Generative Design Systems
DECO2013
Design Grammars
Design Grammars

- Design grammars are production systems that can generate designs according to a set of rules (the grammar).

- A well-defined design grammar will generate designs that adhere to design constraints.
Advantages of Using a Design Grammar

- Design grammars give the designer the potential to evaluate a large number of alternative designs without laborious work.

- The designs produced by a grammar can contain designs that might have been overlooked without the aid of a grammar.

- Paving the way for possible innovative designs.
The Language of Grammars

- A set of non-terminal symbols
- A set of terminal symbols
- A set of rules, where a rule is of the form:
  - LHS → RHS
    - LHS and RHS contain terminal or non-terminal symbols
    - LHS must contain at least one non-terminal symbol
An Example Grammar

- Design Grammar
  - $S \rightarrow aSb$
  - $S \rightarrow ba$

- Produces:
  - $abab$ $aababb$ $aaaaababbb$ $aaaaababbbb$
Interpreting a Production

› In a design grammar, symbols produced by the grammar are interpreted as design elements

› E.g., commands to a drawing program:

  ‣ a = ↓
  ‣ b = →

  abab = [diagram]
  aababbb = [diagram]
Interpreting a Production

- The production of a design grammar can be interpreted in different ways
  - E.g., components for a mechanical system:
    - $a = \text{spring}$
    - $b = \text{mass}$
    - $abab = \text{spring mass spring mass}$
    - $aababb = \text{mass spring mass spring mass}$
Rules for roof fixing:
1. Place wall plate over single brick wall
2. Place wall plate over outer leaf of double brick wall
3. Place double header over timber stud of framed wall
4. Notch rafters over wall plate

Rules for rafter ends:
5. Finish rafter with vertical end.
6. Finish rafter with perpendicular end.
7. Finish rafter with right-angle end.

Example of a Real Design Grammar
Context-Free vs Context-Sensitive

- Grammars can be written to be either context-free or context-sensitive
  - Context-free grammars do not represent the context of a symbol in the rules
  - Context-sensitive grammars represent the context of a symbol using a prefix and suffix
Applying the Rules of a Grammar

- How the rules are applied has a significant impact on what can be produced
  - Rules applied in sequence
    - One symbol replaced at a time
      - Chomsky Grammars
  - Rules applied in parallel
    - All symbols replaced at once
      - Lindenmayer Grammars
Lindenmayer Grammars

- Lindenmayer Grammars were developed to model the growth of natural forms.
- Lindenmayer Systems are a very good way to produce organic forms, e.g. fractals, trees, flowers.
a
n=5, δ=25.7°
F
F → F [+F] F [-F] F

b
n=5, δ=20°
F
F → F [+F] F [-F] [F]

c
n=4, δ=22.5°
F
F → FF - [-F+F+F] + [+F-F-F]
n=5, δ=25.7°

F
F → F[+F]F[-F]F
F → F[+F]F
F → F[-F]
Deterministic vs Non-Deterministic

- Some rule interpretation systems can support non-deterministic firing of rules, e.g. L-Systems
  - Multiple rules can be specified for the same LHS and the rule fired is chosen at random
  - Some systems allow the different rules to be weighted so that they fired with different probabilities