



Discrete Event Systems

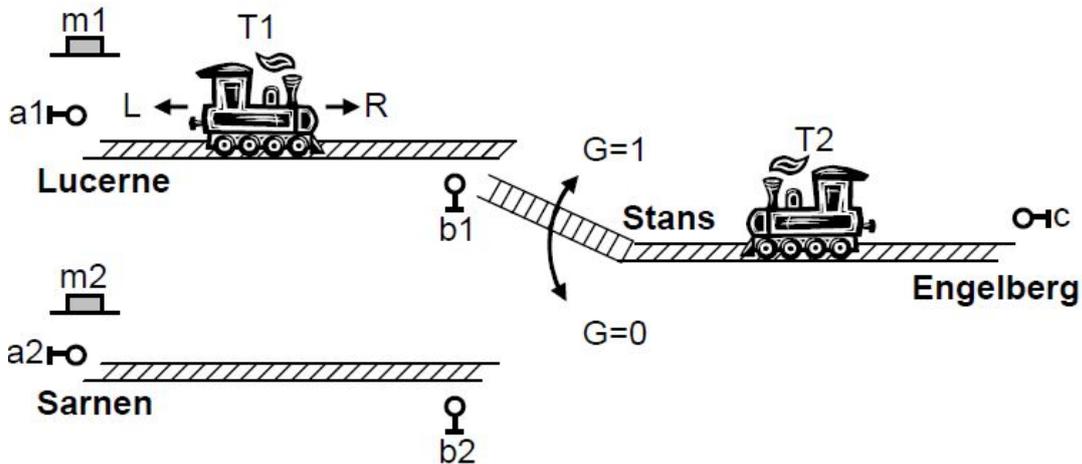
Exercise Sheet 10

1 The Winter Train Problem

We consider two trains T1 and T2 transporting skiers from Sarnen and Lucerne to Engelberg. Because there is only one ground rail track from Stans to Engelberg, at most one train might be between these two villages at any time. There is a switch in Stans, which either connects the track between Sarnen and Engelberg xor the track between Lucerne and Engelberg. After the train conductor has pressed a button m in (Sarnen or Lucerne), its train moves to Engelberg, but might have to wait in Stans until the other train has left the critical section. Once arrived in Engelberg, the train waits for 100s and then returns.

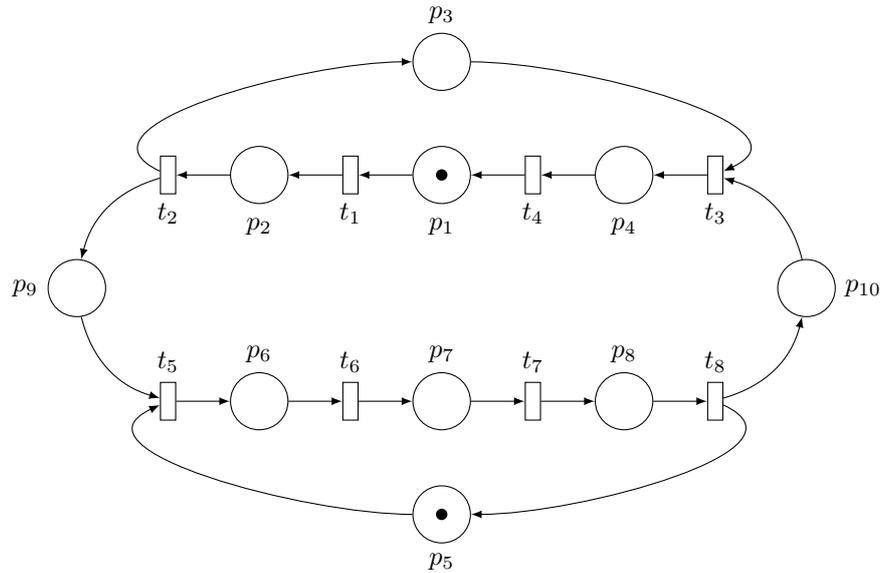
The sensors a_1, a_2, b_1, b_2 and c indicate the presence of a train with the value 1, otherwise, the value is 0. The switch in Stans is accessed through a variable G , as indicated in the picture. Finally, the motion of the trains is regulated by assigning 'R', 'L' or 'S' to the train, to move right, left, or stop, respectively.

The situation is shown below. Draw the corresponding State Chart using the notation introduced in the Figure!



2 Structural Properties of Petri Nets and Token Game

Given is the following petri net N_1 :

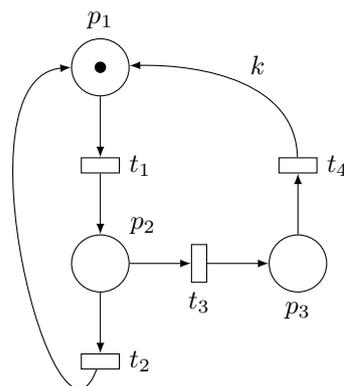


- What are the pre and post sets of transitions t_5 and t_8 and of place p_3 ?
- Which transitions are enabled after t_1 and t_2 fired?
- Determine the number of tokens in N_1 before and after t_2 fired.
- Play the token game for N_1 and construct the reachability graph.

Hint: You may denote the states in such a way that the index indicates the places that hold a token in this state, for example $\vec{s}_0 = (1, 0, 0, 0, 1, 0, 0, 0, 0, 0) =: s_{1,5}$.

3 Basic Properties of Petri Nets

Given is the following petri net N_2 :



Explain the terms *boundedness* and *deadlock-freeness* using this example, i.e. for which values of $k \in \mathbb{N}$ is the petri net N_2 bounded/unbounded and not deadlock-free?

4 Mutual Exclusion

Your task is to model a system as a petri net in which two processes want to access a common exclusive resource. This means that the two processes have to exclude each other mutually from the concurrent access to the resource (e.g. a critical program section). More concrete, this means:

1. A process executes its program.
2. In order to enter the critical section, a given mutex variable must be 0.
3. If this is the case, the process sets the mutex to 1 and executes its critical section.
4. When done, it resets the mutex to 0 and enters an uncritical section.
5. Then the procedure starts all over again.