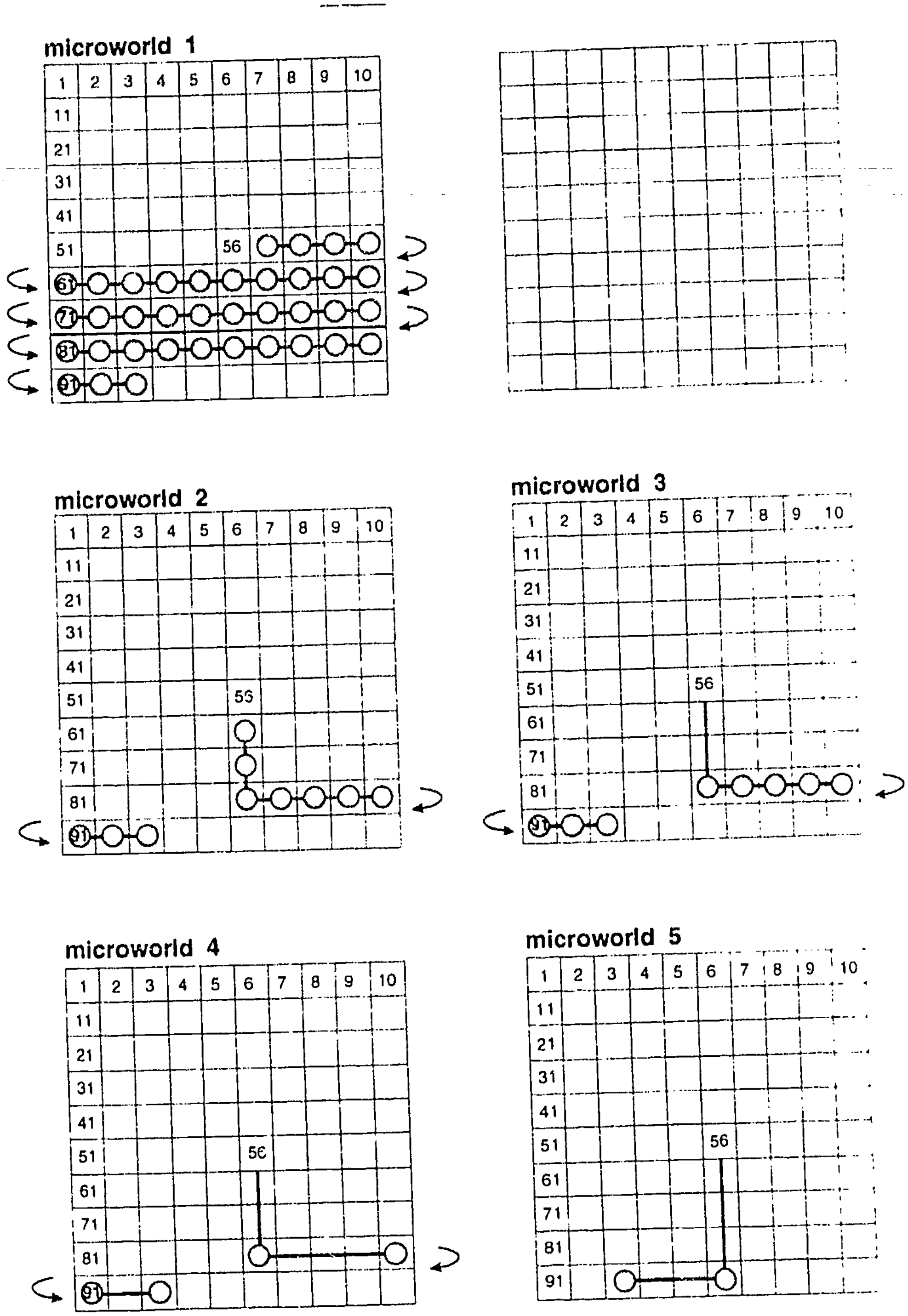


Figure 4. Different solution paths as a consequence of adding constraints to the problem space. The problem is: 56 + 37.

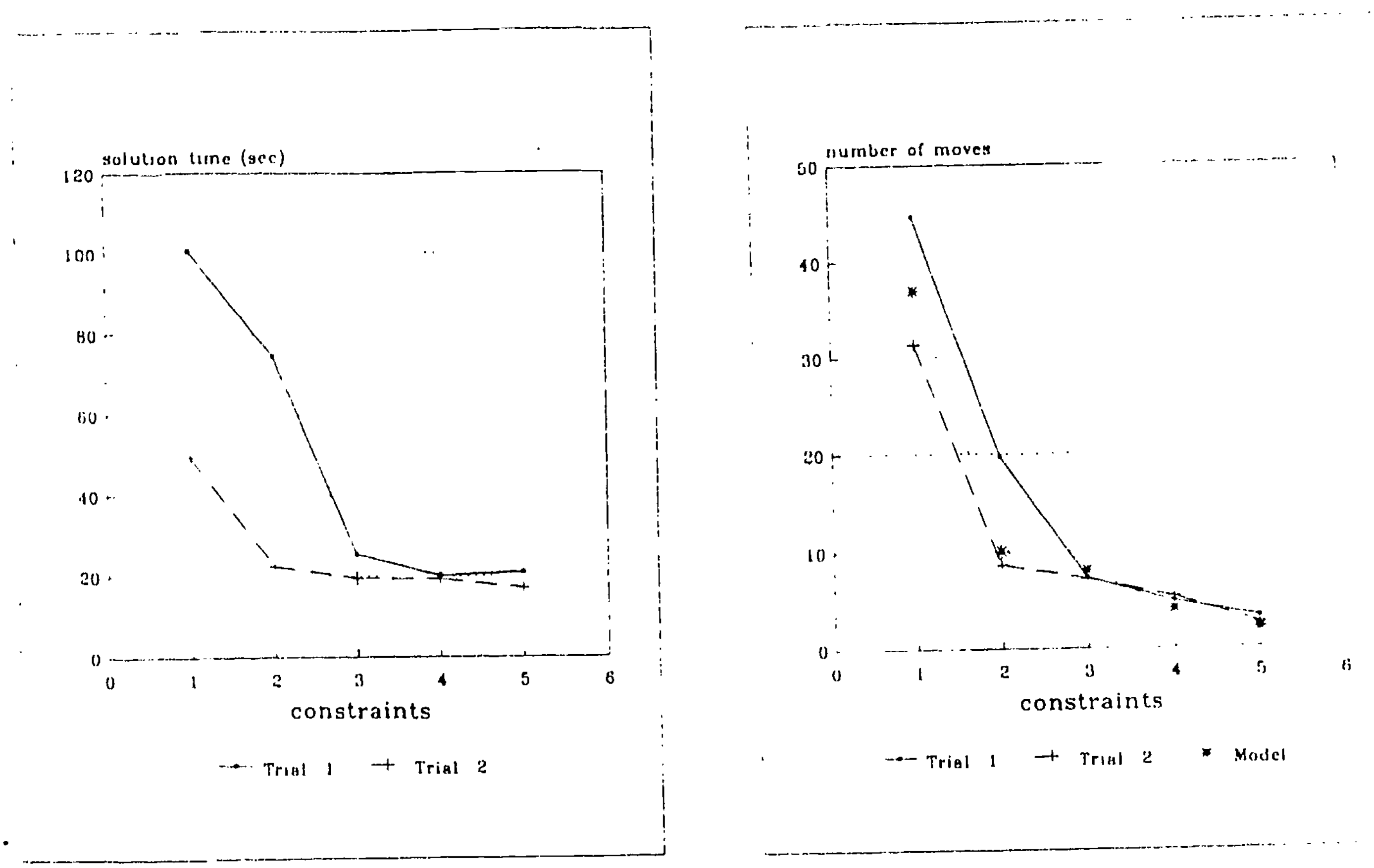
The subjects were divided into two groups. Sixteen students were assigned to the experimental group (E-group), and four students were assigned to the control group (C-group). One of the E-students did not participate in the last three sessions of the experiment. In five sessions the E-students worked through a standard set of problems in each of the microworlds.

Each session took about half an hour. In every session the students were informed that 'something' had changed in the microworld, and that it was their task to find out what it is and how to deal with it while solving the addition problems. Each time a problem was completed successfully the subjects were required to read the problem aloud and mention the answer just found.

The sessions took place every other day. Three days after the fifth session the students were tested in a sixth session in an unconstrained microworld. The C-students worked themselves through the same set of problems, but they always worked in the non-constrained version of the microworld. Their results will not be discussed further.

Results

Let us jump to the end of the story to discuss the evidence related to the question: is this type of microworld effective at all. Do students acquire the target algorithm in every microworld and can their behavior gradually be shaped to the ultimate 'expert' performance? In the following section on the effectiveness of the microworld data is provided that supports the claim of a qualitative change in calculational behavior without teacher intervention.



Figures 5a and 5b.
Mean solution time in seconds and number of moves as function of the number of constraints of the microworlds.

Evidence for improved effectiveness. We compared the performances of students on a particular problem that was given as 15th and final (30th) problem in every session: **56 + 37 = ..;** As Figure 4 shows, in every following microworld the solution path for the problem is slightly different because these solution paths are increasingly more constrained.

The experimental data were analyzed following a doubly multivariate repeated measures design with CONSTRAINTS, i.e. the set of five microworlds that differ in the number of constraints imposed on the possible moves of the pawn. and TRIAL, i.e. the 15th versus the 30th problem of each session, as within subject factors. For each combination two dependent variables were measured - solution time and number of moves. The analyses were conducted on logarithmic transformations of the dependent variables.

The multivariate test for the CONSTRAINT effect was significant for both dependent measures (Pillais T = .97831; df = 8,7; $p<.01$). Inspection of the univariate results (Helmert contrasts) suggested that for solution time this significancy exclusively originated from the difference between microworld 1 and 2 versus the succeeding microworlds. There was no significant decrease in solution time beyond the second microworld. The multivariate test for TRIALS also was significant for both dependent measures (Pillais T =.73619; df =2, 13).

The multivariate test for the interaction effect CONSTRAINTS x TRIAL on both measures were marginally significant (Pillais T = .81212; df = 8, 7; $p<.05$). As Mauchly's test of sphericity indicated that the covariance matrix of the transformed variables could not be considered a scalar matrix, we corrected the degrees of freedom for the corresponding univariate F-test. It showed that the interaction effect was significant at .05 level for solution times (F = 4.6; df(corrected) = 1, 20; $p<.05$) but not for number of moves (F = 2.67; df(corrected) = 1, 20; n.s.).

A repeated measures MANOVA on mean time per move with CONSTRAINTS and TRIAL as within-subject factors showed only a significant main effect for CONSTRAINTS

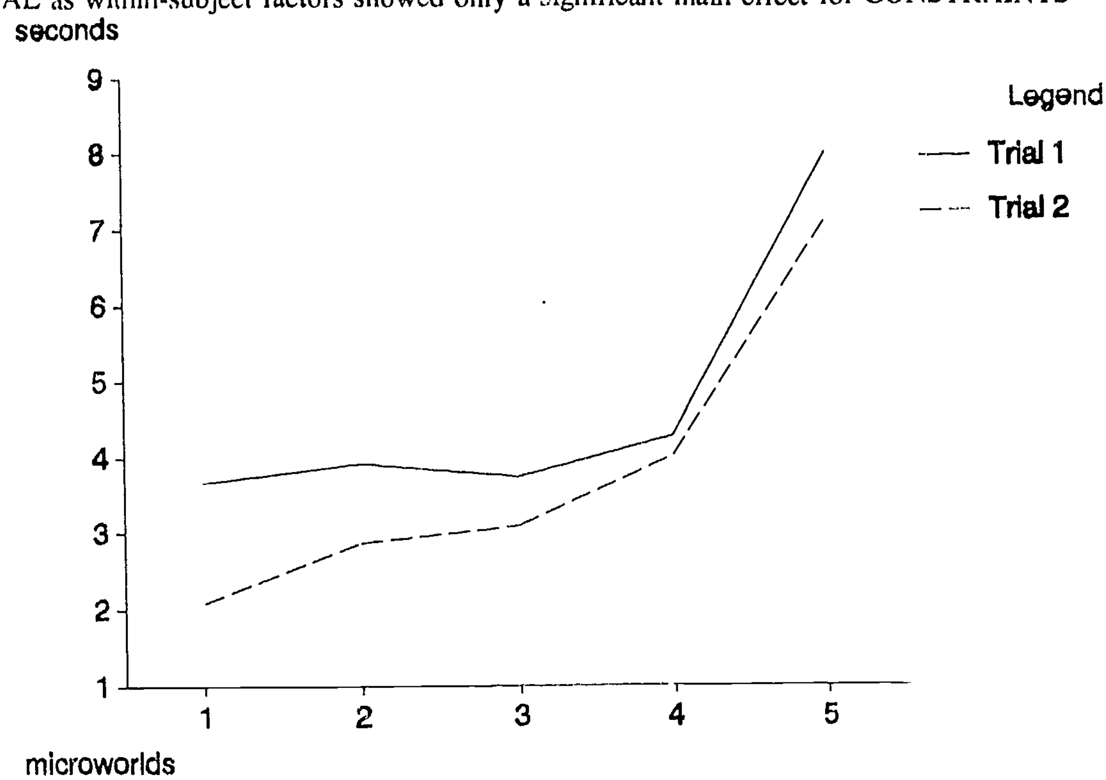


Figure 6. Mean time per move in seconds as function of the number of constraints and experience.

(Pillais T = .56956; df = 11, 4; p<.05). Inspection of the univariate inverse Helmert contrasts indicated that this significancy was due to the difference between microworld 5 and the preceding microworlds (see Figure 6). No significant reduction in mean time per move as a function of increased exposure could be detected, i.e. the factor TRIAL was not significant. Nor was the interaction CONSTRAINTS x TRIAL. This suggests that the decrease in solution times related to the factor TRIAL in the previous analysis is mainly due to a reduction in number of moves.

As Figure 5.b shows, at the final problem of every session the actual number of moves as performed by the students closely follows the number of moves required given the addition problem at hand and the current set of constraints. In order to investigate whether students indeed succeeded in acquiring a strategy that matches the constraints of the microworld they were exposed to during a particular session, a multivariate repeated measures analysis was conducted on a difference score composed of the actual number of moves minus the number of moves given the set of constraints in a particular microworld. The constant effect was not significant (F = 1.06; df = 1, 14; n.s.) indicating that the overall discrepancy between the two scores was zero. However, the CONSTRAINT effect was significant (Pillais T = .72693; df = 11, 4; p<.01). Inspection of the cell means indicated that students used less moves in the first three microworlds than would be allowed, and slightly more moves in microworld 4 and 5.

Retention. The performances of the students did not differ significantly between the fifth microworld and the unconstrained microworld. This result holds for solution time as well as number of moves (Pillais T = .16227; df = 12, 2; n.s.). A qualitative analysis of their performances showed that 7 students (43.75%) solved the problem in a way similar to the calculational strategy corresponding with microworld 4, 5 students (31.25%) utilized the microworld 5 strategy, and 4 students (25%) choose a one-move solution. They apparently relied on mental calculation.

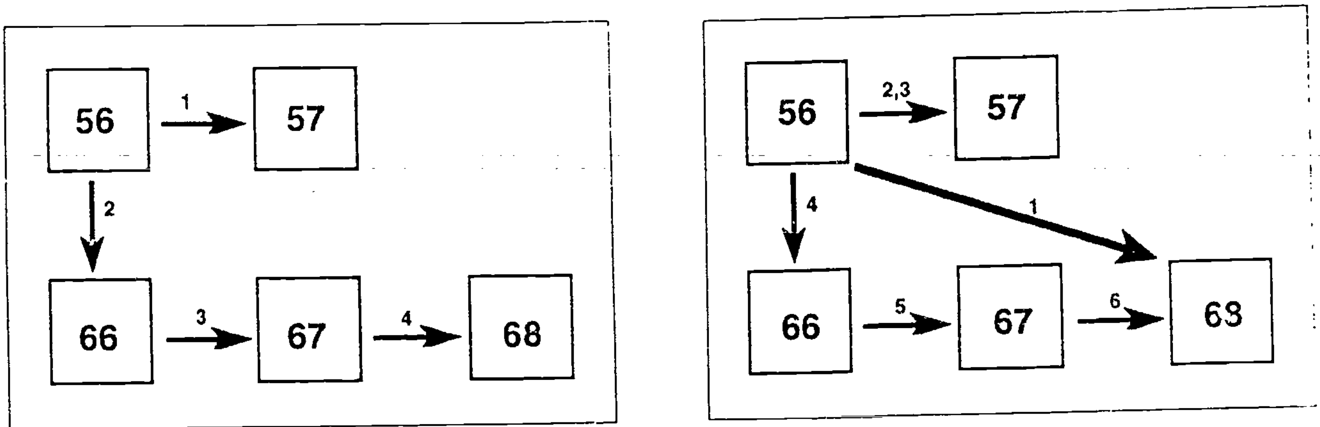
Evidence for weak search methods. At the end of a session, some degree of proficiency in dealing with the task environment can be expected. To figure out how students gradually acquire this proficiency we should look at the first confrontations with a new problem. Is it indeed possible to model student's behavior in terms of weak search methods?

In this section we will discuss the performances of three students, who for the first time were confronted with a newly imposed constraint in microworld 2. The students can be considered representative for the variety of performances in the sample. The constraint implies the following rule:

IF (goal is adding two numbers & second addend > 9)
THEN (first use a difference-10 operator)

Consider the problem: **56 + 12 = ..**, which is a relatively easy problem that many students were able to solve by mental calculation. In microworld 1 this problem could be solved by using a <move-1> operator, i.e. moving the pawn step-by-step towards the goal state.

As Figures 7 and 8 suggest two choices are very prominent at the beginning of the problem solving process. First, <move-1>, i.e. move the pawn from position 56 to 57. This is a rule acquired in microworld 1, but is no longer an admissible operator given the second addend. The second prominent alternative is to bring the pawn directly to the goal state, i.e. position 68. This solution path also does not satisfy the newly imposed constraint. In fact, during the first two trials students tried 11 times (34% of the total number of trials) to move the pawn from position 56 to position 57. The goal state was the target of an intended move 9 times (28% of the total number of trials).



S Ø8

S Ø4

PROBLEM: 56 + 12 =

Figuur 7. Problem solving graphs of the subjects S04 and S08 at the first confrontation of a newly imposed constraint in microworld 2.

The subjects S04 and S08 represent a very rational behavior. During their first couple of moves they either checked the old microworld 1 rule, or tried to move directly to the goal state. This makes sense, because based on their experience in microworld 1 only one operator seemed to be available to them: <move-1>. Or alternatively, a goal state so nearby apparently is very appealing. After having tested both candidate states one or two times both students tried a <move-10> operator and quickly solved the problem.

Subject S02 behaved somewhat differently. After having tried the <move-1> operator, he tried 7 times in a row to move the pawn to the goal state. Subsequently, the old microworld 1 rule was tested for the second time. Probably, then S02 realized that he had no options left based on his knowledge of addition and his previous experience with the microworld. To find a way out of the impasse , he started systematically testing spatially adjacent states. Notice that he only intended to move to orthogonally neighboring positions. However, if his choices were guided by his knowledge of addition he would not have tried to move to the positions 46 and 55. Obviously, some sort of spatial logic for selecting candidate states has temporarily taken over.

One particular function of the ten-square is that it provides for heuristic information, i.e. information that helps to reduce the number of candidate states to be evaluated. If the ten-square would not provide heuristic information, this would imply that every number, or every position has an equal chance to be choosen as the next position for the pawn. This null hypothesis was tested for the first move proposed by the students on the problem at hand: 56 + 12 = ..; The prediction could be rejected (Pearson Chi Square $X^2 = 176.97$, df = 3). The subsequent prediction that there would be no difference between adjacent and non-adjacent positions was also rejected (Likelihood Ratio Chi Square $X_L = 34.04$, df = 1). In conclusion, not every position is considered a candidate position, and adjacent positions have a higher chance of being choosen than non-adjacent positions.

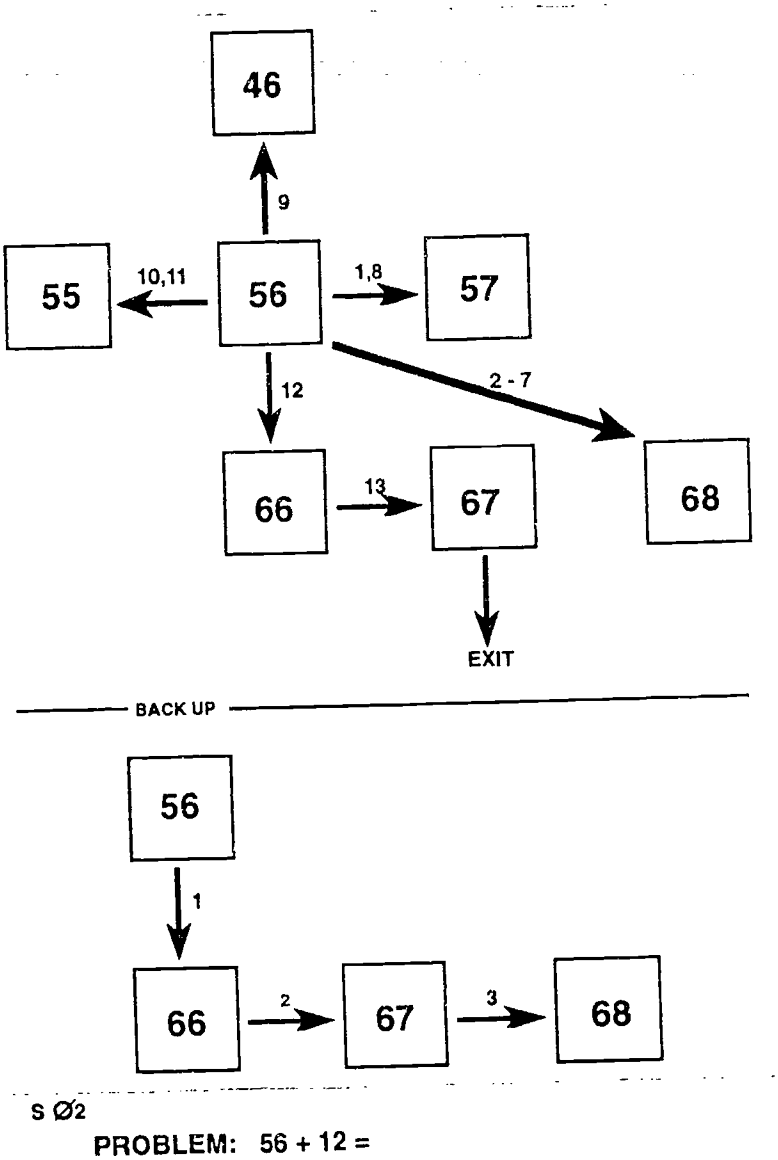


Figure 8. Problem solving graph of subjects S02 at the first confrontation of a newly imposed constraints in microworld 2.

DISCUSSION

This paper presents a microworld that differs from traditional microworlds in that the task environment does not comprise more states and operations than a problem space in which a particular cognitive skill can be developed, would include. It is hypothesized that this condition will enable students to acquire this cognitive skill simply as a result of a problem solving acitivities without the help of a teacher. In order to facilitate this process of acquisition of a cognitive skill by problem solving, certain additional features were built in the microworld. Firstly, in order to gradually mould the initial step-by-step searching behavior of the novice into the efficient two-step addition algorithm of the 'expert', a series of five microworlds is designed. In every microworld a new constraint made a previously transparent solution problematic. This increasingly constraining of the solution path is supposed to trigger problem solving activities. Secondly, the microworld is designed so as to activate a sort of spatial logic which is expected to guide problem solving activities.

It can be concluded that in every session continuing experience with the task has turned solving addition problems into a routine action. This conclusion is based on two findings: (a) the significant reduction in solution time within each session, and (b) the fact that at the end of every session the sequence of operators seems to be known beforehand - there is no trying out of (inadmissible) operators anymore. The actually number of moves as performed by the students closely followed the number of moves required given the addition problem and the current set of constraints.

The evidence presented in support of the role of the ten-square as a mental model for guiding the problem solving behavior is only fragmentary and deserves further study. However, the results do suggest that the ten-square is effective in narrowing down the number of candidate positions to be considered in the process of finding a path between the initial state and the goal state. This is important in a discovery learning environment, because trying out new problem states with a low probability of success, most likely, will negatively influence the student's motivation.

A question to be studied next relates to the transfer from solving addition problems using the ten-square to situations where the addition problems must be solved mentally, i.e. without the support of a spatial representation of the number system.

REFERENCES

Abelson, H., & diSessa, A.A. (1980).**Turtle Geometry: the computer as a medium for exploring mathematics** Cambridge (MA): M.I.T. Press.

Beem, A.L. (1990). **Aspects of memory structure and processes in mental arithmetic.** Doctoral Dissertation Leiden University, Leiden.

Beem, A.L., & Ippel, M.J. (1990). A structuring principle for the memory representation of simple arithmatic facts. In: H.Mandl, E. de Corte, N. Bennett, & H.F. Friedrich (Eds.) **Learning and Instruction. European research in an international context. Volume II.** (pp. 445-460). Oxford: Pergamon.

Borys, S., Spitz, H.H., & Dorans, B.A. (1982). Tower of Hanoi performance of retarded young adults and nonretarded children as a function of solution length and goal state. **British Journal of Experimental Child Psychology, 33,** 87- 110.

Byrnes, M.A., & Spitz, H.H. (1979). Developmental Progression of the Tower of Hanoi Problem. **Bulletin of the Psychonomic Society, 14,** 379-381.

Card, S.T., Moran, P.,& Newell, A. (1983). **The Psychology of Human-Computer Interaction.** Hillsdale (N.J.): Lawrence Erlbaum Associates.

Holland, J.H., Holyoak, K.J., Nisbett, R.E., & Thaggard, P.R. (1986). **Induction, Processes of Inference, Learning and Discovery.** Cambridge (M.A.): The M.I.T. Press.

Ippel, M.J., & Meulemans, C.J.M. (1990). A Computer Simulation of the Acquisition of a Computational Skill in a Discovery-Oriented Microworld. In J.M. Pieters, K. Breuer, & Simons, P.R.J. (Eds.). **Learning Environments: Contributions from Dutch and German Research** (pp. 177-192). Heidelberg: Springer-Verlag.

Ippel, M.J., Kemmere, P.F.G., & Meulemans, C.J.M. (September 1989). **The design of a problem space for learning algorithms for mental addition and subtraction of two-digit numbers.** Paper presented at the Third European Conference on Learning and Instruction, Madrid, September 4 - 7.

Klahr, D. (1985). Solving problems with ambiguous subgoal ordering: Preschoolers' Performance. **Child Development, 56,** 940-952.

Newell, A. (1981). Reasoning, Problem Solving, and Decision Processes: The Problem Space as a Fundamental Category. In: R. Nickerson (Ed.). **Attention and Performance, Vol. 8.** Hillsdale (N.J.): Lawrence Erlbau Associates.

Newell, A., & Simon, H.A. (1972). **Human Problem Solving.** Englewood Cliffs (N.J.): Prentice Hall.

Papert, S. (1980). **Mindstorms: Children, Computers and Powerful Ideas.** New York: Basic Books.

Resnick, L.B. (1989). Developing Mathematical Knowledge. **American Psychologist, 44,** 2, 162-169.

Resnick, L.B., & Ford, W.W. (1981). **The Psychology of Mathematics for Instruction.** Hillsdale (N.J.): Lawrence Erlbaum Associates.

Rich, E. (1986). **Artificial Intelligence.** New York: McGraw-Hill.