Crash course – Petri nets
General definitions
Coverability

Xiaoxi He

Basic definitions

- State ⇔ Marking (Do not confuse states and places !!!)
- **Pre** and **Post** sets for transitions : Pre set: $\bullet t := \{p \mid (p,t) \in F\}$

  Post set: $t\bullet := \{p \mid (t,p) \in F\}$,

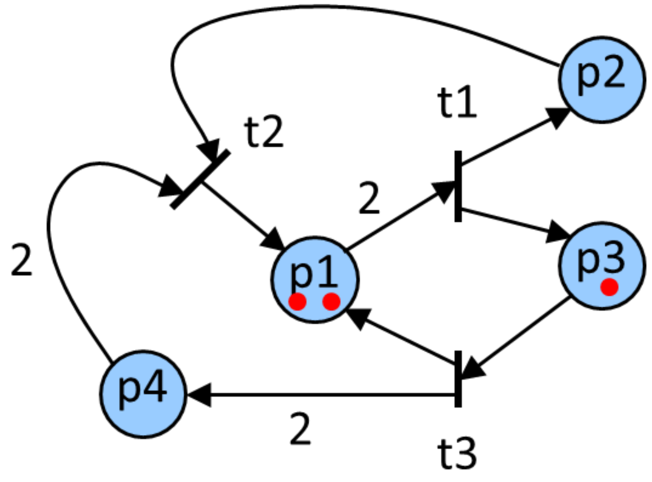
  (likewise for places)
- Upstream $\boldsymbol{W}^-$ and Downstream $\boldsymbol{W}^+$ incidence matrices:

$$W^- = \underbrace{\begin{bmatrix} & \cdots & \\ \vdots & \ddots & \vdots \\ & \cdots & \end{bmatrix}}_{\text{Transitions}} \Big\} \text{Places} \quad , W^-(i,j) = \begin{cases} w & \text{if } p_i \in \bullet t_j \text{ and has weight } w \\ 0 & \text{otherwise} \end{cases}$$

- Incidence matrix: $A = W^+ - W^-$

Basic definitions

- *Token game*
  From a marking M_0, for a firing sequence vector T, the marking obtained is



$$M = M_0 + A \cdot T$$

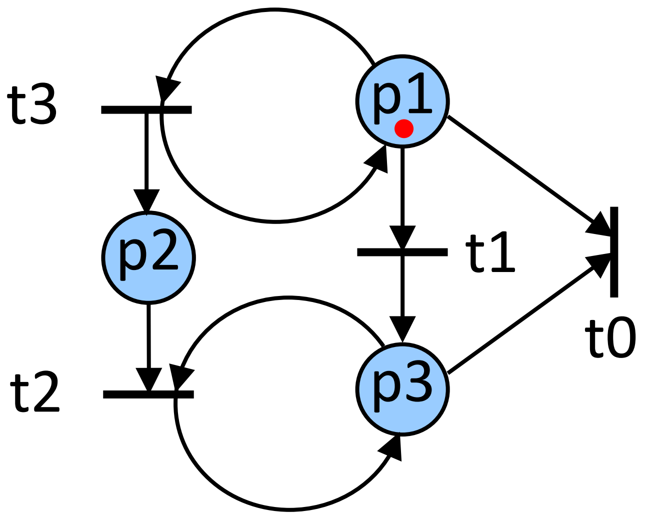
1 firing of t_3

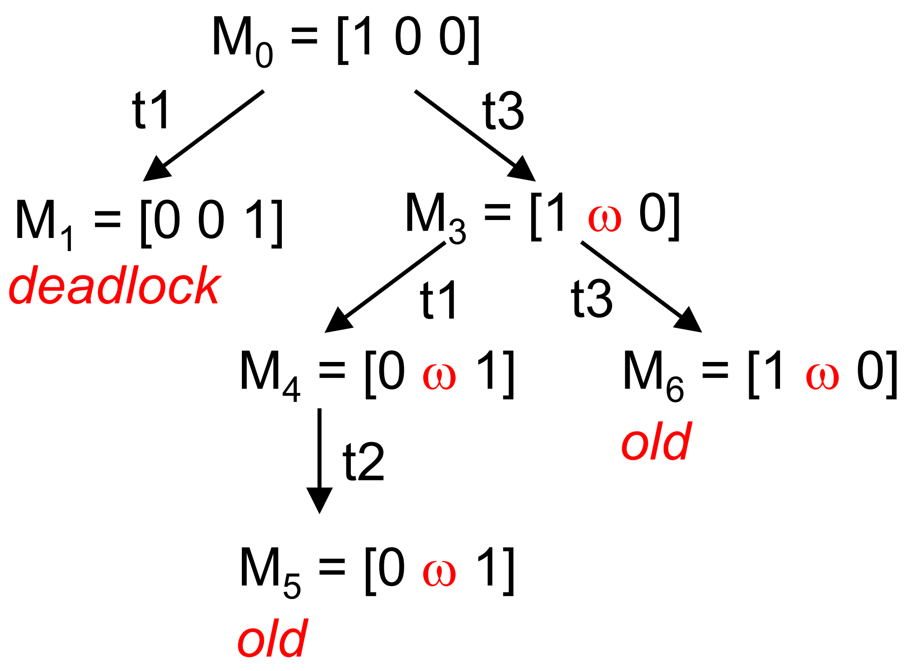
$$\begin{bmatrix} 3 \\ 0 \\ 0 \\ 2 \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \\ 1 \\ 0 \end{bmatrix} + \begin{bmatrix} -2 & 1 & 1 \\ 1 & -1 & 0 \\ 1 & 0 & -1 \\ 0 & -2 & 2 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

BEWARE ! All firing sequences are not necessary allowed by the net…

Coverability Tree

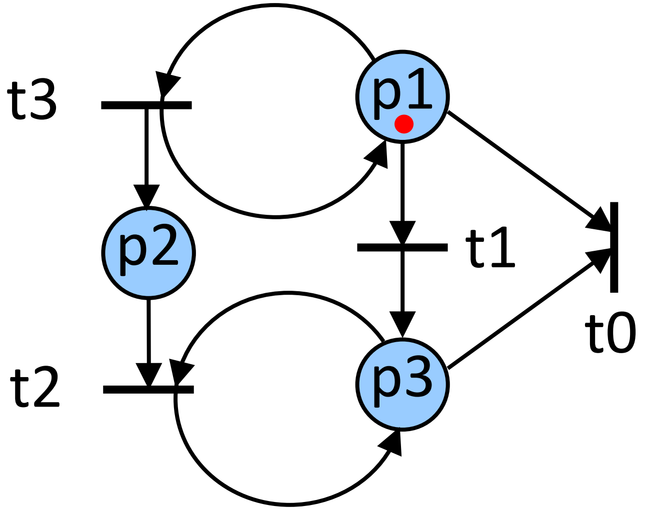
- **Question:** What token distributions are reachable?
- **Problem:** There might be infinitely many reachable markings, but we must avoid an infinite tree.
- **Solution:** Introduce a special symbol ω to denote an arbitrary number of tokens:

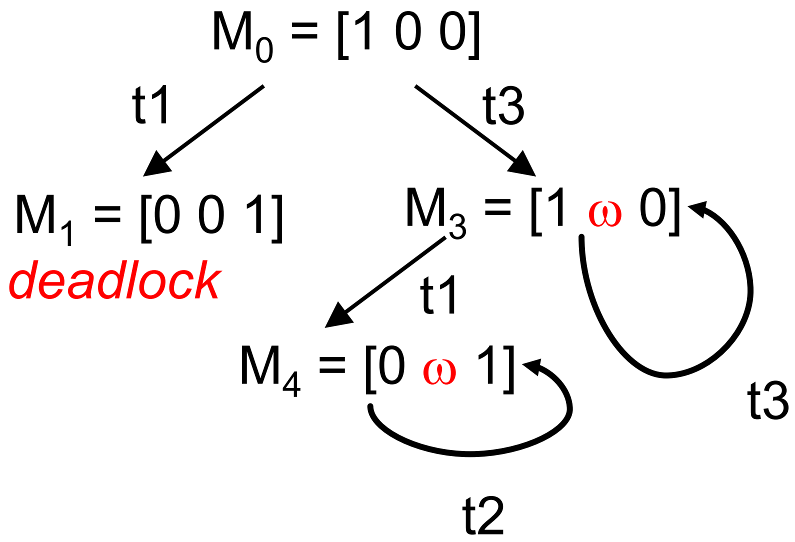




Coverability Graph -> Merge nodes

- **Question:** What token distributions are reachable?
- **Problem:** There might be infinitely many reachable markings, but we must avoid an infinite tree.
- **Solution:** Introduce a special symbol ω to denote an arbitrary number of tokens:



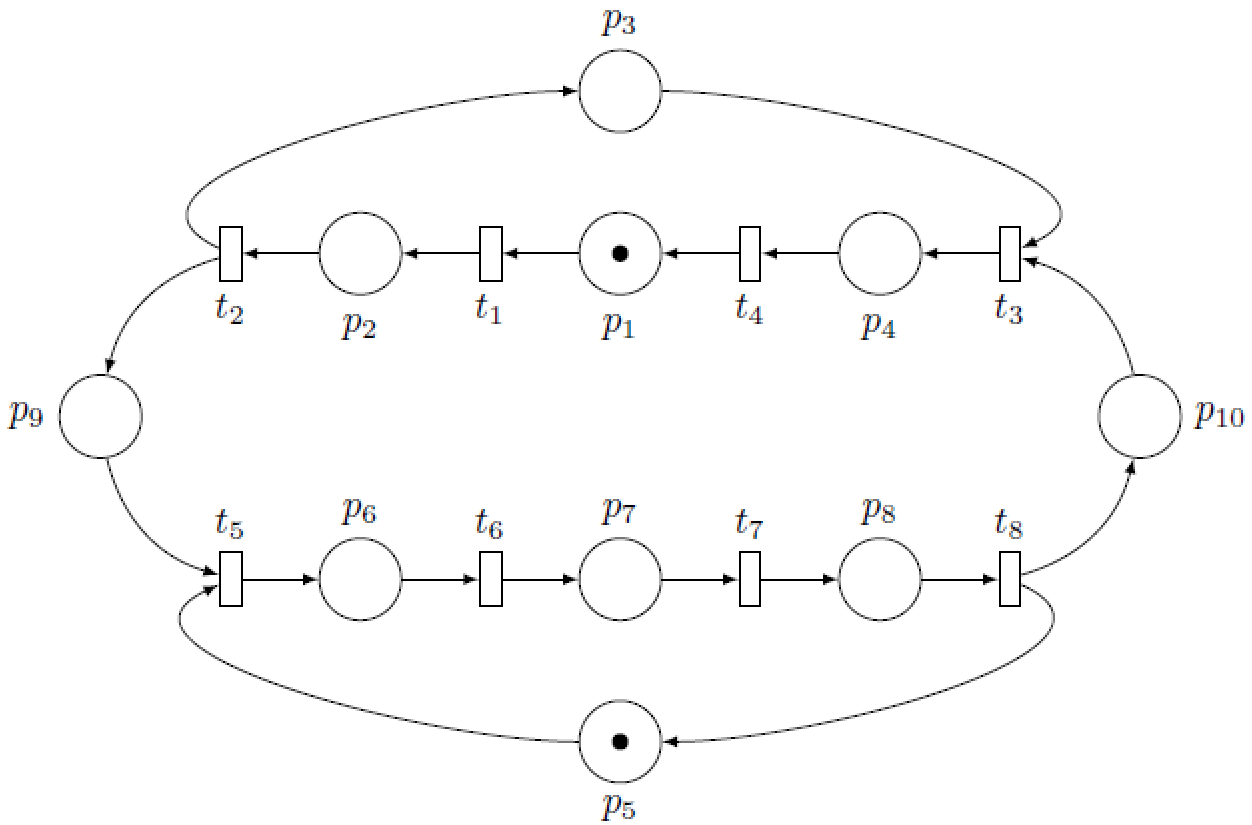


1 Structural Properties of Petri Nets and Token Game

a) $\bullet t_5 = \{p_5, p_9\}$, $\quad t_5\bullet = \{p_6\}$

$\bullet t_8 = \{p_8\}$, $\quad t_8\bullet = \{p_{10}, p_5\}$

$\bullet p_3 = \{t_2\}$, $\quad p_3\bullet = \{t_3\}$



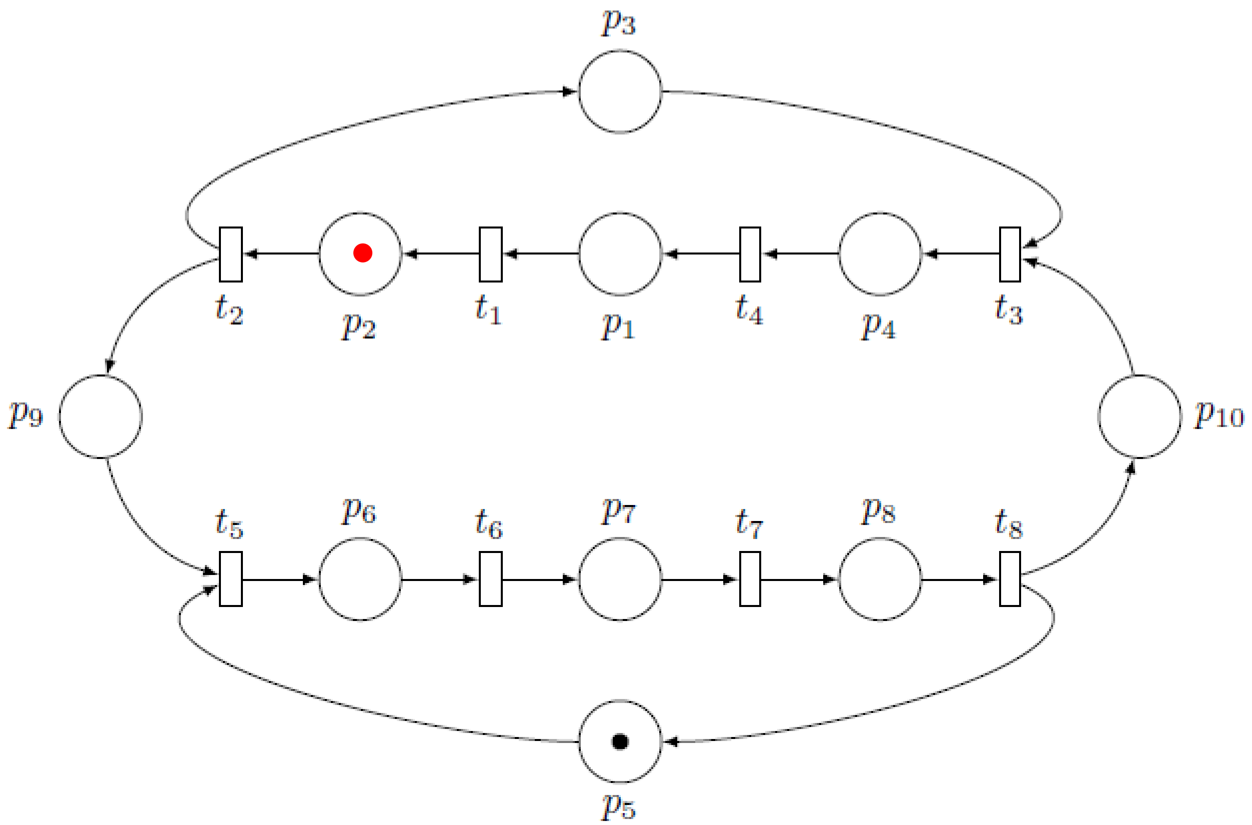
1 Structural Properties of Petri Nets and Token Game

a) $\bullet t_5 = \{p_5, p_9\}, \quad t_5\bullet = \{p_6\}$

$\bullet t_8 = \{p_8\}, \quad t_8\bullet = \{p_{10}, p_5\}$

$\bullet p_3 = \{t_2\}, \quad p_3\bullet = \{t_3\}$

b) t1 fires… t2 fires…
→ t5 is enabled
→ t3 is not



1 Structural Properties of Petri Nets and Token Game

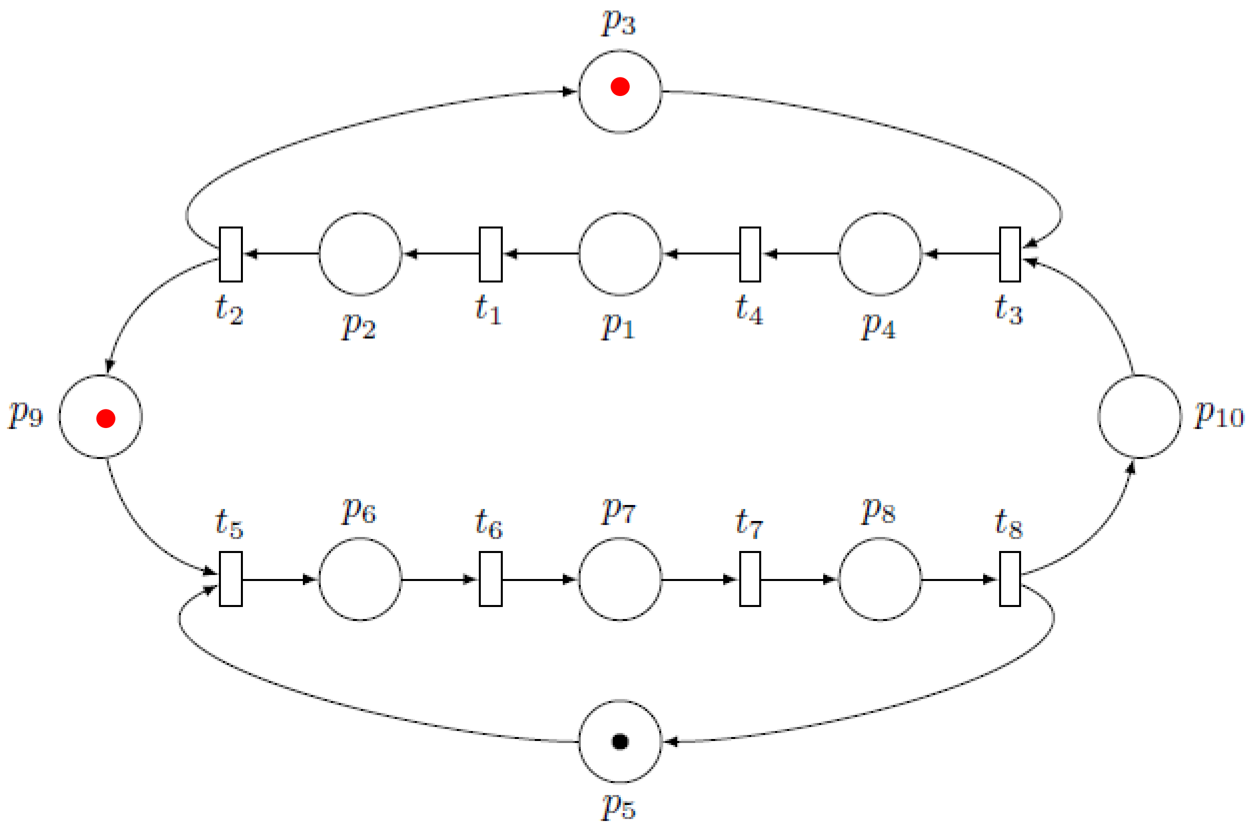
a) $\bullet t_5 = \{p_5, p_9\}$, $t_5\bullet = \{p_6\}$

$\bullet t_8 = \{p_8\}$, $t_8\bullet = \{p_{10}, p_5\}$

$\bullet p_3 = \{t_2\}$, $p_3\bullet = \{t_3\}$

b) t1 fires… t2 fires…
→ t5 is enabled
→ t3 is not

c) 3 tokens in the net after t2 has been fired.

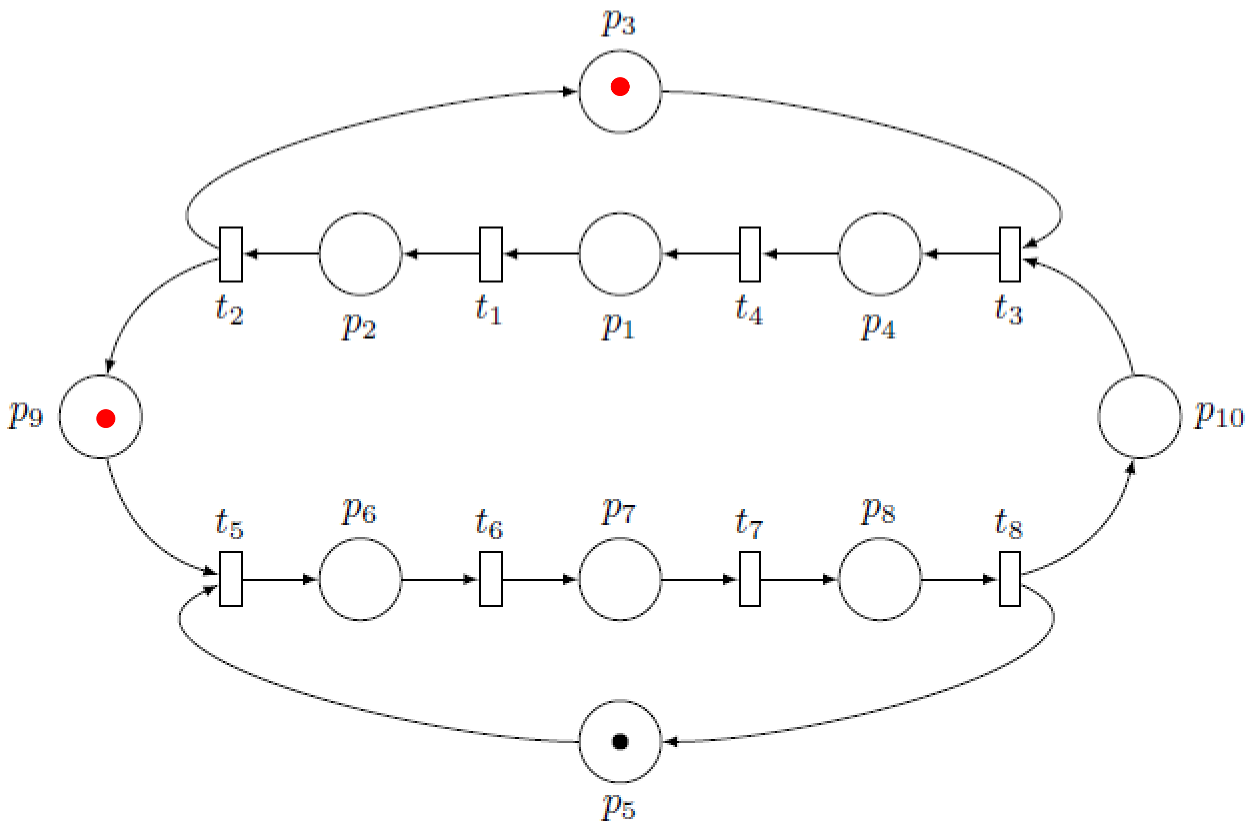


1 Structural Properties of Petri Nets and Token Game

a) $\bullet t_5 = \{p_5, p_9\}$, $\quad t_5\bullet = \{p_6\}$

$\bullet t_8 = \{p_8\}$, $\quad t_8\bullet = \{p_{10}, p_5\}$

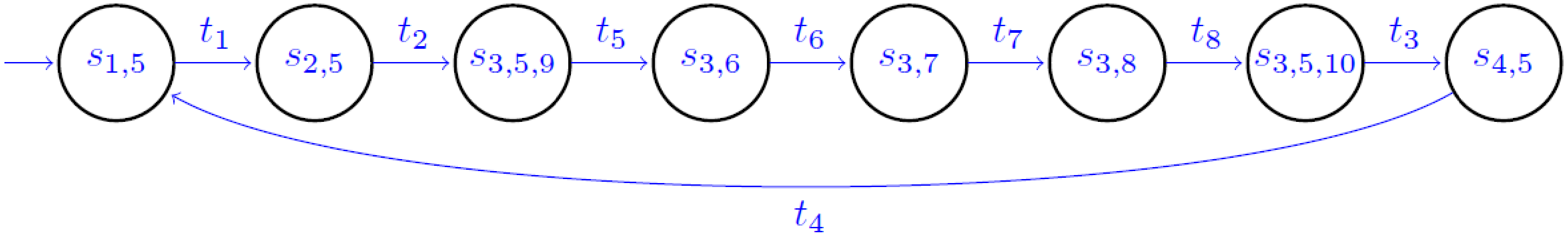
$\bullet p_3 = \{t_2\}$, $\quad p_3\bullet = \{t_3\}$



b) t1 fires… t2 fires…
→ t5 is enabled
→ t3 is not

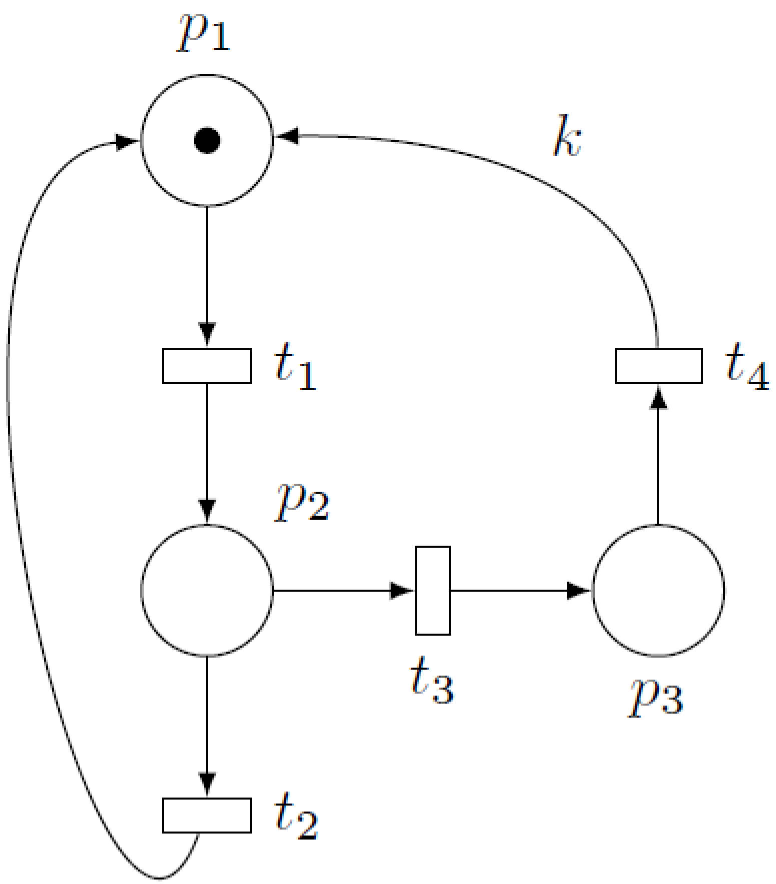
c) 3 tokens in the net after t2 has been fired.

d)



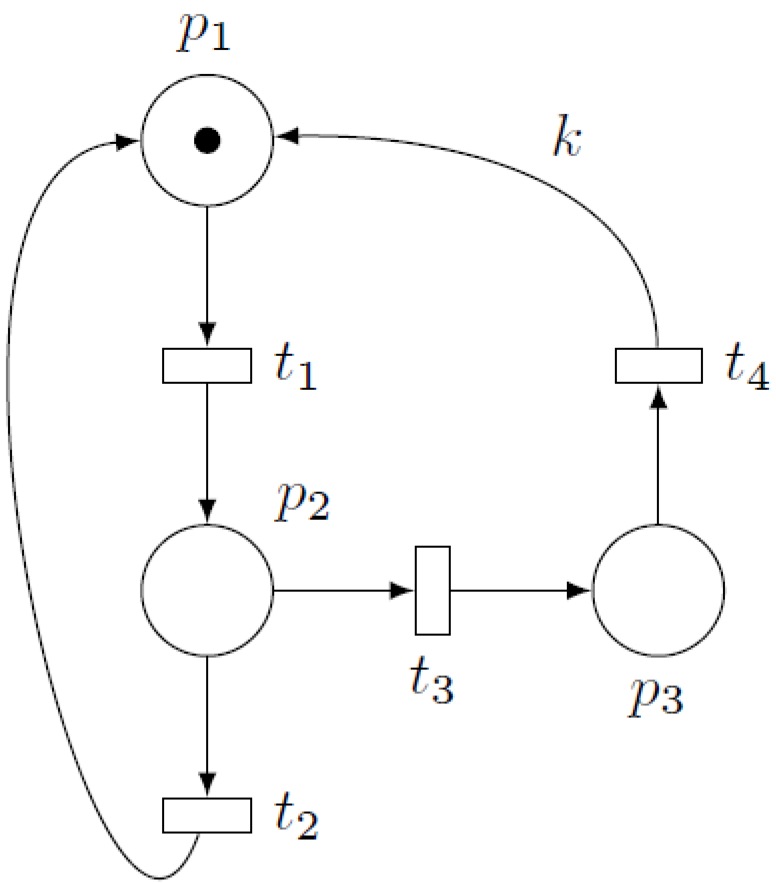
2 Basic Properties of Petri Nets

- For which k is the net bounded?
- For which k is the net deadlock free



2 Basic Properties of Petri Nets

- Bounded for any $k \leq 1$
- Deadlock-free if $k \geq 1$



Crash course – Petri nets

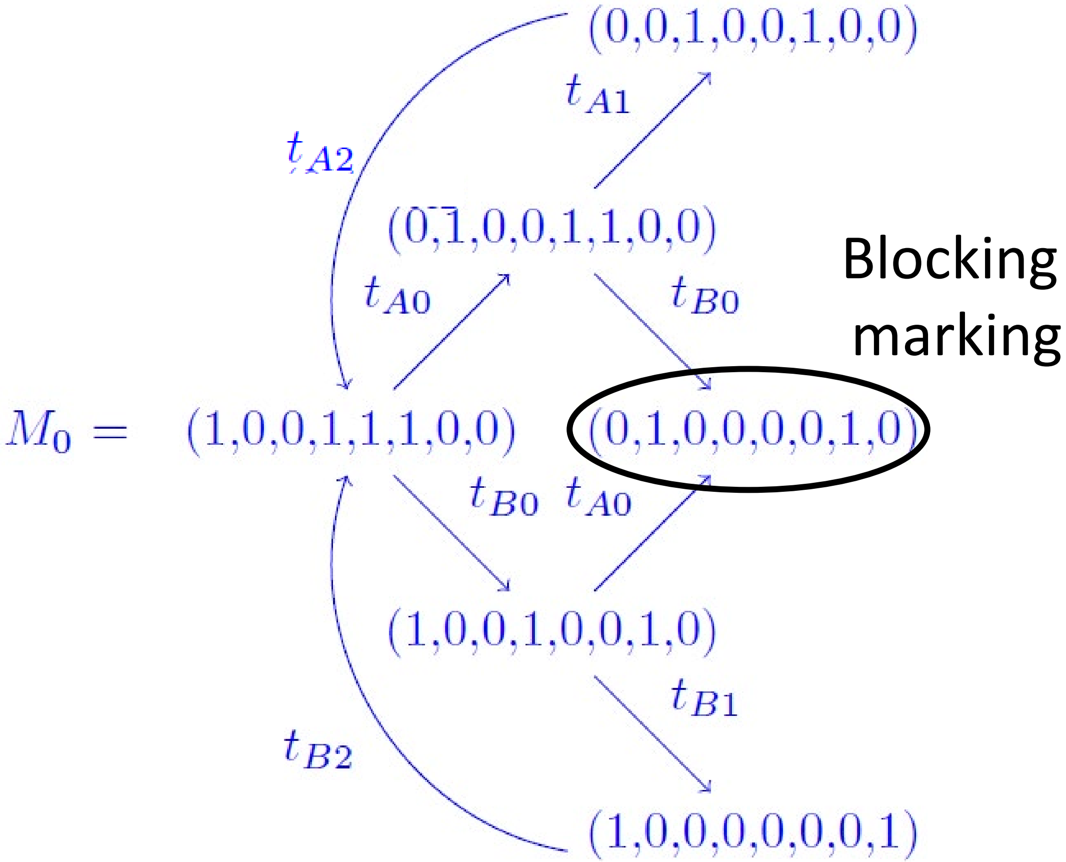
General definitions

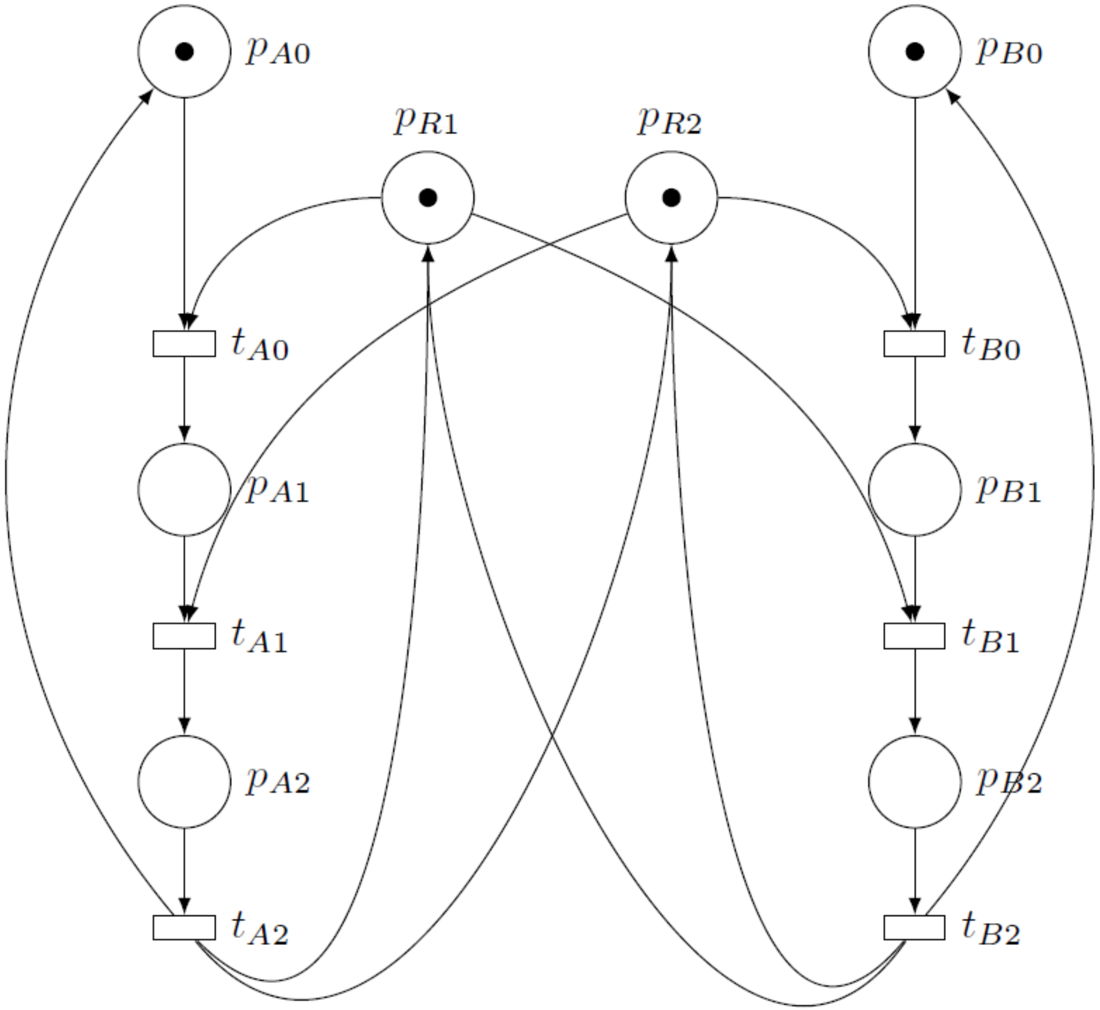
Coverability

Your turn to work!

3 Identifying a deadlock

a) Example of blocking sequence:
$t_{A0} t_{B0}$



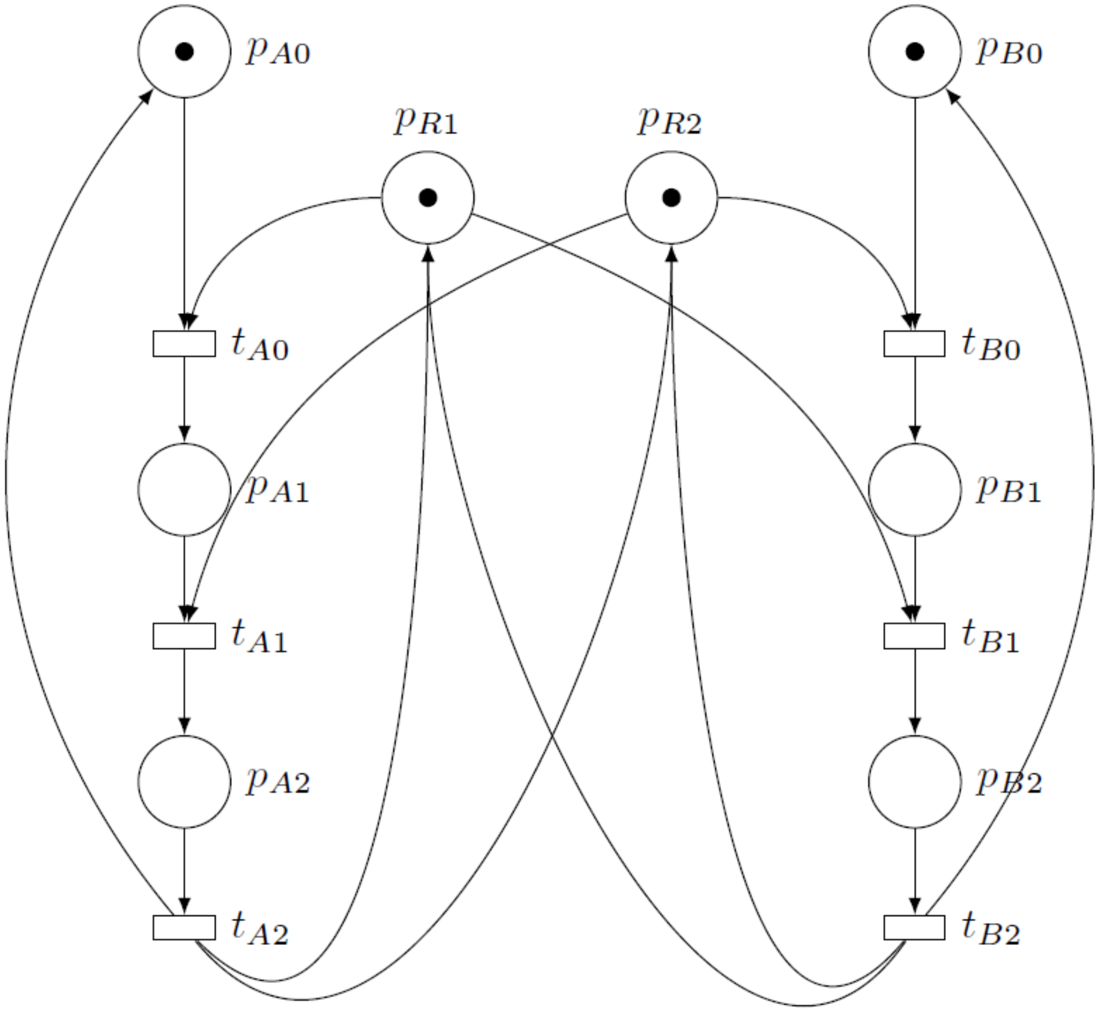


3 Identifying a deadlock

b) Just read it from the graph

$$W^+ = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}, W^- = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A = W^+ - W^- = \begin{bmatrix} -1 & 0 & 1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & -1 & 1 \\ 0 & -1 & 1 & -1 & 0 & 1 \\ 0 & 0 & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix}$$

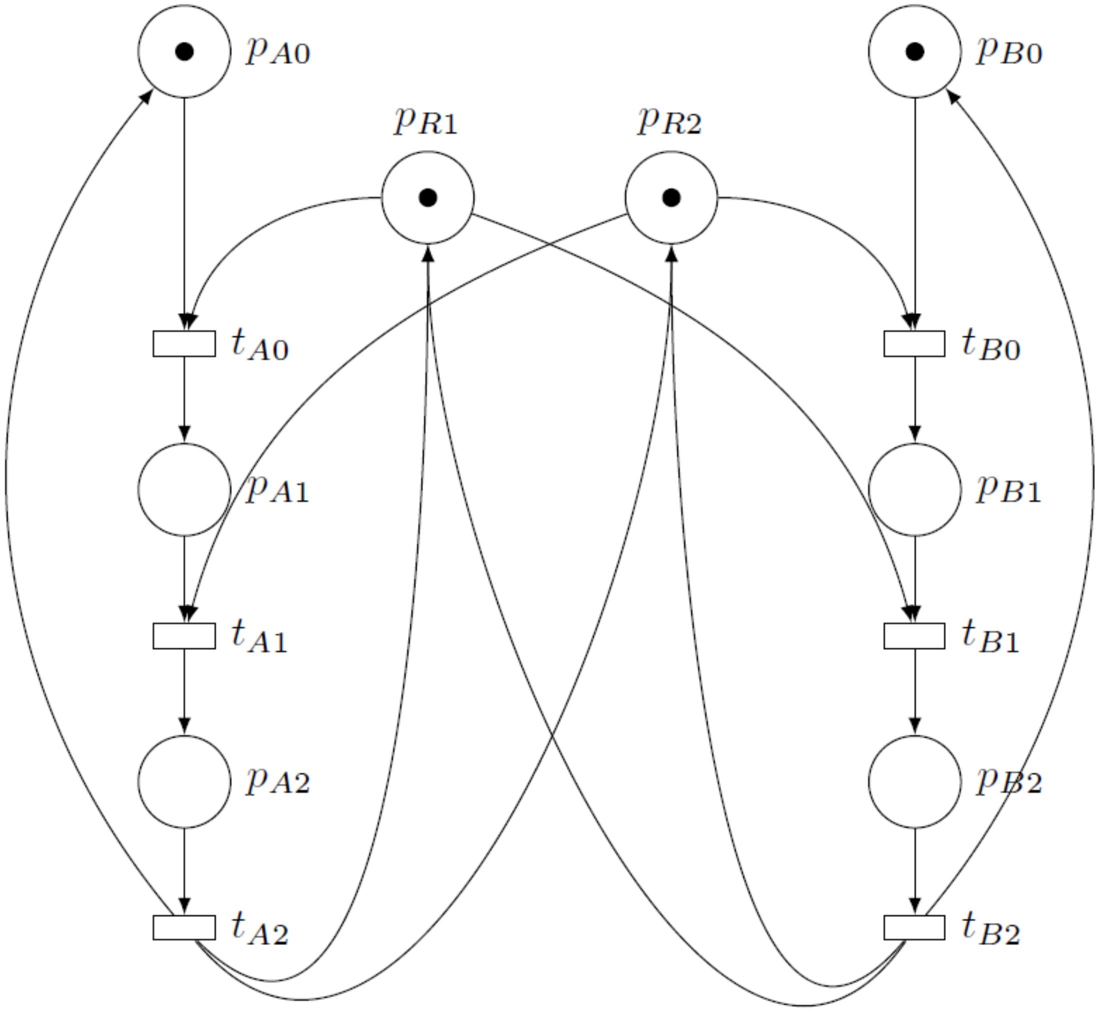


3 Identifying a deadlock

b) Just read it from the graph

$$W^+ = \left[\begin{array}{ccc|ccc} 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ \hline 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{array}\right], \quad W^- = \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ \hline 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right]$$

$$A = W^+ - W^- = \left[\begin{array}{ccc|ccc} -1 & 0 & 1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ \hline -1 & 0 & 1 & 0 & -1 & 1 \\ 0 & -1 & 1 & -1 & 0 & 1 \\ \hline 0 & 0 & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \end{array}\right]$$



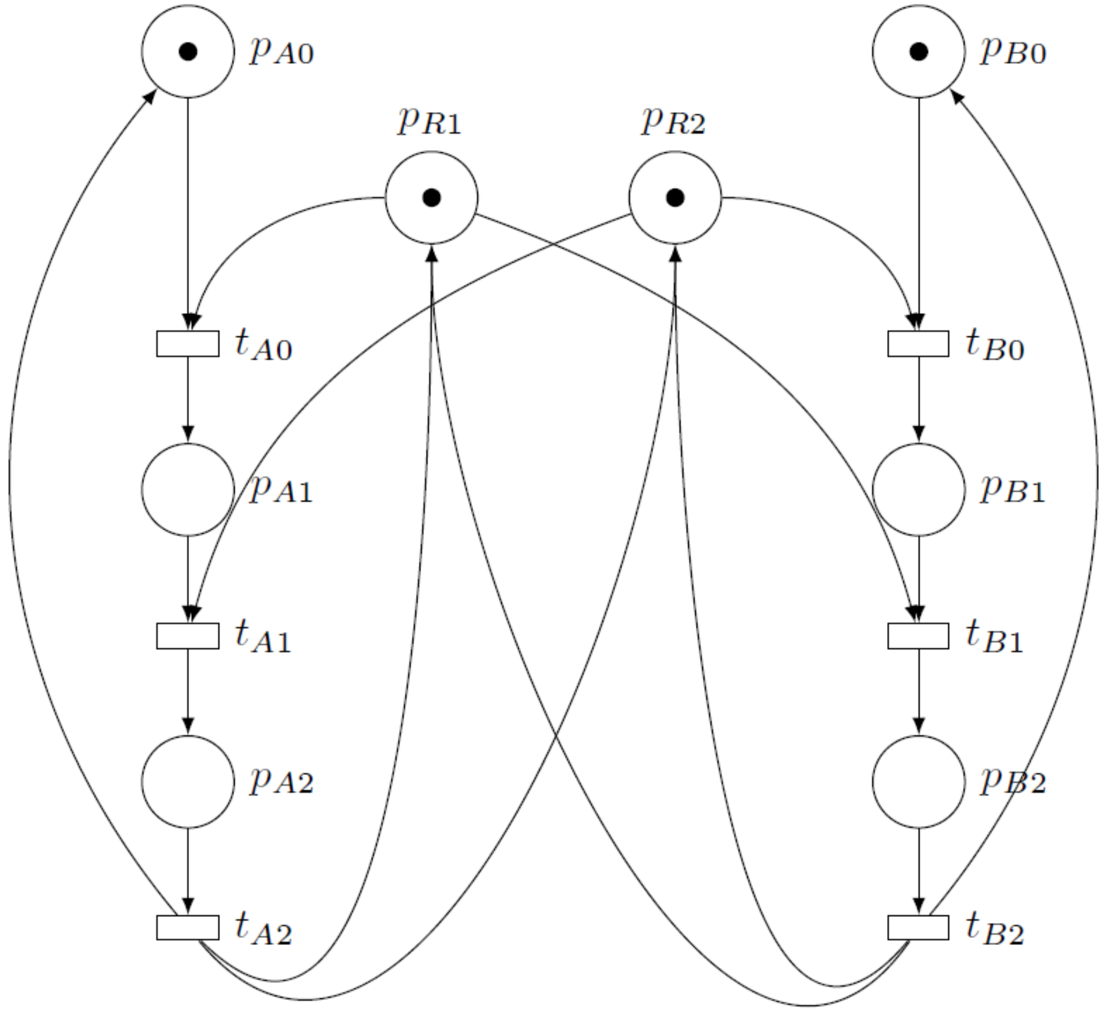
3 Identifying a deadlock

b) Just read it from the graph

$$A = W^{+} - W^{-} = \begin{bmatrix} -1 & 0 & 1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & -1 & 1 \\ 0 & -1 & 1 & -1 & 0 & 1 \\ 0 & 0 & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix}$$

$$M_{deadlock} = M_0 + A \cdot \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -1 \\ 1 \\ 0 \\ -1 \\ -1 \\ -1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

$t_{A0} t_{B0}$

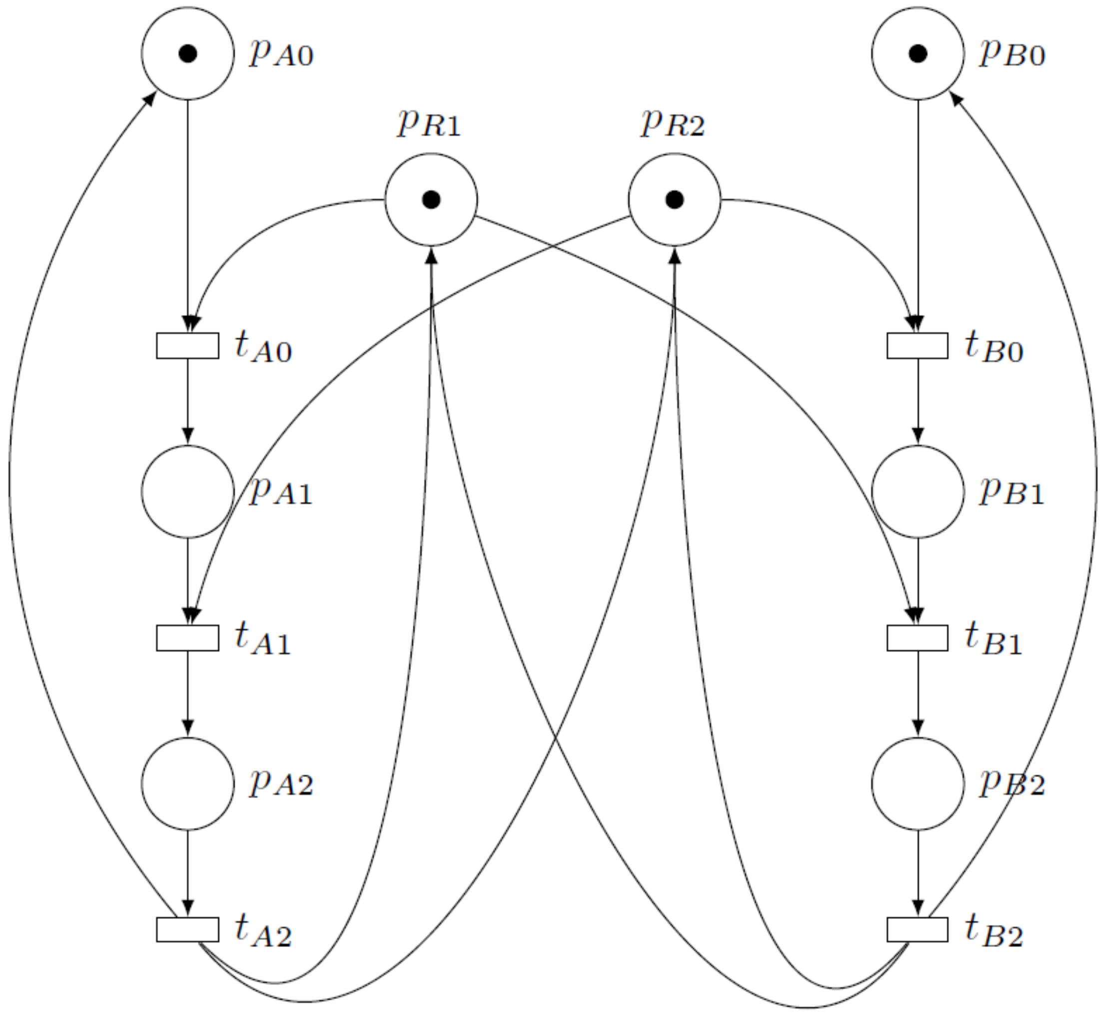


3 Identifying a deadlock

c) Proving marking is blocking

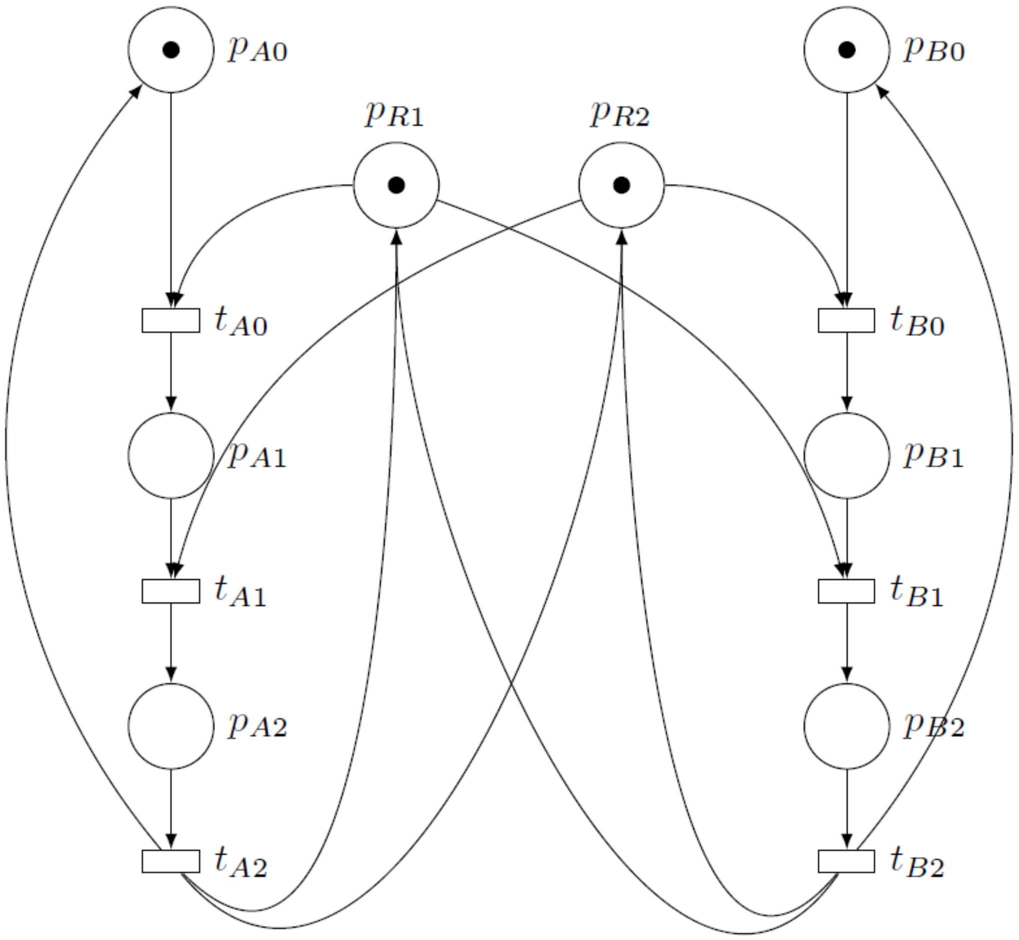
$$W^- = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad M_{deadlock} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

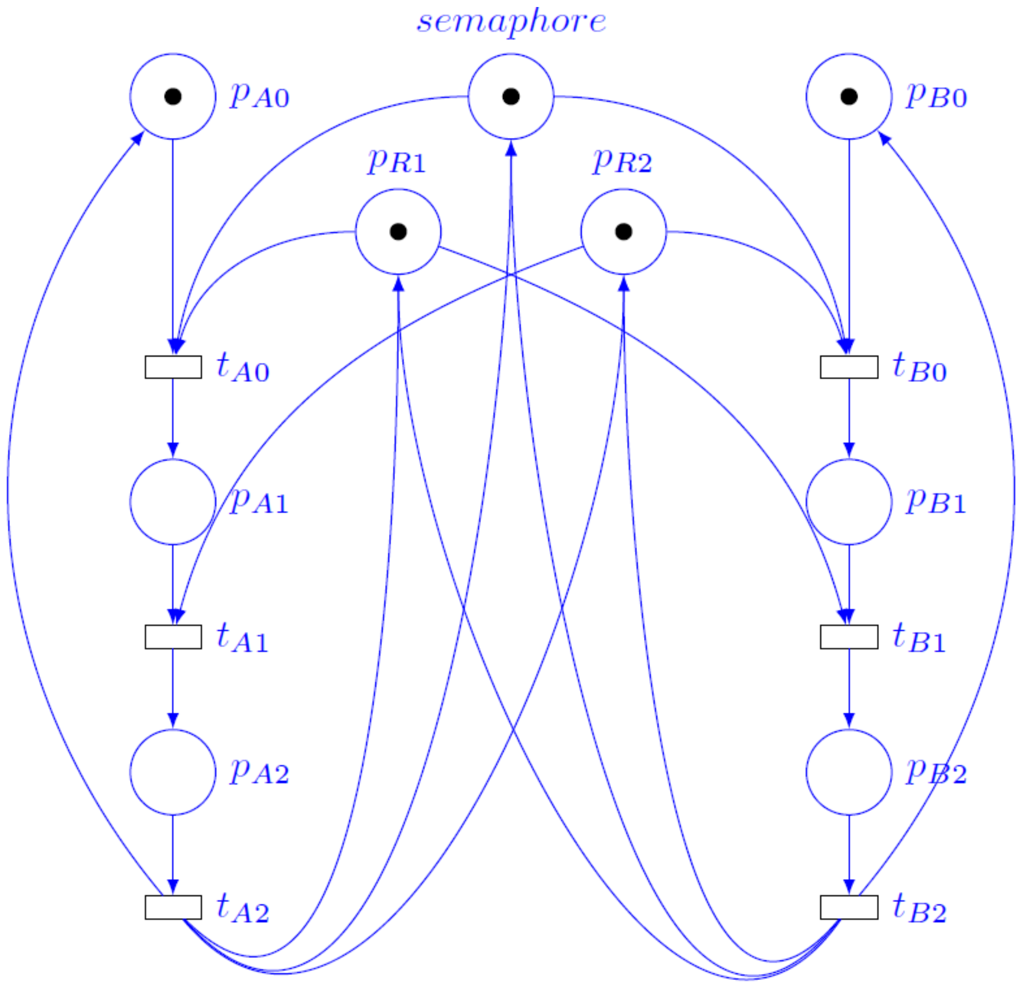
Do not cover any column



3 Identifying a deadlock

d) Correct by adding a semaphore

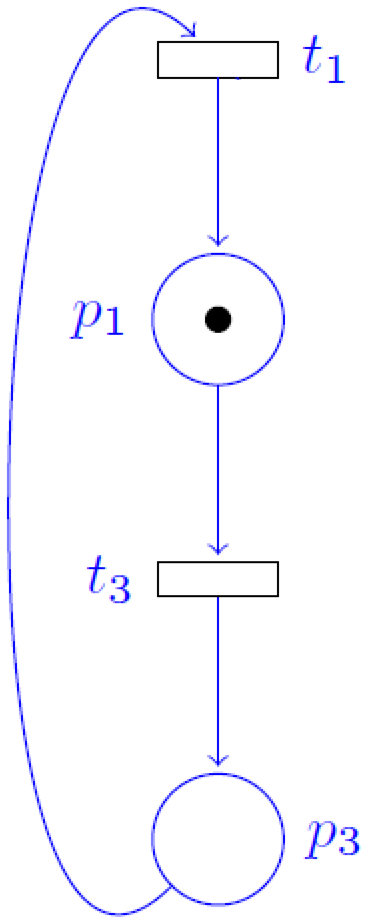




4 From mutual exclusion to starvation

a) Derive the net from the specification

1. One process executes its program.

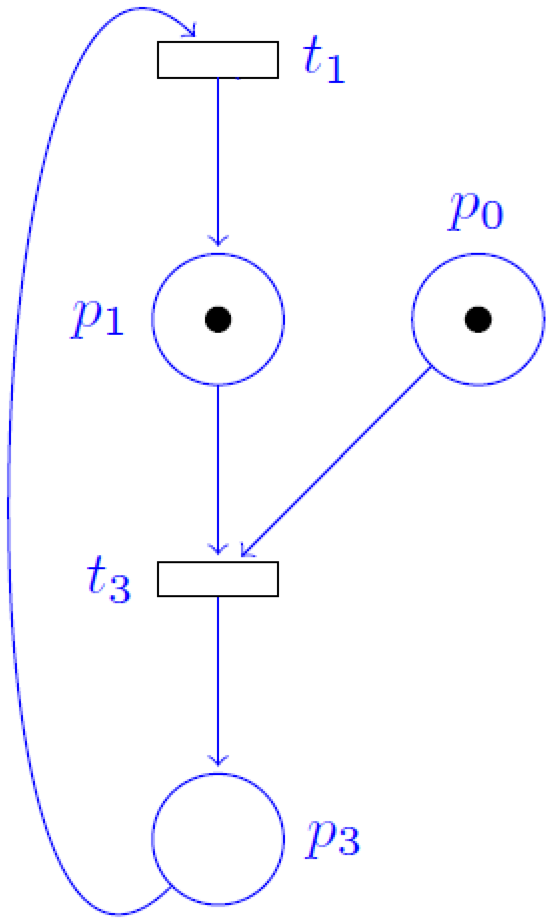


4 From mutual exclusion to starvation

a) Derive the net from the specification

1. One process executes its program.

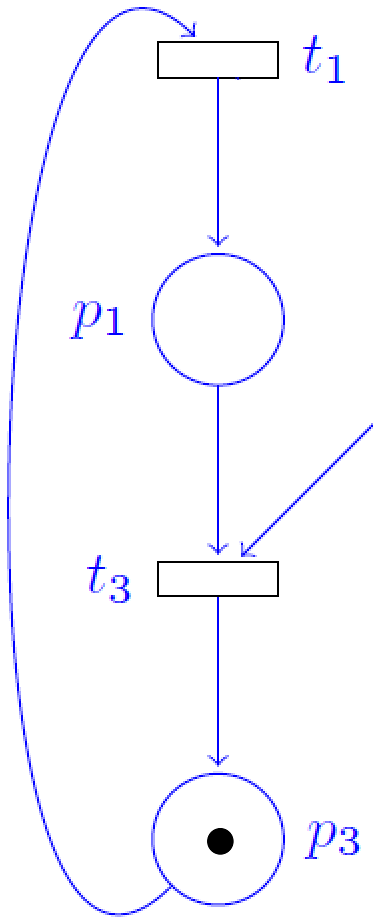
2. In order to enter the critical section, the mutex value must be 1 (i.e. the mutex is available).



4 From mutual exclusion to starvation

a) Derive the net from the specification

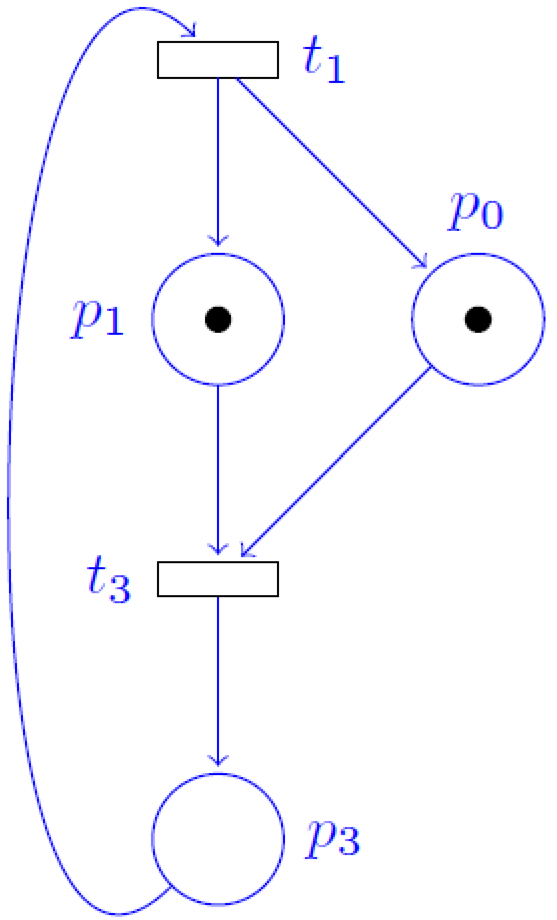
1. One process executes its program.
2. In order to enter the critical section, the mutex value must be 1 (i.e. the mutex is available).
3. If this is the case, the process sets the mutex to 0 and executes its critical section.



4 From mutual exclusion to starvation

a) Derive the net from the specification

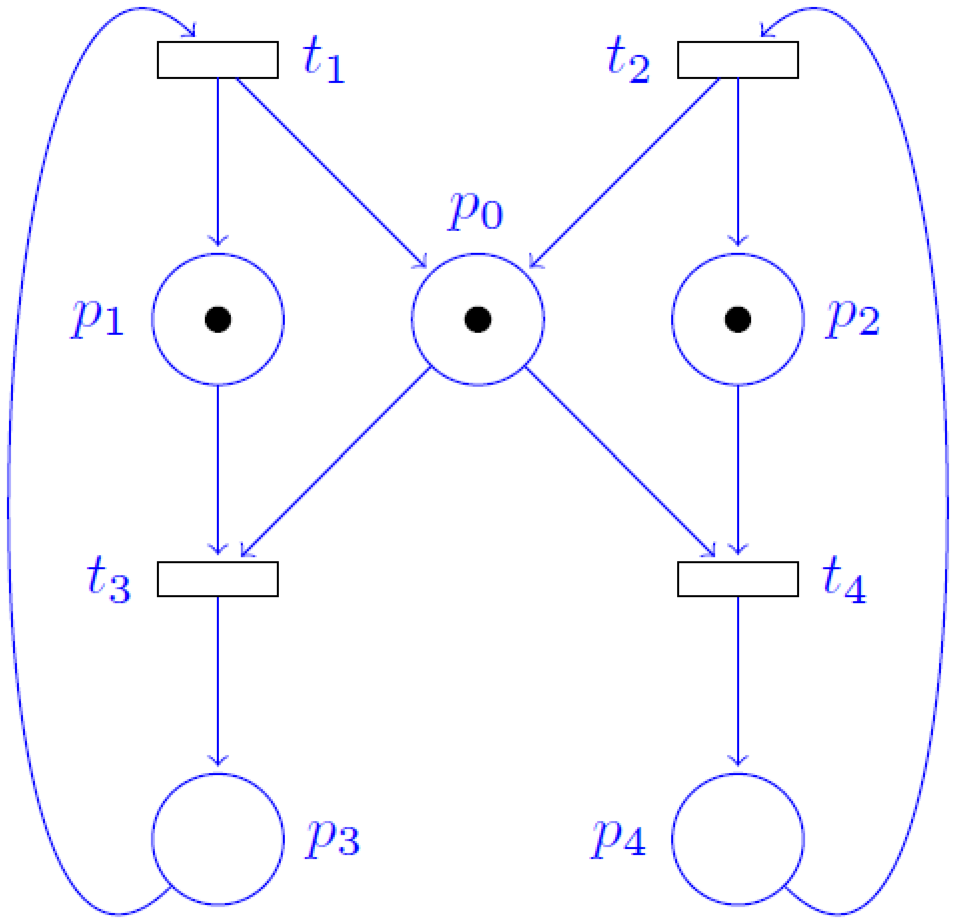
1. One process executes its program.
2. In order to enter the critical section, the mutex value must be 1 (i.e. the mutex is available).
3. If this is the case, the process sets the mutex to 0 and executes its critical section.
4. When it is done, it resets the mutex to 1 and enters an uncritical section.
5. It loops back to start.



4 From mutual exclusion to starvation

a) Derive the net from the specification

1. One process executes its program.
2. In order to enter the critical section, the mutex value must be 1 (i.e. the mutex is available).
3. If this is the case, the process sets the mutex to 0 and executes its critical section.
4. When it is done, it resets the mutex to 1 and enters an uncritical section.
5. It loops back to start.



4 From mutual exclusion to starvation

b) How to avoid starvation ?

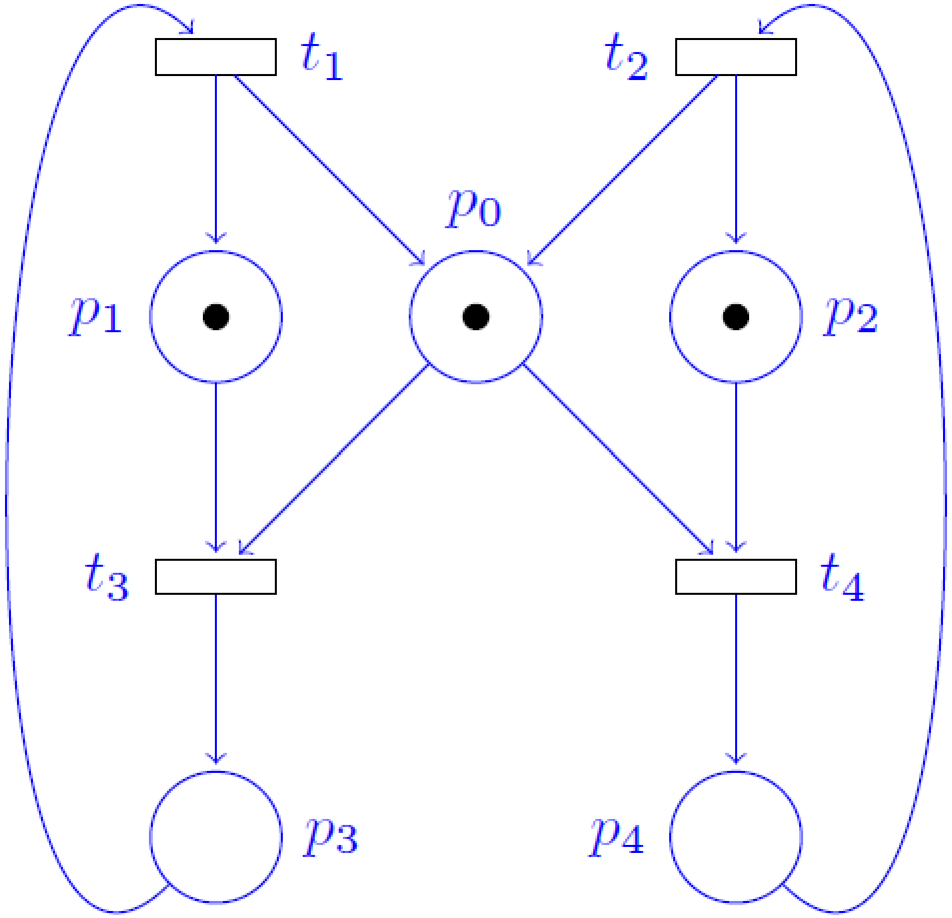
Add a semaphore/resource kind of place

→ Consumed by one process

→ Generated by the other process

To avoid starvation in both direction, you need two of such places

The total number of tokens in those places in the maximal number of possible execution in a row.



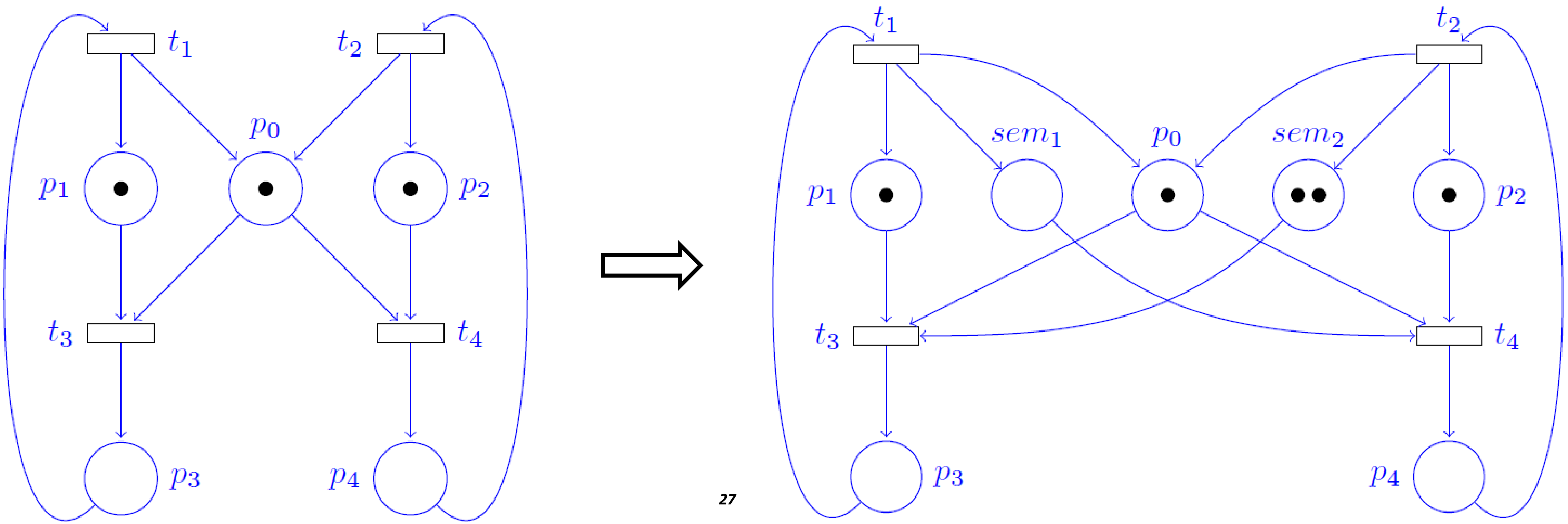
4 From mutual exclusion to starvation

b) How to avoid starvation ? Add a semaphore/resource kind of place

→ Consumed by one process → Generated by the other process

To avoid starvation in both direction, you need two of such places

The total number of tokens in those places in the maximal number of possible execution in a row.



4 From mutual exclusion to starvation

c) What’s the problem with this?

→ If B does not executes anymore, A is forced to stop as well. And vice versa.

What would you propose as specification?

For example:

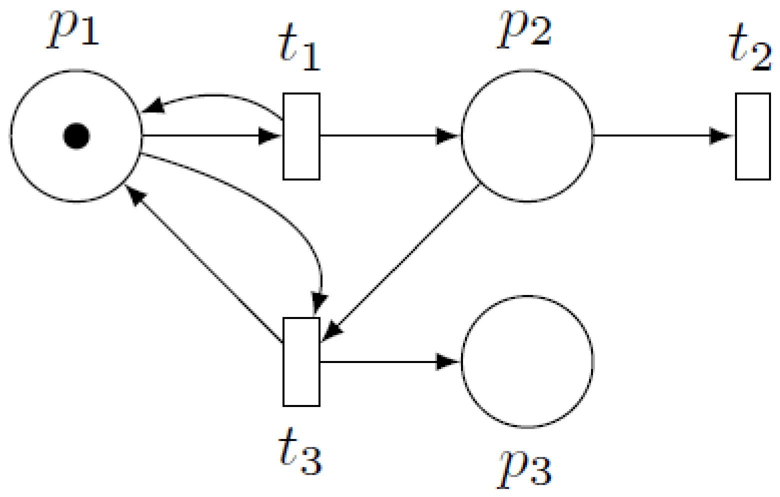
→ “If both processes want to access the resource, they get it in turns.”

d) Bonus Try to implement this specification in your Petri Net… (Is it possible?)

5 Coverability tree and graph

a) Coverability tree

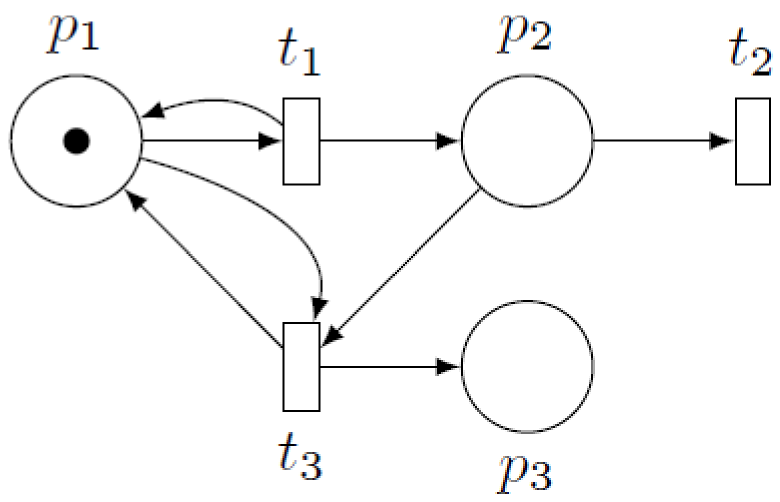
$M_0 = (1{,}0{,}0)$



5 Coverability tree and graph

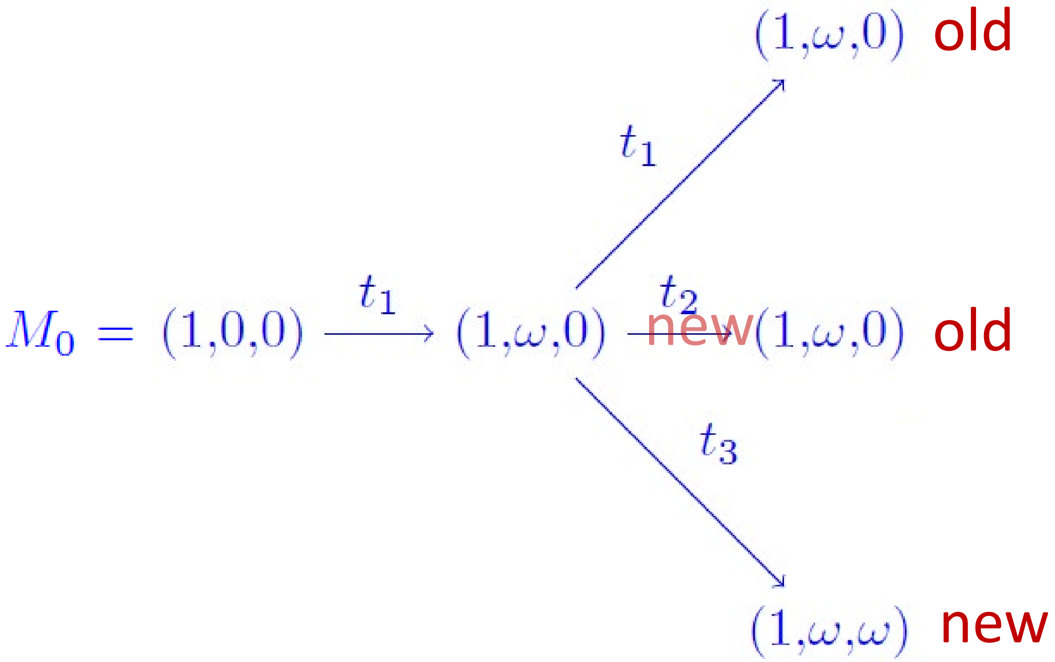
a) Coverability tree

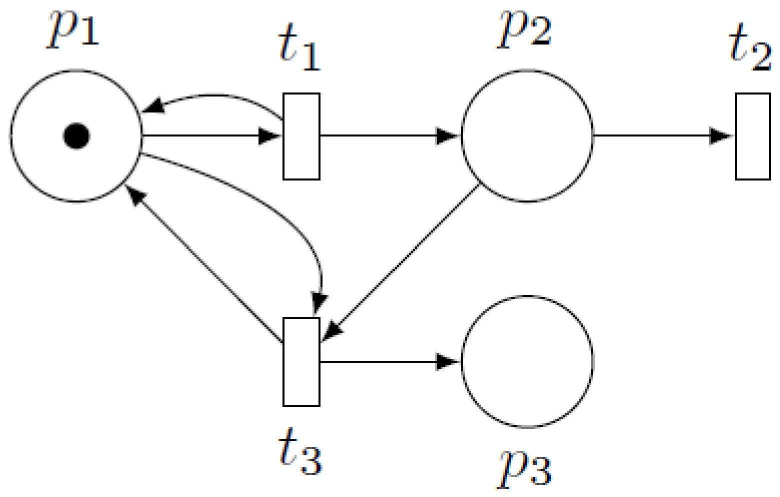
$M_0 = (1{,}0{,}0) \xrightarrow{t_1} (1{,}\omega{,}0)$ new



5 Coverability tree and graph

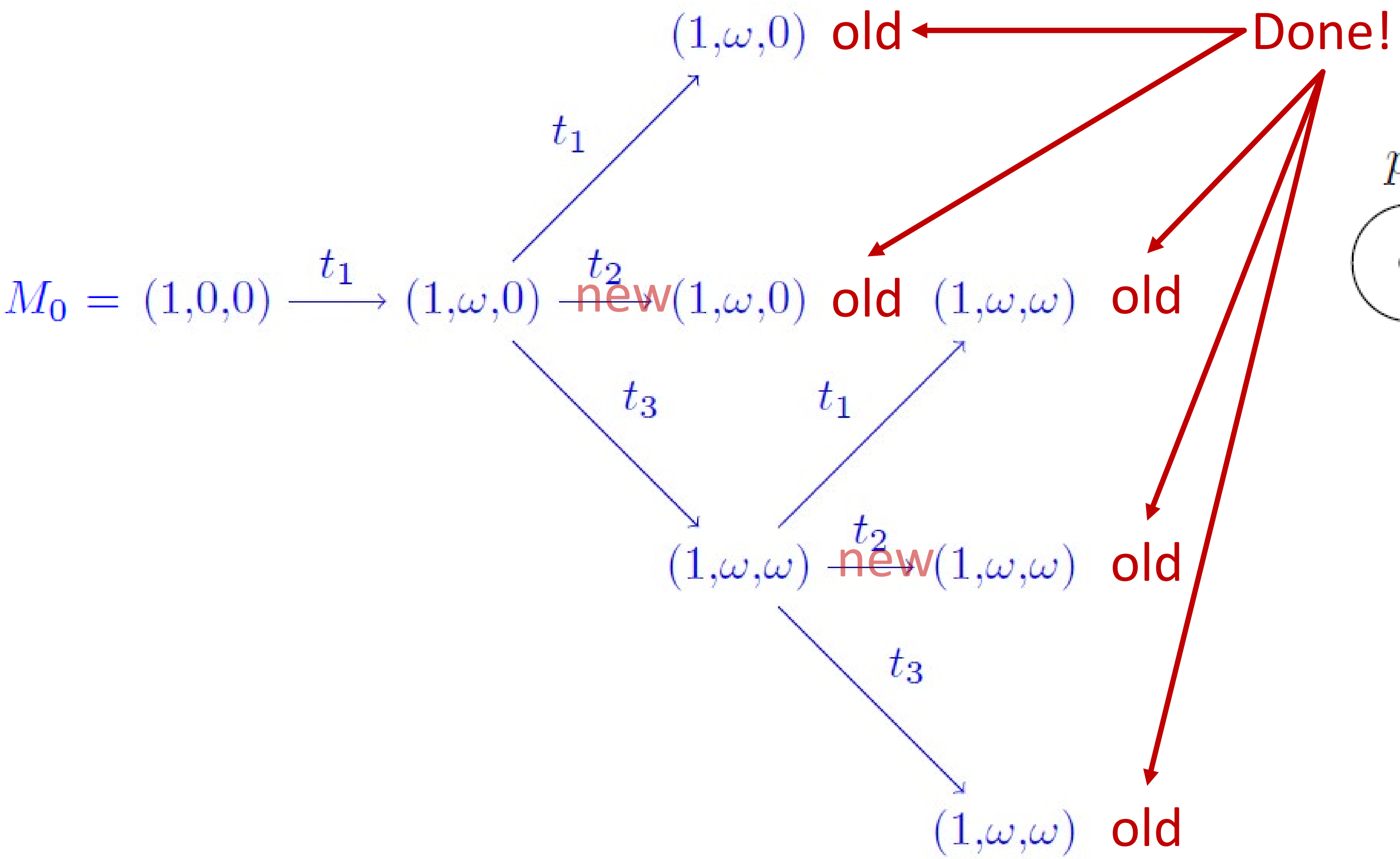
a) Coverability tree

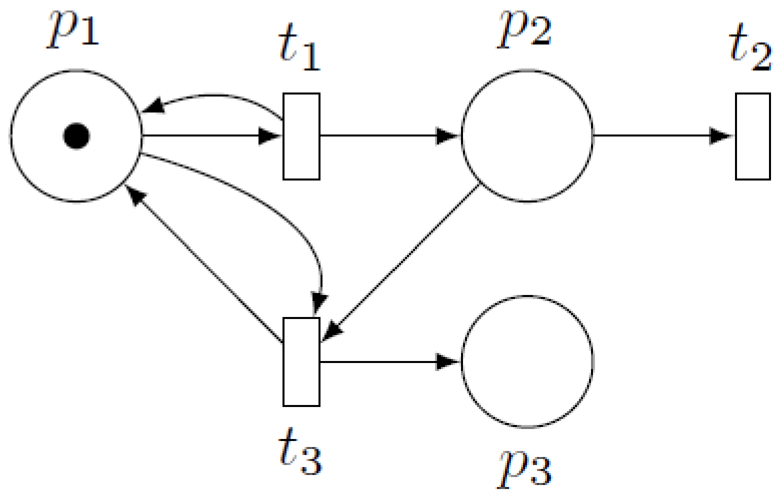




5 Coverability tree and graph

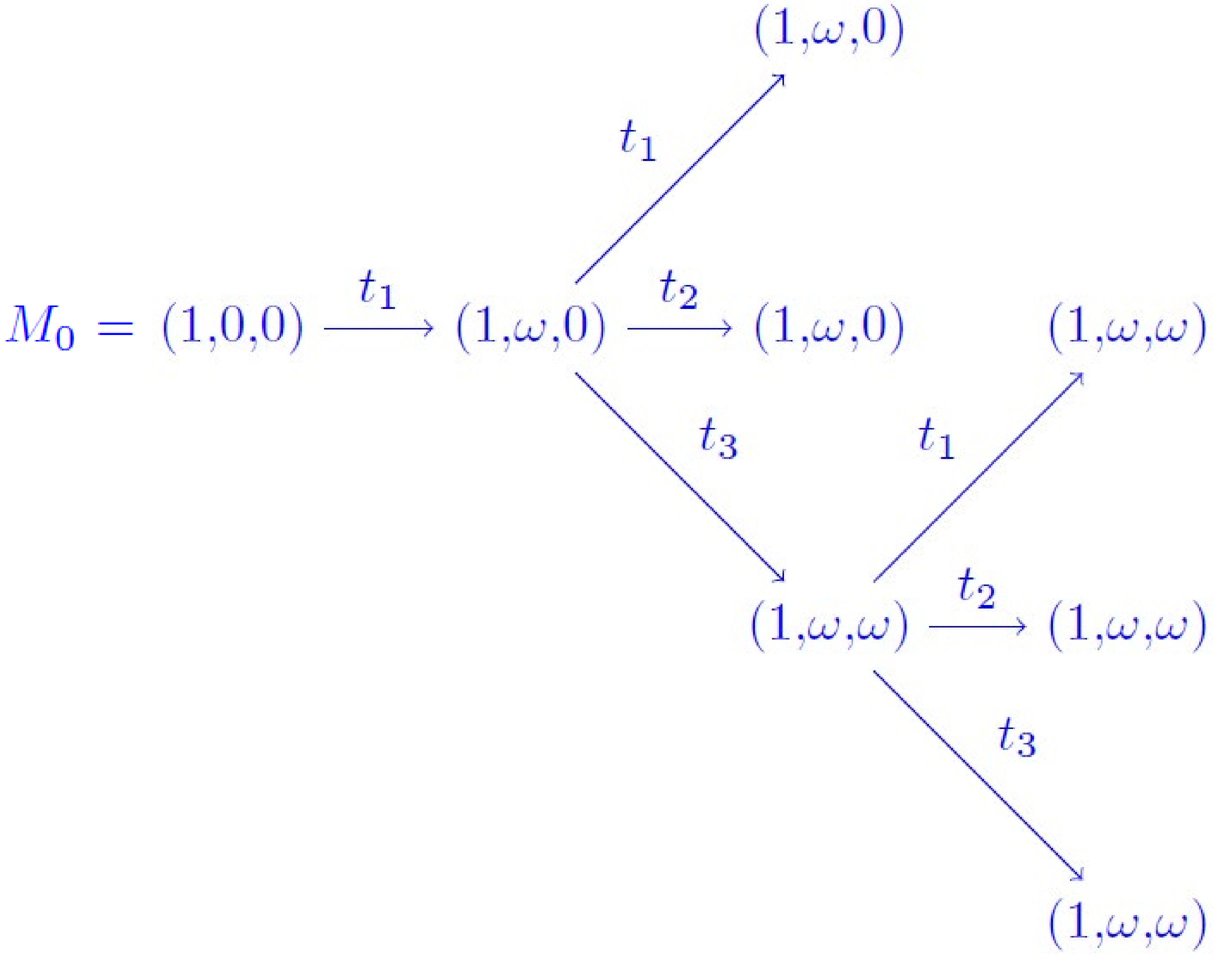
a) Coverability tree

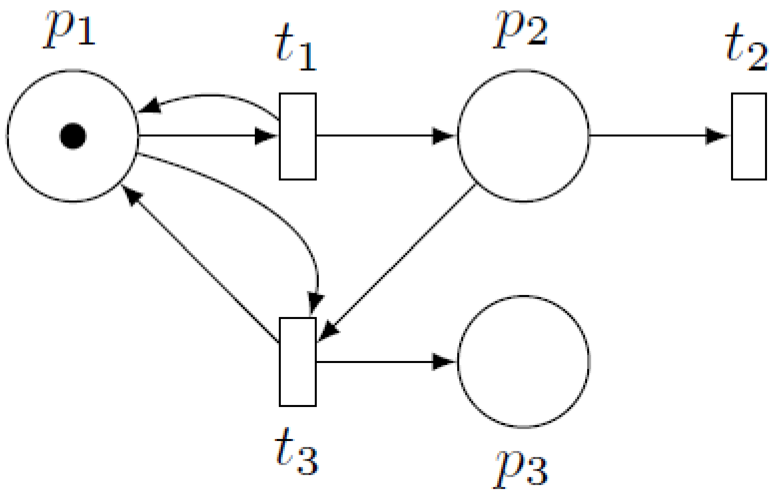




5 Coverability tree and graph

b) Coverability graph





$$M_0 = (1,0,0) \xrightarrow{t_1} \underbrace{(1,\omega,0)}_{t_1,t_2} \xrightarrow{t_3} \underbrace{(1,\omega,\omega)}_{t_1,t_2,t_3}$$

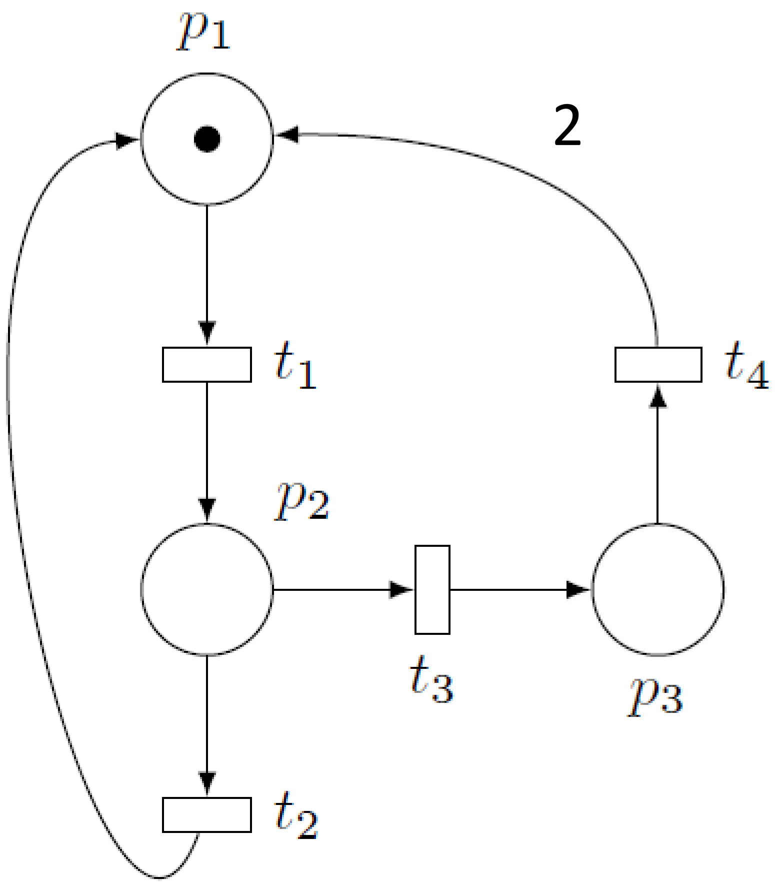
6 Reachability Analysis for Petri Nets

a) Not feasible in general because infinite number of states
→ When do we stop if looking for a non-reachable marking?
Coverability? Always finite!
→ Can only prove non-reachability in the general case.

b) Is $s = (101, 99, 4)$ reachable?
→ Start with necessary condition using
the incidence matrix: $\exists\, F, s = s_0 + A \cdot F$?

$$\mathbf{A} = \begin{pmatrix} -1 & 1 & 0 & 2 \\ 1 & -1 & -1 & 0 \\ 0 & 0 & 1 & -1 \end{pmatrix}$$

$$\begin{pmatrix} -1 & 1 & 0 & 2 \\ 1 & -1 & -1 & 0 \\ 0 & 0 & 1 & -1 \end{pmatrix} \cdot \begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{pmatrix} = \begin{pmatrix} 100 \\ 99 \\ 4 \end{pmatrix} = s - s_0$$

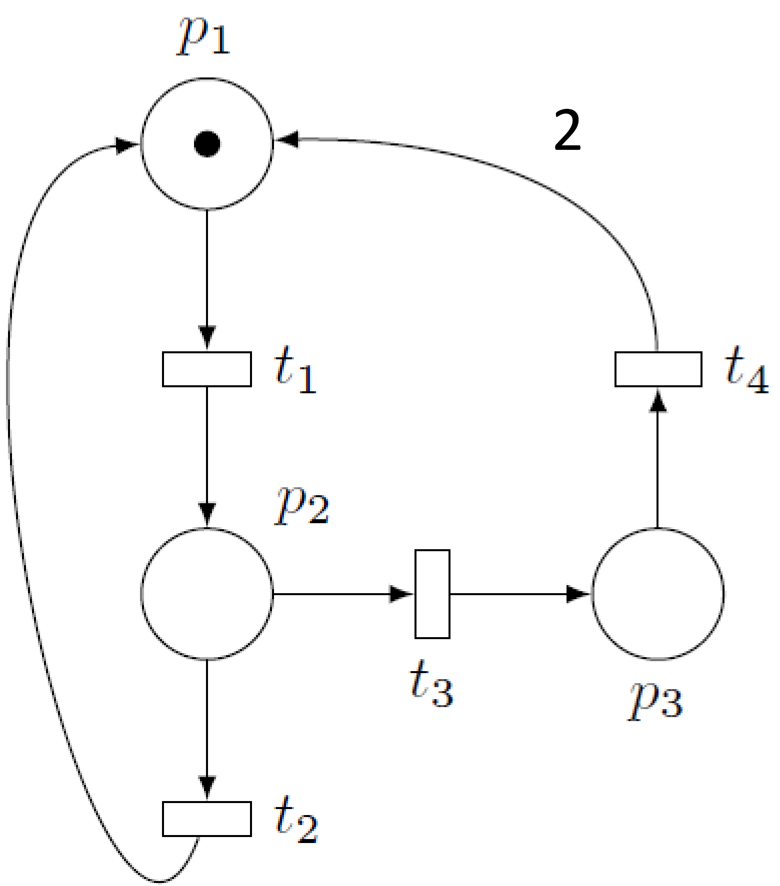


6 Reachability Analysis for Petri Nets

b) Is $s = (101, 99, 4)$ reachable?

$\rightarrow$ Start with necessary condition using the incidence matrix: $\exists\, F, s = s_0 + A \cdot F$?

$$\begin{pmatrix} -1 & 1 & 0 & 2 \\ 1 & -1 & -1 & 0 \\ 0 & 0 & 1 & -1 \end{pmatrix} \cdot \begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{pmatrix} = \begin{pmatrix} 100 \\ 99 \\ 4 \end{pmatrix} = s - s_0$$



No systematic approach… Look at the net and try it out.

$F_1 = (203, 0, 203, 203) \Rightarrow s_1 = (204, 0, 0)$

$F_2 = (103, 0, 0, 0) \Rightarrow s_2 = (101, 103, 0)$

$F_3 = (0, 0, 4, 0) \Rightarrow s_3 = (101, 99, 4) = s$

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Crash course – Petri nets
Introduction

See you next week!