Universität Freiburg
Institut für Informatik
Prof. Dr. P. Fischer

Georges-Köhler Allee, Geb. 51
D-79110 Freiburg
peter.fischer@informatik.uni-freiburg.de

Exercises
Distributed Systemes: Part 2
Summer Term 2014
31.7.2014
Solution Proposal

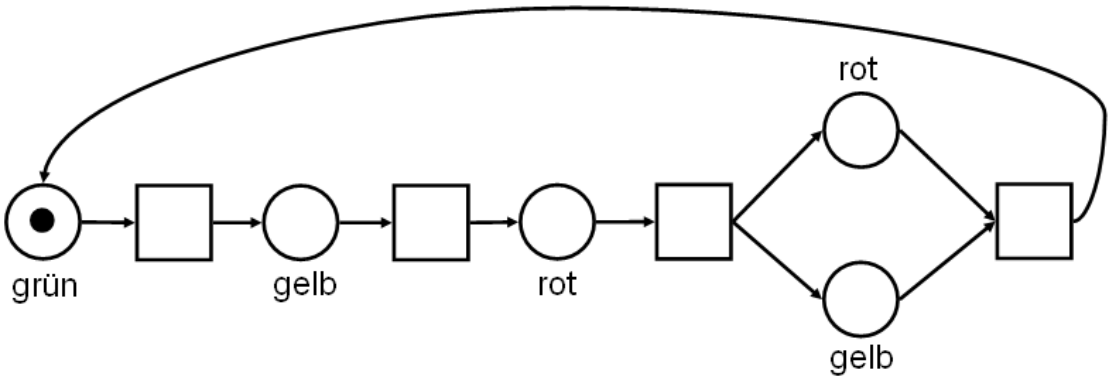
6. Exercise sheet: Petri nets

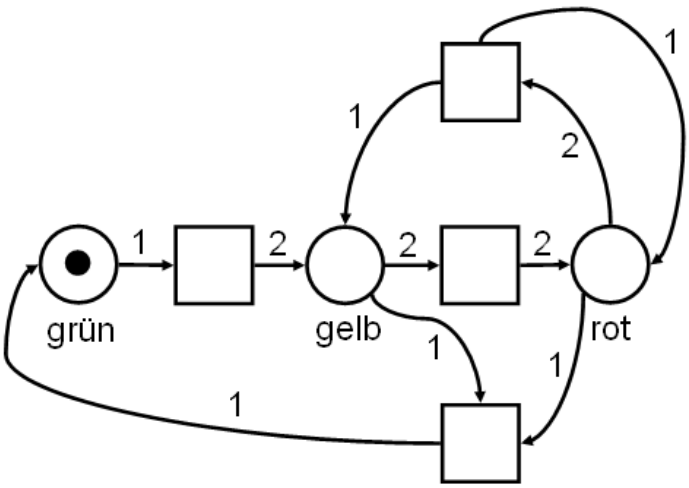
Exercise 1
Model a traffic light by a Petri-Net.

(1) You can use any number of places, however only multiplicity 1 is allowed.

(2) Now only 3 places (one for each color) may be used, but there are no restrictions on the multiplicities.

Solution:

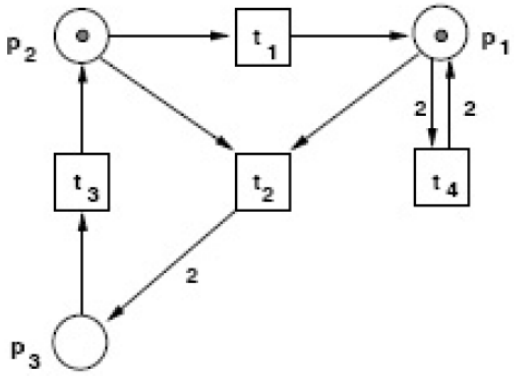




Exercise 2
Prove or give a counterexample: $m[q\rangle m' \Leftarrow m' = m + \Delta q$.

Solution:

Consider $m = (1, 1, 0)$ and $m' = (0, 1, 1)$ and further $q = t_2 t_3$, where $\Delta q = (-1, 0, 1)$. We have $m' = m + \Delta q$ and $m[q\rangle m'$.



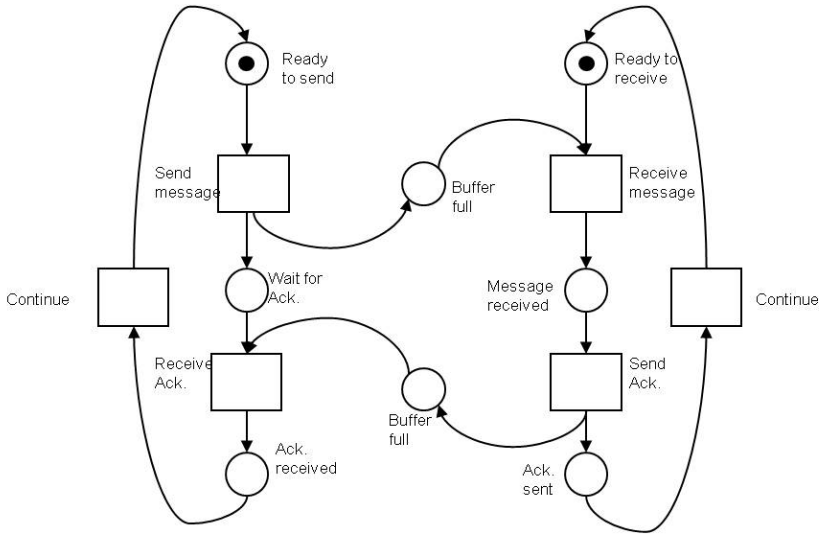
Consider now $m = (1, 0, 0)$ and $m' = (0, 0, 1)$ and further $q = t_2 t_3$, where $\Delta q = (-1, 0, 1)$. We have $m' = m + \Delta q$, however $m[q\rangle m'$ does not hold.

Exercise 3

(1) Model the following Handshaking protocol by a Petri-Net:
Two processes P1 and P2 mutually exchange messages. P1 is the sender and P2 the receiver. P1 starts in state *Ready-to-Send*. When it has sent a message to P2, it moves into the state *Ready-to-Receive* and waits for an acknowledgement ACK sent by P2. Once the acknowledgement has been arrived, P1 can send more messages. P2 starts in state *Waiting-for-Messages*. If it receives a message, it confirms by sending an acknowledgement ACK to P1 and waits for more messages.

(2) Give the reachability tree.

Solution:



(1)

(2) Consider the vector
$(ReadytoSend, ReadyToReceive, MsgBuffer, WaitForAck, MessageReceived, AckBuffer, AckSent, AckReceived)$
The initial marking m_0 is $(1, 1, 0, 0, 0, 0, 0, 0)$.
The only possible transition is *Send message*, leading to
$(0, 1, 1, 1, 0, 0, 0, 0)$.
Now, the only possible transition is *Receive message*, leading to
$(0, 0, 0, 1, 1, 0, 0, 0)$.
At this marking, we can only perform *Send Ack*, giving us
$(0, 0, 0, 1, 0, 1, 1, 0)$.
We now have the choice of two options:
a) Performing *Continue for Receive* yields $(0, 1, 0, 1, 0, 1, 0, 0)$

b) Performing *Ack Received* yields $(0, 0, 0, 0, 0, 0, 1, 1)$

For the first case, the next possible steps are *Ack Received* $(0, 1, 0, 0, 0, 0, 0, 1)$ and then $Continue\ for\ Send$, leading to m_0.

For the second case, we have choice to perform either *Continue* first, leading to $(1, 0, 0, 0, 0, 0, 1, 0)$ or $(0, 1, 0, 0, 0, 0, 0, 1)$. In either case, the other *Continue* is the only option, leading to m_0