THE-MORE-IT-CHANGES-THE-SAMER-IT-GETS PRINCIPLE IN THE CONTEXT OF MATHEMATICS AND INFORMATICS EDUCATION

George Gachev, Evgenia Sendova, Iliana Nikolova

University of Sofia, Bulgaria: gachev@fmi.uni-sofia.bg, iliana@fmi.uni-sofia.bg

Bulgarian Academy of Sciences: jsendova@mit.edu

Abstract: *The paper presents some observations from the authors' teaching experience with teachers and students in a Logo and a Toon Talk environment. The effect of having different representations of the same notion is explored in the context of some important mathematics and informatics concepts and structures. A comparison is made between using visual (TT-like) and script based programming languages for educational purposes.*

1. The importance of having different representations

Plus ça change plus c'est la meme chose is a French saying which Douglas Hofstadter [2] finds to be a very meaningful educational principle. His own translation is "The more it changes the samer it gets" and he interprets it as follows: *The more different manifestations you observe of one phenomenon, the more deeply you understand that phenomenon, and therefore the more clearly you can see the vein of sameness running through all these different things. Or put another way, experience with a wide variety of things refines your category system and allows you to make incisive, abstract connections based on deep-shared qualities.*

The idea that *different representations show a concept in a different light, highlighting some of its aspects and hiding others* is emphasized by many educators [8].

WebLabs, a three-year European research project on the use of programming and web-based collaboration in mathematics and science education, is exploring the hypothesis that at least some of the apparent complexity of mathematical and scientific ideas is due to the *representational infrastructure* with which they are expressed [11]. The symbols traditionally used to express mathematical ideas, and the rules for transforming them, are an essential part of the way of thinking in these two domains. WebLabs research is asking whether these representations are unique, or whether it is possible to design a system where students express and construct their ideas in novel ways, in order to make them more accessible [12].

2. The computer- and the web environments

WebLabs uses an environment for visual programming called *Toon Talk* in which the source code is animated thus allowing for abstract computational concepts to be represented by concrete analogues, instantiated in cartoon-like characters [10]. *ToonTalk* has some unique features suitable for visualizing and exploring mathematics concepts and ideas when working with junior-high school students. The mathematical activities are integrated in a natural way with cultivating some programming skills. The programs in *ToonTalk* take the form of animated robots, which can be named, picked up and trained to perform a certain sequence of elementary steps. A bird is the metaphor for the output of a procedure. After the training, the robots run *forever* if the initial conditions are satisfied. Here is how a program for generating the sequence of the natural numbers looks like (Fig. 1).

The robot is first trained to add 1 to 1

The concrete first addend in the robot's *thought bubble* is then erased to generalize the process

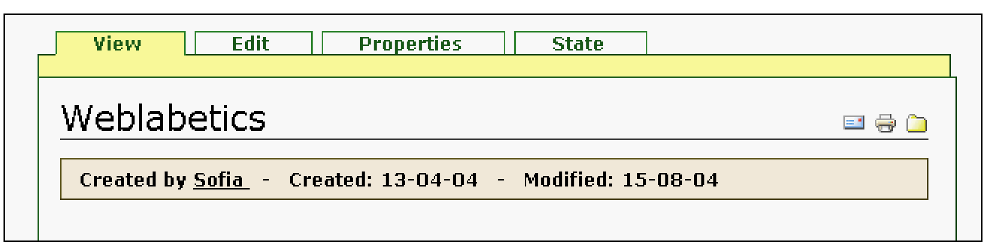
Now the robot can add 1 to any current number in the box and then pass the result to the bird (*forever*)

Fig. 1. Training a robot to count

When using *ToonTalk* as a means for modeling, the students learn how to work with visual computer environments in a natural way. In the context of carefully designed educational activities they gain knowledge about important processes and phenomena from mathematics and science, and compare their understanding with the rest of the participants. The communication is carried out by the so-called *Webreports* – a specially designed concept enabling the young learners to share and discuss the problems they have solved, and even more interesting – the problems they have formulated and implemented by means of *ToonTalk* robots.

3. *Weblabetics* - a graphical scripting language designed by kids to represent TT constructs

The language the students are expected to publish their web-reports in is English but the native tongue is also accepted for local communications. On one hand this arrangement supports the feeling of the participants of being part of an international community, on the other - it could be a reason for frustration. Let us illustrate this phenomenon with excerpts from a web-report [18] created by a group of Bulgarian kids participating in WebLabs:



When browsing in search for interesting sequences we had very unexpected experience. We moved step-by-step through the sequences suggested by Nikmous, Kiriakos, Irakli - all in Greek. The sequences are very *clear but when the comments are in a language, which we don't understand, it is very annoying. So our teacher asked us:* ***Can you think of a way to express ToonTalk ideas so that anyone could understand them?*** *Yana suggested to use pictures for representing the ToonTalk characters and drew some on the board* (Fig. 2):

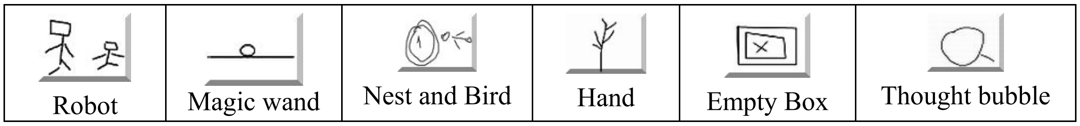


Fig. 2: Pictures representing ToonTalk tools

The teacher challenged us to translate our Counting Robot in the new language. We all thought that this was easy but soon realized that we didn't have symbols for actions in our alphabet (or rather – weblabetics). So we added arrows for "puts" and "takes". Here is the Counting Robot in weblabetics (Fig. 3):

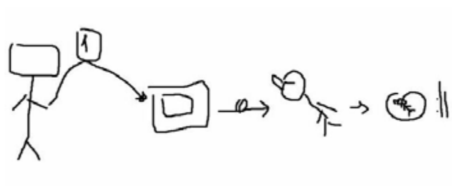


Fig. 3: TT program code (Counting Robot) in *weblabetics*

*Isn't this clear for everybody? Well, just in case you lack the experience: A robot puts 1 in a box, then copies the content, gives it to a bird, which puts it in its nest. Afterwards everything is repeated. Do you see the "***:||***" sign at the end – this is the music symbol for a repetition – Peter thought of it! In short, this is our old friend – the Counting Robot (in new clothes…)*

We hope that now it would be easier to talk about ToonTalk and our ideas to everyone in the WebLabs project. Our teacher told us the story of the Babylonian tower - a common language for everyone is more effective than many languages for a few.

Children were faced with a typical e-learning problem while trying to learn collaboratively over distance – the language problem. In an attempt to overcome it, they reached the idea of designing a graphical scripting language for visual programming.

Using different representations of the same phenomenon proved to be very fruitful in a slightly different situation – when the students had to prove that their robots produce the same sequence of numbers.

4. Proving the equivalence of robots

After hearing from Yishay Mor, a researcher from the UK WebLabs team, that there is a new challenging sequence on the "Guess my robot" web page [13] published by a Portuguese girl (Rita), the Sofia teachers (the first and the second authors) asked the students to solve it as a homework. Two students, Nasko and Ivan, took the challenge and reported to the class that they had guessed the rule and had even built robots generating it. The teachers decided to use this as a basis for the topic "Different representations of number sequences" [16].

Teacher: *Nasko, please try to translate your robot in algebraic language?*
Nasko: *Here is the relationship between the consecutive terms (Fig. 4)*

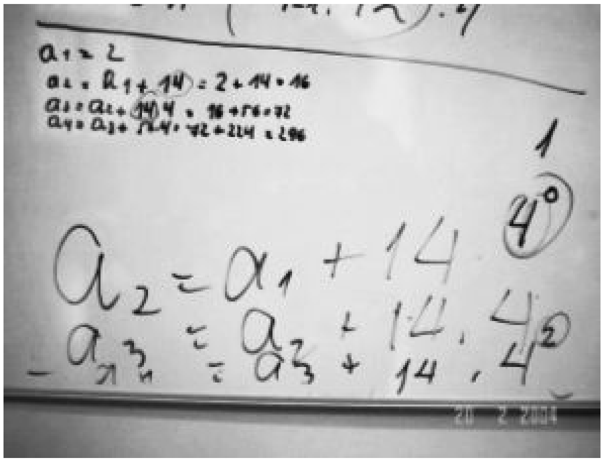


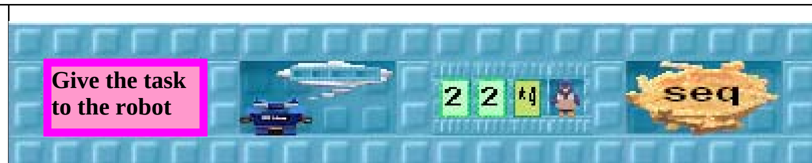
Fig. 4. Nasko's robot in algebraic language

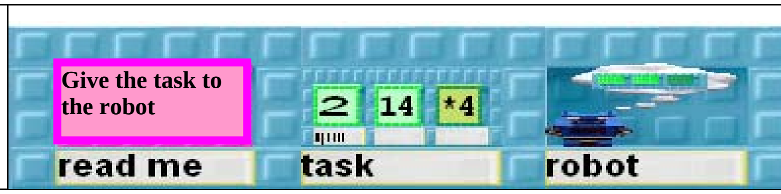
Fig. 5. Teddy translating from algebraic into a "robot" language

Vessela: *Ah, it means we have to present 1 as* 4^0
Teacher: *Rita's robot in algebraic language looks like this:*
$y_1=2, \quad y_n= 4(y_{n-1} + 2)$
Are these sequences the same?
Teddy: *Ah, this looks much easier. Let me try to translate it in a "robot language"* (Fig. 5)
Teacher: *Do you think that your and Nasko's robot will produce the same sequence?*
Teddy: *Sure! Absolutely!*
Teacher: *How do you know? Take for example the sequence 3, 5, 7, ... Which is the next term*?
Teddy: *9, of course!*
Teacher: *Do you agree, George?*
Teacher: *I am thinking of 11.*
Teacher: *Of course!* (Both of us are thinking of a subsequence of the prime numbers (of course!!!))
Teddy: *Oh, I see, so you could extend the sequence in more than one way... – adding twice 2, and then adding twice 4, etc. So, I could write it in algebraic way as follows:* $y_n = y_{n-1} + 2^n$
Vessela: *What about* $y_n = y_{n-1} + y_{n-2} - 1$?
Ivan: *My robot also uses 2 previous terms :*$y_{n+1} = y_n + (y_n - y_{n-1})*4$ *and its first numbers coincide with Rita's ones.*
Teddy: *Oh, how could we compare so many robots?*
Teacher: *I am not telling you! But I hope to hear your ideas next time. Please consider for the following questions from our English partners Celia Hoyles and Richard Noss*:

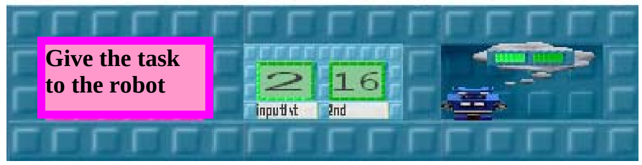
- *How did you guess the rule of Rita's sequence?*
- *Which is easier for you – to translate from the robot language into algebraic one, or vice versa?*
- *How could we check if two robots produce the same sequence?*

Thus the sequences of Rita and Nasko which are to be proved equivalent could be described in algebraic and in a robot language as follows [7]:

Rita: $y_{n+1} = (y_n + 2).4$ $y_0 = 2$	

Nasko: $y_n = y_{n-1} + 14.4^{n-1}$ $y_0 = 2$	

And the sequence of Ivan is described as follows:

Ivan: $y_{n+1} = y_n + (y_n - y_{n-1})*4 = 5y_n - 4y_{n-1}$ $y_0 = 2$ $y_1 = 16$	

In essence both Ivan and Nasko work after the formula:

$$y_{n+1} = (y_n - y_{n-1})*4 + y_n$$

The only difference being that Nasko has a_1 and $(y_2 - y_1) = d_1$ and constructs y_2, whereas Ivan has y_2 and y_1 and constructs $(y_2 - y_1) = d_1$.

Or, in Logo notation:

```
to Rita  :current
   make "intermediate :current + 2
   make "next :intermediate * 4
   print :next
   Rita :next
end

to Nasko   :previous  :difference
   make “current :previous + :difference
   make "next_difference :difference * 4
   make "next  :current + :next_difference
   print :next
   Nasko :current :next_difference
end

to Ivan :previous :current
   make "difference :current - :previous
   make "next_difference :difference * 4
   make "next  :current + :next_difference
   print :next
   Ivan :current :next
end
```

So, the sequence of Rita is generated by the following three instructions:

```
Rita 2

Nasko 2 14

Ivan 2 16
```

In an e-mail conversation with Yishay he wrote:

The problem with Nasko's robot is that it doesn't really output the result, so there's room for interpretation regarding when the actual sequence term is produced. Still, to match Rita's sequence, I prefer this reading:

```
to Nasko :previous  :difference
   make “current :previous + :difference
   make "next_difference :difference * 4
   print :current
   Nasko :current :next_difference
end
```

I believe yours is identical to two iterations of this one.

Here follows the proof of the equivalence of the robots by means of the theory of difference equations.

The above sequences could be considered as solutions of homogeneous (Ivan) and inhomogeneous (Rita, Nasko) difference equations of first (Rita and Nasko) and second (Ivan) order, respectively, described generally as follows:

(1) $a_k y_{n+k}+a_{k-1}y_{n+k-1}+\ldots+a_0 y_n=b,\ \ a_0 a_k \neq 0.$

Each sequence $\{y_n\}^{\infty}_{n=0}$ such that (1) holds for any k+1 consecutive members of it is called a *solution of* (1)

The rules of Rita and Nasko are first order non-homogeneous difference equations. Their solutions can be found easily by using recurrence technique.

For the Rita’s rule we find:

(2) $y_{n+1}=(y_n+2).4$

$y_{n+1}=4y_n+8$

$y_n=4^n y_0+8(4^n\text{-}1)/(4\text{-}1)=4^n y_0+(8/3).(4^n\text{-}1)=4^n(y_0+8/3)\text{-}8/3$

$y_0=2$

$y_n=(14/3).4^n\text{-}8/3$

For the Nasko’s rule it follows:

(3) $y_n=y_{n-1}+14.4^{n-1}$

$y_0=2$

$y_{n+1}=y_n+14.4^n = y_{n-2}+14.\ 4^{n-1} = \ldots = y_0+14(1+4+4^2+\ldots+4^{n-1}) =2+14(4^n\text{-}1)/(4\text{-}1)= (8/3).(7.4^{n-1}\text{-}1)$

Thus the solutions of the equations of Rita and Nasko are equivalent since:

$(14/3).4^n\text{-}8/3=8/3.(7.4^{n-1}\text{-}1)$

The Ivans’ rule is a second order homogeneous difference equation:

(4) $y_{n+1}=y_n+(y_n\text{-}y_{n-1}).4$

or, which is the same:

$y_{n+1}\text{-}5y_n+4y_{n-1}=0$

To find its solution we use the characteristic equation:

$z^2\text{-}5z+4=0$

whose roots are: $z_1=4,\ z_2=1$.

Thus the general solution of equation (4) is:

$y_n=C_1.4^n+C_2.$

The constants C_1 and C_2 are determined from the initial conditions
$y_0=2; y_1=16:$
$y_0=C_1+C_2=2$
$y_1=4C_1+C_2=16$

$C_1=14/3$
$C_2= -8/3$

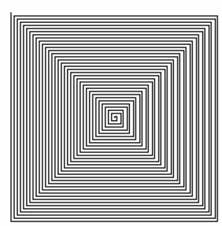
$y_n=(14/3).4^n-8/3=(2/3).(7.4^n-4)$, or

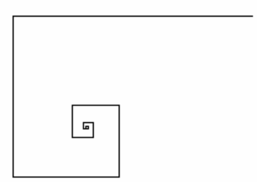
$y_n=(8/3).(7.4^{n-1}-1)$

Since the three above difference equations have the same solution the robots of Ivan, Nasko and Rita are equivalent.

It is important for the kids to see that the questions the researchers ask them are not always trivial and sometimes we, the teachers, don't know the answer. The moral for them is to feel that each representation of a sequence has its own advantage and that the ability to "jump" from one to another might be very insightful. In this sense instead of saying "the robot computes the sequence" we could say "the robot IS the sequence"… [7]

The sequences the Weblabs kids generate in order to challenge their peers are rarely pure arithmetic or geometric progressions. But when drawing spirals in Logo it is a very good idea to discuss with the students that they could be considered as a graphical representation of an arithmetic (geometric) progression:

to arithmetic_spiral :a :d
    if :a > 200 [stop]
    fd :a rt 90
    arithmetic_spiral :a + :d :d
end
	

to geometric_spiral :a :q
    fd :a rt 90
    geometric_spiral :a*:q :q
end
	

If a sound (whose frequency depends on the length of the segment being drawn) is introduced in the above procedures one can *hear* the difference in the growth of the two sequences. Another interesting spiral is the so-called "golden spiral" where the size of the revolutions grows with the golden ratio (Fig. 6). It can be constructed by inscribing circular arcs of 90 degrees in a sequence of squares as follows: start with a rectangle whose sides have the golden ratio, then divide the rectangle into a square and a rectangle (where the new rectangle also has the golden ratio between its sides, and so on ad infinitum (Fig. 7) [9]

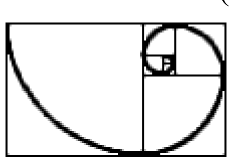


Fig. 6. The golden spiral sections

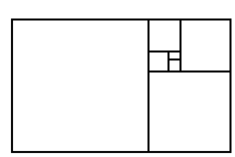


Fig. 7. The golden spiral

It is interesting for the students to come across another representation of the golden ratio – as a limit of two consecutive terms of the Fibonacci sequence.

5. Fibonacci sequence and some of its variations:

The Fibonacci sequence:

1, 1, 2, 3, 5, …

in which every term from the third on is a sum of the two previous can be easily modeled in Logo by using some operations on lists [*ibid*]:

```
to sequence :list
   output lput :list (last :list) + last bl :list
end
```

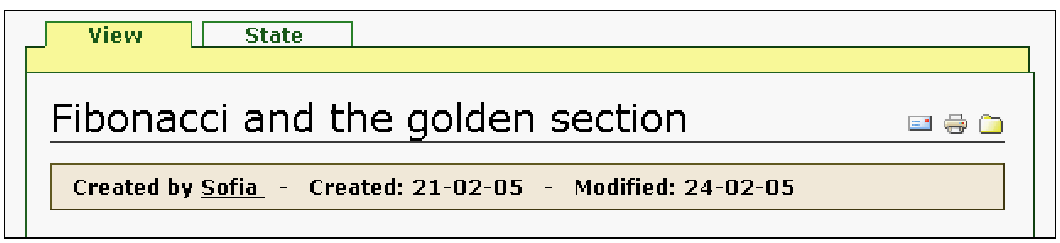
If we give a list of the first *n* Fibonacci numbers as input the output will be the list of the first *n*+1 Fibonacci numbers. The ratio of two consecutive terms could be found by the following operation:

```
to ratio :list
   output (last :list)/last bl :list
end
```

Plotting the graph of this ratio as the input list of the Fibonacci numbers grows provides the students with a better understanding about its behavior, viz. tending towards the golden ratio.

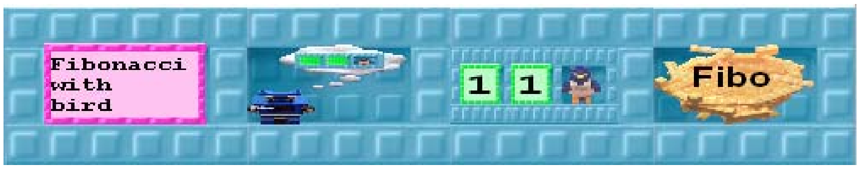
If we draw a binary tree whose left branches are twice bigger than his right ones we could find an interesting connection with the Fibonacci numbers, viz. if we assume that a branch of unit length grows for a unit time, after *n* time units the tree will have as many branches as the *n*-th term of the Fibonacci sequence.

Here is how the Fibonacci numbers were treated in a ToonTalk context [20]



Teacher: *Here is a sequence:*
1, 1, 2, 3, 5, 8, ….
Make a robot generating it.
Teacher: *This is very nice and tricky construction but it is not very easy to see the consecutive terms of the sequence. Now modify this robot so that it could output this sequence to another one.*
Ivan: *Ah, you want me to use a bird and a nest. Here you are:*
Teacher: *Are there initial conditions under which all the terms will be equal?*
Ivan: *0 0*
Teacher: *Can you study the ratio of every two consecutive numbers of this sequence?*
Ivan: *Let me try…*

He first makes a new robot but it does not work properly since he gets the proper term 3/2 but the next one in his construction is 5/2 (instead of 5/3). After realizing this Ivan goes to the board to describe in a meta-language the action of his new robot and to debug it. Then he produces two consecutive robots:



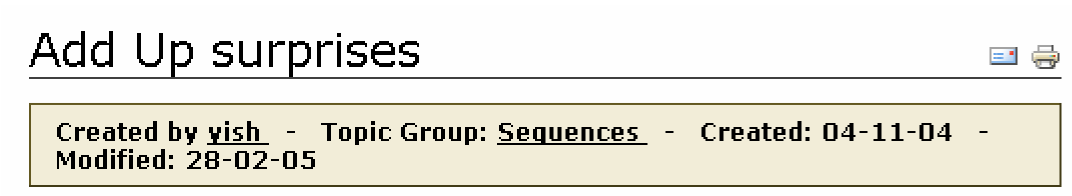
When Mitty was given the task of constructing the Fibonacci sequence he made the following robot:

Teacher: *Why do you use so many holes?*
Mitty: *It's more readable that way.*
Teacher: *Do you think it could be made by using only 2 holes?*
Mitty: *Impossible!*

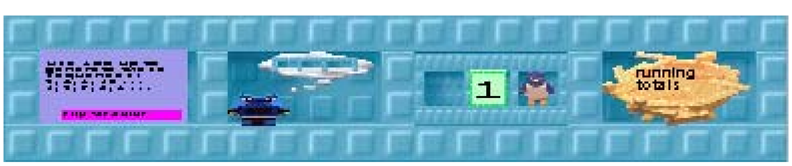
After observing the first solution of Ivan: *Ah, he is a sly old dog! But my solution is clearer. And if you start watching Ivan's robot from some bigger numbers you would never guess what he is doing...*

Note that Ivan's robot uses only two input holes. It copies and adds the content of the second hole over the first and vice versa. It is essential that the robot should run from the second hole. Suppose we have y_2=1 in the first hole and y_3=2 in the second. Adding the second to first will actually produce $y_3 + y_2 = y_4$ (3 in this case) and we will have a_4 and a_3 in the boxes after operation. Further adding first to second box will produce a_5 and etc. If Ivan had trained the robot to start the process with the first hole the resulting sequence would have been different.

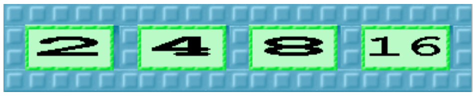
The reaction to this report was done by Yishay Mor and commented further by Ken Kan.

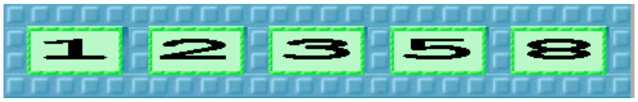


CHALLENGE: CAN YOU GENERATE THESE SEQUENCES WITH THE ADD-UP ROBOT?



I was playing with my Add Up robot, and discovered I can generate the sequences below with it. Can you?



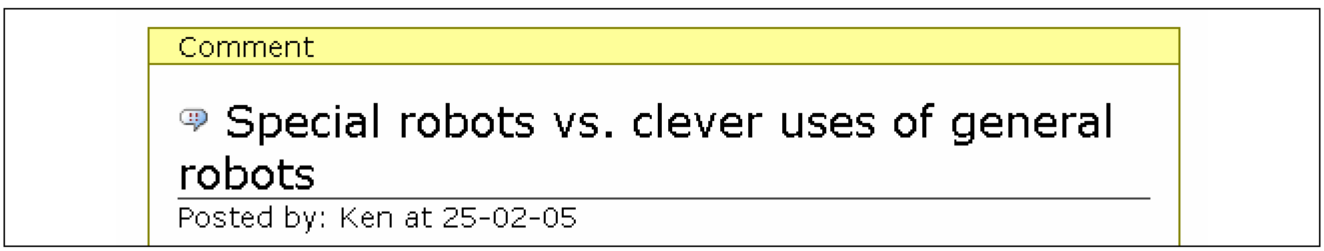


The challenge was serious even for us, the teachers. Here is how we proceeded:

Step 1: We put the two first terms of Fibonacci sequence in the nest thus using the nest as a FIFO buffer.
Step 2: Then we placed the nest in the first hole and 0 in other (using the hint that *the snake is eating its tail*).

In the Logo programming bible [1] the author writes:
There are several different ways to approach recursion, and my experience is that each person learns best by a different method. If you have troubles understanding this one, try the next. Even if you do understand this chapter easily you should still read the other versions. Each approach includes some important ideas not found in the others...

Using recursion in the above challenge gives us an additional insight about the connection between the two sequences. After discussing the problem from mathematical and informatics perspective we were excited to hear the opinion of the ToonTalk developer Ken Kahn:



I see this as an instance of a general issue in programming and constructionism. Some programming languages provide generic functions that one can specialize to solve problems. APL is an extreme example where you tend to create arrays and vectors and apply operations to these composite entities. In contrast sometimes it is clearer to just build up some program from simpler "primitives" even if it isn't as concise. But as Yishay points out sometimes when using the higher-level constructs you can see that two concepts that were thought of as very different can differ in a tiny way that a high level construct is used.

Conclusion:

In a panel discussion on the educational value of computer programming diSessa [4] proposes the idea that *the intellectual power the programming representations can have for learning science is at least comparable to, if not greater than, algebra*. One can easily adopt it in the context of learning mathematics – gaining the flexibility of moving from a programming to algebraic representation of a sequence contributes to a deeper understanding of the mathematical ideas. And such understanding helps back in verifying one's programs.

After comparing our observations with those of our project partners [6,12] we realized that we could explore together interesting research questions concerning the *complexity of the sequences*, such as*: How to measure their mathematical, computational, cognitive and pedagogical* *complexity? How are they correlated? How much of perceived complexity is inherent to the object under study, and how much – to the representational infrastructure being chosen?*

As WebLabs researchers we feel really enriched by the research process and insights so far. It has been an intellectual challenge and pleasure, as well. The concept for different representations, supported by relevant tools for computer explorations and for virtual collaboration created a rich and stimulating learning environment in which children develop their thinking and intuition and deal with notions, ideas and constructs that are usually introduced at much higher age. In order to deeper reflect on the WebLabs experiences and to judge their educational value, a next step would be to generalize the WebLabs findings and to enable the wider community to gain from it.

References

1. Harvey, B. (1988) *Computer Science Logo Style, Volume 1: Intermediate programming*. The MIT Press.
2. Hofstadter*, D*. (1985) *Metamagical Themas*: Questing for the mind and Pattern. Basic *Document*
3. Klopfer, E. et al. (1999) *Programming Revised – The Educational Value of Computer Programming*. http://www.cs.berkeley.edu/~abegel/papers/icls04.pdf
4. Loethe, H. (2003) Concept Oriented Learning Environments: A Project Using Sequences in Scheme, in *"Be Creative...re-inventing technology on education", Proceedings of Eurologo 2003*, Portugal
5. Mor, Y, Sendova, E. (2003) ToonTalking about mathematics, in *Proceedings of the International Congress MASSEE* 2003, p.36-43 Books, p. 253
6. ICMI, Study 14 (2003) Application and Modeling in Mathematics Education. *Discussion* http://www.weblabs.org.uk/wlplone/Members/Sofia/my_reports/Report.2004-04-13.0957
7. Mor, Y. et al. (2004) Thinking in Process. *MicroMath,* vol. 20/2*, p. 22*
8. Parnafes, O. and DiSessa A.A. (2004) Relations between types of reasoning and computational representation, *Journal of Computers for Mathematical Learning*, Vol.9, No.3 pp.251-280
9. Sendov, Bl. et al. (1989) *Mathematics and Informatics – a textbook for 9th grade*, Narodna Prosveta, Sofia
10. Toon Talk
    http:// www.toontalk.com
11. WebLabs: New representational infrastructures for e-Learning
    http://www.weblabs.eu.com
12. Weblabs: New representational infrastructures for e-Learning. *Operational Plan for Evaluation and Validation*,. November 2004
13. Guess my robot Game
    http://www.weblabs.org.uk/wlplone/Members/Rita/my_reports/Report.2004-01-27.1557
14. Webreport**:** *Bugging and debugging*
    http://www.weblabs.org.uk/wlplone/Members/Sofia/my_reports/Report.2004-11-12.2714
15. Webreport: *Convergence and divergence activities in Sofia*
    http://www.weblabs.org.uk/wlplone/Members/Sofia/my_reports/Report.2004-11-29.3038
16. Webreport: *Different representations of number sequences*
    http://www.weblabs.org.uk/wlplone/Members/Sofia/my_reports/Report.2004-04-05.2630
17. Webreport: *Harmonic series*
    http://www.weblabs.org.uk/wlplone/Members/ioe/my_reports/Report.2004-03-03.3328
18. Webreport: *Weblabetics*
    http://www.weblabs.org.uk/wlplone/Members/Sofia/my_reports/Report.2004-04-13.0957
19. Webreport: The fascinating infinite series
    http://www.weblabs.org.uk/wlplone/Members/Sofia/my_reports/Report.2004-06-24.3055

20. Fibonacci and the golden section
http://www.weblabs.org.uk/wlplone/Members/Sofia/my_reports/Report.2005-02-21.0006

Acknowledgements:

We acknowledge the support of the *WebLabs* project: "*WebLabs*: new representational infrastructures for e-learning"; contract number IST 2001-3220

We express our deep gratitude to Dr. Stefka Dimova (Sofia University) who showed us how to prove the equivalence of the robots by means of the theory of difference equations.

We are very thankful to all the children participating in the project for their enthusiasm and creative ideas.