**Technische Universität München (I7)**
**Prof. J. Esparza / P. Meyer**

**Summer term 2015**
**Thursday 21st May, 2015**

# Solution

## Petri nets – Homework 3

Discussed on Thursday 21st May, 2015.

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

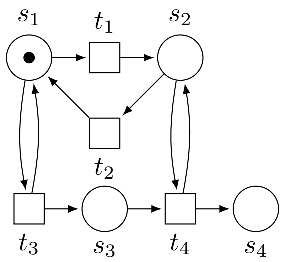
### Exercise 3.1 Coverability

Construct the coverability graph for the Petri net below.

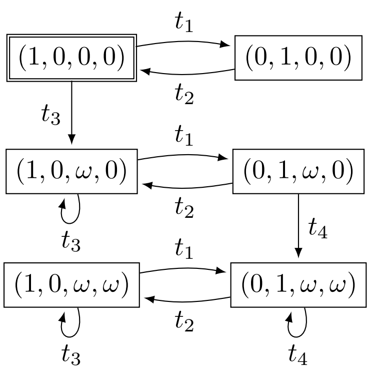
(a) List the unbounded places of the Petri net.

(b) Decide if the following markings $M = (s_1, s_2, s_3, s_4)$ are coverable:

$$M_1 = (1,0,0,0) \quad M_2 = (0,0,1,0) \quad M_3 = (1,1,0,0) \quad M_4 = (0,0,1,1) \quad M_5 = (1,0,1,0) \quad M_6 = (0,1,0,1)$$




**Solution:** The coverability graph is as follows:




(a) The places $s_3$ and $s_4$ are unbounded, as there are $\omega$-markings $M, M'$ with $M(s_3) = \omega$ and $M'(s_4) = \omega$.

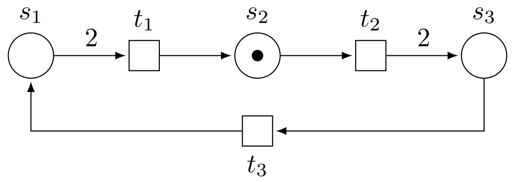
(b) The markings $M_1$, $M_2$, $M_4$ and $M_5$ are covered by $(1,0,\omega,\omega)$ and $M_6$ is covered by $(0,1,\omega,\omega)$. The marking $M_3$ is not coverable, as there is no $\omega$-marking $M$ with $M(s_1) \geq 1$ and $M(s_2) \geq 1$.

### Exercise 3.2 Reachability in Petri nets with weighted arcs

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 $M'$.




**Solution:** We use the following approach: We replace each place in the Petri net with weighted arcs with a ring of places in the Petri net without weighted arcs. The size of the ring is given by the maximum input or output weight, and the sum of the tokens in the ring represent the number of tokens in the original place. The tokens can move around freely in the ring, and a transition with a weighted arc that puts or takes $k$ tokens into or out of the original place is now connected with unweighted arcs to $k$ of the places in the ring.

Formally, let $S = \{s_1, \ldots, s_n\}$ be the places of the Petri net $N$. For each $s_i \in S$, define an integer $k_i$ by

$$k_i := \max(\{W(t, s_i) \mid t \in T\} \cup \{W(s_i, t) \mid t \in T\})$$

The places in the Petri net $N'$ are the sets of ring places for each original place. The transitions are the original transition, plus a fresh set of ring transitions.

$$S' = \bigcup_{s_i \in S} \{s_{i,j} \mid 1 \leq j \leq k_i\} \qquad\qquad T' = T \uplus \{t_{s_{i,j}} \mid s_{i,j} \in S'\}$$

The flow relation connects each transition $t$ to a number of ring places $s_{i,j}$ given by the weight between $t$ and $s_i$. We also connect the ring places and transitions cyclically.

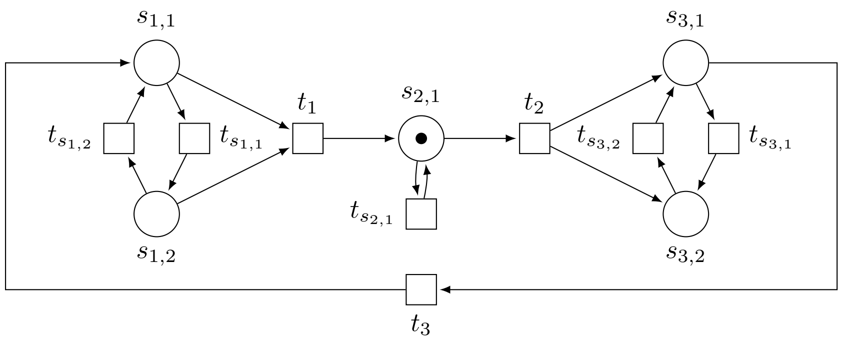
$$\begin{aligned} F' = &\{(s_{i,j}, t) \in S' \times T \mid W(s_i, t) \geq j\} \ \cup \ \{(t, s_{i,j}) \in T \times S' \mid W(t, s_i) \geq j\} \ \cup \\ &\{(s_{i,j}, t_{s_{i,j}}) \mid s_{i,j} \in S'\} \ \cup \ \{(t_{s_{i,j}}, s_{i,1+(j \bmod k_i)}) \mid s_{i,j} \in S'\} \end{aligned}$$

The initial and target marking are given by having all the tokens in the first place of each ring.

$$M_0'(s_{i,j}) = \begin{cases} M_0(s_i) & \text{if } j = 1 \\ 0 & \text{otherwise} \end{cases} \qquad\qquad M'(s_{i,j}) = \begin{cases} M(s_i) & \text{if } j = 1 \\ 0 & \text{otherwise} \end{cases}$$

By construction, if $M$ is reachable in $N$, then there is a reachable marking $M'$ in $N'$ with $\sum_j M'(s_{i,j}) = M(s_i)$ for all $s_i \in S$. We can use the ring transitions to move all tokens in $s_{i,j}$ to $s_{i,1}$ to reach the target marking.

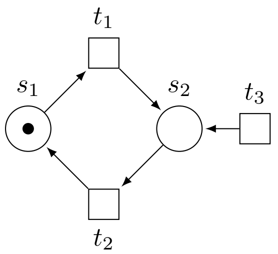
Applying the construction to the Petri net and marking above gives us the Petri net below, along with the marking $M'$ given by $M'(s_{1,1}) = 2$ and $M'(s) = 0$ for all places $s \neq s_{1,1}$.



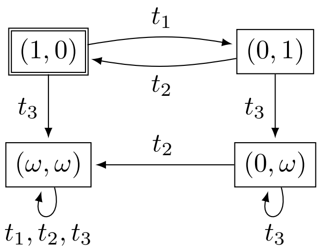

## Exercise 3.3 Uniqueness of the coverability graph

In the algorithm for the construction of the coverability graph, the search strategy (breadth-first or depth-first search and traversal order for visiting child nodes) is not specified. Show that the coverability graph obtained is not unique by exhibiting a Petri net and two different coverability graphs for this Petri net obtained by the algorithm with different search strategies.

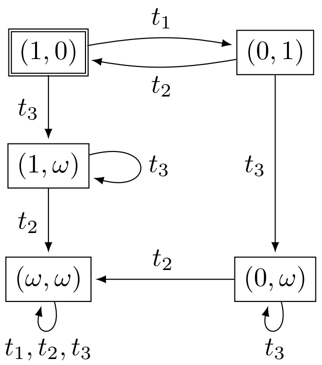
**Solution:** Take the following Petri net:




If, for the construction of the coverability graph, the path $M_0 = (1,0) \xrightarrow{t_1} (0,1) \xrightarrow{t_2} (1,0) \xrightarrow{t_3} (1,1)$ is first explored, then $(1,1)$ strictly covers and is reachable from $(0,1)$ and $(1,0)$, so the node $(\omega,\omega)$ is added to the coverability graph. The resulting graph is:




If instead, the path $M_0 = (1,0) \xrightarrow{t_3} (1,1)$ is explored first, then $(1,1)$ only strictly covers and is reachable from $(1,0)$, so the node $(1,\omega)$ is added to the coverability graph, resulting in the following graph:




## Exercise 3.4 Backwards reachability with transfer arcs

Another variant of Petri nets are nets with transfer arcs, a generalization of nets with reset arcs:

**Definition 3.4.1** (Nets with transfer arcs)**.** A *net with transfer arcs* $N = (S, T, F, R)$ consists of two disjoint sets of places and transitions, a set $F \subseteq (S \times T) \cup (T \times S)$ of arcs, and a set $R \subseteq (S \times T) \cup (T \times S)$, disjoint from F, of *transfer arcs*.

A transition $t$ is *enabled* at a marking $M$ of $N$ if $M(s) > 0$ for every place $s$ such that $(s,t) \in F \cup R$. If $t$ is enabled then it can *occur* leading to the marking $M'$ obtained after the following operations:

1. Let $k$ be the sum of the tokens in all places $s$ such that $(s,t) \in R$, i.e., $k := \sum_{\{s \in S \mid (s,t) \in R\}} M(s)$.
2. Remove one token from every place $s$ such that $(s,t) \in F$.
3. Remove all tokens from every place $s$ such that $(s,t) \in R$.
4. Add one token to every place $s$ such that $(t,s) \in F$.
5. Add $k$ tokens to every place $s$ such that $(t,s) \in R$.

Show that the abstract backwards-reachability algorithm can be applied to Petri nets with transfer arcs by showing that the transition relation is monotonic.

**Solution:** To show monotonicity, we need to show that for every $x \xrightarrow{t} y$ and every $x' \geq x$, there is $y' \geq y$ such that $x' \xrightarrow{t} y'$.

First, if $t$ is enabled at $x$, then $x'(s) \geq x(s) > 0$ for every place $s$ such that $(s,t) \in F \cup R$. Therefore $t$ is also enabled at $x'$ and we can obtain $y'$ from $x' \xrightarrow{t} y'$. To show that $y' \geq y$, we show that each of the steps from the operations above is monotonic.

For a transition $t$ and $i \in [1,5]$, we denote by $m \xrightarrow{t_{|i}} m'$ that applying step $i$ from the 5 operations above on $m$ results in the marking $m'$. We then break $x \xrightarrow{t} y$ down into $x \xrightarrow{t_{|1}} x_1 \xrightarrow{t_{|2}} x_2 \xrightarrow{t_{|3}} x_3 \xrightarrow{t_{|4}} x_4 \xrightarrow{t_{|5}} y$ and $x' \xrightarrow{t} y'$ into $x' \xrightarrow{t_{|1}} x'_1 \xrightarrow{t_{|2}} x'_2 \xrightarrow{t_{|3}} x'_3 \xrightarrow{t_{|4}} x'_4 \xrightarrow{t_{|5}} y'$. Assuming $x' \geq x$, we show that $x'_1 \geq x_1$, $x'_2 \geq x_2$, $x'_3 \geq x_3$, $x'_4 \geq x_4$ and finally $y' \geq y$.

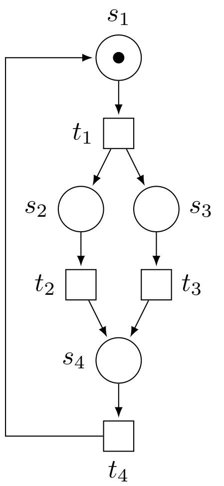
1. $x \xrightarrow{t_{|1}} x_1$, $x' \xrightarrow{t_{|1}} x_1'$ and $x' \geq x$: We set $k' := \sum_{\{s \in S | (s,t) \in R\}} x'(s) \geq \sum_{\{s \in S | (s,t) \in R\}} x(s) =: k$. The number of tokens remains unchanged, therefore $x_1' = x' \geq x = x_1$.
2. $x_1 \xrightarrow{t_{|2}} x_2$, $x_1' \xrightarrow{t_{|2}} x_2'$ and $x_1' \geq x_1$: For every $s$ with $(s,t) \in F$, We have $x_2'(s) = x_1'(s) - 1 \geq x_1(s) - 1 = x_2'(s)$ and for every other $s$, we have $x_2'(s) = x_1'(s) \geq x_1'(s) = x_2(s)$, therefore $x_2' \geq x_2$.
3. $x_2 \xrightarrow{t_{|3}} x_3$, $x_2' \xrightarrow{t_{|3}} x_3'$ and $x_2' \geq x_2$: For every $s$ with $(s,t) \in R$, we have $x_3'(s) = 0 = x_3(s)$ and for every other $s$, we have $x_3'(s) = x_2' \geq x_2 = x_3(s)$, therefore $x_3' \geq x_3$.
4. $x_3 \xrightarrow{t_{|4}} x_4$, $x_3' \xrightarrow{t_{|4}} x_4'$ and $x_3' \geq x_3$: For every $s$ with $(t,s) \in F$, we have $x_4'(s) = x_3'(s) + 1 \geq x_3(s) + 1 = x_4(s)$ and for every other $s$, we have $x_4'(s) = x_3' \geq x_3 = x_4(s)$, therefore $x_4' \geq x_4$.
5. $x_4 \xrightarrow{t_{|5}} y$, $x_4' \xrightarrow{t_{|5}} y'$ and $x_4' \geq x_4$: For every $s$ with $(t,s) \in R$, we have $y'(s) = x_4' + k' \geq x_4 + k = y(s)$ and for every other $s$, we have $y'(s) = x_4' \geq x_4 = y(s)$, therefore $y' \geq y$.

## Exercise 3.5 Number of tokens in bounded nets

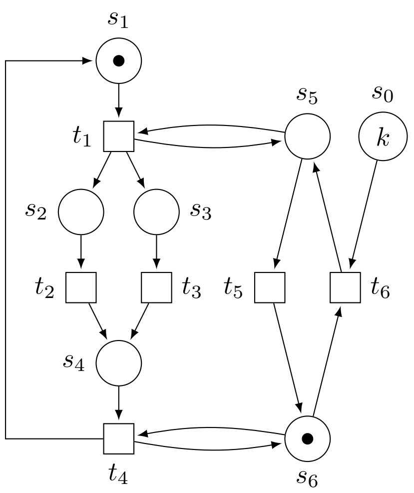
Give a family of bounded Petri nets $\{N_k\}_{k \in \mathbb{N}}$ such that the size of $N_k$ is bounded by $O(k)$ (that is, there is a $c \in \mathbb{N}$ such that for all $N_k = (S, T, F, M_0)$, we have $|S| + |T| + |F| \leq ck$ and $\forall s \in S : M_0(s) \leq ck$), but each $N_k$ has a reachable marking $M$ and a place $s$ with $M(s) \geq 2^{2^k}$.

*Hint:* Construct a net that doubles the number of tokens in a place. Modify it so that one occurrence sequence for doubling removes exactly one token from a certain place. Use this construct again or the construct from the lecture to put $2^k$ tokens into that place.

**Solution:** In the following net, we can fire $t_1 t_2 t_3 t_4 t_4$ to duplicate a token in $s_1$. If there are $n$ tokens in $s_1$, the firing sequence $t_1^n t_2^n t_3^n t_4^{2n}$ doubles the number of tokens in $s_1$.




By modifying the net as follows, we ensure that to fire $t_1^n t_2^n t_3^n t_4^{2n}$, we need to move the token from $s_6$ to $s_5$ and back and remove one token from $s_0$. Now the net is bounded, and with $k$ tokens in $s_0$, we can put up to $2^k$ tokens in $s_1$.



We can duplicate the net and use the output place $s_1$ as the input place $s_0$ for the other net. In the following net, we can fire the transitions in the right net to put $2^k$ tokens in $s_1'$, and then fire the transitions in the left net to put $2^{2^k}$ tokens in $s_1$. The net has a constant size, and we have $M(s) \leq k$ for all places $s$.