Lindenmayer Systems and Their Applications

CSE 30151: Course Project 2

Lauren Ferrara

March 22, 2016

Relevance: In this paper, I discuss L-Systems, their relation to formal language theory, and their impact on the study of developmental biology. Additionally for this project, I wrote programs to demonstrate how L-Systems work similarly to context-free grammars and to show how they can be used to create computer graphics.

Effort: I spent 14 hours working on this project in order to develop a relevant topic, conduct background research, and write the programs and paper.

L-Systems as Context-free Grammars

A Lindenmayer system, or L-System, consists of:

1. An alphabet of symbols: This alphabet consists of both the set of variables that can be replaced and those that cannot (terminals).
2. Initial axiom: This is the string of symbols from the alphabet defining the initial state of the system.
3. A collection of production rules: These rules have a predecessor and a successor. If each production rule in an L-system refers only to an individual symbol as the predecessor, it is a context-free grammar. Otherwise, it is considered context-sensitive.

Applications in Biology

In 1968, Aristid Lindenmayer, a Hungarian theoretical biologist and botanist, created L-Systems to describe how plant cells behave and to model growth processes in plant development. The key difference between L-Systems and other Chomsky grammar, which makes L-Systems better suited for modeling biological growth, is that production rules in L-Systems are applied in parallel, simultaneously replacing all variables in a strings. This helps model biological development, because "productions are intended to capture cell divisions in multicellular organisms, where many divisions may occur at the same time."[1]

The original L-System was used by Lindenmayer to model the growth of algae. The alphabet for this model consists of variables A and B but no terminals. The axiom, or initial state, is A. The production rules are as follows:

$$A \rightarrow AB$$
$$B \rightarrow A$$

The output of this system at each iteration can be determined using the first Python code submitted, as displayed here:

```
# L-system for algae growth using regular CFG-type definition

# variable A
def algae_a(n):
    if n == 0:
        print('A', end="")
    else:
        algae_a(n-1)
        algae_b(n-1)

# variable B
def algae_b(n):
    if n == 0:
        print('B', end="")
    else:
        algae_a(n-1)

iterations = 8 # how many rounds of substitutions to make
for i in range(iterations):
    print('n =', i, end=" ")
    algae_a(i) # enters axiom state
    print('')
```

Output:

```
n = 0 A
n = 1 AB
n = 2 ABA
n = 3 ABAAB
n = 4 ABAABABA
n = 5 ABAABABAABAAB
```

[1] Prusinkiewicz, Przemyslaw, and Aristid Lindenmayer. "Chapter 1. Graphical Modeling Using L-Systems." The Algorithmic Beauty of Plants. New York: Springer-Verlag, 1990. Print

```
n = 6 ABAABABAABAABABAABABA
n = 7 ABAABABAABAABABAABABAABAABABAABAAB
```

An interesting feature of this L-system is that the length of each string generated follows the Fibonacci sequence, skipping the first 1. Therefore, this L-system can be modeled with the concatenation, $G(n) = G(n-1)G(n-2)$, as written in the second Python program submitted:

```
# L-system for algae growth using G(n) = G(n-1)G(n-2) concatenation algorithm

def algae(n):
    if n == 0:
        print('A', end="")
    elif n == 1:
        print('AB', end="")
    else:
        algae(n-1)
        algae(n-2)

iterations = 8 # how many rounds of substitutions to make
for i in range(iterations):
    print('n =', i, end=" ")
    algae(i)
    print('')
```

Applications in Computer Graphics

Graphical application of L-Systems occur in two steps: 1) generate a string of symbols using the L-System and 2) interpret that string as a sequence of commands for drawing on a computer screen. L-Systems were not applied to picture generation until 1984 when Aono and Kunii used them to create realistic-looking images of trees and plants. Soon, what is called "turtle interpretation" became the chosen method for converting from a string of symbols to a drawing of an L-System. Turtle interpretation is the idea that there is a turtle on the screen with certain features pertaining to its position. This turtle follows commands, created from production rules of the L-System, to move around the screen, drawing a line in its path. A state of the turtle is a triplet (x, y, a), where the coordinates (x, y) represent the turtle's position, and angle α, called the turtle's heading, is interpreted as the direction in which the turtle is facing[2]. This is the method used in the python programs submitted using the turtle library. The following are images generated from these programs:

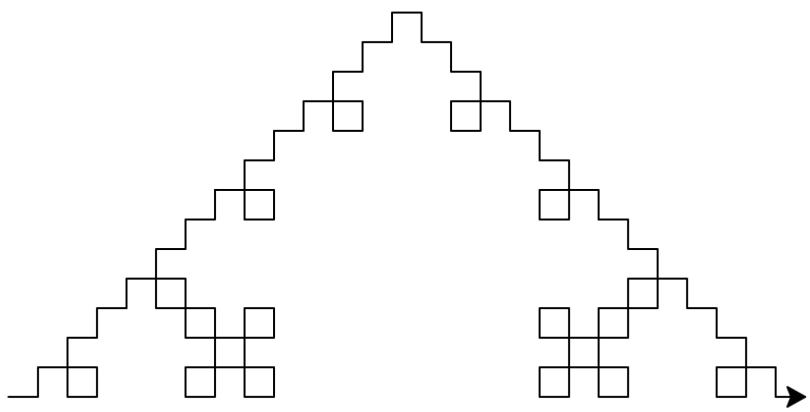
[2]Prusinkiewicz, Przemyslaw. "Graphical Application of L-Systems." ResearchGate. Web. 22 Mar. 2016.

The Koch Curve is created using an L-System with alphabet of variable F and terminals + and -. The axiom state is F. The production rule is:

$$F \rightarrow F + F - F - F + F$$

To convert this L-System to a turtle graphic, the + sign is taken as a 90 degree turn to the left, the - sign is a 90 degree turn to the right, and F is a move forward of a certain size.

Figure 1: Koch Curve, 90 degree angles, with n = 3 and size = 15

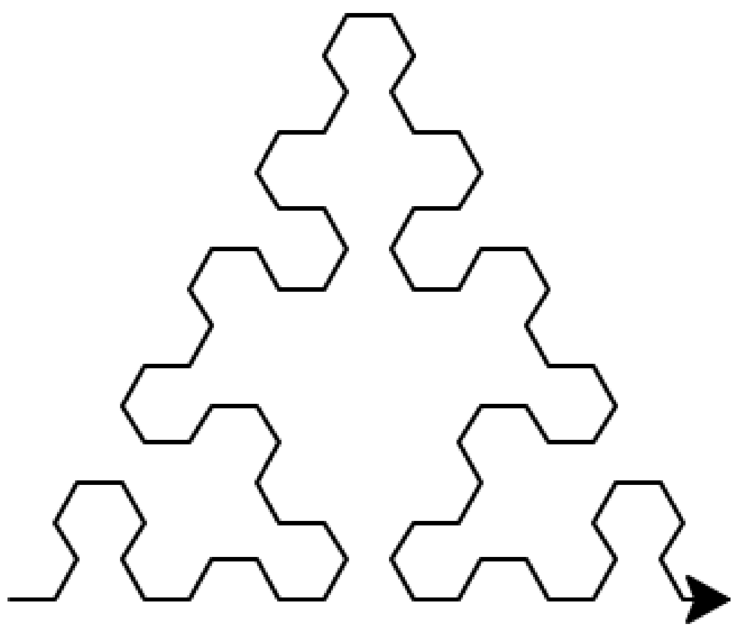


The Sierpinski Triangle is created using an L-System with alphabet of variables A and B and terminals + and -. The axiom state is A. The production rule is:

$$A \rightarrow +B - A - B+$$
$$B \rightarrow -A + B + A-$$

To convert this L-System to a turtle graphic, the + sign is taken as a 60 degree turn to the left, the - sign is a 60 degree turn to the right, and both A and B are moves forward of a certain size.

Figure 2: Sierpinski Triangle with n = 4 and size = 10

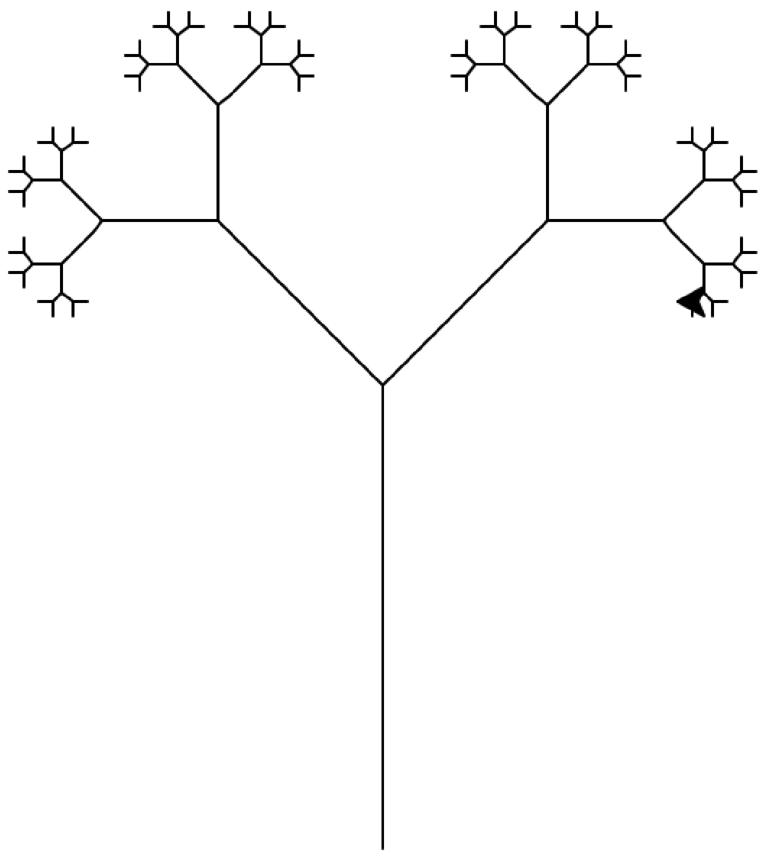


The Pythagoras tree is created using an L-System with alphabet of variables 0 and 1 and terminals [and]. The axiom state is 0. The production rule is:

$$0 \rightarrow 1[0]0$$
$$1 \rightarrow 11$$

To convert this L-System to a turtle graphic, the 0 and 1 mean to move forward by a set size. [pushes the current position and angle onto a stack and turns left by 45 degrees.] pops the position and angle, returns to them without drawing, and turns right 45 degrees.

Figure 3: Pythagoras Tree with n = 6 and size = 5



The fractal plant is created using an L-System with alphabet of variables X and F and terminals +, -, [, and]. The axiom state is X. The production rule is:

$$X \rightarrow F - [[X] + X] + F[+FX] - X$$
$$F \rightarrow FF$$

To convert this L-System to a turtle graphic, F means to draw forward. X does not correspond to any action, but is a placeholder to shape the curve. The - sign is taken as a 25 degree turn to the left. The + sign is a 25 degree turn to the right. [pushes the current position and angle onto a stack.] pops the position and angle and returns to them without drawing.

Figure 4: Fractal Plant with n = 4 and size = 20