Exploring infinite processes through Logo programming activities of recursive and fractal figures

Ana Isabel Sacristán

Department of Mathematics Education, Cinvestav

Av. IPN 2508, Zacatenco, 07360 México DF, MEXICO

asacrist@mail.cinvestav.mx

Abstract

Logo is a powerful language that allows for explorations of advanced mathematical topics such as that of infinite sequences and series. In this paper, I revisit the design of a computer microworld for the exploration of such infinite processes. In this microworld, students of different ages (some as young as 14) constructed and investigated graphical models of infinite sequences of the type $\{1/k^n\}$, as well as fractal figures, conceived here as “limit-objects” of infinite graphical sequences, such as the Koch curve. Students gave meaning to the processes under study by coordinating the visual and numeric outputs with the symbolic code contained in the procedures that they themselves had written.

Keywords

Logo, microworlds, infinity, infinite sequences, fractals

1. Introduction

At Eurologo 1999, I presented a paper (Sacristan, 1999) where I introduced the design of a Logo microworld for the study and exploration of infinite processes, specifically infinite sequences and series. The way in which that microworld mediated students’ learning was researched (see Sacristan, 1997) with subjects of different ages: 14 year-olds, high-school students, college students and even teachers, and refers to an advanced mathematical topic. Considering that the topic of the 2005 Eurologo conference is “Digital Tools for Lifelong Learning”, and that such microworld is an excelent example of the use of Logo as a useful tool for any age or topic, I decided to revisit it and present results from that study not included in the previous paper.

2. An infinite processes microworld.

The challenge was to create situations and ways in which infinity could become more accessible. Infinite processes such as infinite sequences and series, and the ideas of limit and convergence are concepts that are often a source of difficulty for students. One of the problems is that infinity is not "extractable" from sensory experience, it is a mental construct which often defies common sense. Furthermore, the areas of mathematics where infinity appears are also those that have traditionally been presented to students mainly from an algebraic/symbolic perspective making it difficult to create a link between formal and intuitive knowledge.

In an attempt to make the infinite more "concrete", I built a computational “set of open tools” (see diSessa, 1997) — a microworld — which could provide its users with insights into a range of infinity-related ideas. This microworld, using Logo, provided a means for students to construct and explore different types of representations — symbolic, visual, unfolding (using movement) as well as numerical — of infinite processes via programming activities. The computer environment was used to construct symbolic representations (in the programming code), simultaneously integrating them with dynamic or “unfolding” visual and numeric representations. This is an important point: the emphasis was on creating meanings through the construction of different types of external representations of the infinite processes under study.

The way in which the activities of the microworld were conceived is that they create the necessity to coordinate the different representational systems involved: the graphical and numerical representations, as well as the process they represent, are described and controlled through the symbolic (programming) code. In this way, the programming activities may facilitate the integration at a cognitive level of the different aspects of the mathematical object under study, by first linking them on an external level (see Papert, 1993). Furthermore, by having multiple types of representations linked through the programming code, the construction of bridges between the different contexts in which the infinite is encountered may be facilitated. This is a significant issue since it has been found by several researchers (e.g. Nuñez, 1993) that the context and situation in which infinite processes are encountered profoundly influences students conceptions of such processes.

A further issue is that the use of the computer allows the observation of the temporal evolution of a process, rather than just the final state (or result) of the process, as was the case before the advent of computers in school mathematics: thus, the behaviour of infinite processes can be investigated.

2.1. The activities of the microworld: explorations of models of infinite sequences and fractals.

The central topic of the microworld was the convergence (and divergence) of infinite sequences and series, and limits, through the use of recursive geometric figures. Self-similar figures and fractals — which are the limits of infinite graphical sequences— were used for introducing a different kind of setting for the idea of the limit of a sequence, and for presenting some of the results that students sometimes find paradoxical and that can come about when working with the infinite. Fractals also have an intrinsic *recursive structure.* The activities of the microworld thus involved the visual study of these recursive structures, and the investigation of the building *process* of the geometrical objects involved. By observing the movements of the turtle a sequence of geometrical objects can be seen as *processes* in time; that is, through the geometric representation, the different steps of the process can be seen. Additionally, the possibility of producing successive levels of a figure as approximations to the "real fractal" can be thought of as a (potentially) infinite process, yet it is an exploration of a geometric object that has already been given through the symbolic code. Progressing through the different levels of the picture can thus be interpreted as approaching an object that "is already there".

The activities and procedures were divided into two sections:

a) explorations of classical infinite sequences and their corresponding series, through geometric models such as spirals, bar graphs, and staircases.

b) fractal explorations

The procedures given here[1] are examples of how the representations of the infinite sequences and processes can be constructed, but each student constructed his own procedures according to his/her understanding. Although we do not provide here any details (for that, see Sacristán, 1997), these activities were complemented with paper-and-pencil activities such as the construction of tables where numerical values were recorded and the relationships between the different elements involved in each process could be made explicit.

3. Spirals, bar-graph and staircases: models of infinite sequences and their corresponding series

In the exploratory activities, sequences such as $\{1/2^n\}$, $\{1/3^n\}$, $\{(2/3)^n\}$, $\{2^n\}$ etc., and then $\{1/n\}$, $\{1/n^{1.1}\}$,..., $\{1/n^2\}$, and the sequences of their corresponding partial sums, were investigated through visual models such as spirals, bar graphs, staircases, and straight lines, and the corresponding Logo procedures, with a complementary analysis of the numerical values (their progressions and the apparent limits, if any existed or appeared to exist).

These geometric models are a straightforward way of translating arithmetic series into geometric form. For instance, in the 'spiral' type of representation each term of the sequence is translated into a length, visually separated by a turn, so that the total length of the spiral corresponds to that of the sum of the terms (the corresponding series). Thus, for instance for the sequence $\{1/2^n\}$, the visual process and added lengths of the spiral would represent the series: $\frac{1}{2}+\frac{1}{4}+\frac{1}{8}+\frac{1}{16}+\ldots$, a notation which is descriptive of the process involved — the ellipsis points indicating an indefinite continuity of the process (a *potentially* infinite process). On the other hand, in the symbolic computer code, the same series can also be represented by a notation corresponding to $\sum_{n=1}^{\infty}\frac{1}{2^n}$, which is an object in itself (an *actual* infinite object), and does not explicitly indicate the infinite process it describes. (This is independent of the convergence or divergence of the series, although when there is convergence it is easier to think of the series as a "complete" object, e.g. when $\sum_{n=1}^{\infty}\frac{1}{2^n}=1$). This illustrates how the same (mathematical) object can be represented both as a (complete) *object* as well as in terms of a *process*.

It was hoped that through the observation of the visual (and numeric) behaviour of the models, students would be able to explore the convergence, and the type of convergence, or divergence, of a sequence and its corresponding series, and predict the behaviour at infinity. The different geometric models for the same sequence were meant to provide different perspectives of the same process. An aspect that was considered important for this, was that the students carried out the transformations of the models themselves by changing the computer code. It was intended that this involvement would help build links between the symbolic representation (in the code) and the different models.

[1] The procedures in this paper are actual procedures created by some of the students; only the names of the procedures were translated into English from the original Spanish.

3.1. An example of the ways in which students coordinated the different representations within the microworld: Endless movement and the link with the recursive (iterative) structure of the code – the interaction between the symbolic code and the visual output

Here, I present an example to illustrate some of the ways in which students used and coordinated the elements of the exploratory medium *to construct meanings for the infinite*. That is, the microworld gave the students means to *make sense of what they saw on the screen via the programming code*: the interactions between the code and its outputs.

We introduced the visual models to the students by giving them the procedure below and asking them to predict its behaviour.

```
TO DRAWING :L
PU
FD :L
RT 90
WAIT 10
DRAWING :L / 2
END
```

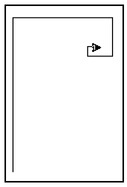

Figure 1. Spiral model for the sequence $\{1/2^n\}$.

This procedure makes the turtle walk through a spiral with arms each having half the length of the previous one (see Figure 1). It is a first approach to the infinite sequence $\{1/2^n\}$, which was chosen because of its simplicity. It should also be noted that this is a tail-recursive procedure without a stop condition, so the procedure could potentially continue indefinitely, in an effectively iterative way. This procedure — which in the beginning does *not* draw but does show the turtle moving — was designed to induce students to reflect on the behaviour of the turtle and of the process itself.

The idea of having the pen up produced interesting results: in particular, the students had to visualize the actual pattern without relying on the computer drawing; it induced students to try to make sense of the relationships between the code and the graphical output: For instance, most students did not expect to see the turtle endlessly spinning without leaving a trace. In order to explain to themselves this unexpected behavior and make sense of why the turtle was endlessly spinning, the students had to re-examine the procedural code and carefully observe the movements of the turtle (which are made easier to see by the WAIT command). Victor, a 23 year old college student, was one subject who immediately remarked that the procedure would never stop because the recursive structure of the code represented an infinite process. He explained it was because the procedure called itself without anything telling it to stop, so it never would; the process of turning and walking half the previous distance would continue repeating itself and would never stop. By analyzing the code Victor was able to connect to it the behavior of the visual output (in this case the *movements* of the turtle) since he correctly predicted the outcome and was able to justify that visual behavior through the code. He *linked the recursive structure of the code with the infinitude of the process.*

A modified procedure (with the Pen down) produced an inward spiral with the turtle then turning endlessly in its center. Victor and his partner, Alejandra (another college-level student) pointed out that although the turtle seemed to be just turning in the same spot, in reality there was "a variation”. There were two factors here: a) The turtle kept turning and b) the turtle turned at same spot. The first factor could have served as an indicator that the process continued, but it was the fact that the students seemed to be able to disregard the *visual appearance* of the turtle — spinning in apparently the same spot — that suggests that they understood that the underlying (mathematical) process continued, and that they were able to link the output with the code and the process.

Later in the activities, when the students modified the procedure to give out numeric values, they would confirm the continuation of the process by still getting an output of values, even when the turtle seemed stuck:

> Alejandra: Apparently it is stopping on the screen, but it is still walking because we are still getting the values.

It was thus that students were able, via a process of experimenting backwards and forwards from code to figure, to make sense of the behavior of the turtle, which seemed to be spinning on the same spot realizing that the amount that the turtle moved each time was halved. The key point here is that the analysis of the code allowed them to: 1) recognize in the recursive structure a potentially infinite process; and 2) to quantify the movement, to explain that although the turtle seemed to be turning without moving forward, in reality there *was* a variation.

Thus, by coordinating the visual and symbolic — in the order visual to symbolic to visual — and later complementing it through numerical explorations, their understanding of the process became integrated and potentially misleading visual appearances could be ignored. This interplay between the code and its outputs, which led students to make sense of what they observed by linking all the elements, is illustrated in Figure 2:

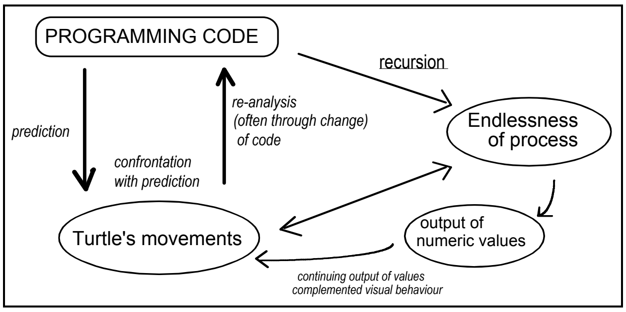

Figure 2. The interplay between the code and its output to make sense of the endless movement: the graphical image gains meaning from the symbolic representation.

3.2. Modifying the original procedure to generate other visual models of infinite sequences and series

The above procedure was the generic activity for subsequent activities. This initial procedure was to be modified (see further below) so that the turtle actually left a trace (by putting the Pen down), as well as adding a stop condition. In fact, another activity that produced interesting results, was derived from students' investigations of the stop values in the stop condition and the study of the relationship of the stop-value with the number of segments drawn in the geometric model.

An important modification was that of transforming the spiral into other types of graphical models, because different models provide different perspectives of a same process:

- Through a **straight Line** (see Figure 5) the spiral can be stretched out in order to observe its total length (which represents the value of the corresponding series) ; in the case of sequences with corresponding convergent series, this model can visually demonstrate that convergence.

- The **Staircase** model (see Figure 3) is a way of combining the Spiral mode, where the different terms of a sequence can be discerned, but, like the Line model it also allows for the visualisation of the behaviour of the corresponding series.

- Finally, separating each term of the sequence into a **Bar Graph** (see Figure 4) allows for the sequence of terms to be seen side to side.

When students construct themselves the procedures for each of the models, they can connect all of these models to one unique process underlying all the models and described in the programming code. By ending up with a general procedure and a set of visual models, the behaviour of each sequence, and of its corresponding series, can be explored through each of its models.

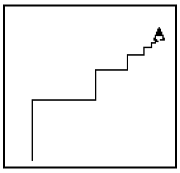

Figure 3. Stair model corresponding to the sequence $\{1/2^n\}$.

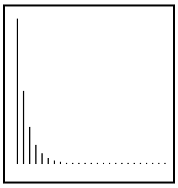

Figure 4. Bar graph corresponding to the sequence $\{1/2^n\}$.

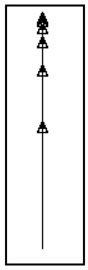

Figure 5. Line model corresponding to the sequence $\{1/2^n\}$.

Below we give an example of a new general procedure DRAWING after modification of the original procedure, and its subprocedures FUNCTION (determining the transformation carried out at each step of the process) and MODEL (determining the visual model to be used):

```
TO DRAWING :L
IF :L < 1 [STOP]
MODEL
DRAWING ( FUNCTION :L )
END
```

```
TO FUNCTION :L
OP :L / 2— for the sequence
{1/2n} —
END
```

```
TO MODEL
SPIRAL
END
```

where `SPIRAL` can be replaced by any of the following:

for a spiral:

```
TO SPIRAL
FD :L
RT 90
WAIT 10
END
```

for a bar graph:

```
TO BARGRAPH
JUMP
FD :L
END

TO JUMP
PU
SETY -100
RT 90 FD 5 LT 90
PD
END
```

for a staircase:

```
TO STAIR
FD :L/2
LT 90
FD :L/2
RT 90
END
```

for a straight line:

```
TO LINE
FD :L
WAIT 10
END
```

The students played with these procedures and made changes such as that which modifies the original sequence ($\{1/2^n\}$) into a new one, such as that which corresponds to $\{1/3^n\}$. The comparison activities between different sequences of the same type proved in our study to be very useful in the analysis of the behaviour and rate of convergence (or divergence) of those sequences.

At a later stage, it became important to change the approach in the construction of the models, so that the visual models are no longer produced simply through the process of transforming a previous segment, but rather as actual *models* of a sequence described as a *list* of values. This is a significant change: (i) First, producing a sequence as a list and *then* modelling it, highlights the idea that the geometric figures are models of a mathematical process which can be symbolically described and independently expressed; whereas before the numerical values of the segments were seen merely in terms of the measure of the segments. It is constitutes a step in the formalisation of the process. (ii) Additionally by adding the idea of producing lists of the values of the sequences, these potentially infinite sequences originally seen as *processes* can now be seen as *sets,* a conceptually different representation of the same idea. (iii) This change involves making explicit the description of the sequences, and therefore helps to differentiate a sequence from its corresponding series (which was originally represented as the "total length" of a model). (iv) Furthermore, the new approach facilitates the numerical analysis of the sequence, and allows for the use of a scale variable (which acts as a kind of "zoom") independent of the numerical values of the sequence.

The new approach requires a different way of producing the visual models, as well as the need of procedures for generating the sequences and storing the first *n* terms into a list. Thus, the previous procedures need to be modified to draw visual models of *terms* of sequences. This is an useful activity because it helps to create an interaction between the numerical values at each stage and its corresponding visual representations, and assists in the analysis of the behaviour of the overall. Examples of the new procedures are given below:

To generate the first N terms of a sequence:

```
TO SEQUENCE :N
IF :N = 1 [OP FN 1]
OP SE (SEQUENCE :N - 1) (FN :N)
END
```

where the sequence is given as a function of N, and not of L. For instance:

```
TO FN :N
OP  1 / POWER 2 :N
END
```

The drawing procedure would then be as the one below, changing in each model "`:L` " by "`:SCALE * FIRST :LIST`":

```
TO DRAWSEQUENCE :LIST :SCALE
IF :LIST = []  [STOP]
FD :SCALE * FIRST :LIST
MODEL
DRAWSEQUENCE BF :LIST :SCALE
END
```

In order to investigate the values of the series and partial sums of each sequence the following procedures were used in combination with SEQUENCE to generate the input list (e.g. typing `SUML SEQUENCE` <number of terms>). The PARTIALSUMS procedure gives as output the *list* of partial sums of a sequence of *n* terms; while SUML is a procedure which adds all the terms in a list (sequence).

```
TO PARTIALSUMS :LIST
IF :LIST = [] [OP [] ]
OP SE (PARTIALSUMS BL :LIST) (SUML :LIST)
END
```

```
TO SUML :LIST
IF :LIST = [] [OP 0]
OP (FIRST :LIST)+ SUML BF :LIST
END
```

4. Fractal figures as "limit objects".

Fractal figures, such as the Koch curve and snowflake, and the Sierpinski are *limit* objects generated through infinite geometric sequences. These fractals "exist" as limits of infinite processes; yet, once "produced" they can also be conceived in terms of sets consisting of infinitely many parts, where each part is *self-similar* to the whole (thus highlighting the recursive/iterative fundamental nature of the infinite). For obvious reasons, these figures provide a rich ground for the exploration of infinite processes and of "infinite" objects: they give the opportunity to study a different kind of limit, including the visual sequence that leads to it; furthermore, the programming code reflects in its recursive structure, each of the steps of the sequence. (Fractals also have the advantage of being beautiful, attractive, and fun, and of being a more contemporary area of mathematics).

These activities are useful for confronting students with the idea of "what happens at infinity", by having them "visualise" an infinite process by observing its behaviour through the, albeit finite, computer-based approximations. Through these explorations, paradoxical situations emerge (see further below). In order to solve such paradoxes it is useful to include numerical explorations to complement the visual models; for such explorations "measurement" procedures can be used, and the numerical data can be structured into tables of values.

4.1. The Koch curve and snowflake.

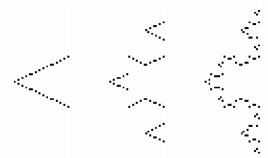

Figure 6. Construction process of the Koch curve.

The Koch curve is constructed by replacing in each step, each (sub)line-segment with a self-similar figure to the original generating figure (see Figure 6). The explorations activities involve measuring the perimeter of this curve. The use of tables for recording the values for the number of segments and size of each segments are useful for this purpose since they allow to “visualise” in the numeric, the behaviour of the different elements present. A similar approach was used to study the Koch snowflake produced by joining in a triangular fashion three Koch curves (see Figure 7), its perimeter and its area. The procedures for these figures are given below:

For generating the Koch curve:

```
TO CURVE :L :LEVEL
IF :LEVEL = 1 [FD :L STOP]
CURVE :L / 3 :LEVEL - 1
LT 60
CURVE :L / 3 :LEVEL - 1
RT 120
CURVE :L / 3 :LEVEL - 1
LT 60
CURVE :L / 3 :LEVEL - 1
END
```

And for drawing the snowflake:

```
TO SNOWFLAKE :L :N
REPEAT 3 [CURVE :L :N RT 120]
END
```

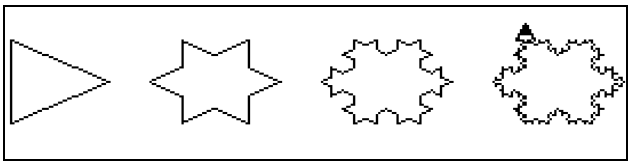

Figure 7. Construction process of the Koch snowflake.

Additionally, procedures such as the following, which computes the area of an equilateral triangle, were used as tools for computing the area of the snowflake, and in the Sierpinski triangle activity below:

```
TO AREATRI :SIDE
OP (POWER :SIDE 2) * (SQRT 3) / 4
END
```

4.2. Koch curve 'paradoxes': solving the paradox by coordinating two simultaneous infinite processes.

For some students, the idea of an infinite perimeter formed by an infinite number of "zero-length" segments caused anxiety. This was particularly the case of Manuel and Jesus, two high-school students. Nuñez (1993) has explained that many paradoxes arise in dealing with the infinite when there are several competing components (processes) present. In our case, Jesus was aware that there were two types of processes involved in the change of the perimeter: the increase in the number of segments, and the decrease in the size of those segments. He realised that *the behaviour of the numerical values* pointed towards the perimeter becoming very large, infinite. But when they considered that the segments *at infinity* measured zero, this seemed to indicate to them that *at infinity* the perimeter would measure zero! In fact, by focusing on the latter process, Jesus would challenge the idea of the divergence of the perimeter: "The segments are getting smaller... The perimeter cannot be infinite...". His partner, Manuel, had a different perspective: he focused more on how the zero-sized segments would affect the *shape* of the figure first concluding that it would become a "smooth" curve with no segments, "an infinite sequence of points". By thinking of an infinite number of zero-sized segments, the students were dealing with what in Calculus is called the indeterminate form of a limit; in this case: $\infty \times 0$. The students realised it was necessary to carry out algebraic and numeric explorations to solve the paradoxes. A breakthrough came when Jesus became interested in how each of the factors (the rate of decrease in the size of each segment vs. the rate of increase of the perimeter in the number of segments) behaved in relationship to each other. He used numerical explorations (structured in a table) to explore the behaviour of the perimeter, verify his hypothesis, and become convinced of the divergence of the perimeter by observing that the perimeter's increase was faster than the segments' convergence to zero. Whereas the indeterminate form of a limit is traditionally solved through algebraic manipulation, in this case Jesús solved the paradox through analysis of the *behaviour* of each of the elements involved, observing specifically the difference in the *rate* of divergence or convergence of each of the elements and coordinating the two processes involved. This was possible thanks to the exploratory design of the microworld which combined visual representations with numeric investigations.

4.3. Explorations of the Sierpinski triangle.

Another fractal activity included in the microworld ,and useful for analysing the behaviour at infinity, was the study of the Sierpinski triangle. This fractal is constructed by a process that "takes away", at each step, one fourth of the area of each part (see Figure 8), and is such that the area at infinity becomes nil. The procedure for constructing such a figure is:

```
TO TRI :SIDE :LEVEL
IF :LEVEL = 0 [STOP]
REPEAT 3 [TRI :SIDE / 2
 :LEVEL - 1   FD :SIDE  RT 120]
END
```

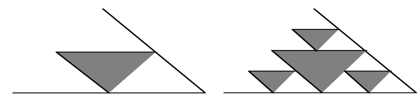

Figure 8. Construction process of the Sierpinski triangle.

If analysed, the code of this procedure reflects the structure of the procedure in that in each triangle, there are three similar triangles of half the size.

5. Concluding remarks

It is relevant to point out that the procedures and activities described above are simple the point of departure for much more diverse activities. The learner must be given enough freedom to explore in his/her own way the given processes and objects, as well as to use the given ideas as a basis for the exploration of other ideas, through modifying the existing procedures or constructing new ones (this is why we call the microworld: a set of open tools). As an example of this, in our study (Sacristán, 1997) we experienced how the fractal activities inspired some of the students to write a procedure that would generate a graphical construction of Cantor's Set in order to investigate its behaviour.

To summarise, the students were able to reconstruct or redesign the tools, and the links between them, and express themselves through the *programming* activities of the microworld. In this way they had an opportunity for constructing meanings through *doing*, action and expression. The constructive actions that connected the different elements, thus played a mediating role in the exploratory activities for the creation of meanings for the infinite processes under study.

References

diSessa A (1997), *Open Toolsets: New Ends and New Means in Learning Mathematics and Science with Computers*, Proceedings of PME-21, Pehkonen E (Ed.), Finland 47-62

Noss R & Hoyles C (1996), *Windows on Mathematical Meaning. Learning Cultures and Computers*, Kluwer Academic Press.

Nuñez R (1993), *En Deçà du Transfini. Aspects psychocognitifs sous-jacents au concept d'infini en mathématiques* Éditions Universitaires, Fribourg, Suisse

Papert S (1993), *The Children's Machine*, Basic Books, New York

Sacristán AI (1997), *Windows on the Infinite: Constructing Meanings in a Logo-based Microworld*, PhD Dissertation, University of London, Institute of Education, London, England

Sacristán AI (1999), *Creating meanings for the infinite through a Logo-based microworld*, Supplement to the Proceedings of the Seventh European Logo Conference EuroLogo '99, Virtech Ltd, Sofia, Bulgaria, 475-484.