HW/SW Codesign

Exercise 1: StateCharts

27.09.2017

Stefan Draskovic
Stefan.draskovic@tik.ee.ethz.ch
-- slides by Mirela Botezatu --
StateCharts

• Specification model
  • Suitable for complex systems, since they can model **hierarchy** and **concurrency**
  • Transitions can be guarded by conditions
  • Transitions can be associated with actions
Basic States and Super-States

- **Basic States**: states not composed of other states
- **Super-States**: states containing other states

**S**: Super-State

**E**: Sub-State of S

**E**: Basic State

**S**: Ancestor State of E

Diagram:

- States: A, B, C, D, E, S, Z
- Edges: g, h, i, j, k, m, f
Hierarchy: OR-Super-States

- **OR-Super-State:**
  - a Super-State A in which *exactly one* of its (immediate) sub-states (X, Y, Z) is active whenever A is active

**sub-states are related by “exclusive-or”**
Concurrency: AND-Super-States

- **AND-Super-State:**
  - A Super-State A in which *all* its (immediate) sub-states (B, C) are active whenever A is active

sub-states are related by "and"
Tree Representation

B: OR-Super-State

A: AND-Super-State

C: Basic State

- Super-States have at least one children
- Basic States have no children
Set of States

• Basic States
  • State

• OR-Super-States
  • Union of children

• AND-Super-States
  • Cartesian product of children

(computed traversing the tree from leaves to root)
Edge Labels

- **Events**: exist only until next model evaluation
- **Conditions**: refer to values of variables that keep their value until they are reassigned
- **Actions**: assignments to variables or creations of events
Edge Labels: Three-phases evaluation

1. **Evaluate** effect of *external changes* on events and conditions

2. **Compute** set of transitions to be made in the *current step* and *right hand sides* of assignments

3. **Activate** transitions, assign new values to variables
In the Exercise, *external events* are assumed not to occur contemporarily.

**b**: Event  
[D2]: Condition  
**c**: Action  
(event generation)
Exercise 1.1.d

- Compute the set of states

- **Hint**: expand the final formula: it will be useful for Exercise 1.1.f, since it will list all possible states for Super-State A

- **Example**:
  \[(Q_1 \cup Q_2) \times (Q_3 \cup Q_4) = (Q_1, Q_3) \cup (Q_1, Q_4) \cup (Q_2, Q_3) \cup (Q_2, Q_4)\]
Exercise 1.1.e

- A sequence of events $a, b, e, b, d, b$ is applied: what is the sequence of states that are passed through?

<table>
<thead>
<tr>
<th>Event</th>
<th>State B</th>
<th>State C</th>
<th>State A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial</td>
<td>1</td>
<td>G</td>
<td>1,G</td>
</tr>
<tr>
<td>$a$</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$b$</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$e$</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$b$</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$d$</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$b$</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Exercise 1.1.f

• Convert the StateChart to a FSM

• Hints:
  1. start from drawing all the possible states for Super-State A (computed in exercise 1.1.d)
  2. Draw all possible transitions
  3. Remove possible unnecessary states
Different notation:

**Action:** ok

**Trigger:** coin_in

is equivalent to:

**coin_in/ok**
Exercise 1.1.c: Solution

A: AND-Super-State

B, C, D: OR-Super-States
Exercise 1.1.d: Solution

\[ Z_A = Z_B \times Z_C \]
\[ = (Z_1 \cup Z_2) \times (Z_G \cup Z_D) \]
\[ = (Z_1 \cup Z_2) \times (Z_G \cup (Z_{D_1} \cup Z_{D_2})) \]
\[ = (Z_1, Z_G) \cup (Z_1, Z_{D_1}) \cup (Z_1, Z_{D_2}) \cup (Z_2, Z_G) \cup (Z_2, Z_{D_1}) \cup (Z_2, Z_{D_2}) \]
Assumption used in the solution:
External events can only be detected after a stable state has been reached (see lecture slide 2-46)

1. Event a activates edge G->D (i.e., G->D1) in Super-State C
2. Edge G->D creates event c, that in turn activates edge 1->2 in Super-State B, which has no actions associated
3. Transitions G->D1 and 1->2 are activated, so that the global transition for Super-State A is (1,G)->(2,D1)
Exercise 1.1.e: Solution

No variables in this StateChart!
Exercise 1.1.f: Solution

From Exercise 1.1.d:

\[ Z_A = Z_B \times Z_C \]
\[ = (Z_1 \cup Z_2) \times (Z_G \cup Z_D) \]
\[ = (Z_1 \cup Z_2) \times (Z_G \cup (Z_{D1} \cup Z_{D2})) \]
\[ = (Z_1, Z_G) \cup (Z_1, Z_{D1}) \cup (Z_1, Z_{D2}) \cup (Z_2, Z_G) \cup (Z_2, Z_{D1}) \cup (Z_2, Z_{D2}) \]
Exercise 1.1.f: Solution

1,G 

1,D1 

1,D2 

2,G 

2,D1 

2,D2

Diagram with nodes and connections.
Exercise 1.1.f: Solution
Exercise 1.1.f: Solution
Exercise 1.1.f: Solution
Exercise 1.1.f: Solution
Exercise 1.1.f: Solution
Exercise 1.1.f: Solution
Exercise 1.1.f: Solution
Exercise 1.1.f: Solution
Exercise 1.1.f: Solution
Exercise 1.1.f: Solution

Final FSM (non minimized)
State (1,D1) is not reachable!
Exercise 1.1.f: Solution

Final FSM (minimized)
Exercise 1.1.g: Solution

"honest" trace

Two steps of the same evaluation

\[ A_1.0 \xrightarrow{\text{coin-in/ok}} A_1.1 \]
\[ A_2.A \xrightarrow{\text{ok/}} A_2.B \]
\[ A_2.B \xrightarrow{\text{req-tea/start-tea}} A_2.D \]
\[ A_2.D \xrightarrow{\text{drink-ready/done}} A_2.A \]
\[ A_1.1 \xrightarrow{\text{done/}} A_1.0 \]

"cheater" trace

Two steps of the same evaluation

\[ A_1.0 \xrightarrow{\text{coin-in/ok}} A_1.1 \]
\[ A_2.A \xrightarrow{\text{ok/}} A_2.B \]
\[ A_2.B \xrightarrow{\text{req-tea/start-tea}} A_2.D \]
\[ A_2.D \xrightarrow{\text{cancel/coin-out, reset}} A_1.0 \]
\[ A_1.1 \xrightarrow{\text{drink-ready/done}} A_2.A \]
Exercise 1.1.g: Solution

2: New State

block: New Event