Prof. Bernd Finkbeiner, Ph.D.
Dipl.-Inf. Rüdiger Ehlers
Markus Rabe, M.Sc.
Sebastian Hahn, B.Sc.

Summer term 2012
Problem Set 2
Due: Wednesday, 2nd May 2012

Embedded Systems

Please indicate your **name**, **matr. number**, **email address**, and which **discussion session** you have been allocated to. We encourage you to collaborate in **groups** of up to **three** students. Only one submission per group is necessary.

Exercise 1: Modeling with StateCharts

Figure 1 shows the control of a simple vending machine (in the StateCharts formalism). Figure 2 lists all occurring events together with their meaning. A typical interaction of the vending machine with the environment is:

- Initially the system is in the 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 user presses 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 coffee.

(b) The control of the vending machine has a bug that allows the user to cheat. Find it.

(c) Fix the bug.

(d) Now, 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.

Exercise 2: Timed StateCharts

Consider the AND-state in Figure 3 that models a system with two concurrent processes `P1` and `P2` accessing a shared resource. The StateChart comprises the global variable `id` that is initially set to 0, and the external events `try1`, `try2`, `set1`, `set2`, `retry1`, `retry2`, `enter1`, `enter2`, `exit1`, and `exit2`. Furthermore, the timing behavior is parameterized by the integer

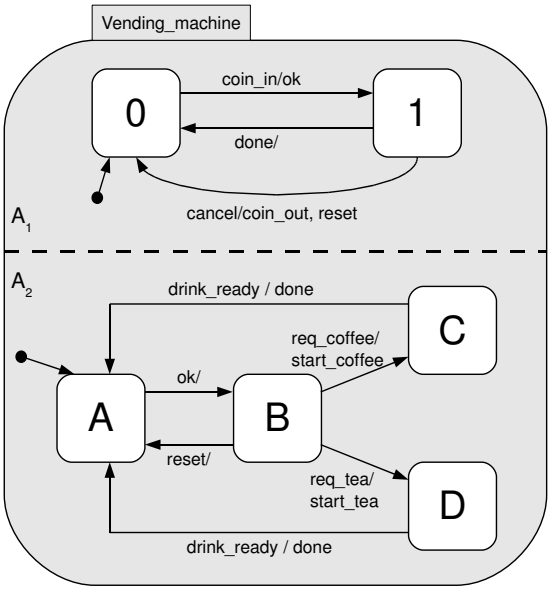

Figure 1: A vending machine.

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

Figure 2: Events for the vending machine in Figure 1.

constants `D` and `T`. The system is considered as *safe* if it is never the case that `P1` is in `crit1` and `P2` is in `crit2` at the same time.

In case of conflicting update assignments we assume an *interleaving semantics*. For instance, if the assignments id:=1 and id:=2 are to be executed concurrently, then the semantics nondeterministically determines whether (1) first id:=1 then id:=2 is executed, or (2) first id:=2 then id:=1 is executed.

(a) Assume `D` = 10 and `T` = 5. Is the system safe? Justify your answer either by giving an argument why it is safe, or by providing a (short) trace (i.e., a scenario of delays and events) leading to an unsafe state, where `P1` is in `crit1` and `P2` is in `crit2` at the same time.

(b) Give the most general characterization how `D` and `T` must be chosen such that for *any* occurring of events, the system will always remain safe.

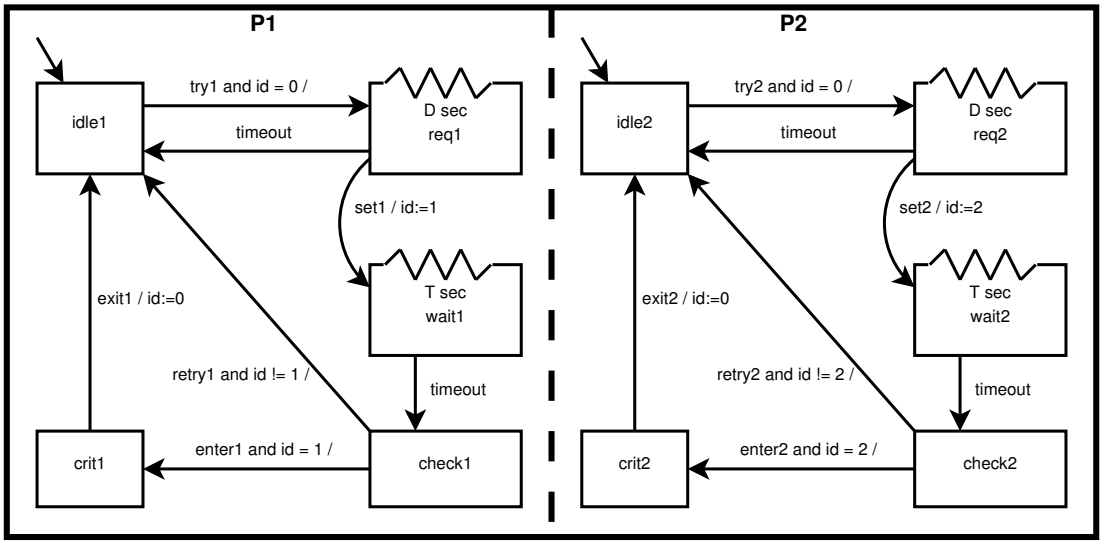

Figure 3: A timed mutual exclusion protocol.

Exercise 3: MATLAB / Simulink

Download the Simulink model of the damped harmonic oscillator from the course web page.
(a) Let

$$y_s = \lim_{t\to\infty} y(t);$$
$$t_s(d) = \inf\{t \in \mathbb{R}_0^+ : \ \forall t' \geq t.\ |y(t') - y_s| \leq d\}.$$

Approximate y_s and $t_s(0.2)$ with a precision of 1 (by simulation) for the parameters $k = 10$, $m = 1.2$, $y_0 = 15$, and $R = 0.1$.

Hint: You can increase the precision of your simulation when you select under *Simulation* $\rightarrow$ *Configuration Parameters* a *fixed-step* solver and decrease the *Fixed-step size*.

(b) Extend the model such that the suspension $u(t)$ varies with a 0.5Hz cosine with an amplitude of 1. Use the following differential equation:

$$\ddot{y}(t) = -\frac{1}{m}\left(k\left(y(t) - \frac{1}{k}u\left(\frac{t}{4}\right)\right) + R\dot{y}(t)\right)$$

In your submission, please provide a print out or a drawing of your Simulink model. State the parameters of all changed or newly added function blocks.