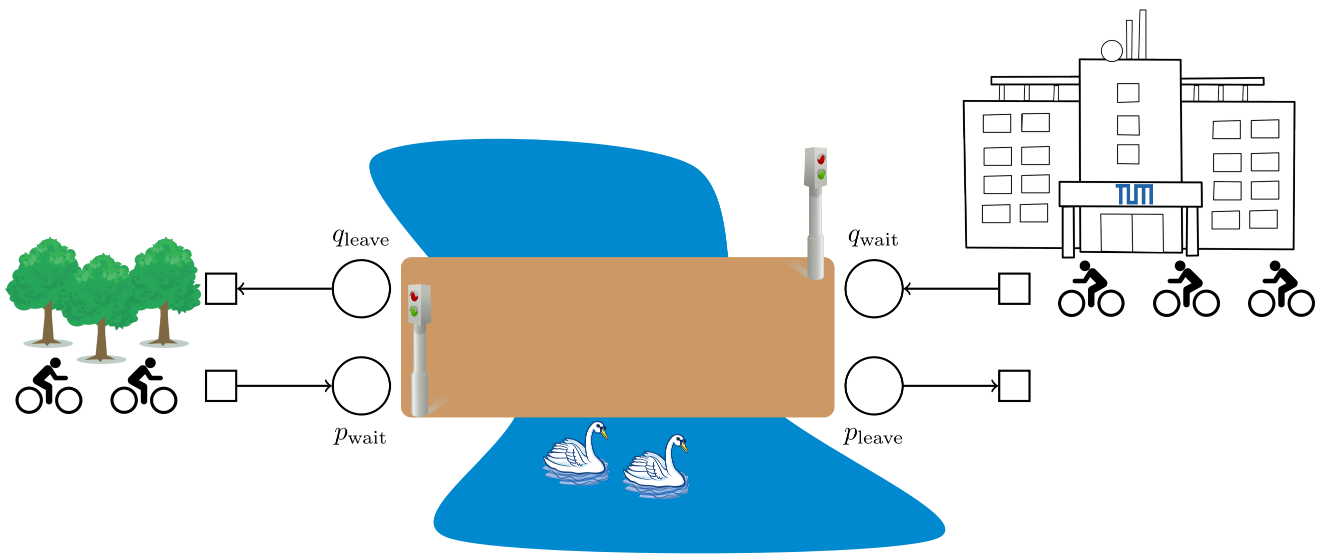
**Technische Universität München** **Summer term 2017**
**I7**
**Prof. J. Esparza / Dr. M. Blondin** **01.05.2017**

# Petri nets — Homework 1

Due 09.05.2017

**Exercise 1.1 (adapted from [1, ex. 2.22])**

Consider a new (fictive) bridge connecting TUM to the other side of the Isar. Since this bridge is narrow, it can only be used in one direction at a time. Moreover, for safety reasons, there should not be more than six cyclists at a time on the bridge. The university wants the bridge to be equipped with a system controlling green and red lights on both ends of the bridge. For each direction, when the green light is on, cyclists are allowed to get onto the bridge; and when the red light is on, cyclists are *not* allowed to get onto the bridge.




Model the bridge as a Petri net (with weighted arcs) by extending the partial model shown above. Cyclists should flow from $p_{\text{wait}}$ to $p_{\text{leave}}$, and from $q_{\text{wait}}$ to $q_{\text{leave}}$. Assume that, initially, the left green light is on, the right red light is on, and the bridge is empty. Make sure that the model respects safety, i.e. that bikes are not allowed to go in opposite directions, and that the bridge cannot hold more than six bikes.

**Exercise 1.2**

Consider Lamport's 1-bit mutual exclusion algorithm:

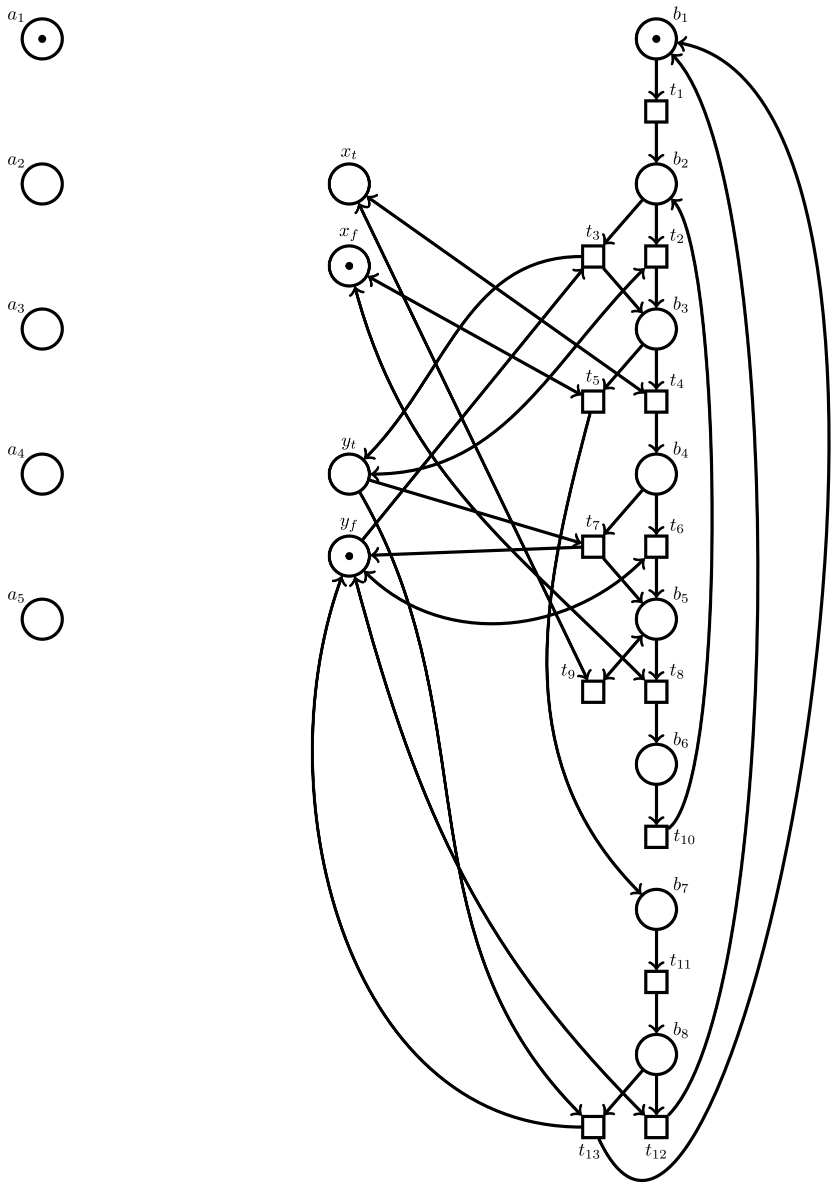
First process

```
1.  while True:
2.    x = True
3.    while y: pass
4.    # critical section
5.    x = False
```

Second process

```
1.  while True:
2.    y = True
3.    if x then:
4.      y = False
5.      while x: pass
6.      goto 2
7.    # critical section
8.    y = False
```

The algorithm can be modeled by a Petri net $\mathcal{N}$ where each program location (i.e. line of code of a process) is associated to a place, and where the shared binary variables `x` and `y` are associated to two places each. In more details, $\mathcal{N} = (P, T, F)$ where $P = \{a_1, \ldots, a_5, b_1, \ldots, b_8, x_t, x_f, y_t, y_f\}$. A token in $a_i$ (resp. $b_i$) indicates that the first (resp. second) process is at line $i$; a token in $x_t$ (resp. $y_t$) indicates that `x` (resp. `y`) has value `True`; and a token in $x_f$ (resp. $y_f$) indicates that `x` (resp. `y`) has value `False`. The initial marking of $\mathcal{N}$ is $M_0 = \{a_1, b_1, x_f, y_f\}$. We give a partial Petri net that only models the second process:




(a) Complete the above Petri net $\mathcal{N}$ so that it also models the first process. You should not add new places, only transitions and arcs. Note that `pass` is a "no operation", i.e. an operation without any effect.

(b) Complete the given `APT` file for $\mathcal{N}$ accordingly, and verify whether

  (i) $(\mathcal{N}, M_0)$ is bounded;

  (ii) $(\mathcal{N}, M_0)$ is live.

(c) Complete the given `LoLA` file for $\mathcal{N}$ accordingly, and verify whether

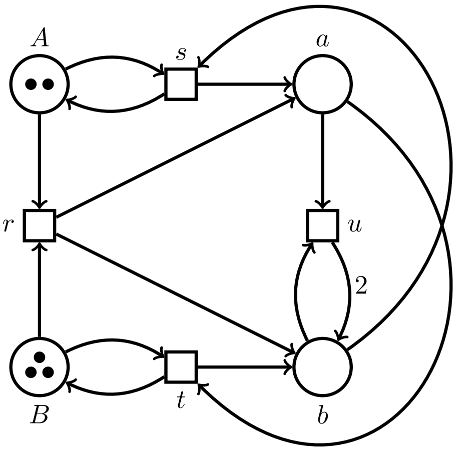
  (i) $(\mathcal{N}, M_0)$ is deadlock-free;

  (ii) a process can be at multiple program locations at the same time;

  (iii) whether both processes can reach their critical sections simultaneously.

**Exercise 1.3**

Consider the following Petri net $\mathcal{N}$ (with weighted arcs):




(a) Describe the preset and postset of each transition and each place.

(b) Give a subnet of $\mathcal{N}$ with two places and one transition.

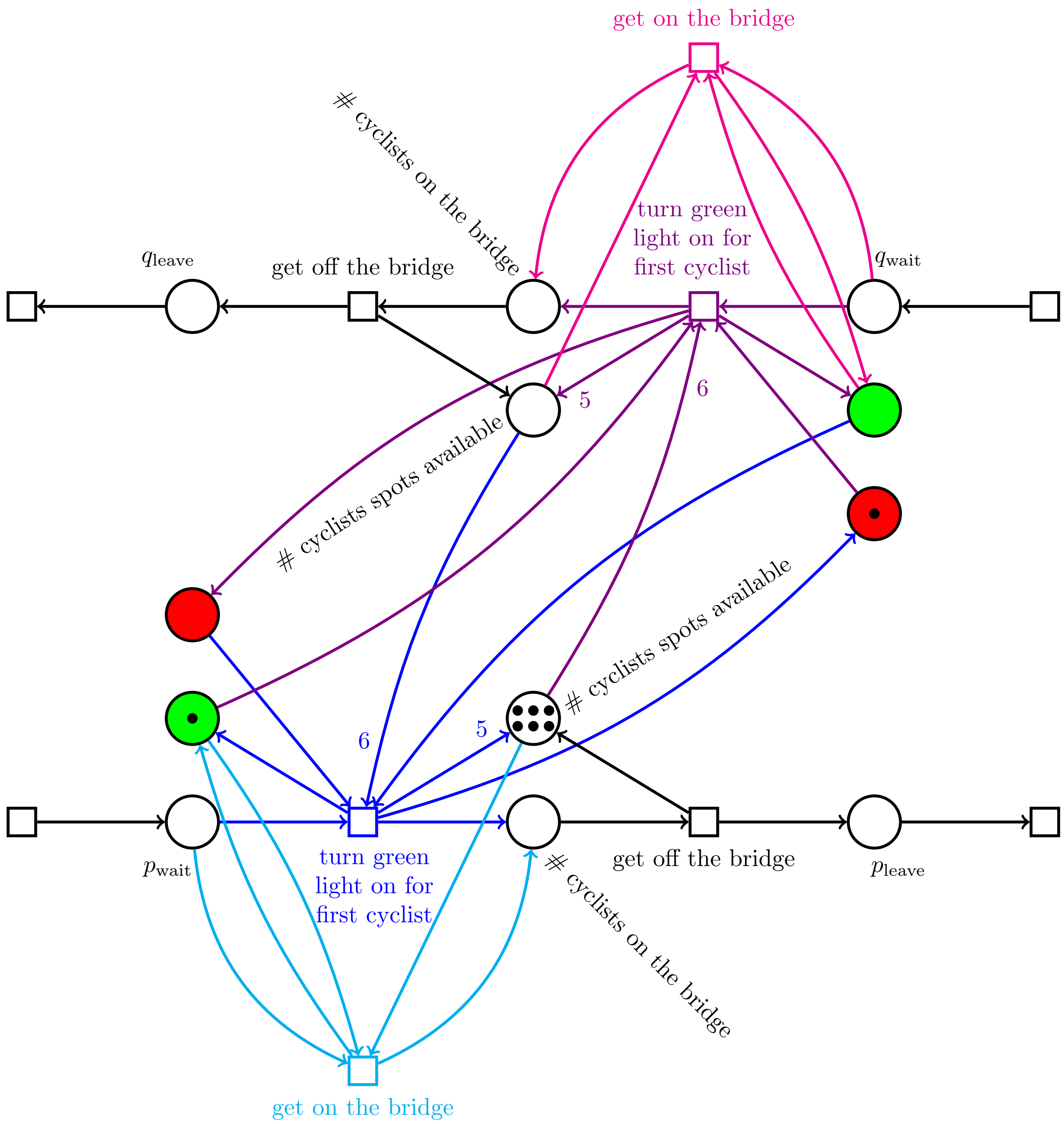
(c) Is $\mathcal{N}$ $k$-bounded for some $k \in \mathbb{N}$? If so, for what $k$? If not, why?

(d) Is $\mathcal{N}$ live? deadlock-free? Justify your answers.

(e) Let $M$ be a marking such that $M(a) = M(b) = 0$. Which dead markings can be reached from $M$? To answer the question, first use `PIPE` to simulate $\mathcal{N}$ from multiple such markings. It is not necessary to prove your answer, but you can try. [Hint: ]

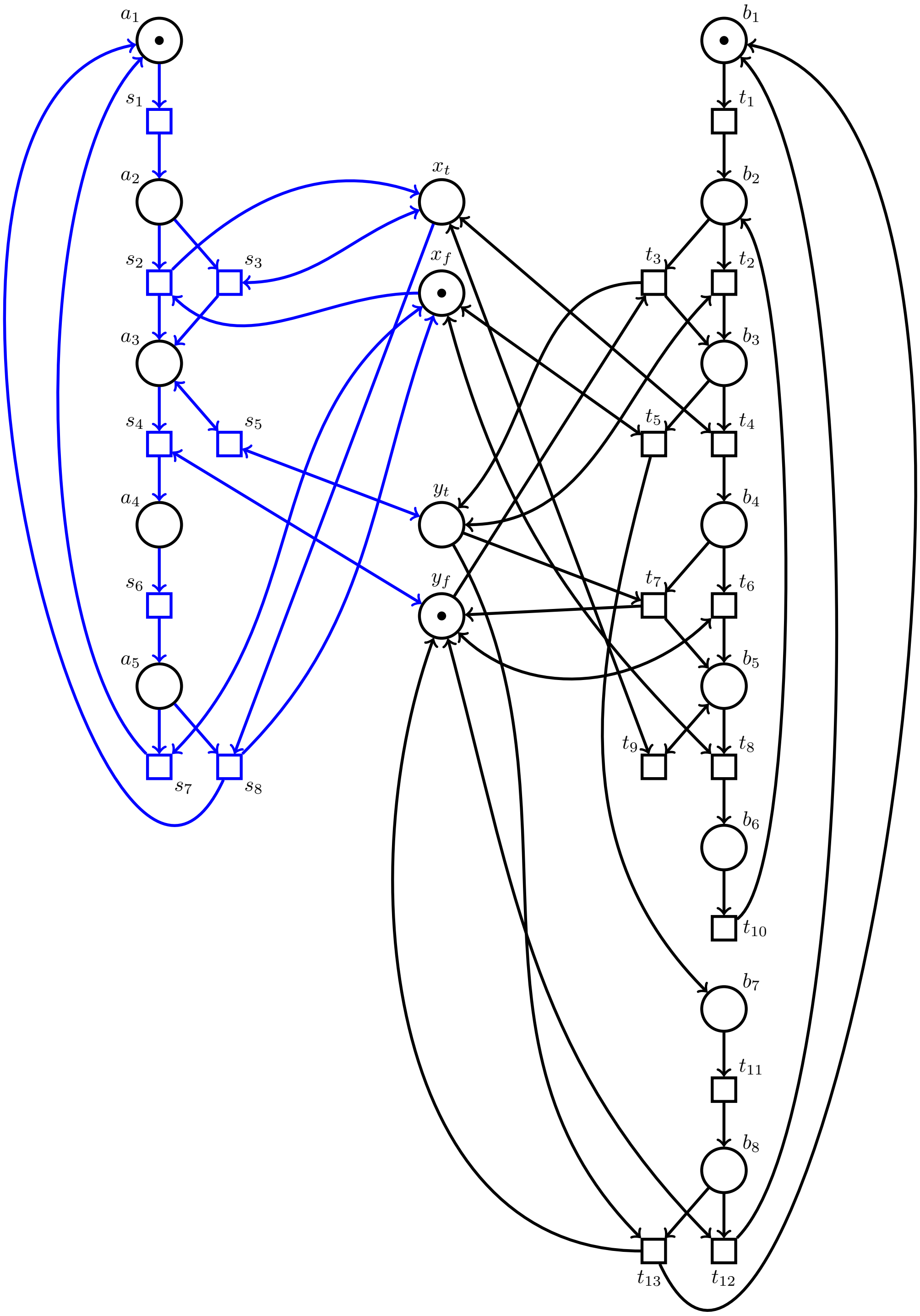
**Solution 1.1**

The bridge can be modeled as follows:



**Solution 1.2**

(a)



(b) (i)
```
> java -jar apt.jar bounded lamport.apt
bounded:  Yes
smallest_K: 1
```

(ii)
```
> java -jar apt.jar strongly_live lamport.apt
strongly_live:  No
```

(c) (i)
```
> lola lamport.lola -f "REACHABLE DEADLOCK"
lola:  result:  no
lola:  The net does not satisfy the given formula.
```

(ii)
```
> lola lamport.lola -f "REACHABLE (a1 + a2 + a3 + a4 + a5 > 1) OR (b1 + b2 + b3 + b4
+ b5 + b6 + b7 + b8 > 1)"
lola:  result:  no
lola:  The net does not satisfy the given formula.
```

(iii)
```
> lola lamport.lola -f "REACHABLE (a4 > 0 AND b7 > 0)"
lola:  result:  no
lola:  The net does not satisfy the given formula.
```

**Solution 1.3**

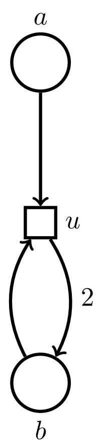
(a) Transitions presets and postsets:

$$^{\bullet}r = \{A, B\}, \qquad r^{\bullet} = \{a, b\},$$
$$^{\bullet}s = \{A, b\}, \qquad s^{\bullet} = \{A, a\},$$
$$^{\bullet}t = \{B, a\}, \qquad t^{\bullet} = \{B, b\},$$
$$^{\bullet}u = \{a, b\}, \qquad u^{\bullet} = \{b\}.$$

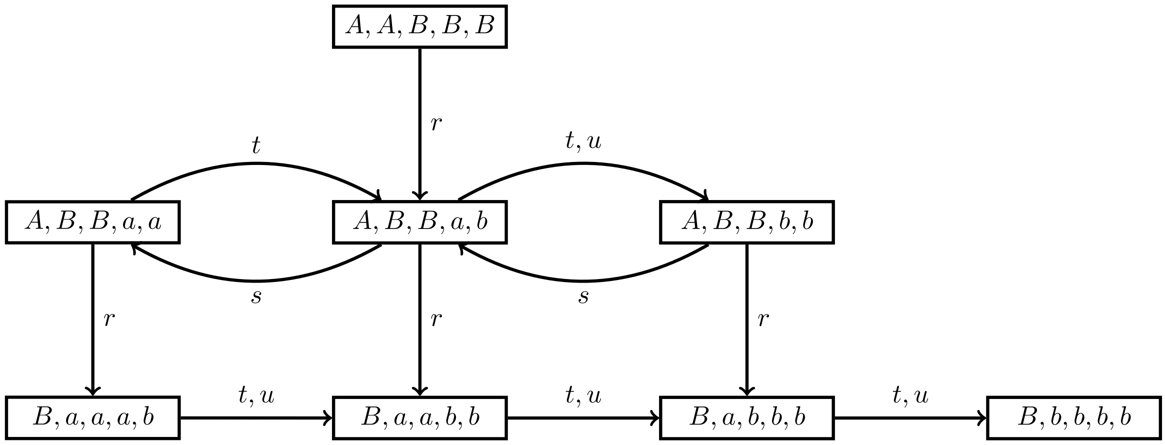
Places presets and postsets:

$$^{\bullet}A = \{s\}, \qquad A^{\bullet} = \{r, s\},$$
$$^{\bullet}B = \{t\}, \qquad B^{\bullet} = \{r, t\},$$
$$^{\bullet}a = \{r, s\}, \qquad a^{\bullet} = \{t, u\},$$
$$^{\bullet}b = \{r, t, u\}, \qquad b^{\bullet} = \{s, u\}.$$

(b) The only such subnet is $\mathcal{N}' = (S', T', W')$ where $S' = \{a, b\}$, $T' = \{u\}$, $W(a, u) = 1$, $W(u, a) = 0$, $W(u, b) = 2$ and $W(b, u) = 1$:

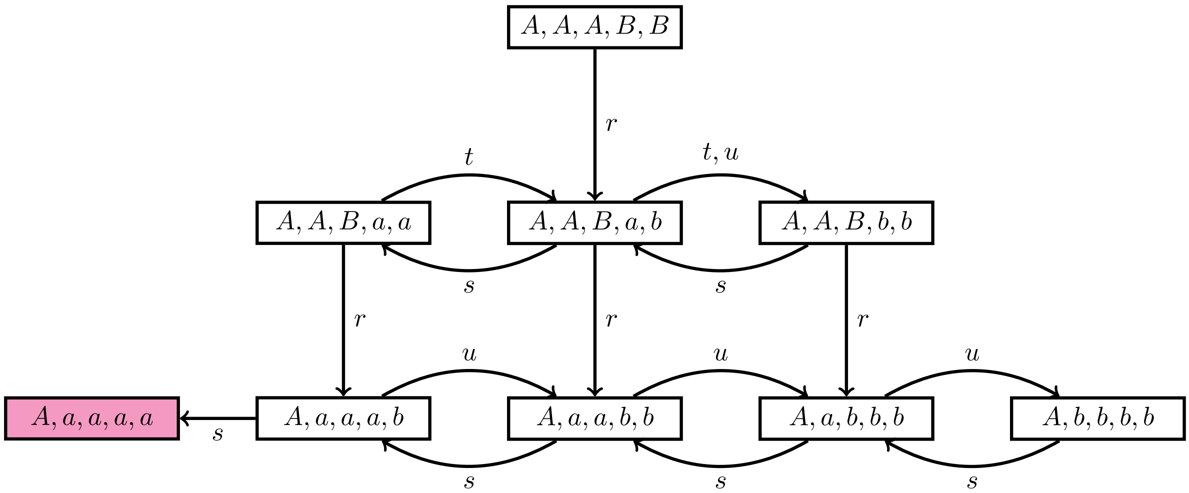


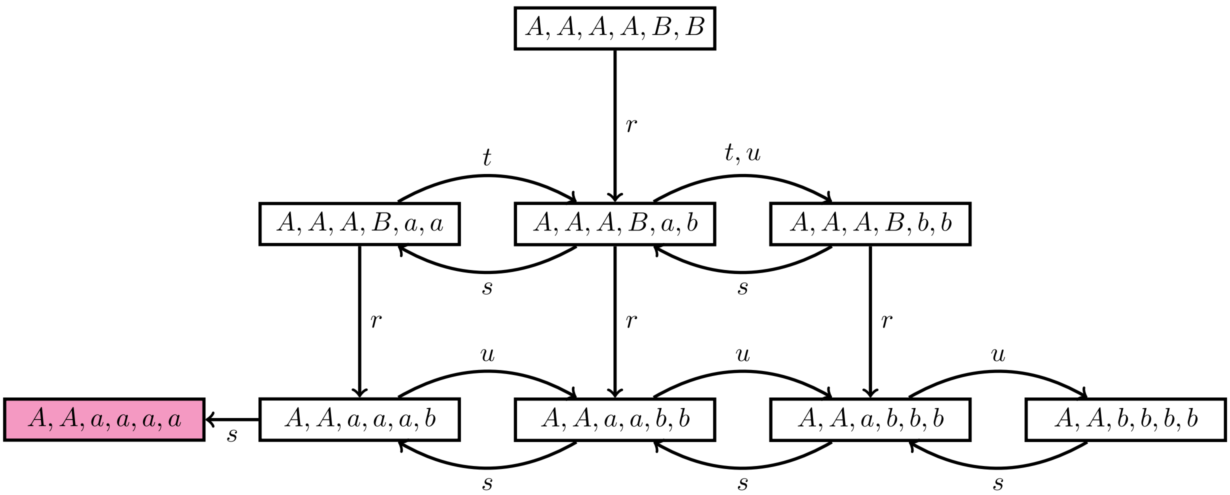
(c) Yes, it is 5-bounded since the initial marking contains 5 tokens and each transition preserves the number of tokens. Alternatively, it can be seen by exploring the reachability graph:




(d) No, it is neither live, nor deadlock-free, since $\{A, A, B, B, B\} \xrightarrow{rrtt} \{B, b, b, b, b\}$ and $\{B, b, b, b, b\}$ is dead.

(e) We already built the reachability graph of such a marking in (c). Let us build reachability graphs for other markings to get further intuition:



It appears that a unique dead marking is reachable. Simulations with PIPE suggest that this marking is:

$$M' = \begin{cases} \{\underbrace{A, A, A, \ldots A,}_{M(A)-M(B) \text{ times}} \underbrace{a, a, a \ldots, a}_{2\cdot M(B) \text{ times}}\} & \text{if } M(A) > M(B), \\ \\ \{\underbrace{B, B, B, \ldots B,}_{M(B)-M(A) \text{ times}} \underbrace{b, b, b \ldots, b}_{2\cdot M(A) \text{ times}}\} & \text{if } M(B) \geq M(A). \end{cases}$$

★ Let us prove this claim for the case where $M(A) > M(B)$. The other case is similar. First note that $M'$ is dead, and that

$$M \xrightarrow{(rs)^{M(A)-M(B)}} M'.$$

Let $M''$ be a dead marking reachable from $M$. We must prove that $M'' = M'$. The only transition that can decrease the amount of tokens in $A$ and $B$ is $r$. Therefore, there exists $\lambda \in \mathbb{N}$ such that $M''(A) = M(A) - \lambda$ and $M''(B) = M(B) - \lambda$. If $\lambda < M(B)$, then $r$ is enabled which contradicts $M''$ being dead. Therefore, $\lambda = M(B)$ which implies that $M''(A) = M(A) - M(B)$ and $M''(B) = 0$. Note that $M''(A) > 0$ since $M(A) > M(B)$. If $M''(b) > 0$, then $s$ is enabled, which contradicts $M''$ being dead. Thus, $M''(b) = 0$. Observe that $\mathcal{N}$ is conservative, i.e. all transitions leave the number of tokens unchanged. Therefore,

$$\begin{aligned} M''(a) &= [M(A) + M(B) + M(a) + M(b)] - [M''(A) + M''(B) - M''(b)] \\ &= [M(A) + M(B)] - M''(A) \\ &= [M(A) + M(B)] - [M(A) - M(B)] \\ &= 2 \cdot M(B). \end{aligned}$$

□

# References

[1] Wil van der Aalst, Massimiliano de Leoni, Boudewijn van Dongen, and Christian Stahl. Course business information systems: exercises, 2015. Available at http://wwwis.win.tue.nl/~wvdaalst/old/courses/BIScourse/exercise-bundle-BIS-2015.pdf.