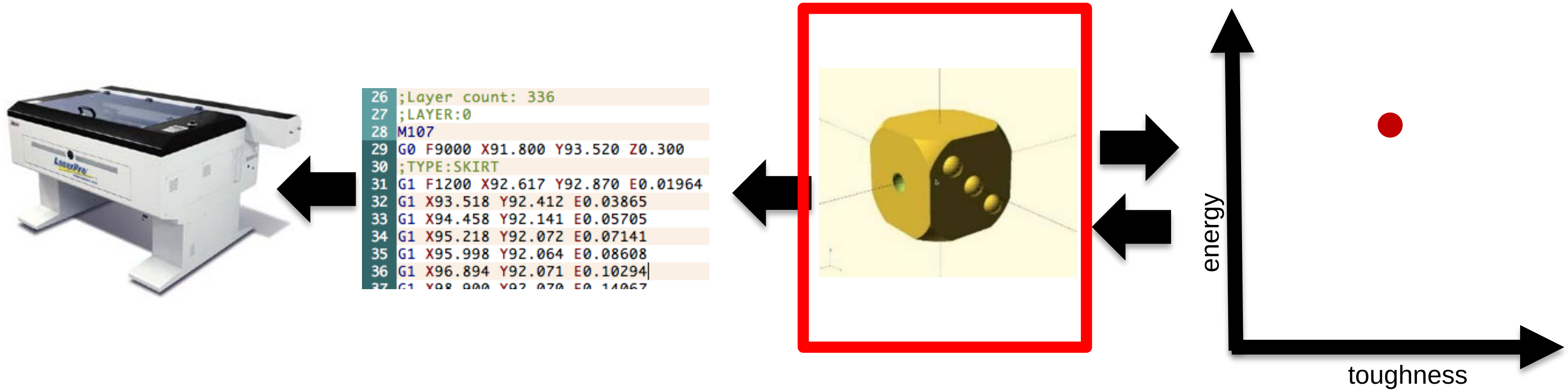
Procedural Modeling

Wojciech Matusik

CSAIL & EECS MIT

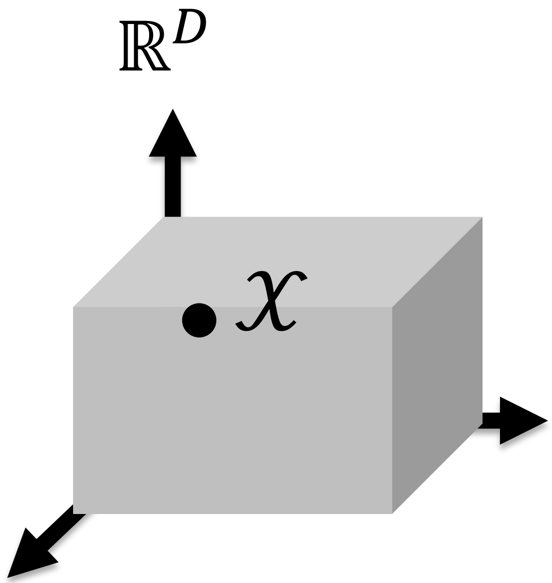
Computational Design Stack



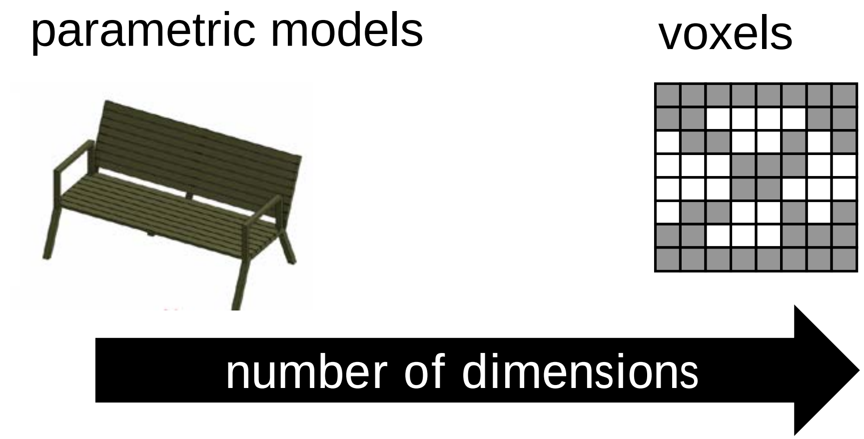
Hardware | **Machine Code** | **Direct Design** | **Functional Design**

Design Space

- Each design can be mathematically represented as a point in $\mathbb{R}^D$



Design Space



Parametric Design

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix}$$

parameters

$\Rightarrow$ **Parametric Design** $\Rightarrow$

Procedural Modeling

- Goal:
  - Describe 3D models algorithmically
- Advantages:
  - Automatic generation
  - Concise representation
  - Parameterized classes of models
  - More general than CAD parametric models
  - Dimensionality of design space might vary

Formal Grammars and Languages

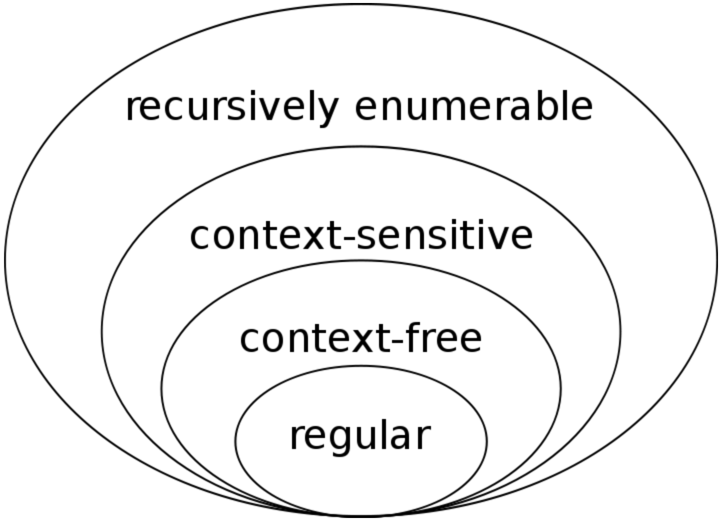
- A finite set of **nonterminal symbols: {S, A, B}**
- A finite set of **terminal symbols: {a, b}**
- A finite set of **production rules: S $\rightarrow$ AB**
- A **start symbol: S**

- Generates a set of finite-length sequences of symbols by recursively applying production rules starting with **S**

Formal Grammars and Languages

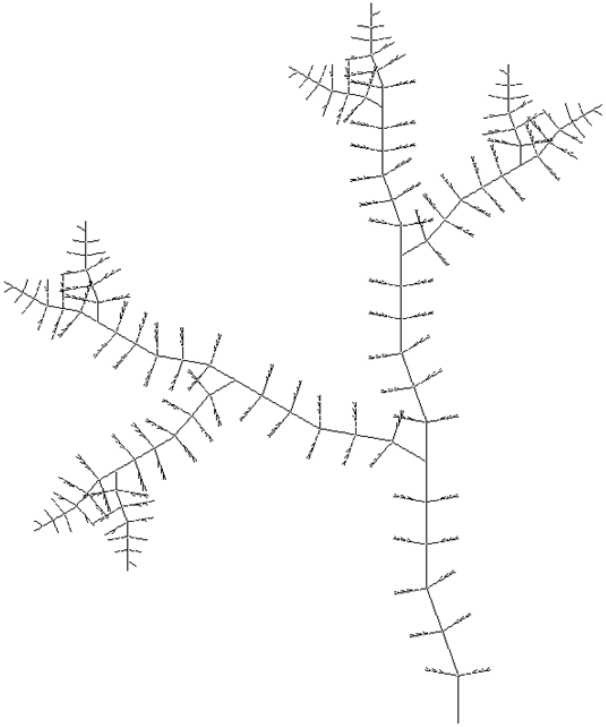
- Chomsky's four types of grammars

Grammar	Languages	Automaton	Production rules (constraints)
Type-0	Recursively enumerable	Turing machine	$\alpha \rightarrow \beta$ (no restrictions)
Type-1	Context-sensitive	Linear-bounded non-deterministic Turing machine	$\alpha A \beta \rightarrow \alpha \gamma \beta$
Type-2	Context-free	Non-deterministic pushdown automaton	$A \rightarrow \gamma$
Type-3	Regular	Finite state automaton	$A \rightarrow a$ and $A \rightarrow aB$



L-systems (Lindenmayer systems)

- A model of morphogenesis, based on formal grammars (set of rules and symbols)
- Introduced in 1968 by the Swedish biologist A. Lindenmayer
- Originally designed as a formal description of the development of simple multi-cellular organisms
- Later on, extended to describe higher plants and complex branching structures



L-system Example 1: Algae

- **nonterminals** : A B
- **terminals** : none
- **start** : A
- **rules** : $(A \rightarrow AB)$, $(B \rightarrow A)$

$n = 0$: A

$n = 1$: AB

$n = 2$: ABA

$n = 3$: ABAAB

$n = 4$: ABAABABA

$n = 5$: ABAABABAABAAB

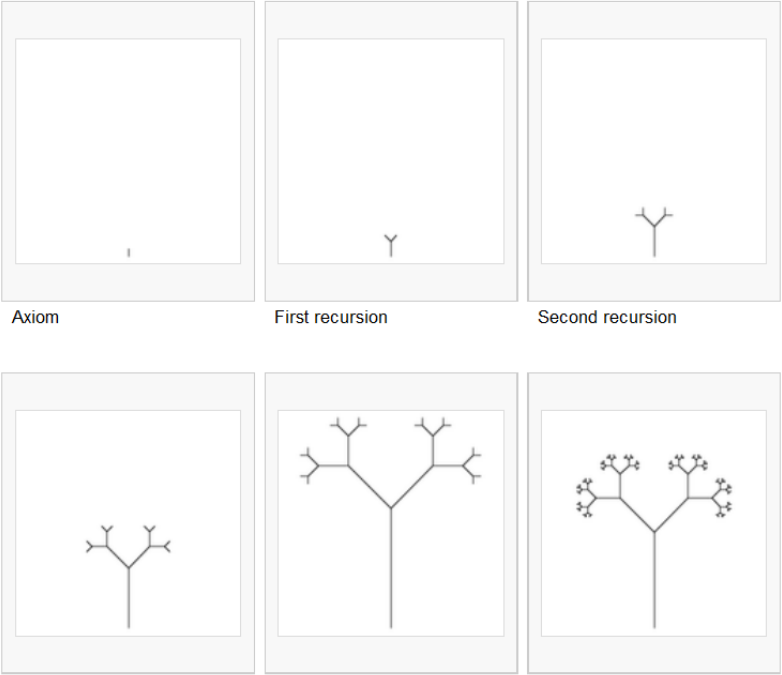
L-system Example 2

- **nonterminals** : 0, 1
- **terminals** : [,]
- **start** : 0
- **rules** : $(1 \rightarrow 11)$, $(0 \rightarrow 1[0]0)$

start:	0
1st recursion:	1[0]0
2nd recursion:	11[1[0]0]1[0]0
3rd recursion:	1111[11[1[0]0]1[0]0]11[1[0]0]1[0]0

L-system Example 2

- **Visual representation: turtle graphics**
  - **0: draw a line segment ending in a leaf**
  - **1: draw a line segment**
  - **[: push position and angle, turn left 45 degrees**
  - **]: pop position and angle, turn right 45 degrees**



Axiom | First recursion | Second recursion

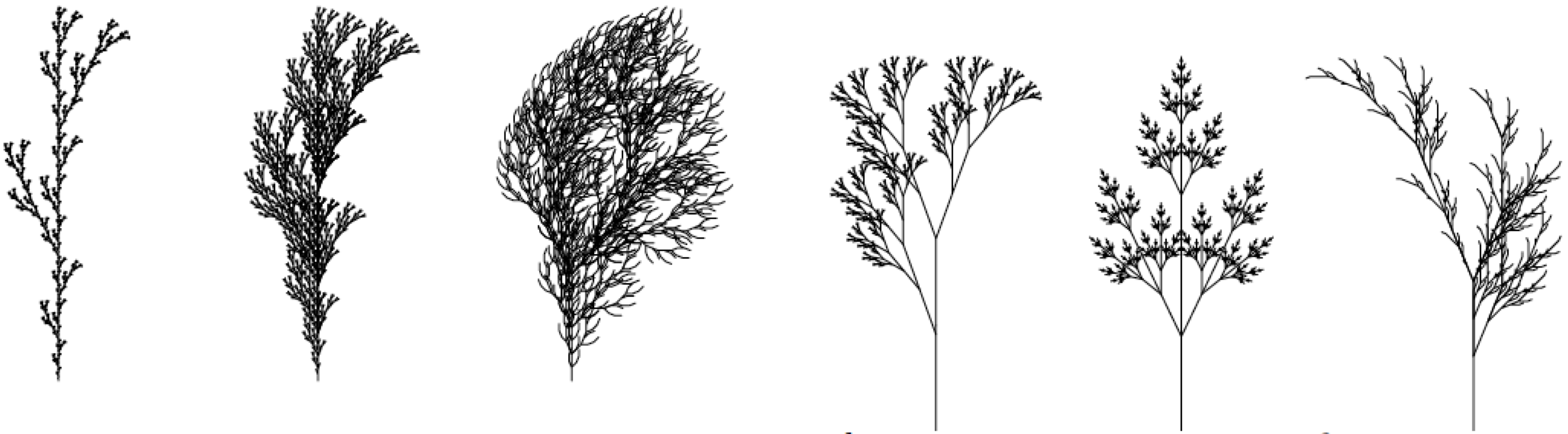
Third recursion | Fourth recursion | Seventh recursion, scaled down ten times

L-system Example 3: Fractal Plant

- **nonterminals** : X, F
- **terminals** : + - [ ]
- **start** : X
- **rules** : (X → F-[[X]+X]+F[+FX]-X), (F → FF)

L-Systems Examples

- Tree examples



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]

d
n=7, δ=20°
X
X → F[+X]F[-X]+X
F → FF

e
n=7, δ=25.7°
X
X → F[+X][-X]FX
F → FF

f
n=5, δ=22.5°
X
X → F-[[X]+X]+F[+FX]-X
F → FF

L-Systems Examples

Types of L-Systems

- *Deterministic*: If there is exactly one production for each symbol

  $0 \longrightarrow 1[0]0$

- *Stochastic*: If there are several, and each is chosen with a certain probability during each iteration

  $0\ (0.5) \longrightarrow 1[0]0$

  $0\ (0.5) \longrightarrow 0$

Types of L-Systems

- *Context-free:* production rules refer only to an individual symbol
- *Context-sensitive:* the production rules apply to a particular symbol only if the symbol has certain neighbours

$$
\begin{aligned}
S &\rightarrow aSBC \\
S &\rightarrow aBC \\
CB &\rightarrow HB \\
HB &\rightarrow HC \\
HC &\rightarrow BC \\
aB &\rightarrow ab \\
bB &\rightarrow bb \\
bC &\rightarrow bc \\
cC &\rightarrow cc
\end{aligned}
$$

Types of L-Systems

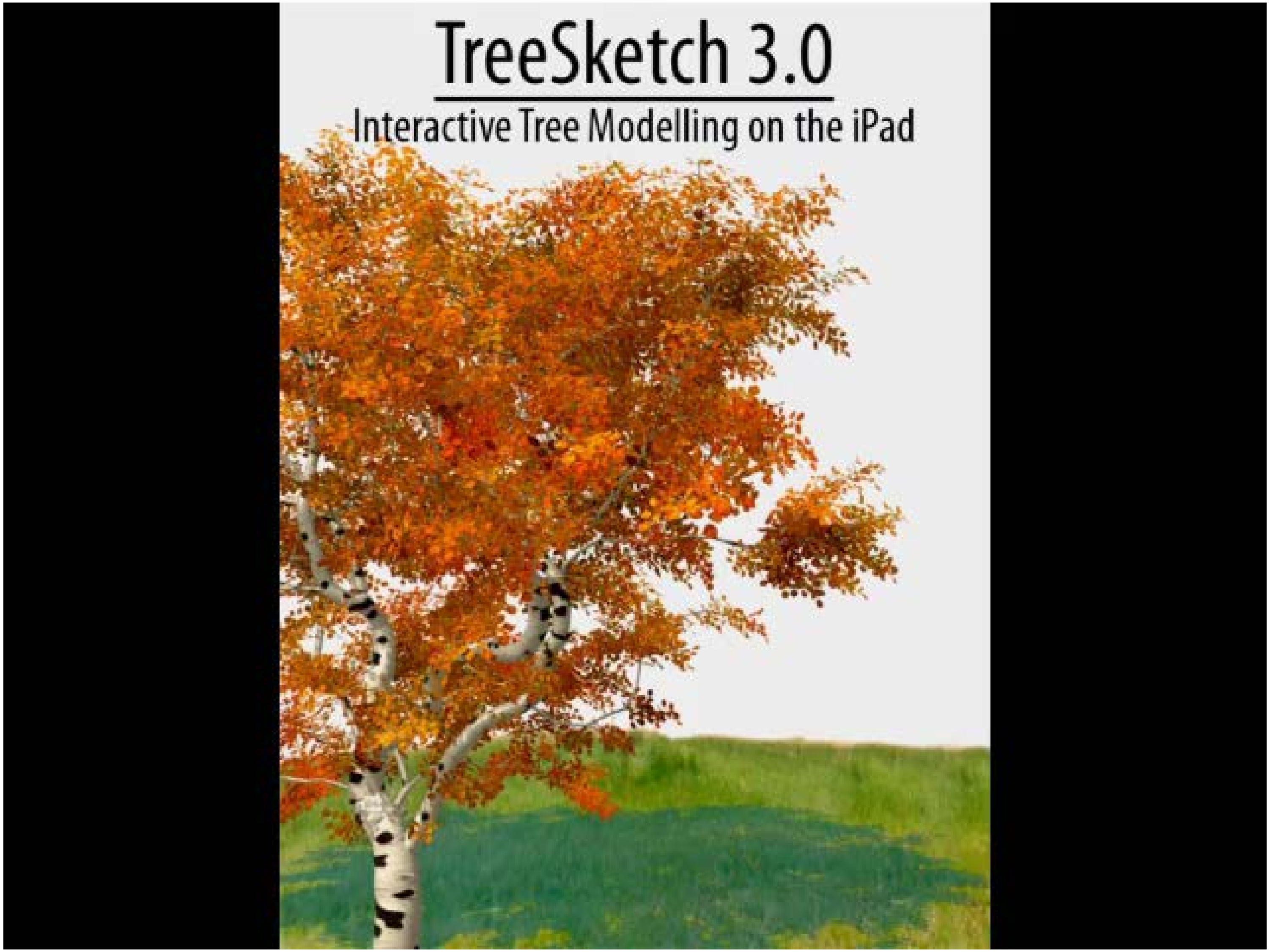
- *Nonparametric grammars:* no parameters associated with symbols
- *Parametric grammars:* symbols can have parameters
  - Parameters used in conditional rules
  - Production rules modify parameters

  - $A(x,y) : x = 0 \rightarrow A(1, y+1)B(2,3)$

Applications: Plant Modeling

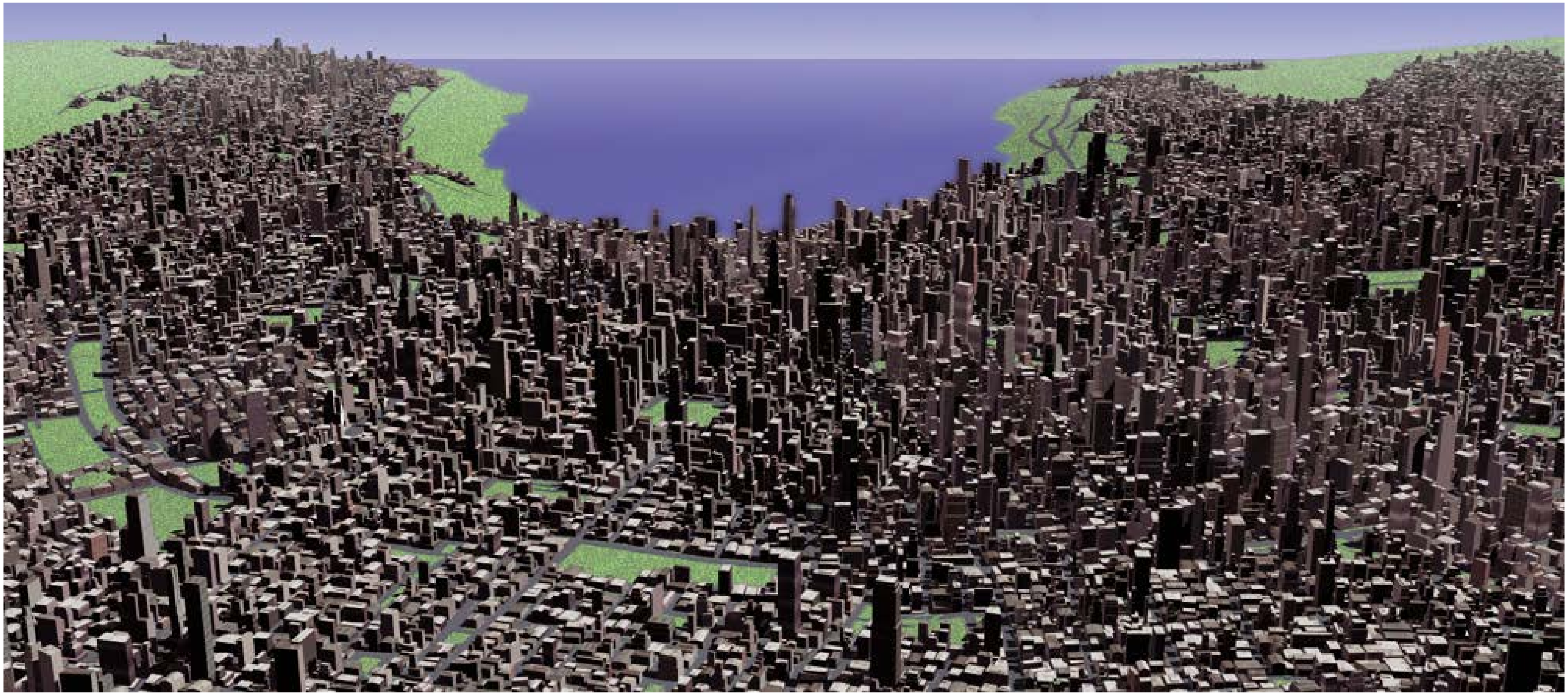
- Algorithmic Botany @ the University of Calgary
  - Covers many variants of L-Systems, formal derivations, and exhaustive coverage of different plant types.
  - http://algorithmicbotany.org/papers
  - http://algorithmicbotany.org/virtual_laboratory/

TreeSketch: Interactive Tree Modeling



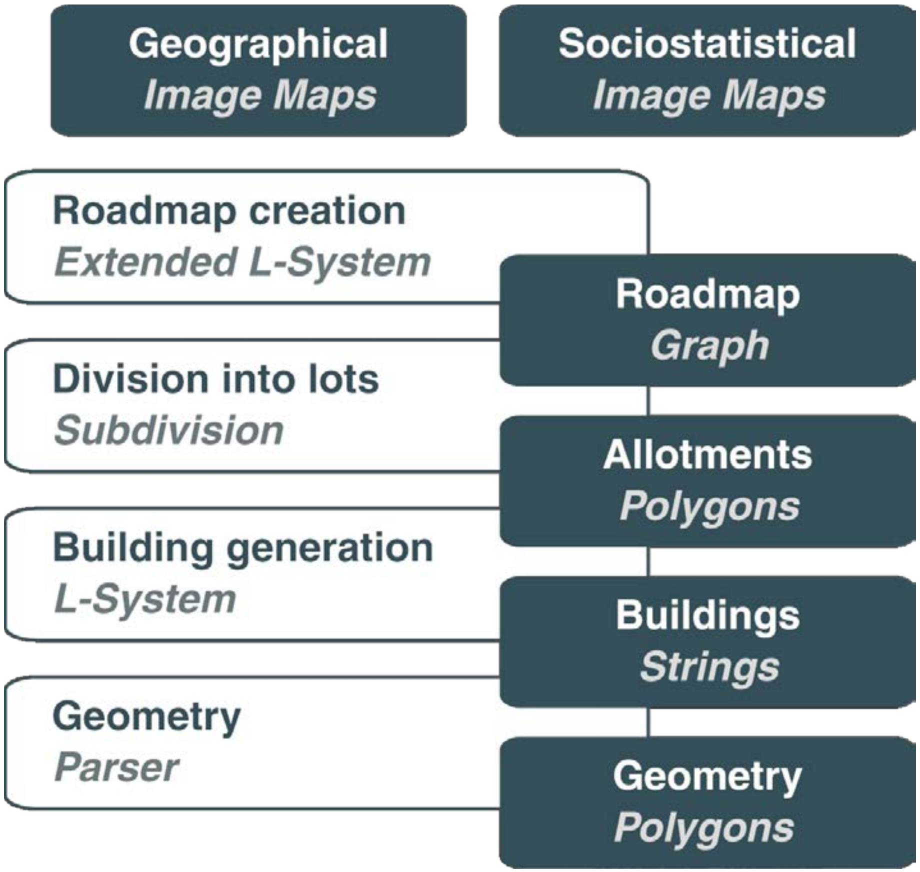
http://vimeo.com/68195050

Procedural Modeling of Cities



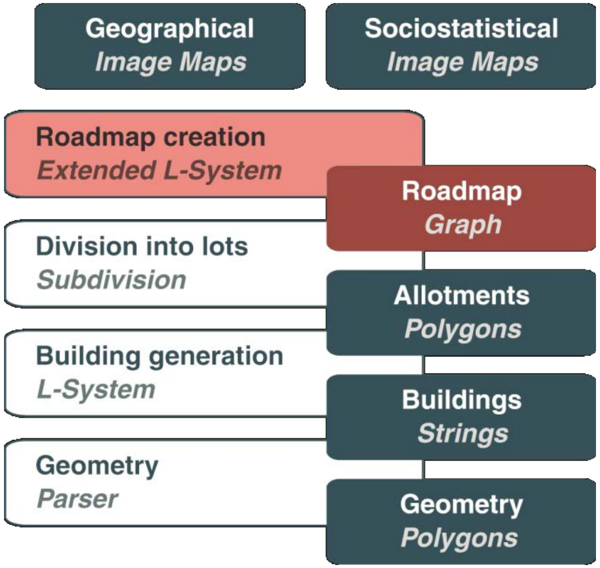
Procedural Modeling of Cities / Yoav Parish, Pascal Müller, Siggraph 2001

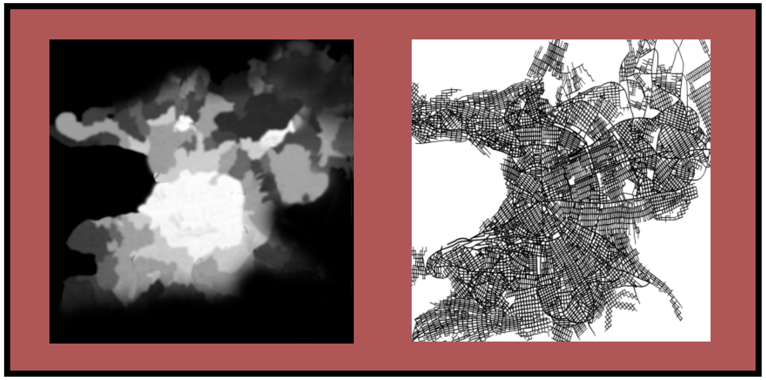
System Pipeline



Procedural Modeling of Cities / Yoav Parish, Pascal Müller, Siggraph 2001

Module 1: Streetmap Creation

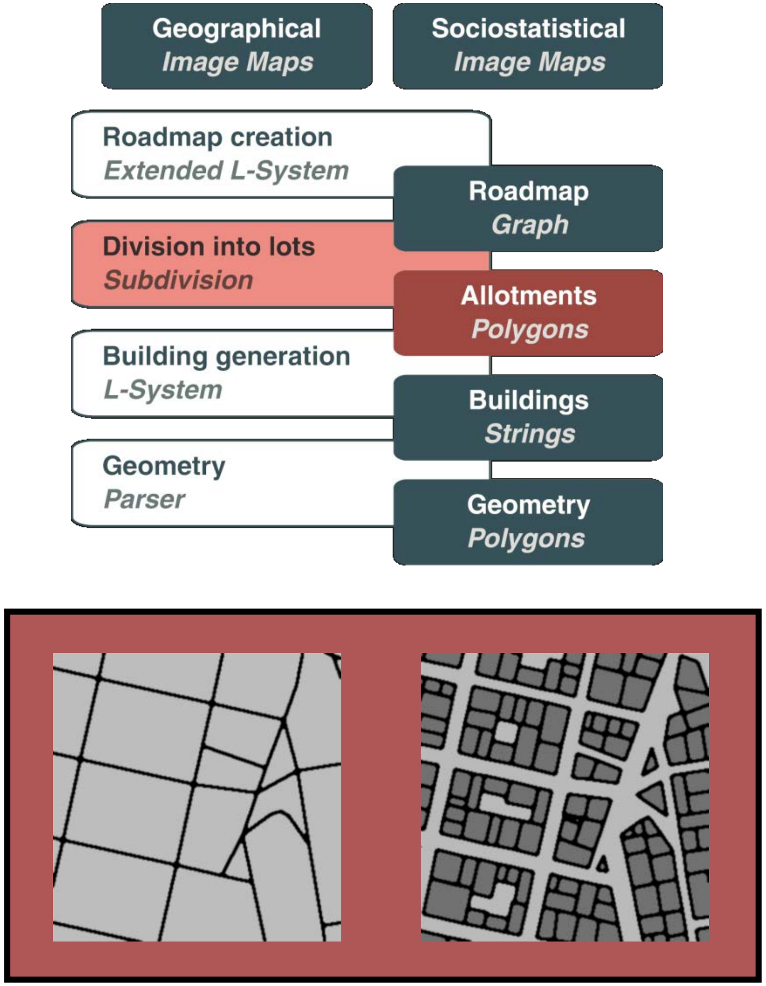




- Input:
  Image maps, parameters for rules
- Output:
  A street graph for interactive editing

Procedural Modeling of Cities / Yoav Parish, Pascal Müller, Siggraph 2001

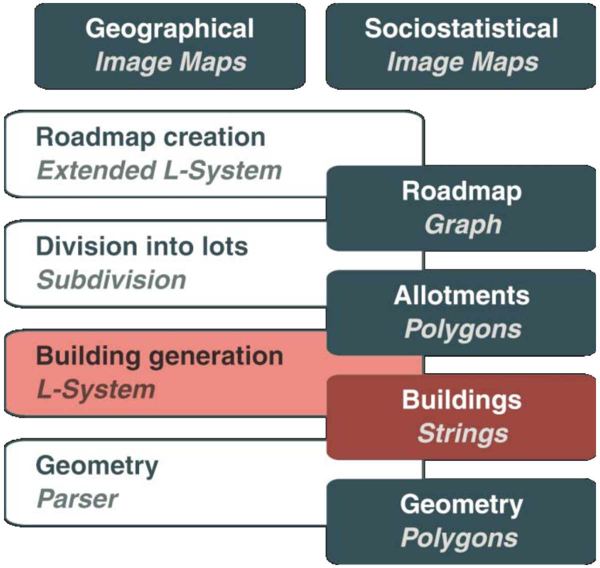
Module 2: Division into Lots

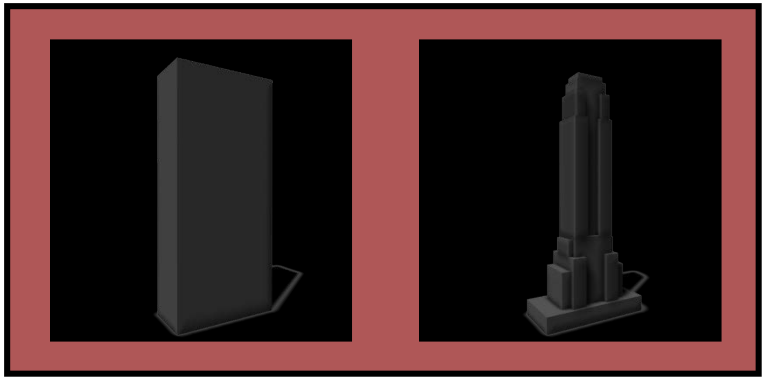


- Input:
  Street graph, area usage map
- Output:
  Polygon set of allotments for buildings

Procedural Modeling of Cities / Yoav Parish, Pascal Müller, Siggraph 2001

Module 3: Building Generation

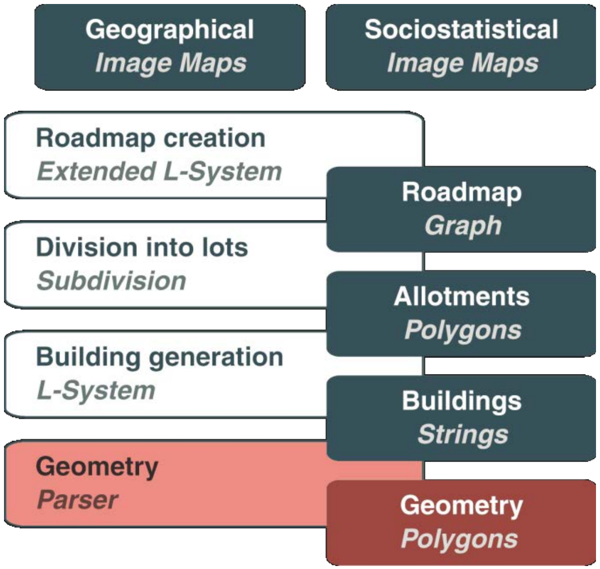


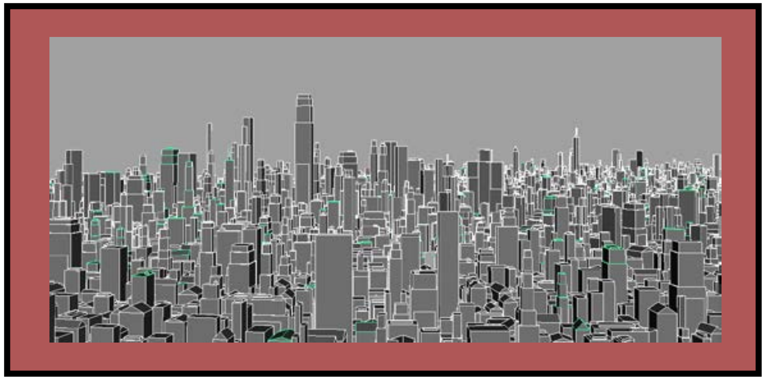


- Input:
  Lot polygons, age map and zone plan
- Output:
  Building strings with additional info

Procedural Modeling of Cities / Yoav Parish, Pascal Müller, Siggraph 2001

Module 4: Geometry and Facades





- Input:
  Strings and building type
- Output:
  City geometry and facade texture (procedural shader)

Procedural Modeling of Cities / Yoav Parish, Pascal Müller, Siggraph 2001

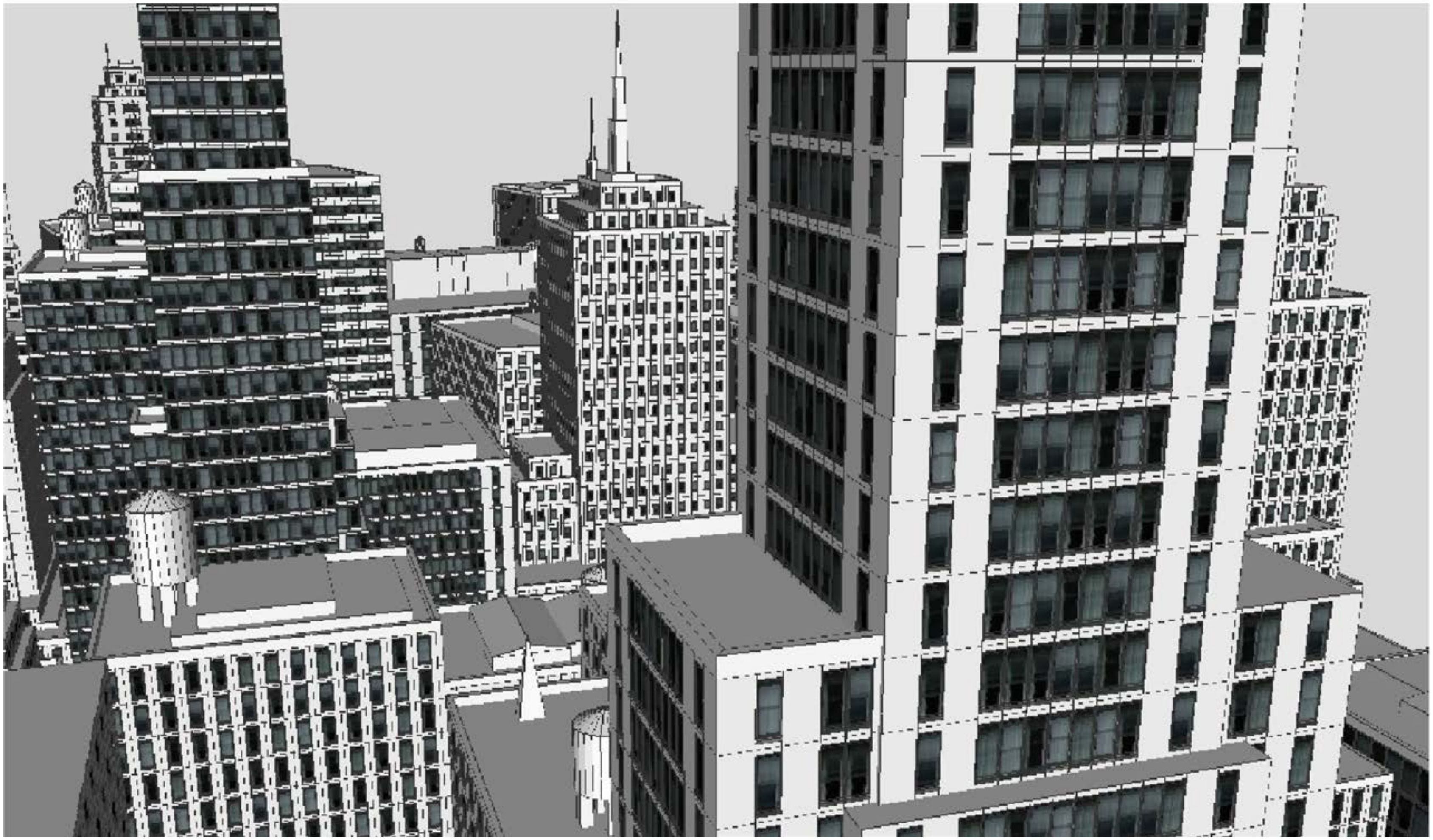
Procedural Modeling of Buildings

- Pompeii

Procedural Modeling of Buildings / Müller et al, Siggraph 2006

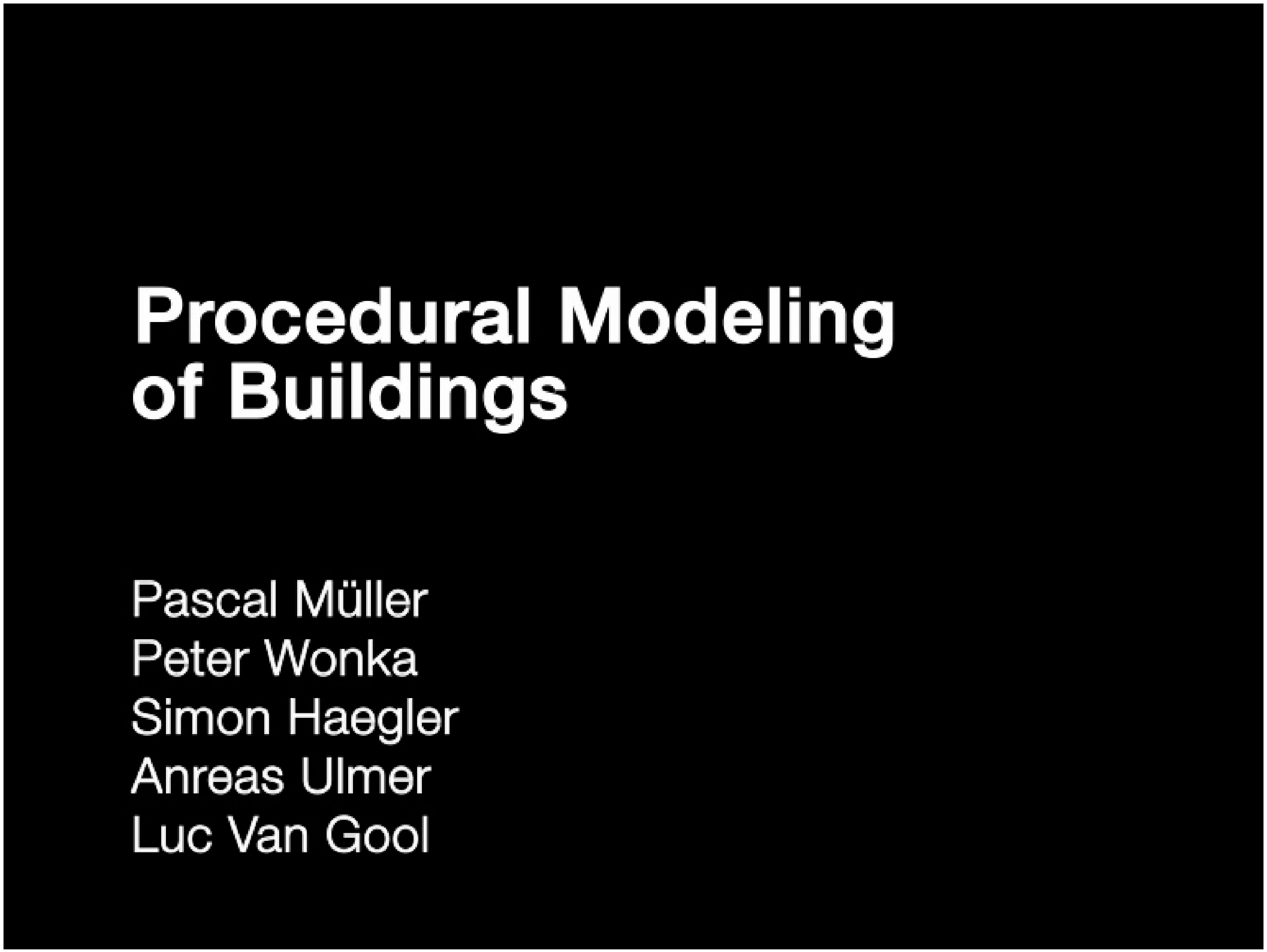
Procedural Modeling of Buildings

- Modern architecture



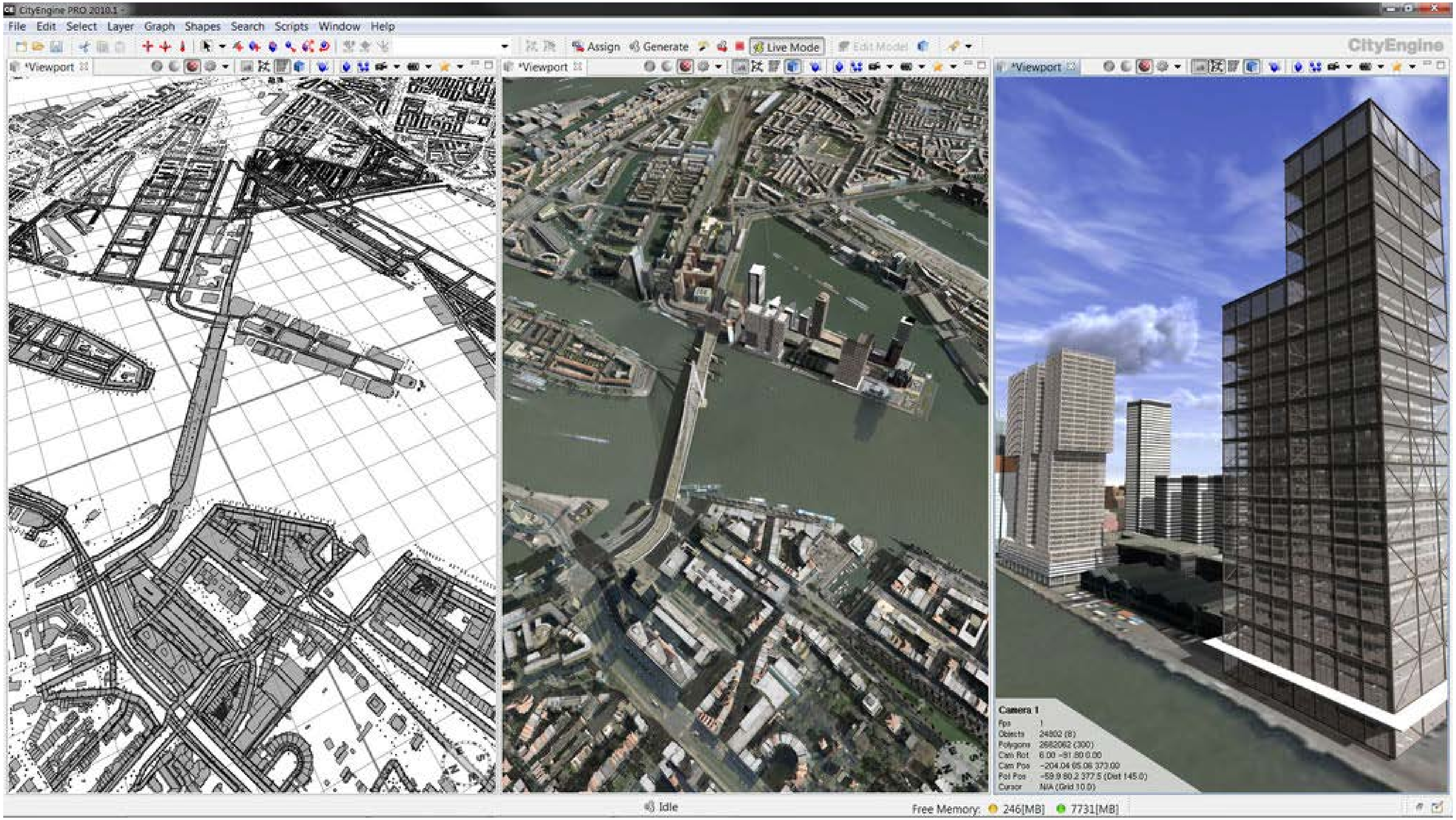
Procedural Modeling of Buildings / Müller et al, Siggraph 2006

Procedural Modeling of Buildings



Procedural Modeling of Buildings / Müller et al, Siggraph 2006

CityEngine



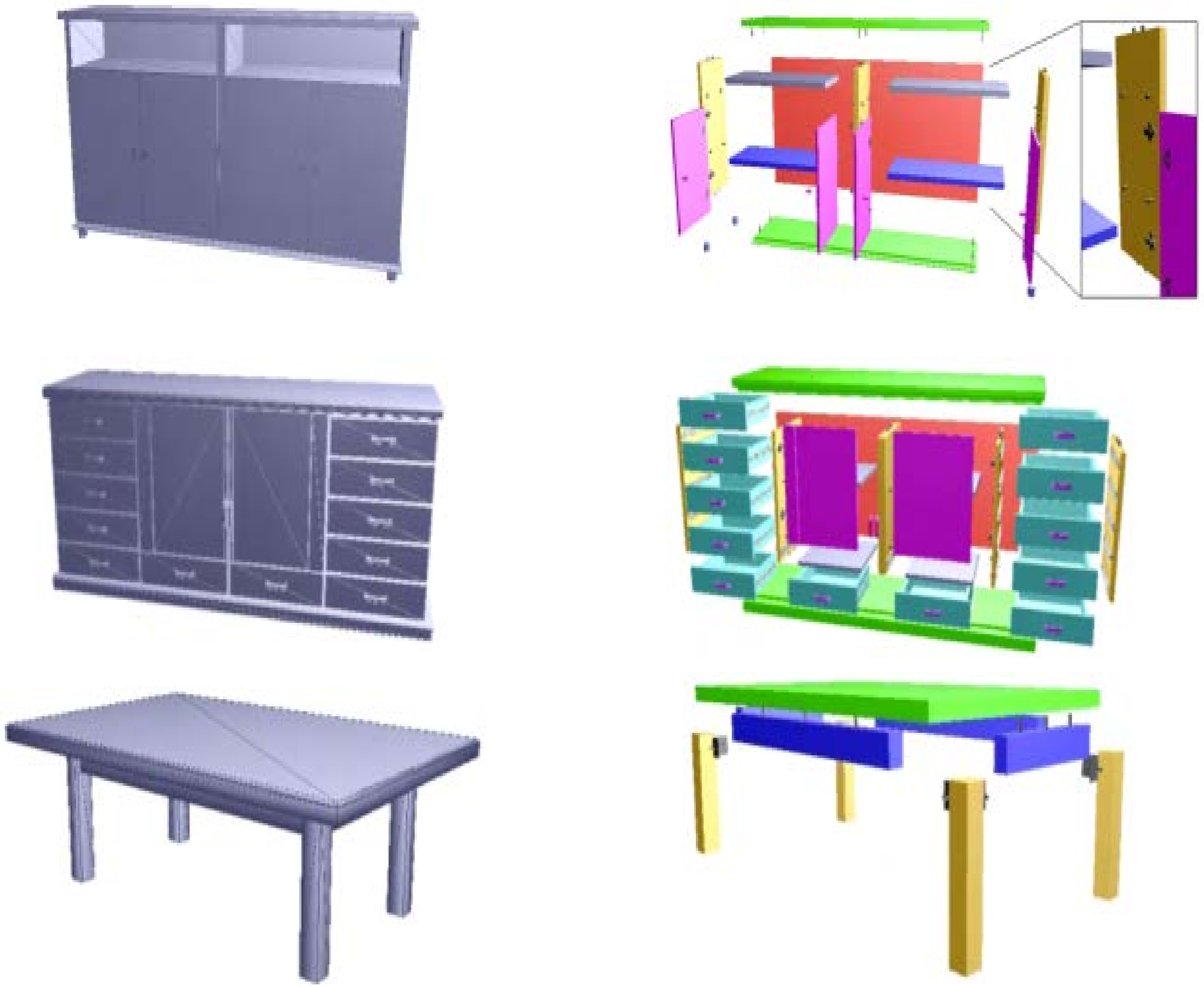
http://www.esri.com/software/cityengine/

CityEngine

http://www.youtube.com/watch?v=aFRqSJFp-I0

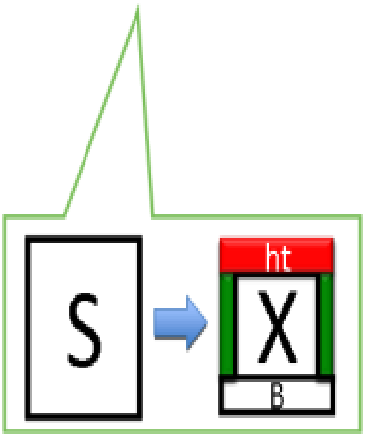
http://www.esri.com/software/cityengine/

Furniture Design using Formal Grammar

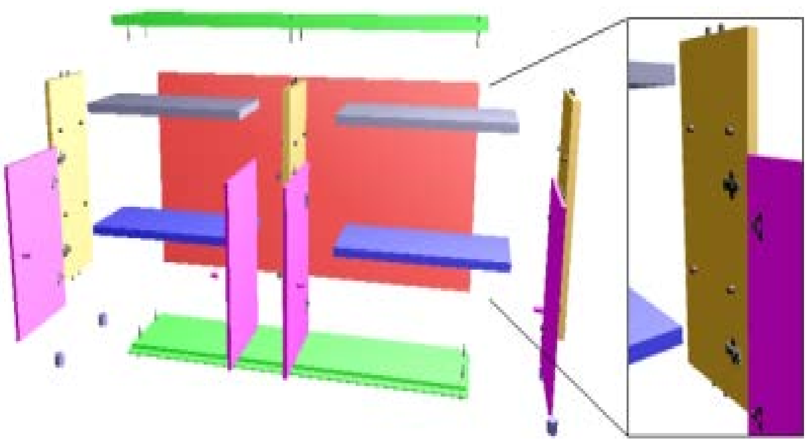


Converting 3D Furniture Models to Fabricable Parts and Connectors, Lau et al., Siggraph 2011

Furniture Design using Formal Grammar



Formal
grammar



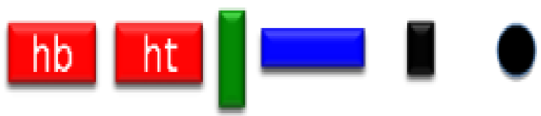
Separate parts
and
connectors

Formal Grammar for 2D Cabinets

$$N = \{\boxed{S}, \boxed{B}, \boxed{X}, \boxed{Y}\}$$

Non-terminal Symbols
- Collection of Parts

$$\Sigma = \{hb, ht, v, ha, leg, wheel\}$$



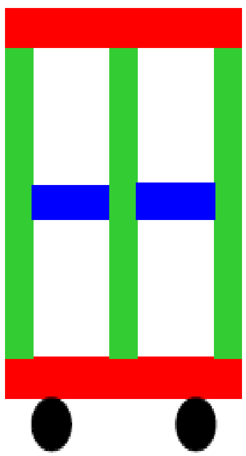
Terminal Symbols
- Separate Parts

P : Set of Production Rules

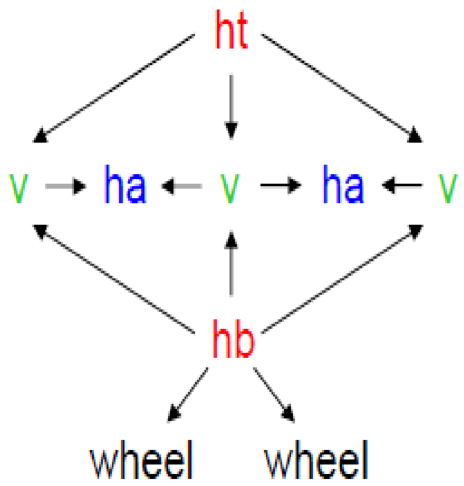
$\boxed{S}$: Start Symbol

The language specifies a directed graph,
and each graph represents parts and connectors

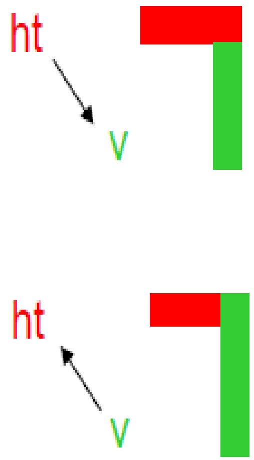
Representation of 2D Cabinets



Example 2D Cabinet



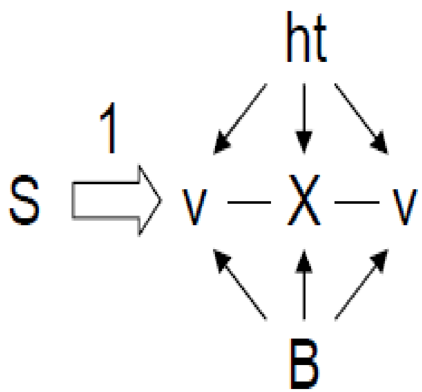
Corresponding Graph

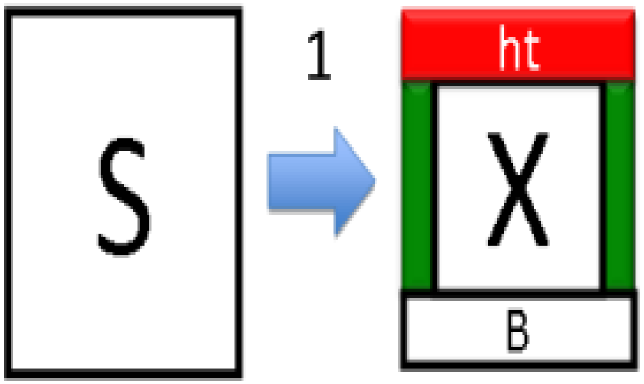


Positioning of Parts

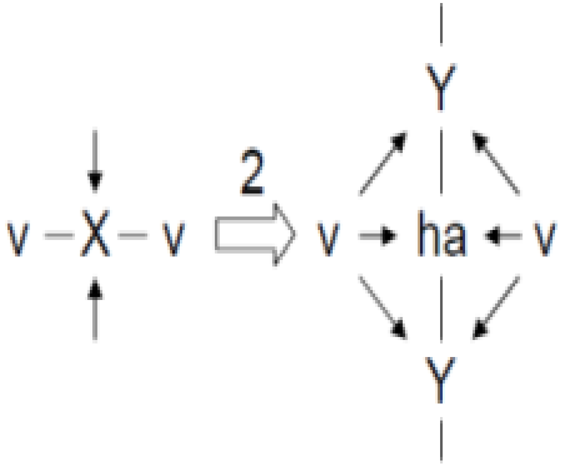
Examples of Production Rules

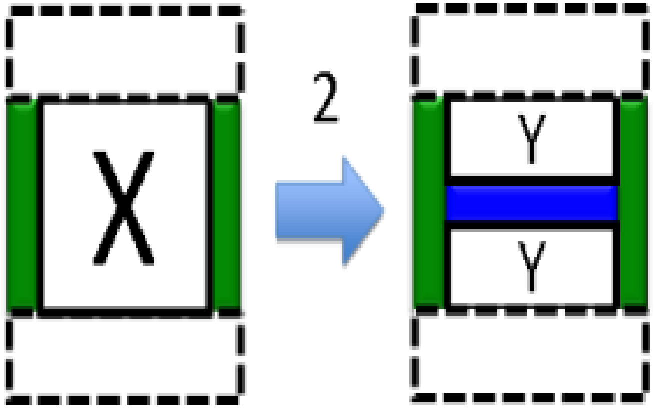
Production Rule 1





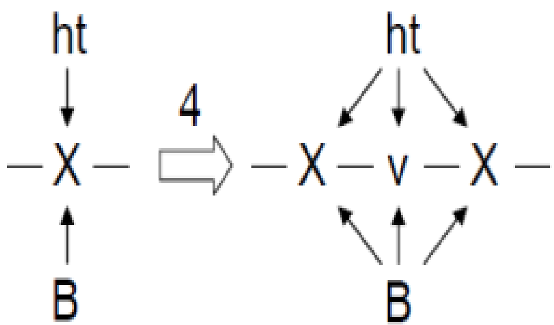
Production Rule 2

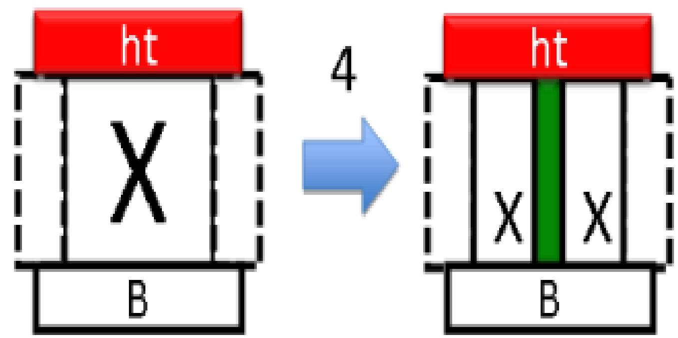




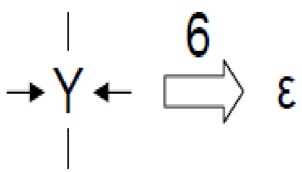
Examples of Production Rules

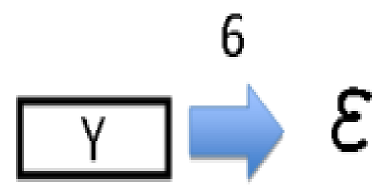
Production Rule 4



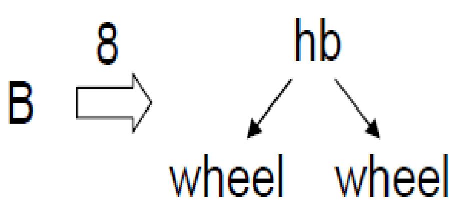


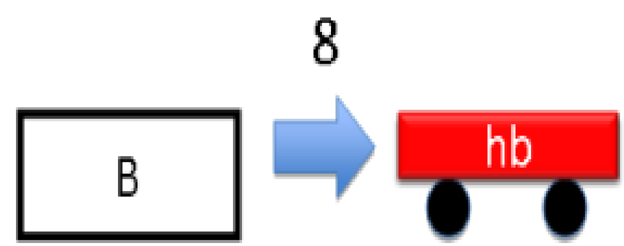
Production Rule 6



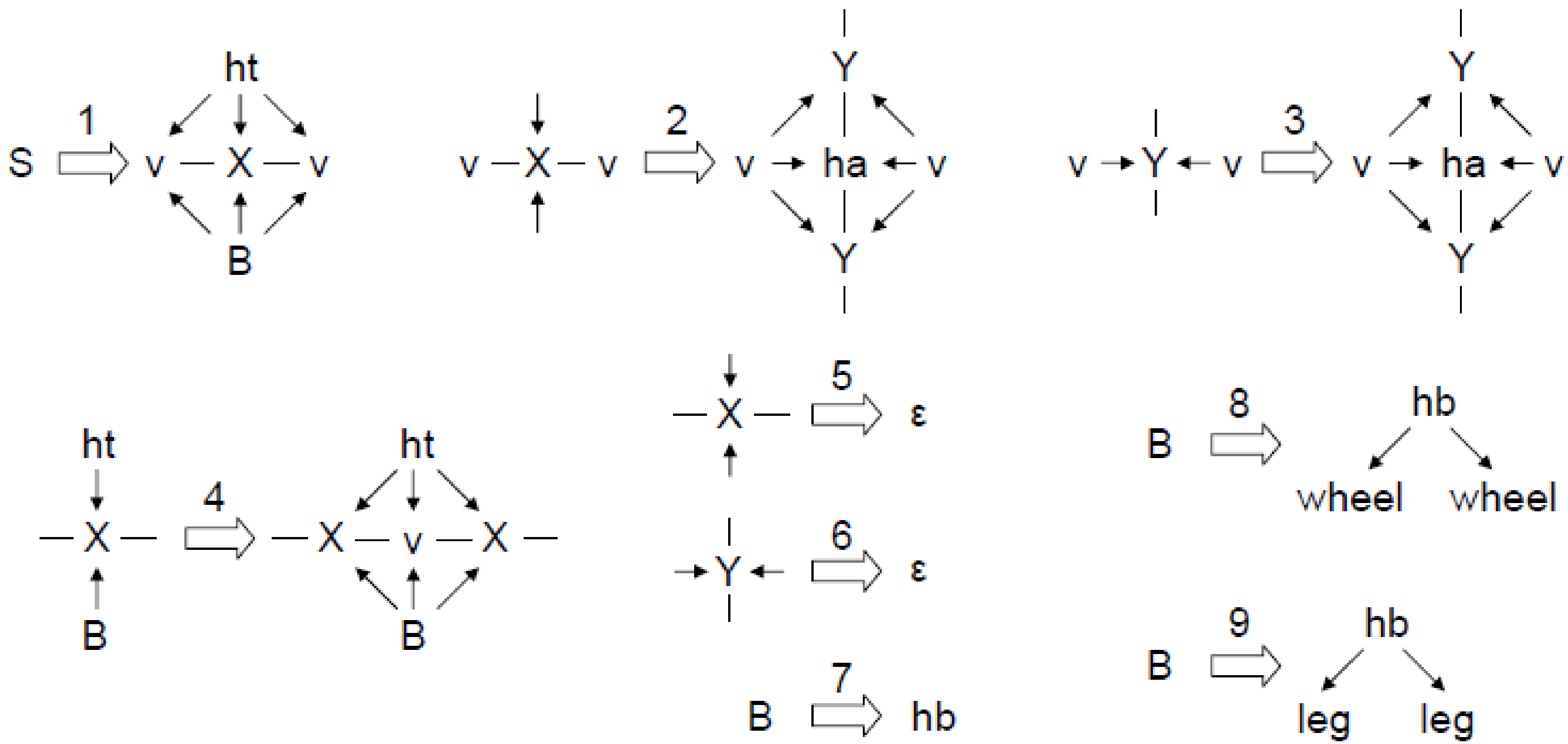


Production Rule 8

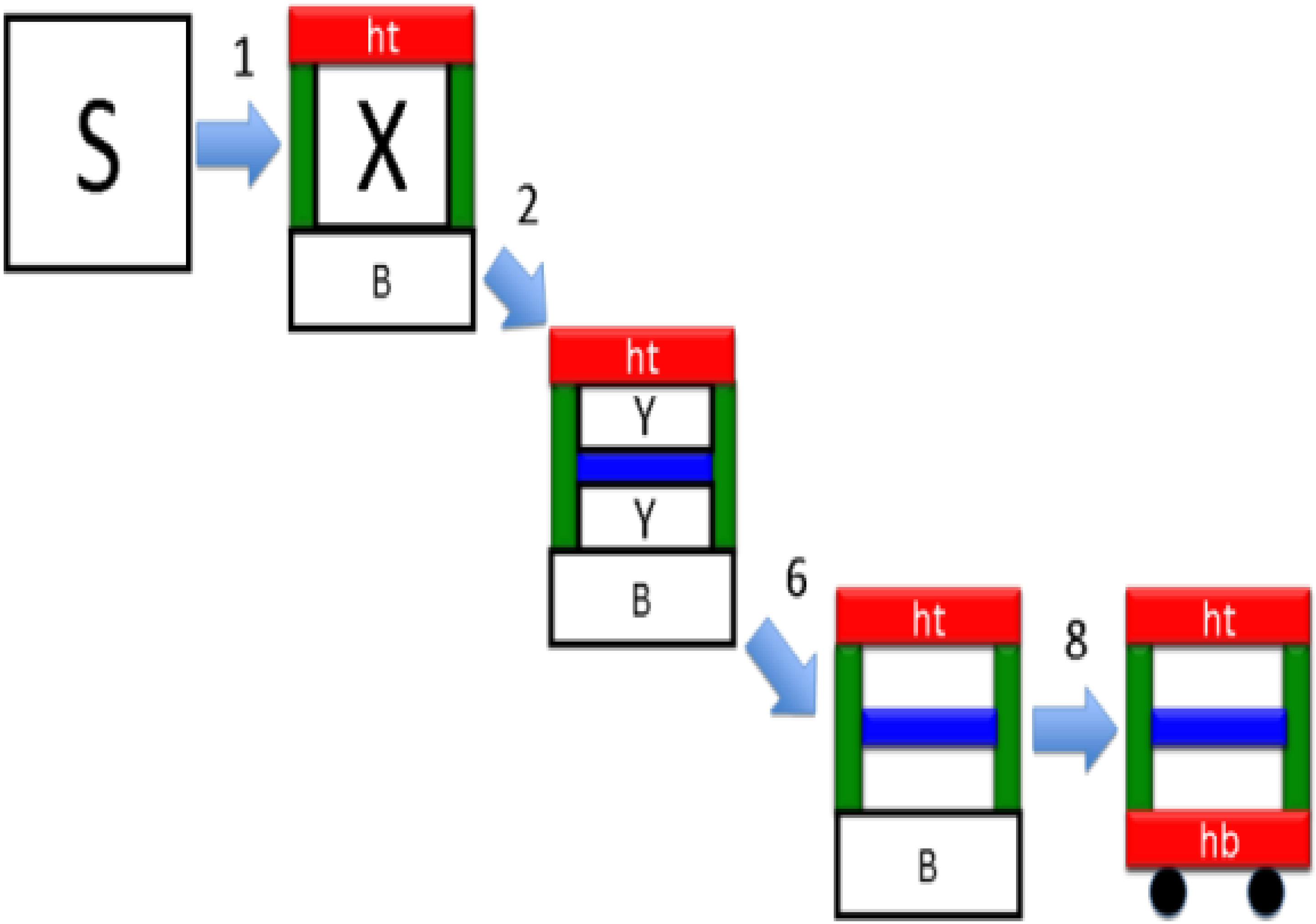




All Production Rules

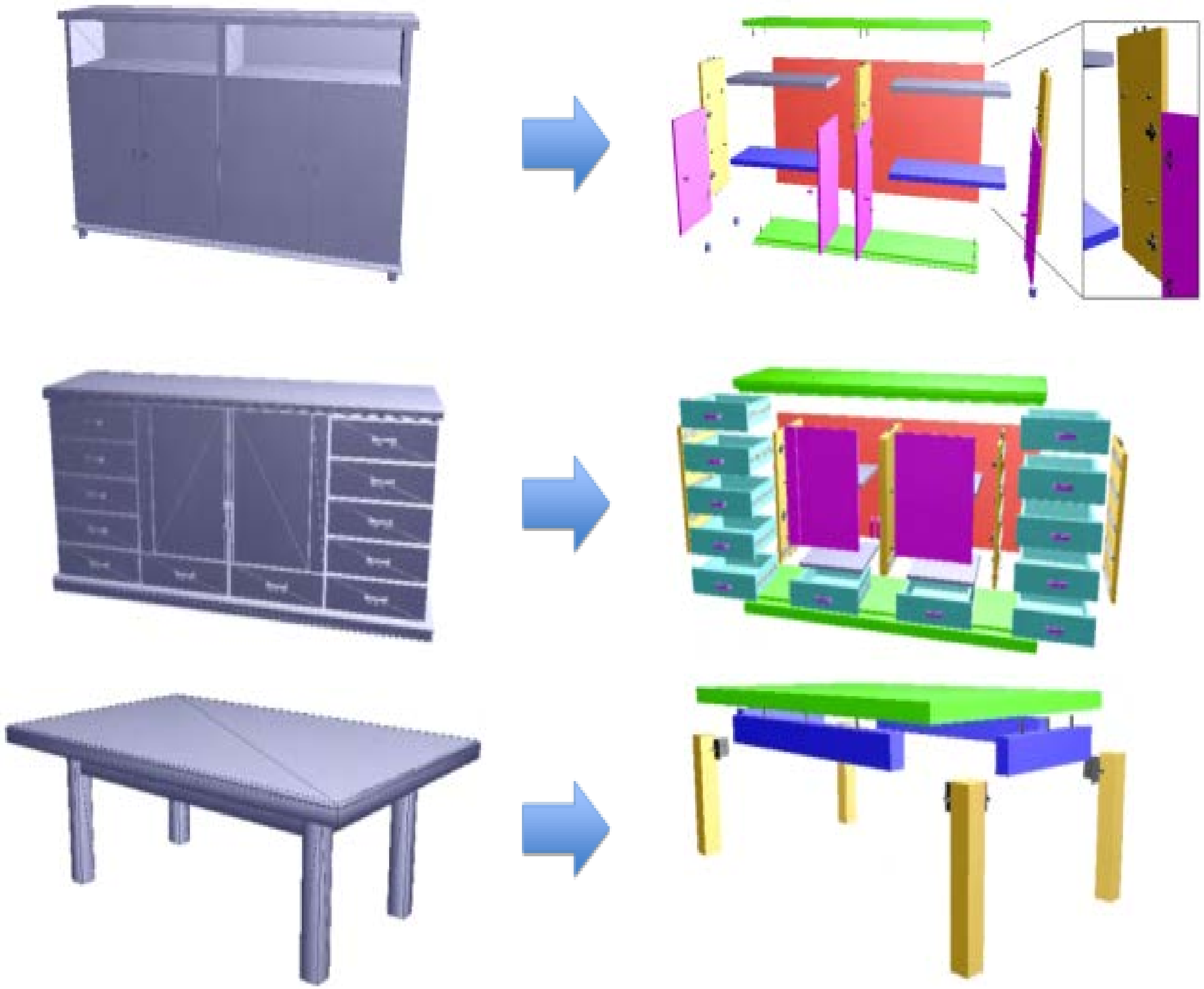


Sequence of Production Rules



Inverse Procedural Modelling

Input:
3D model



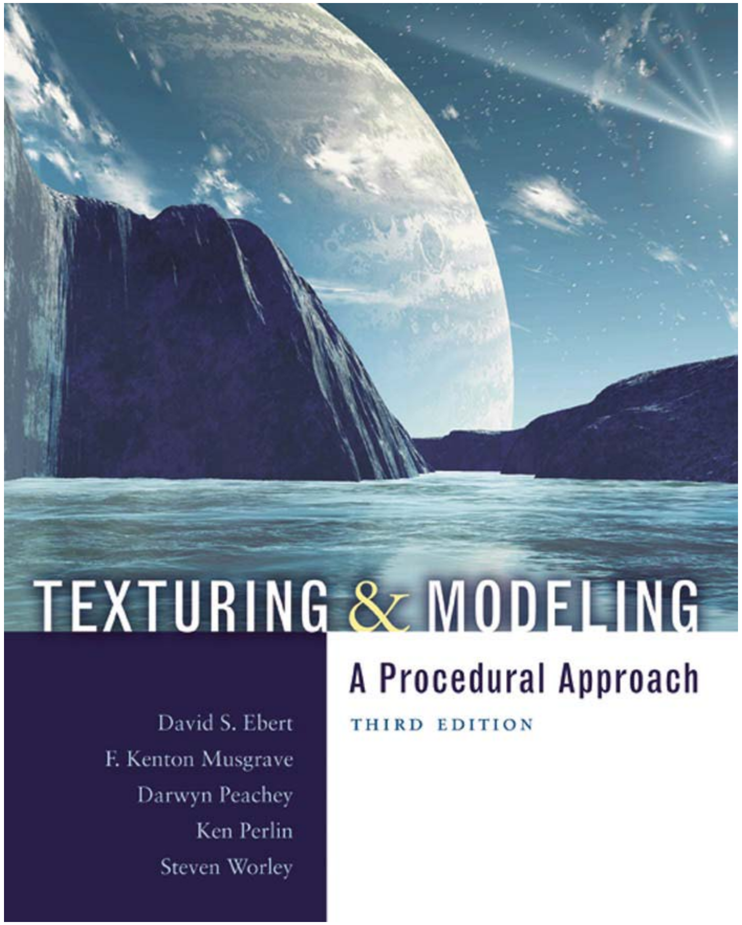
Output:
Fabricatable
Parts and
Connectors

Converting 3D Furniture Models to Fabricable Parts and Connectors, Lau et al., Siggraph 2011

Results: IKEA ALVE Cabinet

Further Reading in Procedural Techniques

- Texturing and Modeling - A Procedural Approach



That’s All For Today

- Readings:
  - The Algorithmic Beauty of Plants (Chapter 1)
    - *Chapter 1:Graphical modeling using L-systems*
    - *http://algorithmicbotany.org/papers/abop/abop-ch1.pdf*
  - *Procedural Modeling of Buildings / Müller et al, Siggraph 2006*
  - Converting 3D Furniture Models to Fabricable Parts and Connectors, Lau et al., Siggraph 2011