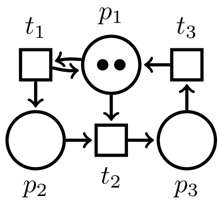
Technische Universität München **Summer Term 2019**
I7
Prof. J. Esparza / Ch. Weil-Kennedy / P. Meyer **12.08.2019**

Petri nets — Revision Exercise Sheet

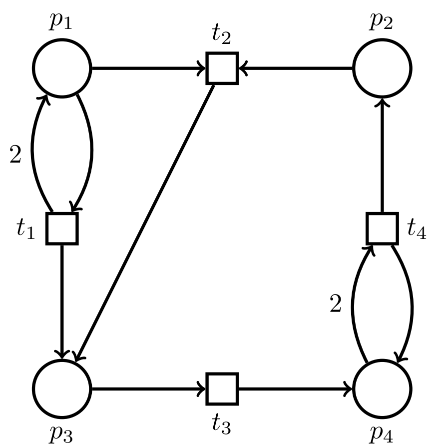
Exercise R.1

(From SS 2016, Exercise sheet 2)
Apply the backwards reachability algorithm to the Petri net below to decide if the marking $M = (0, 0, 2)$ can be covered. Record all intermediate sets of markings with their finite representation of minimal elements.



Exercise R.2

(From SS 2018, Exercise sheet 5)
Consider the following Petri net (with weights) $\mathcal{N}$:



(a) Identify the traps of only one place. Identify the proper siphons of this net.
Hint: there are 4 siphons.

(b) Use siphons/traps to prove or disprove that $\mathcal{N}$ is live from $M_0 = \{p_2, 3 \cdot p_4\}$,

(c) Can the marking equation be used to prove or disprove that $\{p_2, p_4\} \xrightarrow{*} \{p_1, 3 \cdot p_2, p_3\}$? Is so, why? If not, can traps or siphons help?

Exercise R.3

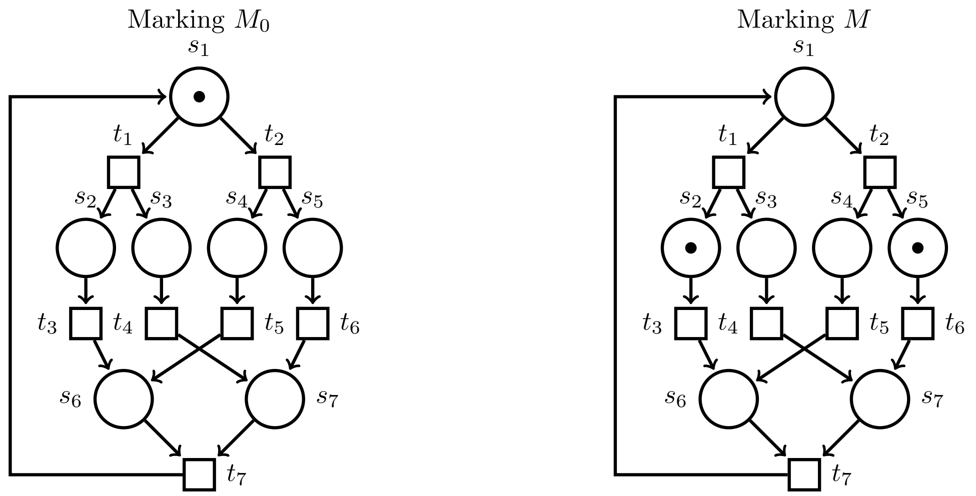
(From SS 2016, Exercise sheet 3)
For a Petri net (N, M_0) and a transition t of N, we define liveness levels in the following way:

- t is L_0-live (or dead) if t occurs in no firing sequence σ of N enabled at M_0.
- t is L_1-live if t occurs in some firing sequence σ of N enabled at M_0.
- t is L_2-live if for any $k \in \mathbb{N}$, t occurs at least k times in some firing sequence σ of N enabled at M_0.
- t is L_3-live if t occurs infinitely often in some infinite firing sequence σ of N enabled at M_0.
- t is L_4-live if for any reachable marking M from M_0, t occurs in some firing sequence σ of N enabled at M, i.e. t can always fire again. *Note*: If this holds for all transitions, this coincides with our standard definition of liveness for Petri nets.

For each $i \in \{0, 1, 2, 3\}$, exhibit a Petri net (N, M_0) and a transition t of N such that t is L_i-live, but not L_{i+i}-live.

Exercise R.4
(From SS 2019, Exercise sheet 5)

1. Give a procedure that, given a net $\mathcal{N}$, constructs a boolean formula φ satisfying the following properties:
   - The formula contains variables r_s for each place $s \in S$,
   - if φ is satisfiable, then $\mathcal{N}$ has a trap,
   - and if φ is not satisfiable, then $\mathcal{N}$ has no trap.
   - Additionally, if A is a model of φ, then the set given by $R = \{s \mid A(r_s)\}$ is a trap of $\mathcal{N}$.
2. Apply your procedure to the Petri net on the left below and give the resulting constraints.
3. Adapt your procedure such that, given two marking M_0 and M, it adds additional constraints to ensure that any trap R obtained as a solution by the constraints is marked at M_0 and unmarked at M. The constraints should be satisfiable iff a trap marked at M_0 and unmarked at M exists.
4. Construct the constraints for the Petri net below with the markings M_0 and M.
5. Use your constraints and the trap property to show that M is not reachable from M_0 in the net below.



Exercise R.5

(From SS 2016, Exercise sheet 2)
Reduce the reachability problem for Petri nets with weighted arcs to the reachability problem for Petri nets without weighted arcs.

For that, describe an algorithm that, given a Petri net with weighted arcs $N = (S, T, W, M_0)$ and a marking M, constructs a Petri net $N' = (S', T', F', M_0')$ and a marking M' such that M is reachable in N if and only if M' is reachable in N'. The algorithm should run in polynomial time (you may assume unary encoding for the weights in the input, although it is also possible with a binary encoding).

Apply the algorithm to the Petri net below with the target marking $M = (2, 0, 0)$ and give the resulting Petri net N' and marking M'.