Fractals and L-systems

“L-systems are a mathematical formalism proposed by the biologist Aristid Lindenmayer in 1968 as a foundation for an axiomatic theory of biological development. Two principal areas include generation of fractals and realistic modelling of plants.

Central to L-systems, is the notion of rewriting, where the basic idea is to define complex objects by successively replacing parts of a simple object using a set of rewriting rules or productions. The rewriting can be carried out recursively.

The most extensively studied and the best understood rewriting systems operate on character strings. Chomsky's work on formal grammars (1957) spawned a wide interest in rewriting systems. Subsequently, a period of fascination with syntax, grammars and their application in computer science began, giving birth to the field of formal languages.

Aristid Lindenmayer's work introduced a new type of string rewriting mechanism, subsequently termed L-systems. The essential difference between Chomsky grammars and L-systems lies in the method of applying productions. In Chomsky grammars productions are applied sequentially, whereas in L-systems they are applied in parallel, replacing simultaneously all letters in a given word. This difference reflects the biological motivation of L-systems. Productions are intended to capture cell divisions in multicellular organisms, where many division may occur at the same time.”

Simple Cells

Think of this as a system for modeling the growth of a 1-dimensional line of algae. Cell type A can divide into an A next to a B, while cell type B must mature into an A before it can divide.

**variables:** A, B  
**start:** B  
**rules:** (A → AB), (B → A)

Fill in the blanks for each generation.

<table>
<thead>
<tr>
<th>t</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>B</td>
</tr>
<tr>
<td>1</td>
<td>A</td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>ABAAB</td>
</tr>
<tr>
<td>5</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>ABAABABAABABAABABAABABA</td>
</tr>
<tr>
<td>8</td>
<td>ABAABABAABAABABAABABAABABAABABAABABA</td>
</tr>
</tbody>
</table>

Count the number of cells in each generation... do you notice anything? (Optional: Can you try to prove why this is happening?)
Dragon Curve

This system is for generating the rules to draw a dragon curve.

variables: A, B  
start: B  
rules: (B → B−A), (A → A+B)  
after-rules: A or B means “draw forward”  
+ means “turn left 90°”  
− means “turn right 90°”

Fill in the blanks for each generation.

t = 0:  
t = 1:  
t = 2:  
t = 3:  
t = 4:  
t = 5:  

Try to draw a Dragon Curve using one of the results that you get!