Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Institut für
Technische Informatik und
Kommunikationsnetze

Prof. L. Thiele

Hardware/Software Codesign - HS 15

Exercise Sheet 1

Issue Date: 16. September 2015
Discussion Date: 23. September 2015

1.1 StateCharts

The lecture has introduced Harel's *StateChart* formalism. StateCharts are a popular specification model for embedded systems.

1.1.a) Advantages of StateCharts

What are the most important extensions of the StateChart model in comparison to an ordinary finite state machine (FSM)?

StateCharts can model hierarchy and concurrency. Transitions can be guarded (conditionally enabled). Furthermore, transitions can be associated with actions. Actions can perform computations on variables, as well as generate new events.

1.1.b) Disadvantages of StateCharts

What are the disadvantages of the StateChart formalism?

Although StateCharts scale better than ordinary FSMs, they grow in size for large systems and tend to be hard to understand. There is only limited potential for re-use. Actions associated with transitions provide a powerful extension, but on the other hand, the extensive use of actions moves parts of the system state information from the states themselves to the variables. This hidden state makes system analysis difficult.

1.1.c) Tree of states for StateChart

Given the StateChart in Figure 1. Draw the state space of the StateChart as a tree, which shows the hierarchy

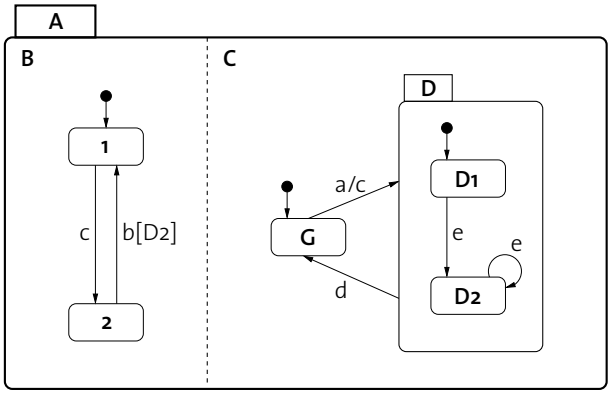

Figure 1: StateChart

of states and denotes the state types (basic state, sequential states, and parallel states).

The state space is shown in Figure 2.

1.1.d) Formal computation of state space

How would you formally compute the set of states? Compute the set of states for the hierarchical automata which is defined by the StateChart from Fig. 1.

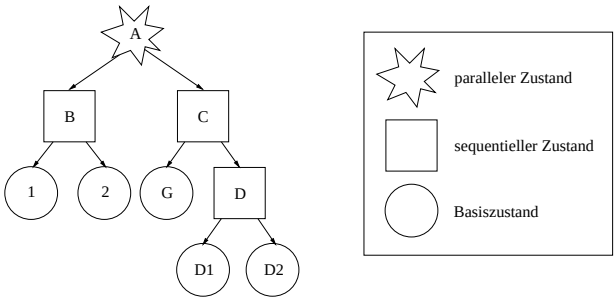

Figure 2: State tree: hierarchy and type of states

$$
\begin{aligned}
Z_A &= Z_B \times Z_C \\
&= (Z_1 \cup Z_2) \times (Z_G \cup Z_D) \\
&= (Z_1 \cup Z_2) \times (Z_G \cup (Z_{D1} \cup Z_{D2})) \\
&= (Z_1, Z_G) \cup (Z_1, Z_{D1}) \cup (Z_1, Z_{D2}) \cup (Z_2, Z_G) \cup (Z_2, Z_{D1}) \cup (Z_2, Z_{D2})
\end{aligned}
$$

1.1.e) Analysis

The automaton defined by the StateChart from Fig. 1 passes through a number of states, when external events are applied. Show the sequence of state that are passed through, starting from the initial state, for the following sequence of events: a,b,e,b,d,b. Use a table notation.

Event	State B	State C	State A
Initial	1	G	1,G
a	2	D1	2,D1
b	2	D1	2,D1
e	2	D2	2,D2
b	1	D2	1,D2
d	1	G	1,G
b	1	G	1,G

1.1.f) Conversion of StateChart to a finite state machine (FSM)

Draw a finite state machine which is equivalent to the StateChart from Fig. 1. Minimize the number of states.

Figure 3 show an equivalent FSM that is not minimized. Since state (1,D) is unreachable, it can be removed, and the minimized FSM in Figure 4 results.

1.1.g) StateChart model of a vending machine

The StateChart model of a simplified vending machine is shown in Figure 5.

- Describe the trace of transitions occurring when the user inserts a coin and orders a tea.
- The control of the vending machine has a bug that allows the user to cheat. Describe the trace of transitions that illustrate the bug.
- Draw the corresponding StateChart that fixes the bug.
- *Optional:* Extend the StateChart such that it accepts 0.05, 0.10, 0.20, and 0.50 coins. Coffee costs 0.75 and tea costs 0.50.

- $A_1.0 \xrightarrow{\text{coin\_in/ok}} A_1.1$
  $A_2.A \xrightarrow{\text{ok/}} A_2.B$
  $A_2.B \xrightarrow{\text{req\_tea/start\_tea}} A_2.D$
  $A_2.D \xrightarrow{\text{drink\_ready/done}} A_2.A$
  $A_1.1 \xrightarrow{\text{done/}} A_1.0$

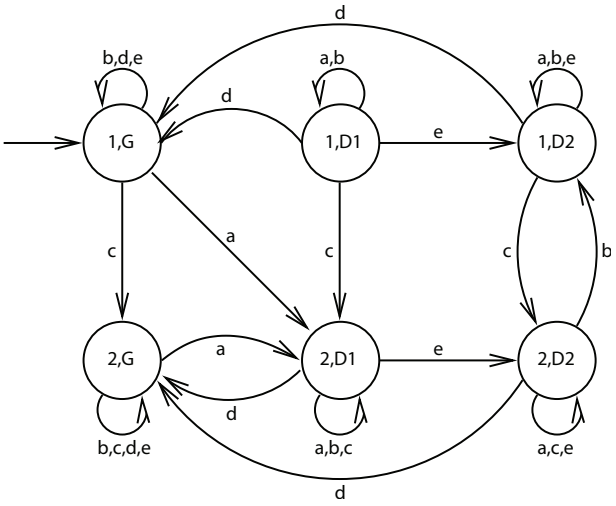

Figure 3: Equivalent FSM (not minimized)

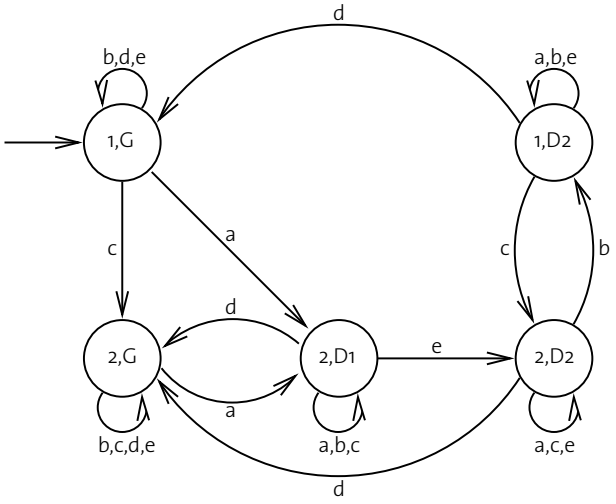

Figure 4: Equivalent FSM (minimized)

- $A_1.0 \xrightarrow{\text{coin\_in/ok}} A_1.1$
  $A_2.A \xrightarrow{\text{ok/}} A_2.B$
  $A_2.B \xrightarrow{\text{req\_tea/start\_tea}} A_2.D$
  $A_1.1 \xrightarrow{\text{cancel/coin\_out, reset}} A_1.0$
  $A_2.D \xrightarrow{\text{drink\_ready/done}} A_2.A$
- One possible solution is shown in Figure 6.

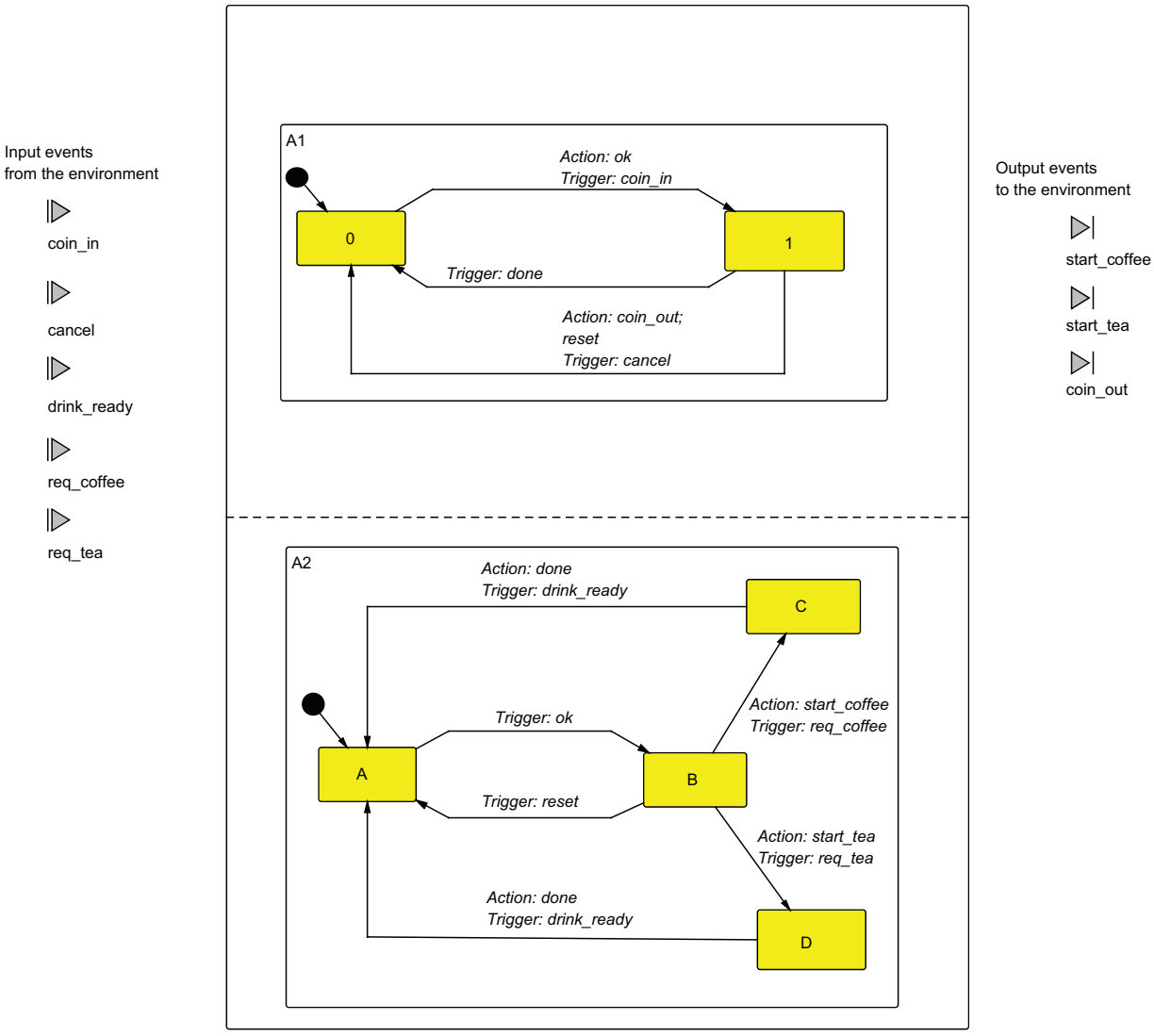

Figure 5: Vending Machine

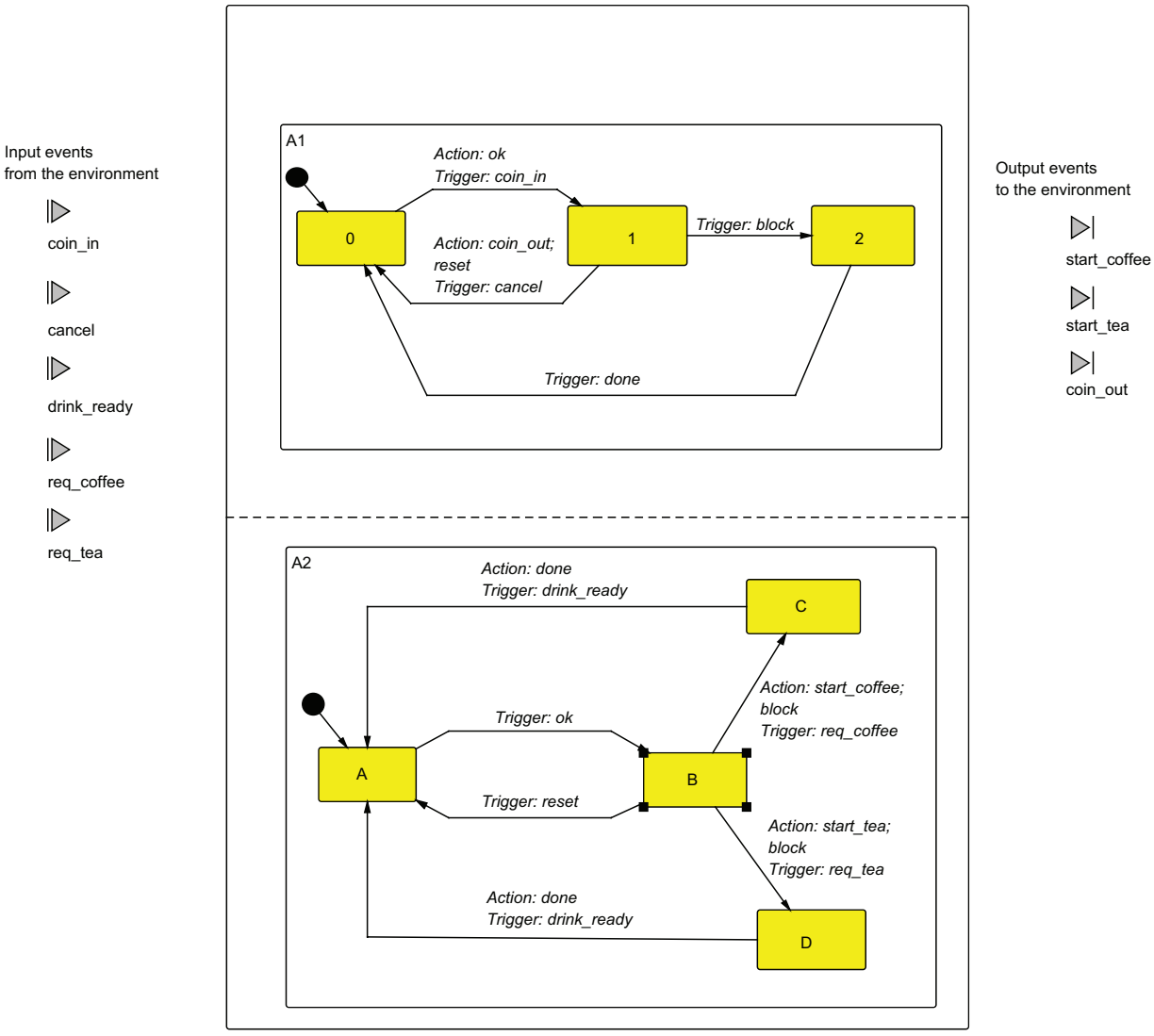

Figure 6: Fixed Vending Machine