# Petri nets – Homework 8

Discussed on Thursday $2^{\text{nd}}$ July, 2015.

*For questions regarding the exercises, please send an email to meyerphi@in.tum.de or just drop by at room 03.11.042.*

## Exercise 8.1 Commoner's Liveness Theorem for general Petri nets

(a) Exhibit a non-live Petri net $(N, M_0)$ where every proper siphon contains a trap marked at $M_0$.

(b) Exhibit a live Petri net $(N, M_0)$ with a proper siphon $R$ of $N$ that does not contain a trap marked at $M_0$.

## Exercise 8.2 Hack's Boundedness Theorem for general Petri nets

A structural component of Petri nets are S-components (Definition 5.3.5 in the script):

**Definition 8.2.1.** [S-component] Let $N = (S, T, F)$ be a net. A subnet $N' = (S', T', F')$ of $N$ is an *S-component* of $N$ if

1. $T' = {}^\bullet S' \cup S'^\bullet$ (where ${}^\bullet s = \{t \in T \mid (t, s) \in F\}$, and analogously for $s^\bullet$).
2. $N'$ is a strongly connected S-net.

The fundamental propery of S-components is (Proposition 5.3.6 in the script):

**Proposition 8.2.1.** Let $(N, M_0)$ be a Petri net and let $N' = (S', T', F')$ be an S-component of $N$. Then $M_0(S') = M(S')$ for every marking $M$ reachable from $M_0$.

(a) Prove: Let $(N, M_0)$ be a Petri net. If every place of $N$ belongs to an S-component, then $(N, M_0)$ is bounded.

(b) Exhibit a live and bounded Petri net $(N, M_0)$ with a place $s$ of $N$ that does not belong to any S-component.

## Exercise 8.3 Minimal path length in 1-bounded Petri nets

For each $n \in \mathbb{N}$, give a live and 1-bounded Petri net $(N, M_0)$ and a marking $M$ of $N$ such that the size of the net grows linearly, but the length of the minimal occurrence sequence $\sigma$ with $M_0 \xrightarrow{\sigma} M$ grows exponentially, i.e., $|\sigma| \in \Omega(2^n)$.

*Hint*: Try to model a counter with a binary encoding that counts incrementally from 0 to $2^n$ by firing one or more transitions for each step.

## Exercise 8.4 Reducing SAT to reachability in free-choice systems

Reduce the satisfiability problem for boolean formulas in conjunctive normal form to the reachability problem in free-choice systems.

For that, give a polynomial time translation that, for a given formula $\varphi$, produces a free-choice system $(N, M_0)$ and a marking $M$ such that $\varphi$ is satisfiable iff $M$ is reachable in $(N, M_0)$. Describe your reduction informally and give the resulting Petri net when applying it to the formula below.

$$\varphi = (\neg x_1 \lor \neg x_2 \lor x_3) \land (x_2 \lor \neg x_3)$$

## Exercise 8.5 Simulating a bounded stack

A *bounded stack* is a tuple $K = (\Gamma, k)$, where $\Gamma$ is the stack alphabet and $k$ is the stack size. A *configuration* of the stack is a sequence $\gamma \in \Gamma^*$. On a stack, we can perform the following actions:

- Push an element $c \in \Gamma$ on the top of the stack if the stack has less than $k$ elements.
- Pop an element $c \in \Gamma$ from the stack if the top element is $c$.
- Assert that the stack is empty if the stack has no elements.

More formally, for a bounded stack $(\Gamma, k)$, we have the *actions* $A_\Gamma = \{push_c, pop_c \mid c \in \Gamma\} \cup \{empty\}$ which induce the following transitions rules on the configurations:

$$\begin{aligned}
\epsilon &\xrightarrow{empty} \epsilon \\
\gamma &\xrightarrow{push_c} \gamma c \quad \text{for } c \in \Gamma,\ \gamma \in \Gamma^*,\ |\gamma| < k \\
\gamma c &\xrightarrow{pop_c} \gamma \quad \text{for } c \in \Gamma,\ \gamma \in \Gamma^*
\end{aligned}$$

A sequence of actions $w = a_1 a_2 \dots a_n \in A_\Gamma^*$ is a *computation* of a stack $K$ if there exist configurations $\gamma_1 \gamma_2 \dots \gamma_n$ such that $\epsilon \xrightarrow{a_1} \gamma_1 \xrightarrow{a_2} \gamma_2 \dots \gamma_{n-1} \xrightarrow{a_n} \gamma_n$ is a transition sequence according to the above transition rules. For example, with $K = (\{a, b\}, 2)$, the sequence *empty push$_a$ push$_b$ pop$_b$ pop$_a$ empty push$_b$* is a computation with the intermediate configurations

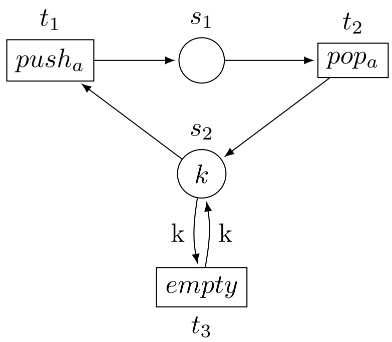
$$\epsilon \xrightarrow{empty} \epsilon \xrightarrow{push_a} a \xrightarrow{push_b} ab \xrightarrow{pop_b} a \xrightarrow{pop_a} \epsilon \xrightarrow{empty} \epsilon \xrightarrow{push_b} b.$$

A Petri net $(N, M_0)$ together with a labeling function $h_l : T \to A_\Gamma \cup \{\tau\}$ *simulates* the computations of a bounded stack $K = (\Gamma, k)$ if, with the homomorphism $h : T^* \to A_\Gamma^*$ defined by

$$\begin{aligned}
h(\epsilon) &= \epsilon \\
h(t) &= \epsilon \qquad \text{if } h_l(t) = \tau \\
h(t) &= h_l(t) \qquad \text{if } h_l(t) \in A_\Gamma \\
h(\sigma t) &= h(\sigma) h(t),
\end{aligned}$$

we have: $\{h(\sigma) \mid \exists M : M_0 \xrightarrow{\sigma} M\} = \{w \in A_\Gamma^* \mid w \text{ is a computation of } K\}$. Basically, some transitions of the Petri net correspond to actions of the stack, and the occurrence sequences of the net projected onto the actions of these transitions correspond to the computations of the stack.

As an instance, if $\Gamma = \{a\}$, for each $k$, we can give the following Petri net (with weights) which simulates a stack $K = (\{a\}, k)$. The label of each transition is given inside the box for the transition.




For $\Gamma = \{a, b\}$ and a given $k$, give a construction to produce a Petri net that simulates the bounded stack $K = (\{a, b\}, k)$. Give the resulting Petri net when applying the construction with $k = 3$. Describe informally how the Petri net changes as $k$ increases. Ensure that the size of the Petri net only grows linearly with $k$.