

Imaginary Gardens A Model for Imitating Plant Growth

Anne M. Burns
Mathematics Department
Long Island University, C.W. Post Campus
Brookville, NY 11548, USA
Email: aburns@liu.edu

Abstract

A simplified version of the use of L-Systems as a method of modeling plant growth is described. We do not insist on the accuracy of the models; instead we focus on producing aesthetically pleasing imitations of plants that can be used in creating computer-generated landscapes.

1. Introduction

I have always been fascinated by the beauty of plants and flowers, and ever since I bought my first home computer I have searched for ways to imitate the structure and growth patterns of plants using ideas from mathematics and computer science. At first my “plants” were stick figures made up of straight lines for branches and discs for flowers; the screen resolution was 320 by 200, and there were only three colors available. Gradually, as the capabilities of the computers improved, so did my plant models. The sticks became curved rendered tree branches or plant stems and the disks became intricate inflorescences. Along the way I searched out articles suggesting new methods of generating plants. In the late 1980’s a number of computer scientists presented new ideas for modeling plant growth and several presentations were made on this topic at SIGGRAPH conferences. I was not so much interested in the accuracy of the models, but more in the aesthetic possibilities for artistic works. Also, I am a teacher, and this was an exciting new way to get students interested in math. It makes a great topic for a class in computer science. For a comprehensive list of papers that deal with the theory of string rewriting, formal languages and L-Systems see the references.

In this paper I will present my own simplified adaptation of this method of modeling plants that can be used in a class in programming. Processing the strings is a real live application of a stack and its implementation. Writing functions that draw curves and leaves and flowers gives students the opportunity to use some elementary trigonometry and algebra.



Figure 1: *An interpretation of Example 3*

2. L-Systems and String Rewriting

A biologist from the University of Utrecht, Aristid Lindenmayer, using ideas from formal language theory, conceived of a method for modeling the structure and growth of plants by generating new character strings from existing ones by replacing one or more characters in a string by strings of characters in a parallel manner. Here is a simple example (a formal definition can be found in the references): We start with a single character which we call the *axiom*, and a set of rules which we call *productions*. Each production has a left side and a right side. In our first examples we will make the left side of each production a single character and the right side a string of characters. A plant will be associated with a string of characters that initially will consist of a single character, the axiom. To get to the next stage in the plant's development we will traverse the string of characters from left to right. Whenever we encounter a character that is the left side of a production, we replace that character by the right side of that production. By convention, when we encounter a character that is not the left side of any production we replace that character by itself.

Example 1. Our *alphabet* will consist of the set of characters $\{I, [,], (,)\}$ and the axiom will be the character I . Our L-System will have one production: $I \rightarrow III$ This is what our string will look like after one and two stages:

Axiom: I

Stage 1: III, Stage 2: $I[III][III]III$

At stage 3 the string will consist of 148 characters!

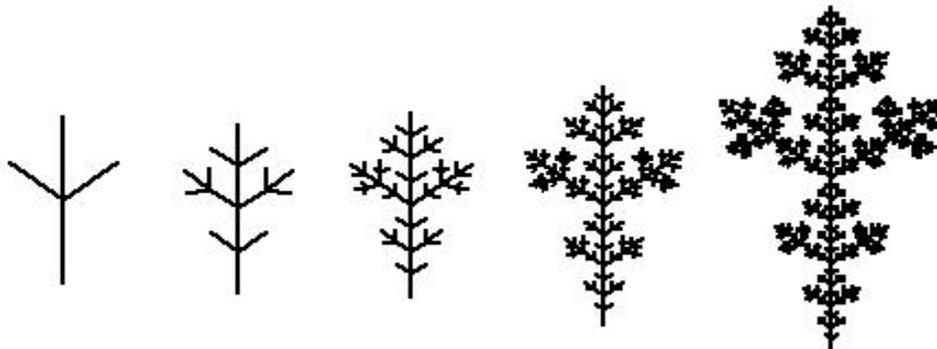


Figure 2: Stages 1 – 5 of a simple example, Example 1

Now let us give a geometric interpretation to this string. We write a computer program and in it we write a function *plant* with parameters *stage*, *x*, *y*, *angle*, *length*. Executing this function, *plant(stage, x₀, y₀, π/2, 10)*, will result in the following steps: the initial point will be set at x_0, y_0 and a variable *angle* will be set to $\pi/2$. A variable *length* will be set to $10/\text{stage}$. The string will be traversed from left to right one character at a time. Each time I is encountered the function will draw a line segment of length *length* at the current angle *angle*. When either $[$ or $($ is encountered, the variables *x*, *y* and *angle* will be “pushed” onto a stack (their current values will be saved) and if the character is $[$ the variable *angle* will be increased by some small angle (for example $\pi/3$), if the character is $($ the variable *angle* will be decreased by some small angle. That is, we interpret $[$ as a branch to the left and $($ as a branch to the right. When a $]$ or $)$ is encountered the old values of *x*, *y* and *angle* are “popped” off of the stack, that is, the parameters that existed before we branched off are recovered. Figure 2 shows this interpretation of several stages of Example 1.

Figure 1 is a second example of a very simple L-system: Axiom I and one production:

$I \rightarrow I[IL[I[IF]]](ILF)I$. Here two new characters have been added to our alphabet, “L” which we interpret as a “leaf” and “F” which we interpret as a “flower”. We also added some curvature to the line segments.



Figure 3. Three “plants” generated by the same Stochastic L-System

We can make our plant growth less predictable by using probabilities in what we might call a “Stochastic L-System”. Figure 3 shows three “plants” resulting from a single L-System. In a stochastic L-System we allow two or more productions to have the same left side (I) but different right sides. In Figure 3 we have used two productions with left side I. The first production has the same right side as the L-System in Figure 2, “I[I][I]”, and the second has right side “I(I)I”. In general if we have n productions with the same left side we assign a probability (from nonnegative numbers p_1, \dots, p_n whose sum is 1) to each production. In the subroutine that replaces the old string with the new, when we encounter an “I” we generate a random number between 0 and 1. This number will determine which of the productions we use. In Figure 3 $p_1 = .45$ and $p_2 = .55$.

3. More Realistic Looking Plants

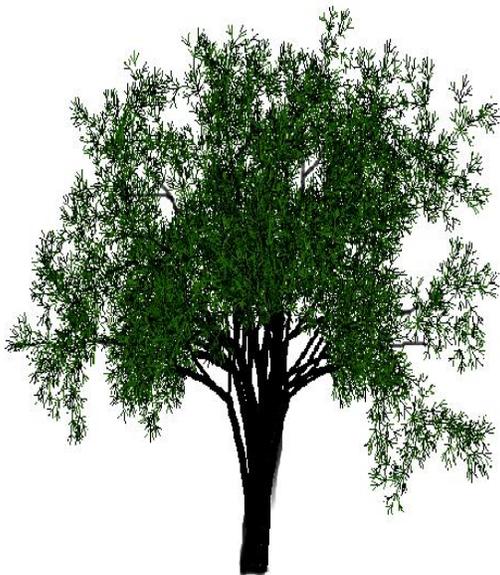


Figure 4a. A “conifer”

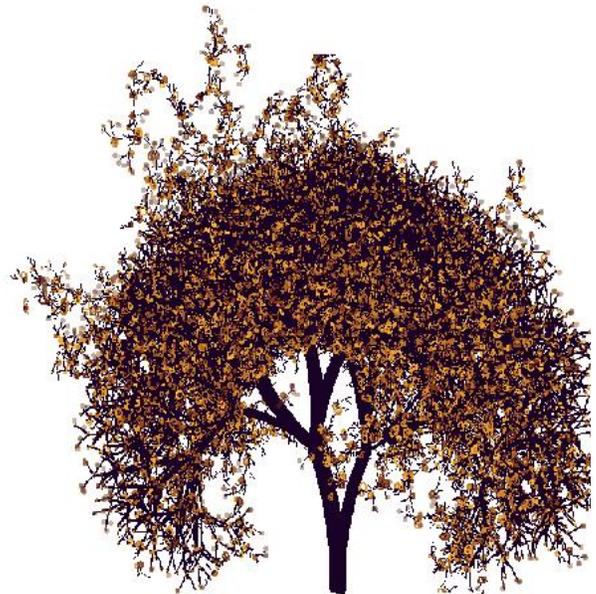


Figure 4b. “Fuzzy tree”

To make the umbel more realistic, we have to use more symbols than just I and F. We need to add some curvature to the stems and some randomness to the length and the amount of curvature of each stem. Adding curvature is relatively easy. We use the function that draws a line segment taking parameters x,y , $angle$, $length$, $thickness$ and $color$ and which returns the new endpoint (x,y) . Define a new function $curve$ with the same parameters but add one more parameter $curvature$. The new function is very simple; divide the parameter $length$ by some $n > 1$ and use the function $segment$ to draw n small segments, each time increasing or decreasing the $angle$ parameter by the $curvature$ angle. As the drawing angle will change during the execution of this function it should return the current $angle$ as well as the current value of x and y .

Figure 7 shows the result embellished by a stem with compound leaves that have been created by almost the exact set of rules that defined the flower, except, as you can see, at each stage there is a division into 3 rather than 5.

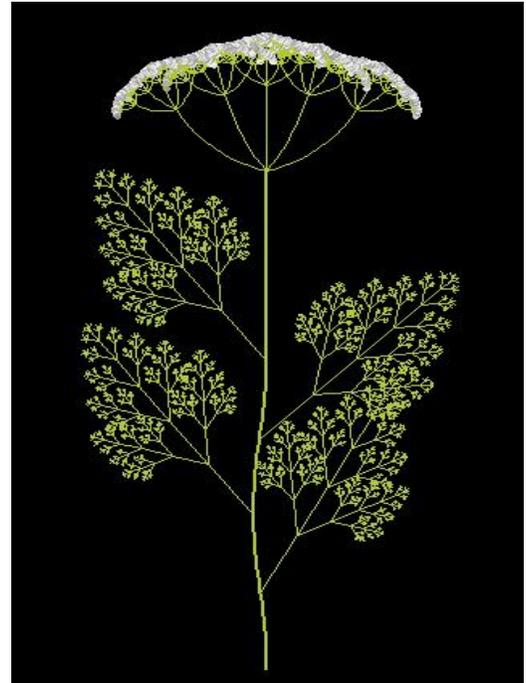


Figure 7 Family *umbelliferae*

“A *monochasium* is a peduncle bearing a terminal flower and, below it, one branch that produces a single lateral flower. The terminal flower is older. This is a *simple monochasium*. A repetition of this on the lateral branches produces a *compound monochasium*.” [2] See Figure 8.

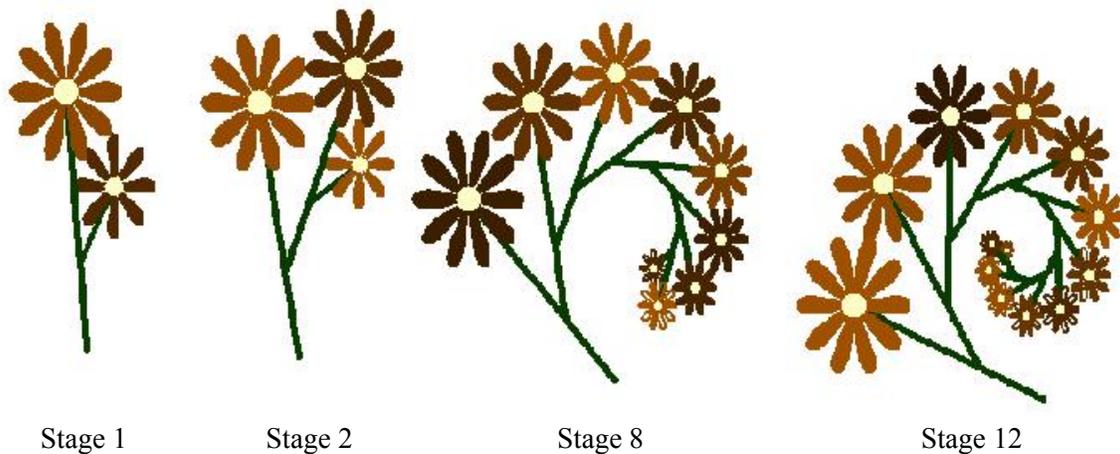


Figure 8. Rules for a monochasium: $I \rightarrow I$, $A \rightarrow I(A)B$, $B \rightarrow B$, Axiom: A

“A *dichasium* is a peduncle bearing a terminal flower and a pair of branches that produce lateral flowers. The oldest flower is the central one. This *simple dichasium* is a common unit making up parts of many more complex inflorescences. A repetition of this on a lateral pair of branches produces a *compound dichasium*.” [2] See Figure 9.

Rules for a dichasium: $A \rightarrow IAB$, axiom: A

I is interpreted as a line segment, A as a line segment topped by a small flower, and B a longer line segment topped by a larger flower.

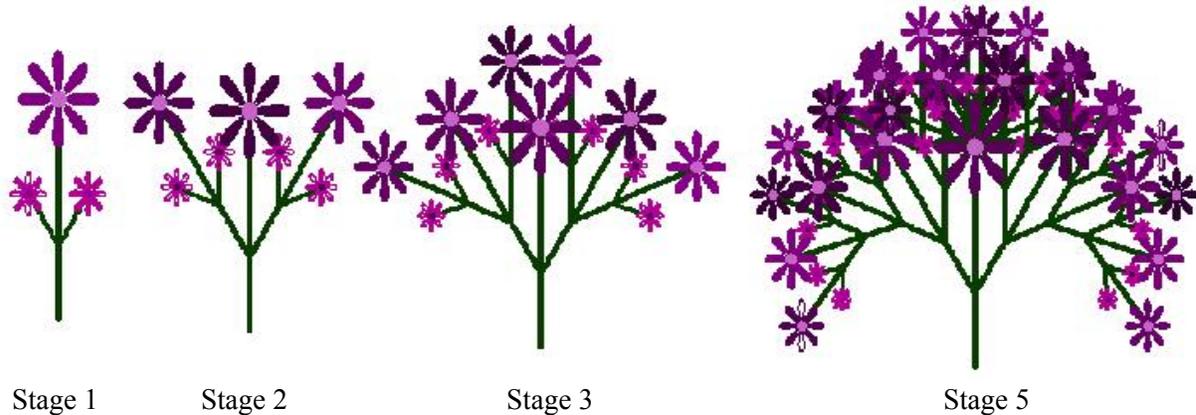


Figure 9. Stages in the development of dichasia

“A *panicle* is a more or less elongated inflorescence with a central axis along which there are branches which are themselves branched. These may be a sequence of blooming from the base upward, but some panicles are made up wholly of dichasia.” [2] See Figure 10.

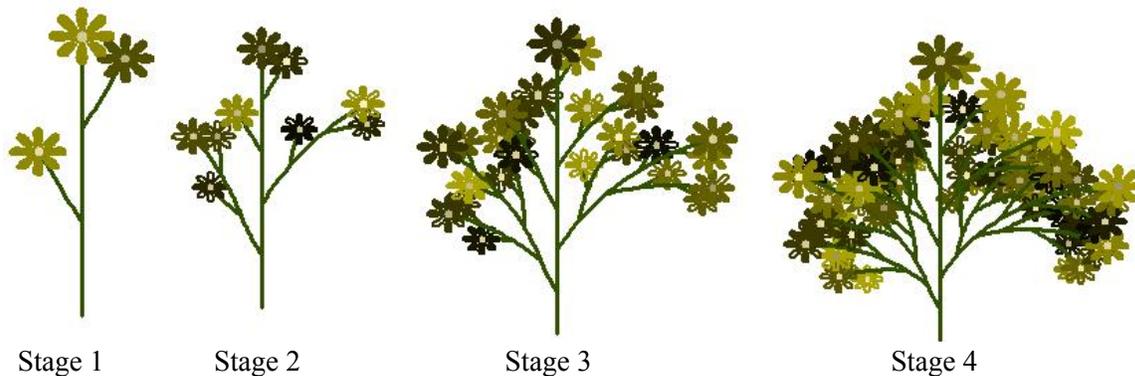


Figure 10. Rules for a panicle: $I \rightarrow A[IF]A(IF)IF$, axiom I

5. Let the Clock Tick – Parametric L-Systems

We can keep a clock ticking and allow variables to represent the time stages in the plant’s growth. For a given left side of a production we can have more than one right side; the particular one chosen will depend on the value of t . To illustrate this idea here is a simplified version of a flowering sequence where the flowers mature earlier on the lower branches. In this example the axiom is I and we let t represent the stage, that is at $t = 0$ the string consists of the axiom I.

The productions are:

$$I \rightarrow A[B]I \quad \text{for } t \leq 2$$

$$I \rightarrow AB \quad \text{for } t = 3$$

$$B \rightarrow Af(0)$$

$$f(n) \rightarrow f(n+1) \text{ for } n \leq 2$$

$$f(3) \rightarrow \varepsilon \text{ (the empty string)}$$

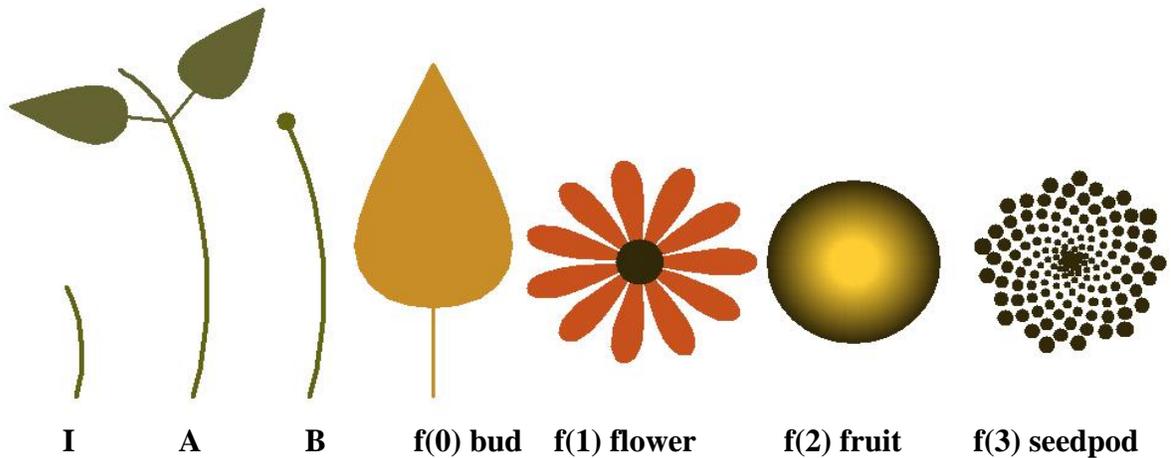


Figure 11. *Representation of string characters for a parametric L-System*

Figure 11 shows the representation of each of the characters in the string. (If you are a teacher, note that writing the routines that draw each of the figures involves a little math as well.)

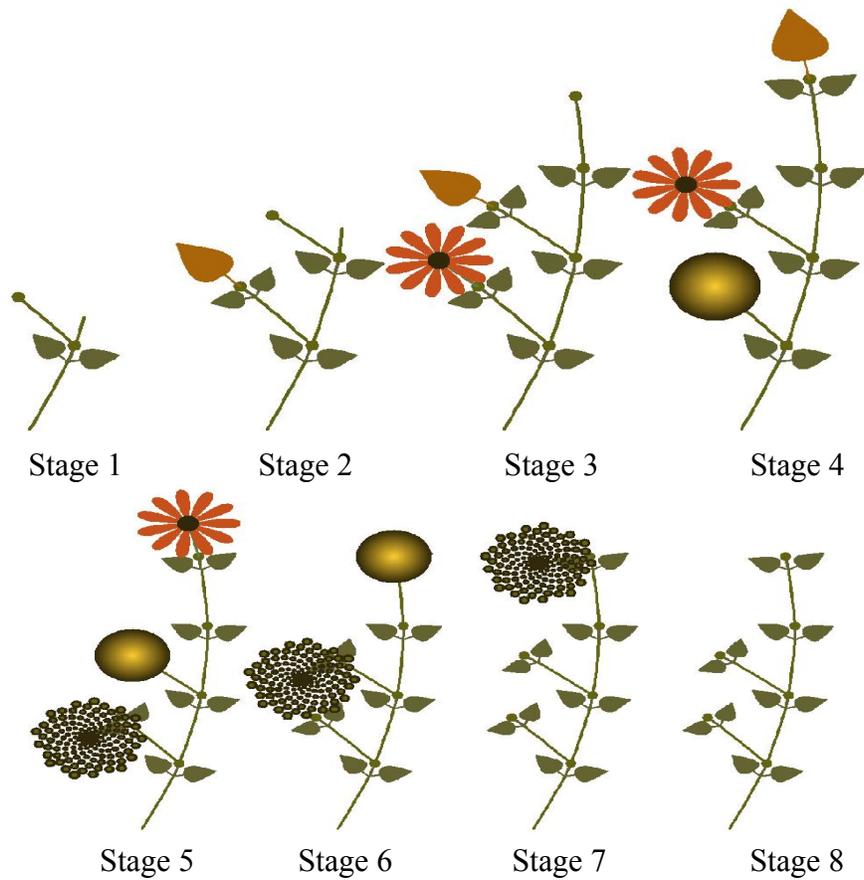


Figure 12. *8 stages in the parametric L-system*

Figure 12 illustrates the flowering sequence and life cycle of the “reproductive organs” of the plant.

Conclusion

String rewriting can be taught to students as a way to generate interest in mathematics and computer programming. Space constraints have allowed me to present only a very simple introduction to this exciting topic. For an excellent explanation of the rules that describe plant development, modeling of cellular layers and fractal properties of plants I highly recommend Prusinkiewicz's beautiful book [3]. Some of the other references are more mathematical and contain a wide variety of applications of formal language theory, L-Systems and string rewriting.



Figure 13. *Imaginary garden*

References

- [1] Anne M. Burns, The Flowering of String Rewriting Systems, *The College Mathematics Journal*, Vol. 23, No. 3, May 1992
- [2] C.L. Porter. *Taxonomy of Flowering Plants*, W.H. Freeman and Co., 1959
- [3] Przemyslaw Prusinkiewicz and Aristid Lindenmayer, *The Algorithmic Beauty of Plants*, Springer-Verlag, 1990
- [4] G. Rozenberg and A. Salomaa (Eds.), *Lindenmayer Systems*, Springer-Verlag, 1992
- [5] G. Rozenberg and A. Salomaa (Eds.), *The Book of L*, Springer-Verlag, 1986
- [6] Arto Salomaa, *Formal Languages*, Academic Press, 1973
- [7] Alvy Ray Smith, Plants, Fractals and Formal Languages, *Computer Graphics*, 19, 3 (July 1984)
- [8] <http://en.wikipedia.org/wiki/L-system>