Procedural Modeling and L-systems

Martin Čermák, Jiří Koutný and Alexander Meduna

Deparment of Information Systems
Faculty of Informatin Technology
Brno University of Technology, Faculty of Information Technology
Božetěchova 2, Brno 612 00, Czech Republic

BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

Advanced Topics of Theoretical Computer Science

FRVŠ MŠMT FR2581/2010/G1

How to get a Model?

We know what a model is, but how/where to get it?

Process of Modeling

- Reality is given (known fact)
- Observation of reality (to see the reality)
- Acquiring knowledge (to learn about the reality)
- Selection of essential knowledge (which features are important)
- Making a model (making the reprezentation of knowledge)

Reality $\rightarrow$ *Knowledge* $\rightarrow$ *Model*

How to get 3D Computer Model of Object?

Three basic principles

- Image based modeling
- Interactive modeling
- Procedural modeling

Image based modeling

- Means to get 3D image of object.
- Rough model of object's geometry.

How to?

- Scanning by 3D scanner.
- Reconstruction based on several images from photocamera.

Disadvantages

- Size of the reality
  - How to scan mountain, skyscraper, cloud, etc.?
- Segmentation of the reality
  - How to reconstruct surface of head under the hairs?

Example: Scanning by 3D Scanner

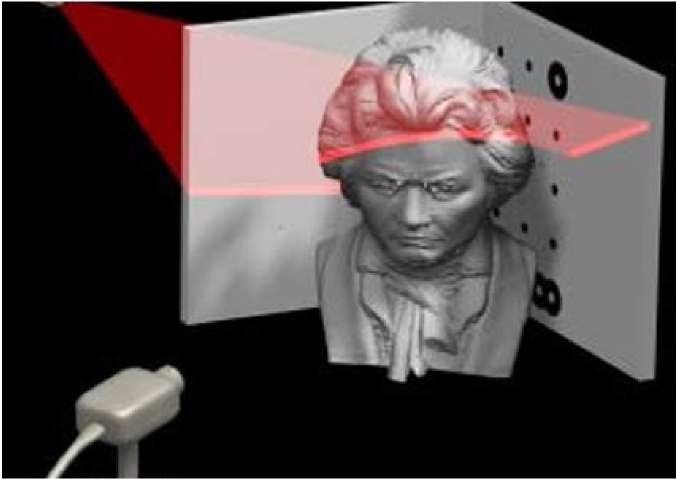


Figure: gadgetell.com

Example: Scanning by 3D Scanner

Figure: flickr.com

Example: Reconstruciton from Photograps

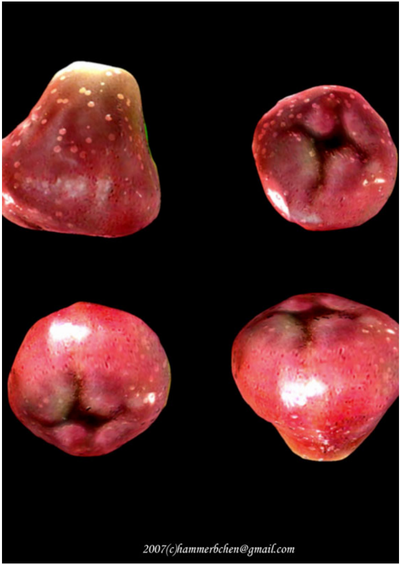


Figure: imageshack.us

Figure: immersiveeducation.org

- Means to create a model in modeling software.
- Semantically completely accurate model.

How to?

- CAD/CAM systems.
- 3D studio, Blender, etc.

Disadvantages

- Time consuming process.
    - To create detail model of object takes many hours, days,...?
- Talent and experiences needed.

Example: Interactive Modeling

Figure: ntlworld.com

Example: Interactive Modeling

Figure: imageshack.us

Example: Interactive Modeling

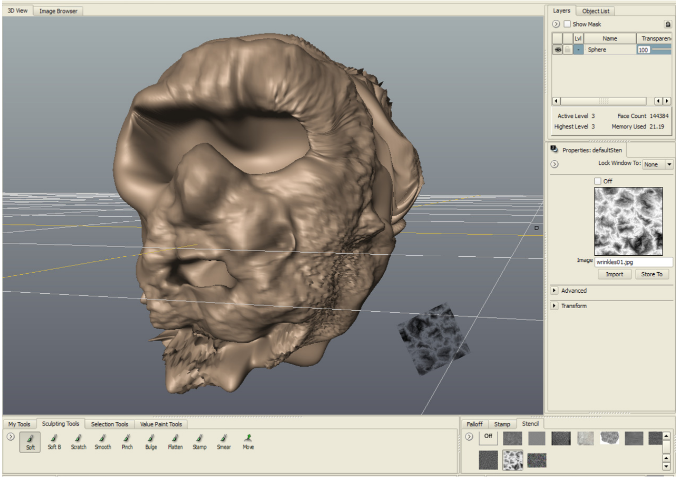


Figure: wikimedia.org

Example: Interactive Modeling

Figure: unrealphd.com

Example: Interactive Modeling

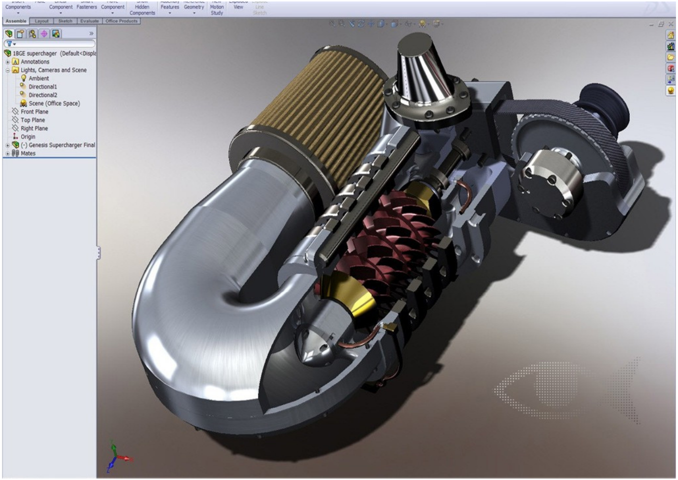


Figure: 3dconn.com

- Means to describe model by *algorithm*.

How to?

- Generating surfaces from curves (in CAD systems).
- Automatical generating of the objects similar to the objects in the nature.

What is it?

Procedural techniques are the algorithms that determine certain characteristics of a computer model

Basic types:

- Algorithm based on grammars.
  - Lindenmayer systems, used for generating plants.
- Fractal geometry.
  - Used for generating mountains, landscapes, stones, etc.
- Particle systems.
  - Used for generating explosions, flocks of birds, falling balls, fire, etc..

Example: Lindenmayer System

Some determinstic 3D branching plants.

Figure: antipode.ca

Example: Lindenmayer System

Figure: sites.google.com

Example: Fractal Geometry

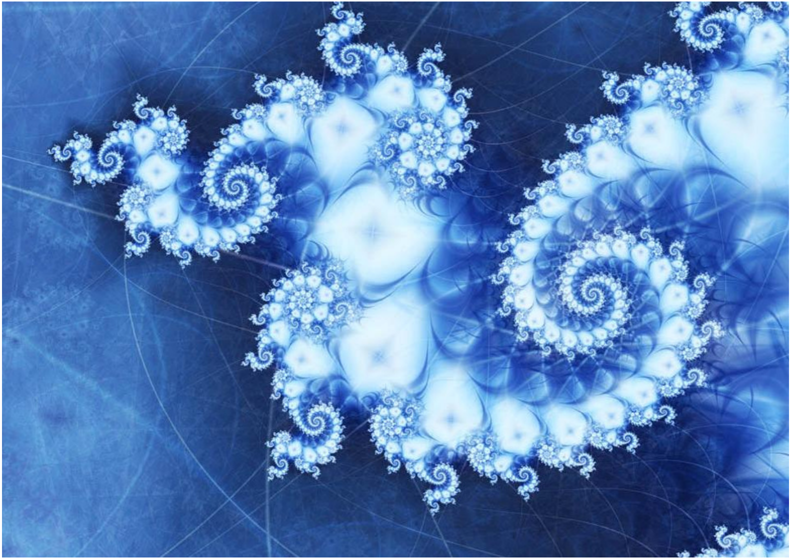


Figure: uknowhy.com

Example: Fractal Geometry

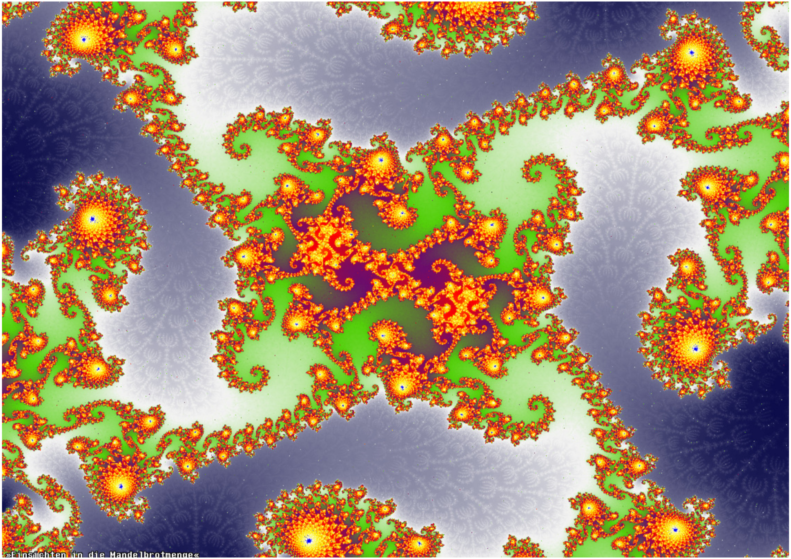


Figure: math.rochester.edu

Example: Particle System

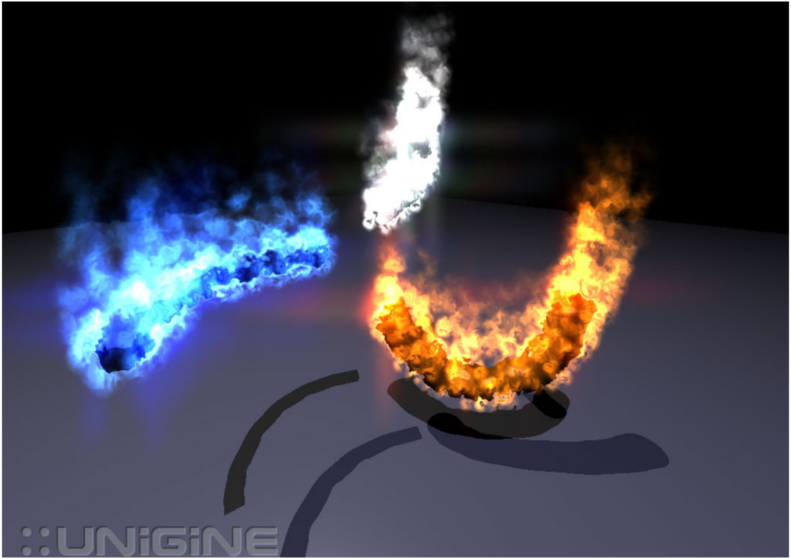


Figure: unigine.com

Example: Particle System

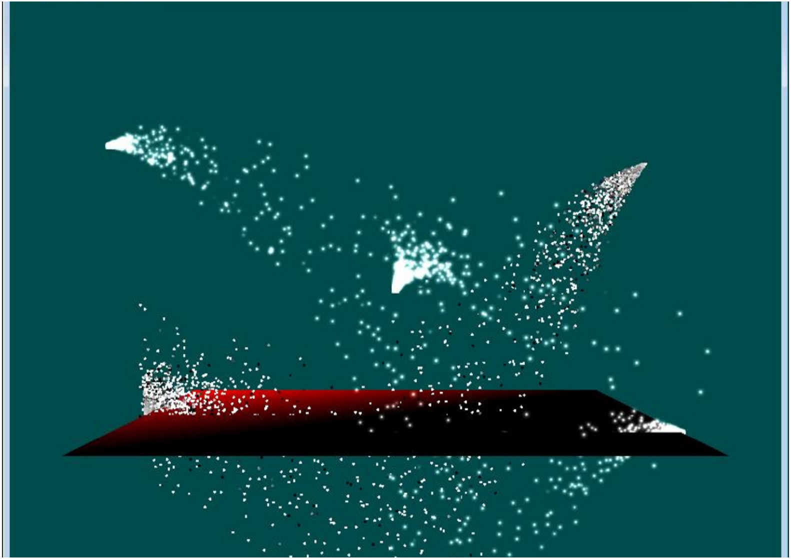


Figure: johnfragkoulis.files.wordpress.com

Basics of Lindenmayer Systems

Figure: algorithmicbotany.org

Motivation

- Development in the generation of synthetic scenes.
- Natural scenes are very complex.
- Analytical description of natural scenes is too difficult/time-consuming, sometimes even impossible.

Selfsimilarity

It seems that a part of the natural object is similar to the object, i.e. natural objects are seflsimilar, i.e. invariant to scaling.

For example:

- A branch of a tree is similar to whole tree.
- A stone is similiar to rock.

Motivation

Based on the character of natural object, there are several systems for describing seflsimilar objects:

- Dynamic systems with fractal structure.
- Iterated function system.
- Stochastic fractals.
- Lindenmayer systems.

Aristid Lindenmayer

- 1925 - 1989
- Hungarian biologist.
- L-systems in 1968.
- Theoretical framework for studying cellular structures.
- Originally no graphical reprezentation.

Later: Przemyslaw Prusinkiewicz, Development in L-systems, Turtle Interpretation, Plants Development.

Aristid Lindenmayer (19251989)

I. Discrete Modules

Based on assumption that organism can be considered as a set of discrete modules.

Example of discrete modules in biology: Part of a tree (trunk, bud, leave, blossom, etc).

II. Development of Modules

Development of modules can be predicted and their mutual transformation can be captured in the form of rewriting rules.

Rewriting rules are of the form $A \rightarrow xyz$, i.e. *predecesor* $\rightarrow$ *successor*.

III. Development in Discrete Time

Development of a base structure (called *Axiom*) in the discontinuous-time (discrete time).

Between each two moments, the structure is transformed so that all the modules are replaced by a right-hand side of the rewriting rules, that is by successors.

I. Module Interpretation

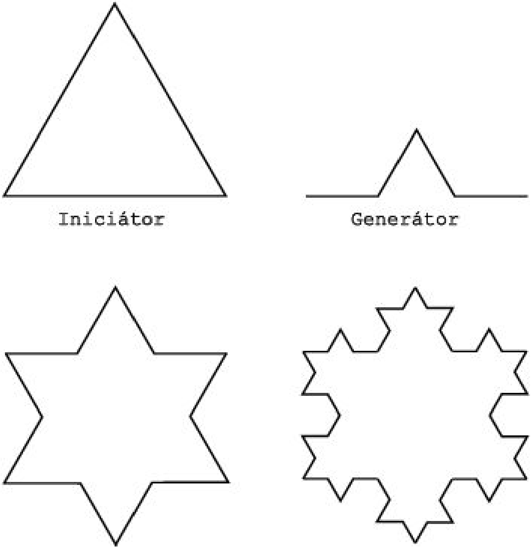
Geometrical meaning assigned to each discrete module.

For example, module F means draw a line, module $+$ means rotation to the left, $-$ means rotation to the right, etc.

II. Turtle Interpretation

Each module is a command for imaginary turtle that goes through the space and interprets geometrical meaning of a module.

Basics of Turtle Interpretation



- L-systems is mathematical theory for describing simple cellular structures.
- Emphasis was placed on a description of the topology.
- Geometric aspects were not been taken into account.
- It was not possible to model more complex organisms such as plants.

The Idea of Rewriting Systems

- We have string of symbols and a set of rewriting rules.
- The idea is to rewrite *all* the symbols in strings by the rules.

For example:

- String of modules: ABC
- Set of rules: $\{A \rightarrow AXY, B \rightarrow BB, C \rightarrow C\}$
- $ABC \Rightarrow AXYBBC \Rightarrow AXYXYBBBBC$

History of Rewriting Systems

- Thue (begining of 20th century): First definition of Rewriting Systems.
- Chomsky (1950s): Sophisticated theory of *Sequential* Rewriting Systems (Chomsky Hiearachy).
- Backus and Naur (1960s): Programing language ALGOL-60 based on Rewriting System Theory.
- Lindenmayer (1968): *Paralel* Rewriting Systems (L-systems)

Difference betwen Chomsky Hierarchy and L-systems

Let us have a string of modules. On each step, Chomsky rewrites just one occurence of selected module, Lindenmayer rewrites all occurences of selected module.

Parallel application of rules in L-systems corresponds to the laws of nature, i.e. leaves on the tree also grow at the same time, not one by one.

Deterministic Context-Free L-Systems

- *d*: Deterministic: just one rule with *A* on the left-hand side
- 0: context-free: no matter what is before/after currently rewrited symbol

Example:

- Let us have strings that contains only symbols *a* and *b*.
- Let us have a rewriting rule for each symbol *a* and *b*, $a \rightarrow ab$, $b \rightarrow a$.
- The rewriting process begins on *axiom*, let axiom be *b*.
- 1st derivatio step: $b \rightarrow a$
- 2nd derivation step: $a \rightarrow ab$
- 3rd derivation step: $ab \rightarrow aba$
- 4th derivation step: $aba \rightarrow abaab$
- 5th derivation step: $abaab \rightarrow abaababa$

Turtle Interpretation

Earlier methods:

- Frijters and Lindenmayer, Hogeweg and Hesper 1974: Basics of geometrical interpretation of strings, especially for capture the branching of plants.
- Smith: Modeling and realistic rendering of plants, based on the results of Hogeweg and Hesper.
- Szilard and Quinton (1979): Alternative approach for interpreting L-systems, d0L-systems for generating fractals.
- Siromoney a Subramanian: Space-filling curves based on L-systems.
- Prusinkiewicz: Interpretation of L-systems based on turtle.

Position of the Turtle in the Space

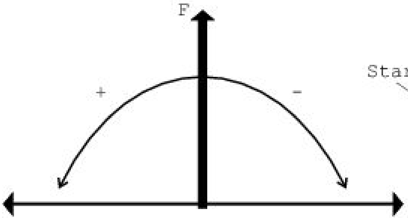
To specify a state of the turtle in the two-dimensional space, we need

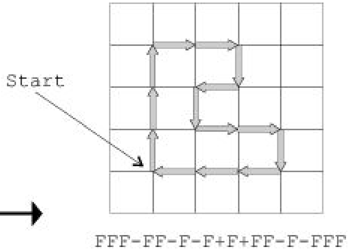
1. x, y: Position of the turtle in Cartesian coordinates.
2. α: Heading, angle representing the direction the turtle is looking.

Common comands for the turtle:

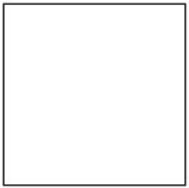
- F: Go straight; change the state to $x' = x + d.cos\alpha$, $y' = y + d.sin\alpha$, where d is step size; draw a line from $[x, y]$ to $[x', y']$.
- f: The same as F, without drawing a line.
- $+$: Rotation of turtle by δ to the left, change α to $\alpha = \alpha + \delta$.
- $-$: Rotation of turtle by δ to the right, change α to $\alpha = \alpha - \delta$.

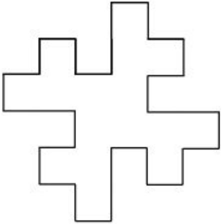
Example: Turtle Graphics in Two Dimensions

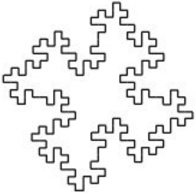




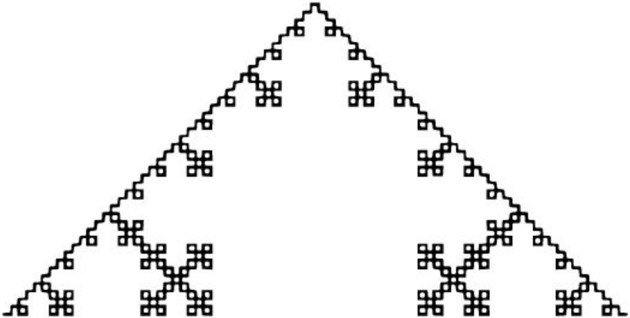
$$
\begin{aligned}
\delta &= 90^\circ \\
\omega &: \quad F - F - F - F \\
p &: \quad F \rightarrow F - F + F + FF - F - F + F
\end{aligned}
$$



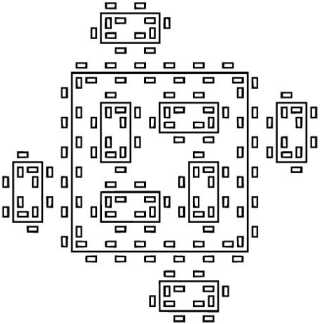




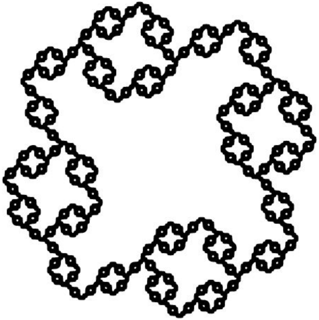
$$
\begin{array}{rrcl}
4g & \delta & = & 90^\circ \\
 & \omega & = & -F \\
 & F & \rightarrow & F+F-F-F+F
\end{array}
\tag{1}
$$



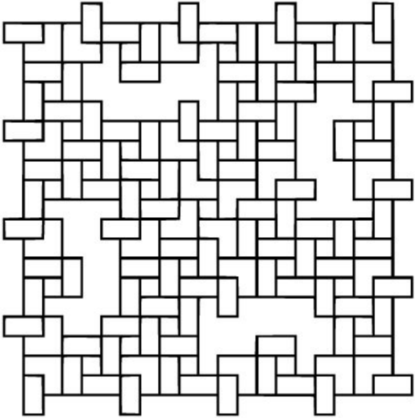
$$
\begin{array}{lrcl}
2g & \delta & = & 90^\circ \\
 & \omega & = & F+F+F+F \\
 & F & \rightarrow & F+f-FF+F+FF+Ff+FF-f+FF-F-FF-Ff-FFF \\
 & f & \rightarrow & ffffff
\end{array}
\tag{2}
$$



$$
\begin{array}{llcl}
4g & \delta & = & 90^\circ \\
 & \omega & = & F-F-F-F \\
 & F & \rightarrow & FF-F-F-F-F-F+F
\end{array}
\tag{3}
$$



$$
\begin{aligned}
3g \quad \delta &= 90^\circ \\
\omega &= F-F-F-F \\
F &\rightarrow FF-F+F-F-FF
\end{aligned}
$$

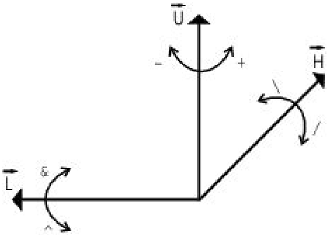


- Based on Abelson and diSessa.
- Orientation of the turtle in three-dimensions is given by three vectors *H*, *U*, *L*, where:

Position of Turtle in 3D

- *H*: heading, the direction, where the turtle is looking and where it is going,
- *U*: up, the direction where the mail of the turtle is,
- *L*: left, the direction of the left hand of the turtle.

The length of each of the vectors is 1 and the vectors are mutually perpendicular.



Rotation of the turtle in three dimensions is given by rotation matrices:

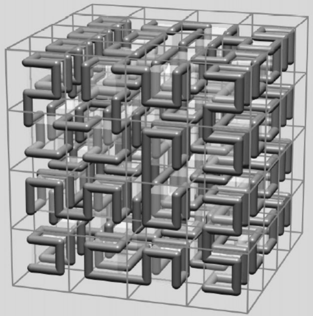
$$
\begin{aligned}
\mathbf{R_H}(\alpha) &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix} \\
\mathbf{R_L}(\alpha) &= \begin{bmatrix} \cos\alpha & 0 & -\sin\alpha \\ 0 & 1 & 0 \\ \sin\alpha & 0 & \cos\alpha \end{bmatrix} \\
\mathbf{R_U}(\alpha) &= \begin{bmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}
\end{aligned}
$$

There is much mode “simple” mathematics.

New symbols for interpretation in three dimensions.

- $+$: Left rotation of the turtle by angle δ using matrix $R_U(\delta)$
- $-$: Right rotation of the turtle by angle δ using matrix $R_U(-\delta)$
- &: Down rotation of the turtle by angle δ using matrix $R_L(\delta)$
- $\wedge$: Up rotation of the turtle by angle δ using matrix $R_L(-\delta)$
- $\backslash$: Right longitudinal axis Rotation of the turtle by angle δ using matrix $R_H(\delta)$
- $/$: Left longitudinal axis rotation of the turtle by angle δ using matrix $R_H(-\delta)$
- $|$: Turn around the turtle using matrix $R_H(180)$

$$
\begin{aligned}
\delta &= 90^\circ \\
\omega &= A \\
p: A &\rightarrow B - F + CFC + F - D\&F \wedge D - F + \&\&CFC + F + B// \\
B &\rightarrow A\&F \wedge CFB \wedge F \wedge D \wedge \wedge - F - D \wedge |F \wedge B|FC \wedge F \wedge A// \\
C &\rightarrow |D \wedge |F \wedge B - F + C \wedge F \wedge A\&\&FA\&F \wedge C + F + B \wedge F \wedge D// \\
D &\rightarrow |CFB - F + B|FA\&F \wedge A\&\&FB - F + B|FC//
\end{aligned}
$$



- Meduna Alexander.
  Automata and Languages: Theory and Applications, London, Springer, 2000.
  Spring, 2000.
- Meduna Alexander, Švec Martin.
  Grammars with Context Conditions and Their Applications.
  Wiley, 2005.
- Przemyslaw Prusinkiewicz, Aristid Lindenmayer
  The Algorithmic Beauty of Plants.
  From URL:
  `http://algorithmicbotany.org/papers/#webdocs`
- Rozenberg Grzegorz, Salomaa Arto.
  Lindenmayer systems.
  Springer-Verlag, 1992.

Thank you for your attention!

End