Prof. Bernd Finkbeiner
Dr. Calogero Zarba
Moritz Hahn

Embedded Systems

Problem 1: A vending machine

Embedded systems are frequently specified using high level design tools based on automata theory. Figure 1 shows the control of a simple vending machine composed of two finite automata A_1 and A_2.
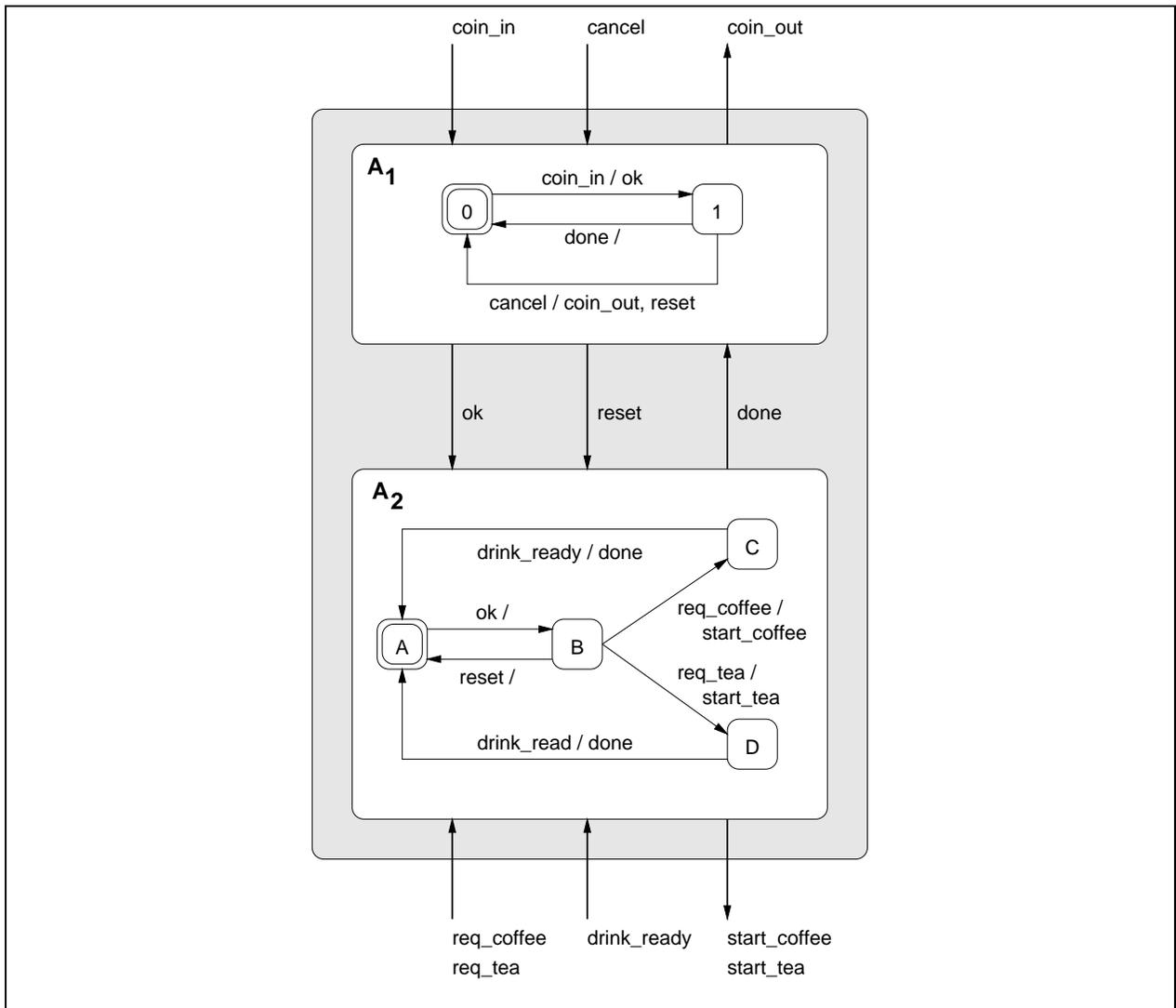

Fig. 1: A vending machine.

The automata communicate between each other and with the environment by consuming and generating *events*. Figure 2 lists all occuring events together with their meaning.

Event	Generated by	Consumed by	Meaning
COIN_IN	environment	A_1	user inserts coin
CANCEL	environment	A_1	user presses cancel-button
REQ_COFFEE	environment	A_2	user presses coffee-button
REQ_TEA	environment	A_2	user presses tea-button
DRINK_READY	environment	A_2	drink is ready
COIN_OUT	A_1	environment	coin returned to user
START_COFFEE	A_2	environment	start preparation of coffee
START_TEA	A_2	environment	start preparation of tea
OK	A_1	A_2	enough coins inserted
RESET	A_1	A_2	coins back to user
DONE	A_2	A_1	drink delivered

Fig. 2: Events for the vending machine in Figure 2.

A transition written as A/B, can be taken only when the event A is observed. When the transition is taken, the event A is consumed and the event B is generated. A transition written as $A/$ does not generate events when it is taken.

A typical interaction of the vending machine with the environment is:

– Initially the automata are in states $\boxed{0}$ and $\boxed{A}$.
– The user inserts a coin, the environment generates the event COIN_IN, A_1 moves to state $\boxed{1}$, and the event OK is generated.
– A_2 consumes the event OK and moves to state $\boxed{B}$.
– The used press the cancel-button, A_1 moves back to state $\boxed{0}$, the events RESET and COIN_OUT are generated.
– A_2 consumes the RESET event and moves back to state $\boxed{A}$.

(a) Describe the trace of transitions occurring when the user inserts a coin and orders tea.
(b) The control of the vending machine has a bug that allows the user to cheat. Find it.
(c) Construct an equivalent automaton Q where no parallelism is involved. The initial state should be $\boxed{0A}$. When the event COIN_IN occurs, Q moves to state $\boxed{1A}$ and the event OK is generated. This causes Q to move from state $\boxed{1A}$ to state $\boxed{1B}$. Now continue yourself.
(d) Fix the bug.
(e) Allow the vending machine to accept coins for €0,05, €0,10, €0,20, and €0,50. Coffee costs €0,75. Tea costs €0,50.

Problem 2: Statecharts

Construct a statecharts equivalent to the one in Figure 3, but that uses no AND-superstates.

Problem 3: Statecharts

Construct a statecharts equivalent to the one in Figure 4, but that uses no superstates and no history.
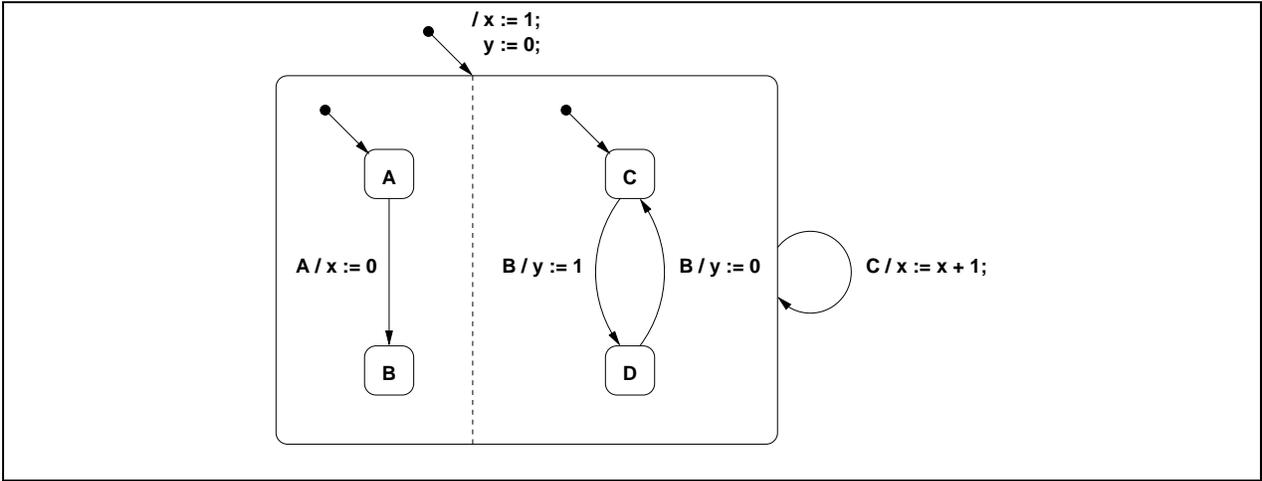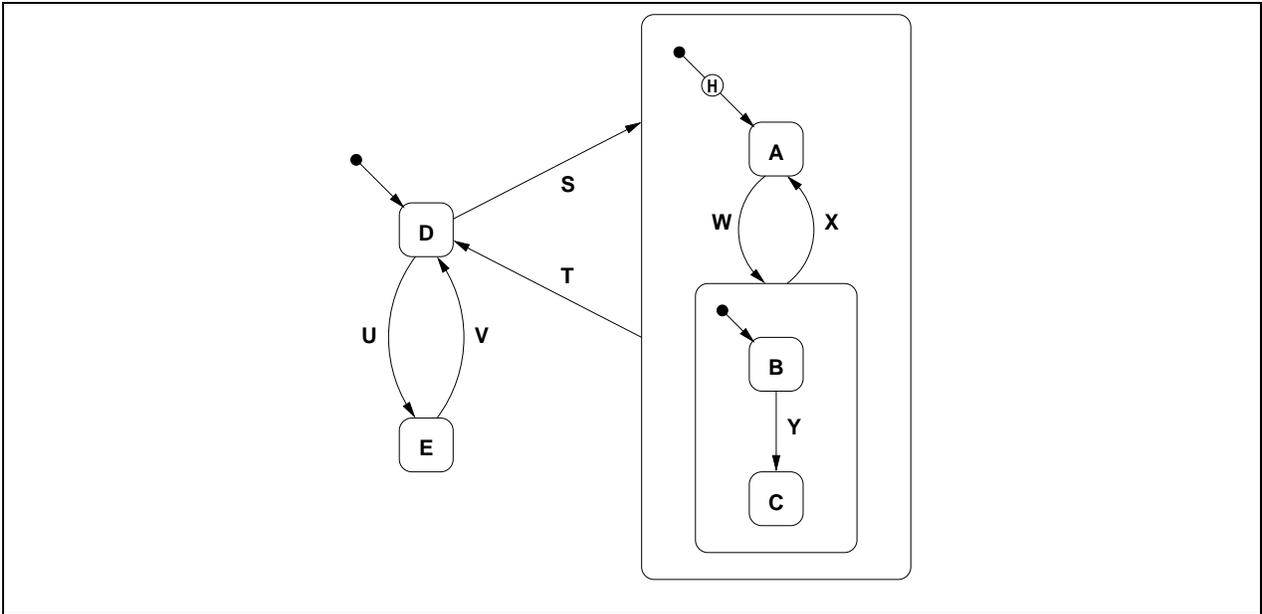
Fig. 3: A state chart with AND-superstates.

Fig. 4: A statechart with hierarchy and history.