L-system in M4

https://github.com/jkubin/L-system

OpenAlt 2018
Josef Kubín

Contents

- What is L-system
- Chomsky grammar and L-system
- Fractals
- Turtle graphic
- L-system in M4
- Koch curve
- Fibonacci tree

What is L-system (Lindenmayer-system)

- Aristid Lindenmayer (1968)
- Parallel rewriting system
  - **A type of formal grammar**
- DOL-system
  - Deterministic Context Free L-system
  - My implementation in M4
- {D,P,E,T}{O,I}L-system

Aristid Lindenmayer (1925-1989)

L-system example (anabeana catenula)

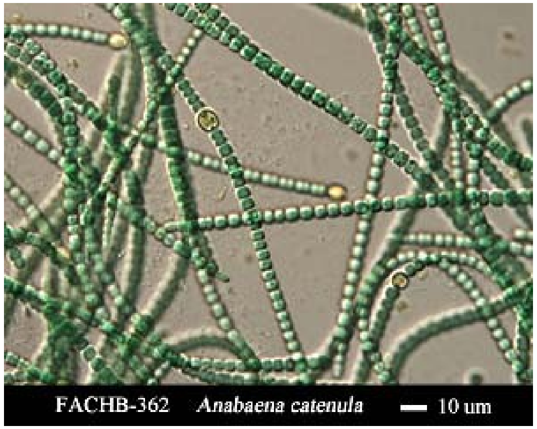
V: {A, B}

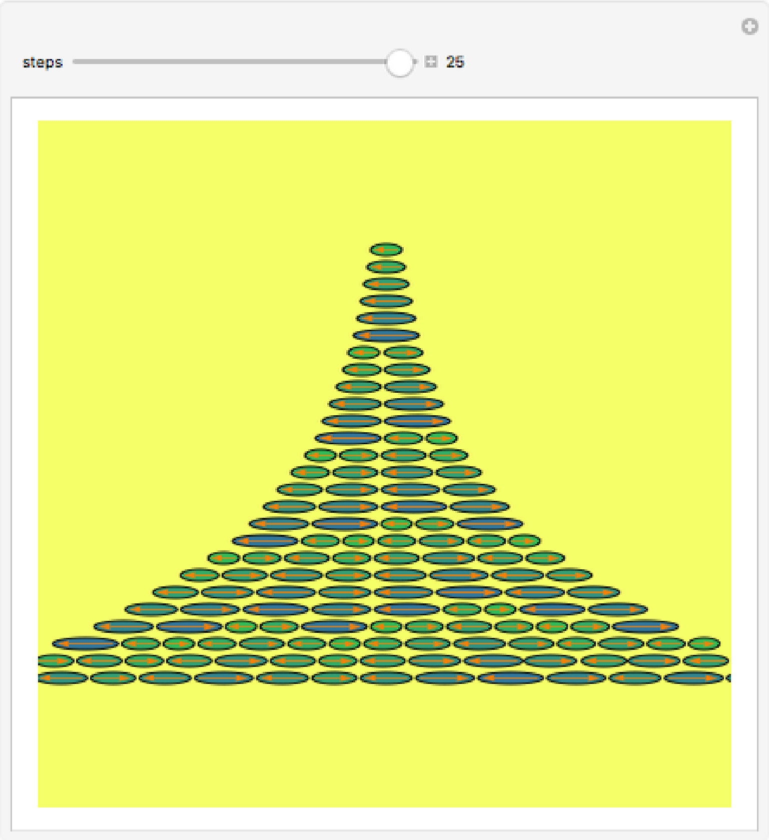
ω: A

P: A → AB

B → A

```
n = 0:         A
              / \
n = 1:       A   B
            /|    \
n = 2:     A B     A
          /| |     |\
n = 3:   A B A     A B
        /| | |\    |\ \
n = 4: A B A A B   A B A
```





L-system

G = (V, ω, P)

V: alphabet, a finite set of variables
and constants

ω: start, **axiom** or initiator

$\boldsymbol{\omega \in V^+}$

P: a fin. set of production (rewrite) rules,

$\mathbf{P \subset V \times V^*}$

Chomsky grammar

$G = (N, \Sigma, P, S)$

N: fin. set of nonterminal symbols

Σ: fin. set of terminal symbols

$$N \cap \Sigma = \emptyset$$

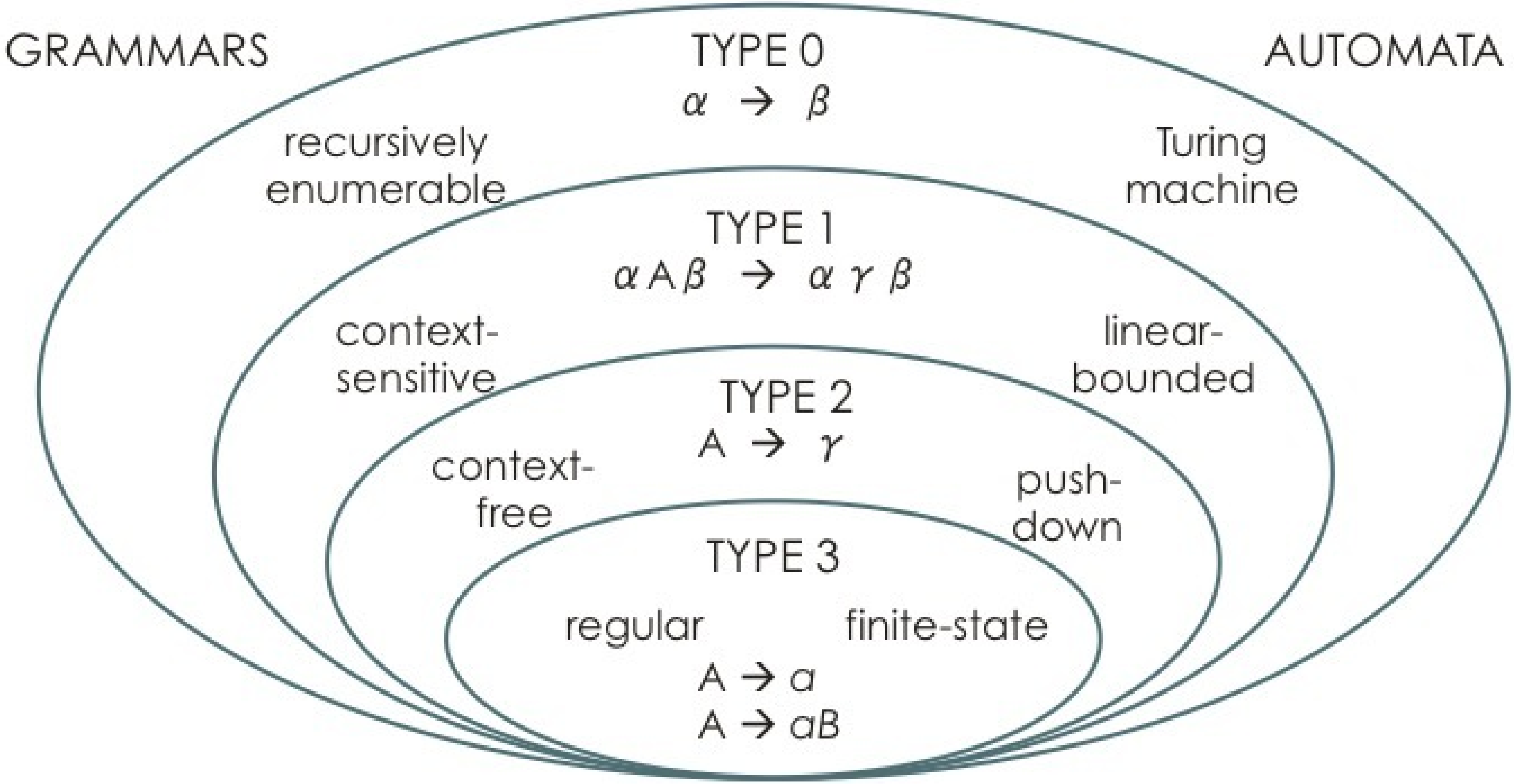
P: fin. set of production (rewrite) rules

$$(N \cup \Sigma)^* N (N \cup \Sigma)^* \rightarrow (N \cup \Sigma)^*$$

S: is the start symbol

$$S \in N$$

Chomsky Hierarchy



GRAMMARS

AUTOMATA

TYPE 0

$\alpha \rightarrow \beta$

recursively enumerable

Turing machine

TYPE 1

$\alpha A \beta \rightarrow \alpha \gamma \beta$

context-sensitive

linear-bounded

TYPE 2

$A \rightarrow \gamma$

context-free

push-down

TYPE 3

regular

finite-state

$A \rightarrow a$

$A \rightarrow aB$

L-system and Chomsky Grammar main difference

- L-system
  - rewriting rules are applied in parallel
- Chomsky
  - rewriting rules are applied sequentially

Fractals

1) Self-similarity
   - Parts resemble the whole
2) Simple rules to generate
   - Seems to be very complicated

L-system in M4

A → AB

B → A

A → A

B → B

```
define(`A', `ifelse(`$1', `0', ``A'', `A(decr($1))B(decr($1))')')
define(`B', `ifelse(`$1', `0', ``B'', `A(decr($1))')')
```

```
A(4) → … → ABAABABA
```

L-system in M4

A → AB

B → A

A → A

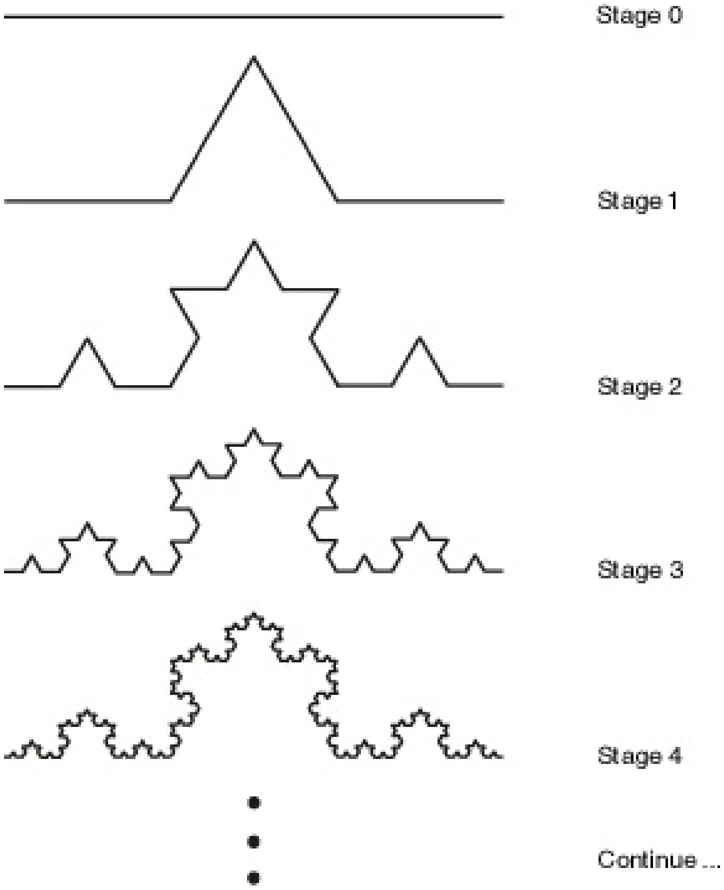
B → B

```
RULE(`A', `AB', `A')
RULE(`B', `A', `B')
```

```
$ m4 lsys.m4 algae.m4
ABAABABA
```

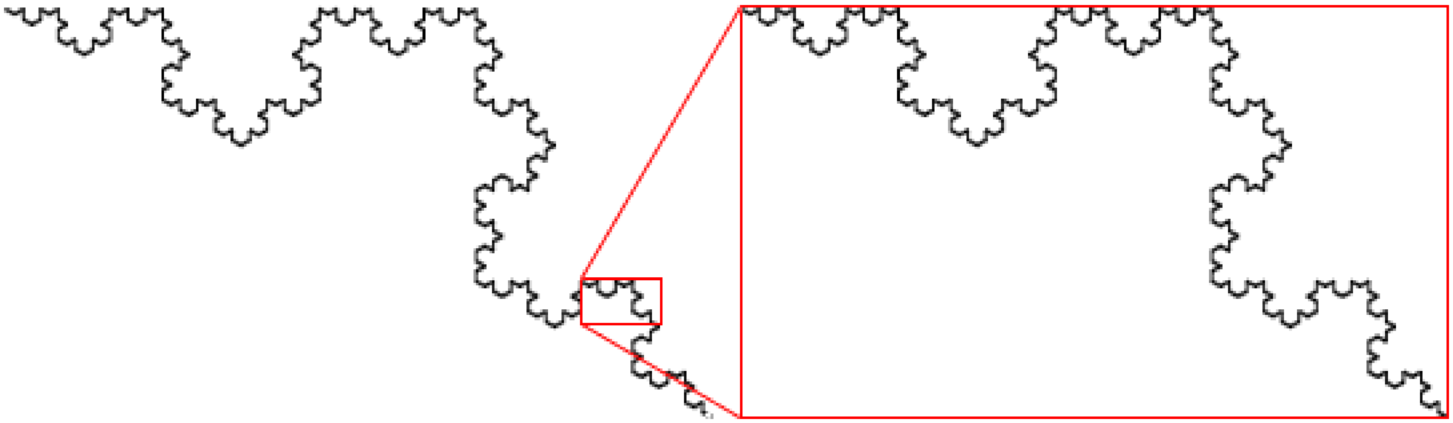
Koch curve (1904)

Helge von Koch (1870-1924)



Koch curve

Self similarity:



Koch curve
L-system

```
ANGLE: 60

VARS: F

AXIOM: F

RULE: F → F+F--F+F
```

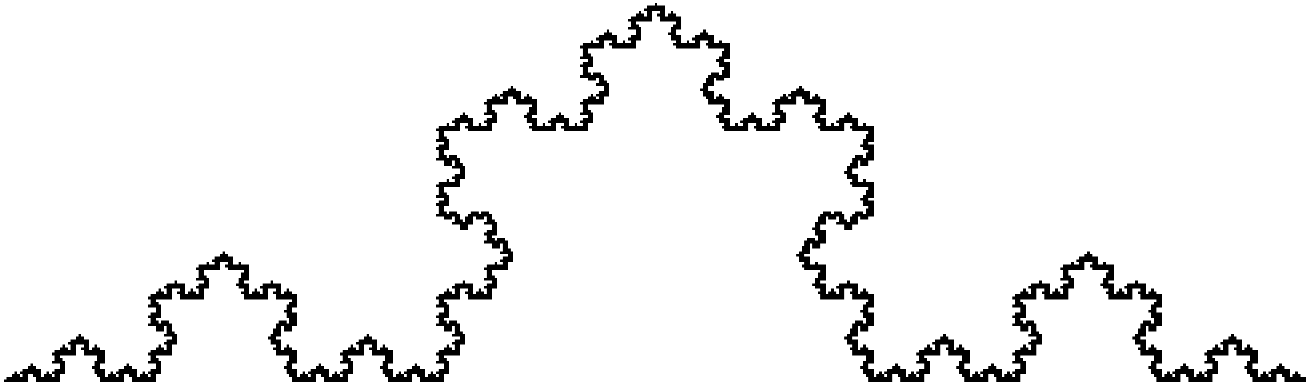
Koch curve
L-system in M4

```
ANGLE(60)
VARS(`F’)
AXIOM(`KOCH’, `F’)
RULE(`F’, `F+F--F+F’, `F’)


KOCH(3) → F(3)
F(3) → F(2)+F(2)--F(2)+F(2)
F(2) → F(1)+F(1)--F(1)+F(1)
F(1) → F(0)+F(0)--F(0)+F(0)
F(0) → F
```

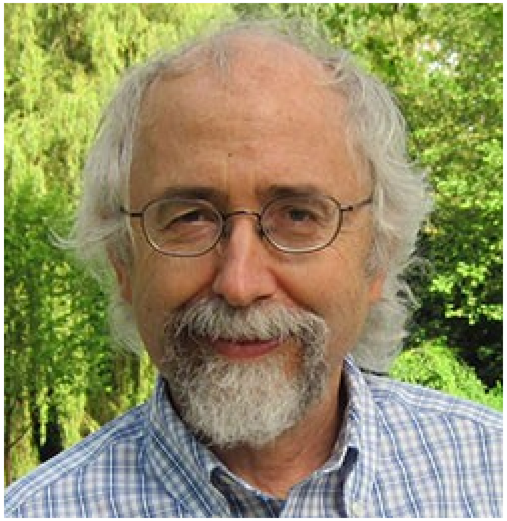
Koch curve
L-system in M4

```
$ m4 lsys.m4 koch_curve.mc | turtle_plotter
```



Turtle graphic

- **Logo** educational programming language
  - Known for using **turtle graphic** (turtle writes lines)
- L-system symbols are **turtle graphic** commands
  - Przemyslaw Prusinkiewicz (1986)



Turtle graphics (2D) in L-system

F move forward a step **d** (a line is drawn)

f move forward a step **d** (without drawing a line)

+ turn to the left by angle **δ**

– turn to the right by angle **δ**

[push current turtle state on the stack

] pop a state from the stack and set turtle

Turtle graphics (3D) in L-system

& pitch down by angle **δ**

^ pitch up by angle **δ**

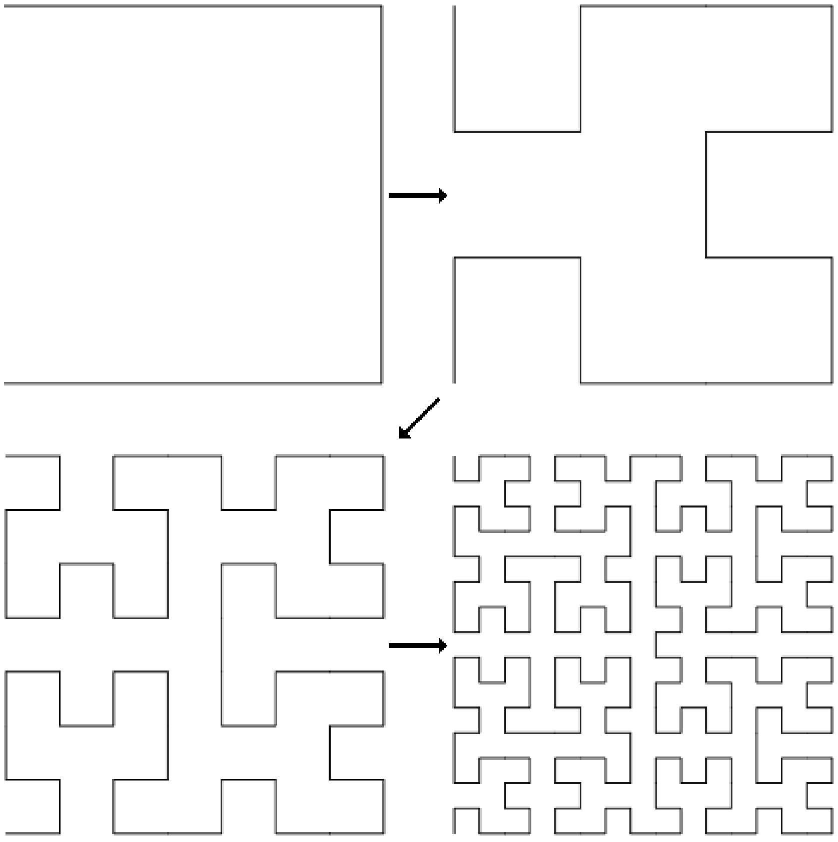
\ roll left by angle **δ**

/ roll right by angle **δ**

| turn around

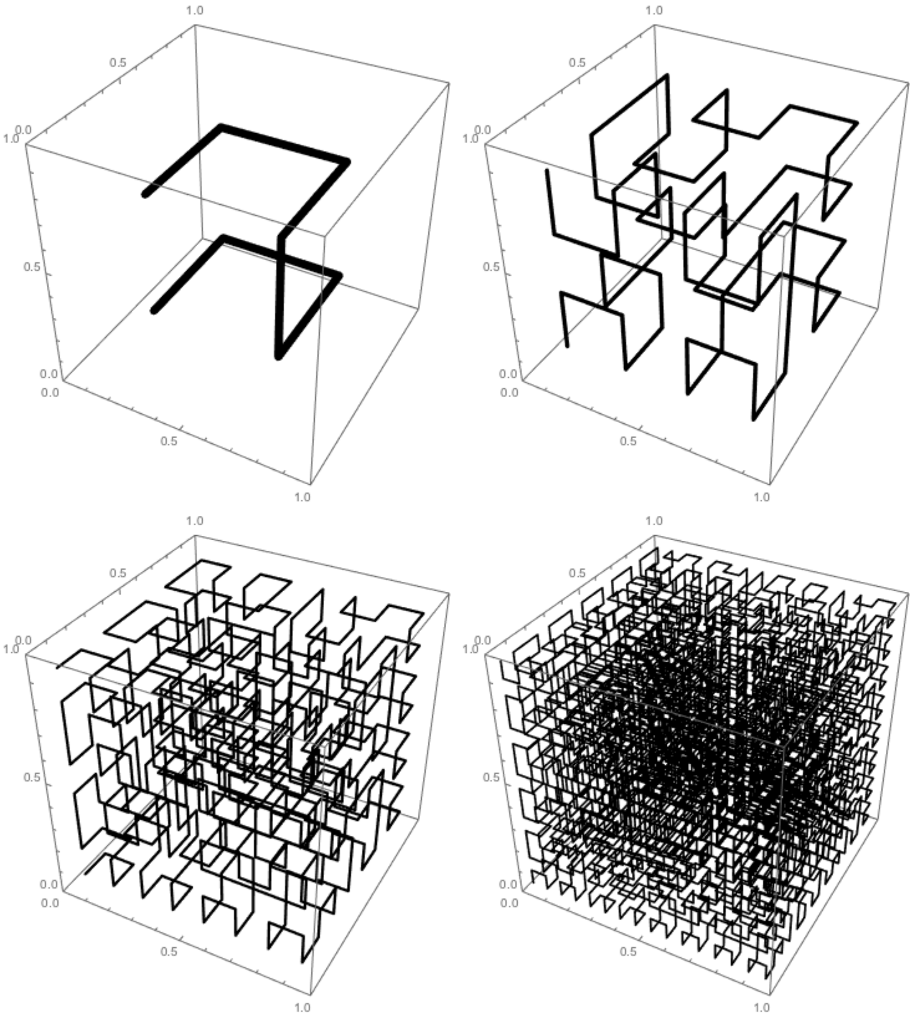
Hilbert curve

```
$ m4 lsys.m4 hilbert_curve.mc | turtle_plotter
```



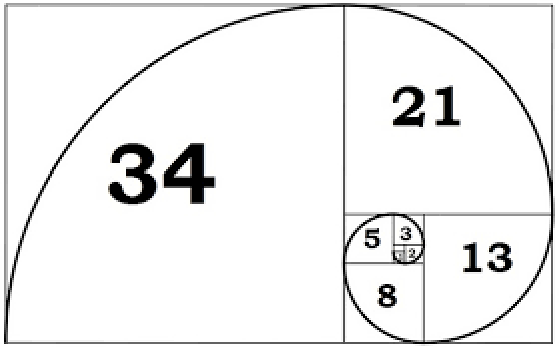
Hilbert 3D curve

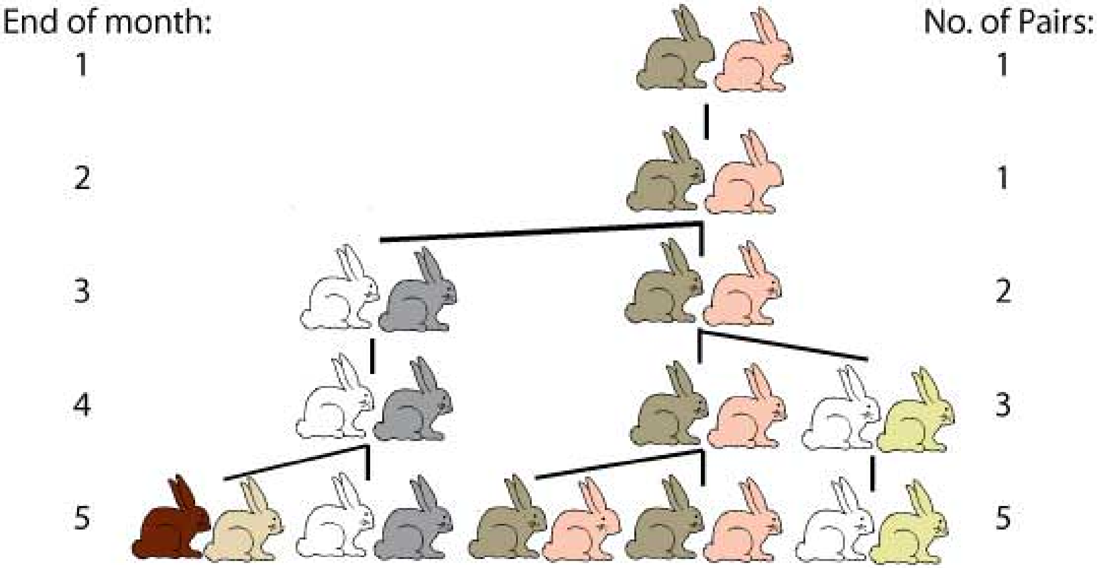
```
$ m4 lsys.m4 hilbert_curve_three_dim.mc
```

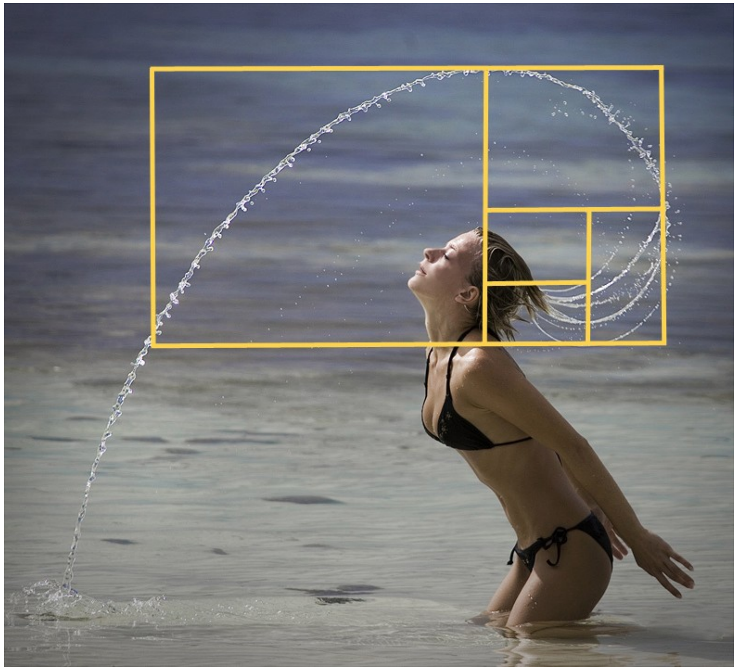


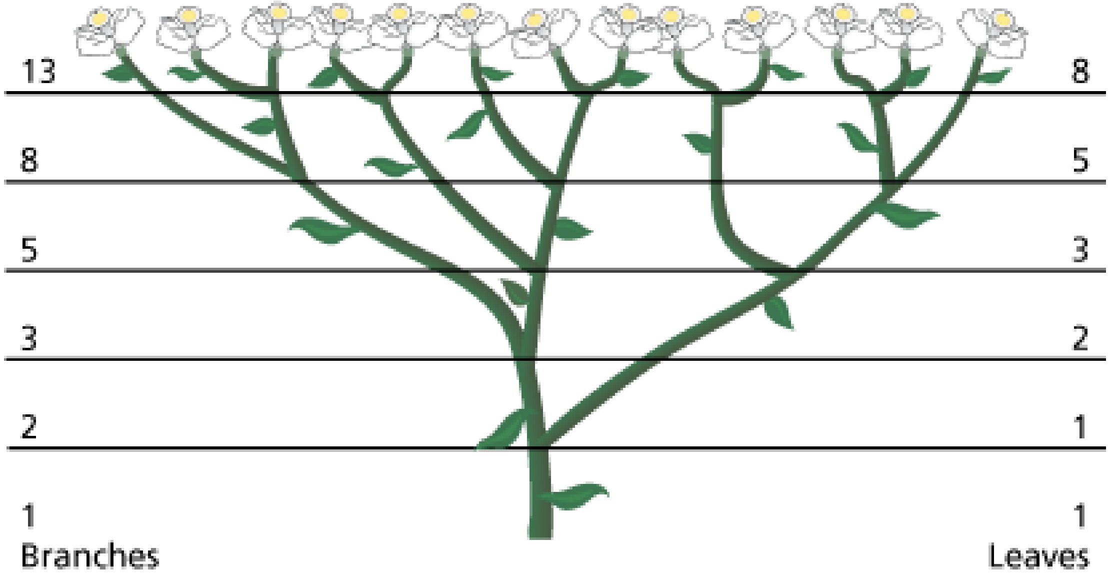
Fibonacci Sequence

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, ...



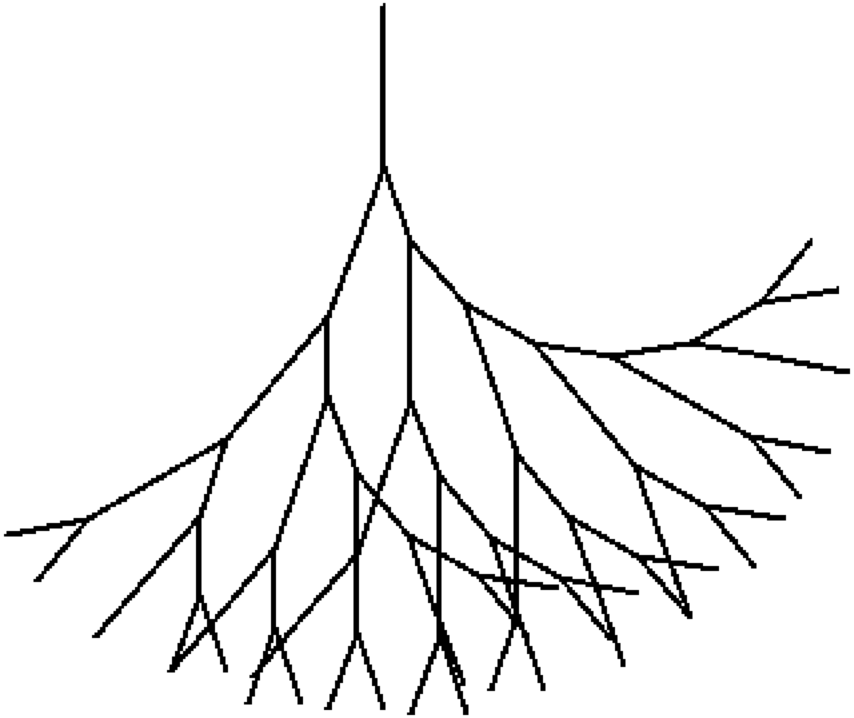






Fibonacci tree

```
$ m4 lsys.m4 fibonacci_tree.mc | turtle_plotter
```



References

https://cs.wikipedia.org/wiki/Helge_von_Koch

https://p2irc.usask.ca/profiles/theme-3/przemyslaw-prusinkiewicz.php

https://da.wikipedia.org/wiki/Leonardo_da_Pisa

https://learnodo-newtonic.com/fibonacci-facts

https://i.stack.imgur.com/Ed8DZ.png

Děkuji za pozornost!

https://github.com/jkubin/L-system

Nezapomeňte vyplnit anketu!
OpenAlt 2018

Josef Kubín