

Learning Mathematics through the Educational Computer Language MAL

Han Hyuk Cho

Dept. of Mathematics Education

As the world goes into an informationized society, the role of computer has been greatly increasing as a tool to compile, keep, handle, and utilize the enormous amounts of information around us. Computer education, therefore, is emphasized nowadays and various efforts have been tried to introduce computers into the curriculum of schools. In Korea the vocational education courses in the primary, middle, and high schools provide such subjects as 'computer and life,' 'the structure of computer and computer language,' and 'flow chart and BASIC programming,' and etc. In mathematics education, however, the computer is seldom used, but flow charting is taught in the high school.

It is expected that it will not be long before the high school math program provides not only flow charting but how to program, as in other foreign countries. BASIC will probably be selected for this purpose because it is generally understood in Korea that 'computer programming' equals 'BASIC programming.' But attention needs to be paid to the fact that there are not a few negative views about BASIC. One of them is that BASIC is based on a clumsy programming method, and it is not appropriate for problem solving, especially for the tools for thinking. Nevertheless we are under the regretful situation in which much discussion is made only about when and how to use BASIC or LOGO without enough consideration being given as to whether these languages are suitable for computer and math education.

It is time to drop the opportunism in which we have adopted BASIC or LOGO language as a programming language for math and computer education without any criticism. It is time to break from the passiveness with which we have received the existing languages and have been only concerned about how to utilize them. We should consider what conditions are required of a programming language for math and computer education, and we should try to design a new language which is best fit for our educational purposes. This paper is to introduce and explain the educational characteristics and utilization of MAL, an educational computer language for math and computer learning, resulting from the kinds of efforts mentioned above.

1. Educational Computer Language MAL


The computer education in primary, middle, and high schools could be divided into two parts, 'education about computers,' and 'education through computers.' Education about computers aims to understand computer hardware and software, and to study algorithms that are basic to computer science. Education through computers aims to use computer hardware and software effectively for better learning. For these two educational purposes we need a computer programming language which can be called an 'educational computer language.' It is desirable that an educational computer language should play a stepping stone's role between the present computer languages and future computer languages. Therefore it is demanded that educational computer languages have the educational characteristics of the past, the present, and the future as follows.

First, an educational computer language should be easy to learn so that students who learned BASIC or LOGO in the past can interact with the computer with this new language without any further learning. Second, an educational computer language must provide the syntax structure of some high-level languages (PASCAL, C, etc.) and their merits (structured programming technique, recursion algorithm, etc.) so that students will use them in the future. Third, an educational computer language must have mathematical grammar and notations, functions, and graphic capability so that the primary, middle, and high school students can use it now for their arithmetic and mathematics learning.

MAL (Mathematical Algorithmic Language) is a computer language that tries to satisfy the three educational characteristics mentioned above. It is designed to overcome some of the shortcomings of BASIC and LOGO languages. Founded on the statements of BASIC that are most widely used, MAL was designed to unify the advantages of several languages including the graphics of LOGO, and the structured programming techniques of C and PASCAL. Following the idea of Papert [13], the learner is supposed to learn arithmetic, mathematics, and various computer languages while he draws pictures in MAL LOGO, and the languages acquired in this way are supposed to be used again for arithmetic and mathematics learning.

A. Mal Language = Mal Basic + Mal Logo + Mal Pascal + Mal C

There are three basic structures used in an algorithm. The corresponding statements of MAL language to those structures are if, for, and def statements. Students who are accustomed to using other languages can use MAL easily, because the statements of MAL contain the basic statements of BASIC, LOGO, PASCAL, and C languages. There are, of course, no strict boundaries among the basic statements since they all work together in MAL. For convenience, however, MAL statements can be divided into four groups; MAL BASIC, MAL LOGO, MAL C and MAL PASCAL statements. For details, let's execute the MAL statements in the mal.mal file.

M> run "mal.mal" 

```

♥1   if 3<5 then print "MAL BASIC (말 베이식)" else print 3<5
♣2   if(3<5) { print("MAL C (말 씨)") ; print "You may use GOTO" }
♥3   for i=1 to 1 ; print "말 베이식" ; next i
♣4   for i=1 to 1 step 1 do begin print("MAL PASCAL (말 파스칼)") end
5     repeat 1 { print "MAL LOGO",. ; print " (말 로고)" }
♥6   def a(x)=x*x-3 ; 함수 b(x)=a(x-3)+5
♣7   def f(x) { for i=1 to 36 ; fd x ; rt 10 ; next i }
♣8   def g(x,y) { fd x ; rt y ; if x<100 then g(x+3,y) }
♥9   a$="ab말" : arx[1]=123 ; ary[50]=321 ;
⊕10  만일 a$=="ab말" 이면 쓰자 "ASCII of a is ",a$[1] 아니면 쓰자 0

```

```

M> run "mal.mal"
①   MAL BASIC (말 베이식)
②   MAL C (말 씨)
     You may use GOTO
③   말 베이식
④   MAL PASCAL (말 파스칼)
⑤   MAL LOGO (말 로고)
⑥   function a() is defined
⑥   function b() is defined
⑦   function f() is defined
⑧   function g() is defined
⑩   ASCII of a is 97

```

```

M> _
>> print input locate if~ for ~ def{goto:return} load save run clear

```

As seen above, statements in the mal.mal file are executed after the statements appeared on the editor screen. Statement ♥1 and ♥3 are if and for statements of BASIC respectively. Statement ♣2 and ♣4 are statements of MAL C and MAL PASCAL respectively. The fifth statement is a statement of MAL LOGO. As shown in the statement ♥6 and ♥9, MAL provides function state-

ment and array for arithmetic and mathematics education (for the syntax of MAL, refer to Cho [2]). Also the statement ⑩ shows that MAL responds to Korean as well as English command so that elementary school students can also learn it with ease. In summary, MAL is an easily accessible language for any learner regardless of what languages he learned in the past.

B. Structured Programming and Recursion

MAL was designed with the assumption that the learner can acquire the syntax of high-level computer languages and their benefits while he uses this language. As in the C and PASCAL languages, MAL has function statements that make structured programming and recursion possible. In this sense MAL overcomes the shortcomings of GW-BASIC which is incapable of structured programming and recursion, and LOGO that is incapable of representing mathematical functions.

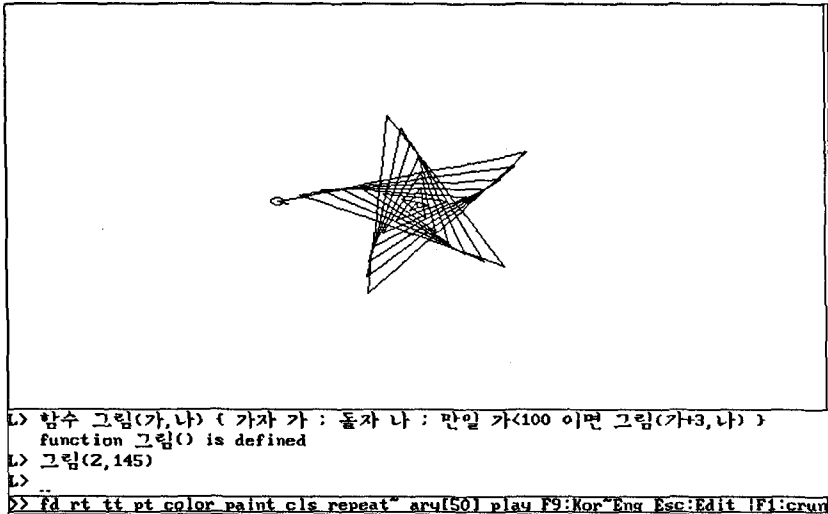
The basis of the structured programming is that the program should be written logically so that it can be read easily, and corrected and changed without difficulty. MAL also provides screen editor for writing and debugging algorithms. Structured programming techniques, one example of which is Top-Down programming technique, conform to the educational goal of improving problem solving ability. Recursion also has educational importance as a method for problem solving since the fundamentals of recursive thinking are to reduce the problem at hand to a simpler one.

L> 함수 그림(가,나) {가자 가: 돌아 나: 만일 가<100 이면 그림(가+3,나)} □

The above recursive statement is the Korean statement of $\text{def } g(x, y) \{ \text{fd } x ; \text{rt } y ; \text{if } x < 100 \text{ then } g(x+3, y) \}$, ♣8 statement in the mal. mal file. As you will see below, once the above command is given the computer responds with the message saying "function 그림() is defined", which means that function 그림(가, 나) is now preserved in the computer memory. Now calling 그림(가, 나) by assigning values to 가 and 나 as seen in 그림(2,123) will produce the following picture.

Note that 그림(가, 나) will react differently depending on the values assigned to 가 and 나. Unlike a compiler language, MAL has (def) function statements that are constructed and used while

↓ 그림(2, 145)



the learner interacts with the computer. Thus, the function statements of MAL can be used effectively for arithmetic and mathematics learning since they can provide a dynamic and interactive environment that other educational instruments can not.

C. MAL Language Designed for Mathematics Learning

Papert [13] said "There must be a way to get the kids to learn to talk to computers the way they learn to talk to people. Why should it be harder to talk about". Following his idea, MAL introduces Turtle Graphics Environment so that the learner can learn arithmetic and mathematics through the dialogue between the turtle and himself. MAL also adopts mathematical syntax and function notation for learning math, and MAL is designed to use structured programming for problem solving and mathematical thinking.

The graphic environment of MAL was designed to overcome some of the shortcomings of LOGO. First of all, unlike LOGO, MAL LOGO commands are created by adding variables to the four basic commands fd, rt, tt, and pt. For example, MAL LOGO commands tt, tt yellow, tt x, y, and tt x, y, d correspond to LOGO commands HT and ST, SETPC 14, SETXY [X Y], and SETXY [X Y] followed by SETH D respectively. Thus it is easy to learn and memorize MAL LOGO commands. Next, the MAL LOGO command

rt 45 makes the turtle turn 45 degrees to the left, and the turning direction is changed by the command rt (RoTate). Once we draw a picture, we can get the mirror image of it by using the command rt. In LOGO, however, to get the mirror image every RT must be changed to LT and vice versa. To overcome this shortcoming Zazkis & Leron [16] propose introducing a FLIP command similar to the MAL LOGO command rt. Next, a MAL LOGO function can be expressed mathematically as seen from $f(x, y)$, while the corresponding LOGO expression $F : X : Y$ is not at all mathematical. This is why Aberson & diSessa adopted mathematical expressions in their book [3] on LOGO. MAL LOGO also improves such confusing LOGO statement as `REPEAT 36 [FD 10 RT 10]`. The corresponding MAL LOGO statement is `repeat 36 {fd 10 : rt 10}`. Finally, MAL LOGO provides additional statements such as goto, array, and a BASIC's for-next statement `for i=1 to 36 ; fd 10 ; next i` using a variable i .

Educational computer languages should be able to handle functions and graphing easily in order to learn mathematics. MAL meets these qualifications for educational computer language since it provides function statements, array, and a graphic environment. In particular, MAL can express the following math functions which BASIC or LOGO can not.

(function by equation) : `def f(x, y)=x*x+y*y`

(conditional function) : `def f(x) {if x<=0 then return x else return -x}`

(recursive function) : `def f(x) {if x<0 return 0 else return f(x-1)+x}`

II . MAL Language and Mathematics Learning

The computer is expected to make a great contribution to the present state of learning with its power of rapid calculations, graphics, animation ability, and immediate feed back. This new technology should have a powerful effect on what is taught at every level of mathematics, as well as on how it is taught. There are three ways in which the computer can be applied to the educational situation. The first method is CPS(Computers Program Students) in which the computer software leads learning as if it were a teacher or an assistant teacher. In the second method SPC (Students Program Computers) the learner takes the initiative in

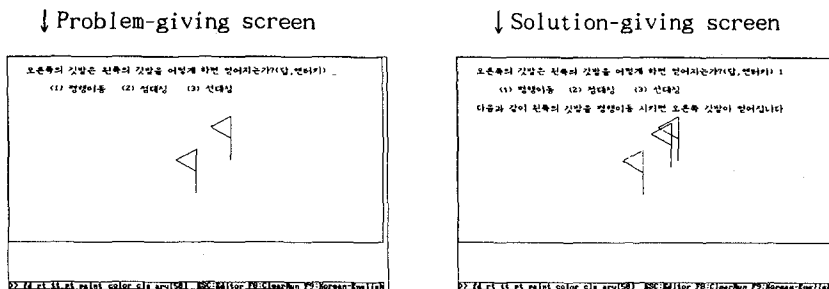
programming for problem solving. This is where he is supposed to create mathematical experiments and make observations. The third method CPS+SPC, a combination of CPS and SPC in the sense that the student learns through both the execution of the program and the program itself. For this kind of integrated learning, we need a computer programming language having an authoring capability in the broadest sense. MAL is designed to be used for all three of these methods.

A. CPS Learning through MAL

The first learning method is CPS, which leads learning by using the educational software. In CPS, the computer plays the role of a teacher or an assistant teacher and the student acquires his or her learning from the computer-teacher. There are many learning types related to this method such as drill-practice, private-tutor, and simulated experiment.

a) Private-tutor program for parallel translation

Computer simulation arouses students' interest by showing them some moving pictures that books or blackboards can never offer. Therefore the computer can be used effectively for helping students understand some difficult dynamic concepts and procedures. The following program presents a geometry problem first, then the answer and the concept of parallel translation are presented by the animated pictures.

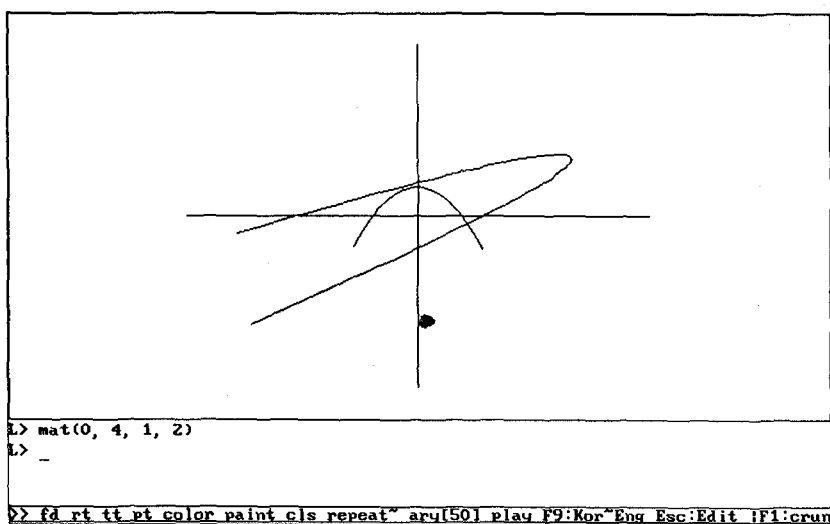


b) Program for the experiment of matrix transformations

The following will transform the graph of the given quadratic function by a matrix transformation. Since the program draws the given graph and the transformed graph at the same time it is

possible to observe the dynamics of the matrix transformation from a monitor. For example, a command such as `mat(0, 4, 1, 2)` performs the matrix transformation of $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$ with entries 0, 4, 1, and 2 respectively.

↓ Linear transformation `mat(0, 4, 1, 2)`

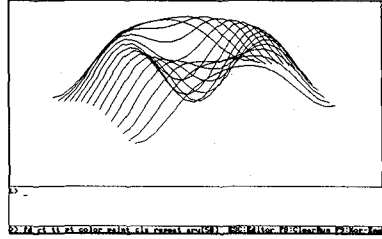
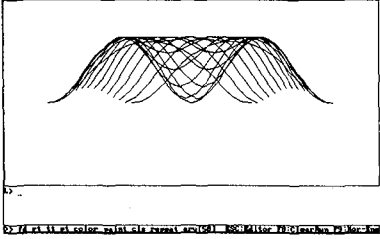


Matrix transformations can be applied to draw 3 dimensional pictures, fractal images, and ellipses. Through such applications students will develop an interest and positive attitude toward mathematics. The first two pictures below are created by a two-variable function ; the third one affine transformations [4] ; and in the last one, ellipses are obtained from a unit circle through rotation.

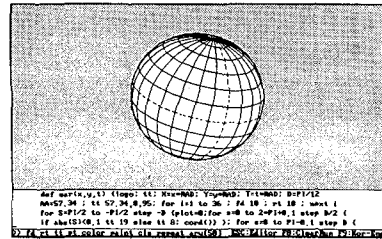
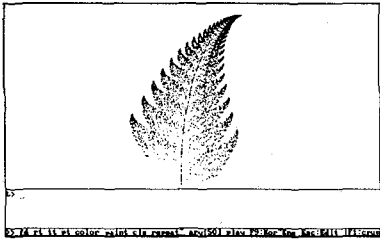
B. SPC Learning through MAL

In the learning method SPC, the student actively acquires his own learning through programming. In this case, the learner plays the role of a teacher and the computer plays the role of a student. One of the learning methods related to SPC is PERCs (Programming Exercise Related to Contents) [9]. In PERCs, the act of developing an algorithm to solve a particular problem, programming it, testing it, and refining it, is itself problem solving. For this learning-through-programming process, the existing computer languages are less appropriate than MAL or LOGO

↓ 3-D rotation by a matrix transformation



↓ fractal image and the earth by matrix transformations



which can offer an environment called 'Mathworlds' constructed by mathematical objects and operations.

a) Programming for the property of integers

Programming math problems can help the learner to acquire mathematical reasoning by making the problem solving process concrete and logical. For programming, one is required to analyze and conceptualize the problem, even if the learner has understood it already. Through the programming process, he is able to have an opportunity to reflect and review his mathematical knowledge.

A prime number is an integer that does not have any factor except 1 and itself. The following program shows how to find all the prime numbers between 2 and 100. We can think of some applications ; for example, to make the def function statement finding the prime numbers, or to double the execution speed by changing 『for i =2 to N-1』 on the third line.

b) Programming for the convergence of iteration sequences

The following program determines the orbit diagram for the

```
(MAL BASIC)
1 for N=2 to 100
2   flag=1
3   for i=2 to N-1
4     if N/i == int(N/i) then flag=0
5     if flag <> 0 then print N, " ",
6   next i
7 next N
```

```
Execution
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73
79 83 89 97
```

long-term iterative behavior of the quadratic function $f_c(x) = x^2 + c$. From the graphs, the learner can predict the convergence of sequences related to the graphs and then can create an experiment on the various aspects of the convergence. For details refer to the 1991 yearbook of NCTM 『Discrete Mathematics across the Curriculum, K-12』 or 『The College Mathematics Journal, Vol. 22』.

```
(MAL PASCAL)
1 def orbit() begin
2   for c=0.25 to -2 step -0.01 do begin
3     x=0 ;
4     for k=1 to 20 do x=x*x+c
5     for k=1 to 50 do begin x=x*x+c : pt(100*c+100,50*x-5 ) end
6   end
7 end
8 def main() begin logo : tt : orbit() end
```

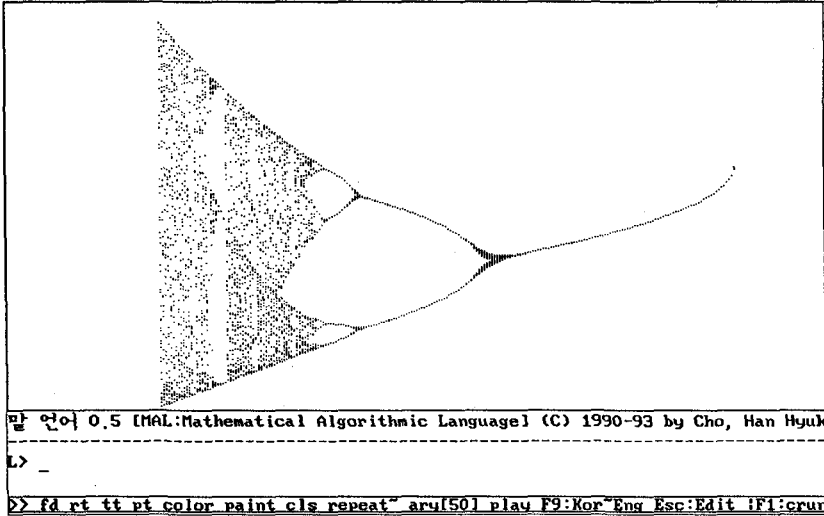
C. CPS+SPC Learning through MAL

In CPS+SPC the learner is supposed to study under the computers instruction by executing the ready-made programs. Next, he explores the source program making modifications as necessary. For such integrated learning as CPS+SPC, an educational programming language must have some authoring capability. However, due to the shortage of such languages, either CPS or SPC has been emphasized.

a) Learning statistics using the Monte Carlo method

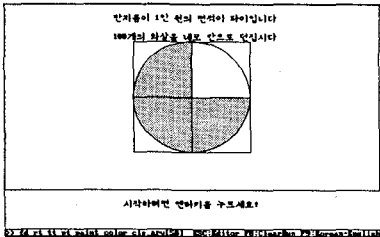
CPS : Grasp of the statistics concepts through mathematical experiment

↓ Orbit diagram of $f_c(x)$

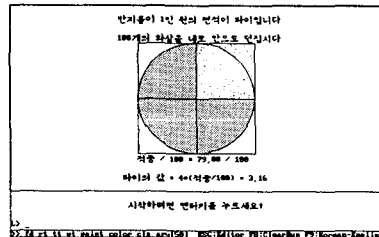


Using the computer's capability to generate random numbers, we can determine the value of π by a special experiment using the Monte Carlo method. For example, after shooting one hundred arrows at the target below we can estimate the value of π by counting the number of hits. This kind of computer experiment leads the learner to understand more about probability and statistics. By executing this program we can obtain the following screens designed to teach probability and statistics.

↓ Beginning screen



↓ π value by the Monte Carlo method



By doing statistical experiments on the approximate values of π obtained by Monte Carlo method, students can compute the average value and its deviation. Students may check whether the graphic representation of the approximate values form a normal

distribution or not. Also by taking the confidence interval, students may infer the real value of π

SPC : Problem solving through programming

Once the learner understands the source program, he or she may alter it to find the area related to an ellipse or hyperbola. This method will arouse the student's interest, and it will help the student to acquire a central understanding of probability theory. Also already mentioned, the learner can approach statistics easily through this method.

	(MAL C)
1	def arrow() { x = rnd(100) / 99 ; y = rnd(100) / 99 ;
2	if(x*x + y*y <= 1) return 1 else return 0 }
3	def pi() { s=0 ;
4	for i=1 to 100 {
5	if(arrow() == 1) s = s + 1 }
6	print(4 * s / 100) }

b) Using and understanding the Gaussian elimination algorithm

The Gaussian elimination algorithm is useful for finding the inverse matrix and for solving a system of equations. The following is an example of CPS+SPC learning where the learner is supposed to become familiar with the Gaussian elimination algorithm through the execution of the program first, and then studying the source program in order to understand the algorithm itself.

CPS : Drill and practice through the execution of program

In learning Gaussian elimination, what really matters is the algorithm, not addition or multiplication. Usually the simple hand calculation process along with some minor mistakes that usually occur are irrelevant to algorithm but sometimes make Gaussian elimination seen difficult. As a result, a program is usually required to solely offer the learner a chance to concentrate only upon the Gaussian algorithm.

The following screens are obtained by the MAL program that can be used for studying the Gaussian elimination algorithm. After running the program, the learner is supposed to input the coefficients of the system, and then find the solution using the following three basic operations.

(i) $m(b, k)$ = multiplying b-th row by k

thor.

Prof. Han Hyuk Cho (hanchoakrsnucl.bitnet)
 Department of Mathematics Education
 Seoul National University
 Seoul, KOREA

References

- Kim, Myoung-Lyoul, 1989, *A Study on the Design of CELL : Computer Educational Little Language*, Ph. D. Thesis, Hong-Ik University.
- Cho, Han Hyuk, 1991, *Computer and Mathematics Learning through MAL Language*, Lecture Note.
- Aberson and diSessa, 1986, *Turtle Geometry : The Computer as a Medium for Exploring Mathematics*. Cambridge, MA. : The MIT Press.
- Barnsley, M., 1988, *Fractals Everywhere*, San Diego, CA : Academic Press.
- Devaney, R., 1991, "The Orbit Diagram and the Mandelbrot Set," *The College Mathematics Journal*, 22, pp. 23-38.
- Engel, A., 1984, *Elementary Mathematics from an algorithmic standpoint*, Translated by F. R. Watson, Staffordshire : Kelle Mathematical Education Publications.
- Hoyles and Sutherland, 1989, *LOGO Mathematics in the Classroom*, London : Routledge.
- Maddux and Johnson, 1988, *LOGO : Methods and Curriculum for Teachers*, New York : The Haworth Press.
- NCTM, 1984, *Computers in Mathematics Education, 1984 Yearbook*, Reston, VA. : The Council.
- _____, 1991, *Discrete Mathematics across the Curriculum, K-12, 1991 Yearbook*, Reston, VA. : The Council.
- Oldknow and Smith, 1983, *Learning Mathematics with Micros*, Chichester : Ellis Horwood Limited.
- Papert, S., 1980, *Mindstorm : Children, Computers and Powerful Ideas*, New York : Basic Books.
- _____, 1980, "Personal Computing and Its Impact on Education," In *The Computers in the School : Tutor, Tutee*, edited by R. P. Taylor, New York : Teachers College Press.
- Ralston, A. 1987, "The Impact of Computer Science on Pre-college Mathematics : A Research Program," In *Developments in School Mathematics Education around the World*, edited by Wirszup & Streit, Reston, VA : NCTM.
- Thompson, P., 1987, "Mathematical Microworlds and Intelligent Computer-Assisted Instruction," In *Artificial Intelligence & Instruction*, ed-

ited by G. P. Kearsey, MA : Addison-Wesley Publishing Company.
Zazkis and Leron, 1991, "Capturing Congruence with a Turtle," *Educational Studies in Mathematics*, 22, pp. 285-295.