2nd Seminar – State-Based Modelling

1 Traffic light

We are designing a controller mechanism for a traffic light.

a. Create a state space for a simple red–yellow–green traffic light that is detailed enough for controlling the traffic light. Make sure that the state space is mutually exclusive and complete.
b. What are the state spaces of the individual bulbs? What is the abstraction relation between the state space of the traffic light and the state spaces of the individual bulbs? How does the state space of the traffic light relate to the Cartesian product of the state spaces of the individual bulbs?
c. What are the valid state transition rules? Create the (simple) state graph!
d. How can we represent the same behaviour in a more compact way with hierarchical states?
e. When measuring the electric energy consumption of the traffic light we are only interested in the number of bulbs emitting light at a given time. Use abstraction on the current state machine to create a new one where the states differ only in their energy consumption!
f. At the end of the red signal there is a period when the green bulb of the perpendicular pedestrian crossing is blinking. Refine the state graph (from before the abstraction) in a way so that this state is presented separately!
g. There are 10 traffic lights along a road with 4 states each. How many states can the whole system have at the most? Do we have to handle every state?

2 Three-tier architecture

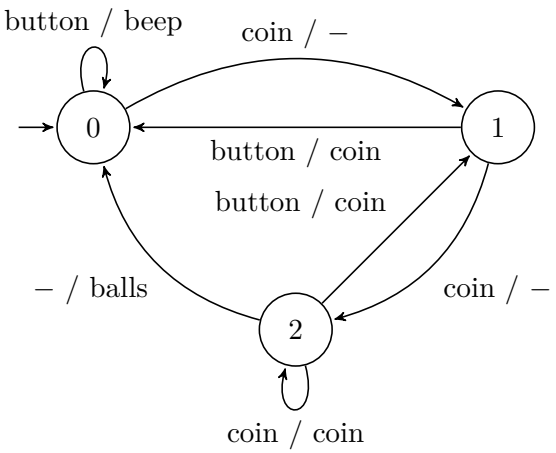
We would like to model an IT system which is built as a *three-layered architecture* in the following manner:

Are the following sets valid state spaces for modelling the behaviour of our system?

a. { Web server in use, Application server in use, Database server in use }
b. { Powered off, Idle, In use }
c. $\mathbb{N}$ (as the number of requests currently under processing)
d. { Processing of the request has not started yet, The servers are working on the request, The request is processed }
e. { True } $*$

3 „What is the output?”

Let's look at state machine M.



Note: the „−” symbol denotes a spontaneous transition when used as an input symbol, and a missing output when used as an output symbol. It's not the *don't care* symbol known from the digital design courses.

a. What real life system could be represented with this state machine? How does it work?
b. Is the behaviour model deterministic? Can we add or remove a single transition rule to change this?
c. Use abstraction on state graph M according to the state partitioning $\{\{0\}, \{1, 2\}\}$!
d. What kind of non-deterministic behaviours can be observed on the abstract model and where?

4 Touchscreen keyboard

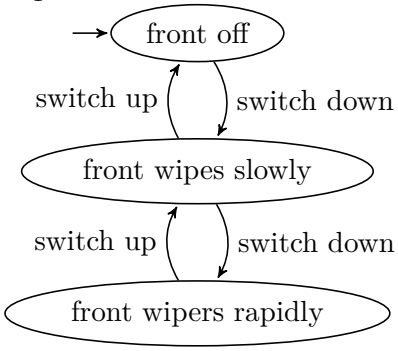
Model a touchscreen keyboard (designed for smartphones) with a state machine as presented during the lecture! The keyboard displays either lower case letters or capital letters or numbers and important symbols or rare symbols. The primary operation mode button switches between letters and numbers/symbols, the secondary operation mode button switches between the subcategories of the previous main categories. Moreover there is a capital letter state which changes automatically to lower case letters after a letter is pressed. Consider only the top left button (`q`/`Q`/`1`/`=`) and the two operation mode switching buttons as inputs, and consider the typed characters as outputs!

5 Windshield wiper

The front windshield wiper in a car has three states (*front off*, *front wipes slowly*, *front wipes rapidly*), the rear wiper has two states (*rear off*, *rear wipes*). The behaviour of the wipers are represented by the state machines M_1 and M_2, respectively.

a. Construct the asynchronous product of M_1 and M_2!
b. How many states and transitions does the resulting model have?
c. (Extra task) Can we represent the following behaviour in the product state machine, or in the state machines of the components: the rear wiper can only be switched on if the front wiper is on?

M_1 state machine:



M_2 state machine: