How had Logo effected to a pedagogue?

FARKAS Károly

John von Neumann Informatics College

Budapest

farkas.karoly@nik.bmf.hu

Abstract

What have I got by Logo?

The Logo language, and logo-pedagogy make us modern teachers. This study is useful for me not only in teaching but it has developed my knowledge, and my way of looking.

In this paper I list some examples, which were interesting and heuristic for me. These matters have come mostly from the turtle geometry, but there are some more in the field of a mother tongue, music, and even gymnastics.

Keywords

Logo-pedagogy, syntonic learning, LEGO-Logo, demonstration

1. Introduction

The Logo language is the main field, the main tool of Logo-pedagogy, which was created by Seymour Papert. The usage and the development of this paradigm mean to me the whole knowledge of the science of education.

Logo-pedagogy has verified most of my previous views, has cleared my doubts, has opened new vistas, and through its application, I have discovered numerous new facts / ideas.

Primarily, on the basis of Logo, I have realised that the use of Information Technology in the teaching process is the best way for the renewal and explorative development of pedagogy.

We can teach with the help of logo-pedagogy in a way which, most of the time, seems to be playing and entertainment to our pupils and students.

Logo has improved and systematized my knowledge not only in pedagogy, but in engineering, epistemology, and philosophy as well. During the cultivation of Logo, I myself could experience the pleasure of learning by discovering in several fields of science.

Below I will provide a few sketchy examples of how Logo has helped to unite separate elements of my knowledge into one system and how it has strengthened my conviction of systemic theory.

2. The computer is able to count well

The syntonic drawing algorithm of a regular polygon with `n` sides (we compare the drawing with own moving) is well known in the circle of Logo-users – on the vertex, it must turn from the moving direction with `360/n` degrees. This basic algorithm induces one thesis of computer use and, what is more, of the pedagogy of informatics-teaching, that is

When we use a computer, we have to make it do the counting instead of us.

When you draw heptagon the suitable algorithm goes like this:

```
repeat 7[fd 20 rt 360 / 7]
```

(Do not you calculate three hundred and sixty divided by seven!) As far as I am concerned, in the teaching of mathematics we could go by the Age of Information, when the application of computing machines, the computer in particular, has become common. Computer, calculator started to be tolerated in schools slowly. The quick spread of mobile phones has contradicted that argument: 'There is no calculator every time at hand.' The spread of calculating devices meant that the skill of calculating algorithms nowadays is only a mental exercise and attitude development, but not practical knowledge.

In primary school, pupils are taught: if you have to divide a fraction by fraction, you must multiply it with the invert of the divisor. It may be just as important: you have to take the divisor into parenthesis.

When we use Excel, we write a formula into the cell, and it is not advisable to count. Or it is more important to know – to use a differential rather than to know how to do a sum in his head. The programme Maple serves it by pushing a button. The engineer must know what kind of correspondence there is between stoop-curve and stress-curve, instead of making a differential. Nowadays, an economist at optimisation uses the Solver in Excel, instead of paper-pencil methods, he must find the mathematical model, explain the result.

3. We can play with re-entering polygon, and a convex polygon with the same algorithm, for example a hexagon and a hexagram.

During my early school studies, concave polygons appeared like some kind of untreatable, difficult things. Drawing them on paper takes a long time indeed. If we look at regular concave polygons, we recognise that it is possible to create them with the same command we use for a convex polygon drawn in Logo. The angle of every peak equals *360°/n*, where *n* is the number of peaks. The more we turn at a given peak, the less we have to turn at the next one. In Logo:

```
repeat :n / 2 [fd 50 rt 360 / :n + :d fd 50 rt 360 / :n - :d]
```

If the value difference of *d* `k*360 + 360/n`, and `k*360 - 360/n` is a part of an open interval, where `k` is a natural number, the polygon is convex; otherwise, it is concave. An increase in the value of `d` causes sharpening of the peaks. This sharpening turns over and the angles start to increase again on repetitive intervals. The edges cross each other. The method of 'kinetic' geometrical exemplification is a good way to demonstrate the close connection between regular polygons.

Computer experiments make the teaching of geometry much more effective. The procedures for demonstrating polygon families are as follows:

```
to polygon :n :a :d
  repeat :n / 2 [fd :a rt 360 / :n + :d fd :a rt 360 / :n
  - :d]
end
to kin: n :a :f
  make "s 0
  repeat 100[polygon :n :a :s wait 10 cg make "s :s + 360
  / :f / :n]
end
```

For example: if `n=6`, adjusting the value of `d` by 20 degrees at every step, we get the following polygons as results:

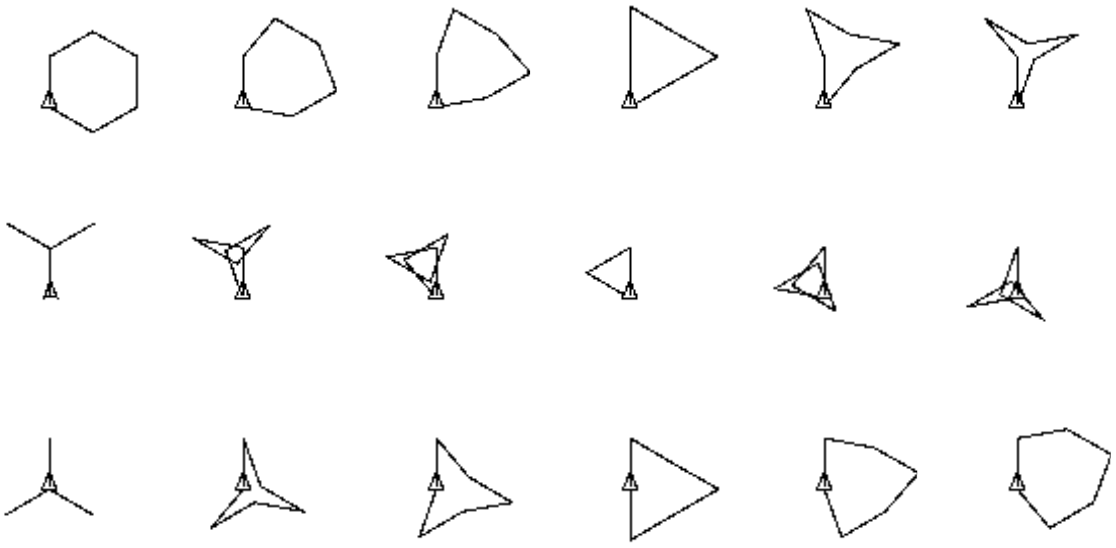

Figure 1. Regular polygons with 6 edges

These shapes repeat periodically. The film like presentation of polygon series is a model of a throbbing system (one possible model of our Universe).

As we can find triangles in the family of hexagons, there are hexagons between regular 12-peak polygons. Some of the 12-peak polygons can be seen in figure 2.

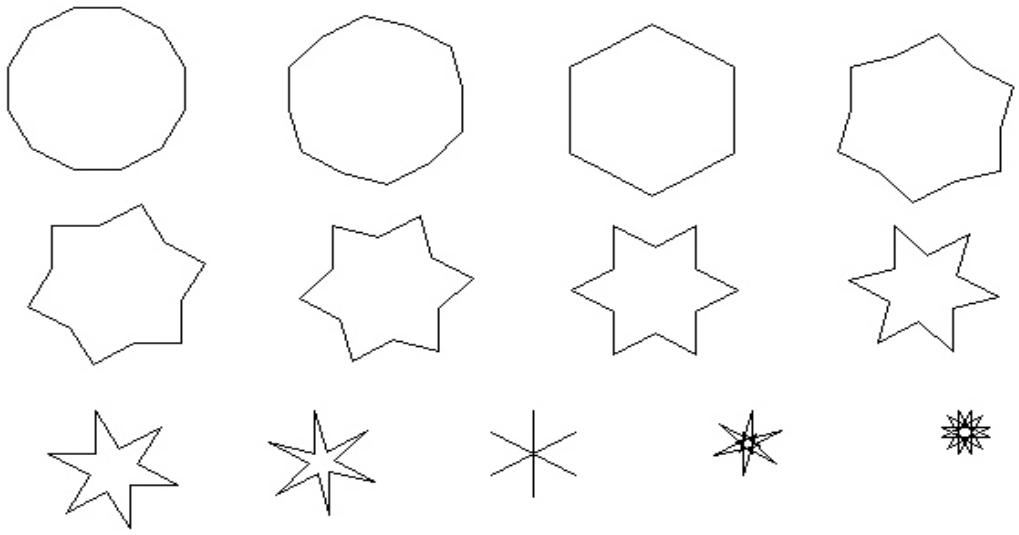

Figure 2. 12-peak polygons

4. We can find several kinds of algorithm to draw a circle in turtle-geometry as well.

Drawing a circle by the following two algorithms is more syntonical than the ‘repeat many times: one step forward, one turn’ method:

One of these procedures can be implemented by a child from grade 3: to step forward by the amount of the circle's radius, place the pen to the paper and draw a little, then lift the pen, go back to the middle and turn, then repeat these steps many times. In Logo:

```
repeat many times
[pu fd radius pd fd a little pu bk radius + a little rt a little]
```

The line's thickness of the circle can be adjusted easily this way, because it depends on the amount of movements forward with the pen placed on the paper. This algorithm is more oriented to the 'look-round-ideology' of a kid.

Another circle drawing algorithm is based on the connection (social model!) between the turtles. (Farkas, 2003) The method features two turtles. 'Adam' is tied to 'Eve'. (Eve is in front of Adam and the distance is constant between them at every time.) Adam is spinning, so Eve is circling. In MicroWorlds language we only have to animate the turtles: we name them, command them to repeat simple moves, then we 'revive' them by clicking.

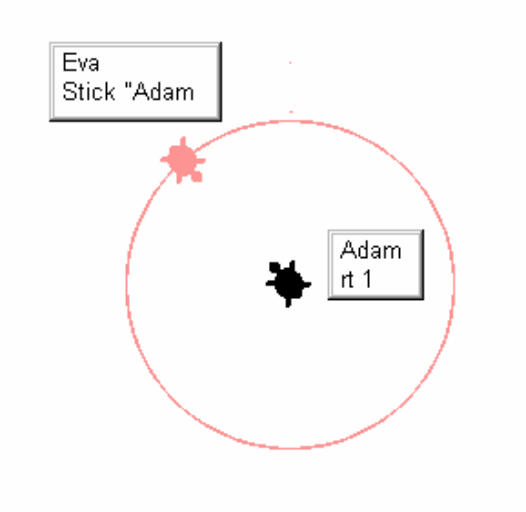

Figure 3. Eve circles around Adam

Procedure ***stick*** exists only in the Macintosh version of MicroWorlds. Let's create it in the IBM version! If a command is missing, we can write it for ourselves. We can teach the turtle, form the software to our, and the students needs. A possible implementation:

```
to stick :a
 ;I stick the currant turtle to :a turtle
 towards :a
 make "d distance :a
 make "alpha ask :a [heading]
 make "x ask :a [xcor]
 make "y ask :a [ycor]
 setpos list :x + :d * sin :alpha :y + :d * cos :alpha
end
```

5. You can draw a spiral, cycloid without profound mathematical knowledge.

"For most people, nothing is more natural than that the most advanced ideas in mathematics should be inaccessible the children." (Papert, 1993 in sec. ed. p 161)

More complex mathematical curves can be taught in an understandable way if we superimpose two motions. For example: If 'Eve' drifts while circling around Adam, then she draws a spiral line.

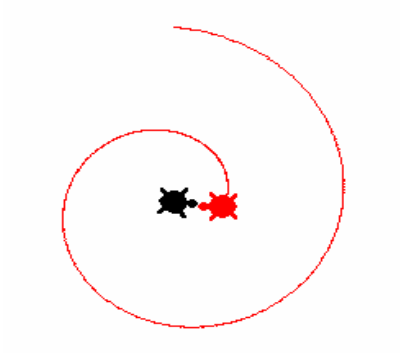

Figure 4. Spiral

Or if one turtle is walking, another is circling, and the third does the two moves together, then this third turtle draws a cycloid curve.

6. The world of spindles

In the book 'Present Physics', Feynmann writes that the most frequent shape is the spiral in our Universe. I thought about this idea. The orbs are nearly spheres. Are not these planets moving on a circle, elliptical or at least somewhat cyclic orbit? At first I did not take into account that the centripetal circling is only a relative motion. The absolute space is dilating in every direction, so these orbits are spiral curves in reality. The frequency of the spiral shape is a consequence of the dilating universe. The following algorithm gives spirals:

```
repeat many times [fd :a rt constant, make "a more]
```

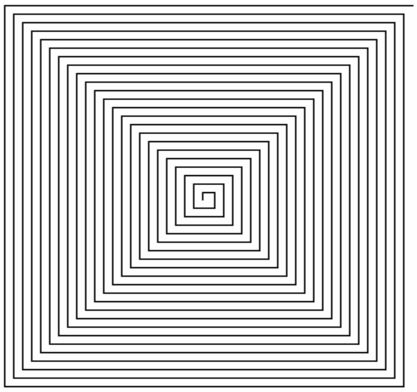

Figure 5. The 'rectangular1 spiral. The value of the constant is 90

The varying of the constant showed that if the value is arbitrary, meaning that the angle of turn is not special (divider of 360° 4, ore 45°, 60°, 72°, 120°), then the spiral draws up. Look at figure 6, where the angle of turning is 91°.

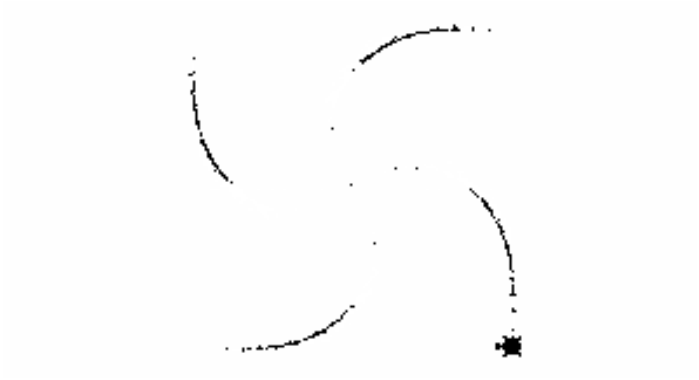

Figure 6. 'Implicit spiral'. The value of constant is 91

We get spirals using the inverse of the previous algorithm, where we increase the angle of turning, but the step size is constant. Probably we were the first to call spirals turtle roses, spindles in Hungary. (Farkas, 1994a, 1996a). An algorithm with varying step size and turn angle also gives a spiral. Analysing these curves can be an exercise for an informatics and/or mathematics class.

7. The sinus curve is the degeneration of a cycloid.

The stretched cycloid can be constructed by the addition of two motions:

```
to vec :a :b
 ;:a transporting, :b relative mowing, first-black turtle
 resulting, second-red transporting, third-blue relative
 curve
 ask [t1 t2] [run :a]
 ask [t1 t3] [run :b]
 vec :a :b
end
```

We get a stretched cycloid by the following command:

```
vec [setx xcor + .5][fd .25 rt 1]
```

I was continuously modifying cycloid parameters, when I realised that the sine curve is a special cycloid. The horizontal drift has to be infinite, or the horizontal part of the circling motion has to tend to zero. The second case can be represented if the plane of red drifting motion is perpendicular to the circling motion. For the presentation, we use a 4th, green turtle. The black sine curve is the resultant of the horizontal drifting motion and the blue relative motion. After these, the sine curve can be constructed this way:

```
to alter
t4’fd 1,57 rt 1 ask [t1 t3] [sety ask "t4 [ycor]]
end
```

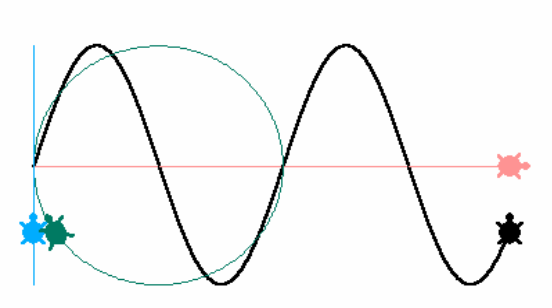

Figure 7. Sine curve: vec [setx xcor + 1][alter]

The additional green turtle (turned into the image plane) is circling, the blue turtle takes its actual vertical projection and evades axis X (the horizontal red line) with the same rate. We can recognise that the alternating motion of the blue turtle has pulsing speed. By varying the parameters of the algorithm, the function transformation can be demonstrated easily.

8. People are not able to choose a random number out of their heads.

People usually do not know enough the random. Numbers do not feel such a minimal chance to reach a direct hit on the lottery. There is an old didactical trick: when I ask my pupils to write ten random numbers out of head and also ten numbers taken from the computer. Taking the two, it is not hard to see which of them was generated by the people. Not only the pupils, but students as well endeavour to use all numbers just. No, it is not the reality; the random is not equitable in the short period. In a lottery, there are many winners - I have never been one. I found to demonstrate the random in the best way, when a turtle weaves to the right during the walk and to the left in a random value. On the third of my pupils made the following algorithm:

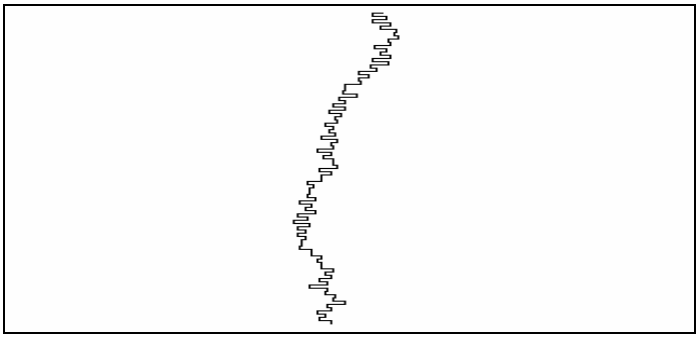

Figure 8. The presentation of random

After the turtle left the screen on top, it appears at the bottom. In figure 9, the turtle surrounds its world more times.

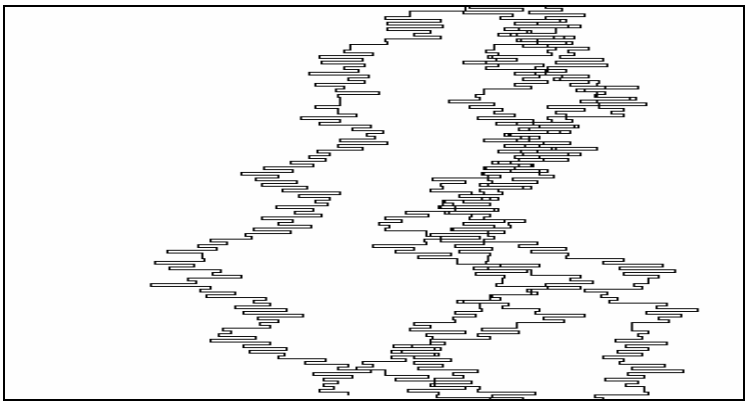

Figure 9. Random walk

If the random were chosen evenly when it weaves to the right-left, the random-tree would be straight. We can think, and we could get the answers to view this model:

Is it possible to have the return-point the same as the starting-point?

Get the turtle to the margin of the screen?

Will equivalent touch every point of the screen?

Will the whole screen get black through the walk ?

9. Tégla-logo. Symmetry is invariable during the change of dimension.

We invented the name 'tégla-logo' (It means Brick-logo) for the Hungarian version of LEGO-Logo. LEGO-Logo has a duplex meaning. It means the controlling of models made of LEGO elements, which is many times realised by one of the Logo implementations. LCSI has always paid attention to constructive pedagogical approaches of robotics. As LogoWriter has a LogoWriteROBOTICS version, a MicroWorlds Ex Robotics version also exists. It's easy to build robots with LEGO, and Logo is an optimal language to control them.

The other meaning of LEGO-Logo is the algorithm of building structures from LEGO or LEGO like elements put in the Logo language. We suggest the use of Brick logo for many aspects. It improves the algorithmic skills, structural thinking, 3D vision and manipulative skills together.

Build means placing a brick. ***Forward*** means the movement in the direction of the brick's 'face'. ***Up*** means the elevation. A brick is four units long and one unit tall. Build steps from bricks:

```
Repeat 3[build forward 2 up 1]
```

To change the sign of the 'horizontal' dimension, we use *bk 2* instead of *forward 2*, so the building will be the mirror image of the original one. Changing the sign in the other dimension means the use of *down 1* instead of *up 1*.

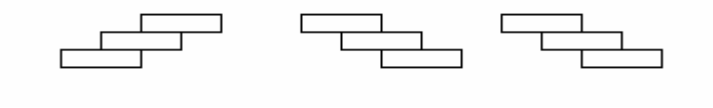

Figure 10. Symmetry

To demonstrate this and to rate the results is really easy, because we are building from real bricks, so the right solution is completely visible. (Farkas, 1996b).

10. The algorithm of chimney-bricklaying is definable in two lines.

The algorithm of chimney building is really spectacular, elegantly short in Brick logo.

```
To first line
repeat 4[build forward 4 rt]
end
To first lift
up 1 rt
end
To second line
repeat 4[build forward 4 lt]
end
To second_lift
up 1 lt
end
To element_of_chimney
first_line first_lift second_line second_lift
end
```

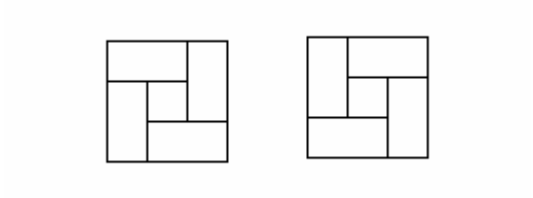

Figure 11. Chimney

11. The prime numbers are the basis for the numerical system

Experimenting with turtle roses or spindles shows the significance of prime numbers. We had publications on this several times. (Farkas, 1994a)

```
to inda :d
 make "s 1
 repeat 2000[fd 5 rt remainder :s 360 make "s :s + :d]
```

```
end
```

Figure 12. Spindles d = 3, d = 7, and d = 21 (three times seven!)

12. Logo, including programming, is not only mathematics

Logo has not improved only my mathematical skills, but affected the way of my interpretation of language thinking. I found the translation of commands into Hungarian essential. I have experienced its importance many times. The ***bk*** command's rough translation '**hátra**' ('BACK') resulted many times in little children turning around and taking the way back like that. This is why we found the expression '**hátrál**' ('RETREAT') more suitable. Another interesting experience was the case, when I tried to explain the ***pen up***, and ***pen down*** commands for undergraduate training-college students. I could not show them the turtle robot, which lifts and lowers its pen. The candidates interpreted the commands with reverse meaning. During their teaching exercises ***pen down*** meant to put down the pen and stop writing.

We created the Hungarian translation of the Logo instruction set for the most frequently used Logo version, Comenius Logo. T-Logo also utilizes these instructions, which is the Hungarian version of LogoWriter, made by Éva Törtely. We apply these practical Hungarian commands in translation of MicroWorlds Logo. (Farkas, 1994b)

Anyone who masters the concept of Logo can create his own language version, adjusted to the given needs of a group of students.

The easiest commands to translate are those without any parameters. In that case, we only have to create a small dictionary.

```
To szivacs
 cg
end
```

(szivacs means sponge.) When translating commands with parameters, we have the chance to adjust the unit sizes. E.g.: small kids can work with cake slices, later with the clock, and the older ones can use the angles.

```
To jobbra :s
 rt :s * 30
end
```

('**jobbra**' means to the right) Other Logo-primitives are also easy to translate for a teacher who has ever used/ tried Logo.

Logo was a useful tool for creating reading improvement of software. The 'Dynamic reading' diskette contains exercises and games written in Logo. All the reading improvement exercises I meant to do with computer were possible to be written in the Logo language. This proved that Logo is not only a game but a useful tool for my job.

Playing with Logo games also improved my musical skills. Writing these music programs improves the audition, and the sense of rhythm. Several teachers recognized that students could memorize tunes more easily if the computer plays those tunes too. There are exercises in music education, which we can program in Logo optimally. There is the tune teaching program, where students have to replay audible tunes by the computer. It is really easy to implement a program like that in Logo.

The algorithmic study of the 'waterfall' juggler trick is presented in Papert's book, Mindstorms. (Papert, 1993) Some of us have tried and can prove that complex motions are easier to memorize if they are studied as algorithms. Those, who practiced gymnastics, or other sports, or played an instrument, can tell you that artistic motion is not simply a physical

activity. Assembling the Magic Cube can be aided by algorithms expressed in the Logo language. Using this aid can show us the way to solve the problem on our own which helps us to understand the micro-world of the cube and improves our 3D vision. (Farkas, 1995)

References

Farkas K (1994a), *The World of Turtle Roses,* Ivan Kalaš ed.: Acta Didactica Universitatis Comeniae Informatics, Issue, Bratislava, Comenius University Slovakia, 63-67.

Farkas K (1994b), *LogoWriter, Programnyelv gyerekeknek* (LogoWriter programme language for children), Múzsák-Reál Könyvkiadó, Budapest.

Farkas K (1995) *First steps in education of Informatics.* Eurologo'95, 5th European Logo Conference Birmingham, England, 65-70.

Farkas K (1996a), *Robot Games and Turtle Roses. Teaching Logo in Hungary,* Learning and Leading with Technology (USA*)*, 62-64. and http://www.jio.hu.

Farkas K (1996b), *Játékos informatika (Playfull informatics),* Dissertation. D17799 I–II.

Farkas K (2003), *Logo and native language. Intrinsic procedures of some curves.* Proceedings of the 9th European Logo Conference, Porto, Portugal, 69.-79.

Papert S (1993), Mindstorms. Children, Computers, and Powerful Ideas, Basic Books.