Computational models
Programs describe the input to an algorithm

Example: graphviz

/* courtesy Ian Darwin and Geoff Collyer, Softquad Inc. */
digraph unix {
    size="6,6";
    node [color=lightblue2, style=filled];
    "5th Edition" -> "PWB 1.0";
    "6th Edition" -> "LSX";
    "6th Edition" -> "1 BSD";
    "6th Edition" -> "Mini Unix";
    "6th Edition" -> "Wollongong";
    "6th Edition" -> "Interdata";
    "Interdata" -> "Unix/TS 3.0";
    "Interdata" -> "PWB 2.0";
    "Interdata" -> "7th Edition";
    "7th Edition" -> "8th Edition";
Decision table

see also: Fowler, chapter 48

Programs describe choices (essentially, a bit-vector)

Example: pricing

13-inch MacBook Pro with Retina display

1. Choose Processor
   Which processor is right for you?
   
<table>
<thead>
<tr>
<th>Processor Type</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.7GHz Dual-core Intel Core i5, Turbo Boost up to 3.1GHz</td>
<td></td>
</tr>
<tr>
<td>2.9GHz Dual-core Intel Core i5, Turbo Boost up to 3.3GHz</td>
<td>+ $100.00</td>
</tr>
<tr>
<td>3.1GHz Dual-core Intel Core i7, Turbo Boost up to 3.4GHz</td>
<td>+ $300.00</td>
</tr>
</tbody>
</table>

2. Choose Memory
   How much memory is right for you?
   
<table>
<thead>
<tr>
<th>Memory Type</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>8GB 1866MHz LPDDR3 SDRAM</td>
<td></td>
</tr>
<tr>
<td>16GB 1866MHz LPDDR3 SDRAM</td>
<td>+ $200.00</td>
</tr>
</tbody>
</table>
Production rule
see also: Fowler, chapter 50

Programs describe conditions + actions
Example: IFTTT
State machine
see also: Fowler, chapter 51

Programs describe states + transitions

Example: Picobot’s syntax
Dependence graph
see also: Fowler, chapter 49

Programs describe a network of dependences (partial order)
Example: make

all: $(TARGETS)

stringtest:  $(STRINGTEST_OBJS)
            $(CXX) $(LDFLAGS) $(LIBS) $(CXXFLAGS) -o @ $(STRINGTEST_OBJS)

clean:
    rm -f $(TARGETS) $(ALL_OBJS)

string-wrapper.o: string-wrapper.cpp string-wrapper.hpp upointer.hpp \
                upointer-private.hpp
stringtest.o: stringtest.cpp string-wrapper.hpp upointer.hpp \
            upointer-private.hpp
Constraint satisfaction

Programs describe (properties of) the solution

Example: SQL

```sql
from Employee e
left join fetch e.department
where e.salary > :limit
```
Computational models

Data interpretation: programs describe input
e.g., graphviz

Decision table: programs describe choices
e.g., pricing

Production rule: programs describe conditions & actions
e.g., IFTTT

State machine: programs describe states & transitions
e.g., Picobot syntax

Dependence graph: programs describe a partial order
e.g., make

Constraint satisfaction: programs describe the solution
e.g., SQL