Botanical Modeling

CS 418

Interactive Computer Graphics

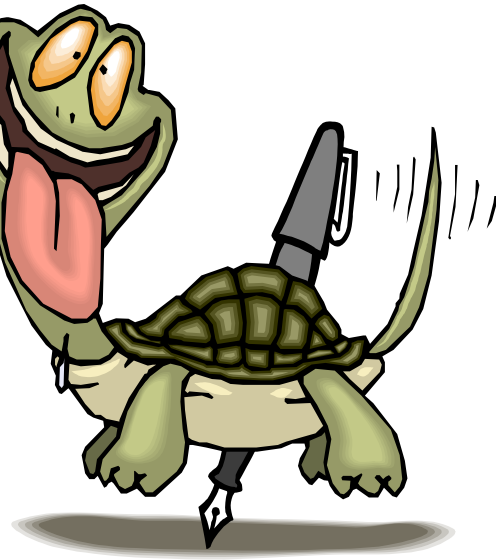
John C. Hart

Botanical Modeling

Deussen, Hanrahan, Lintermann, Mech, Pharr, Prusinkiewicz. Realistic modeling and rendering of plant ecosystems. SIGGRAPH 1998

Turtle Graphics

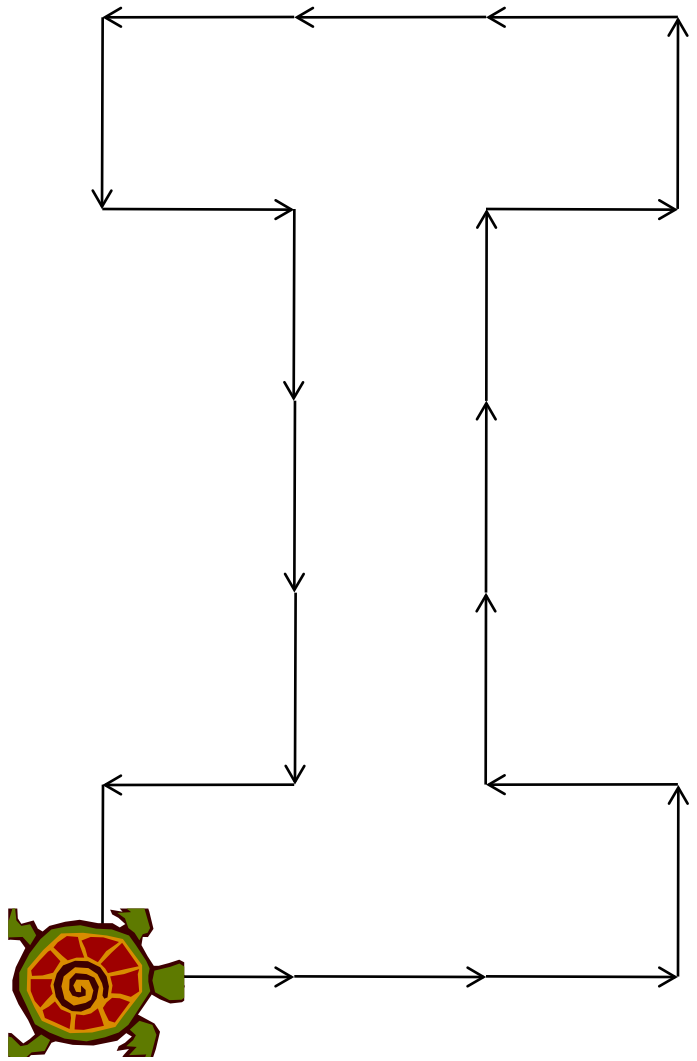
- Correspondence between symbols and graphics commands
- 2-D turtle commands:
  - **F** Draw forward one unit
  - **f** Move forward one unit
  - **+** Rotate turtle left
  - **–** Rotate turtle right
- Strings of symbols describe shapes
- Units (F,f) and angles (+,-) are global constants
- Turtle has state (like OpenGL state) consisting of position and orientation
- Each commands changes turtle's state



Turtle Graphics

F	Draw forward one unit
f	Move forward one unit
+	Rotate turtle left
–	Rotate turtle right

FFF+F+F-FFF-F+F+
FFF+F+F-FFF-F+F+



L-System Grammar

- Invented by Aristid Lindenmeyer
- L-System components
  - Set of symbols, e.g. {F,+,–}
    - Terminal – output, e.g. {+,–}
    - Non-terminal – replaced {F}
  - Production(s), e.g. F→F+F--F+F
  - Axiom(s), e.g. F
- Parallel graph grammar
  - Like a context free grammar
  - But all non-terminal symbols are replaced simultaneously

Symbols: {F,+,–}

Productions:

{F → F+F--F+F}

Axioms: F

F

F+F--F+F

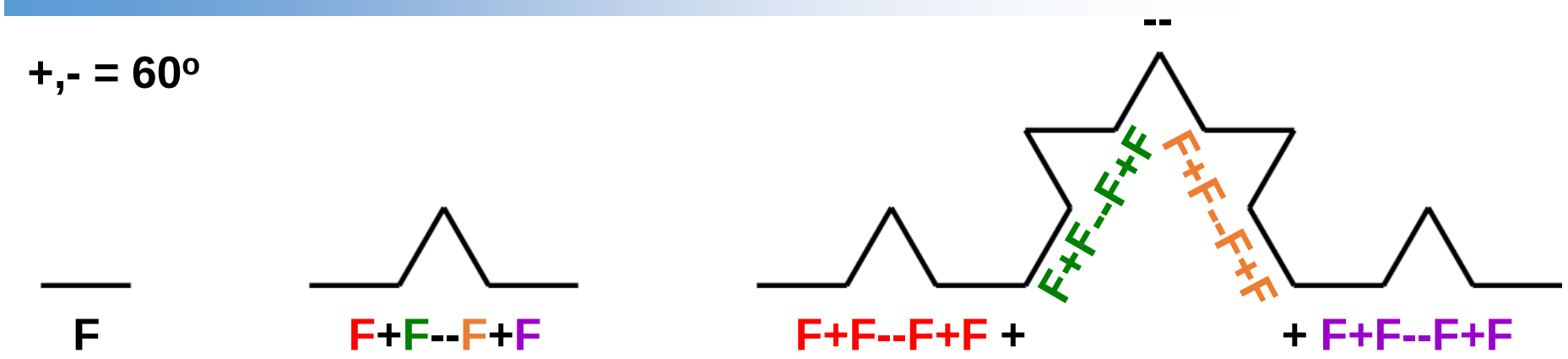
F+F--F+F +

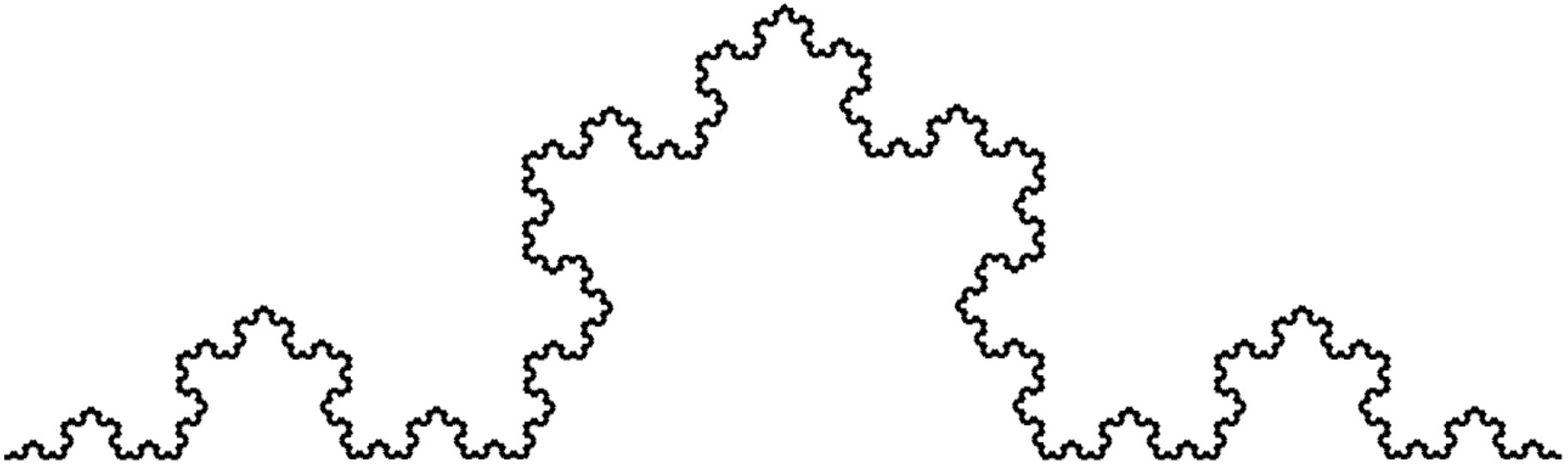
F+F--F+F --

F+F--F+F +

F+F--F+F

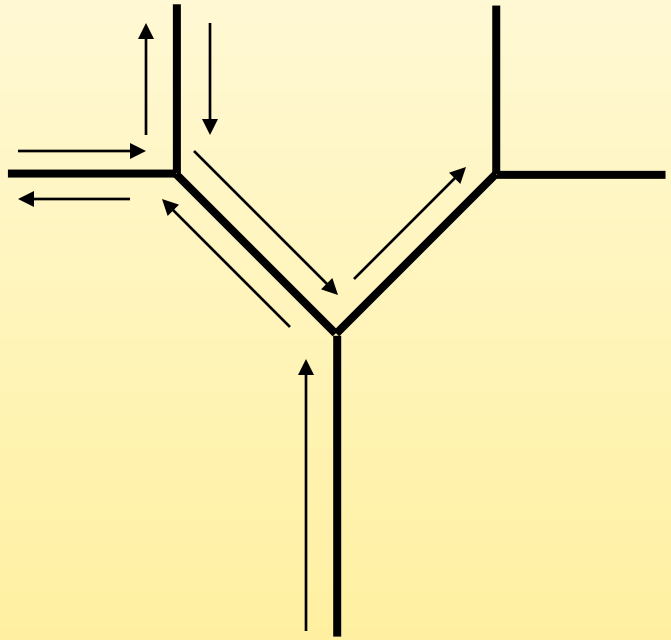
von Koch Snowflake Curve





Branching

- Can use F,+,- to make a large number of squiggly lines
- But want to model branching structures
- Don’t want to have to retrace our steps whenever we branch
- Turtle state stack
  - “[” – pushes state onto stack
  - “]” – pops state from stack
- Use “[” to start a branch and “]” when finished to return to its base



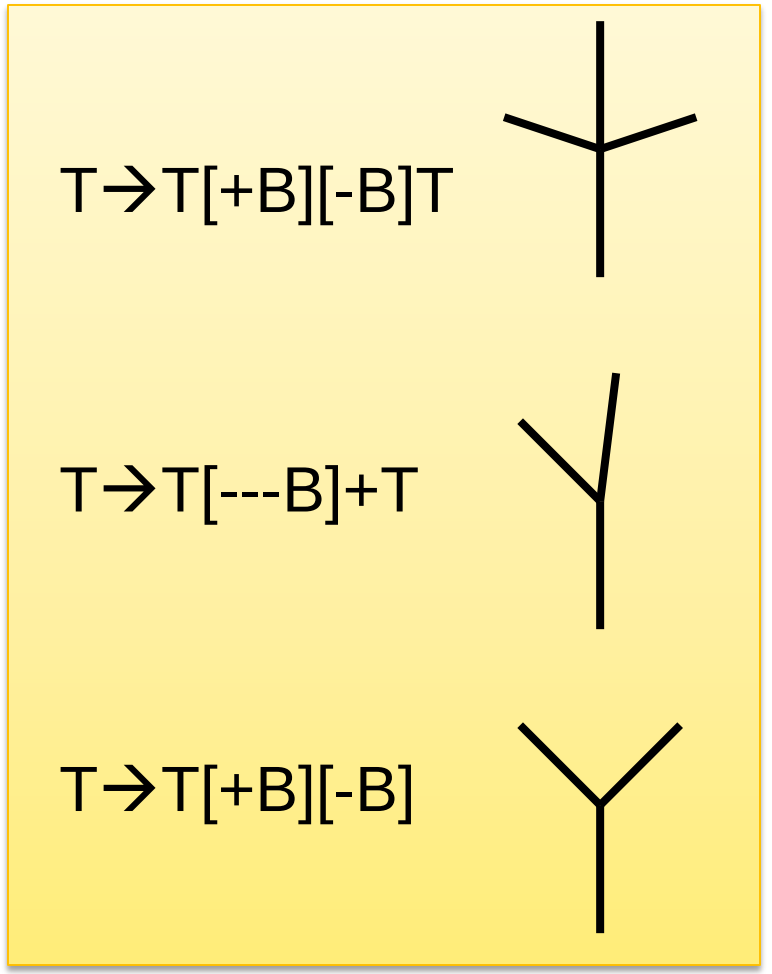
F+F+F----f+F----f+…

vs.

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

Branching Types

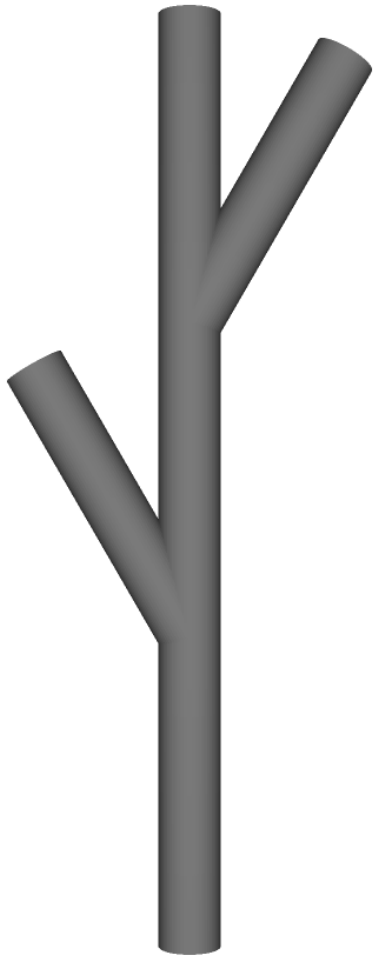
- Monopodial
  - Trunk extends undeviated to top
  - Branches extend perpendicularly
- Sympodial
  - Trunk deviates to top
  - Branches extend perpendicularly
- Binary
  - Trunk terminates at first branching point
  - Branches deviate uniformly

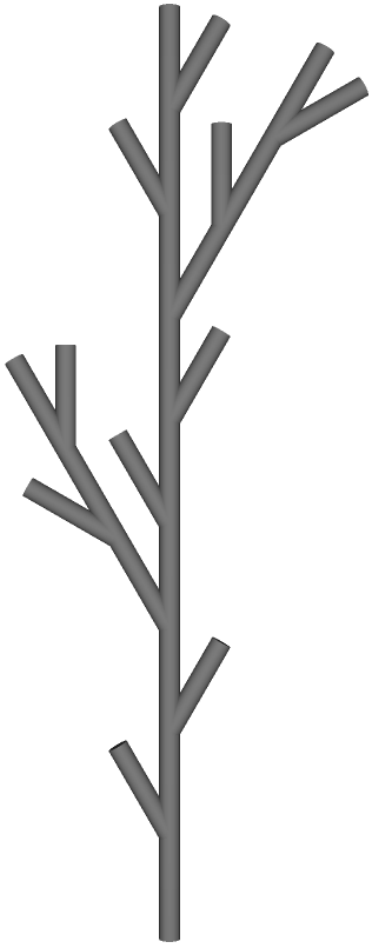


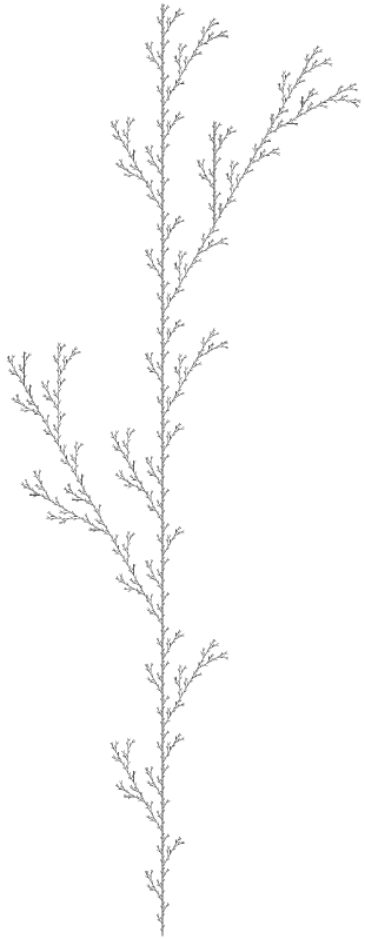
Σ branch areas = base area (da Vinci)

Bush

F→F[+F]F[-F]F







Stochastic L-Systems

- Conditional firing of productions

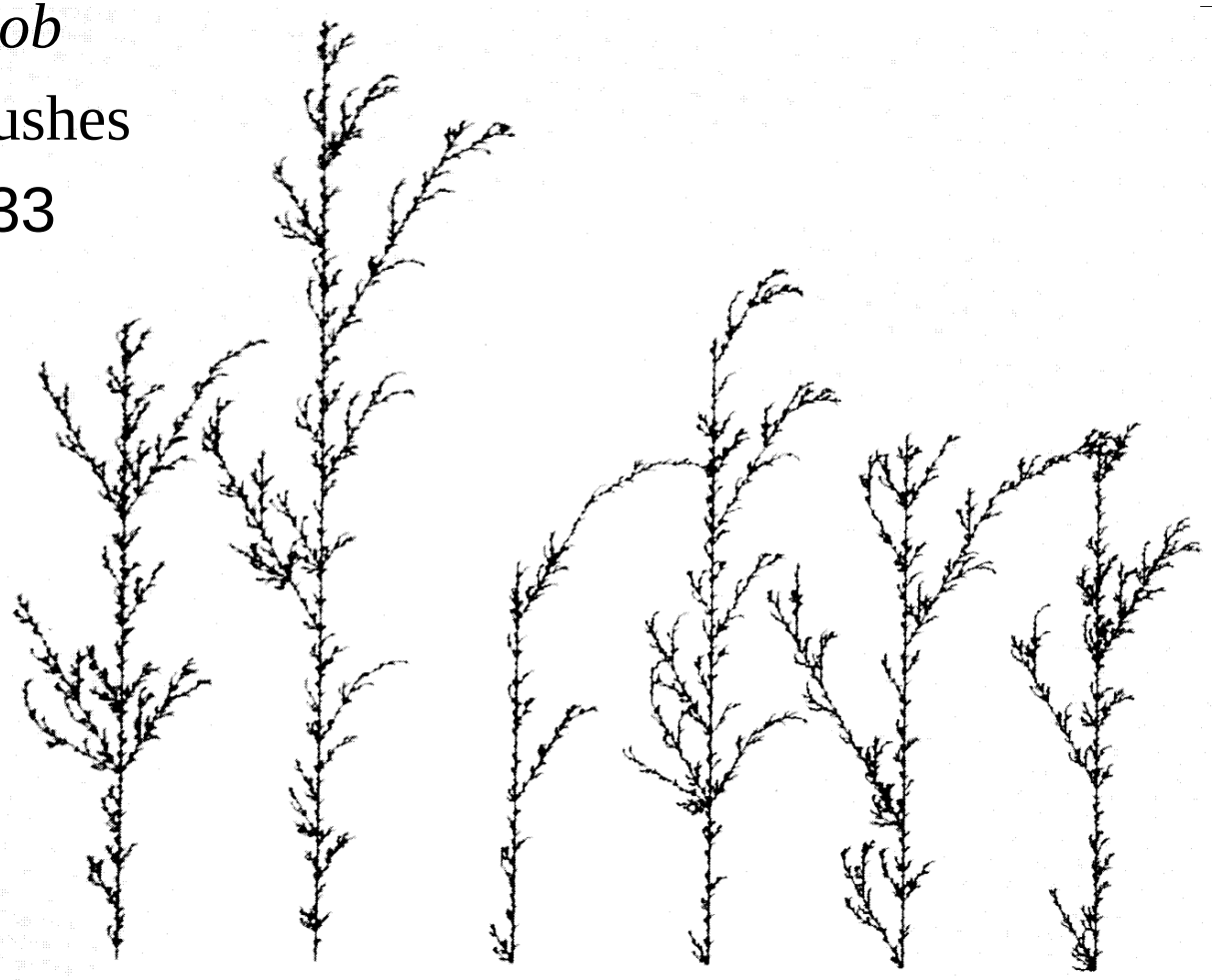
  NT → *result* : *prob*

- Example: Random Bushes

  F → F[+F]F[-F]F : .33

  F → F[+F]F : .33

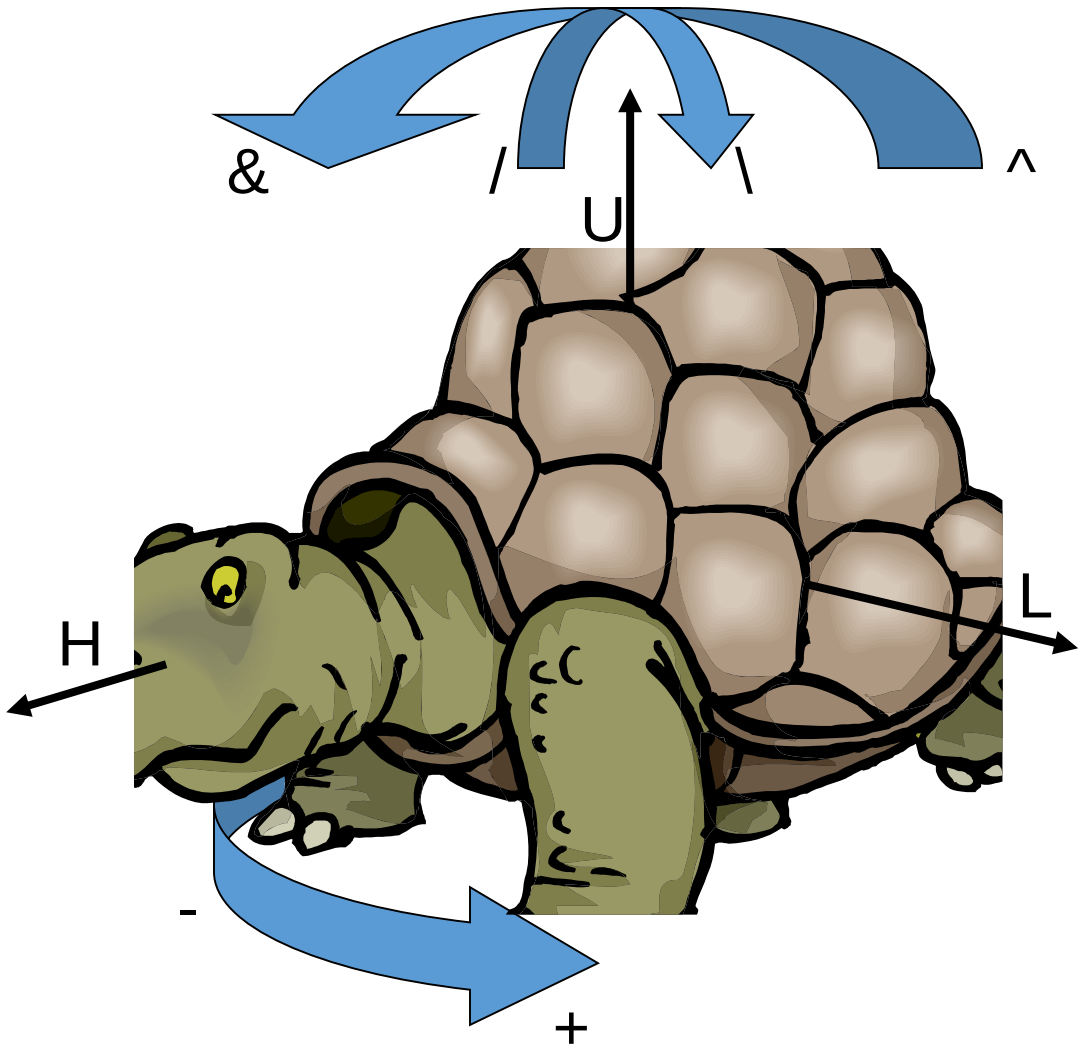
  F → F[-F]F : .34



- If not stochastic, then *deterministic*

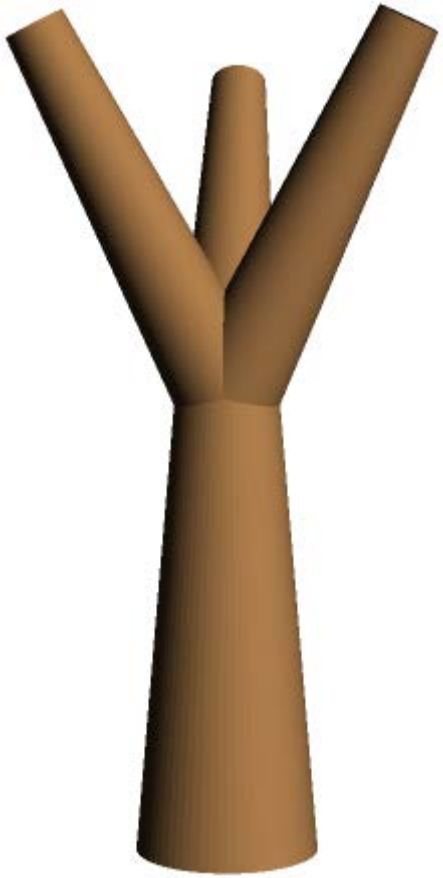
Parameterized 3-D Turtle Graphics

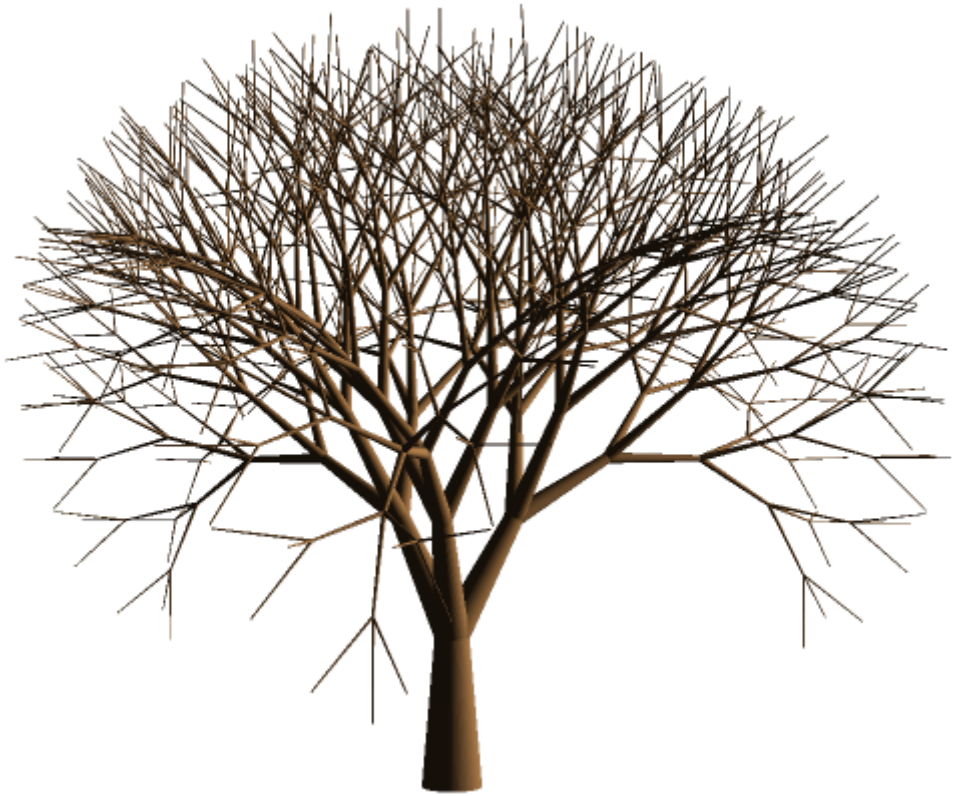
- "+(θ)" – yaw left
- "-(θ)" – yaw right
- "^(θ)" – pitch up
- "&(θ)" – pitch down
- "\(θ)" – roll left
- "/(θ)" – roll right
- "F(x)" – draw x units
- "f(x)" – move x units



Ternary Tree

F → F[&F][&/F][&\F]





Other Turtle Commands

- "!(*w*)" changes width of lines (cylinders)
  - Decreases width if no parameter
- "@O" draws a sphere
  - Useful at branch joints
- Generalized cylinder
  - Delimited by "@Gs" and "@Ge"
  - "@Gc" control point at current turtle position
- Polygons
  - Delimited by "{" and "}"
  - Edge "F", no edge "G", vertex "."

Need the "@" symbol because turtle quickly ran out of ASCII characters

- Bicubic Patches
  - "@PS(*n*)" patch indexed by integer *n*
  - "@PC(*n*,*i*,*j*)" current turtle position is (*i*,*j*)th control point of patch *n*
  - "@PD(*n*)" draws patch *n*
- Color & Texture
  - ";" moved to next color table entry
  - ";(*f*,*b*)" front color *f*, back color *b*
  - "@T(*n*)" use texture *n*
- "@L(string)" label at cur. pos.
- "@S(cmd)" run OS command

Parameterized Productions

- Non-terminals can be parameterized
- Productions can be conditional

  $NT(param) : cond \rightarrow result$

- Example: Long branches

  $B(n) : n > 0 \rightarrow F\ B(n-1)$

  $B(n) : n = 0 \rightarrow F$

- Example: Fibonacci

  $A(n) : n > 1 \rightarrow A(n\text{-}1)A(n\text{-}2)$

  $A(n) : n = 1 \rightarrow X$

  $A(n) : n = 0 \rightarrow \varepsilon$ (empty)

A(2) = A(1)A(0) = X

A(3) = A(2)A(1) = A(1)A(0)X

= XX

A(4) = A(3)A(2)

= A(2)A(1)A(1)A(0)

= A(1)A(0)XX

= XXX

Context Sensitivity

- 0L-System – context free
- 1L-System – one-sided context

  L < *NT* → *result*

  NT > *R* → *result*

  – "<" and ">" part of syntax

  – Only *NT* replaced, not *L* or *R*
- 2L-System – two-sided context

  L < *NT* > *R* → *result*
- Example: Signal Passing

  B < A → B

  B → A

BAAAAAAA

ABAAAAAA

AABAAAAA

AAABAAAA

AAAABAAA

AAAAABAA

AAAAAABA

AAAAAAAB

AAAAAAAA

Signals and Branches

- Which way do the signals propagate?
- Left context simulates *acropetal* propogation (base → leaves)
- Right context simulates basipetal propogation (leaves → base)
- IL-systems
  - Ignore bracketed substrings when pattern matching for context
  - Attempt all combinations of bracketed substrings
- May need to completely ignore other symbols (e.g. +,-)

BC < S > G[H]M → X

ABC[DE][SG[HI[JK]L]MNO]

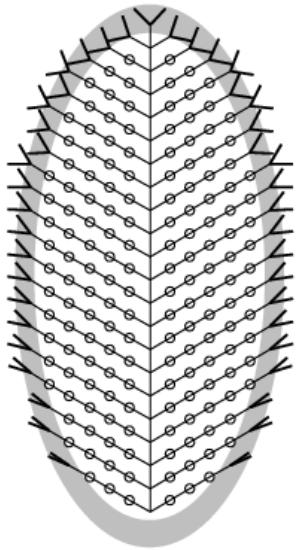
Topiary

- Query symbol “?”
  - “?P(*x*,*y*,*z*)” – position
  - “?H(*x*,*y*,*z*)” – heading
  - likewise turtle up and left vectors
- Variables (*x*,*y*,*z*) replaced by actual values at current turtle position
- Example: Ellipse

  A → [+B][-B]F?P(x,y)A

  B → F?P(x,y)@oB

  ?P(x,y) : $4x^2 + (y-10)^2 > 100$ →
  [+(2y)F][-(2y)F]%
- “%” cuts all symbols to end of branch



Open L-Systems

- Environmental query "?E(x_1,x_2,…)"
- Queries the environment
- Query at current turtle state
- Example: Light sensitivity
  - "?E(r)" how much light does a cluster of leaves of radius r at the current turtle position get?