Petri Net Supervisors for Disjunctive Constraints

Marian V. Iordache
School of Engineering and Eng. Tech.
LeTourneau University
Longview, TX 75607-7001

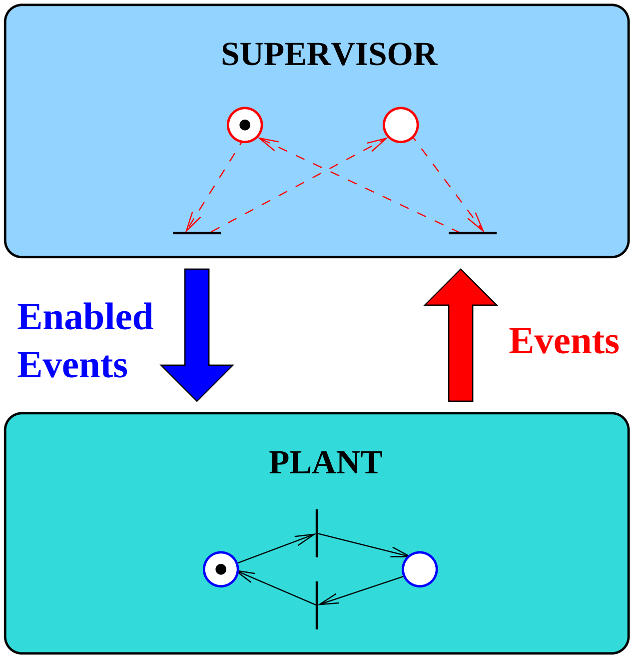
Panos J. Antsaklis
Department of Electrical Engineering
University of Notre Dame
Notre Dame, IN 46556

July 13, 2007

Supervisory Control

SC input describes:

- System to be controlled (PN, automata).
- Properties to be satisfied during operation.
- Implementation constraints
  - controllability
  - observability
  - decentralization



SC result: ***supervisor***

- Represented by a DES (PN, automata).
- Implemented in computer code.

Supervisory Control

SC relevant for

- Coordination problems (e.g. in flexible manufacturing)
- High level decision making in controllers
- Design automation

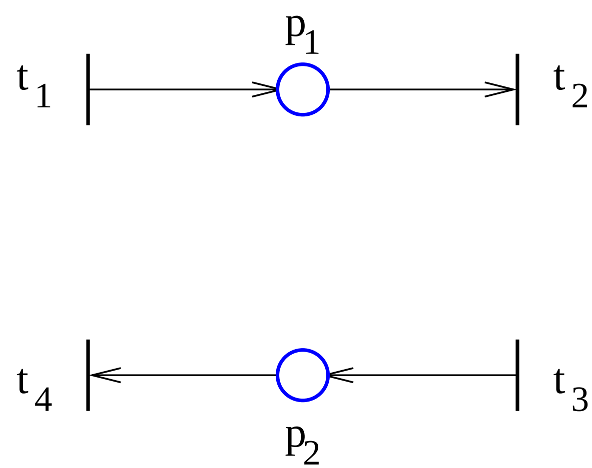
Why Petri Nets?

- Often considerably more compact than automata.
- PN methods include also automata methods (by reach. graph).
- Naturally model concurrency.

PN Specifications

- Mutual exclusion. *Common in Comp. Sci. In [Krogh 91] for AGV coordination. In [Tittus 99] for batch chemical processes.*

  Example: $\mu_1 + \mu_2 \leq 1$



- Fairness constraints. *In [Li 93] for manufacturing application. In [Genrich 80] for a communication protocol.*

  Example: $v_1 - v_3 \leq 0$

- Enabling constraints. *In [Yamalidou 91], for chem. process control. In [Giua and Seatzu 01], for railway networks.*

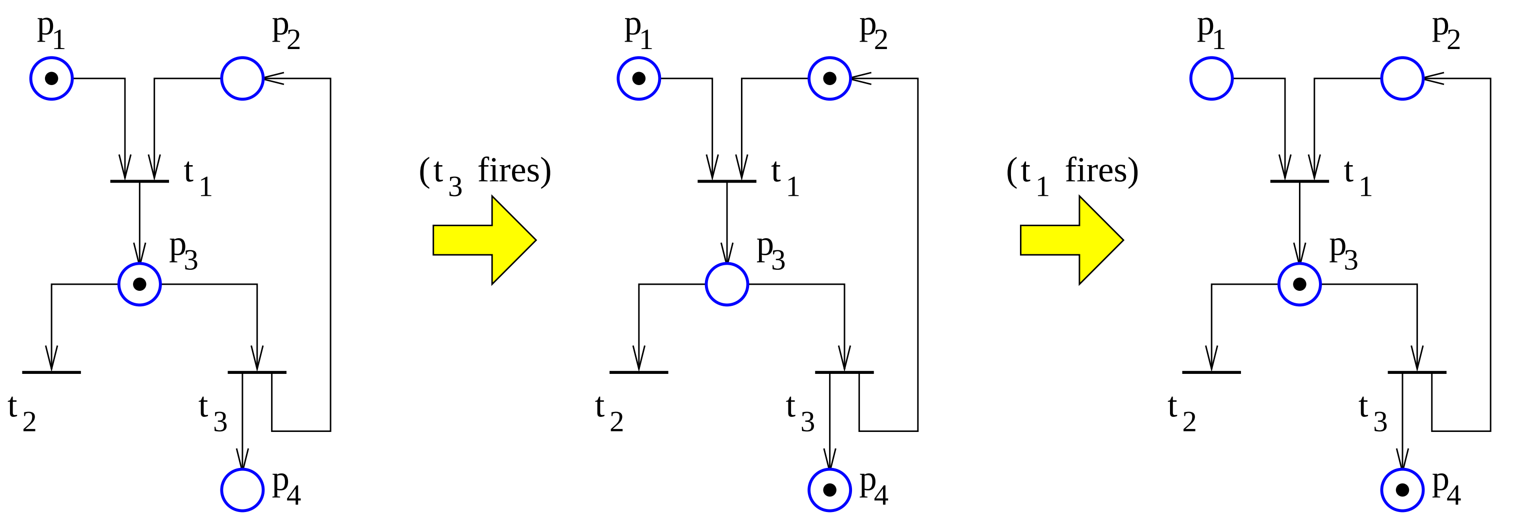
  Example: $q_1 \leq \mu_2$

Background

Notation: μ – the marking, μ_0 – the initial marking, q – the firing vector, and v – the Parikh vector. Let $\mu_i = \mu(p_i)$, $q_j = q(t_j)$ and $v_j = v(t_j)$.

v_j: how many times t_j has been fired.

q_j: how many times t_j is fired at a firing instance.



$$\mu_0 = [\ 1\quad 0\quad 1\quad 0\ ]^T \qquad \mu' = [\ 1\quad 1\quad 0\quad 1\ ]^T \qquad \mu'' = [\ 0\quad 0\quad 1\quad 1\ ]^T$$

$$v = [\ 0\quad 0\quad 0\ ]^T \qquad v = [\ 0\quad 0\quad 1\ ]^T \qquad v = [\ 1\quad 0\quad 1\ ]^T$$

$$q = [\ 0\quad 0\quad 1\ ]^T \qquad q = [\ 1\quad 0\quad 0\ ]^T$$

The general form of the specification discussed so far.

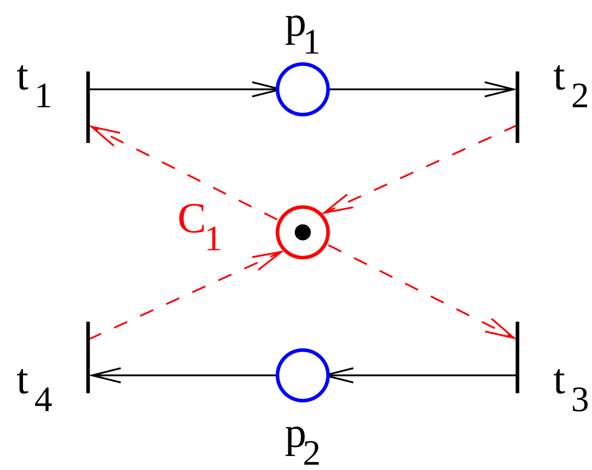
$$L\mu + Hq + Cv \leq b$$

L, H, and Q: integer matrices; b: integer vector.

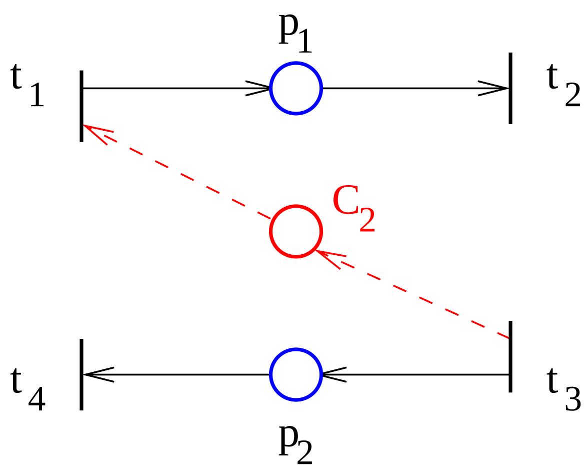
$$\mu = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_m \end{bmatrix} \quad q = \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_n \end{bmatrix} \quad v = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$$

This is the class of constraints that can be enforced by *monitors*.

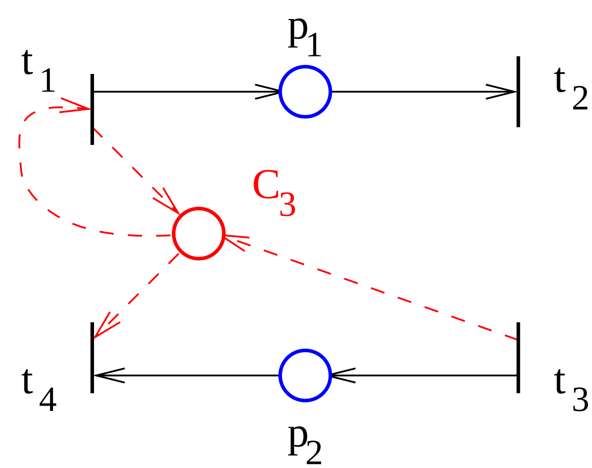
$$\mu_1 + \mu_2 \leq 1$$



$$v_1 - v_3 \leq 0$$



$$q_1 \leq \mu_2$$

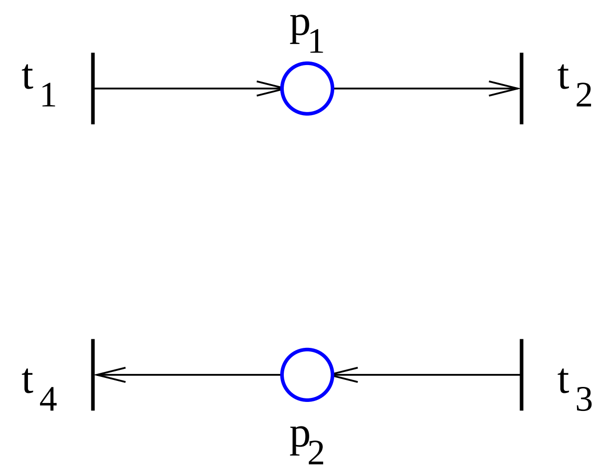


PN Specifications

Monitors implement specs $L\mu + Hq + Cv \leq b$ [Iordache and Antsaklis 2003].

Not all interesting specs have this form.

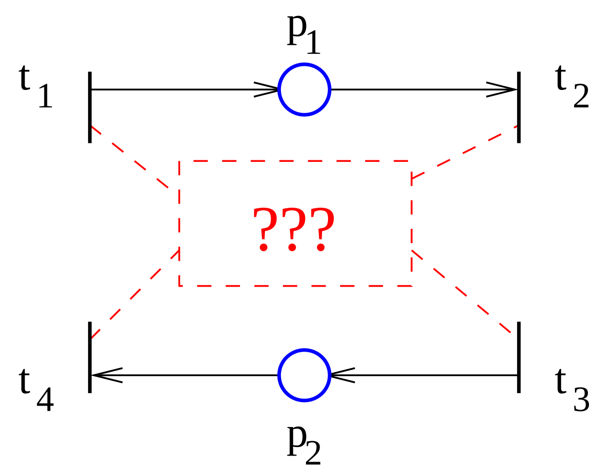
Example: In a single track segment all trains must go in the same direction.



Thus, $[\mu_1 \leq 0] \vee [\mu_2 \leq 0]$.

Problem Statement

$[\mu_1 \leq 0] \vee [\mu_2 \leq 0]$



PN supervisor for $\bigvee_i L_i\mu \leq b_i$ designed in [Iordache and Antsaklis 2005].

However, the solution was suboptimal (in general not least restrictive).

In this paper:

Design optimal PN supervisor for $\bigvee_i L_i\mu + H_i q + C_i v \leq b_i$.

Note: The Cv term can be incorporated in the $L\mu$ term.

Assumptions

- The goal is to design *optimal* PN supervisor *under full controllability and observability.* Refer to the monograph [Iordache and Antsaklis 2006] for work on the general case.

- q and the terms $L_i\mu + C_i v$ are bounded.

- Finite upper and lower bounds are known.

- Two disjunction interpretations proposed.
  - *State based:* state only is constrained.
  - *Dynamic:* state and transients are constrained.

Optimality obtained under no concurrency (i.e. when two or more transitions are not fired at the same time.)

For simplicity, consider a simpler specification

$$\bigvee_i L_i \mu \leq b_i$$

where $L_i \in \mathbb{Z}^{m_i \times n}$ and $b_i \in \mathbb{Z}_i^m$.

Bring the specification to the equivalent form

$$\bigwedge_j \bigvee_{i \in A_j} l_i \mu \leq \beta_i$$

where $l_i \in \mathbb{Z}^{1 \times n}$, $\beta_i \in \mathbb{Z}$ and A_j is a set of integers.

Let δ_i be Boolean variables s.t.

$$\delta_i = 1 \text{ iff } l_i\mu \leq \beta_i \tag{1}$$

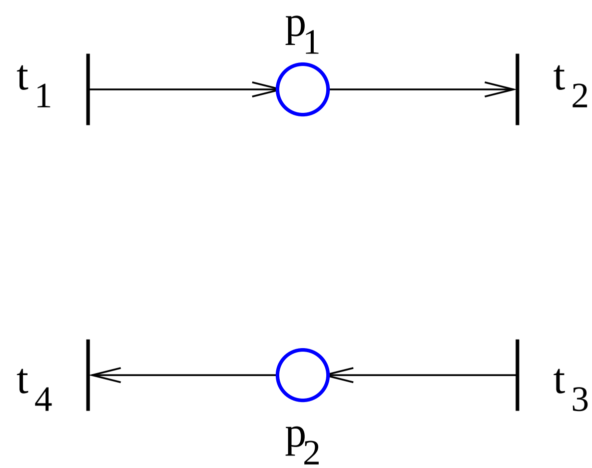
Then $\bigvee_{i \in A_j} l_i\mu \leq \beta_i$ is replaced by

$$\sum_{i \in A_j} \delta_i \geq 1 \tag{2}$$

If $m_i \leq l_i\mu \leq M_i$, then $[\delta_i = 1] \leftrightarrow [l_i\mu \leq \beta_i]$ becomes

$$l_i\mu + (M_i - \beta_i)\delta_i \leq M_i \tag{3}$$

$$l_i\mu + (\beta_i + 1 - m_i)\delta_i \geq \beta_i + 1 \tag{4}$$



In our example let δ_i be Boolean variables s.t.

$$\delta_i = 1 \text{ iff } \mu_i \leq 0 \qquad (5)$$

$[\mu_1 \leq 0] \vee [\mu_2 \leq 0]$ is replaced by $\delta_1 + \delta_2 \geq 1$.

If only two trains are allowed, $0 \leq \mu_i \leq 2$.

Then $[\delta_i = 1] \leftrightarrow [\mu_i \leq 0]$ is equivalent to

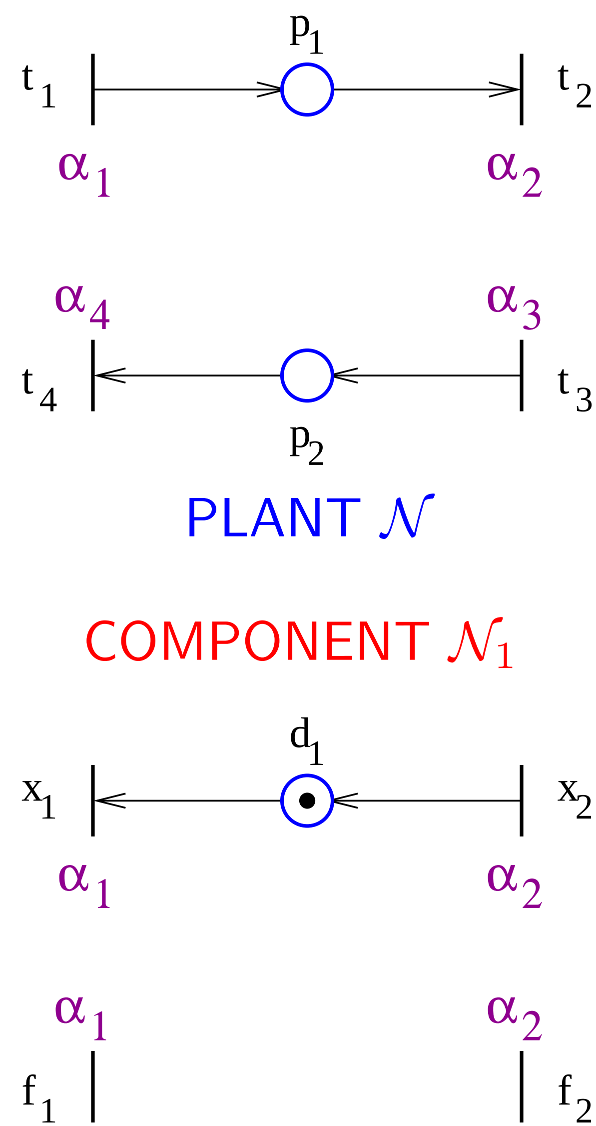
$$\mu_i + 2\delta_i \leq 2 \qquad (6)$$

$$\mu_i + \delta_i \geq 1 \qquad (7)$$

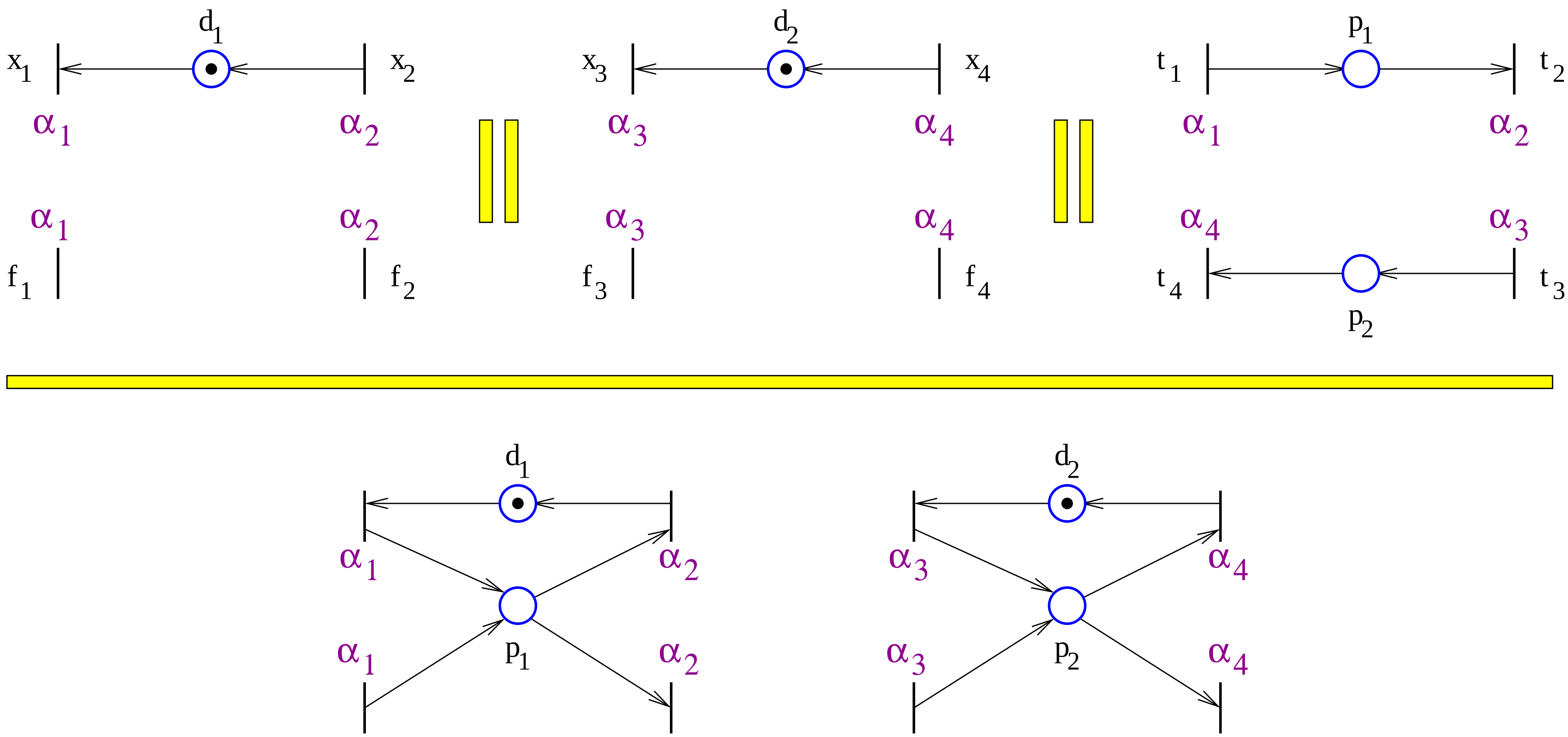
Assign distinct labels to each transition.

For each δ_i create a labeled PN $\mathcal{N}_i$.

1. Create place d_i; we intend $\mu(d_i) = \delta_i$.
2. Create transitions f_j and x_j for each t_j that can affect δ_i.
3. f_j and x_j have the same label as t_j.
4. The intent is that
   (a) f_j should model firings of t_j that do not affect δ_i.
   (b) x_j should model firings of t_j that affect δ_i.

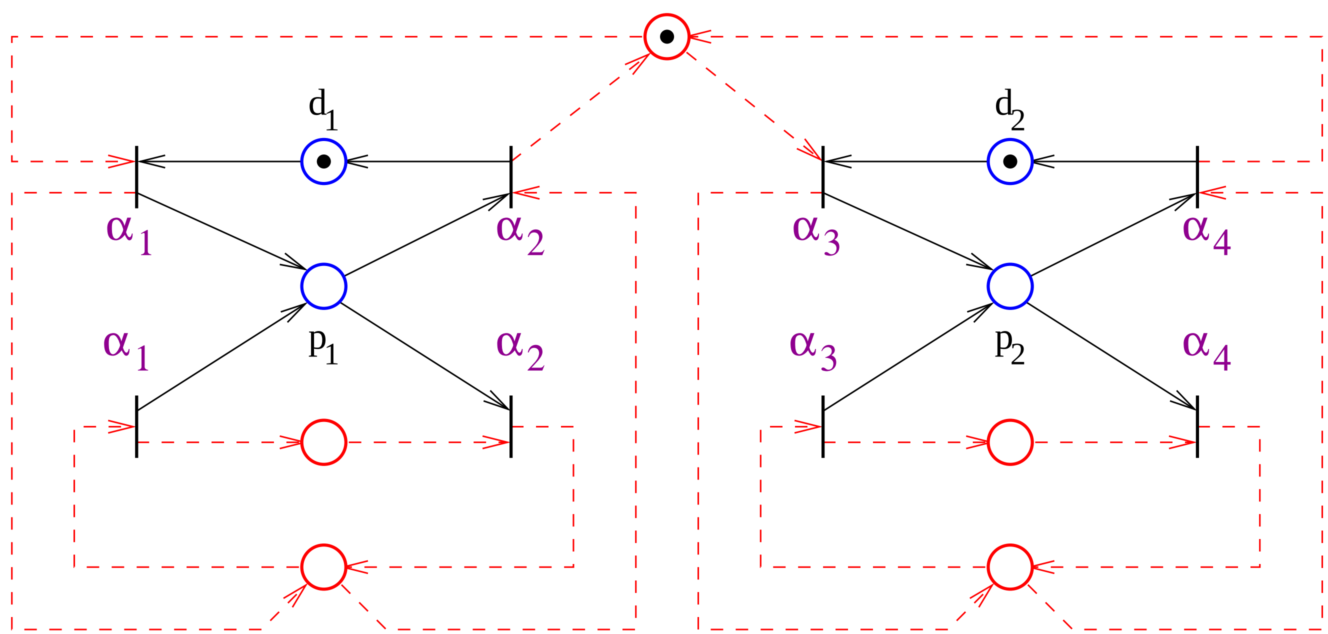


In this step the plant and the components $\mathcal{N}_i$ are composed.



The constraints are enforced on the composed net.

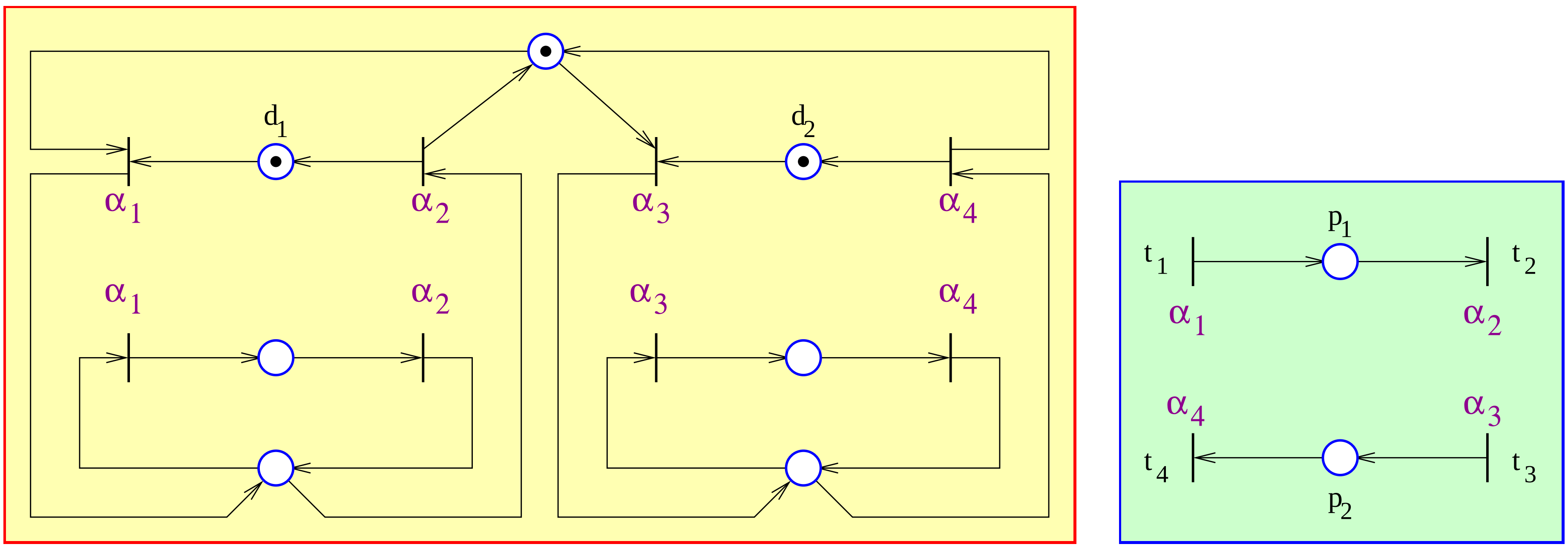
$$\begin{aligned}\mu_1 + 2\delta_1 &\leq 2 \\ \mu_1 + \delta_1 &\geq 1\end{aligned} \qquad \begin{aligned}\mu_2 + 2\delta_2 &\leq 2 \\ \mu_2 + \delta_2 &\geq 1\end{aligned} \qquad \delta_1 + \delta_2 \geq 1$$



Note: Labels are irrelevant at this step. Their role is to describe how to compose the plant to the constructed components.

The supervisor is obtained by deleting the places of the plant and then the transitions left completely disconnected ($\bullet t = t\bullet = \emptyset$).

Supervisor (left) and plant (right).



Performance

+ Least restrictive.

+ The supervisor is a PN.

− Complexity of supervisor.

  − Parallel composition → exp. number of transitions in the worst case.
  + The number of places depends linearly on the number of constraints $l_i\mu + h_i q + c_i v \leq \beta_i$.

Remark: In the general case, not all supervisor transitions are synchronized with plant transitions. (They are fired as soon as enabled.)