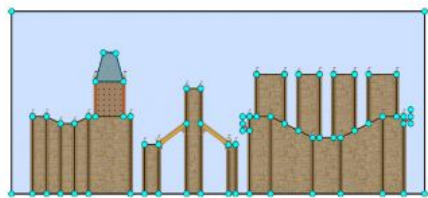
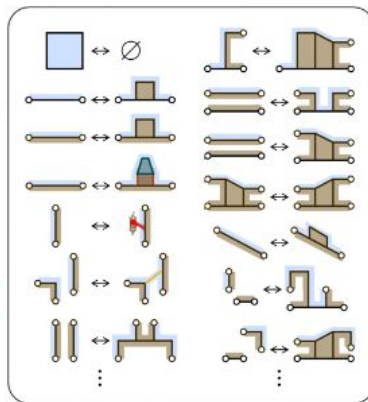


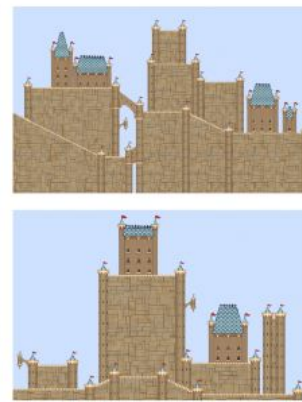
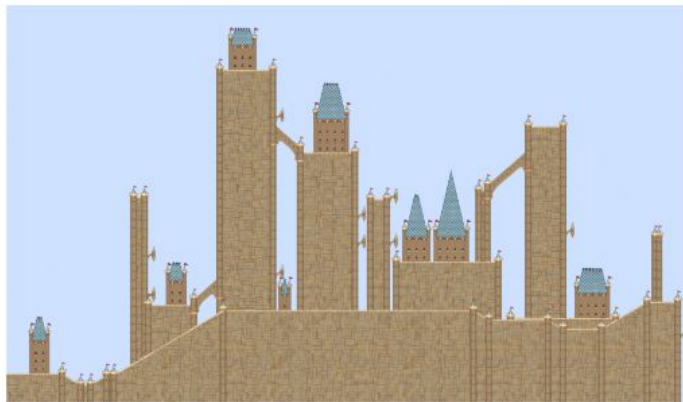
Example-Based Procedural Modeling Using Graph Grammars



(a) Example Shape



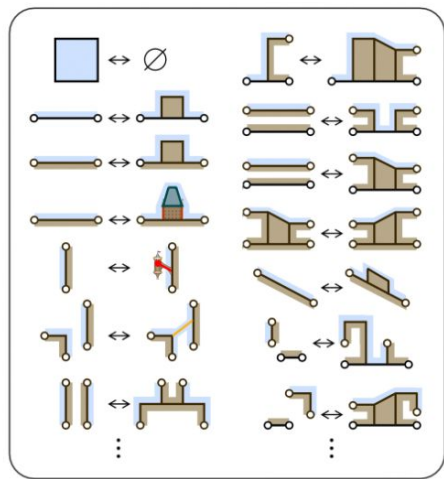
(b) Graph Grammar



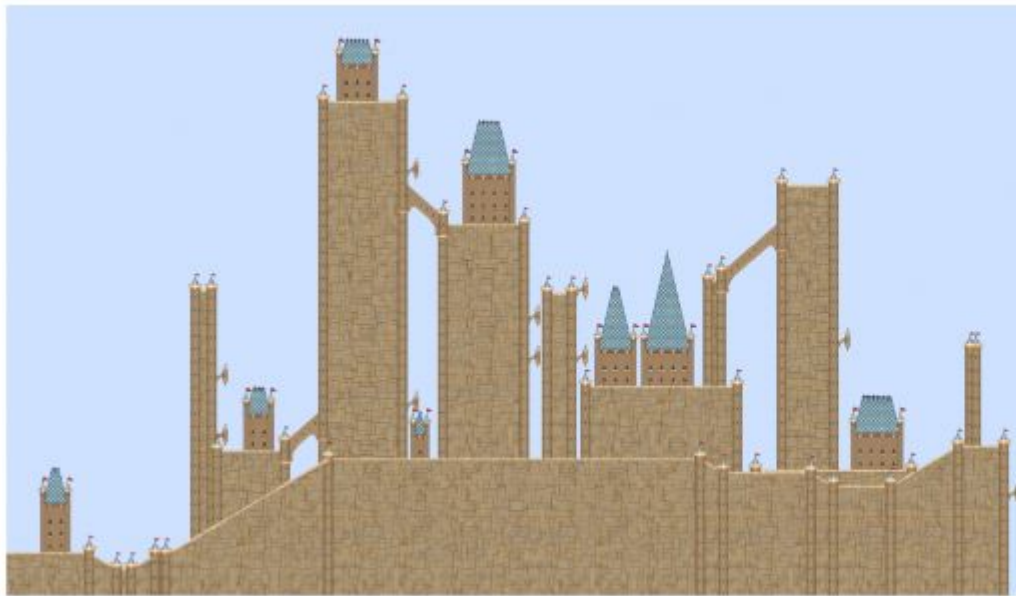
(c) Generated Shapes

Paul Merrel

Example-Based Procedural Modeling Using Graph Grammars



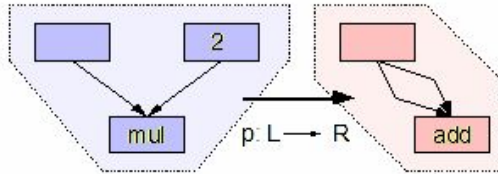
Grammar



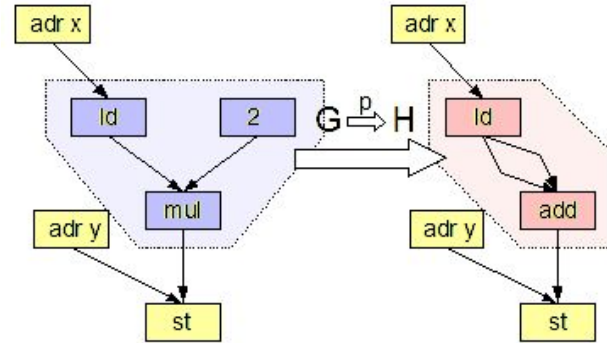
Generated scene

Example-Based Procedural Modeling Using **Graph Grammars**

↓
Set of rules to
modify a graph

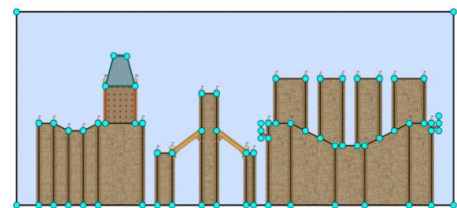


Rule

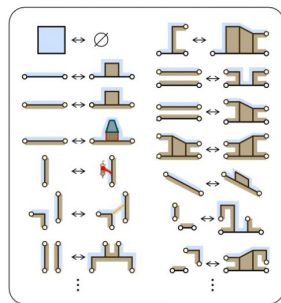
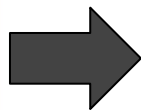


Graph modification

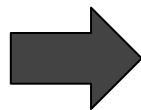
Example-Based Procedural Modeling Using Graph Grammars



Example

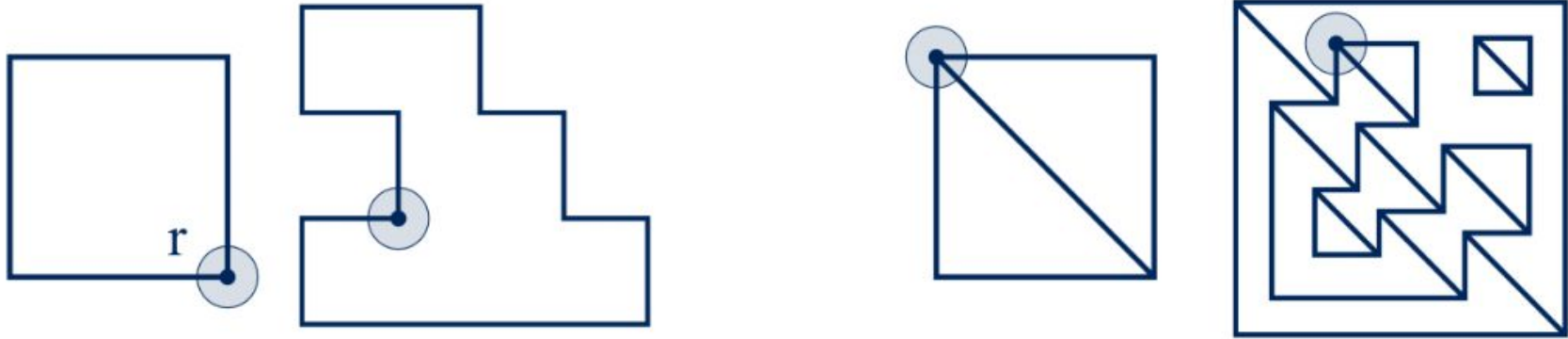


Grammar



Inverse Procedural
Modeling

Goal: local similarity



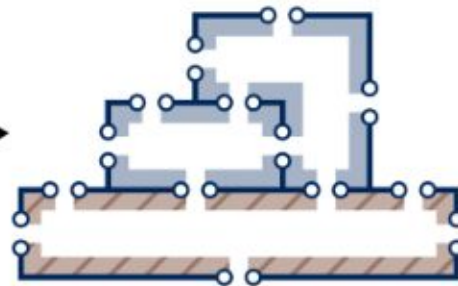
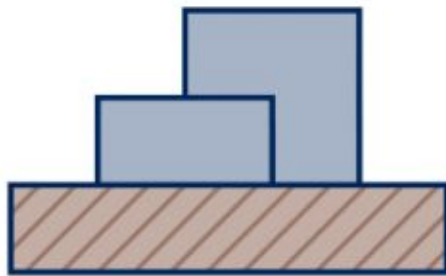
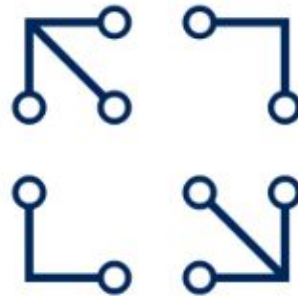
“every small region within the output must match a small region in the input”

Primitives

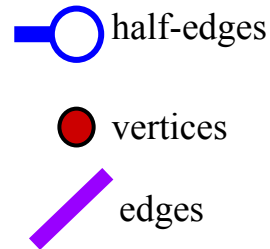
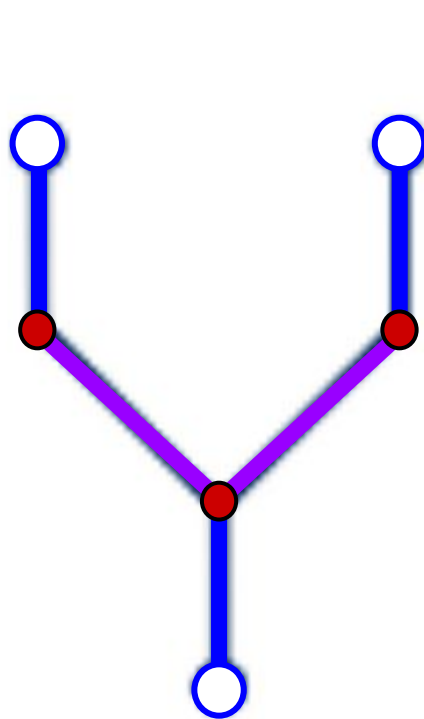
Example



Primitives

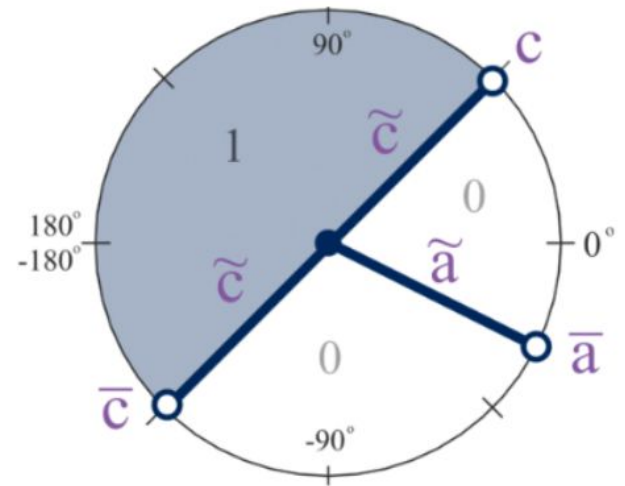


Angle Graph



(Half-)Edges labels

$$\tilde{a} = (0, 0, 150^\circ)$$
$$\tilde{c} = (1, 0, 45^\circ)$$



Primitives → Graph Hierarchy

Generation 0 - Edges



Generation 1 - Primitives



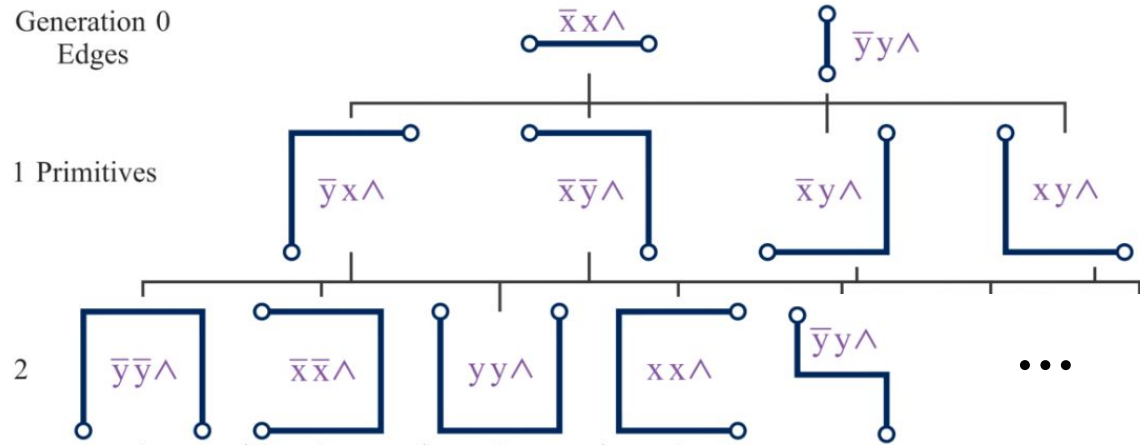
Generation 2



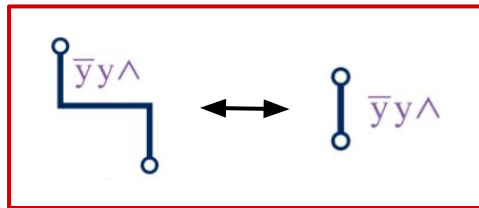
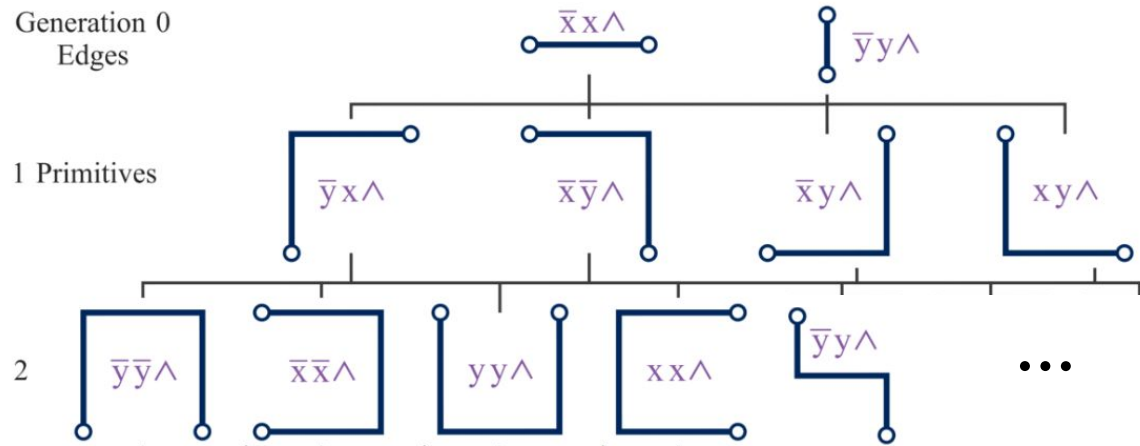
Gluings primitives

Gluings primitives

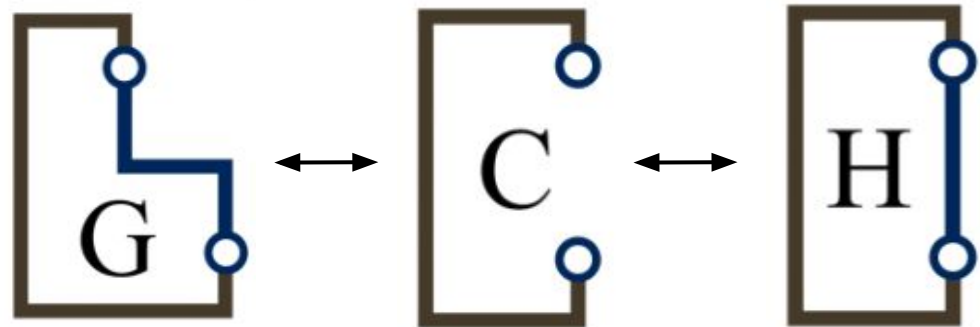
Building a concrete Graph Hierarchy



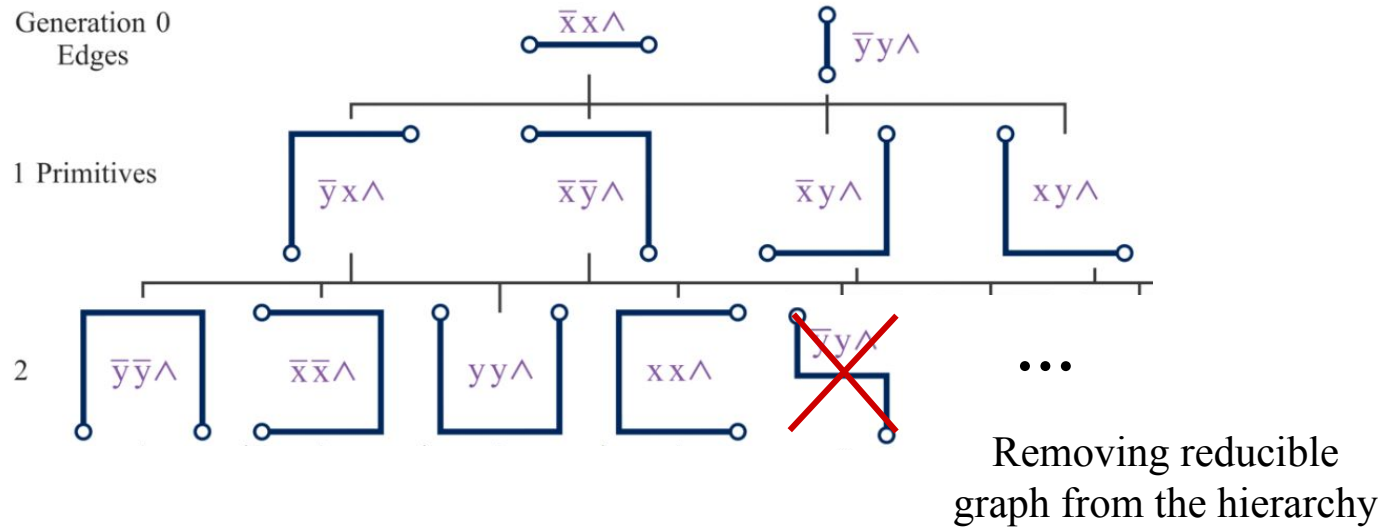
Building a concrete Graph Hierarchy



Rule creation

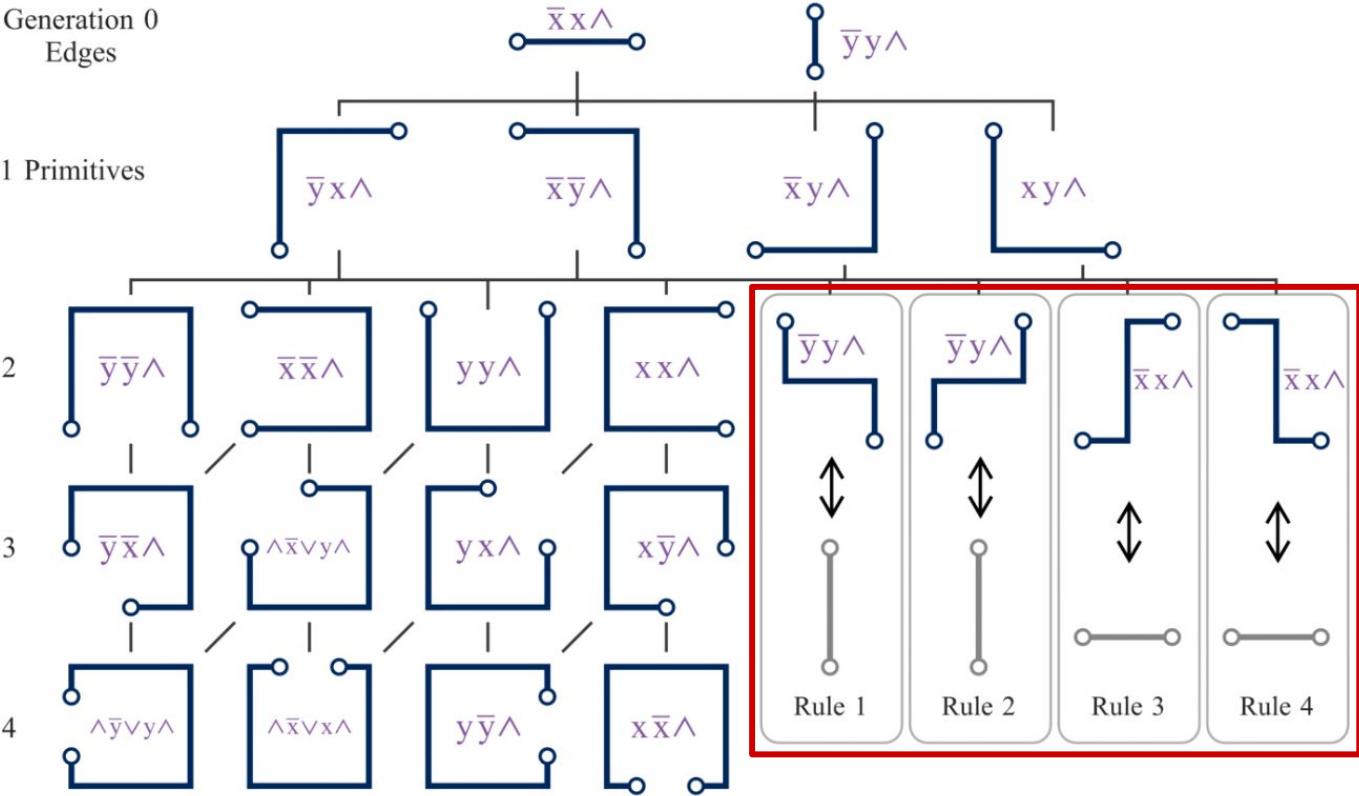


Building a concrete Graph Hierarchy

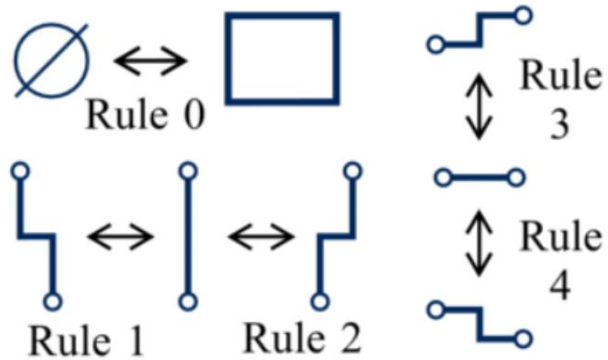


Adding rules in the graph grammar

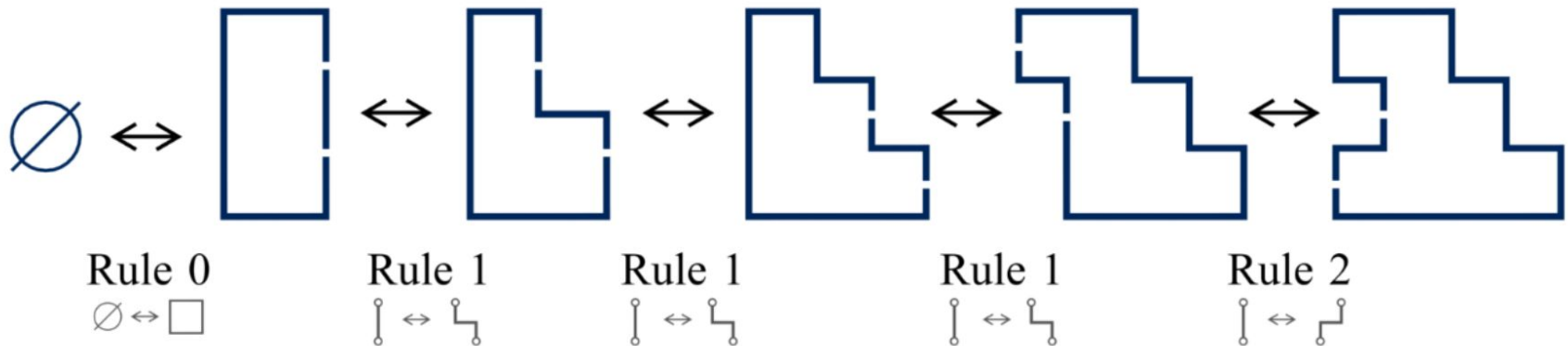
Building a concrete Graph Hierarchy



Using the Graph Grammar

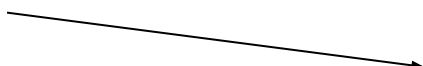


- Start from empty graph
- Use rules randomly, in any direction.

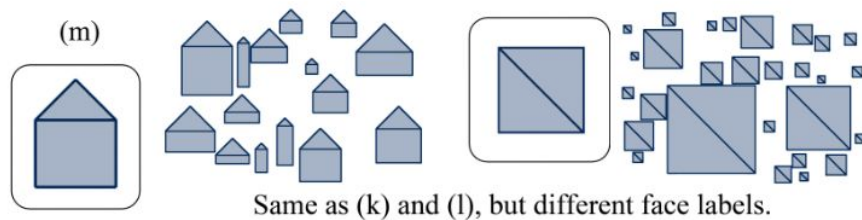
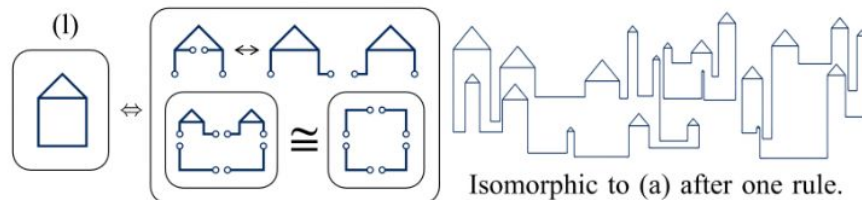
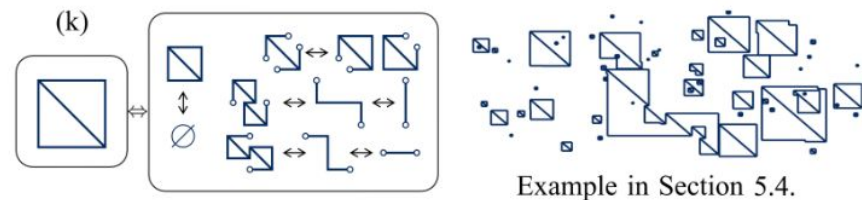
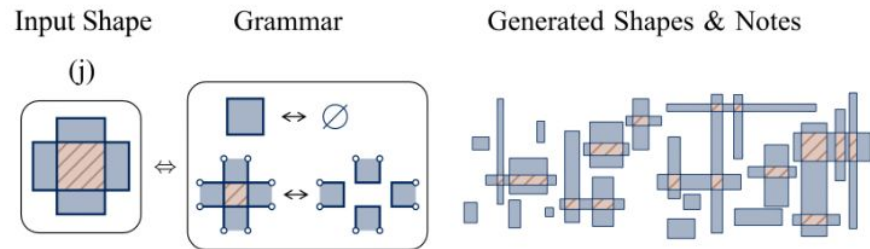
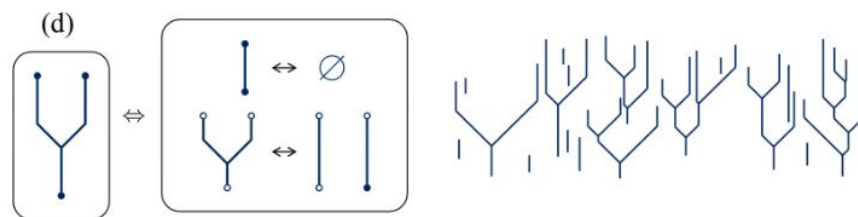
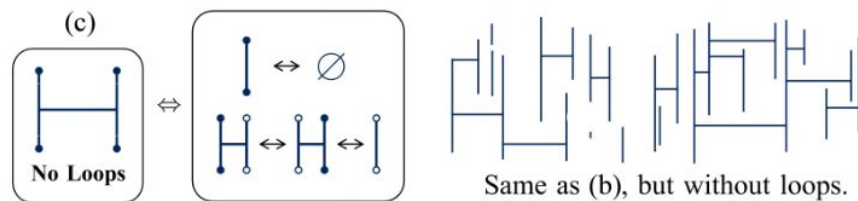
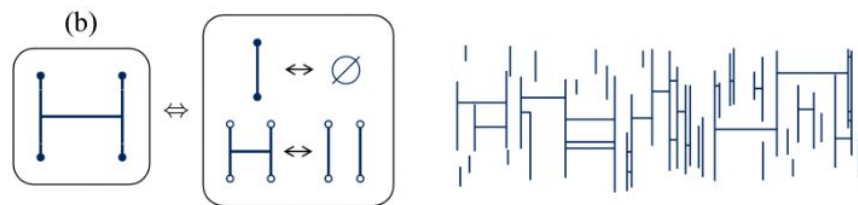
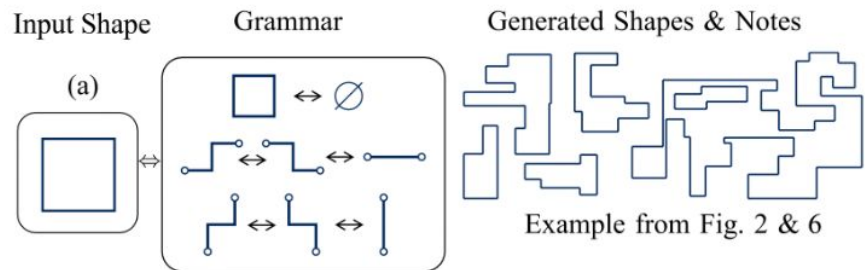


Graph Grammar $\rightarrow$ Angle Graph $\rightarrow$ Graph Drawing

Algorithm 3 Generate Graph Drawing From Grammar

- 1: Start with an empty graph $\emptyset$.
- 2: **for** $n = 1$ to N **do**
- 3: Propose a change to the graph using a rule.  Use a rule
- 4: **for** $i = 1$ to I and not accepted **do**
- 5: Find the space of consistent graph drawings: $\hat{\mathbf{x}} + \mathbf{K}_A \mathbf{\Lambda}$  Solve the vertices positions system
- 6: **for** $j = 1$ to J and not accepted **do**
- 7: Sample from the space $\hat{\mathbf{x}} + \mathbf{K}_A \mathbf{\Lambda}$  Choose a random solution
- 8: **if** sample is acceptable **then** accept new graph drawing.
- 9: Free a vertex. Allow its position to change.  If problem:
allow one more vertex to move

Results



More results ...

