Lindenmayer systems

L- systems were defined by Lindenmayer in an attempt to describe the development of multi-cellular organisms. In the study of developmental biology , the important changes that take place in cells and tissues during development are considered. L-systems provide a framework within which these aspects of development can be expressed in a formal manner . L-systems also provide a way to generate interesting classes of pictures by generating strings and interpreting the symbols of the string as the moves of the cursor.

From the formal language theory point of view, L-systems differ from the Chomsky grammars in the following three ways :

- Parallel re-writing of symbols is done at every step. This is the major difference.
- There is no distinction between nonterminals and terminals(In extended L-system , we try to introduce the distinction) .
- starting point is a string called the axiom .

Definition 1

A 0L system is an ordered triple $\pi = \left(V, w_0, P\right)$ where V is an alphabet , w_0 a non-empty word over V which is called the axiom or initial word ; and P is a finite set of rules of the form $a \rightarrow \alpha, a \in V$ and $a \in V^*$. Furthermore , for each $a \in V$

There is at least one rule with a on the left hand side (This is called the completeness condition)

The binary relation $\Rightarrow$ is defined as follows : If $a_1 \ldots\ldots\ldots a_n$ is a string over V and $a_i \rightarrow w_i$ are the rules in p ,

$$a_1 \ldots\ldots\ldots a_n \Rightarrow w_1 \ldots\ldots w_n$$

$\overset{*}{\Rightarrow}$ is the reflexive transitive closure of $\Rightarrow$. The language generated by the 0L system is:

$$L\left(\pi\right) = \left\{ w \mid w \in V^*, w_0 \overset{*}{\Rightarrow} w \right\}$$

Definition 2

A 0L system $\pi = \left(V, w_0, P\right)$ is deterministic if for every $a \in V$ there is exactly one rule in P with a on the left hand side. It is propagating $\left(\varepsilon - free\right)$, if ε is not on the right hand side of any production. Notations DOLS, P0LS, and DP0LS are used for these systems.

The language generated by these systems are called DOL , P0L, and DP0L languages respectively.

Example 2

Consider the following DP0L system:

$$\pi_1 = \left(\left\{a,b\right\}, ab, \left\{a \to aa, b \to bb\right\}\right)$$

The derivation steps are :

$$ab \Rightarrow aabb \Rightarrow aaaabbbb \Rightarrow .....$$

$$L\left(\pi_1\right) = \left\{a^{2^n} b^{2^n} \mid n \geq 0\right\}$$

Example 2

Consider the following DP0L system:

$$\pi_2 = \left(\sum, 4, P\right)$$

Where $\sum = \{0,1,2,3,..........,9,(,)\}$

P has rules :

$$0 \to 10$$
$$1 \to 32$$
$$2 \to 3(4)$$
$$3 \to 3$$
$$4 \to 56$$
$$5 \to 37$$
$$6 \to 58$$
$$7 \to 3(9)$$
$$8 \to 50$$
$$9 \to 39$$
$$(\to ($$
$$) \to)$$

The 10 steps in the derivation are given below :

1 4

2 56

3 3758

4 $33(9)3750$

5 $33(39)33(9)3710$

6 $33(339)33(39)33(9)3210$

7 $33(3339)33(339)33(39)33(4)3210$

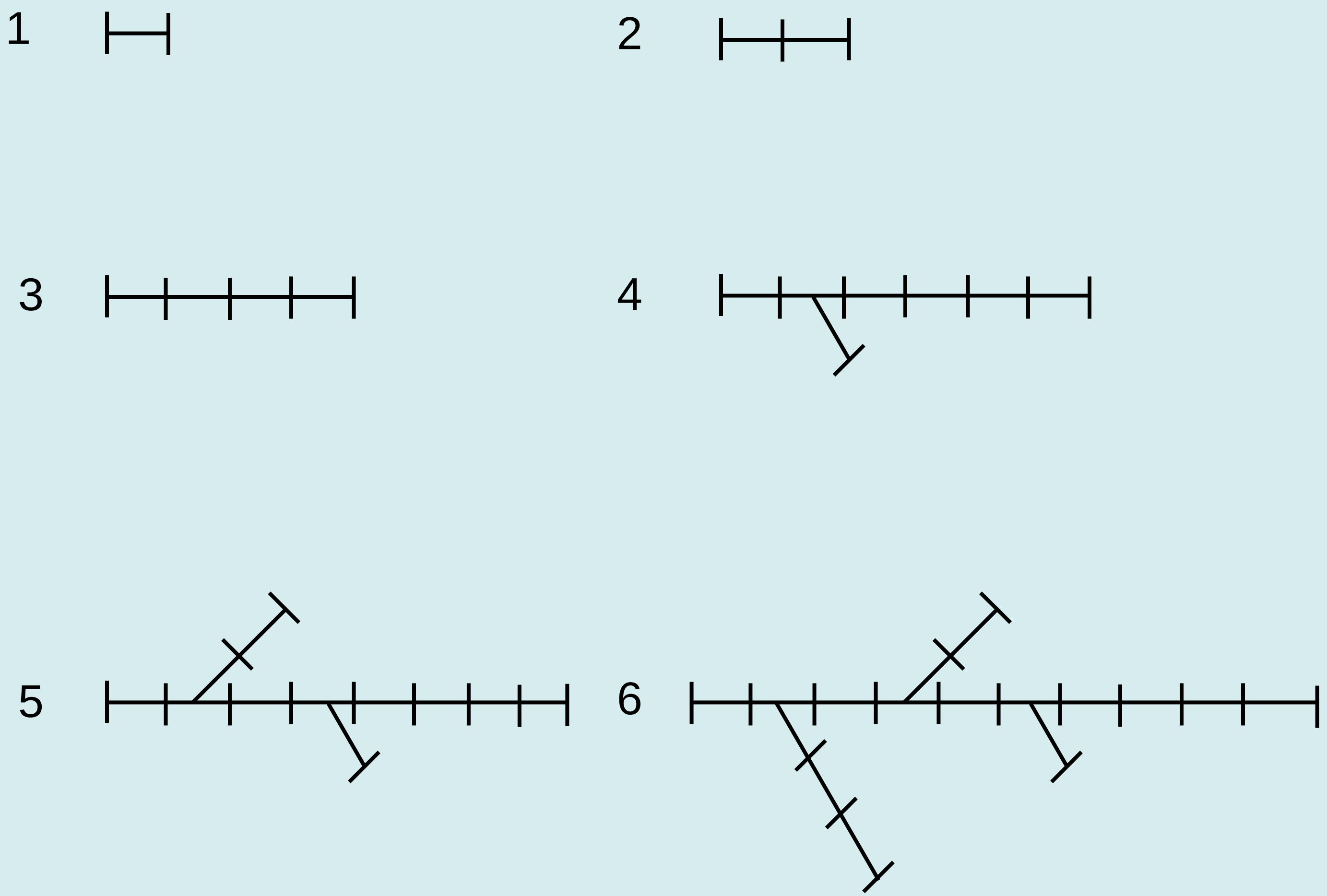
8 $33(33339)33(3339)33(339)33(56)33(4)3210$

9 $33(333339)33(33339)33(3339)33(3758)33(56)33(4)3210$

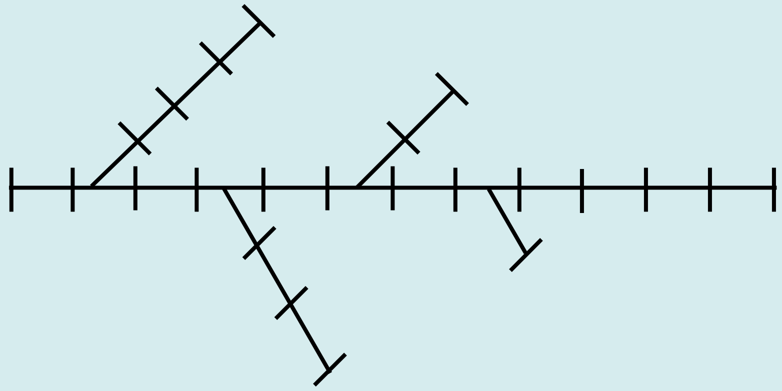
10 $33(3333339)33(333339)33(33339)33\big(33(9)3750\big)33(3758)$

$33(56)33(4)3210$

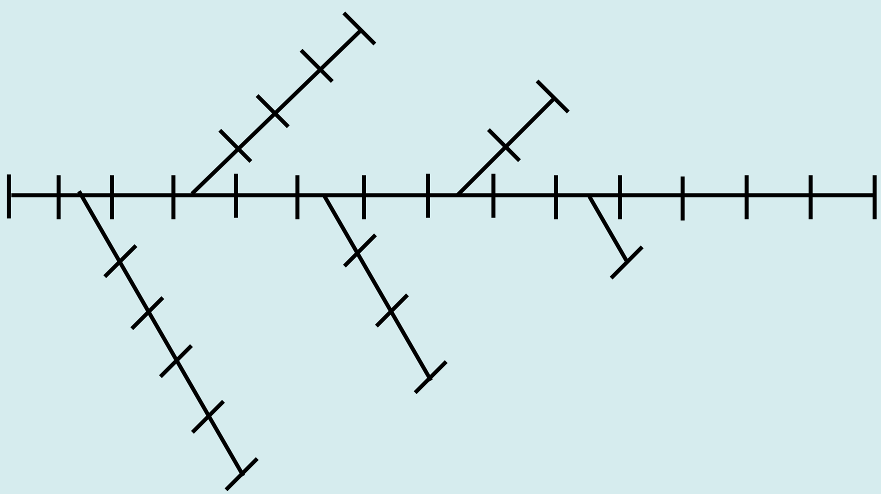
If the symbols are interpreted as steps , cells with the portion within ( ) representing a branch, we get the following growth pattern for each step :



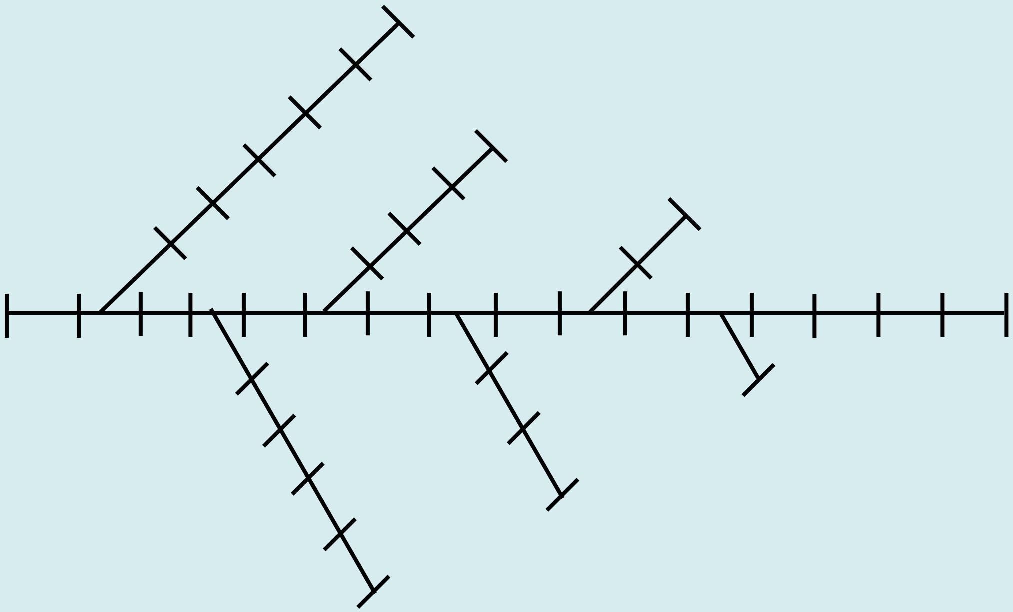
7



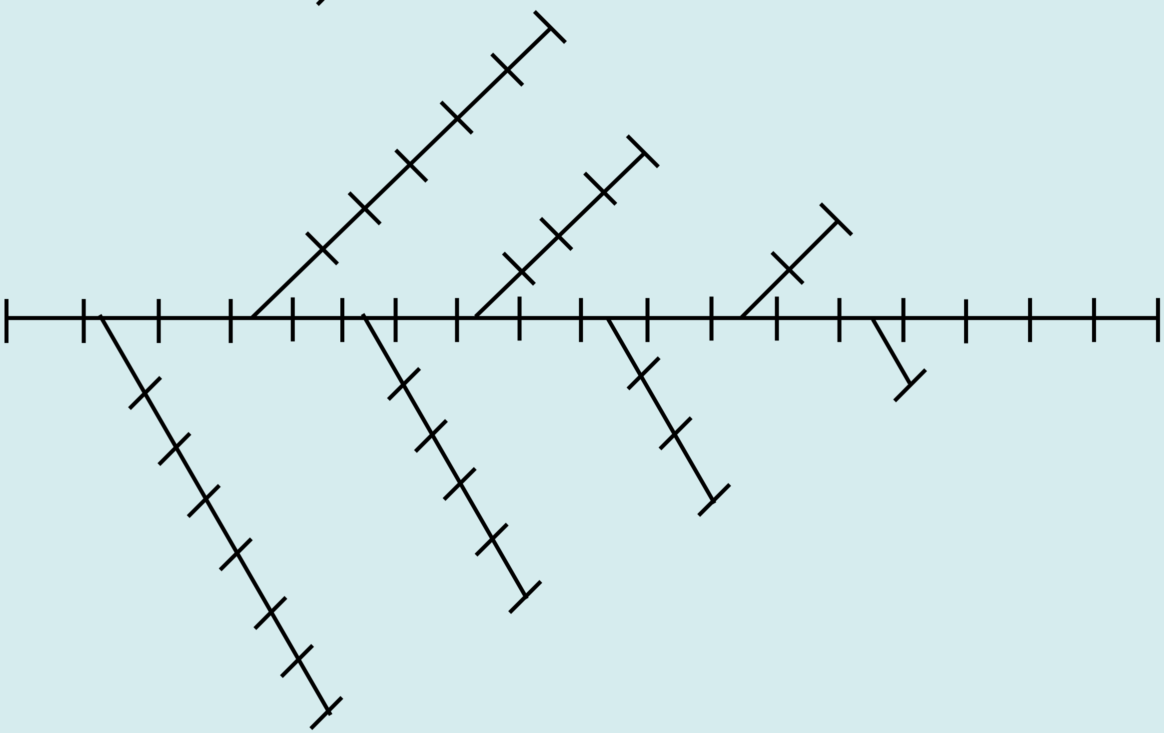
8



9



10



Example

Consider the deterministic and propagating 0L-system, where the alphabet and the production are given by the following table .

1	2	3	4	5	6	7	8	(	)	#	0
2#3	2	2#4	504	6	7	8(1)	8	(	)	#	0

(The right side of each production is in the second row .) starting with the axiom $P_0 = 1$, we get the following words in the language L(0LS).

$$P_0 = 1$$

$$P_1 = 2\#3$$

$$P_2 = 2\#2\#4$$

$$P_3 = 2\#2\#504$$

$$P_4 = 2\#2\#60504$$

$$P_5 = 2\#2\#7060504$$

$$P_6 = 2\#2\#8\left(1\right)7060504$$

It can be verified inductively that for all $n \geq 0$,

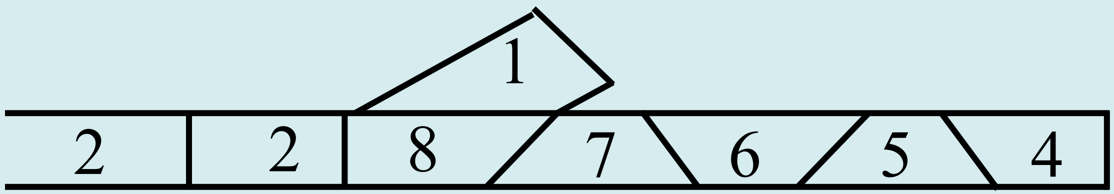
$$P_{n+6} = 2\#2\#8\left(P_n\right)08\left(P_{n-1}\right)0.....08\left(P_0\right)07060504.$$

The developmental stages P_6 , P_8 and P_{13} are illustrated in figure . Parenthesized expressions are branches whose position is indicated by the 8’s . The 0’s are marked by oblique walls drawn alternately right and left inclined . The branches are shown as attached on alternative sides of the branch on which they are borne , and the #’s are marked by vertical walls. It is easy to verify that L(0LS) is context-sensitive but not context-free.

P_6 $2\#2\#8(1)07060504$

P_7 $2\#2\#8(2\#3)08(1)07060504$

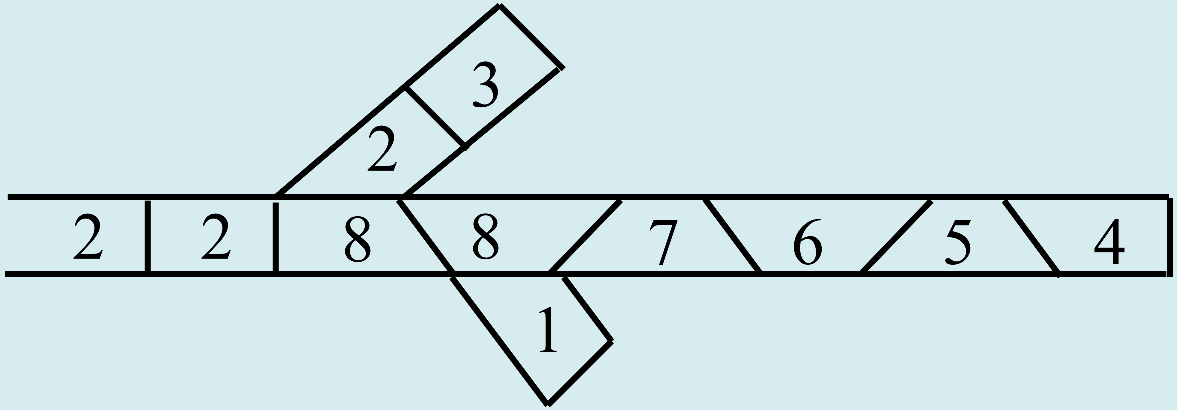
P_8 $2\#2\#8(2\#2\#4)08(2\#3)08(1)07060504$



P_6 $\quad 2\#2\#8(1)07060504$

1	2	3	4	5	6	7	8	(	)	#	0
2#3	2	2#4	504	6	7	8(1)	8	(	)	#	0

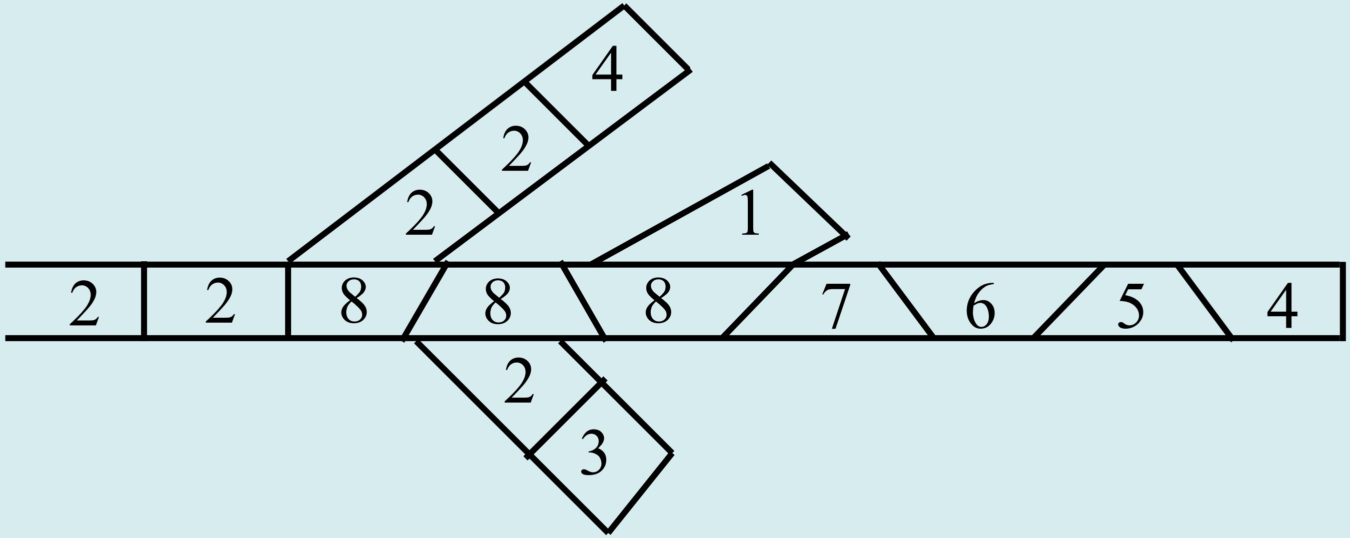
P_7 $\quad 2\#2\#8(2\#3)08(1)07060504$



P_7 $\qquad 2\#2\#8\left(2\#3\right)08\left(1\right)07060504$

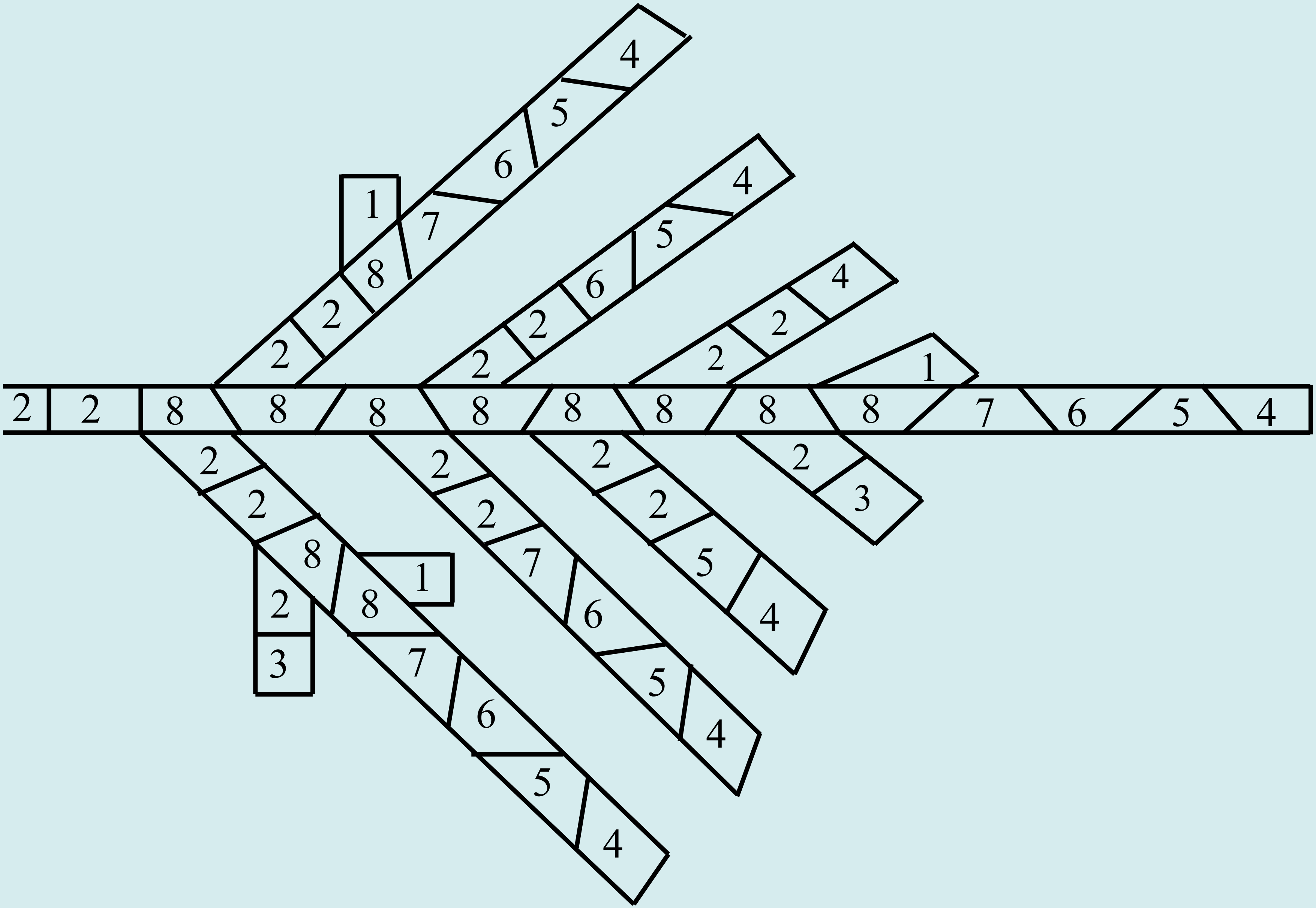
1	2	3	4	5	6	7	8	(	)	#	0
2#3	2	2#4	504	6	7	8(1)	8	(	)	#	0

P_8 $\qquad 2\#2\#8\left(2\#2\#4\right)08\left(2\#3\right)08\left(1\right)07060504$

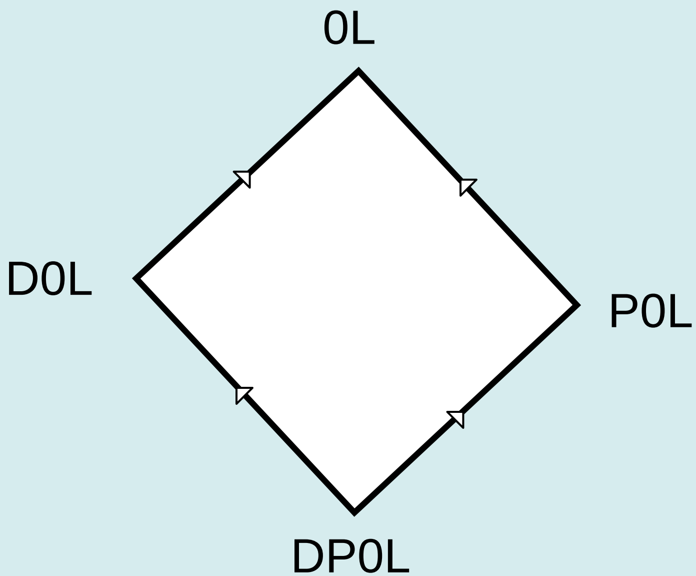


P_8 $\qquad 2\#2\#8\left(2\#2\#4\right)08\left(2\#3\right)08\left(1\right)07060504$

1	2	3	4	5	6	7	8	(	)	#	0
2#3	2	2#4	504	6	7	8(1)	8	(	)	#	0


2 2 8 8 8 8 8 8 8 8 7 6 5 4
2 2 8 7 6 5 4
1
2 2 6 5 4
2 2 4
1
2 2 8 8 7 6 5 4
2 3
1
2 2 7 6 5 4
2 2 5 4
2 3

The following hierarchy can be easily seen as



It can be seen that the family of 0L languages is not closed under any of the usual operation (AFL operation) union , concatenation , Kleene closure , $\varepsilon - free$ homomorphism, intersection with regular sets, and inverse homomorphism . For example even though {a} and {aa} are 0L languages generated by 0L systems with appropriate axioms and rule $a \rightarrow a$ we find their union {a, aa } is not a 0L language . If it is a 0L language the 0L system generating it will have a axiom either a or aa. If a is the axiom , to generate aa , there must be a rule $a \rightarrow aa$, in which case other strings will also be generated . If aa is the axiom to generate , we must have $a \rightarrow a$ and $a \rightarrow \varepsilon$ in which case ε will also be generated. In a similar manner , we can give examples for nonclosure under other operations. This also shows that there are finite languages which are not 0L.

Definition 3

A tabled 0L(T0L) system is an ordered triple $T\pi = (V, w_0, P)$, where V is an alphabet , w_0 is the axiom and P is a finite set of tables. Each table contains rules of the form $a \to \alpha$ for $a \in V$. Each table will have at least one rule with **a** on the left hand side for each $a \in V$ (completeness condition). If $\alpha \neq \beta$ is a derivation step $\alpha = a_{1}.......a_n , \beta = \beta_{1}.......\beta_m,\ a_i \to \beta_i \in t$, where t is a table in P . i.e., in one step, only rules from the same table should be used. $\overset{*}{\Rightarrow}$ Is the reflexive transitive closure of $\Rightarrow$. The language generated is :

$$L(T\pi) = \left\{ w \mid w \in V^*, w_0 \overset{*}{\Rightarrow} w \right\}$$

Example 3

Consider the T0L system :

$$T\pi = \left(\left\{a\right\}, a, \left\{\left\{a \to a^2\right\}, \left\{a \to a^3\right\}\right\}\right).$$

$$L\left(T\pi\right) = \left\{a^i \mid i = 2^m 3^n \ \mathit{for}\, m, n \geq 0\right\}$$

A T0L system is deterministic , if each table has exactly one rule for each $a \in V$. It is propagating if $\varepsilon - rules$ are not allowed. We can hence talk about DT0L systems , PT0L systems , DPT0L systems and the corresponding languages .

Extended System

Nondistinction between nonterminals and terminals affected the closure properties . From formal language theory point of view , extended systems are defined which make the families defined closed under many operations . Here the system has two sets of symbols , terminals and nonterminals , or total alphabet , and target alphabet.

Definition

An E0L system is defined as a 4-tuple $G=\left(V,\Sigma,w_0,P\right)$ where $\left(V,w_0,P\right)$ is a 0L system and $\Sigma\subseteq V$ is the target alphabet. $\overset{*}{\Rightarrow}$ and $\Rightarrow$ are defined in the usual manner. The language generated is defined as :

$$L\left(G\right)=\left\{w \mid w\in\Sigma^*, w_0\overset{*}{\Rightarrow}w\right\}$$

In a similar manner, ET0L systems can be defined specifying the target alphabet.

Example 4

Let G be $\left(\{S, a\}, \{a\}, S, \{S \to a, S \to aa, a \to a\}\right)$ be an E0L system.

The language generated is $\{a, aa\}$ which is not a 0L language .

Systems with Interactions

Definition

A 2L system is an ordered 4-tuple $H=(V,w_0,P,\$)$ where V and w_0 are as in 0L system. $\$\in V$ Is the input from environment and P is a finite set of rules of the form $<a,b,c>\rightarrow w, b\in V, a,c\in V\bigcup\{\$\}, w\in V^*$. $\Rightarrow$ is defined as follows:

$a_1......a_n$ is the sentinel form and $a_1......a_n\Rightarrow\alpha_1.......\alpha_n$

If $(a_{i-1},a_i,a_{i+1})\rightarrow\alpha_i$ *is in* P

for $2\leq i\leq n-1,(\$,a_1,a_2)\rightarrow\alpha_1,(a_{n-1},a_n,\$)\rightarrow\alpha_n\in P$

$\overset{*}{\Rightarrow}$ is the reflexive transitive closure of $\Rightarrow$ .i.e., for rewriting a symbol , the left and right neighbors are also considered. The language generated is defined as

$$L(H)=\left\{w\,|\,w\in V^*,w_0\overset{*}{\Rightarrow}w\right\}$$

If only the right neighbor (or left neighbor) is considered., it is called a 1L system i.e., A 2L system is a 1L system if and only if one of the following conditions hold.

1. for all a , b , c , d in V , P contains $\left(a,b,c\right)\rightarrow\alpha$ if and only if P contains $\left(a,b,d\right)\rightarrow\alpha$ *for all* $d\in V\bigcup\{\$\}$ *or*

2. $\left(a,b,c\right)\rightarrow\alpha$ is in P if and only if for all $d\in V\bigcup\{\$\}\left(d,b,c\right)\rightarrow\alpha$ is in P.

The corresponding languages are called 2L and 1L languages .

Example

Consider 2L systems:

$$H_2 = \left(\{a,b\}, a, P, \$\right)$$

where P is given by

$$(\$, a, \$) \rightarrow a^2 \mid a^3 \mid a^3 b \mid b a^3$$

$$(x, b, y) \rightarrow b \mid b^2$$

$$(a, a, a) \rightarrow a$$

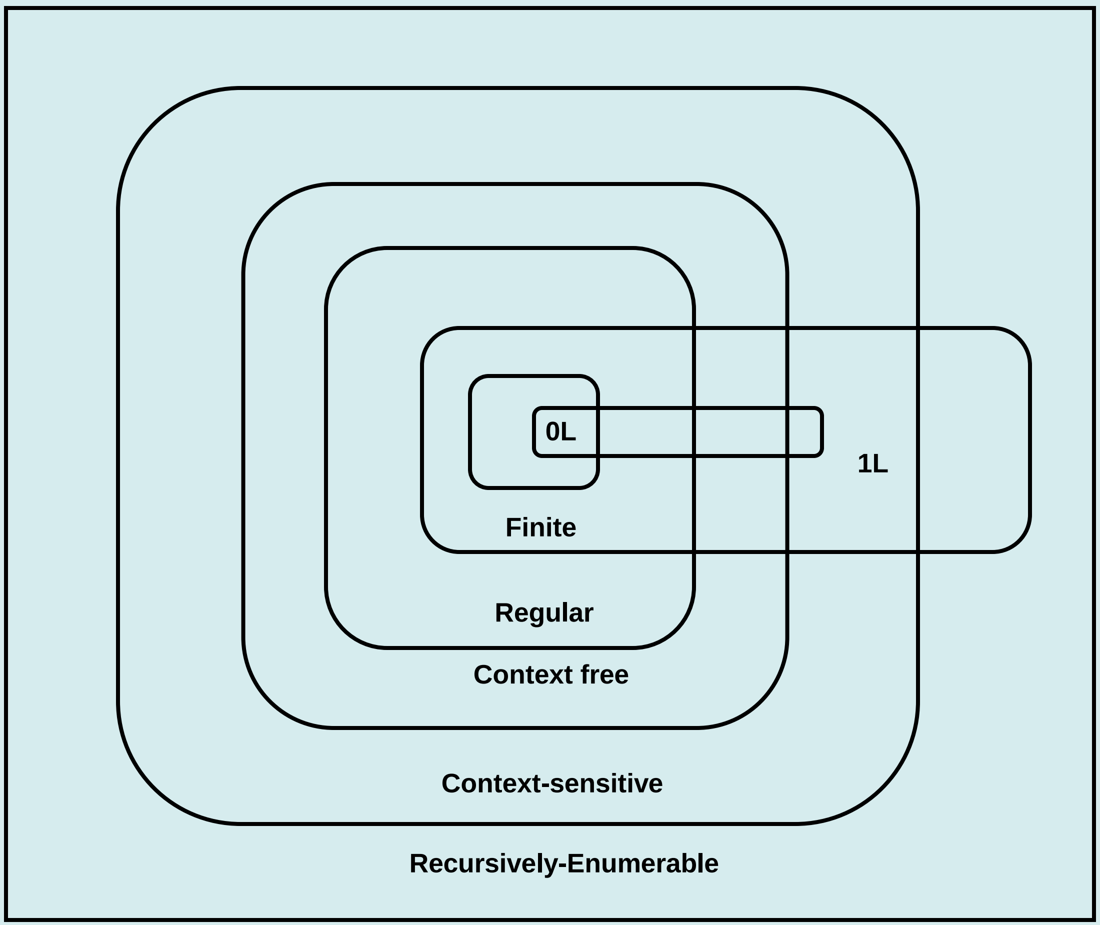
$$(a, a, b) \rightarrow a$$

$$(\$, a, a) \rightarrow a$$

$$x, y \in \{a, b, \$\}$$

$$L(H_2) = \{a, a^2, a^3, a^3 b^*, b^* a^3\}$$

The hierarchy of the language families generated is shown in the below figure:



Deterministic 0L systems are of interest because they generate a sequence of strings called D0L sequence. The next step generates a unique string. The lengths of the sequence of the strings may define a well-known function. Such a function is called a growth function. Consider the following example

Example

Consider the following 0L systems :

1. $S = \left(\{a\}, a, \{a \to a^2\}\right)$, we have

$$L(S) = \left\{a^{2^n} \mid n \geq 0\right\}.$$

The above language is a DP0L-language . The growth function is $f(n) = 2^n$

2. $S = \left(\{a,b\}, a, \{a \to b, b \to ab\}\right)$, the words in the ***L***(S) are :

$$a, b, ab, bab, abbab, bababbab.....$$

The length of these words are the squares of the Fibonacci numbers

3. $S=\left(\{a,b,c\},a,\{a\to abcc,b\to bcc,c\to c\}\right)$, the words in L(S) are :

$$a,abcc,abccbcccc,abccbccccbcccccc......$$

The lengths of these words are the squares of the natural numbers.

4. $S=\left(\{a,b,c\},a,\{a\to abc,b\to bc,c\to c\}\right)$, the words in L(S) are:

$$a,abc,abcbcc,abcbccbccc,abcbccbcccbcccc.....$$

The lengths of these words are the triangular numbers.

5. $S=\left(\{a,b,c,d\},a,\{a\to abcd^5,b\to bcd^5,c\to cd^6,d\to d\}\right)$, the words in ***L***(S) are:

$$a,abcd^5,abcd^5bcd^5cd^6d^5......$$

The lengths of these words are the cubes of natural numbers.

Two D0L systems are growth equivalent , if they have the same growth function.

Let $G=\left(V,w_0,P\right)$ be a D0L system. Then, let there be a n-dimensional row vector π which is the Parikh mapping of w_0. M is a $n\times n$ matrix whose ith row is the Parikh mapping of α_i where $a_i \rightarrow \alpha_i \in P$. η is an

n-dimensional column vector $\begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$ Consisting of 1's. Then, $f\left(n\right)$, the

growth function is given by $f\left(n\right)=\pi M^{\eta}\eta.$

Example

$$\pi = \left(\{a,b,c\}, a, \{a \to abcc, b \to bcc, c \to c\}\right).$$

The strings generated in a few steps and their lengths are given below.

w_0	a	1
$step\,1$	$abcc$	4
$step\,2$	$abccbcccc$	9
$step\,3$	$abccbccccbcccccc$	16

It is obvious that $f(n) = (n+1)^2$

$\pi\ is\ (1,0,0)$

$$M \text{ is } \begin{pmatrix} 1 & 1 & 2 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{pmatrix} \quad , \quad \eta \text{ is } \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

$$\pi M \eta \text{ is } = (1,0,0) \begin{pmatrix} 1 & 1 & 2 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{pmatrix} \qquad \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

$$= (1\ 0\ 0) \begin{pmatrix} 4 \\ 3 \\ 1 \end{pmatrix} = 4$$

$$\pi M^2 \eta = 9 \text{ and so on.}$$

The growth function f (n) is termed as malignant, if and only if there is no polynomial p(n) such that $f(n) \leq p(n)$ for all n .

Applications :

L systems are used for a number of applications in computer imagery. It is used in the generation of fractals, plants, and for object modeling in three dimensions . Applications of *L*-systems can be extended to reproduce traditional art and to compose music.

Two-Dimensional Generation of Patterns :

Here, the description of the string is captured as a string of symbols . An *L*-system is used to generate this string. This string of symbols is viewed as commands controlling a LOGO-like turtle. The basic commands used are move forward, make right turn, make left turn etc. Line segments are drawn in various directions specified by the symbols to generate the straight line pattern . Since most of the patterns have smooth curves , the positions after each move of the turtle are taken as control points for *B*-spline interpolation. We see that this approach is simple and concise.

Fractals Generated by L-Systems

Many fractals can be thought as a sequence of primitive elements . These primitive elements are line segments. Fractals can be coded into strings . Strings that contain necessary information about a geometric figure can be generated by *L*-systems . The graphical interpretation of the string can be described based on the motion of a LOGO – like turtle.

A state of the turtle is defined as a triplet (x, y ,A), where the Cartesian coordinates (x, y) represent the position of the turtle and angle A , called the turtles heading, is interpreted as the direction in which the turtle is facing. Given the step size ***d*** and the angle δ , the turtle can move with respect to the following symbols.

f : move forward a step length *d*. The state of the turtle changes to $\left(x', y', A\right)$, where $x' = x + d * \cos(A)$ *and* $y' = y + d * \sin(A)$ A line is drawn between the points $\left(x, y\right)$ and $\left(x', y'\right)$

F: move forward as above but without drawing the line .

+ : Turn the turtle left by an angle δ . The next state of the turtle will be $\left(x, y, A+\delta\right)$

Positive orientation of the angle is taken as anti-clockwise .

- : Turn the turtle as above but in clockwise direction .

Interpretation of a String

Let S be a string and (x_0, y_0, A_0) be the initial state of the turtle , and step size ***d*** , angle increment δ are the fixed parameters . The pattern drawn by the turtle corresponding to the string S is called turtle interpretation of the string S.

Consider the following *L* system.

Axiom : w : $f + f + f + f$

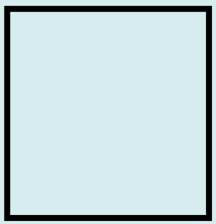
Production : $f \to f + f - f - ff + f + f - f$

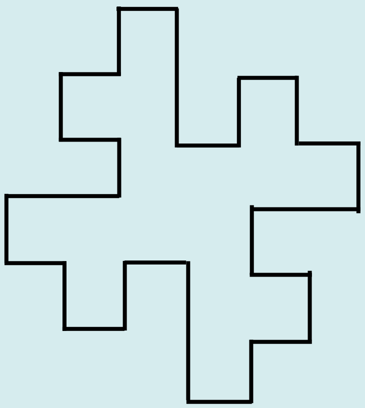
The above L-system is for 'Koch island'

The images corresponds to the string generated for different derivation steps n=0,1,2 is shown in the Figure. The angle increment δ is 90^0 . The step size *d* could be any positive number . The size of the 'Koch island ' depends on the step size and the number of derivation steps .

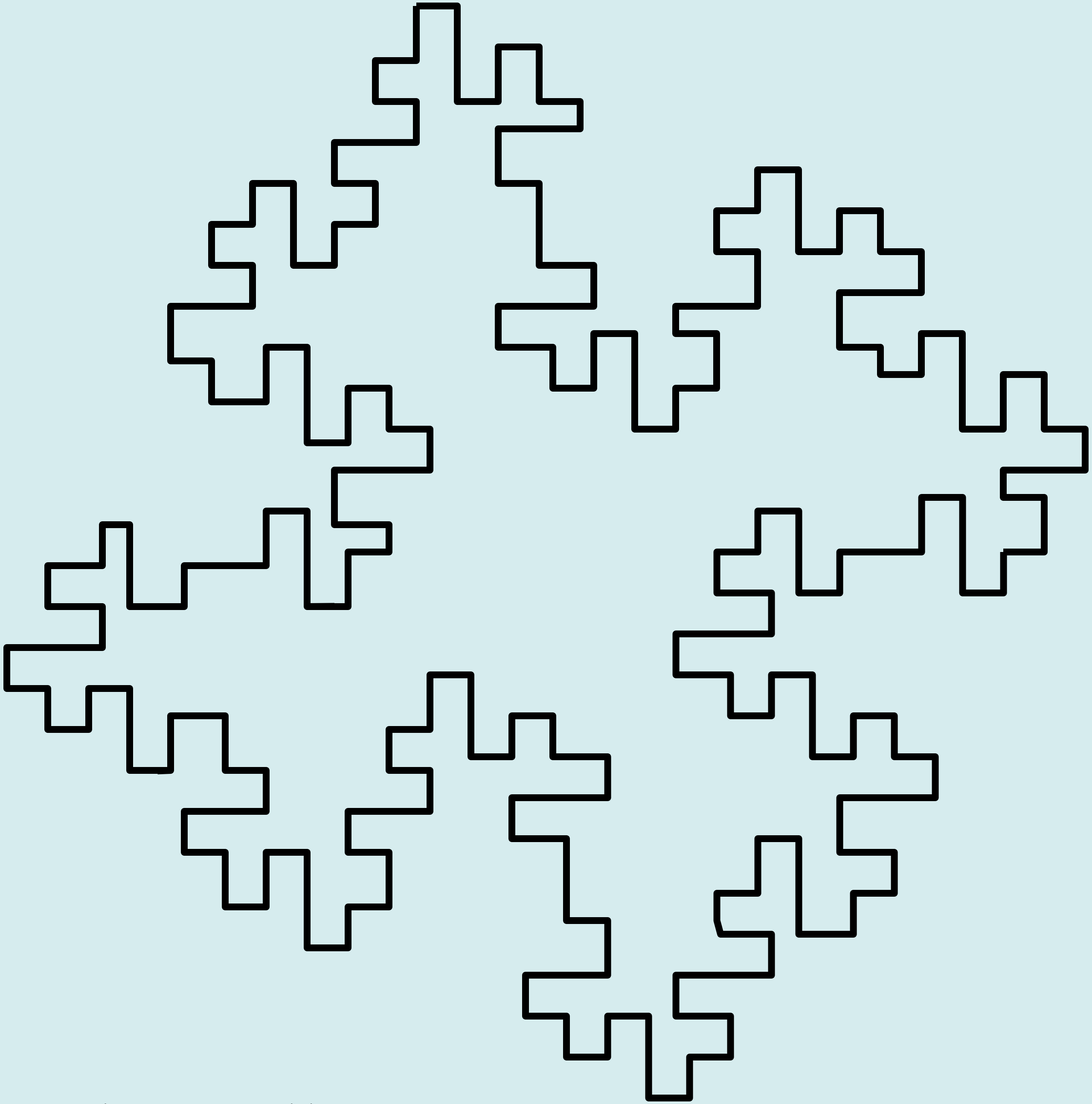
Koch construction are a special case of *L*-systems. The initiator corresponds to the axiom in the *L*-systems. The generator is represented by the single production.

Koch island





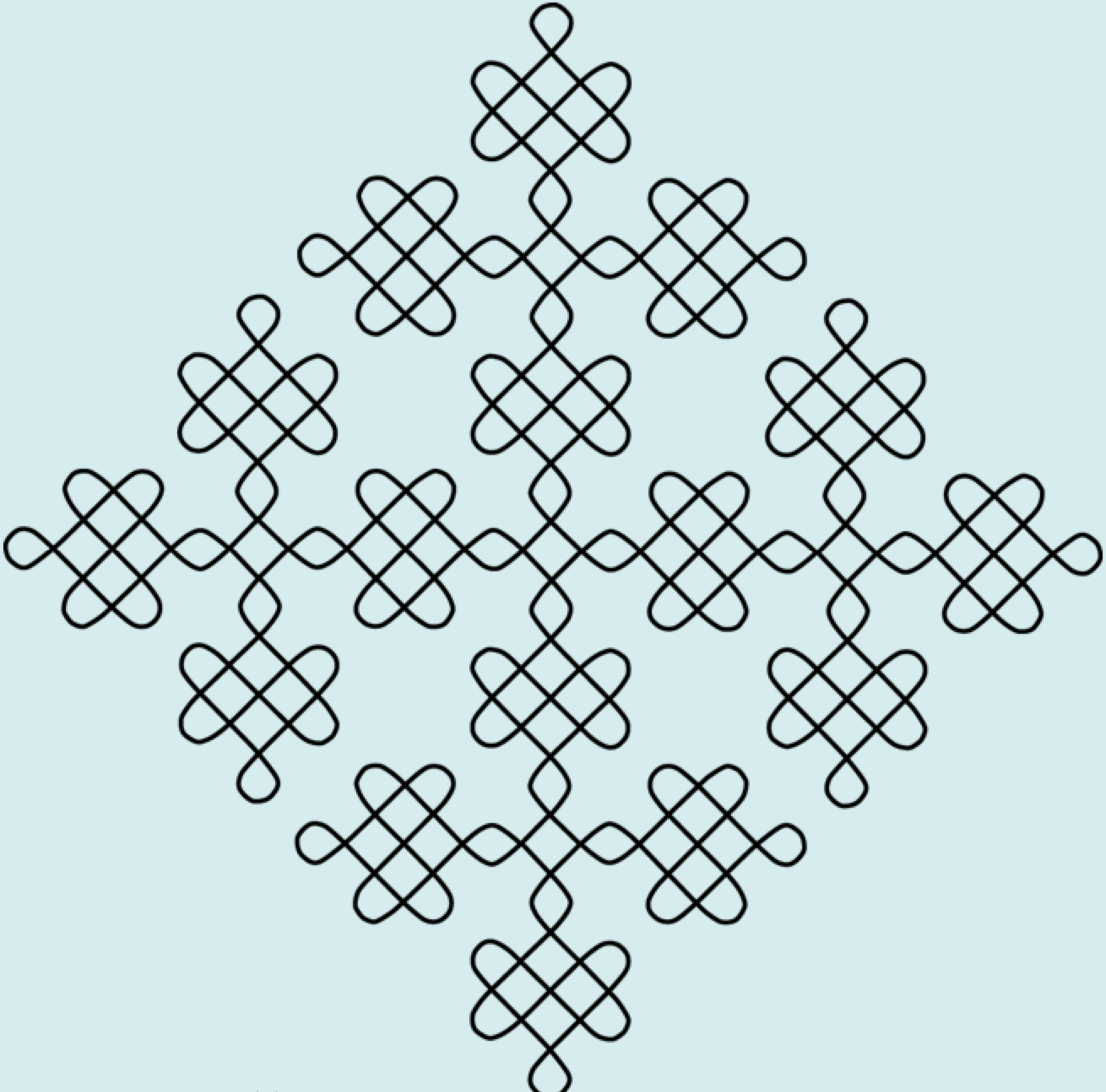
$$f \rightarrow f + f - f - ff + f + f - f$$



Fractals generated by *L*-system

Interpolation

Consecutive positions of the turtle can be considered as control points specifying a smooth interpolating curve. B-spline interpolation is used for most of the kolam patterns.



Kolam patterns generated by *L* - system

Candies

The above figure shows the kolam, pattern ‘Candies’ which can be generated by the following *L*-system.

Axiom : $\left(-D--D\right)$

Productions :

$$A \rightarrow f++f\ f\ f\ f--f--f\ f\ f\ f++f++f\ f\ f\ f--f$$

$$B \rightarrow f--f\ f\ f\ f++f++f\ f\ f\ f--f--f\ f\ f\ f++f$$

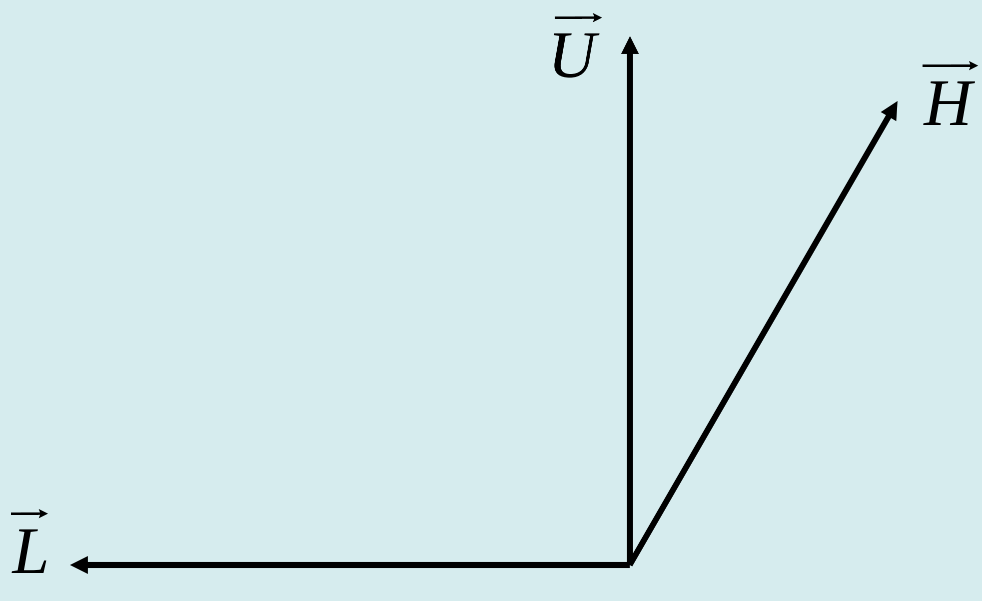
$$C \rightarrow B\ f\ A--B\ f\ A$$

$$D \rightarrow C\ f\ C--C\ f\ C$$

Angle increment $= 45^o$

Generation of Plant Structure

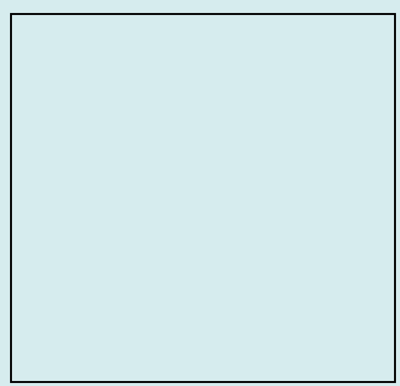
For generating living organisms like plants , a three-dimensional turtle is used. A three dimensional turtle is different in some respects , compared to the two-dimensional one. It has additional parameters for width and color and is represented by a 6-tuple < P, H , L, U, *w*, c > where the position vector P represents the turtle's position in Cartesian coordinates , and vectors H, L and U represents the turtle's orientation. *w* represents the width of the turtle. *c* corresponds to the color of the lines drawn.



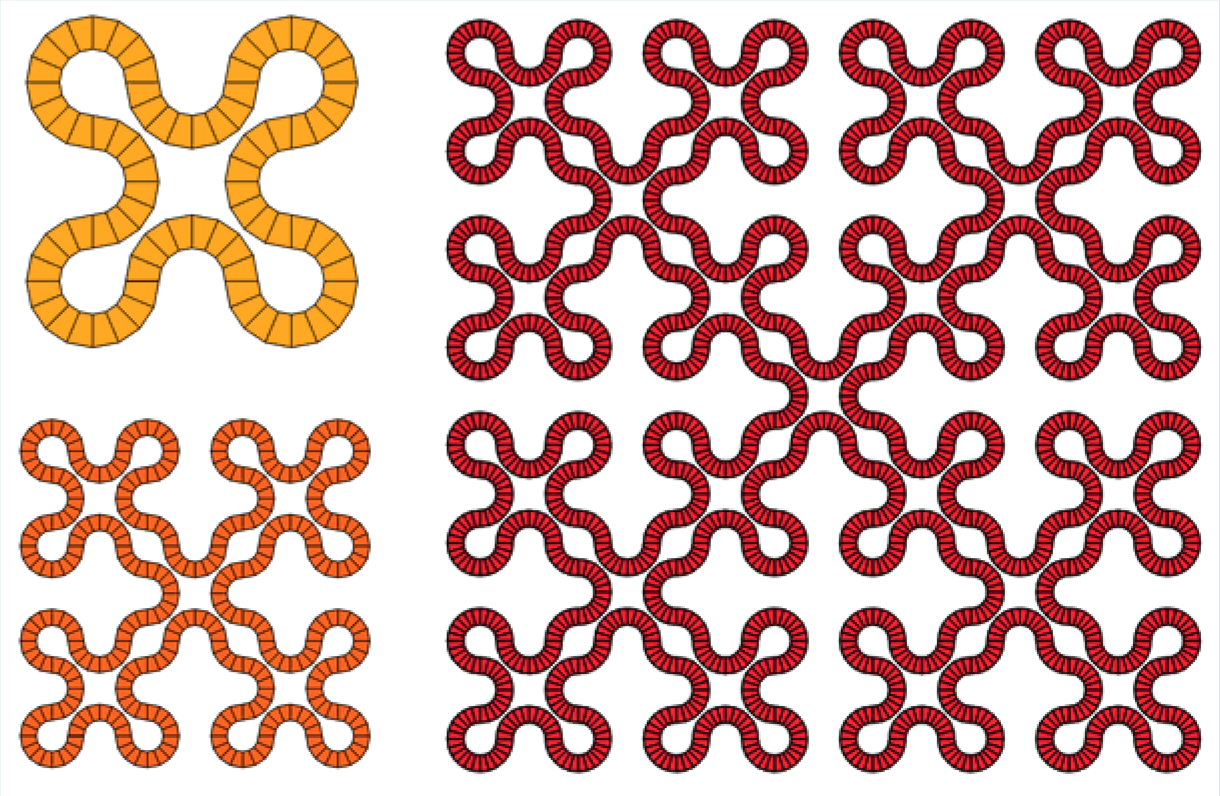
For generating plant like structures, some special L-systems are used . They are bracketed $2L$-systems and stochastic L-system. In bracketed L-systems , brackets are used to separate the branches from the parent stem. In stochastic L-systems , probabilities are attached to the rules which can capture the variations in plant development.

L Systems and Computer Imagery

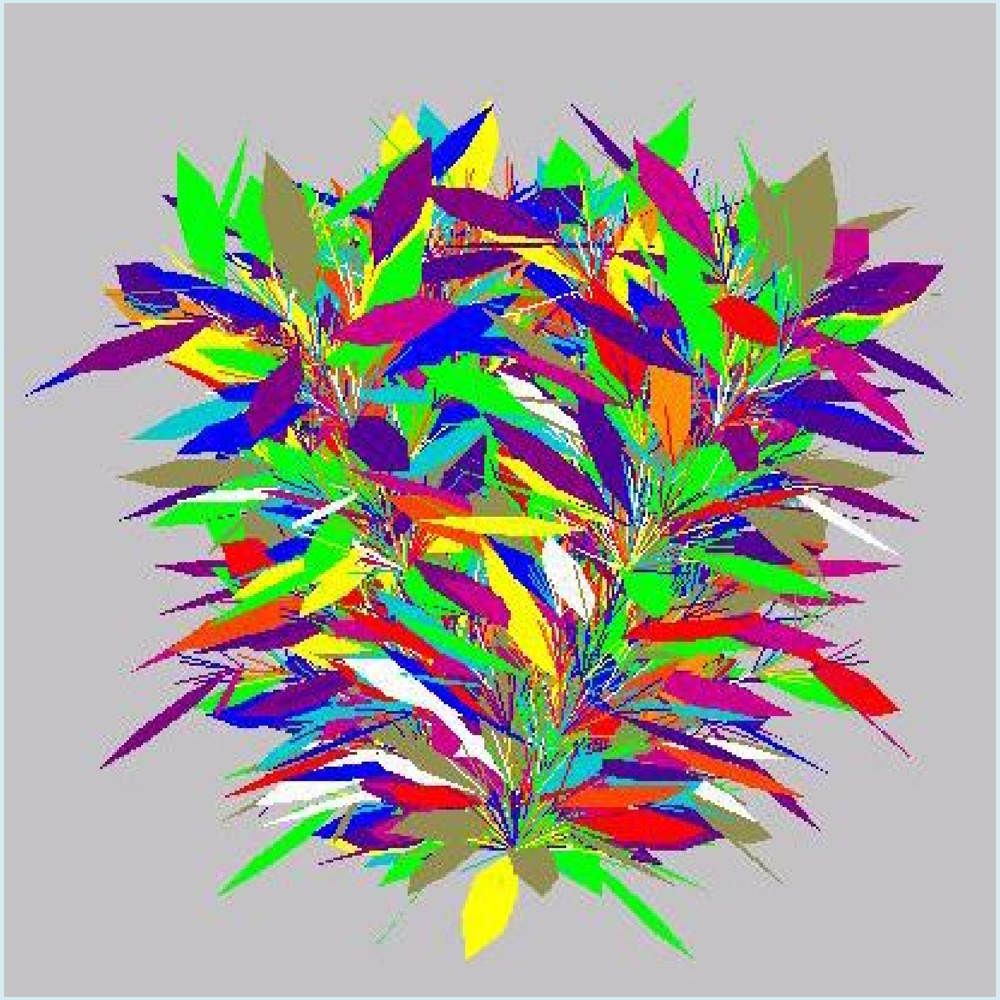
- F+ F+ F+ F
- F ↔ draw a line of unit length
- \+ ↔ turn anti clock wise through 90 degrees



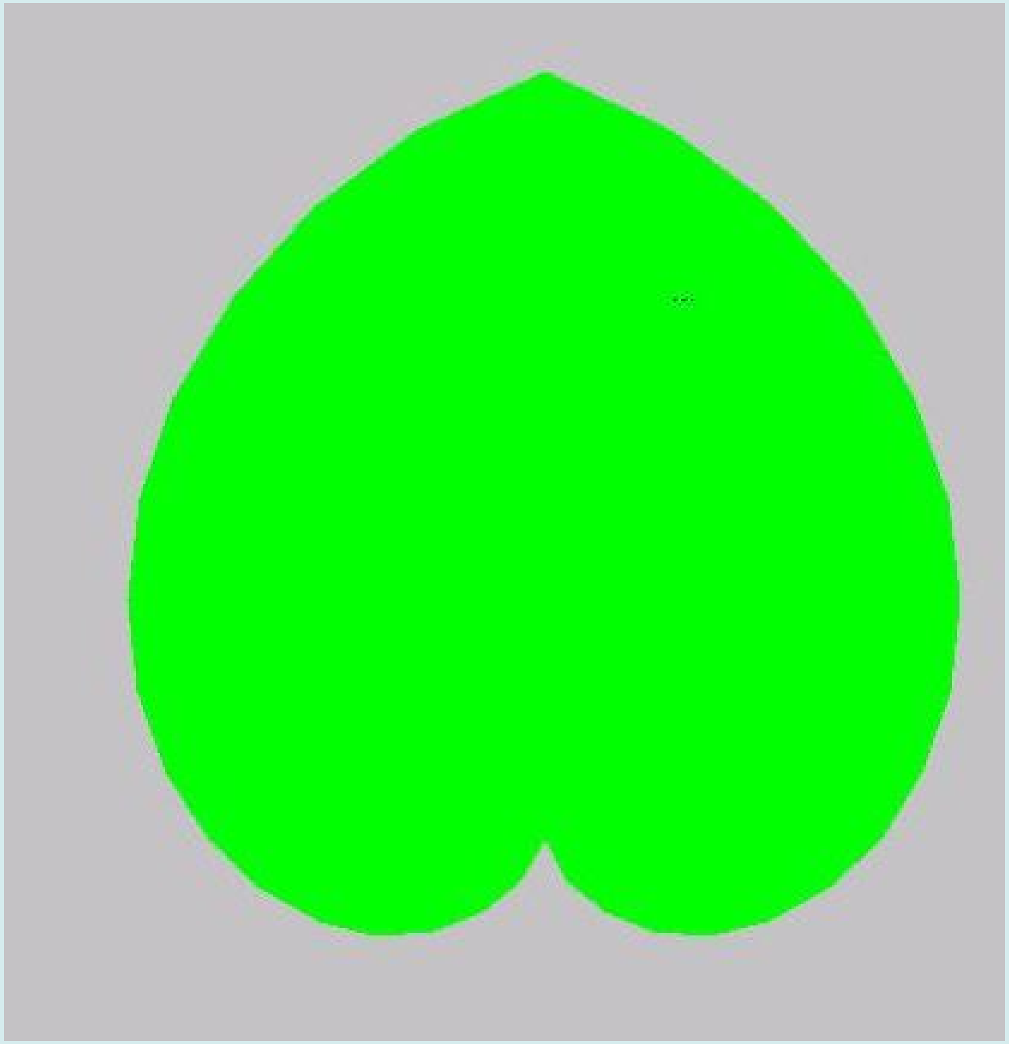
Kolam Patterns



L System and Computer Imagery Contd…



L System and Computer Imagery Contd…



L System and Computer Imagery Contd…