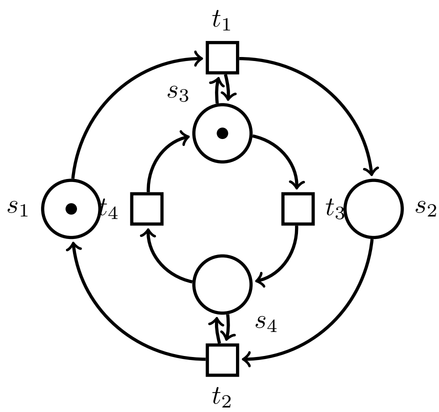
Petri nets — Exercise sheet 2

Exercise 2.1

(a) Give a net $\mathcal{N}$ and two markings M and M' such that $M \leq M'$, $(\mathcal{N}, M)$ is bounded, and $(\mathcal{N}, M')$ is *not* bounded.

(b) Give a net $\mathcal{N}$ and two markings M and M' such that $M \leq M'$, $(\mathcal{N}, M)$ is deadlock-free, and $(\mathcal{N}, M')$ is *not* deadlock-free.

(c) Give a net $\mathcal{N}$ and two markings M and M' such that $M \leq M'$, $(\mathcal{N}, M)$ is bounded *and* live, and $(\mathcal{N}, M')$ is *not* bounded. *Hint:* Add a place and arcs to the following net to obtain a solution:



Exercise 2.2

Let $\mathcal{N} = (P, T, W)$ be a net with weighted arcs. Let $M, M' \in \mathbb{N}^P$, $\sigma, \sigma' \in T^*$ and $t \in T$ be such that $M \xrightarrow{\sigma t \sigma'} M'$. Prove or disprove the following statements:

(a) if t does not consume any token, i.e $W(p,t) = 0$ for every $p \in P$, then $M \xrightarrow{t \sigma \sigma'} M'$.

(b) if t consumes no more tokens than it produces, i.e $W(p,t) \leq W(t,p)$ for every $p \in P$, then $M \xrightarrow{t \sigma \sigma'} M'$.

(c) if t produces no more tokens than it consumes, i.e. $W(t,p) \leq W(p,t)$ for every $p \in P$, then $M \xrightarrow{\sigma \sigma' t} M'$.

Exercise 2.3

Show that nets with place capacities and nets with weighted arcs are equivalent to standard nets. More precisely, sketch two algorithms solving the two following problems. The worst-case running time of your algorithms should be exponential.

(a)

INPUT: a net with place capacities $\mathcal{N} = (S, T, F, K)$, and two markings M and M'.

OUTPUT: a net $\mathcal{N}' = (S', T', F')$, and two markings L and L', such that $M \xrightarrow{*} M'$ in $\mathcal{N}$ if and only if $L \xrightarrow{*} L'$ in $\mathcal{N}'$.

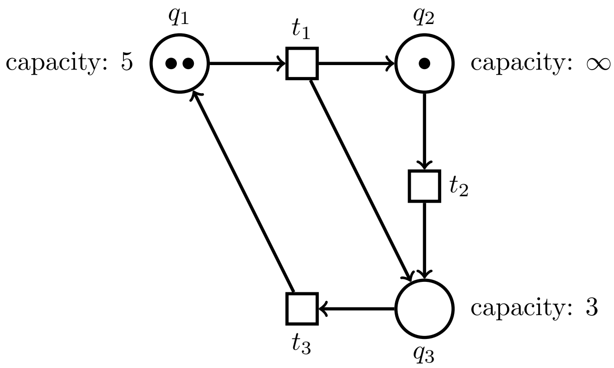
Apply your algorithm on the net below to the left with $M = \{2 \cdot q_1, q_2\}$ and $M' = \{2 \cdot q_1, q_3\}$.

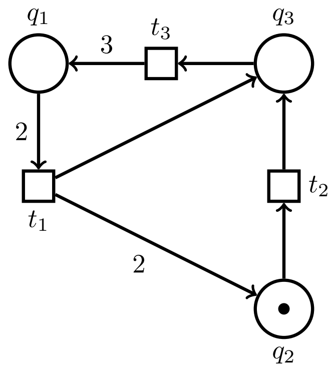
(b)

INPUT: a net with weighted arcs $\mathcal{N} = (S, T, W)$ and a markings M and M'.

OUTPUT: a net $\mathcal{N}' = (S', T', F')$, and two markings L and L', such that $M \xrightarrow{*} M'$ in $\mathcal{N}$ if and only if $L \xrightarrow{*} L'$ in $\mathcal{N}'$.

Apply your algorithm on the net below to the right with $M = \{q_2\}$ and $M' = \{q_1, 2 \cdot q_2, q_3\}$.





Exercise 2.4

We define *signed nets* as $\mathcal{N} = (S, T, F, \Sigma, s_{\mathcal{N}})$, where (S, T, F) is a classic net, Σ is a finite alphabet of *labels* and $s_{\mathcal{N}} : T \to \Sigma$ is called the *signature* of $\mathcal{N}$. Notice that the signature can assign the same label to two different transitions. The signature can be extended to a function $s_{\mathcal{N}}$ from T^* to Σ^*, using $s_{\mathcal{N}}(\sigma t) = s_{\mathcal{N}}(\sigma) s_{\mathcal{N}}(t)$ for $t \in T, \sigma \in T^*$. We call *free* a signed net such that $\Sigma = T$ and $s_{\mathcal{N}}$ is the identity.

For $M_0 \in \mathbb{N}^S$ a marking, we call $(\mathcal{N}, M_0)$ a *signed Petri net*. For $\mathcal{C}_t$ a set of markings, we call $(\mathcal{N}, M_0, \mathcal{C}_t)$ a *terminal signed Petri net*, and $\mathcal{C}_t$ is its set of *terminal markings*. We define

$$L_t(\mathcal{N}) = \{w \in \Sigma^* | \exists \sigma \in T^*, \exists C \in \mathcal{C}_t . M_0 \xrightarrow{\sigma} C \wedge s_{\mathcal{N}}(\sigma) = w\}$$

the *terminal language* for $(\mathcal{N}, M_0, \mathcal{C}_t)$.

(a) Show that regular languages (defined by automata) are included in terminal languages. *Hint:* Think of how to transform a DFA into a signed net.

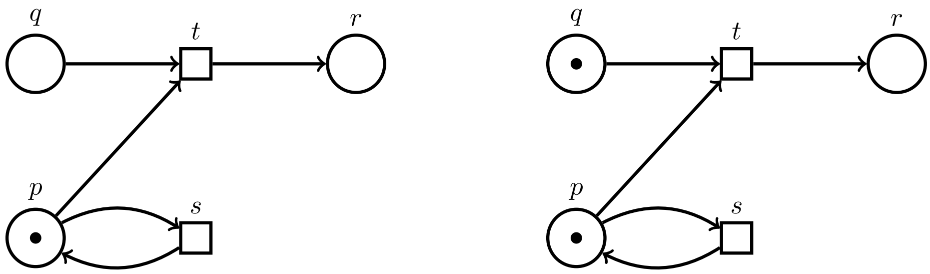
(b) Terminal languages can express more than just regular languages. For instance, show that the non context-free language $\{a^n b^n c^n | n \in \mathbb{N}, n \geq 1\}$ is a terminal language for a certain terminal signed Petri net with a finite terminal set of markings.

(c) If we consider only *free* terminal signed Petri nets, that is terminal signed Petri nets without duplication of labels, then there are regular languages that are not terminal languages. There are even finite languages that are not terminal languages, for example $L = \{abc, ba\}$. Show that L is not a terminal language for any free terminal signed Petri net.

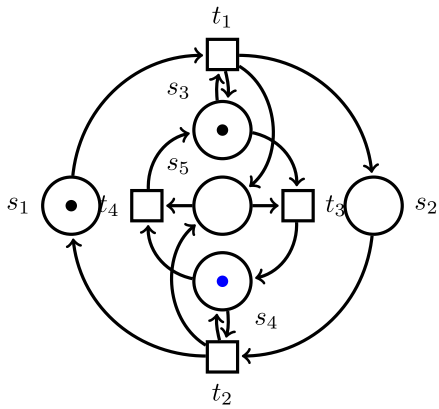
Solution 2.1

(a) The following net is bounded from the empty marking since its reachability set is empty. However, it is not bounded from $\{p\}$ since repetitively firing t increases the number of tokens in q.

(b) The following net is deadlock-free from $\{p\}$ since s is always enabled. However, it is not deadlock-free from $\{p, q\}$ since $\{p, q\} \xrightarrow{t} \{r\}$ and $\{r\}$ is dead.



(c) The following Petri net is live and bounded with the black tokens, but not bounded with the additional blue token in s_4, as repeatedly firing $t_1 t_2$ can put an arbitrary number of tokens in s_5.



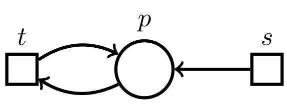
Solution 2.2

(a) True. Let $A, A' \in \mathbb{N}^P$ be such that $M \xrightarrow{\sigma} A \xrightarrow{t} A' \xrightarrow{\sigma'} M'$. Since $W(p,t) = 0$ for every $p \in P$, t is enabled at any marking. We have $A' - A \geq \mathbf{0}$ with $(A' - A)(p) = W(t, p)$ for every $p \in P$. Thus, $M \xrightarrow{t} M + (A' - A)$ and, by monotonicity, $M + (A' - A) \xrightarrow{\sigma} A + (A' - A)$. Therefore,

$$M \xrightarrow{t} M + (A' - A) \xrightarrow{\sigma} A' \xrightarrow{\sigma'} M'.$$

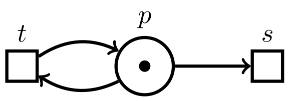
Notice that symmetrically, the following is also true: if t does not produce any token, i.e. $W(t, p) = 0$ for every $p \in P$, then $M \xrightarrow{\sigma\sigma' t} M'$.

(b) False. Consider the following Petri net:



We have $0 \xrightarrow{st} 1$ and $W(p,t) = W(t,p)$, yet ts cannot be fired from 0.

(c) False. Consider the following Petri net:



We have $1 \xrightarrow{ts} 0$ and $W(t,p) = W(p,t)$, yet st cannot be fired from 1.

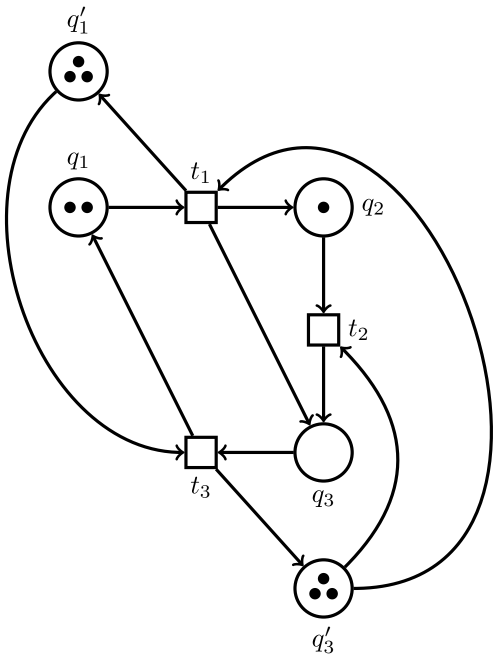
Solution 2.3

(a) We define $\mathcal{N}' = (S', T', F')$ as:

$$\begin{aligned} S' &= S \cup \{q' : q \in P \text{ s.t. } K(q) \neq \infty\}, \\ T' &= T, \\ F' &= F \cup \{(q', t) : (t, q) \in F', K(q) \neq \infty\} \cup \{(t, q') : (q, t) \in F', K(q) \neq \infty\}. \end{aligned}$$

Marking L is defined as the marking such that $L(q) = M(q)$ for every $q \in Q$ and $L(q') = K(q) - M(q)$ for every $q \in Q$ such that $K(q) \neq \infty$. Marking L' is similarly defined as $L'(q) = M'(q)$ for every $q \in Q$ and $L'(q') = K(q) - M'(q)$ for every $q \in Q$ such that $K(q) \neq \infty$.

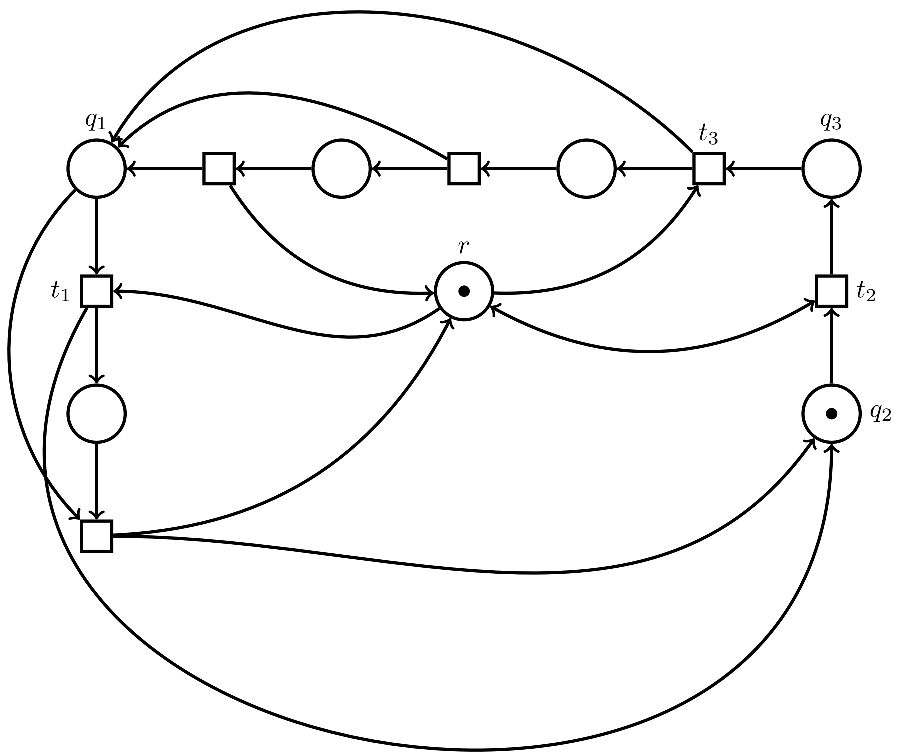
The resulting net is:



and the resulting markings are:

$$\begin{aligned} L &= \{2 \cdot q_1, 3 \cdot q_1', q_2, 3 \cdot q_3'\}, \\ L' &= \{q_1, 4 \cdot q_1', 2 \cdot q_2, q_3, 2 \cdot q_3'\}. \end{aligned}$$

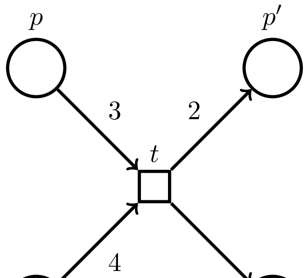
(b) Let us first give a net for the given net with weighted arcs:



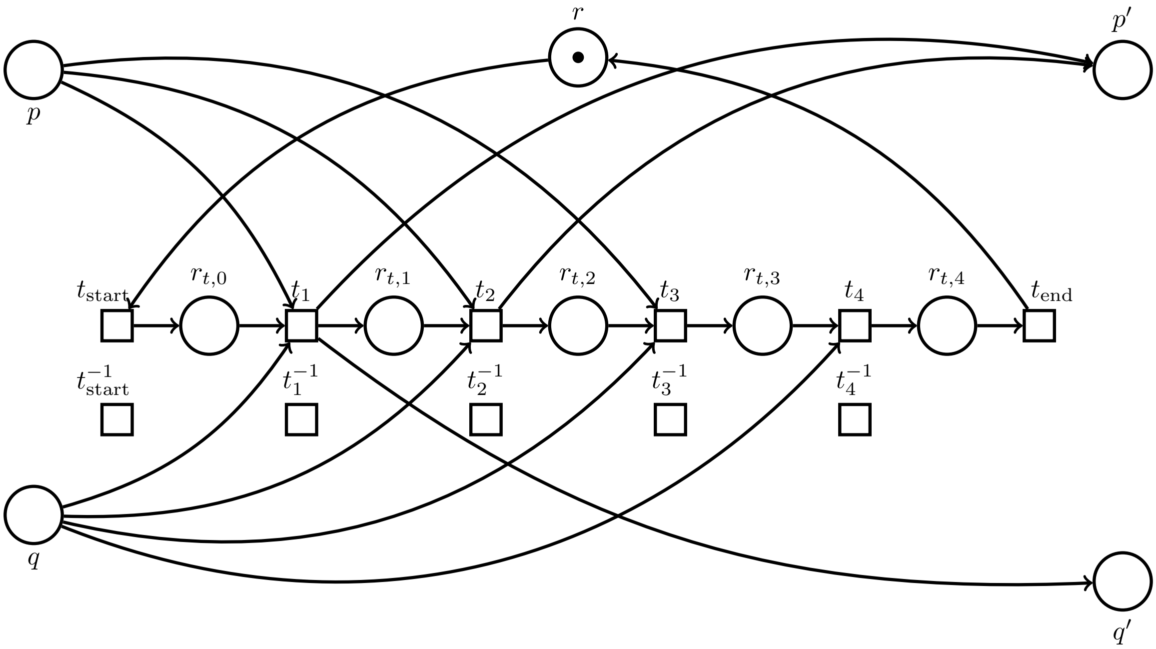
where the markings are:

$$L = \{q_2, r\},$$
$$L' = \{q_1, 2 \cdot q_2, q_3, r\}.$$

More generally, a transition such as:



can be converted into the following gadget that simulates the transition:

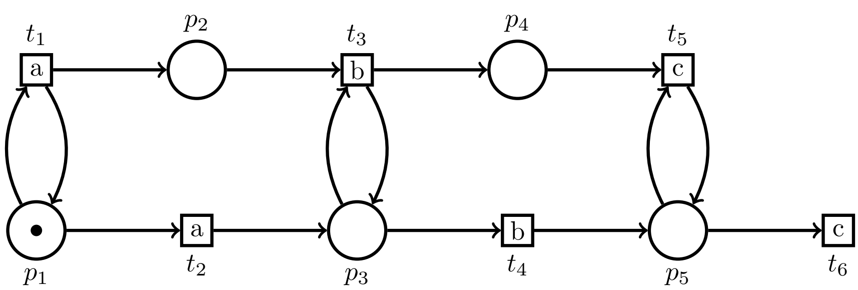


where place r must be shared by all gadgets, and s^{-1} denotes the inverse transition of s, i.e., $(p, s^{-1}) \in F \iff (s, p) \in F$ and $(s^{-1}, p) \in F \iff (p, s) \in F$ for every place p (the arcs are not drawn for readability reasons). Place r is a control place that enforces that only one transition of the original net is being simulated at one time. The token in place r is consumed at the beginning of the simulation, and replaced at the end. The inverse transitions are there so that we do not create a deadlock inside the simulated transition. Indeed consider a marking of the gadget with four tokens in q, one token in r and two tokens in p. The original transition being simulated would not have been enabled, but here transition t_{start} is enabled and we can get to place $r_{t,2}$ by taking t_1 then t_2. But then we are "stuck" and the transition t_3 is not enabled. The inverse transitions allow us to "backtrack" and bring the tokens back into places p and q.

Solution 2.4

(a) Let L be a regular language given by a DFA (deterministic finite automaton) $\mathcal{A}$. Without loss of generality, we can take $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ with a single initial state q_0, transition function $\delta : Q \times \Sigma \rightharpoonup Q$ and final states $F \subseteq Q$.
We transform $\mathcal{A}$ into a terminal signed Petri net $(\mathcal{N}, M_0, \mathcal{C}_t)$. The set of places of $\mathcal{N}$ is Q, its set of labels is Σ and M_0 is the marking that puts one token in q_0 and 0 elsewhere. For every transition $t : q \xrightarrow{a} q'$ in $\mathcal{A}$, we define a transition t in $\mathcal{N}$ such that $(q, t) \cup (t, q')$ is in the flow of $\mathcal{N}$ and $s_{\mathcal{N}}(t) = a$. We define the set of terminal markings $\mathcal{C}_t$ as the markings M_q that put one token in q and 0 elsewhere, for every final state $q \in F$.

(b) We display a terminal signed Petri net $(\mathcal{N}, M_0, \mathcal{C}_t)$ whose terminal language is $\{a^n b^n c^n | n \in \mathbb{N}, n \geq 1\}$.
Let $(\mathcal{N}, M_0)$ be the signed Petri net illustrated below, with $M_0 = (1, 0, 0, 0, 0)$. Let $\mathcal{C}_t$ be the singleton set $\{(0, 0, 0, 0, 0)\}$.



This petri net appears in [1, Abb. 6.1].

(c) We reason by contradiction. Let us assume L is a terminal language for a terminal free signed Petri net $(\mathcal{N}, M_0, \mathcal{C}_t)$. Since a free signed net labels transitions uniquely, we must have at least three transitions a, b, c in $\mathcal{N}$. Transition sequences ab and ba are both enabled from M_0 because abc and ba are in the language. The monotonicity lemma for regular Petri nets still holds for signed Petri nets, as labelling the transitions does not modify the proof of this lemma. Therefore initial marking M_0 is modified in the same way by the action of transition sequences ab and ba. That is, there exists M such that $M_0 \xrightarrow{ab} M$ and $M_0 \xrightarrow{ba} M$. Since ba is in the language, M must be a terminal marking; but then ab is also in L, and this is not the case.

References

[1] Lutz Priese and Harro Wimmel. Petri-netze. ISBN 3-540-44289-8, 2003.