Industrial Automation

(Automação de Processos Industriais)

Supervised Control of Discrete Event Systems

Supervision Controllers (Part 1/2)

http://www.isr.tecnico.ulisboa.pt/~jag/courses/api20b/api2021.html

Prof. Paulo Jorge Oliveira, original slides
Prof. José Gaspar, rev. 2020/2021

Syllabus:

…

Chap. 8 - DESs and Industrial Automation [2 weeks]

Chap. 9 – Supervised Control of DESs [1 week]

*** SCADA**
*** Methodologies for the Synthesis of Supervision Controllers**
*** Failure detection**

Some jokes available in http://members.iinet.net.au/~ianw/cartoon.html

The End.

Some pointers on Supervised Control of DES

History:

The SCADA Web, http://members.iinet.net.au/~ianw/
Monitoring and Control of Discrete Event Systems, Stéphane Lafortune,
http://www.ece.northwestern.edu/~ahaddad/ifac96/introductory_workshops.html

Tutorial:

http://vita.bu.edu/cgc/MIDEDS/
http://www.daimi.au.dk/PetriNets/

Analysers & Simulators:

http://www.nd.edu/~isis/techreports/isis-2002-003.pdf (Users Manual)
http://www.nd.edu/~isis/techreports/spnbox/ (Software)

Bibliography:

* SCADA books http://www.sss-mag.com/scada.html
* K. Stouffer, J. Falco, K. Kent, "**Guide to Supervisory Control and Data Acquisition (SCADA) and Industrial Control Systems Security**", NIST Special Publication 800-82, 2006
* Moody J. e Antsaklis P., **“Supervisory Control of Discrete Event Systems using Petri Nets,”** Kluwer Academic Publishers, 1998.
* Cassandras, Christos G., **"Discrete Event Systems - Modeling and Performance Analysis,"** Aksen Associates, 1993.
* Yamalidou K., Moody J., Lemmon M. and Antsaklis P. **Feedback Control of Petri Nets Based on Place Invariants** http://www.nd.edu/~lemmon/isis-94-002.pdf

Supervision of DES

And

Now

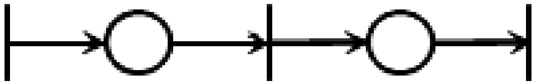
Something

Completely

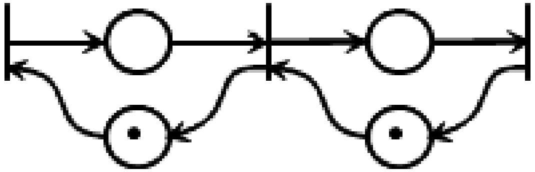
Different

Supervision of DES: *motivation*

Given one Unsafe Petri net can one obtain a Safe Petri net?



Unsafe Petri net



Safe Petri net

In many cases yes: **Supervision** of DES may be achieved using linear algebra based methodologies. *See the next slides!*

Other possible goals:

- Supervise and bound the work of the supervised DES
- Reinforce that some properties are verified
- Assure that some states are not reached
- Performance criteria are verified
- Prevent deadlocks in DES
- Constrain on the use of resources (e.g. mutual exclusion)

Supervision of DES

Some history on Supervised Control

- Methods for finite automata [Ramadge et *al.*], 1989
    - some are based on brute-force search (!)
    - or may require simulation (!)

- Formal verification of *software* in Computer Science (since the 60s) and on *hardware* (90, ...)

- Supervisory Control Method of Petri Nets, method based on *monitors* [Giua et *al.*], 1992.

- Supervisory Control of Petri Nets based on **Place Invariants** [Moody, Antsaklis et *al.*], 1994 (shares some similarities with the previous one, but deduced independently!...).

Supervision of DES

Advantages of the Supervisory Control of Petri Nets

- Mathematical representation is clear (and easy)
- Resorts only to linear algebra (matrices)
- More compact then automata
- Straightforward the representation of infinity state spaces
- Intuitive graphical representation available

The representation of the controller as a Petri Net leads to simplified Analysis and Synthesis tasks

Supervision of DES

Place Invariants

Place invariants are sets of places whose token count remains always constant. Place invariants can be computed from integer solutions of $w^T D = 0$. Non-zero entries of w correspond to the places that belong to the particular invariant.

Supervisor Synthesis using Place Invariants [ISIS docs]:

What type of relations can be represented in the method of Place Invariants?

- Sets of linear constraints in the state space
- Representation of convex regions (there are extensions for non-convex regions)
- Constraints to guarantee liveness and to avoid deadlocks *(that can be expressed, in general, as linear constraints)*
- Constraints on the events and timings *(that can be expressed, in general, as linear constraints)*

Methods of Analysis/Synthesis

Method of the Matrix Equations *(just to remind)*

The dynamics of the Petri net state can be written in compact form as:

$$\mu(k+1) = \mu(k) + Dq(k)$$

where:

$\mu(k+1)$ - marking to be reached
$\mu(k)$ - initial marking
$q(k)$ - firing vector (transitions)
D - incidence matrix. Accounts the balance of tokens, giving the transitions fired.

Methods of Analysis/Synthesis

How to build the Incidence Matrix? (*just to remind*)

For a Petri net with *n* places and *m* transitions

$$\mu \in N_0{}^n$$

$$q \in N_0{}^m$$

$$\boxed{D = D^+ - D^-}\ , \quad D \in Z^{n\times m}, \quad D^+ \in N_0^{n\times m}, \quad D^- \in N_0^{n\times m}$$

The enabling firing rule is $\boxed{\mu \geq D^- q}$

Can also be written in compact form as the inequality $\mu + Dq \geq 0$, interpreted element-by-element.

Note: in this course all vector and matrix inequalities are read element-by-element unless otherwise stated.

Supervision of DES

$$w^T \mu(k+1) = w^T \mu(k) + \underbrace{w^T D}_{=0} \underbrace{q(k)}_{\forall k}$$

eq,d

Place Invariants

Place invariants are sets of places whose token count remains always constant. Place invariants can be computed from integer solutions of $w^T D = 0$. Non-zero entries of w correspond to the places that belong to the particular invariant.

Supervisor Synthesis using Place Invariants [ISIS docs]:

What type of relations can be represented in the method of Place Invariants?

- Sets of linear constraints in the state space
- Representation of convex regions (there are extensions for non-convex regions)
- Constraints to guarantee liveness and to avoid deadlocks *(that can be expressed, in general, as linear constraints)*
- Constraints on the events and timings *(that can be expressed, in general, as linear constraints)*

Methods of Synthesis

Some notation for the method

- The supervised system is modelled as a Petri net with n places and m transitions, and incidence matrix

$$\boxed{D_P \in Z^{n\times m}.}$$

$$\begin{bmatrix} \mu_p(k+1) \\ \mu_c(k+1) \end{bmatrix} = \begin{bmatrix} \mu_p(k) \\ \mu_c(k) \end{bmatrix} + \begin{bmatrix} D_p \\ D_c \end{bmatrix} q(k)$$

- The supervisor is modelled as a Petri net with n_C places and m transitions, and incidence matrix

$$\boxed{D_C \in Z^{n_C\times m}.}$$

- The resulting total system has an incidence matrix

$$\boxed{D \in Z^{(n+n_C)\times m}.}$$

Methods of Synthesis

Theorem: Synthesis of Controllers based on Place Invariants (T1)

Given the set of linear state constraints that the supervised system must follow, written as

$$\boxed{L\mu_P \leq b, \quad \mu_P \in N_0^{\,n}, \quad L \in Z^{n_C \times n} \quad and \quad b \in Z^{n_C}.}$$

If $\boxed{b - L\mu_{P_0} \geq 0,}$

then the controller with incidence matrix and the initial marking, respectively

$$\boxed{D_C = -LD_P,} \quad \text{and} \quad \boxed{\mu_{C_0} = b - L\mu_{P_0},}$$

enforce the constraints to be verified for all markings obtained from the initial marking.

Methods of Synthesis

Theorem - proof outline :

The constraint $L\mu_P \leq b$ can be written as $L\mu_P + \mu_C = b$, using the slack variables μ_C. They represent the marking of the n_C places of the controller.

To have a place invariant, the relation $w^T D = 0$ must be verified and in particular, given the previous constraint:

$$w^T D = \begin{bmatrix} L & I \end{bmatrix} \begin{bmatrix} D_P \\ D_C \end{bmatrix} = 0, \text{ resulting } \boxed{D_C = -LD_P.}$$

From $L\mu_{P_0} + \mu_{C_0} = b$, follows that $\boxed{\mu_{C_0} = b - L\mu_{P_0}.}$

∎

Some more details :

$$L\mu_p \le b \longrightarrow D_c = -LD_p, \quad \mu_{c0} = b - L\mu_{p0}$$

Using slack variables μ_c

allows $L\mu_p \le b \longrightarrow L\mu_p + \mu_c = b$

which can be written in matrix form as $[L \ \ I]\begin{bmatrix}\mu_p\\ \mu_c\end{bmatrix} = b$

Given the dynamics

$$\begin{bmatrix}\mu_p(k+1)\\ \mu_c(k+1)\end{bmatrix} = \begin{bmatrix}\mu_p(k)\\ \mu_c(k)\end{bmatrix} + \begin{bmatrix}D_p\\ D_c\end{bmatrix} q(k)$$

one has

$$[L \ \ I]\left(\begin{bmatrix}\mu_p\\ \mu_c\end{bmatrix} + \begin{bmatrix}D_p\\ P_c\end{bmatrix} q(k)\right) = b$$

$$\underbrace{\qquad\qquad\qquad}_{= b \longleftarrow} \qquad \cdots \underbrace{\qquad\qquad}_{=0}$$

$$[L \ \ I]\begin{bmatrix}D_p\\ D_c\end{bmatrix} q(k) = 0$$

desired $\forall q(k)$

so $[L \ \ I]\begin{bmatrix}D_p\\ D_c\end{bmatrix} = 0$

$$LD_p + D_c = 0 \longrightarrow D_c = -LD_p \;/\!/$$

The initial state is direct:

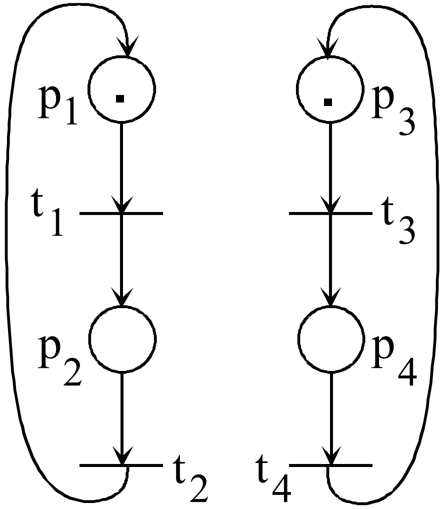
$$L\mu_p + \mu_c = b \longleftarrow \forall \begin{bmatrix}\mu_p\\ \mu_c\end{bmatrix}$$

$$L\mu_{p0} + \mu_{c0} = b$$

$$\mu_{c0} = b - L\mu_{p0} \;/\!/$$

Methods of Synthesis

Example of controller synthesis: Mutual Exclusion



Linear constraint: $\mu_2 + \mu_4 \leq 1$

that can be written as:

$$L\mu_P \leq b \qquad \begin{bmatrix} 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \\ \mu_4 \end{bmatrix} \leq 1.$$

Incidence Matrix $D_P = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix}$ and initial marking $\mu_{P_0} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}.$

Methods of Synthesis

Example of controller synthesis: Mutual Exclusion

1) Test

$$b - L\mu_{P_0} = 1 - \begin{bmatrix} 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} = 1 \geq 0.$$

OK.

2) Compute

$$D_C = -LD_P = -\begin{bmatrix} 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} -1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix} = \begin{bmatrix} -1 & 1 & -1 & 1 \end{bmatrix},$$

and

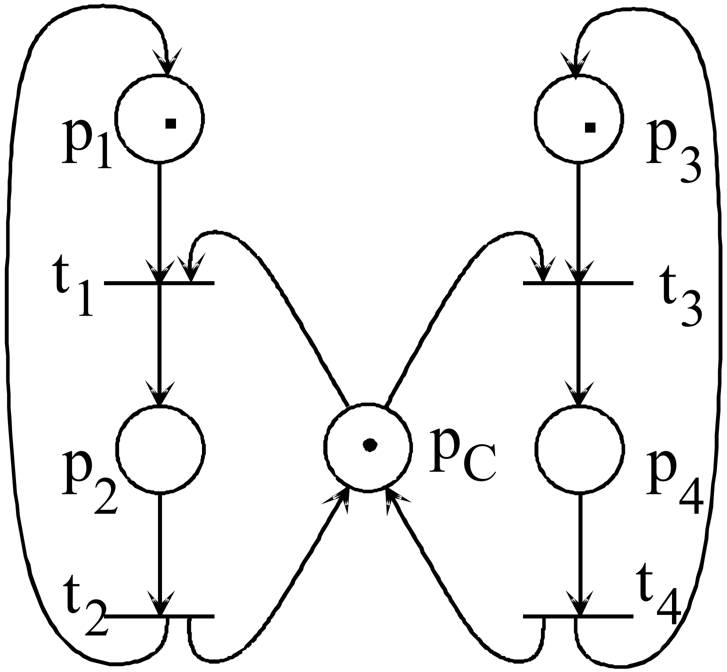
$$\mu_{C_0} = b - L\mu_{P_0} = 1.$$

OK.

Methods of Synthesis

Example of controller synthesis: Mutual Exclusion

3) Resulting in



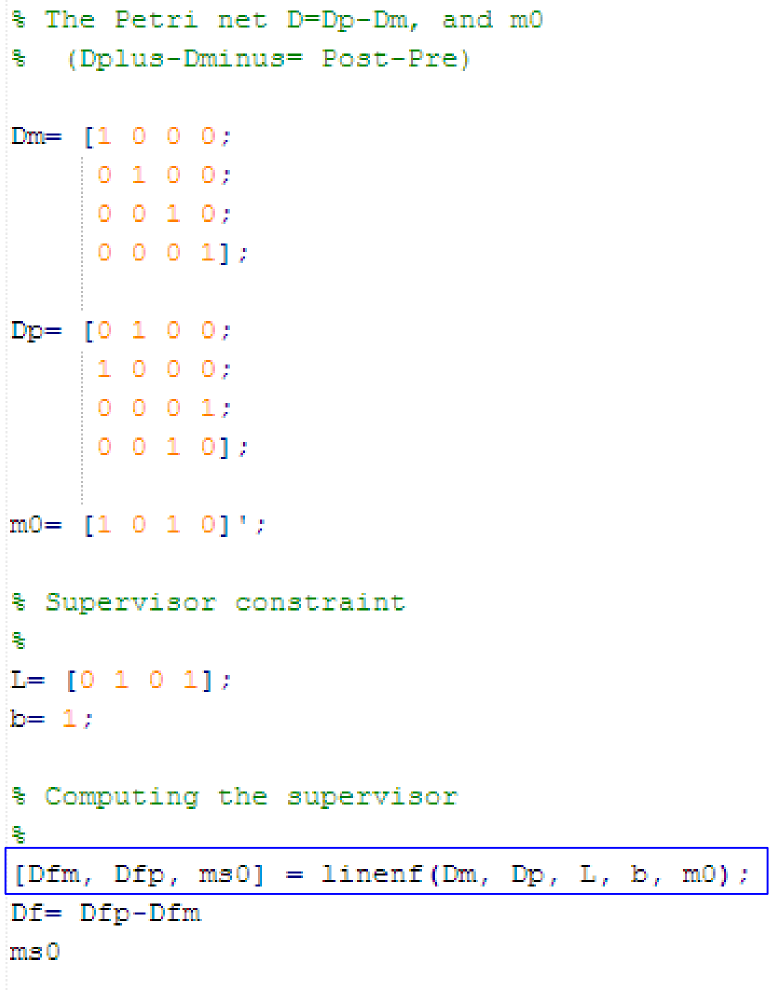
$$D = \left[\begin{array}{cccc} -1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 1 & -1 \\ \multicolumn{4}{c}{\boxed{-1 \quad 1 \quad -1 \quad 1}} \end{array}\right]$$

$$\mu_0 = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ \boxed{1} \end{bmatrix}$$

OK.
UAU!!!.

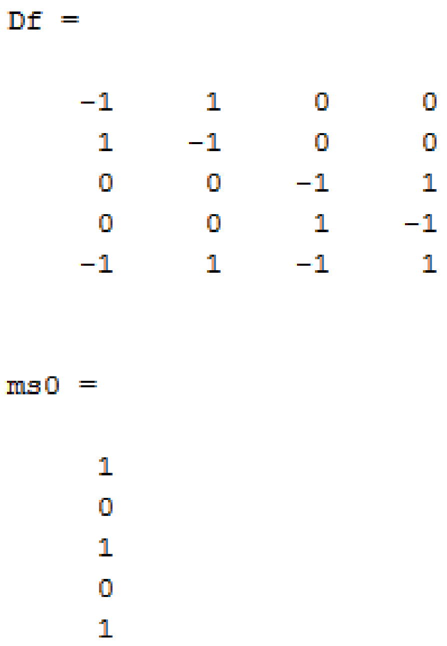
Methods of Synthesis

Example of controller synthesis: Mutual Exclusion



```
% The Petri net D=Dp-Dm, and m0
%  (Dplus-Dminus= Post-Pre)

Dm= [1 0 0 0;
     0 1 0 0;
     0 0 1 0;
     0 0 0 1];

Dp= [0 1 0 0;
     1 0 0 0;
     0 0 0 1;
     0 0 1 0];

m0= [1 0 1 0]';

% Supervisor constraint
%
L= [0 1 0 1];
b= 1;

% Computing the supervisor
%
[Dfm, Dfp, ms0] = linenf(Dm, Dp, L, b, m0);
Df= Dfp-Dfm
ms0
```

Result using the function **linenf.m** of the toolbox SPNBOX:



```
Df =

    -1     1     0     0
     1    -1     0     0
     0     0    -1     1
     0     0     1    -1
    -1     1    -1     1


ms0 =

     1
     0
     1
     0
     1
```

Methods of Synthesis

Definition:

Maximal permissivity occurs when (i) all the linear constraints are verified and (ii) all legal markings can be reached.

Lemmas:

L1) **The controllers** obtained with T1 **have maximal permissivity.**

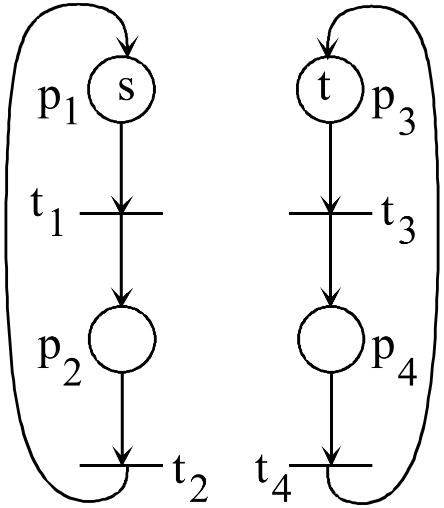
L2) Given the linear constraints used, **the place invariants** obtained with the controller synthesized with T1 **are the same** as the invariants associated with the initial system.

Methods of Synthesis

Example of controller synthesis

$\forall s \in N_0, \forall t \in N_0, \forall n \in N_0$

n Readers / 1 Writer



Linear constraint $\mu_2 + n\mu_4 \leq n$

(max *n* readers or 1 writer)

That can be written as:

$$L\mu_P \leq b \qquad \begin{bmatrix} 0 & 1 & 0 & n \end{bmatrix} \begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \\ \mu_4 \end{bmatrix} \leq n.$$

Incidence Matrix

$$D_P = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix}$$

and initial marking

$$\mu_{P_0} = \begin{bmatrix} s \\ 0 \\ t \\ 0 \end{bmatrix}.$$

Methods of Synthesis

Example of controller synthesis

n Readers / 1 Writer

1) Test

$$b - L\mu_{P_0} = n - \begin{bmatrix} 0 & 1 & 0 & n \end{bmatrix} \begin{bmatrix} s \\ 0 \\ t \\ 0 \end{bmatrix} = n \geq 0.$$

OK.

2) Compute

$$D_C = -LD_P = -\begin{bmatrix} 0 & 1 & 0 & n \end{bmatrix} \begin{bmatrix} -1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix} = \begin{bmatrix} -1 & 1 & -n & n \end{bmatrix},$$

and

$$\mu_{C_0} = b - L\mu_{P_0} = n.$$

OK.

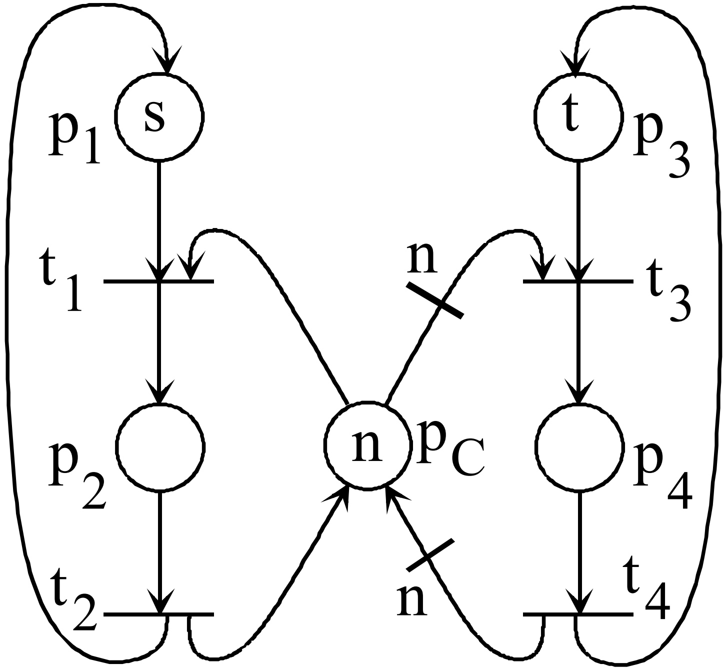
Methods of Synthesis

Example of controller synthesis

n Readers / 1 Writer

3) Resulting in

$$D = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 1 & -1 \\ \boxed{-1 \quad 1 \quad -n \quad n} \end{bmatrix}$$



$$\mu_0 = \begin{bmatrix} s \\ 0 \\ t \\ 0 \\ \boxed{n} \end{bmatrix}$$

OK.
UAU!!!.

Supervision of DES

Advantages of the Method of the Place Invariants [ISIS docs]:

Other characteristics that can impact on the solutions?

- Existence and uniqueness

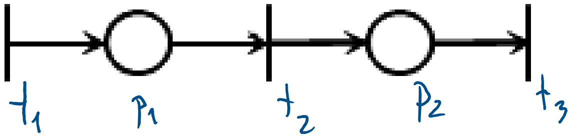
- Optimality of the solutions (e.g. maximal permissivity)

- Existence of transition non-controllable and/or not observable (remind definitions for time-driven systems)

In general the solutions can be found solving:

Linear Programming Problems, with Linear Constraints

Given one Unsafe Petri net can one obtain a Safe Petri net?



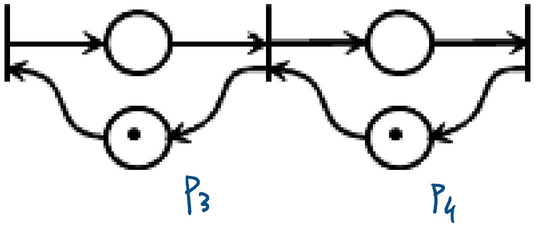
Unsafe Petri net

$$\underbrace{\begin{array}{c|c|c|c|}
 & t_1 & t_2 & t_3 \\ \hline
\mu_1 & +1 & -1 & \\ \hline
\mu_2 & & +1 & -1 \\ \hline
\end{array}}_{D_p}$$

$$\begin{cases} \mu_1 \leq 1 \\ \mu_2 \leq 1 \end{cases} \longleftrightarrow L\mu \leq b \quad \Rightarrow \quad L = \begin{bmatrix} 1 & \\ & 1 \end{bmatrix}, \; b = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$D_c = -L D_p = -I D_p = -D_p \;, \quad \mu_{c_0} = b - L\mu_{p_0} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} //$$

$$D_c = -D_p = \left[\begin{array}{c|c|c} t_1 & t_2 & t_3 \\ -1 & +1 & \\ \hline & -1 & +1 \end{array}\right] \begin{array}{c} \\ p_3 \\ p_4 \end{array} \quad \Rightarrow$$



Safe Petri net