

## 교육용 컴퓨터 언어 MAL을 통한 수학 학습

조 한 혁(서울대학교)

현대의 정보화사회에서 컴퓨터의 역할이 증대되고 있고, 이에 따라 컴퓨터 교육의 중요성이 강력히 부각되고 있다. 이미 여러나라의 교육과정에 컴퓨터가 도입되고 있으며, 한국에서는 국민학교와 중고등학교에 컴퓨터와 BASIC 프로그래밍등이 도입되고 있다. 그러나 수학교과에서의 컴퓨터 활용은 거의 이루어지지 않고 있으며, 단지 고등학교 일반수학의 수열단원에서 알고리즘과 순서도만이 간단히 지도되고 있을 뿐이다. 머지않아 한국의 수학교과에도 미국, 일본등과 같이 순서도뿐 아니라 그에 대한 프로그래밍의 지도가 이루어지리라 기대된다.

컴퓨터는 컴퓨터의 도스(DOS)와 함께 구입하게 되는데, 이때 도스와 함께 베이식언어도 기본적으로 제공되게 된다. 이러한 이유등으로 초중학교에서는 컴퓨터 프로그래밍이 BASIC 프로그래밍으로 인식되고 있으며, 이에 따라 BASIC 언어가 자연스럽게 교육용 컴퓨터 언어로 부상하고 있다. 그러나 BASIC 언어는 건전한 프로그래밍 방법에 어긋나는 졸렬한 방법과 기교를 사용하는 언어로서, 사교의 도구 및 컴퓨터 교육에 부적절하다는 여러 비판적인 주장이 있다. 그럼에도 불구하고, BASIC 또는 LOGO 언어가 컴퓨터교육에 적합한 언어인지에 대한 논의도 없이 이들 프로그래밍 언어를 언제 어떻게 컴퓨터와 수학교육등에 활용할 것인지에 대한 논의만 현재 무성하게 이루어지고 있다.

우리는 아무런 비판없이 BASIC 또는 LOGO 언어를 수학교육과 컴퓨터 교육을 위한 프로그래밍 언어로 채택하는 편의주의에 빠질 것이 아니라 교육적으로 요구되는 프로그래밍 언어의 조건을 먼저 생각해 보아야 할 것이다. 즉, 기존의 언어를 수동적으로 받아들여 그 언어의 활용에만 관심을 가지는 것에서 벗어나, 능동적으로 컴퓨터 교육과 수학교육에 적합한 언어가 어떠한 것인가를 살펴보고 또 그러한 언어를 제작하여 활용하려는 노력이 요구되는 것이다. 본 논문에서는 이러한 생각에서 제작된 교육용 컴퓨터 언어 말(MAL)의 설계 원칙을 살펴보고, 또한 말 언어를 통한 수학학습과 지도에 대한 가능성을 살펴본다.

# I 장: 컴퓨터 언어를 통한 학습방법

컴퓨터는 정보화 사회에서 무엇을 가르쳐야 하는가의 문제 뿐만 아니라 어떻게 가르쳐야 하는가에도 의미하는 바가 크다. 컴퓨터와 컴퓨터 언어를 통해 교육의 현장에서 학습을 유도하는 방법에는 CPS 방법, SPC 방법, 그리고 CPS+SPC 방법등이 있다. 이러한 종합적인 학습이 이루어 지기 위해서는 넓은 의미의 저작 기능을 가진 컴퓨터 프로그래밍언어가 필요하게 되는데, 베이식등의 컴퓨터 언어는 이러한 기능이 없기에 새로운 교육용 컴퓨터 언어가 필요하게 된다.

## 1절: CPS 학습방법

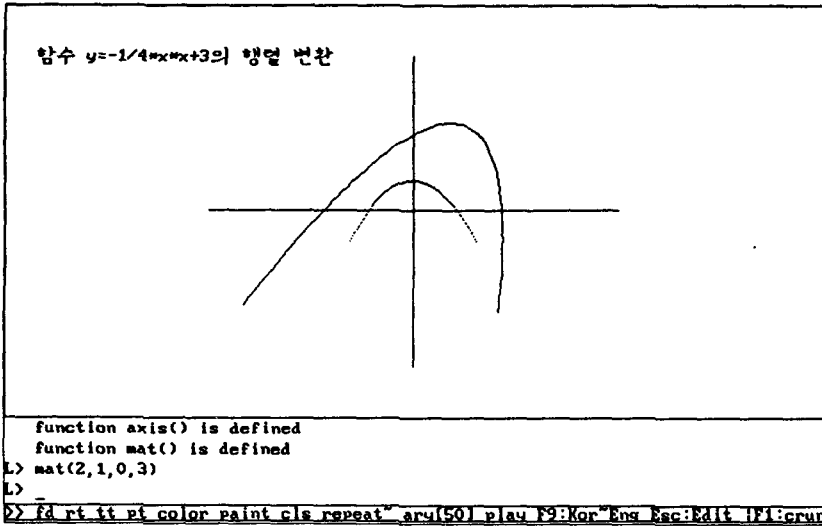
CPS (Computers Program Students ) 학습방법은 컴퓨터 프로그램이 교사 또는 준 교사의 위치에서 학생들의 학습을 유도하는 방법이다. 즉 학습하고자 하는 내용과 관련되어 미리 작성된 컴퓨터 프로그램의 실행을 통해 학습을 유도하는 방법으로 이러한 학습의 유형에는 모의 실험형, 개인교수형, 반복 연습형등이 있다. 요사이 CAI 를 표방하는 대부분의 교육용 소프트웨어들이 이러한 CPS 학습을 위해 제작되고 있다. 그러나 컴퓨터를 통한 CPS 학습의 진가는 기존의 시청각 도구등으로 만들 수 없었던 학습환경을 제공할 때 나타난다. 따라서 단순히 교과서 또는 참고서를 컴퓨터 화면에 그대로 옮겨 놓은 교육용 소프트웨어는 바람직한 CPS 학습용이라 할 수 없다. 대화적이고 역동적인 학습환경을 위해 교육용 컴퓨터 언어는 LOGO 또는 MAL 언어의 함수문을 가져야 한다.

행렬변환에 대한 모의실험 프로그램 : 다음은 주어진 이차함수의 그래프를 행렬에 의해 일차변환 시키는 프로그램이다. 다음의 화면에서 보듯 학생들은 L> 프롬프트 뒤에서 함수  $\text{mat}(a,b,c,d)$ 의 변수  $a,b,c,d$ 의 값을 변화시키며 닳음변환, 회전변환, 그리고 역변환이 없는 변환등을 예측하고 또한 실험할 수 있다. 이 때, 주어진 그래프와 변환 후의 그래프가 변화의 순서에 따라 화면에 동시에 그려지게 동적인 일차변환을 볼 수 있다. 예를 들어 『 $\text{mat}(2,1,0,3)$   $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$ 』 명령은 행렬  $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$ 의  $a,b,c,d$ 의 값이 각각 2, 1, 0, 그리고 3 인 행렬변환을 보여준다.

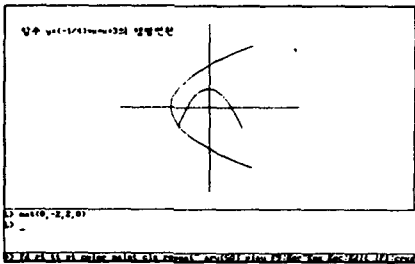
그 다음에 있는 원편의 그림은 L> 프롬프트에서  $\text{mat}(0,-2,2,0)$ 의 행렬변환이고, 그 오른편은  $\text{mat}(2,0,0,5)$ 에 의한 변환인데 이 것은 주어진 2차함수의 그래프를 x축으로 2배 y축으로 5배 확대시키는 변환이다. 또한 프로그램의 원문을 고치면 다른 함수의 그래프도 행렬변환 시킬 수 있다. 이와 같이 말 언어의 함수문은 파

스칼라나 씨의 함수와는 달리 컴퓨터와 대화하며 만들어지고 또한 사용되어질 수 있기에 산수와 수학 학습에 획기적으로 사용될 수 있다.

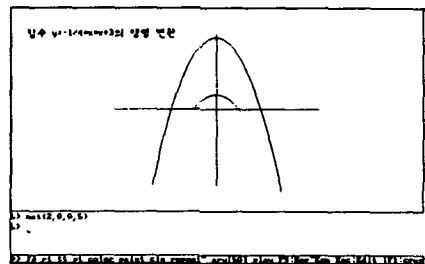
↓  $\text{mat}(2, 1, 0, 3)$  에 의한 행렬변환



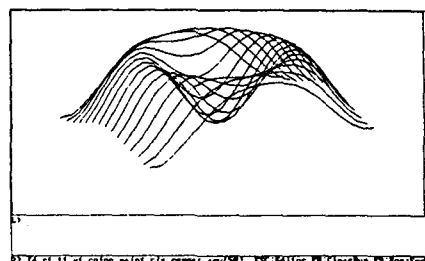
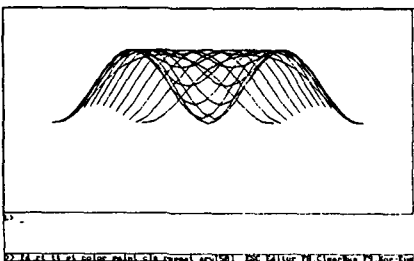
L>  $\text{mat}(0, -2, 2, 0)$



L>  $\text{mat}(2, 0, 0, 5)$



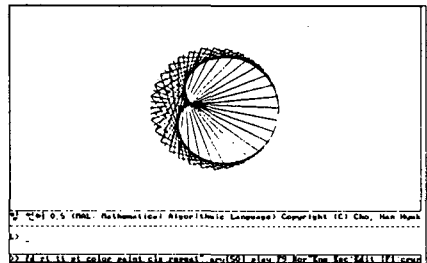
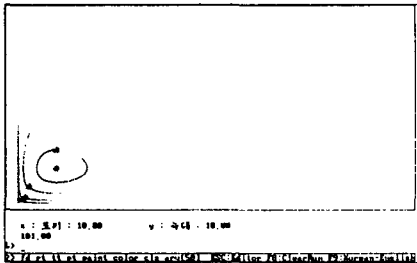
이러한 행렬변환은 다음과 같이 3차원 그림의 회전, 프랙탈 그림, 그리고 타원을 이용한 그림등을 그리는데 활용될 수 있다. 학생들은 이러한 예를 통하여 행렬변환에 대한 흥미와 탐구심을 갖게 되고, 또한 수학에 대한 바람직한 태도를 갖는데 도움이 될 것이다.





```

말 언어-(한글 말 베이식)
함수 mis(xs,ys) : cls:tt: input 16,4,"발사지점의 엑스좌표 : ",: print xs
input 18,4,"발사지점의 와이좌표:",:print ys:tt xs,ys:rt(150,80):fd 3
반복 i=-75 부터 75:만일 (-2*i-xc)*(-2*i-xc)+(80-yc)*(80-yc)<1 이면
input 22,4,"명중!":input 16,4,"격추":locate 16,26:print xc
입자 18,4,"격추": 위치 18,26 : 쓰자 yc
for i=1 to 9 : fd 3 :fd -3:rt 40: next i: goto 10: 만일끝
XC=xc:YC=yc:DC=dc:tt white:tt -2*i,80,180:tt: 가자 2 : 거북
tt XC,YC,DC: tt yellow: rt(-2*(i+1),80): fd 3 : 다음 i
10: locate 24,1:함수끝: 함수 main() : 로고: 색: mis(0,-80): 함수끝
    
```



시간 t에서의 토끼와 늑대의 수를 각각 x(t), y(t)라 하면 다음이 성립한다 [8].

$$P'(t) = \begin{cases} dx/dt = ax - bxy \\ dy/dt = -cx + dxy \end{cases}$$

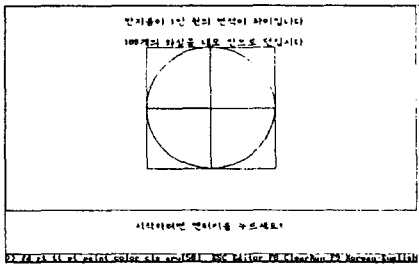
이제 eat(xp,yp)는 토끼와 늑대의 초기마리수를 각각 xp, yp라하고 a=b=2, c=d=1로 잡았을 때 얻어지는 미분방정식의 해곡선이라 하자. 위의 그림은 xp,yp에 여러 값을 주고 그려본 그림이다. eat(10,10)을 실행시키면 평형점을 그리며, 또한 이 평형점 주위의 생태곡선은 타원을 이룸을 볼 수 있다. 이와 같이 주어진 문제를 해결하기 위해 알고리즘을 찾아내어 프로그래밍을 하고, 시험해 보고, 다듬고 하는 자체가 바로 문제해결이다.

### 3절: CPS + SPC 학습방법

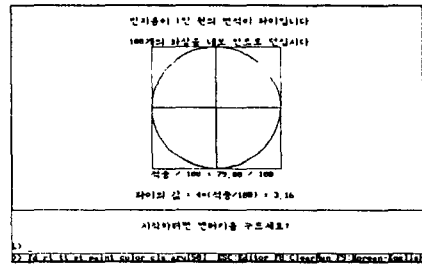
CPS+SPC 학습방법은 프로그램 실행을 통해 CPS 학습도 하고 또한 프로그램 자체의 학습을 통해 SPC 학습도 하는 방법이다. 학생들은 먼저 완성된 프로그램의 실행에 의한 컴퓨터의 지시에 의해 학습을 하고, 다음 그 프로그램 원문의 이해와 수정등을 통해 SPC와 CPS학습을 동시에 하게 된다. 이러한 CPS+SPC 학습과 같은 종합적 학습을 위해서는 저작기능을 가진 교육용 프로그래밍 언어가 필요하게 되는데, 지금까지 이러한 언어의 부족등의 이유로 CPS 또는 SPC 의 한 측면만이 강조되어 왔다.

몬테카를로 방법을 이용한 수학적 확률실험 : 난수를 발생시킬 수 있는 컴퓨터의 특성을 이용해 몬테카를로 방법이라 알려진 특별한 모의 실험을 통해  $\pi$ 의 값을 추정할 수 있다. 예를들어 다음과 같은 표적에 100 개의 화살을 쏜 후 명중된 화살의 개수를 세어 확률적으로  $\pi$ 의 값을 추정할 수 있다. 이와같이  $\pi$ 의 값을 구하는 컴퓨터 실험을 통해 자연스럽게 확률과 통계에 대한 학습을 유도할 수 있다. 다음은 말 언어로 짠 프로그램의 실행을 통해 확률과 통계 학습을 유도하는 화면이다. 이처럼 몬테카를로 방법에 의해 구해진  $\pi$ 의 근사값들을 통계처리 하면 평균및 분산을 구해진다. 또한 근사값들을 그래프로 표현하며 그것들이 정규분포를 이루는 가를 실험할 수 있고, 또한 그 값들을 이용해 우리가 알고있는  $\pi$ 의 참값을 신뢰구간을 잡아 검정할 수도 있다.

↓수학적 확률 실험 초기 화면



↓몬테카를로 방법에 의한 파이의 값



확률과 통계에 대한 이러한 접근은 학생들에게 흥미를 고취시킬 수 있으며, 그 원리가 확률 개념에 기반을 둔 것이므로 확률적 아이디어의 학습에 도움을 준다. 또한 앞서 언급한 바와같이 통계로의 자연스러운 접근을 가능케 한다. 다음은 위에서와 같이 화살을 백개를 쏘아 명중한 것을 세는 프로그램이다. 이 프로그램의 원문을 고쳐  $y=x^2$ 에 의해 만들어지는 면적을 근사적으로 구하기 위해 2행에 있는  $x*x+y*y <= 1$  을 어떻게 고쳐야하는지 생각해볼 수도 있다.

```

말 언어-(말 씨)
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 ) }
    
```

## II장: 컴퓨터 언어 “말”의 교육적 설계

초중고교에서의 컴퓨터 교육은 컴퓨터에 대한 교육과 컴퓨터를 통한 교육으로 크게 나누어질 수 있다. 컴퓨터에 대한 교육의 핵심은 컴퓨터 이해 및 컴퓨터 과학의 기본이 되는 알고리즘 학습에 있으며, 컴퓨터를 통한 교육의 핵심은 컴퓨터 하드웨어와 소프트웨어를 교육현장에서 유용하게 활용하는 것에 있을 것이다. 이 때, 알고리즘 학습등을 위해 컴퓨터 프로그래밍 언어가 필요하게 되는데, 이 경우 사용될 수 있는 언어를 교육용 컴퓨터 언어라 한다. 바람직한 교육용 컴퓨터 언어는 현재의 여러 컴퓨터 언어와 미래의 컴퓨터 언어 사이에서 징검다리 역할을 할 수 있는 바람직한 컴퓨터 언어로서, 다음과 같은 과거와 현재 그리고 미래에 대한 교육적 특징을 갖고 있는 언어라야 한다.

첫째, 교육용 컴퓨터 언어는 배우고 쓰기가 쉬워야 하며, 특히 과거에 BASIC이나 LOGO 언어등을 배운 학생이 쉽게 이 언어로 컴퓨터와 이야기를 나눌 수 있어야 한다. 둘째, 교육용 컴퓨터 언어는 현재의 초중고교 학생이 산수와 수학등을 학습하기 쉽도록 수학적 기호 체계와 문법, 함수적 표현력, 그리고 그래픽 표현 능력을 갖춘 언어라야 한다. 셋째, 교육용 컴퓨터 언어는 미래에 학생들이 대학등에 가서 사용할 고급 컴퓨터언어 (PASCAL, C 언어등) 들의 문법구조와 여러 장점 (구조적 프로그래밍 기법과 리컬전 알고리즘등) 을 이 언어를 쓰면서 배울 수 있어야 한다.

말 언어는 Mathematical Algorithmic Language 의 머리글자인 MAL을 이름으로 갖는 언어로서 앞에서 지적된 교육적 특징을 지향하여 설계 제작된 교육용 컴퓨터 프로그래밍 언어이다. 말 언어는 기존의 BASIC이나 LOGO 언어등이 갖고있는 교육상의 단점을 극복하고자 개발된 교육용 컴퓨터 언어로서, 현재 널리 쓰이는 베이식 언어의 명령문위에 로고 언어의 그래픽과 씨와 파스칼 언어의 구조화 프로그래밍 기법등의 장점을 통합해 설계되었다. 말 언어는 말 언어의 그래픽 상태에서 거북이와 산수와 수학에 관계되는 그림등을 그리며 자연스럽게 여러 컴퓨터 언어와 알고리즘을 배우고, 또한 이렇게 배운 알고리즘은 다시 산수와 수학학습 등에 쓰이도록 설계되어 있다.

### 1절: 과거학습을 위한 설계

현실적으로 교육용 컴퓨터 언어의 문법이 과거에 학생들이 배운 컴퓨터언어의 문법 또는 수학적 표현방법과 크게 어긋나는 것이라면 학생들에게 수용되기

힘들 것이다. 그러기에 교육용 컴퓨터 언어는 먼저 산수와 수학을 통해 학생들이 익힌 수리적 표현법을 이 언어에서 자연스럽게 쓸 수 있어야 하며, 또한 베이식 또는 LOGO 언어등에서 배운 지식도 이 언어에서 사용할 수 있어야 할 것이다. 이러한 관점에서 볼 때 기존의 베이식과 로고언어는 여러 문제점을 갖고 있다. 예를들어 LOGO 에서는 변수 X를 :X라고 표기하고 X에 36을 대입하는 표현을 MAKE "X 36 과 같이 쓰며, 또한 함수를 F :X :Y 과 같이 비수학적으로 표현하고 있다. 한편 베이식 언어의 배열문과 그래픽 명령도 수학적 표현을 쓰지 않는다. 이러한 이유로 Aberson & diSessa [5] 는 LOGO 의 함수문을 수학적으로 제안하였으며, 말 언어의 함수  $f(x,y)$  과 그래픽 명령은 수학적으로 표현된다.

말 언어는 한글 명령과 영어 명령을 동시에 알아 듣기에 저학년 학생이 쓰기에 쉽다. 또한 과거에 기존의 GW-베이식이나 로고 언어등을 배운 학생이 쉽게 쓸 수 있도록 베이식 언어의 기본 명령문 위에 로고 언어의 그래픽 명령문을 더해 제작되었고, 특히 베이식과 로고와 파스칼과 씨 언어의 if, for 그리고 def 명령문을 동시에 알아 듣도록 설계되었다. 물론 이 모두가 하나의 언어인 말에서 수행되는 것이므로 뚜렷한 경계는 없으나, 편의상 기존 언어와의 유사성을 가지고 말 베이식, 말 로고, 말 파스칼, 말 씨의 4가지 명령으로 구분할 수 있다. 예를 들어 다음에 보이는 컴퓨터 화면을 보자. 이 것은 말 언어의 에디터 (===선위) 에서 작성된 프로그램의 일부를 블럭으로 표시한 후, F1키로 실행시킨 모습으로 에디터와 디버깅의 예를 보여주고 있다.

```

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 가 : next i
◆4 for i=1 to 1 step 1 do begin fd 45 : print "Hello" end
5 repeat 1 { print "MAL LOGO : print "(말 로고)" }
♠6 def a(x)=x*x-3 : 함수 { 리턴 b(x)=a(x)+5
a$="ab말" : arx[1]=123 : ary[50]=321 :
만일 a$="ab말" 이면 쓰자 "ASCII of a is ",a$[1] 아니면 쓰자 0
=====
M> load "mal.mal"
M>
② MAL C (말 씨)
You may use GOTO
③ MAL 3: undefined variable 가
④ MAL 4: fd used outside MAL LOGO
⑤ MAL 5: missing "
⑥ function a() is defined
MAL 7: the last function is undefined
M> -
>> print input locate if~ for ~ def{goto:return} load save run clear
    
```



이 때, 블럭으로 표시된 프로그램이 말 언어의 실행화면(===선 아래)에서 실행되며 프로그램의 에러가 지적되고 있다. ♣2 은 C언어 스타일의 명령으로 에러가 없기에 정상적으로 실행이 되었고, ♥3 은 베이식 스타일의 명령문으로 정의되지 않은 변수 『가』 때문에 에러를 냈고, ◆4 은 파스칼 스타일의 명령문인데 그래픽 명령인 fd 를 텍스트 상태에서 사용했기에 에러를 냈고, 5번의 로고 명령문은 print 다음에 스트링을 " 로 닫지 않았기에 에러를 내었고, 마지막으로 ♠6 는 함수 b() 의 선언을 ) 으로 닫지 않았기에 에러를 발생시켰다. 이와 같이 말 언어는 베이식과 로고와 파스칼과 씨 언어의 if, for 그리고 def 명령문을 동시에 알아 들도록 설계되어 말 언어의 문법에 융통성을 주고있다. 프로그래밍의 진정한 지도 목표는 특정한 컴퓨터 언어의 문법 학습이 아니라 알고리즘의 이해와 표현 그리고 실행이다. 따라서 명확한 의도로 기술된 알고리즘이 너무 예민한 프로그래밍 언어의 문법 때문에 에러 처리가 된다면 곤란하다.

## 2절: 현재학습을 위한 설계

기존의 BASIC과 LOGO언어의 중요한 성공 요인이 각각 접근하기 쉬운 느낌을 주는 친근성과 컴퓨터와의 자연스러운 대화를 유도할 수 있는 Turtle 그래픽 환경이었다. 그러나 외국산 LOGO 는 한글 환경을 제공하지 않으며 또한 Hercules 그래픽을 쓰는 한국의 교육용 컴퓨터에서 정상적으로는 작동되지 않는다. 한편 베이식언어는 한글카드 또는 유틸리티에 의해 한글 입출력만을 제공하며, 또한 각각의 그래픽 카드에 대응하는 명령이 필요하기에 프로그램과 하드웨어와의 호환성에 문제가 있었다. 말 언어는 이러한 문제를 해결하기 위해 언어 자체에 한글 처리 능력이 내장되어 있으며, 한글 입출력은 물론 한글 명령의 사용도 가능하도록 설계되어있다. 예를 들어 GW-베이식의 『 play "cdefg" 』 명령은 말 언어에서 『 음악 "도레미파솔" 』 과 같이 명령된다. 또한 말 언어의 거북이(Turtle) 그래픽 환경은 여러 종류의 컴퓨터에서도 작동되도록 설계되어 있다.

컴퓨터와의 자연스럽게 친근한 대화를 유도할 수 있는 교육적인 환경을 위해서는 인터프리터 방식의 컴퓨터 언어가 적당한데, 이는 학생이 내린 명령에 대한 즉각적인 컴퓨터의 반응을 볼 수 있기 때문이다. 그러나 교육적으로 필요한 대화적 환경은 인터프리터 방식만 가지고 이루어지지 않는다. 교육용 컴퓨터 언어는 "프로그래밍은 컴퓨터 언어로 컴퓨터와 대화하는 것이다. 그러기에 컴퓨터 언어는 어린이가 어른과 대화하며 말을 배우듯이 컴퓨터와 대화할 그 무엇을 통해 배워야 한다." 라는 패럴트의 이론 [14] 같이 대화할 꺼리가 풍부한 환경을 아울러 제공해야 한다. 또한 이러한 그래픽의 환경은 Dr. Hallo 등이 제공하

는 단순한 그래픽 환경이 아니라 대상물들(objects)과 대상물들의 조작(operations)이 함께 존재하는 학습을 위한 그래픽환경 (Mathematical MicroWorld) [15] 이다. 말 언어는 패럴트의 주장을 따라 거북이(Turtle) 그래픽 환경에서 거북과 학습자와의 자연스러운 대화를 통한 컴퓨터와 산수-수학 학습을 유도하고 있다.

말 언어의 그래픽 환경은 산수와 수학 학습을 위해 로고의 그래픽 환경을 다음과 같이 재구성하여 배우고 쓰기 쉽도록 만들어져 있다. 먼저 말 로고의 명령문은 기존의 LOGO 와는 달리 fd, rt, tt, pt 의 네가지 기본 명령위에 변수를 도입해 구성되어 있다. 예를들어 tt은 LOGO의 HT와 ST 명령을, tt c 는 LOGO의 SETPC C 를, tt x,y 는 LOGO의 SETXY [X Y] 를, tt x,y,d 는 LOGO의 SETXY [X Y] 와 SETH D 명령을 나타낸다. 다음, 말 로고의 rt 45 명령은 거북을 45도 회전시키는 명령으로 회전 방향은 rt (RoTate)명령에 의해 결정된다. 그러기에 말 로고에서는 어떤 그림을 그리고 rt 하여 다시 그리면 대칭도형을 얻게된다. 이와는 달리 LOGO 에서는 대칭도형을 얻기위해 RT (Right Turn)를 LT (Left Turn)로 모두 바꾸어 주어야 하는데, 이러한 불편 때문에 Zazkis & Leron [16] 은 말 로고의 rt 명령과 같은 FLIP 명령을 제안하기도 했다. 또한 말 로고에서는 LOGO의 FD 45 RT 45 와 같은 혼동되는 표현을 문분리자를 도입해 FD 45 : RT 45 과 같이 개선했으며, repeat 36 { fd 10 : rt 10} 과 같은 LOGO의 단순한 REPEAT 명령은 물론 for i=1 to 36 : fd 10 : rt 10 : next i 와 같이 i값이 변하는 베이식의 반복문도 사용가능케 했다. 또한 말 로고는 LOGO가 지원하지 못하는 행번호와 goto 명령등의 사용을 가능케 한다.

바람직한 교육용 컴퓨터 언어는 학생들이 학교에서 컴퓨터와 산수 그리고 수학등을 학습하는데 불편이 없도록 산수와 수학의 중요한 개념인 함수와 그래프 등의 표현을 위한 여러 기능을 갖고 있어야 한다. 그러나 LOGO 언어는 수학적 함수문의 단점을 갖고 있고, 또한 BASIC 언어는 조건식과 점화식으로 주어지는 함수를 정의할 수 없고 또한 그래픽 구현에 번거러움이 있어 기하학및 그래픽 학습에 적당치 않다. 말 언어는 현재의 산수와 수학학습을 위해 기존의 BASIC 과 LOGO 의 이 같은 단점을 def 함수문을 사용하여 다음과 같이 극복했다. 먼저 수식으로 주어진 함수는 def f(x,y)=x\*x+y\*y 또는 def f(x,y) { return x\*x+y\*y } 같이 정의될 수 있다. 다음 절대값 함수와 같이 조건식으로 주어진 함수는 def f(x) { if x<0 then return -x else return x } 와 같이 정의될 수 있다. 마지막으로 피보나키 점화식과 같은 점화식으로 주어진 함수도 def f(x) { if x<0 then retrun 0 else retrun f(x-1)+f(x-2) } 와 같이 정의될 수 있다. 한편 말 언어는 함수의 일종인 수열과 행렬을 정의할 수 있도록 배열도 지원한다.

## 3절: 미래학습을 위한 설계

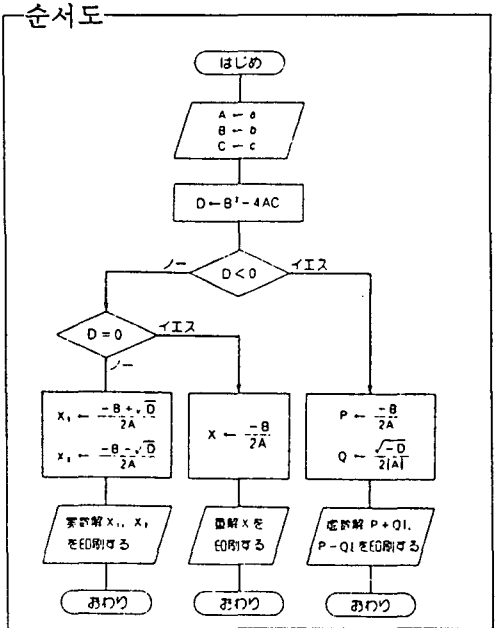
BASIC 언어는 대학등에서 거의 사용되지 않는 언어로서 구조화 프로그래밍 및 리컬전을 할 수 없는 언어이며, LOGO 언어의 문법구조는 미래에 학생들의 수학 및 컴퓨터 언어학습에 별 도움을 줄 수 없다. 그러기에 바람직한 교육용 컴퓨터 언어는 수학적 함수를 표시할 수 없는 LOGO 와 BASIC의 단점과 또한 구조화 프로그래밍 및 리컬전을 할 수 없는 GW-베이식의 단점을 극복해야 한다. 또한 바람직한 교육용 컴퓨터 언어는 미래에 학생들이 대학등에 가서 배울 고급 컴퓨터 언어(C, Pascal)들의 여러가지 문법구조와 장점 (함수 개념, 구조적 프로그래밍, 리컬전등) 을 이 언어를 사용하며 자연스럽게 익힐 수 있어야 한다. 이를 위해 말 언어의 문법은 씨와 파스칼 언어의 문법과 비슷하며, 또한 말 언어는 구조화 프로그래밍과 리컬전을 가능케 하는 *def* 함수문을 갖고있다.

구조화 프로그래밍의 기본은 프로그램을 읽기 쉽고 교정과 변형이 용이하도록 논리적인 방법으로 작성한다는 것이다. 말 언어에서는 *if*, *for* 그리고 *def* 명령문을 사용하여 이러한 구조화 프로그래밍을 할 수 있다. 구조화 프로그래밍 기법의 한 예가 Top-Down(하향식) 프로그래밍 기법인데, 이러한 기법은 문제해결력과 조직적 사고 방식을 중요시 여기는 교육목표에 부합된다. 또한 리컬전은 주어진 문제를 우리가 알고 있는 문제로 단순화시켜 해결하려는 문제해결의 한 방법으로 교육적인 중요성을 갖고 있다. 이러한 구조화 프로그래밍 기법과 리컬전 알고리즘들은 학생들이 미래에 만나게 될 여러 문제해결의 현장에서 아주 중요한 역할을 할 것이다. 말 언어는 이러한 프로그래밍 기법과 알고리즘을 말 언어를 통해 자연스럽게 익힐 수 있도록 설계되어 있다. 예를 들어 다음과 같은 이차방정식의 해를 구하는 프로그램을 보자. 이러한 알고리즘은 우리나라의 수학 교과서에서 순서도로만 지도되고 있는데, 이미 미국, 일본등의 수학교과서에는 순서도와 함께 파스칼 또는 베이식 언어등으로 알고리즘을 같이 지도하고 있다. 이제 말 언어로 표현된 프로그램과 일본 교과서에 나온 베이식 표현을 비교하자.

말 언어-(말 베이식)	
10	input a,b,c
20	let d = b*b - 4*a*c
30	if d==0 then : x = -b/(2*a) : print x : ifend
40	if d<0 then : p = -b/(2*a) : q = abs(sqr(-d)/(2*a))
150	print p, "+", q, "i", " ", p, "-", q, "i" : ifend
50	if d>0 then : s = sqr(d) : x1=(-b+s)/(2*a) : x2 = (-b-s)/(2*a)
80	print x1, x2 : ifend

```

GW- 베이식
10 INPUT A, B, C
20 LET D = B*B-4*A*C
30 IF D=0 THEN 100
40 IF D<0 THEN 130
50 LET S=SQR(D)
60 LET X1=(-B+S)/(2*A)
70 LET X2=(-B-S)/(2*A)
80 PRINT X1, X2
90 STOP
100 LET X=-B/(2*A)
110 PRINT X
120 STOP
130 LET P=-B/(2*A)
140 LET Q=ABS(SQR(-D)/(2*A))
150 PRINT P: "+": Q: "1", P: "-": Q: "1"
160 .ND
    
```



이 때, if 문의 then 다음에 여러 명령을 쓸 수 없는 GW-베이식 언어는 줄 번호 30과 40에서 보듯이 goto 명령을 쓰게 된다. 그러나, 말 언어의 프로그램은 『 if d=0 then : x=-b/(2\*a) : print x : ifend 』에서 보듯이 if 와 ifend 사이에 여러 실행문을 쓸 수 있어 불필요한 goto 문을 쓰지 않고 알고리즘을 표현할 수 있다. 이러한 말 베이식의 문법은 도스 5.0과 같이 제공되는 새로운 큐-베이식의 『 if 3<5 then : 실행문 모임 : ifend 』과 같은 명령도 유사하다.

베이식 언어는 100개가 넘는 명령문을 갖고 있어 베이식 명령 자체를 배우는 일도 쉬운 일은 아니다. 그러나 말 언어는 씨 언어등과 같이 20개 정도의 기본 명령문을 갖고 있으며 나머지는 필요에 따라 만들어 쓸 수 있다. 즉 말 언어의 명령문은 확장가능토록 설계되어 있는데, 예를 들어 말 언어 0.5 판에없는 원을 그리는 명령을 보자. 이 경우 말 언어에서는 주어진 변수 『가』에 대해 다음과 같이 원을 그리는 명령을 만들어 필요할 때마다 사용할 수 있다.

```

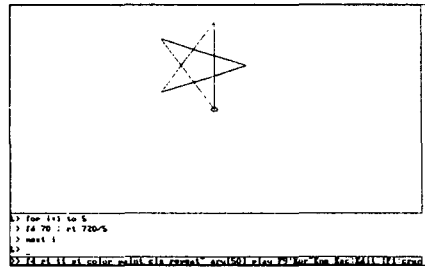
L> 함수 원(가) : 반복 36 하자 ( 가자 가 : 둘자 10 ) : 함수끝
    
```

이와 같은 명령에 컴퓨터는 『 function 원() is defined 』라고 답하게 된다. 이 말은 변수 『가』를 갖는 함수 『원()』이 컴퓨터 메모리에 기억되었다는 말이다. 이후, 원(10) 처럼 변수 『가』에 값을 주면 그에 해당하는 컴퓨터의 반응을 보게 된다. 이와 같이 원을 정의함으로 원에 대한 명령을 갖게 되고 말 언어는 언어자체가 확장되게 된다.

### Ⅲ장: “말” 언어를 통한 수학지도의 예

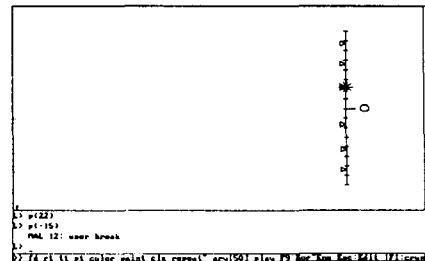
말 언어는 그래픽 상태에서 거북이와 산수와 수학에 관계되는 그림등을 그리며 자연스럽게 여러 컴퓨터 언어와 알고리즘을 배우고, 또한 이렇게 배운 알고리즘은 다시 산수와 수학학습등에 쓰이도록 설계되어 있다. 특별히 말 언어의 명령문은 기하, 대수, 확률, 함수, 그래프등을 학습하기에 유용하도록 설계되어 있다. 이 장에서는 말 언어를 통한 간단한 산수-수학의 지도의 예들을 제시한다.

도형에 대한 지도 : 초·중학교의 기본 도형은 변과 각으로 이루어져 있다. 말 언어의 fd 명령과 rt 명령은 각각 변과 각에 관한 그래픽 명령이다. 예를들어 오른쪽과 같은 별을 그리는 문제 상황을 살펴보자. 이 별은 50 발자국을 가고 720도를 5로 나눈 각만큼 도는 일을 5번 반복하여 얻어진다. 이 그림을 그리는 작업에 도형의 변, 외각, 내각등에 대한 산수 문제가 들어가 있다.



좌표에 대한 지도 : 말 언어에서 logo 명령으로 그래픽 화면에 가면 화면의 중앙에 거북이가 머리를 90도 방향(12시 방향)으로 하고 앉아 있다. 이 그래픽 화면은 거북이의 위치를 원점으로 x축과 y축의 좌표 평면으로 주소가 주어져 있으며, 거북이를 옮기는 tt 명령과 점을 찍는 pt 명령등을 할 때, 학생들은 좌표라는 언어를 써서 컴퓨터와 대화를 하게 된다.

오른쪽의 그림은 line.mal 프로그램의 실행화면이다. 학생이 p(22) 처럼 명령을 하면 좌표 22를 향해 거북이가 날라가서 만일 있으면 그 곳의 삼각형이 터지는 게임용 학습 프로그램이다.



변수에 대한 지도 : 말 언어에서의 변수는 마치 이름이 적힌 상자와 같은 일을 하는데, 이 상자는 숫자를 넣기도 하고 뺄 수도 있다. 다음의 예를 보자.

M> 가=10

M> 쓰자 가+10

이 때, 변수 「가」에 10이 담기게 되고, 그 다음의 명령에 의해 「가」에 있는 값에 10이 더해져서 화면에 20이 찍히게 된다. 다음은 for-next 문에서의 변수이다.

M> 반복 가=1 부터 2 : 쓰자 가 : 다음 가

1  
2

이와같이 변수 「가」는 1부터 2까지 차례로 담아지면 화면에 숫자를 쓰게 된다.

함수와 그래프에 대한 지도 : 수학에서의 함수는 주어진 엑스값에 와이값을 하나씩 대응시키는 법칙이다. 베이식에서는 이러한 함수를 DEF FNF(X)=X\*X 과 같이 정의한다. 말 언어에서는 베이식을 따라 def f(x)=x\*x 과 같이, 또는 씨 언어를 따라 def f(x) { return x\*x } 과 같이 두 가지의 서로 다른 방법으로 정의할 수 있다.

M> def f(x)=1/x

function f() is definiend

M> f(0)

MAL 2: divided by zero

이것은 역비례 관계적인  $y=1/x$  를 함수로 표현하고, 『 f(0) 』 명령을 한 것이다. 이 때 x에 0이 대입되기에  $1/x$  는  $1/0$  이 되어 분모가 0이라는 에러 메시지가 화면에 나온다. 다음은 f(0)을 0이라 하여 새롭게 f(x)를 정의한 것이다.

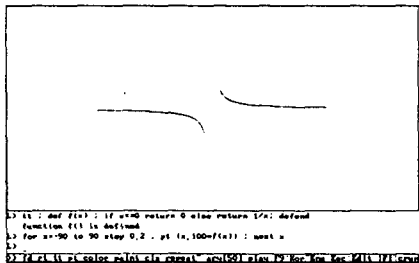
L> def f(x) : if x=0 return 0 else return 1/x : defend

function f() is defined

L> for x=-90 to 90 step 0.2

> pt ( x, 100\*f(x) )

> next x



여기에서의 함수 f(x)는 함수  $f(x)=1/x$  를 x가 0일 때 0의 값으로 대응시키도록

고친 것이다. 함수의 그래프는 『 (x, f(x)) 』 의 모임이다. 따라서 함수  $100*f(x)$  의 그래프를 그리려면 x를 변화시키며 『 pt ( x 100\*f(x) ) 』 명령을 하면 된다.

합동과 닮음 변환에 대한 지도 : 다음의 깃발 함수를 보자.

L> 함수 깃발(x)

> 가자 x

> for i=1 to 3 : 가자 x : 돌자 120 : next i

> 가자 -x

> 함수끝

이 함수는 주어진  $x$ 에 대해 깃발을 그린다. 이제 깃발(20)과 깃발(30)이 그리는 그림의 닭음비는 무엇일까? 또한 깃발(20) 과 『돌자 45: 깃발(20)』은 점대칭인가? 깃발(20) 과 『거북 20, 20: 돌자 45: 깃발(20)』은 합동인가?

방정식과 부등식에 대한 지도 : 닭과 개의 다리수가 합해서 34이고 닭과 개는 합해서 12마리이다. 닭은 몇마리인가를 보여주는 다음 프로그램을 보자.

```
L> for x=0 to 12
  >   if 2*x+4*(12-x) == 34 then print x
  > next x
```

교실에서는 등식의 성질에 의해 기계적으로 방정식을 풀지만, 여기서는  $x$ 에 값을 대입하며 등식이 성립하는  $x$ 값을 찾아 해를 구하게 된다. 이 때의 변수개념은 함수  $f(x)=2x+3$  와  $g(x)=2x+2$  의 그래프가 만나는 교점의  $x$ 좌표를 구하는 문제에 유용하다. 변수  $x$ 의 값을 변화시키며  $y$ 의 값이 같은가를 찾아보는 작업은 단순히 등식의 성질에 의해 방정식  $2x+3=3x+2$  를 기계적으로 푸는 것과는 다른 일이다.

```
L> for x=0 to 10
  >   if 2*x+3 == 3*x+2 then print x
  > next x
```

이와 같이 컴퓨터 언어의 변수 개념은 수학지도의 현장에 매우 유용할 수 있다.

정수의 성질에 대한 프로그래밍 : 수학 문제를 프로그래밍 해 봄으로써, 그 과정을 구체화하고 또 논리적으로 구성해 보는 수학적 사고를 이끌어낼 수 있다. 즉, 이미 이해하고 있는 개념이라도 프로그래밍을 하기 위해서는 그 개념을 분석하고 의식화하는 과정이 요구되는데, 이를 통해 이미 알고 있는 수학의 내용을 구체화하고 또 반성해 보는 기회를 가지게 된다. 다음은 2부터 100 사이의 소수를 모두 구하는 프로그램이다. 이를 응용하여 소수를 구하는 함수문을 만들거나, 또는 3행에 있는 『for i=2 to N-1』를 변화시켜 실행속도를 두 배 빠르게 할 수도 있다.

말 언어-(말 베이식)	
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

실행

```
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
```

(끝나는 수열?) 미해결 문제로의 도전에 대한 지도 : 『 주어진 정수  $a_1$  으로 부터 다음의 세 가지 규칙을 따라 만들어지는 수열은 반드시 1로 수렴하는가 ? 』 는 아직 풀리지 않은 수학기제이다. ①  $a_n$  이 짝수이면  $a_{n+1}$  의 값은  $a_n$  을 2로 나눈 값이다. ②  $a_n$  이 홀수이면  $a_{n+1}$  의 값은  $a_n$  에 3을 곱하고 1을 더한 값이다. ③  $a_n$  이 1이면 수열 만드는 것을 끝낸다. 예를 들어,  $a_1$  의 값이 7 이면  $a_2$  는 ②에 의해 22가 되고  $a_3$  는 ①에 의해 11 이 되고 .... 이렇게 계속하면 1을 만나게 된다. 1974년 까지  $a_1$  의 값이  $10^{40}$  보다 작으면 반드시 수열은 1로 끝남이 확인되었다. 이와 같은 문제는 수학이 이미 완성되고 해결된 것으로 인식하는 학생들에게 수학의 맛을 느끼게 한다. 이제, 주어진  $a_1$  에서 100항 까지 수열을 만드는  $f(n)$ 을 다음과 같이 정의하고  $n$ 에 7을 넣어  $f(7)$ 을 실행시켜 보자.

말 언어

```
1 def f(n)
2   for i=1 to 100
3     if n>1 then p = n/2 else goto 100
4       if p == int(p) then { n = n/2 : print " ", n }
5       if p <> int(p) then { n = n*3+1: print " ", n }
6   next i
7   100: defend
```

실행

```
M> f(7) ☐
22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1 ....
```

논리와 명제에 대한 지도 : 말 언어에서는 두 개 이상의 명제들은 『 and , or , not 』 등의 『 논리 연산자 』 를 써서 서로 연결될 수 있다. 이와 같이 연결되어 만들어진 명제의 참과 거짓은 논리에 의해 계산된다.

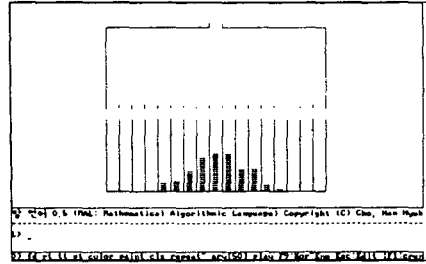
```
M> def f(a,b) {   for i=1 to a                               ☐
>                 if a%i==0 and b%i==0 then print i       ☐
>                 next i   }                               ☐
```

이 함수는 주어진 정수  $a$  와  $b$  를 동시에 나눌 수 있는 수 (공약수) 를 구하는 함수이다. 공약수를 찾으려면  $i$ 를 1부터  $a$ 까지 변화시키며  $a\%i$ 와  $b\%i$ 가 동시에 0인 것을 찾으면 되기 때문에 『  $a\%i==0$  and  $b\%i==0$  』 를 쓴다 ( $a\%i$ 는  $a$ 를  $i$ 로 나



는 나머지). 이제 『 f(20,30) 』 과 같이 a에 20을, b에 30을 넣고 함수를 실행시키자. 이 때, 화면에 나타나는 것이 a와 b의 공약수인가 확인해보자.

확률과 통계에 대한 지도 : 못이 바둑판 모양으로 박힌 판자를 세우고 위에서 여러 공들을 떨어뜨릴 때, 밑에 쌓이는 공들은 어떤 모양을 이루게 될까? 『 run "gong.mal" 』 명령을 하면 gong.mal 파일에 써 있는 프로그램이 실행되어 오른쪽과 같은 컴퓨터 실험 화면이 생기게 된다. 이 때, 공들의 분포는 이항분포를 이루게 되며, 시행횟수가 증가하면 이항분포는 점점 정규분포에 가까운 분포를 이루게 된다. 이제 이러한 프로그램등을 사용하여 할 수 있는 학습을 생각하자.

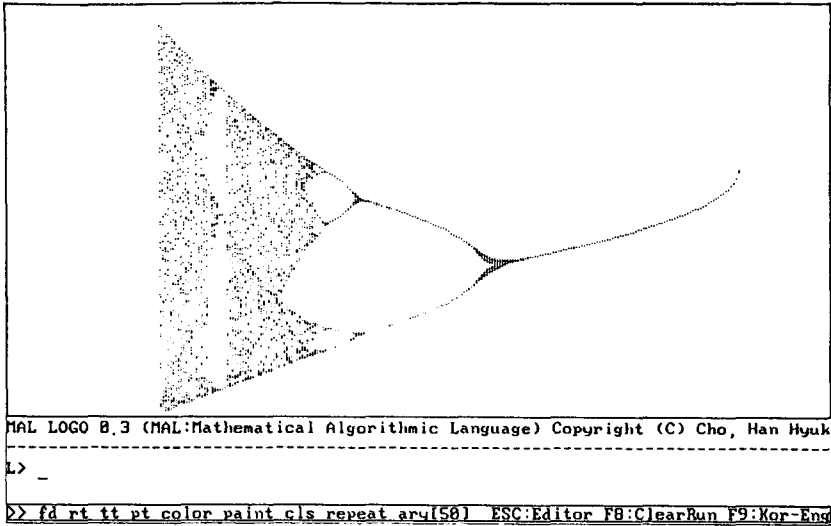


수열의 수렴성에 대한 지도 : 점화수열  $a_n = 6 / (5 - a_{n-1})$  은 n의 값이 커지면 어떻게 될까? 수렴값이 L이던  $L=6/(5-L)$ 이 되어 방정식을 풀면 L의 값이 두 개 구해진다. 그렇다면 어떠한 초기값  $a_1$  에 의해 어떠한 극한을 갖게 될까? 이러한 문제는 컴퓨터를 통해 학생들에게 수열의 극한에 대한 탐구를 가능케 하는 문제이다. 이와 비슷한 문제로 프로그래밍을 통해 프랙탈 기하와 극한개념에 대한 학생들의 흥미와 탐구심을 자극할 수 있는 다음의 문제를 보자. 먼저 이차함수  $f_c(x)=x*x+c$  을 생각하고, 주어진 c의 값에 대해  $a_0=c, a_{n+1}=(a_n)^2+c$  과 같이 귀납적 정의로 생기는 수열을 생각하자. 이제 c의 값을 변화시켜 여러가지 수열을 만들 수 있는데, 이러한 수열들의 수렴-발산을 컴퓨터 프로그래밍을 통해 탐구해 보자. 자세한 내용은 NCTM의 1991년도 yearbook 인 『Discrete Mathematics across the Curriculum, K-12』, 또는 NCTM과 협력하여 나온 『Fractals for the Classroom』을 참조하라.

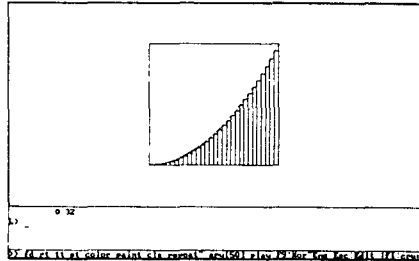
```

말 언어-(한글 말 베이식)
1 함수 올비트()
2     반복 가=0.25 부터 -2 증감 -0.01
3     x=0 :
4     for k=1 to 20 : x=x*x+가 : next k
5     for k=1 to 50 : x=x*x+가 : 점(100*가 + 100,50*x-5) : next k
6     다음 가
7 함수끝
8 함수 main() : 로고 : 거북 : 올비트() : 함수끝
    
```

↓ 실행결과 화면 ( run "orbit.mal" ) .



구분구적법의 지도 : 다음의 함수  $f(n)$  는 주어진  $n$  에 대해,  $y=x*x$  의 그래프와 0 부터 1까지의 구간을  $n$ 등분하여 만들어 지는 사각형들의 면적값을 대응시킨다. 예를 들어, 옆의 컴퓨터 실행화면에서 보는 것은 구간을 30등분한 사각형들의 모습과 면적의 합을 화면에 보여준다. 이제  $f(n)$ 의 수렴성을 살펴보자.



```


gubun.mal의 일부분
1  def f(a) { sum=0; for k=1 to a-1: A=a*a: sum=sum+k*k/(A*a); next k
2  return sum }
    
```

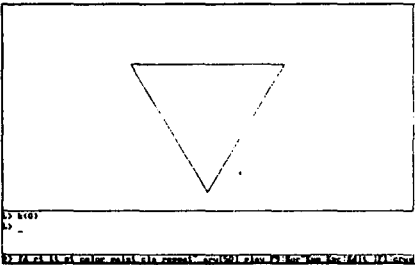
프랙탈 그래픽과 극한에 대한 지도 : 다음은  $x$ 의 값이 커짐에 따라 삼각형에서 점점 코흐(Koch) 프랙탈 커브로의 변화를 볼 수 있게 하는  $koch()$  함수이다.


```

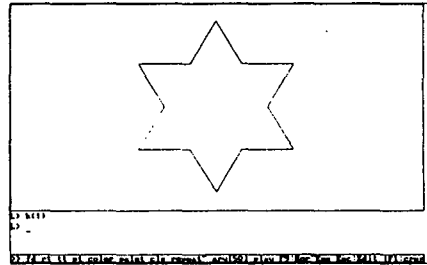
koch.mal의 일부분-(말 씨)
10 def koch(a,n) {
11     10: if(n==0) { fd a : return }
12     20: koch(a/3,n-1) : rt -60
13     30: koch(a/3,n-1) : rt 120
14     40: koch(a/3,n-1) : rt -60
15     50: koch(a/3,n-1)
16 }
17 def k(x) { tt 60,25,180): repeat 3 {koch(120,x): rt 120 } }
    
```


이제  $d=120$  이라 하고, 함수  $k(x)$  에서  $x$  값이 변함에 따라 이 함수가 그리는 도형이 어떻게 변화하는지 알아보자.  $x=0$  이면, 주어진 길이  $d(=120)$ 만큼 가서 120도 도는 과정을 3번 반복하게 되므로 한 변의 길이가 120인 삼각형이 그려진다. 만약  $x=1$  이면,  $d/3$  에 대해  $x=0$ 인 과정을 3번 실행하고,  $x=2$ 라면 .... 따라서, 함수  $k()$  는 다음과 같이 주변 길이의 합은 무한이고 면적은 유한값인 도형으로 점점 수렴해 간다. 이제 길이와 면적에 대한 식을 세우고 극한값을 구하여 보라.

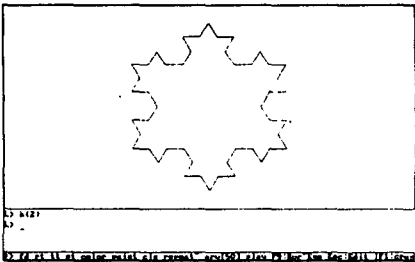
L> k(0) 




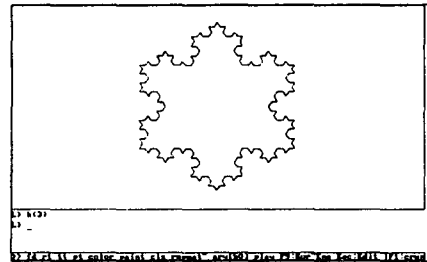
L> k(1) 



L> k(2) 



L> k(3) 



## 참고문헌

- [ 1 ] 조 한혁, 교육용 컴퓨터언어 말 0.5 학습 매뉴얼, 아가페교육연구소, 1992.
- [ 2 ] \_\_\_\_\_, 교육용 컴퓨터 언어 MAL 을 통한 Fractals 표현, 대한수학교육학회지, 1992.
- [ 3 ] \_\_\_\_\_, 교육용 컴퓨터언어의 설계에 대한 연구, 서울대 사대논총, 1992.
- [ 4 ] H. Cho, Learning Mathematics through the Educational Computer Language MAL, SNU J. Education Research (1992).
- [ 5 ] Aberson, and diSessa. Turtle Geometry: The Computer as a Medium for Exploring Mathematics. Cambridge, MA.: The MIT Press, 1986.

- [ 6] M. Barnsley, *Fractals Everywhere*. San Diego, CA: Academic Press, 1988
- [ 7] R. Devaney, "The Orbit Diagram and the Mandelbrot Set." *The College Mathematics Journal* 22 (1991) 23-38.
- [ 8] A. Engel, *Elementary Mathematics from an algorithmic standpoint*. Translated by F. R. Watson. Staffordshire: Kelle Mathematical Education Publications, 1984.
- [ 9] Hoyles, and Sutherland. *LOGO Mathematics in the Classroom*. London: Routledge, 1989.
- [10] Maddux, and Johnson. *LOGO: Methods and Curriculum for Teachers*. New York: The Haworth Press, 1988.
- [11] NCTM, *Computers in Mathematics Education, 1984 Yearbook*. Reston, VA.: The Council, 1984.
- [12] \_\_\_\_, *Discrete Mathematics across the Curriculum, K-12. 1991 Yearbook*. Reston, VA.: The Council, 1991.
- [13] S. Papert, *Mindstorm: Children, Computers and Powerful Ideas*. New York: Basic Books, 1980.
- [14] \_\_\_\_\_, "Personal Computing and its Impact on Education." In *The Computers in the School: Tutor, Tool, Tutee*. edited by R. P. Taylor. New York: Teachers College Press, 1980.
- [15] P. Thompson, "Mathematical Microworlds and Intelligent Computer-Assisted Instruction." In *Artificial Intelligence & Instruction*. edited by G. P. Kearsey. MA: Addison-Wesley Publishing Company, 1987.
- [16] Zazkis, and Leron. "Capturing Congruence with a Turtle." *Educational Studies in Mathematics* 22 (1991) 285-295.

---

참고 : 본 논문의 컴퓨터 실행화면은 말 언어 버전 0.3 과 0.5 를 통해 얻어진 것이다. 버전 1.0 미만의 말 언어 (말 언어 0.3 또는 0.5판) 는 상업성이 없는 경우 『 학교와 가정 』 에서 교육적인 목적을 위해 자유롭게 복사하여 사용할 수 있다. 그러나 저작권법에 의해 『 학교와 가정 』 외의 장소 (예: 학원이나 기업체 등의 장소) 에서 사용하려는 경우와 상업성을 위해 사용하려는 경우에는 반드시 저자의 서면 승락을 먼저 얻어야 한다 (전화: (02)-568-9391).

---