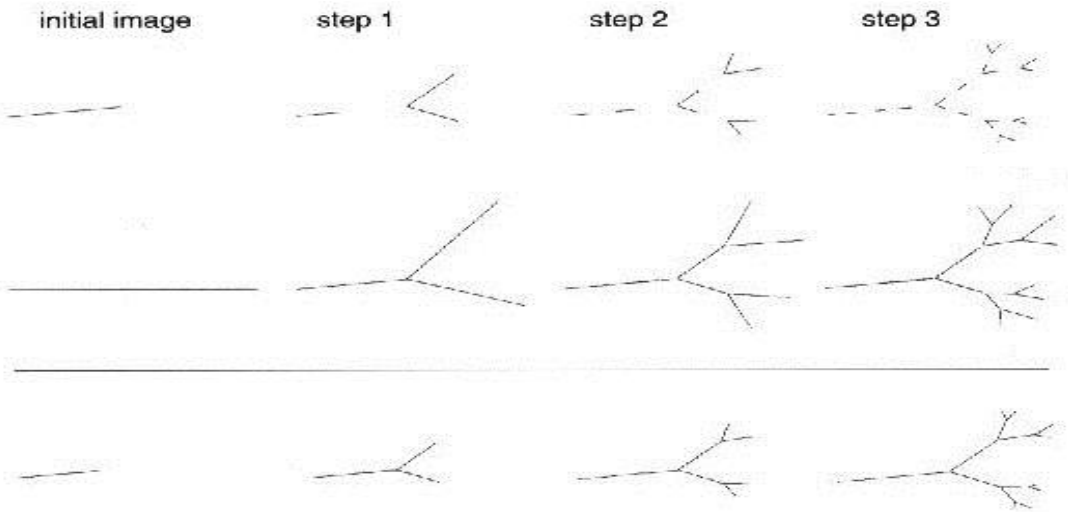
Fractals

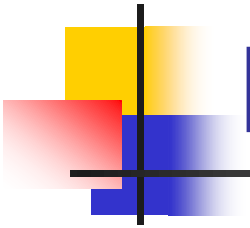
Part 7 : L - systems

Martin Samuelčík
Department of Applied Informatics

How to model growth?

- We want to model growth and shape changes in time
- We have just result
- Hard when using IFS



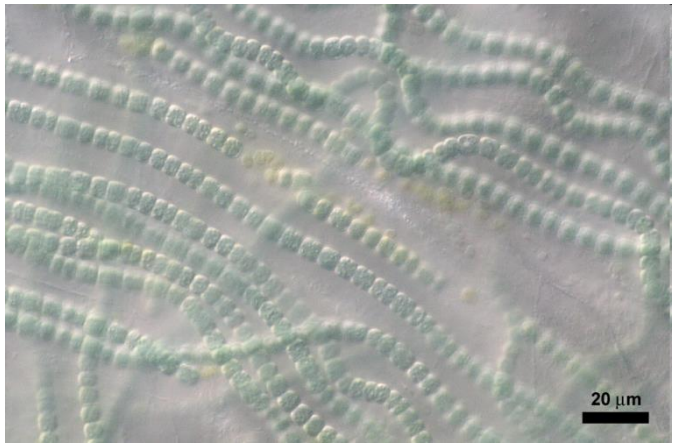


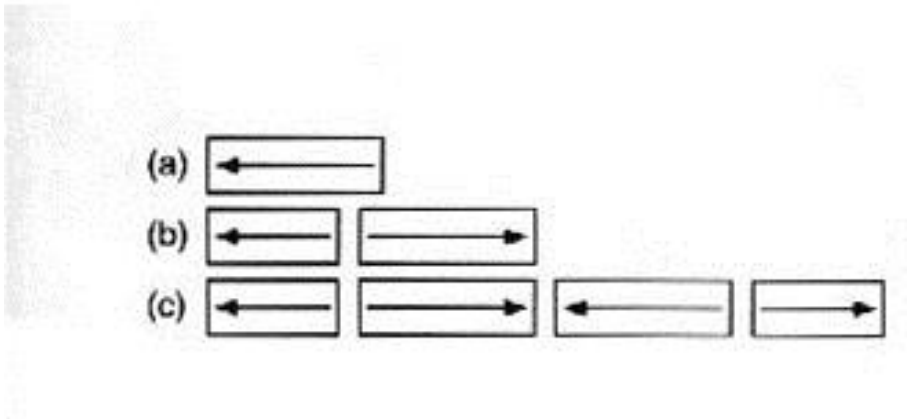
Development in time

- Biologist Aristid Lindenmayer
- Invented formal description for plants growth
- Based on biology observations
- Form <-> Growth
- Suitable for computer implementation

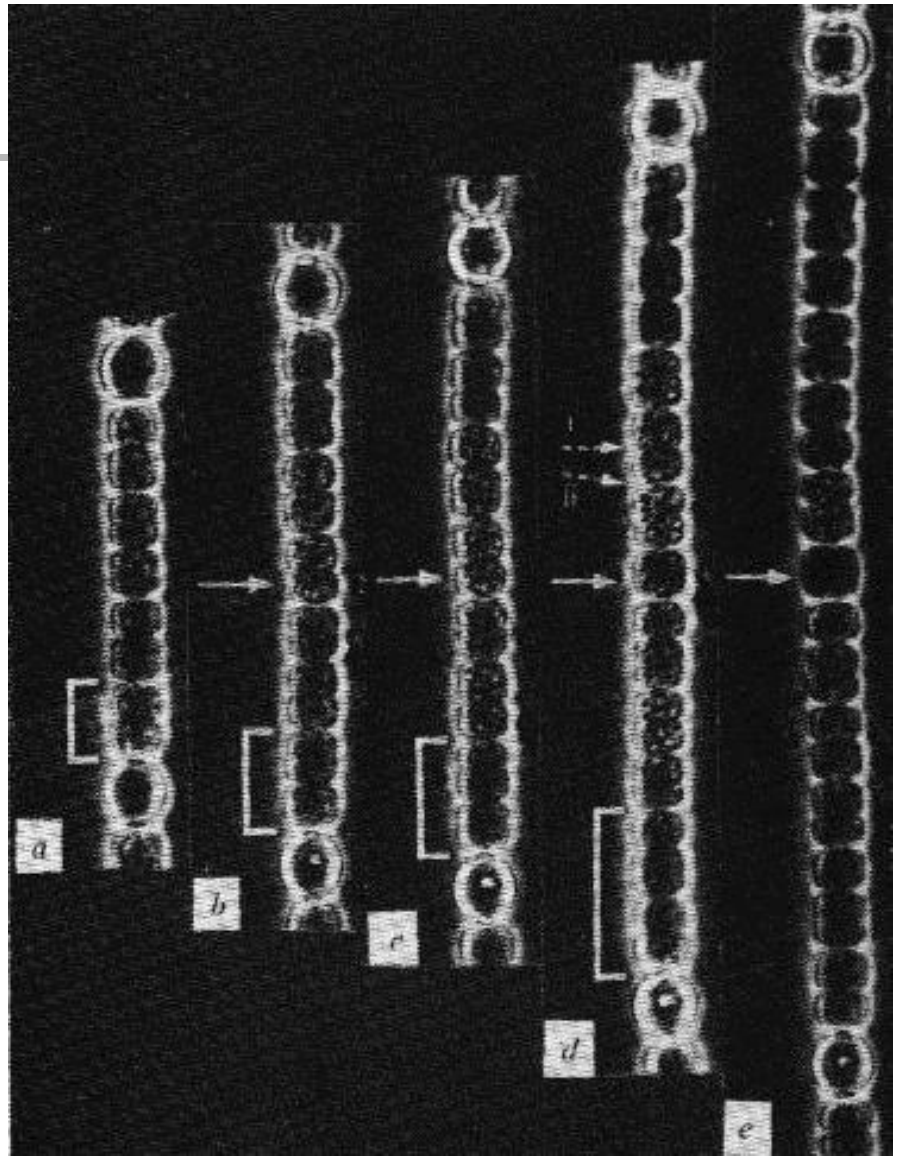
Example of growth

- Anabaena catenula
- 2 types of cells
- Cells are divided differently if they are left or right daughters
- Cells are divided into small and big ones





Example



Using Algebra

- A is small cell, B is big cell
- Arrow means left or right daughter
- Applying rules we get new iterations and system is growing
- One set of rules:

$$\overleftarrow{A} \rightarrow \overleftarrow{A}\overrightarrow{B}$$

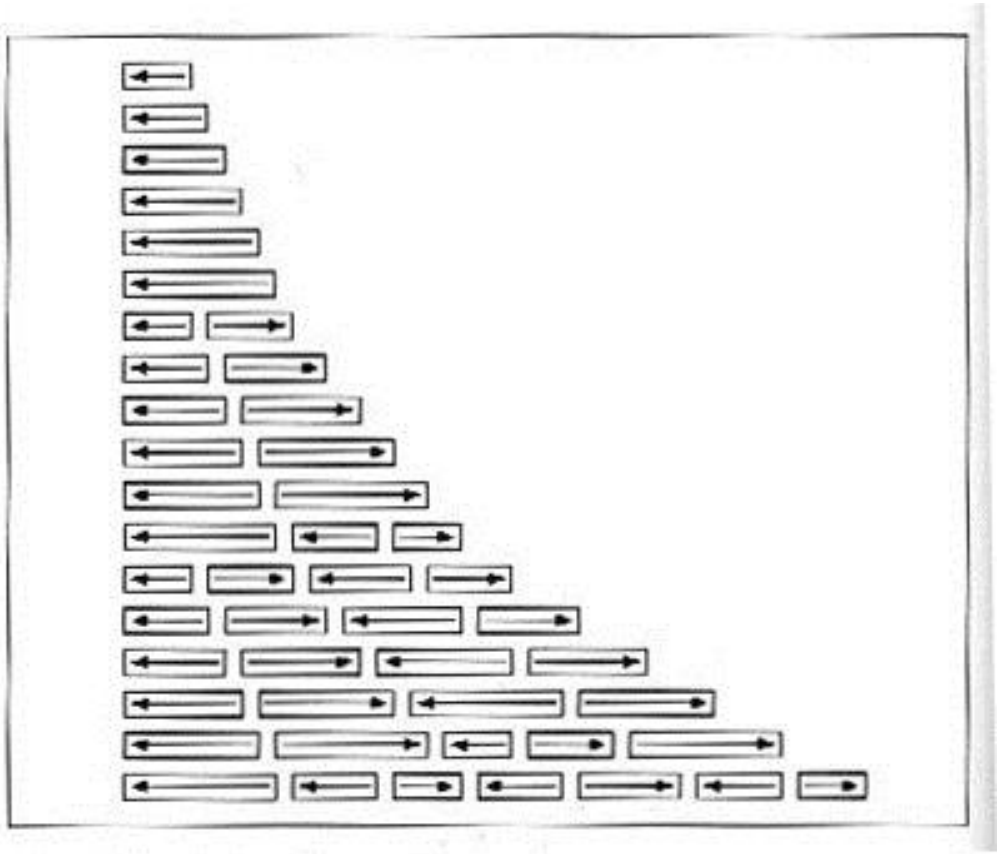
$$\overrightarrow{A} \rightarrow \overleftarrow{B}\overrightarrow{A}$$

$$\overleftarrow{B} \rightarrow \overleftarrow{A}\overrightarrow{B}$$

$$\overrightarrow{B} \rightarrow \overleftarrow{B}\overrightarrow{A}$$

Extending

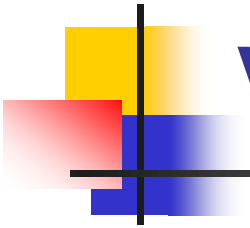
- Small cells must grow longer to perform dividing



L - systems

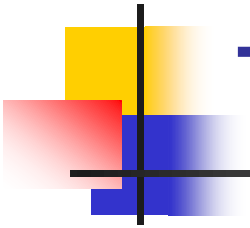
- Alphabet: $V = \{a_1, a_2, \ldots, a_n\}$
- Production map:

  $P: V \rightarrow V^* \; ; \; a \rightarrow P(a)$
- Axiom $a^{(0)}$ in V^*
- For each symbol in V there is one production rule P(a)
- V^* is set of all strings



Visualization

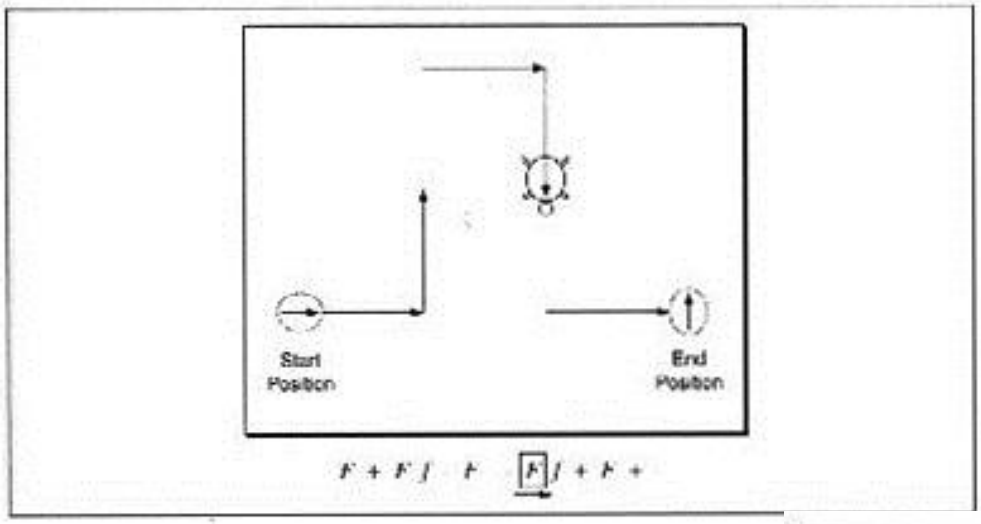
- Graphical interpretation of strings
- Is not predetermined in any way
- Based on analyze and observation of object in nature
- Can be in 2D, 3D
- Using many primitives for part of strings

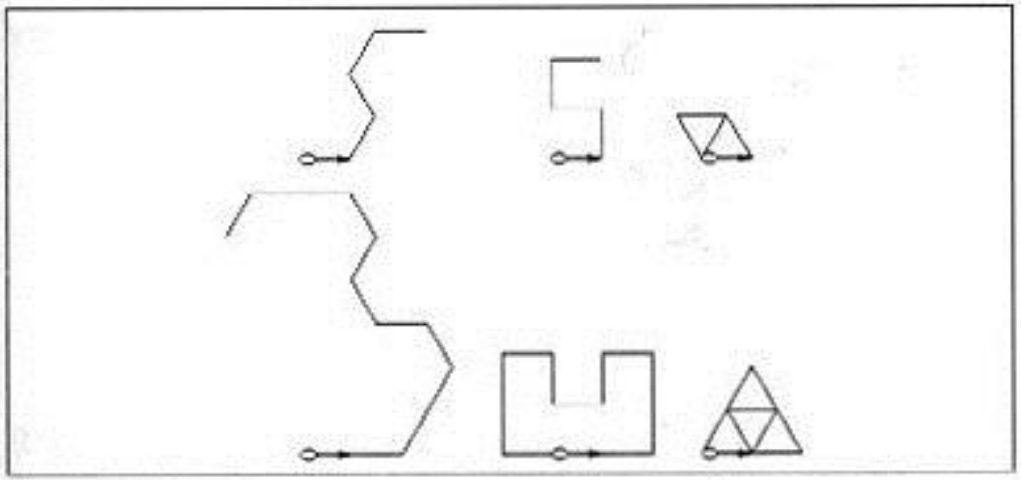


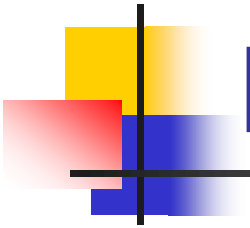
Turtle graphics

- One of graphical interpretation of L-systems
- Turtle is drawing line based on basic commands
- F - move forward drawing line
- f - just move forward
- +,- turn left (right) by the angle

Turtle graphics 2

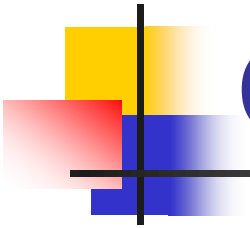






More symbols

- Composite movements
- L = + F – F – F +
- R = - F + F +F –
- S = FF + F + FF – F – FF
- Z = FF – F – FF + F + FF



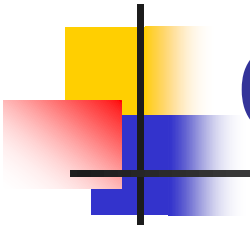
Growing Classical Fractals

- L-system: Koch curve
- Axiom: F
- Production rules:

  F -> F + F - - F + F

  + -> + ; - -> -
- Parameter: angle = 60°
- Possible scaling

Cantor set

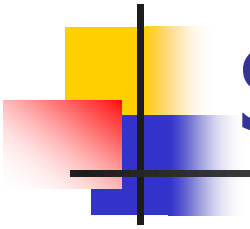


- L-system: Cantor set
- Axiom: F
- Production rules:

  F -> F f F

  f -> f f f

  + -> + ; - -> -
- Parameter: angle = 0°-360°

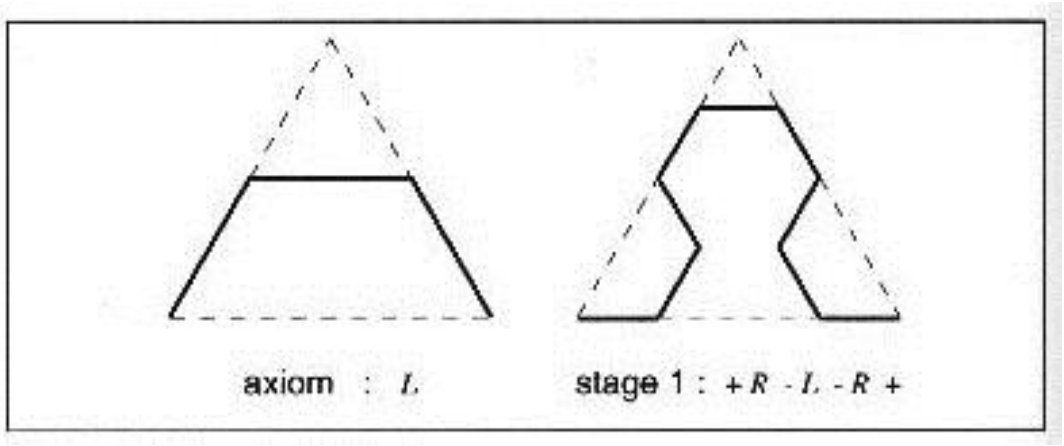


Sierpinski arrowhead

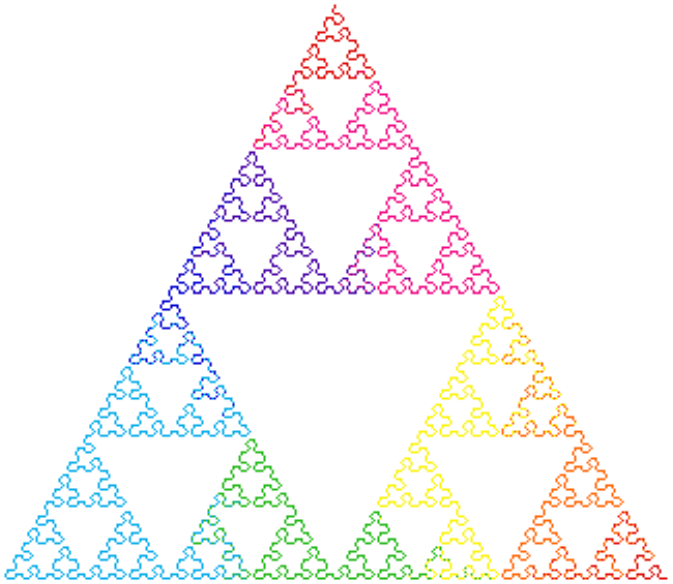
- L-system: Sierpinski arrowhead
- Axiom: L
- Production rules:

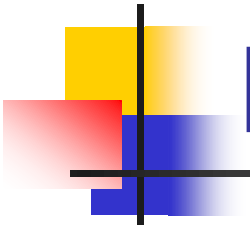
  L -> + R - L - R +
  R -> - L + R + L -
  + -> + ; - -> -
- Parameter: angle = 60°
- L = +F-F-F+, R = -F+F+F-

Sierpinski arrowhead 2



YF, X -> YF+XF+Y, Y -> XF-YF-X



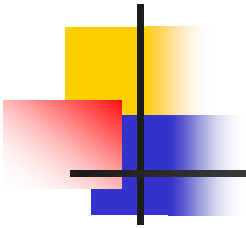


Peano curve

- L-system: Peano curve
- Axiom: F
- Production rules:

  F -> FF + F + F + FF + F + F - F

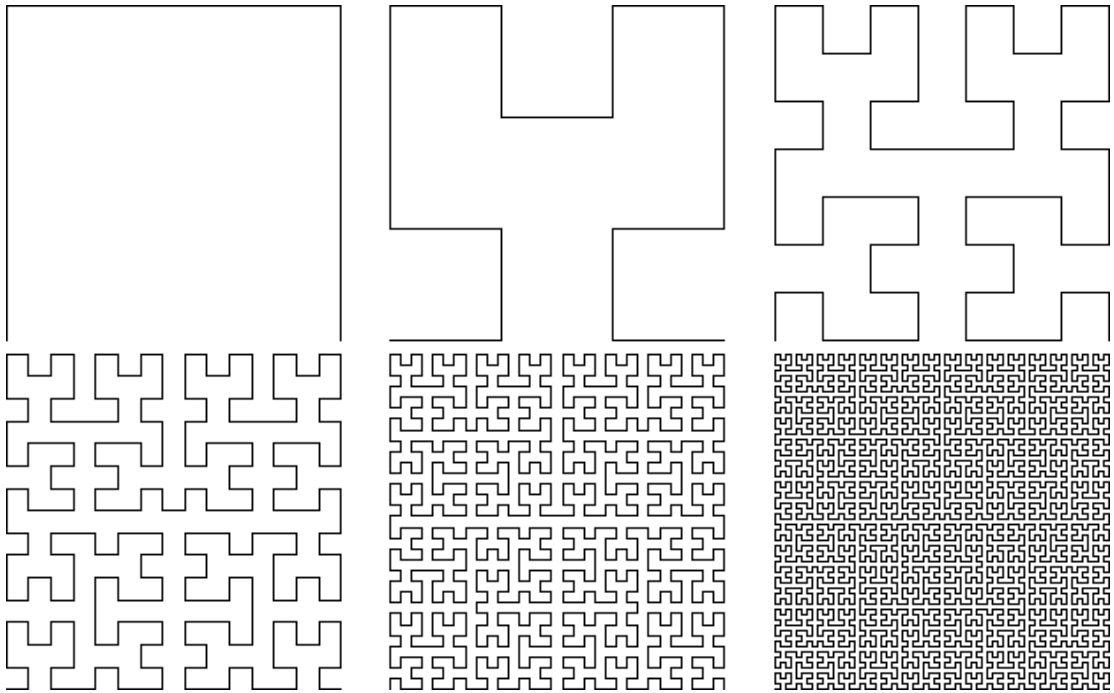
  + -> + ; - -> -
- Parameter: angle = 90°

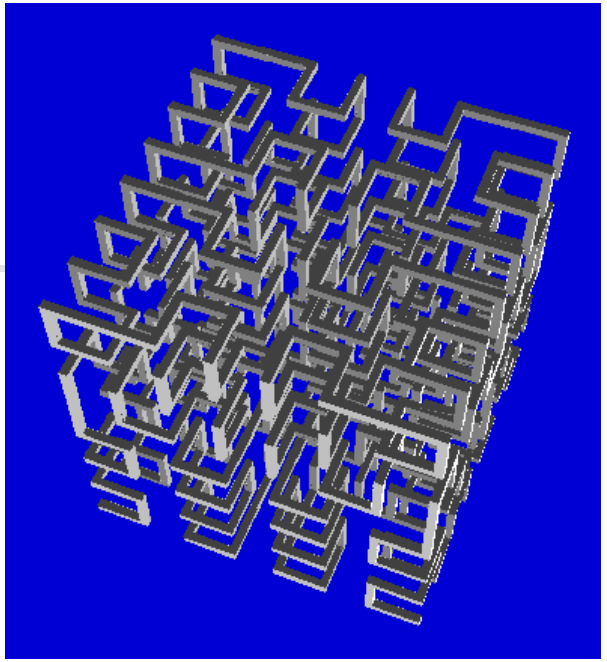


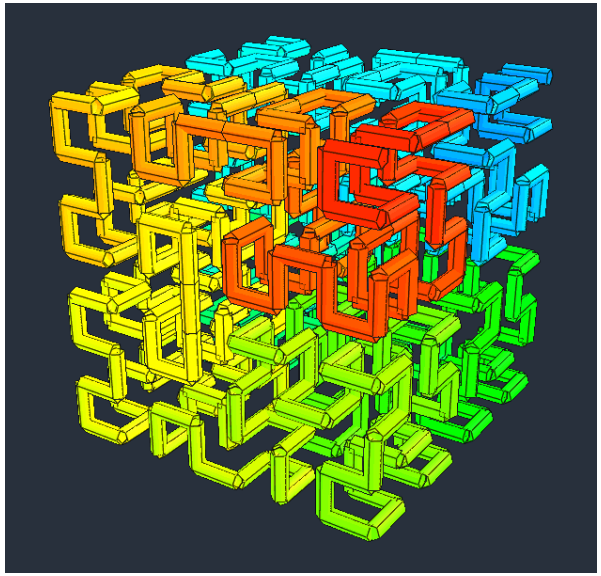
Hilbert curve

- Angle 90, Axiom X, X = –YF+XFX+FY–, Y = +XF–YFY–FX+
- Extension to 3D: up, right & dir vector, more operators, Axiom A
  - +,- = rotate around up
  - &,^ = rotate around right
  - | = rotate around y by 180 degrees
- A -> B-F+CFC+F-D&F^D-F+&&CFC+F+B
- B -> A&F^CFB^F^D^^-F-D^|F^B|FC^F^A
- C -> |D^|F^B-F+C^F^A&&FA&F^C+F+B^F^D
- D -> |CFB-F+B|FA&F^A&&FB-F+B|FC

Hilbert curve 2







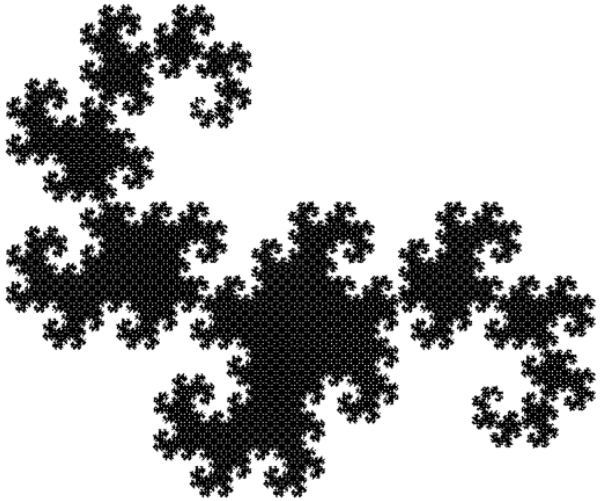
Dragon Curve

- L-system: Dragon curve
- Axiom: D
- Production rules:

  D -> - D + + E

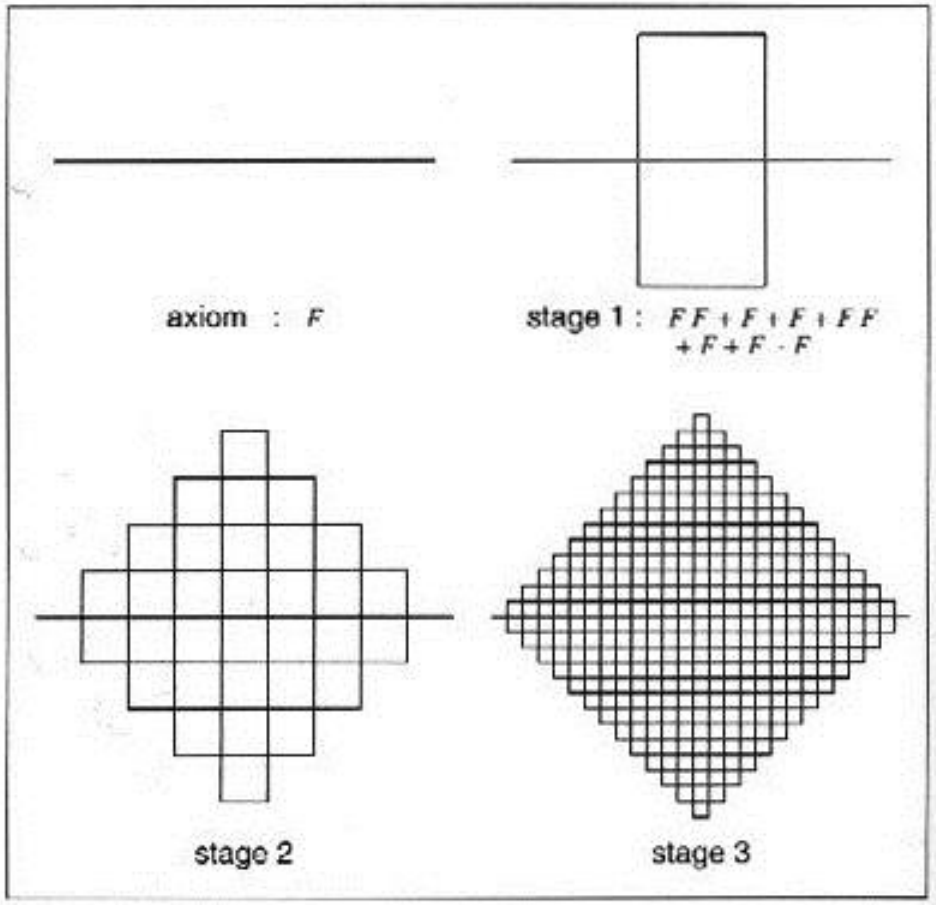
  E -> D - - E +

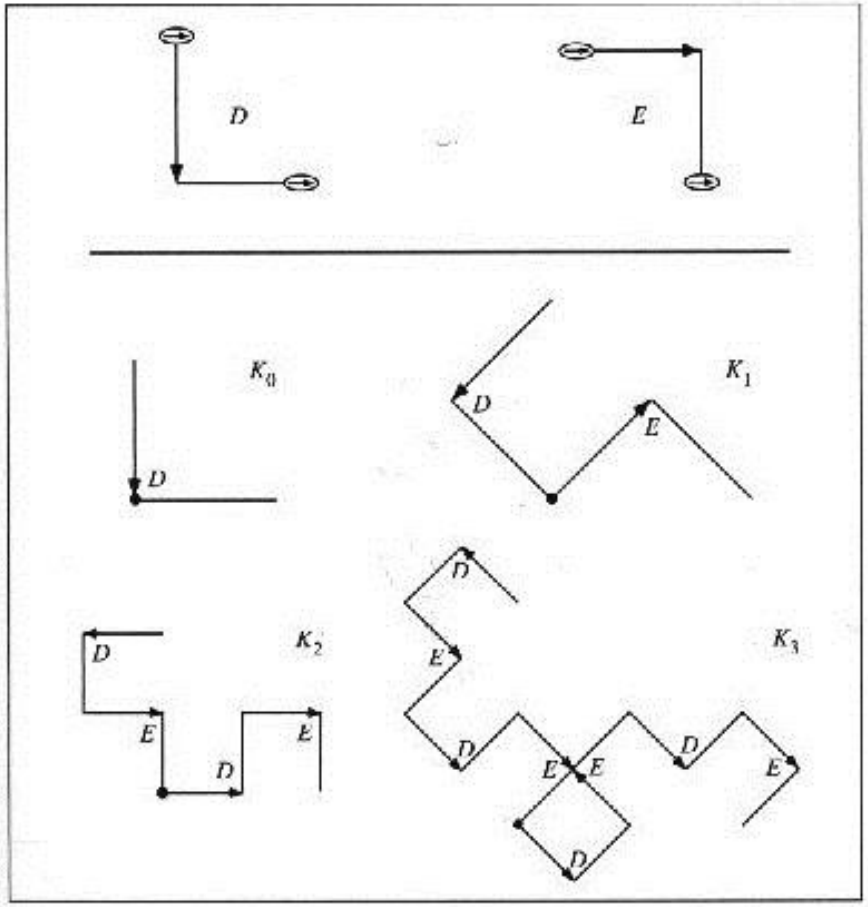
  + -> + ; - -> -

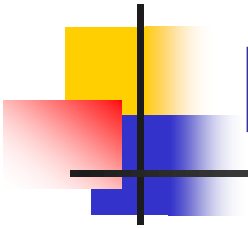


- Parameter: angle = 45°
- D= - - F + + F; E = F - - F + +

Examples

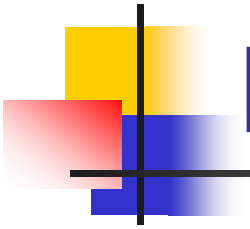






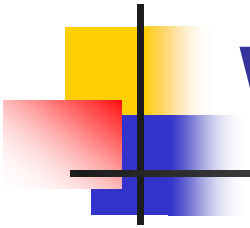
Fractal dimension

- For curves and turtle graphics interpretation
- log(N)/log(d)
- N = effective distance from start to end point after i-th step
- d = distance from start to end point, in turtle steps



Brushes & trees

- Branching structure cannot be described with linear sequential list
- 2 new symbols: [and]
- Left bracket saves current turtle state on stack, right bracket pops state from stack
- In bracket are branches



Weed Plant

- L-system: Weed plant
- Axiom: F
- Production rules:

  F -> F[+F]F[-F]F

  + -> + ; - -> -
- Parameter: angle = 25.7°

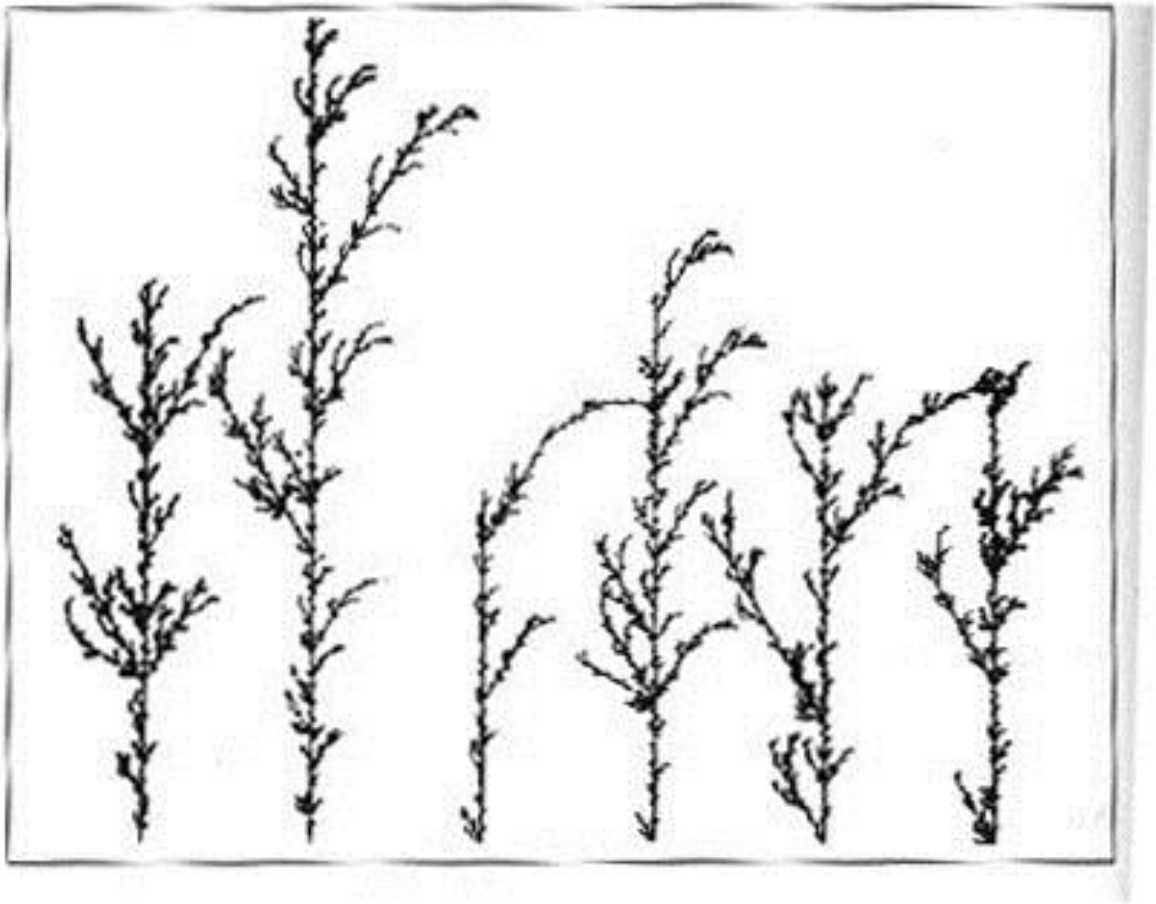
Random Weed Plant

- L-system: Random Weed
- Axiom: F
- Production rules:

  F -> F[+F]F[-F]F (prob. 1/3)
  F -> F[+F]F (prob. 1/3)
  F -> F[-F]F (prob. 1/3)
  + -> + ; - -> -

- Parameter: angle = 25.7°

Weeds



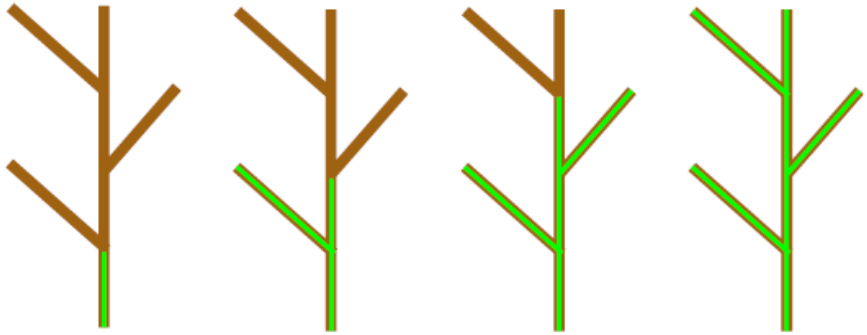
Stochastic L-systems

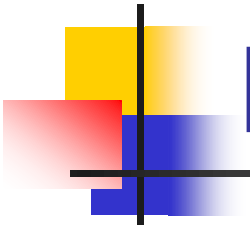
More than one production rule for one symbol, rule is picked by probabilities

Context sensitive

- The selection of a production for a symbol depends on the adjacent symbols in the current string
- Simulating propagation of signal

```
xy < a > z  → ab
x  < a > yz → aa
xyazabyxayzab ⇒ xyabzabyxaayzab
```

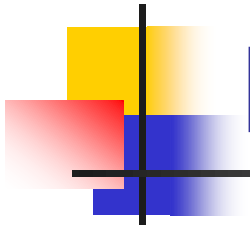




Parametric L-Systems

- A parametric L-system (pL-system) is defined as ordered quadruplet <A,Σ,ω,P>, where:
  - A is an alphabet of symbols
  - Σ is a finite set of parameters
  - ω in (A × R*) is the axiom
  - P from ((A × Σ*) : C(Σ) → (A × E(Σ))* is the set of productions
- C(Σ) denotes a logical and E(Σ) an arithmetic expression with parameters from Σ

Parametric L-Systems



- ω: B(2)A(4,4)
- p1: A(x,y): y<=3 → A(x*2,x+y)
- p2: A(x,y): y>3 → B(x)A(x/y,0)
- p3: B(x) : x<1 → C
- p4: B(x) : x>=1 → B(x-1)

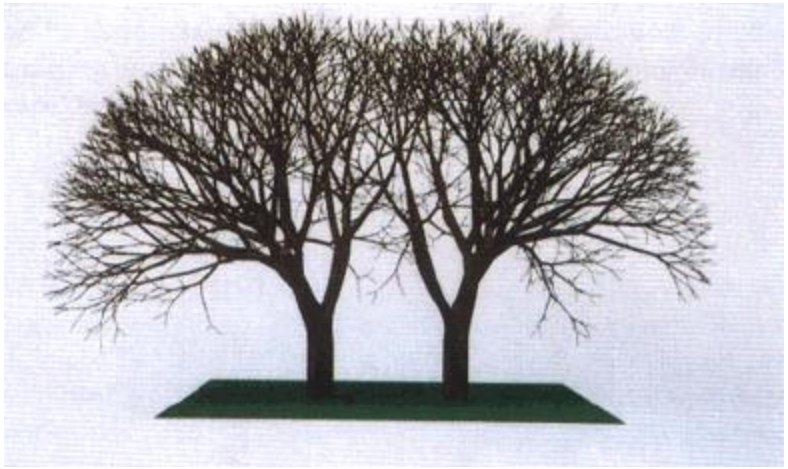
- Result:
  B(2)A(4,4)
  B(1)B(4)A(1,0)
  B(0)B(3)A(2,1)
  C B(2)A(4,3)
  C B(1)A(8,7)
  C B(0)B(8)A(1.142,0)

Environmentally-Sensitive

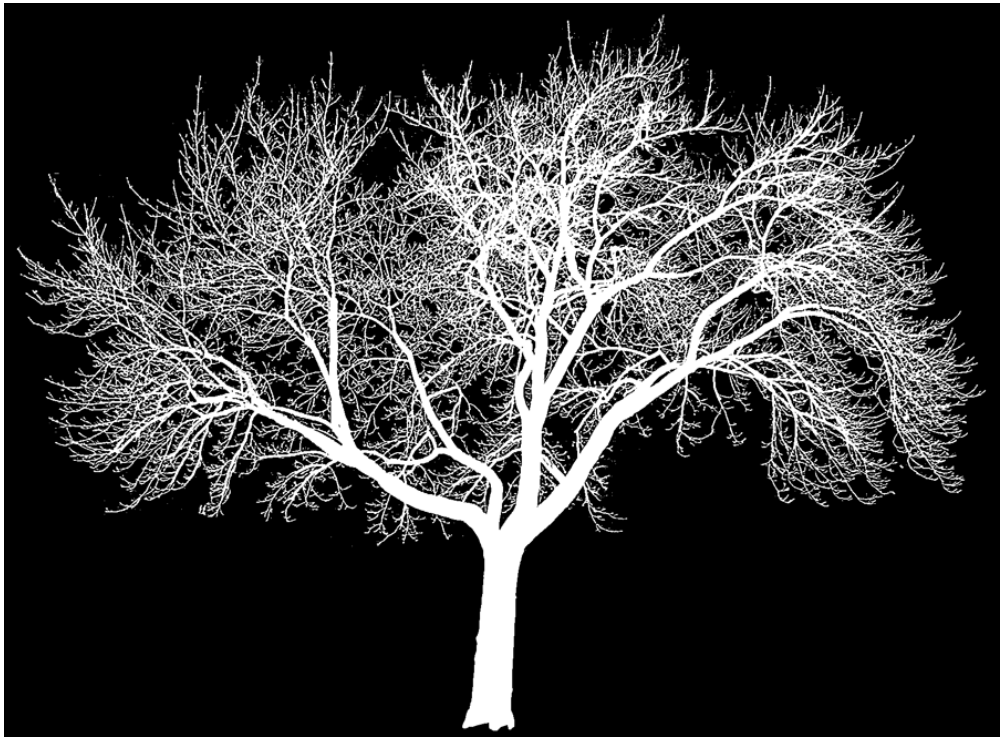
- Green's voxel space automata

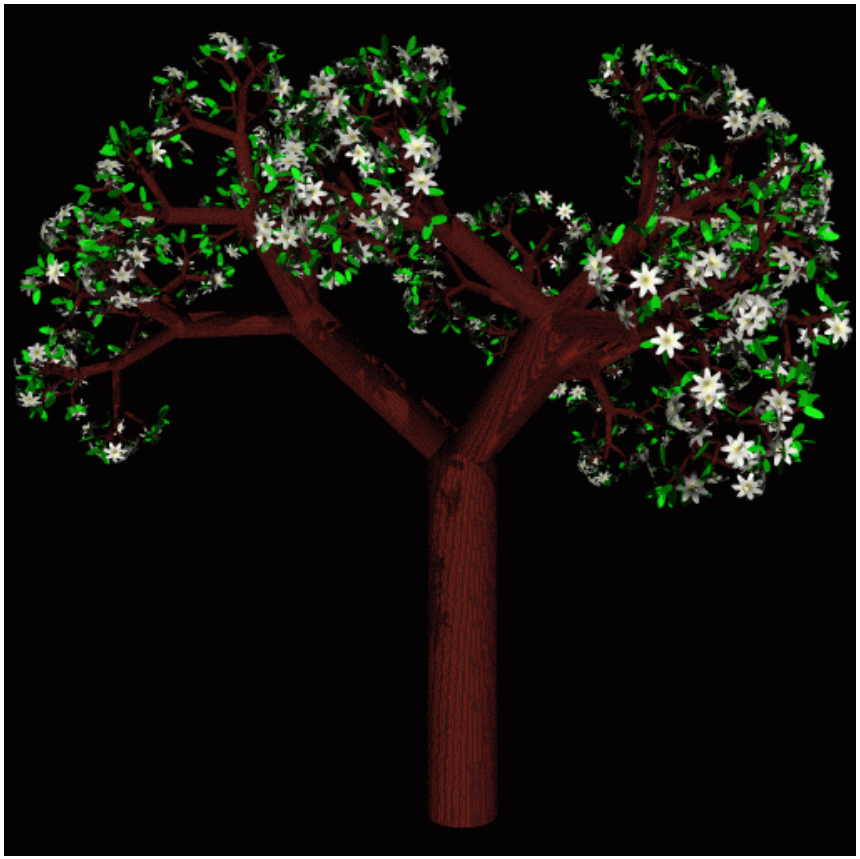
Open L-Systems

- Extension to general query modules and auxilary data structures
- Check for self intersection and avoid them
- Check for biological relevant conditions (soil composition, water, light, wind, ...)

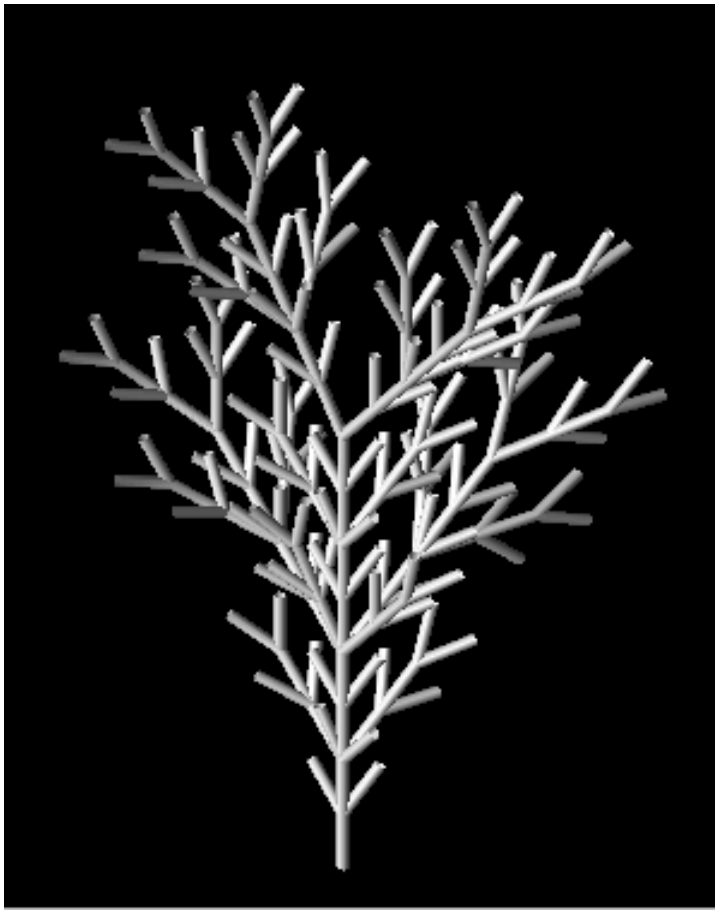


More examples

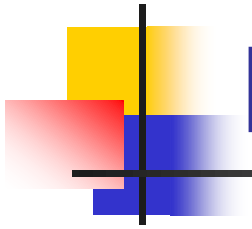




More examples 2



More more examples



End

End of Part 7