1 *Excursion: Lindenmayer Systems ("Plant grammars")

Example 1 Let's have a set of symbols $\Sigma = \{SEED, SEEDLEAF, LEAFS, STALK, BUD, REDFLOWER, BLUEFLOWER\}$. The flower field can be described as a string of Σ^*. Each plant develops by the rules:
SEED $\rightarrow$ SEEDLEAF
SRKKALEAFS $\rightarrow$ LEAFS | STALK
LEAFS $\rightarrow$ STALK
STALK $\rightarrow$ BUD | LEAFS
BUD $\rightarrow$ REDFLOWER | BLUEFLOWER
REDFLOWER $\rightarrow$ SEED | SEED SEED $|\epsilon$
BLUEFLOWER $\rightarrow$ SEED | SEED SEED $|\epsilon$

in which ϵ describes the death of the plant (it disappears without even leaving seeds).
The rules are called *context-free grammar*, because the consequence of each rule is determined only by the symbol itself (if there are several possible consequencies, then one of them is picked).

Example 2 Let's expand the symbol set: $\Sigma_2 = \Sigma \bigcup \{DAY, NIGHT\}$ and define new rules:
SEED $\rightarrow$ SEEDLEAF
SRKKALEAFS DAY$\rightarrow$ LEAFS | STALK
LEAFS DAY$\rightarrow$ STALK
STALK DAY$\rightarrow$ BUD | LEAFS
BUD NIGHT$\rightarrow$ REDFLOWER | BLUEFLOWER
REDFLOWER $\rightarrow$ SEED | SEED SEED $|\epsilon$
BLUEFLOWER $\rightarrow$ SEED | SEED SEED $|\epsilon$
DAY $\rightarrow$ NIGHT
NIGHT $\rightarrow$ DAY

Now the plants grow at day time, but flower at night time. This grammar is called *context-sensitive*, because also the context (the environment factors) affect on the consequence of the rule.

- L-systems $\sim$ simplified abstract grammar, which produces a language to describe the life and development of plants (or other living organisms)
- originally a mathematical model developed by botanist Aristid Lindenmayer to predict the growth of plants
- also used to create artificial organisms and all possible!
- a string produced by the grammar of a L-system can be interpreted as a picture
- similar to fractals, but not necessarily fractals
- General idea: in each time step the symbol of the string is modified by the neighbour symbols and a rule fitting the conditions (can also remain same) $\leftrightarrow$ *c*ell automaton, in which each cell of the grid (vector, 2- or more dimensional array) is a finite automaton
- cut all the classes of Chomsky hierarchy
  - can follow rules of regular languages i.e. of form $A \to aB$ or $A \to Ba$ (the result is one terminal symbol and one nonterminal symbol) or $A \to \epsilon$ ("death of cell", the symbol disappears) or
  - context-free grammar, i.e. rules are of form $A \to V^*$ (i.e. we modify only one symbol at the same time, independent of its neighbours, the result can be any string of terminal and nonterminal symbols, also ϵ) or
  - context-sensitive grammar, i.e. the rrules are of form $\alpha A\beta \to \alpha\omega\beta$ (i.e. we can apply the rule only in some "context", when the neighbours are α and β)
  - unristricted grammar, i.e. the rules are of form $\omega \to \omega'$ (i.e. any string can be modified to any string, string can contain both terminal and nonterminal symbols)
- e.g. one form of algae can be modelled as
  - in the beginning two kind of cells, mark A and B ("axioms")
  - rules:
    $A \to AB$
    $B \to A$

  - e.g. $AB \Rightarrow ABA \Rightarrow ABAAB \Rightarrow ABAABABA$ (3 time steps)

- the simplest L-systems are DOL-systems (contexts-free and deterministic L-system)

- e.g. 2: rules:
  $A \rightarrow CB$
  $B \rightarrow A$
  $C \rightarrow DA$
  $D \rightarrow C$

  - beginning: any "seed" of the alphabet (the starting symbol)
  - e.g. $A \Rightarrow CB \Rightarrow DAA \Rightarrow CCBCB \Rightarrow DADAADAA$

- e.g. 3: Constructing tree structure:
  $A \rightarrow C[B]D$
  $B \rightarrow A$
  $C \rightarrow C$
  $D \rightarrow C(E)A$
  $E \rightarrow D$

  - let's interprete brackets as a brach into left and common paranthesis as a brach to right
  - e.g. $A \Rightarrow C[B]D \Rightarrow C[A]C(E)A$ (2 time steps)
  - as tree:

- Application (WH): Let's create a flower field using graphic tool and the following rules:
  $SEED \rightarrow SEEDLEAF|LSTALK|RSTALK|MSTALK$
  $LSTALK \rightarrow LLEAF|LBUD$ (random colour)
  $RSTALK \rightarrow RLEAF|RBUD$
  $LBUD \rightarrow FLOWER$
  $RBUD \rightarrow FLOWER$
  $MBUD \rightarrow FLOWER$
  $FLOWER \rightarrow SEEDLEAF$
  $MSTALK \rightarrow LSTALK|RSTALK|MBUD|SEEDLEAF$
  $|LLILYSTALK|RLILYSTALK|TWIGS$
  $SEEDLEAF \rightarrow LSTALK|RSTALK|MSTALK|LEAFS$
  $LLILYSTALK \rightarrow LLILY$
  $RLILYSTALK \rightarrow RLILY$
  $LLILY \rightarrow LBERRIES$
  $RLILY \rightarrow RBERRIES$

- The screen is divided into cells, which are scanned in some order, and the possible successor of the symbol is drawn into current position (or neighbour positions)

Literature

- Prusinkiewicz & Lindenmayer: The Algorithmic Beaty of Plants
- Prusinkiewicz & Hanan: Lindenmayer Systems, Fractals, and Plants
- Rozenberg & Salomaa (toim.): Lindenmayer Systems. Impacts on Theoretical Computer Science, Computer Graphics, and Developmental Biology