



Sistemi ad Eventi Discreti

a.a. 2018 - 2019

Prof. Luca Ferrarini

Petri, SFC & LAD

Trasmettitore e ricevitore (1)



- Il sistema è formato da un trasmettitore ed un ricevitore;
- Su necessità, il trasmettitore inizia a processare il messaggio da trasmettere (evento «PROC»);
- Una volta processato il messaggio, il trasmettitore stabilisce una connessione con il ricevitore – il quale deve essere disponibile per la ricezione – ed invia il messaggio (evento «SEND»);
- Il ricevitore è dotato di un buffer per i messaggi in ingresso assumibile come illimitato;
- Il ricevitore, se ha messaggi a disposizione nel proprio buffer, può iniziare a processarli («OPEN»);
- Il ricevitore finisce di processare il messaggio (evento «OK») e si riporta in stato disponibile.

Trasmettitore e ricevitore (2)

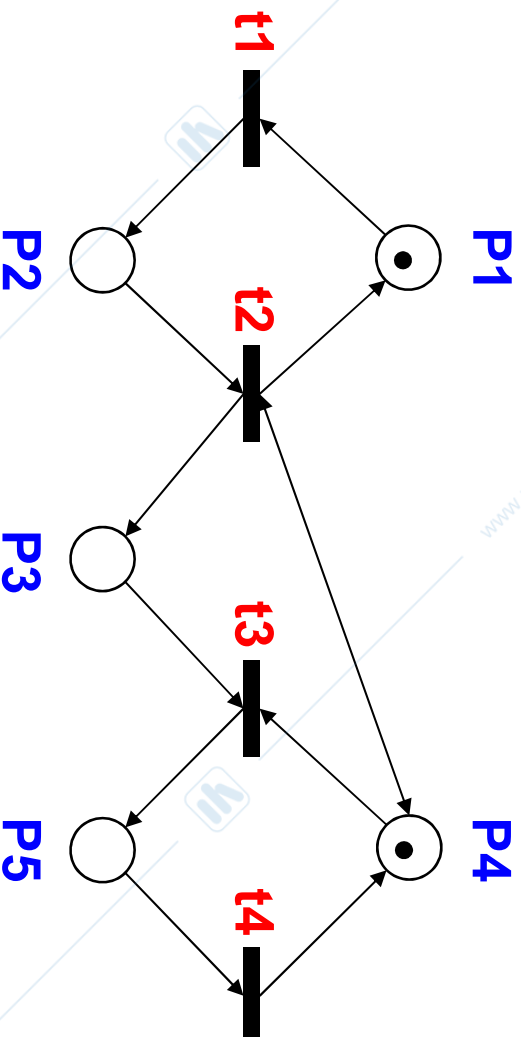


TRANSIZIONI (eventi):

- t1 (PROC)
- t2 (SEND)
- t3 (OPEN)
- t4 (OK)

POSTI:

- P1: trasmettitore disponibile
- P2: trasmettitore prepara messaggio
- P3: buffer ricevitore
- P4: ricevitore disponibile
- P5: ricevitore elabora messaggio



Trasmettitore e ricevitore (3)

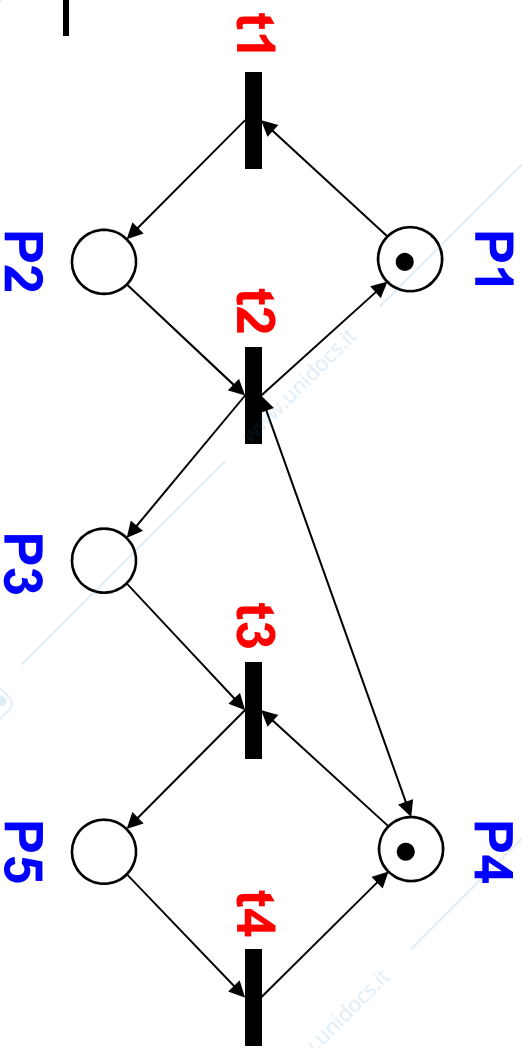


1. Matrice di incidenza
2. Classe della Rdp
3. Calcolo P.I. e conservatività
4. Calcolo T.I.
5. Grafo di raggiungibilità
6. Proprietà della Rdp
7. ...

1. Matrice di incidenza

	T1	T2	T3	T4
P1	-1	1	0	0
P2	1	-1	0	0
P3	0	1	-1	0
P4	0	0	-1	1
P5	0	0	1	-1

ATTENZIONE AGLI AUTOANELLI...



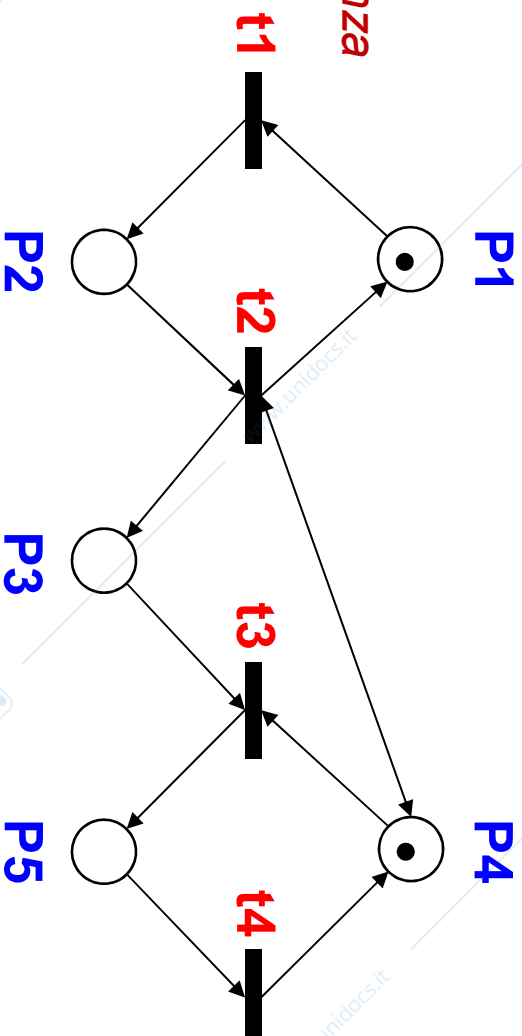
Trasmettitore e ricevitore (4)



2. Classe della Rdp

AUTOANELLI → non è possibile usare i metodi sulla matrice di incidenza

	T1	T2	T3	T4
P1	-1	1	0	0
P2	1	-1	0	0
P3	0	1	-1	0
P4	0	0	-1	1
P5	0	0	1	-1



Macchina a stati finiti → **No** (es: t2)

Grafo marcato → **No** (es: P4)

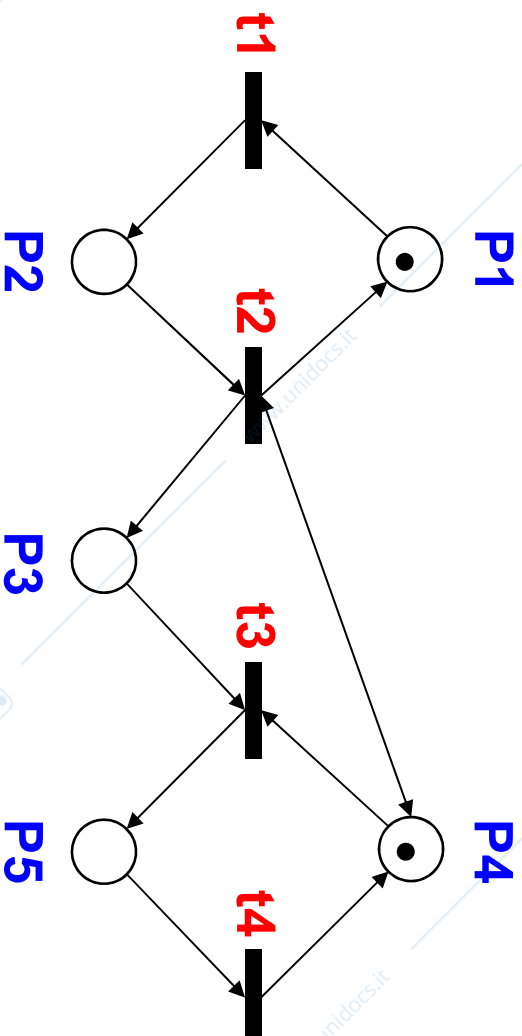
Rete a scelta libera → **No** (es: P4&t2)

Trasmettitore e ricevitore (5)



3. Calcolo P.I. e conservatività

	T1	T2	T3	T4
P1	-1	1	0	0
P2	1	-1	0	0
P3	0	1	-1	0
P4	0	0	-1	1
P5	0	0	1	-1



$$\begin{cases} -x_1 + x_2 = 0 \\ x_1 - x_2 + x_3 = 0 \\ -x_3 - x_4 + x_5 = 0 \\ x_4 - x_5 = 0 \end{cases} \Rightarrow \begin{cases} x_1 = x_2 \\ x_3 = 0 \\ x_5 = x_4 \end{cases}$$

$$P1 = \begin{bmatrix} 1 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad P12 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}$$

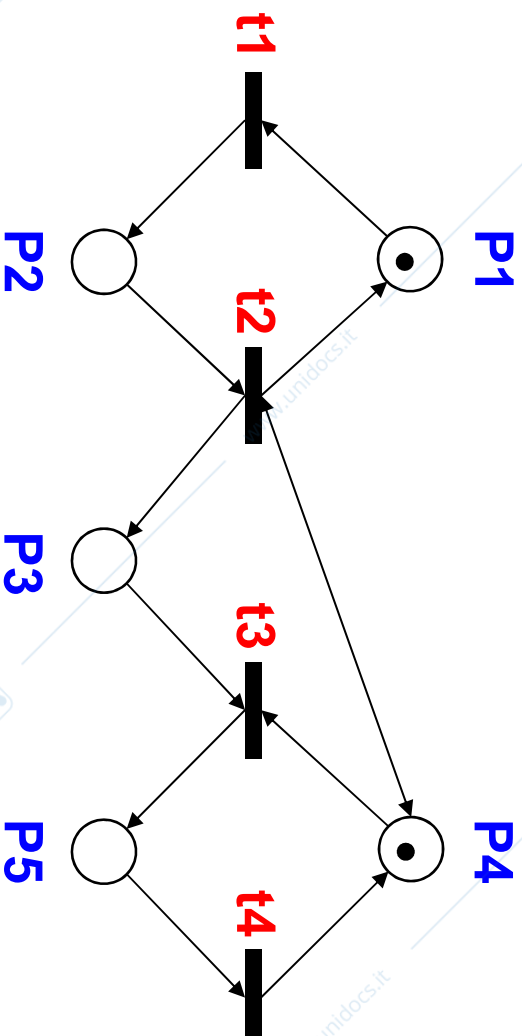
Non conservativa

Trasmettitore e ricevitore (6)



4. Calcolo T.I.

	T1	T2	T3	T4
P1	-1	1	0	0
P2	1	-1	0	0
P3	0	1	-1	0
P4	0	0	-1	1
P5	0	0	1	-1



$$\begin{cases} -x_1 + x_2 = 0 \\ x_1 - x_2 = 0 \\ x_2 - x_3 = 0 \\ -x_3 + x_4 = 0 \\ x_3 - x_4 = 0 \end{cases} \Rightarrow$$

$$\begin{cases} x_1 = x_2 \\ x_2 = x_3 \\ x_3 = x_4 \end{cases}$$

$$T11 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

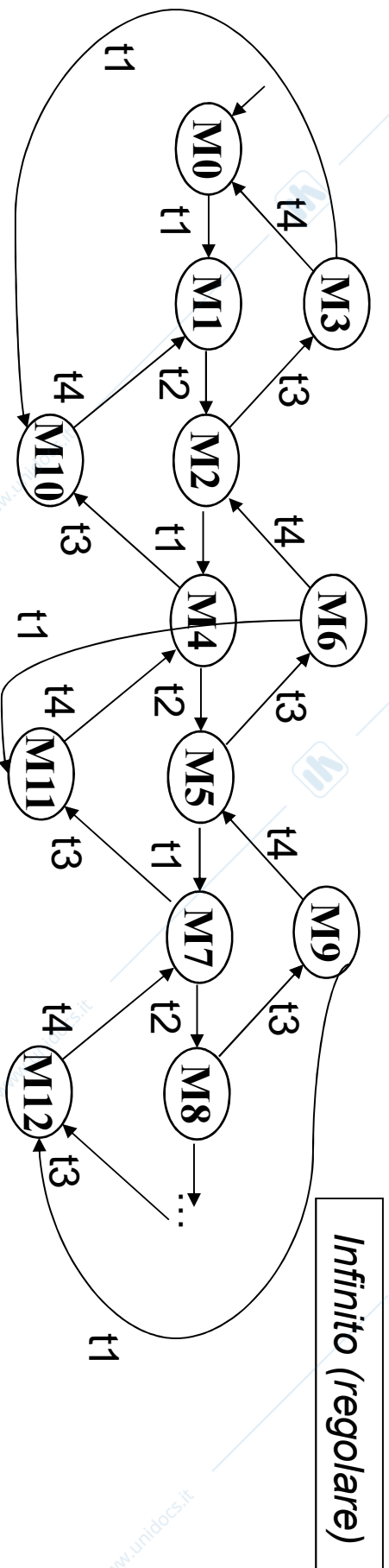
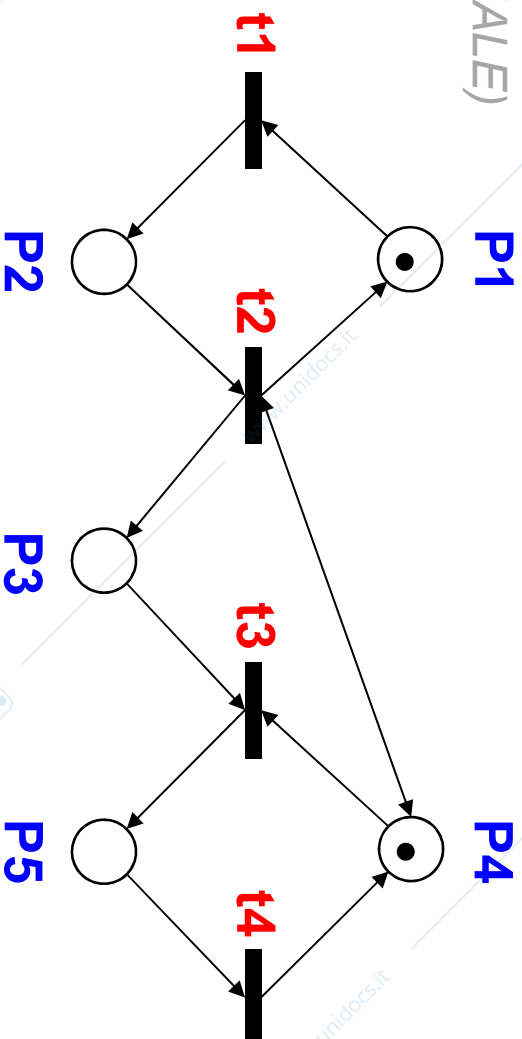
POTREBBE essere reversibile...

Trasmettitore e ricevitore (7)



5. Grafo di raggiungibilità (PARZIALE)

	T1	T2	T3	T4
P1	-1	1	0	0
P2	1	-1	0	0
P3	0	1	-1	0
P4	0	0	-1	1
P5	0	0	1	-1



Infinito (regolare)

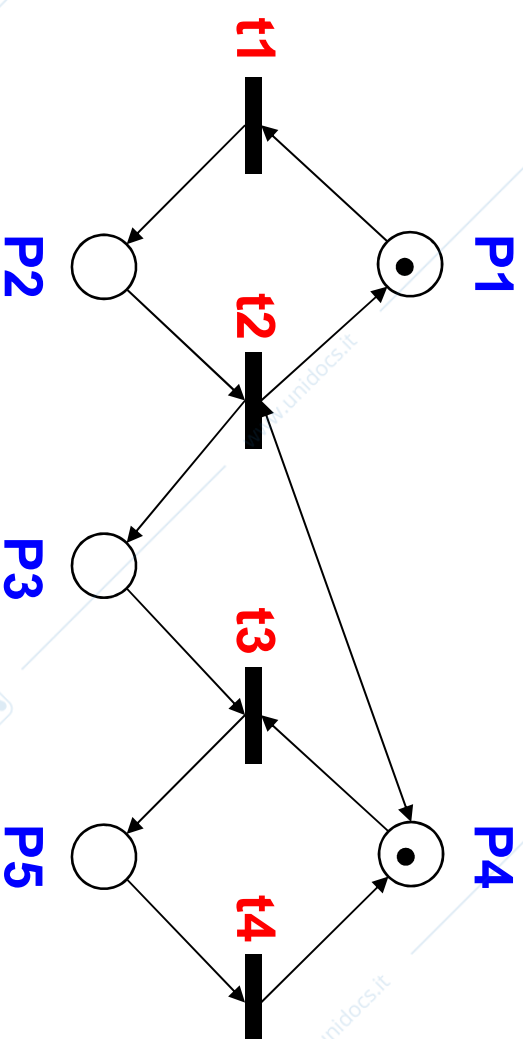
Prof. Luca Ferrarini

Trasmettitore e ricevitore (8)



6. Proprietà della Rdp

Nonostante la relativa semplicità del modello, l'analisi delle sue proprietà non è banale



Limitata \longrightarrow **No** (grafo di raggiungibilità illimitato)

Reversibile \longrightarrow **Sì** (intuitivamente)

Viva \longrightarrow **Sì** (intuitivamente)

Trasmettitore e ricevitore (9)



- Il sistema è formato da un trasmettitore ed un ricevitore;
- Su necessità, il trasmettitore inizia a processare il messaggio da trasmettere (evento «PROC»);
- Una volta processato il messaggio, il trasmettitore stabilisce una connessione con il ricevitore – il quale deve essere disponibile per la ricezione – ed invia il messaggio (evento «SEND»);
- Il ricevitore è dotato di un buffer per i messaggi in ingresso assumibile come illimitato;
- Il ricevitore, se ha messaggi a disposizione nel proprio buffer, può iniziare a processarli («OPEN»);
- Il ricevitore finisce di processare il messaggio (evento «OK») e si riporta in stato disponibile.

VARIANTE:

- Può capitare che, nel processare un messaggio, il ricevitore rilevi un errore («ERROR»). In tal caso, esso richiede una connessione con il trasmettitore («CHECK»), il quale deve essere disponibile, per informarlo. Successivamente, il ricevitore torna disponibile.

Trasmittitore e ricevitore (10)

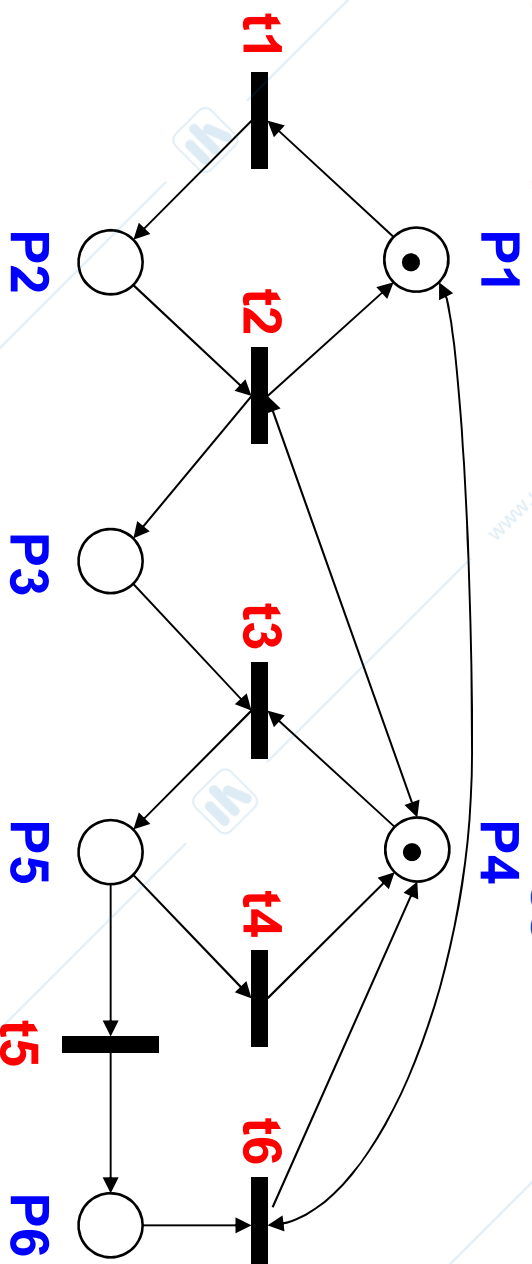


TRANSIZIONI (eventi):

- t1 (PROC)
- t2 (SEND)
- t3 (OPEN)
- t4 (OK)
- t5 (ERROR)
- t6 (CHECK)

POSTI:

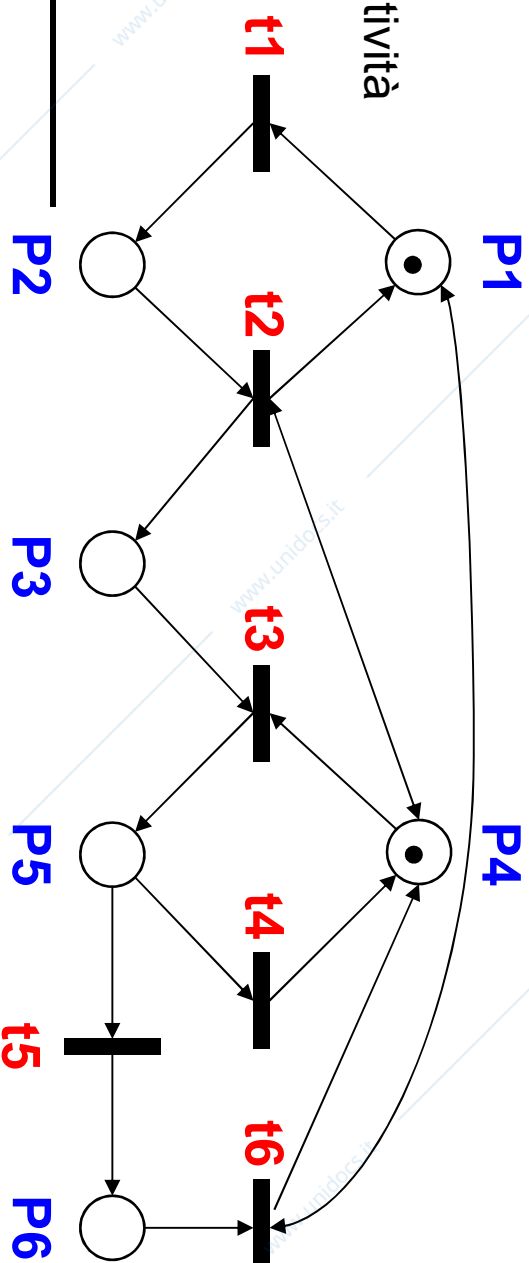
- P1: trasmettitore disponibile
- P2: trasmettitore prepara messaggio
- P3: buffer ricevitore
- P4: ricevitore disponibile
- P5: ricevitore elabora messaggio
- P6: messaggio corretto da rispedire



Trasmettitore e ricevitore (11)



1. Matrice di incidenza
2. Classe della Rdp
3. Calcolo P.I. e conservatività
4. Calcolo T.I.
5. Grafo di raggiungibilità
6. Proprietà della Rdp
7. ...



1. Matrice di incidenza

	T1	T2	T3	T4	T5	T6
P1	-1	1	0	0	0	0
P2	1	-1	0	0	0	0
P3	0	1	-1	0	0	0
P4	0	0	-1	1	0	1
P5	0	0	1	-1	-1	0
P6	0	0	0	0	1	-1

ATTENZIONE AGLI AUTOANELLI...

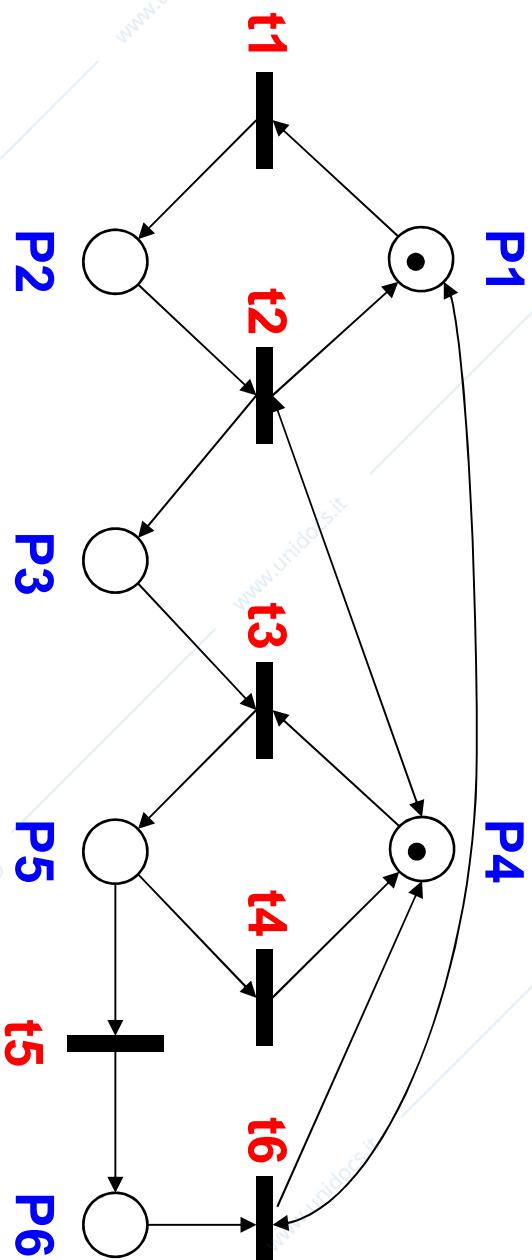
Trasmettitore e ricevitore (12)



2. Classe della Rdp

AUTOANELLI

	T1	T2	T3	T4	T5	T6
P1	-1	1	0	0	0	0
P2	1	-1	0	0	0	0
P3	0	1	-1	0	0	0
P4	0	0	-1	1	0	1
P5	0	0	1	-1	-1	0
P6	0	0	0	0	1	-1



Macchina a stati finiti \longrightarrow **No (es: t2)**

Grafo marcato \longrightarrow **No (es: P4)**

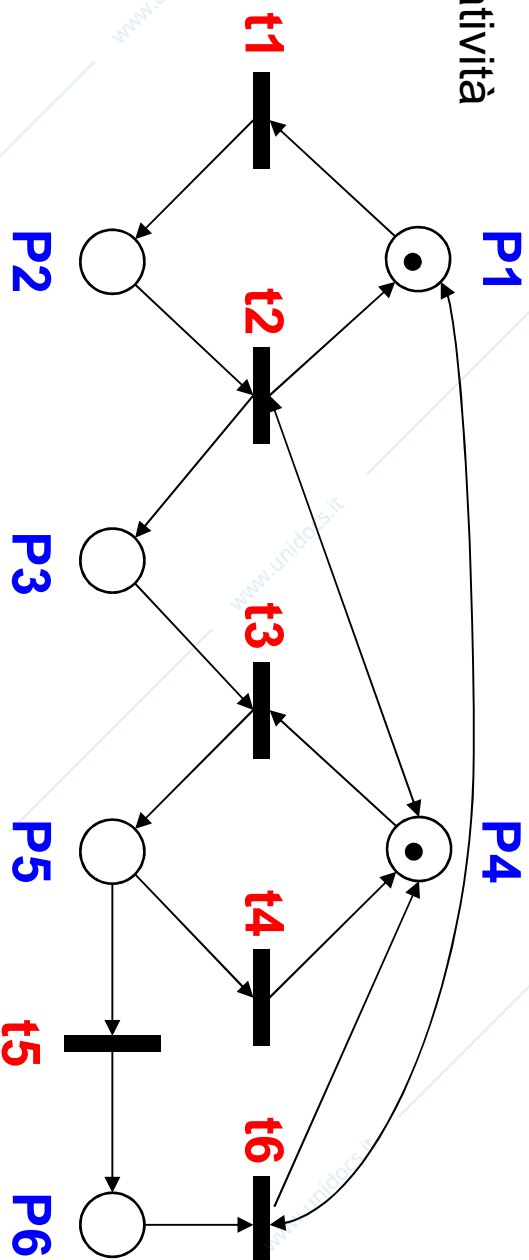
Rete a scelta libera \longrightarrow **No (es: P4&t2)**

Trasmettitore e ricevitore (13)



3. Calcolo P.I. e conservatività

	T1	T2	T3	T4	T5	T6
P1	-1	1	0	0	0	0
P2	1	-1	0	0	0	0
P3	0	1	-1	0	0	0
P4	0	0	1	-1	1	1
P5	0	0	0	1	-1	0
P6	0	0	0	0	1	-1



$$\begin{cases} -x_1 + x_2 = 0 \\ x_1 - x_2 + x_3 = 0 \\ -x_3 - x_4 + x_5 = 0 \\ x_4 - x_5 = 0 \\ -x_5 + x_6 = 0 \\ x_4 - x_6 = 0 \end{cases} \Rightarrow \begin{cases} x_1 = x_2 \\ x_3 = 0 \\ x_5 = x_4 \\ x_6 = x_4 \end{cases}$$

$$P11 = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad P12 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

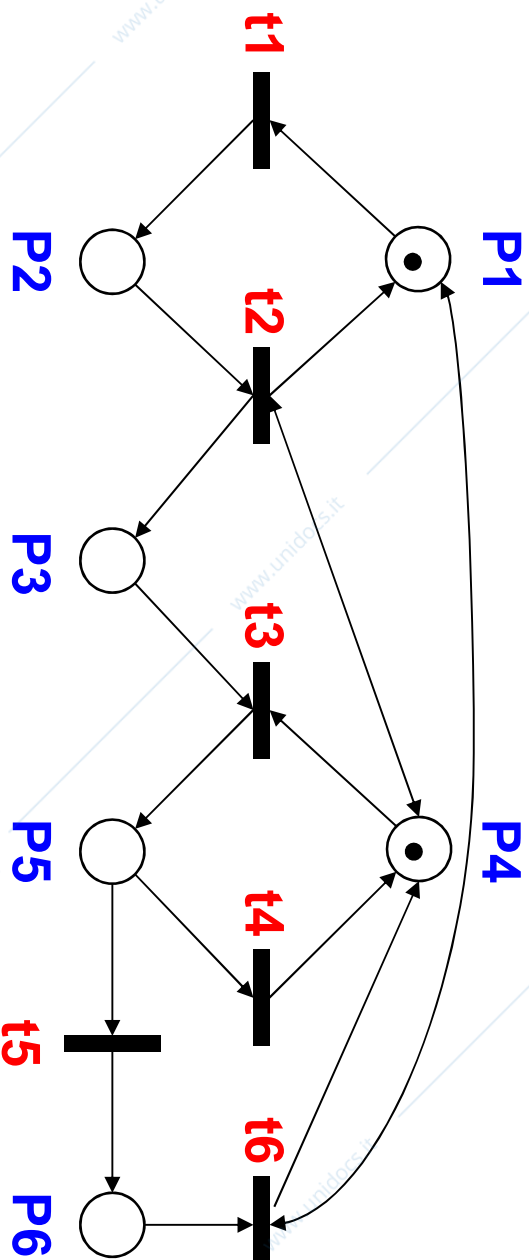
Non è cambiato nulla sulla (non) conservatività

Trasmettitore e ricevitore (14)



4. Calcolo T.I.

	T1	T2	T3	T4	T5	T6
P1	-1	1	0	0	0	0
P2	1	-1	0	0	0	0
P3	0	1	-1	0	0	0
P4	0	0	1	-1	1	1
P5	0	0	0	1	-1	0
P6	0	0	0	0	1	-1



$$\begin{cases} -x_1 + x_2 = 0 \\ x_1 - x_2 = 0 \\ x_2 - x_3 = 0 \\ -x_3 + x_4 + x_6 = 0 \\ x_3 - x_4 - x_5 = 0 \\ x_5 - x_6 = 0 \end{cases}$$

\Rightarrow

$$\begin{cases} x_1 = x_2 \\ x_2 = x_3 \\ x_3 = x_4 + x_5 \\ x_6 = x_5 \end{cases}$$

$$T11 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}, T12 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

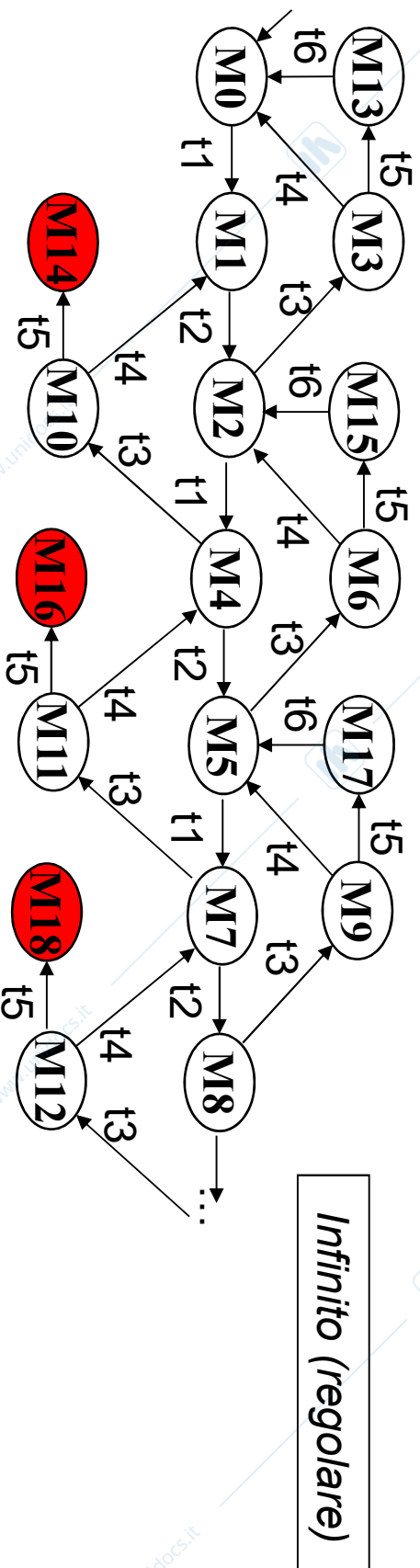
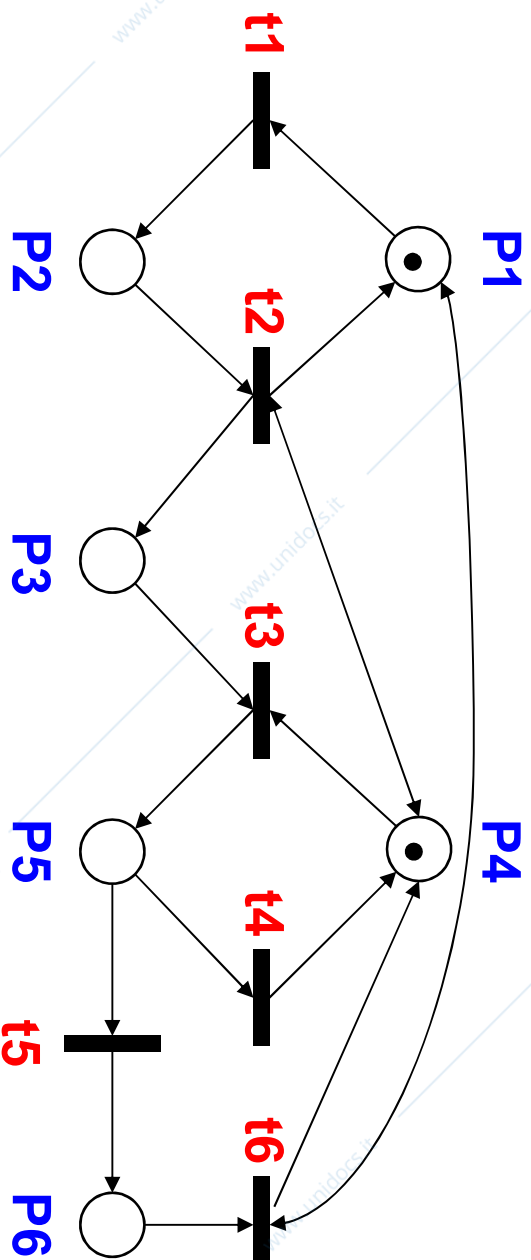
... e non è cambiato nulla sulla «forse-reversibilità»

Trasmettitore e ricevitore (15)



5. Grafo di raggiungibilità (PARZIALE)

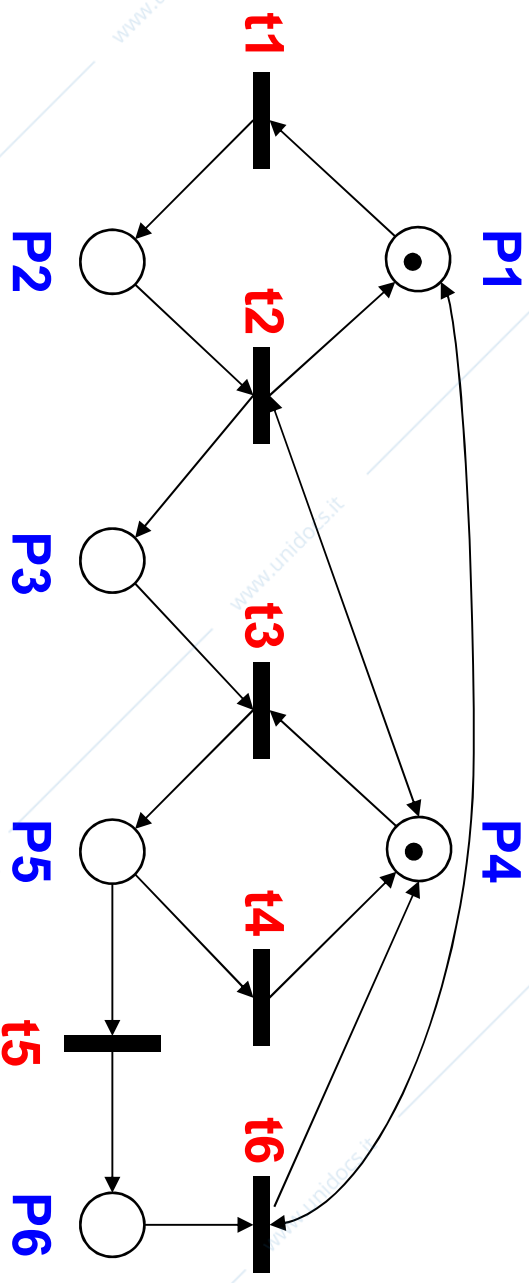
	T1	T2	T3	T4	T5	T6
P1	-1	1	0	0	0	0
P2	1	-1	0	0	0	0
P3	0	1	-1	0	0	0
P4	0	0	1	-1	1	1
P5	0	0	1	-1	-1	0
P6	0	0	0	0	1	-1



Trasmittitore e ricevitore (16)



6. Proprietà della RdpP



- Limitata → **No** (grafo di raggiungibilità illimitato)
- Reversibile → **No** (intuitivamente)
- Viva → **No** (intuitivamente)

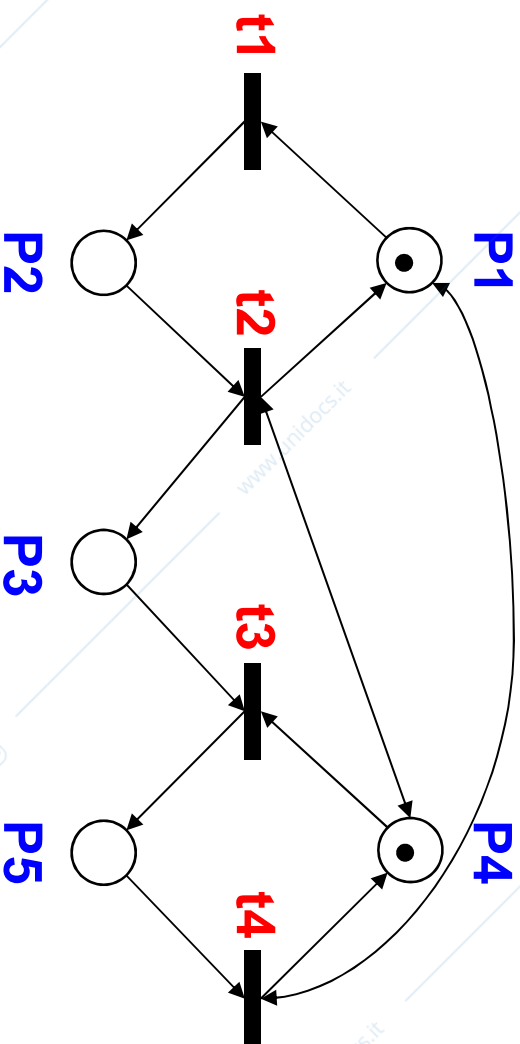
Le proprietà sono cambiate nonostante le conclusioni basate su classe, P.I. e T.I. siano le stesse...

Trasmittitore e ricevitore (17)



PARENTESI:

Avremmo potuto modellare la variante «Errore nel messaggio ricevuto» diversamente (img a destra). In tal caso, anche la matrice di incidenza sarebbe stata identica al caso base, eppure...



Non possiamo accontentarci della matrice di incidenza per analizzare una rete

7. Sifoni

Limitata → **No**

Reversibile → **No**

Viva → **No**

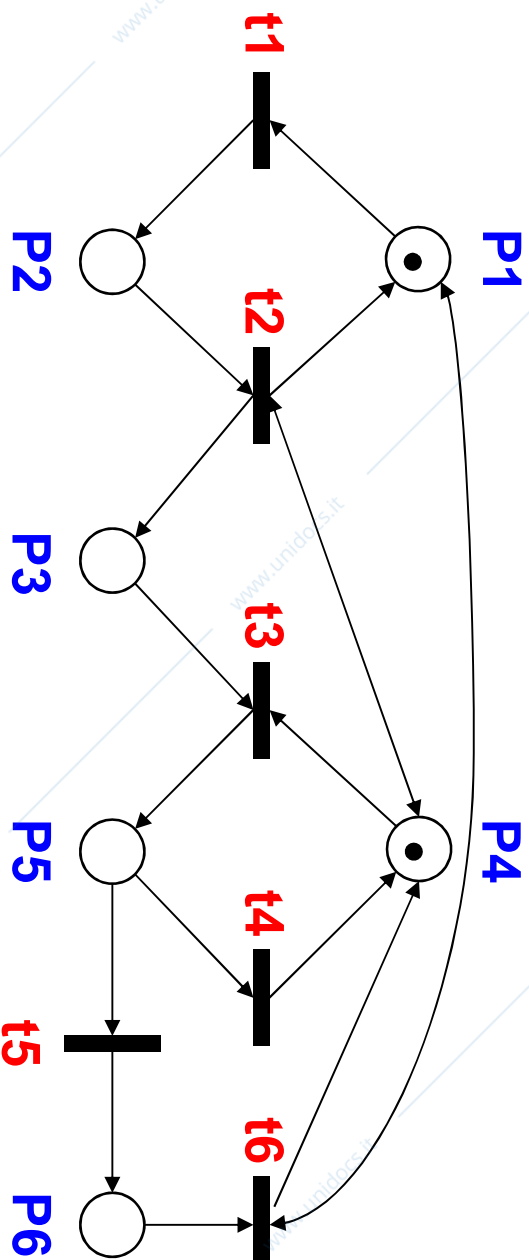
Trasmettitore e ricevitore (18)



7. Studio dei sifoni

AUTOANELLI

	•P	P•
P1	t2, t6	t1, t6
P2	t1	t2
P3	t2	t3
P4	t2, t4, t6	t2, t3
P5	t3	t4, t5
P6	t5	t6



P1-minimi	•S	S•
1	<u>2</u> 6	16
→1 <u>2</u>	261	162
→1 <u>4</u>	26 <u>4</u>	1623
→1 <u>4</u> 5	2643	162345

P2-minimi	•S	S•
2	<u>2</u> 6	16
→1 <u>2</u>	OLD	OLD

Trasmettitore e ricevitore (19)



P3-minimi	•S	S•
3	<u>2</u>	3
→3 <u>2</u>	2 <u>1</u>	3 2
→3 <u>2</u> 1	2 1 <u>6</u>	3 2 1 6
→3 <u>2</u> 1 <u>1</u>	2 4 <u>6</u>	3 2
→3 <u>4</u>	2 4 <u>6</u> 3	3 2 4 5
→3 <u>4</u> 5	2 4 6 3 5	3 2 4 5 6
→3 <u>4</u> 5 <u>6</u>	2 4 6	3 2 1 6
→3 <u>4</u> 1	2 4 6 3	3 2 1 6 4 5
→3 <u>4</u> 1 <u>5</u>	2 4 6 5	3 2 6
→3 <u>4</u> 6	OLD	OLD
→3 <u>4</u> 6 <u>5</u>		

P4-minimi	•S	S•
4	2 4 <u>6</u>	2 3
→4 <u>5</u>	2 4 6 <u>3</u>	2 3 4 5
→4 <u>5</u> 6	2 4 6 3 5	2 3 4 5 6
→4 <u>1</u>	2 4 6	2 3 1 6
→4 <u>1</u> 5	OLD	OLD
→4 <u>6</u>	2 4 6 5	2 3 6
→4 <u>6</u> 5	OLD	OLD

Trasmettitore e ricevitore (20)



P5-minimi	•S	S•
5	<u>3</u>	4 5
↘5 <u>3</u>	3 <u>2</u>	4 5 3
↘↘5 <u>3 2</u>	3 2 <u>1</u>	4 5 3 2
↘↘↘5 <u>3 2 1</u>	3 2 1 6	4 5 3 2 1 6
↘↘↘↘5 <u>3 4</u>	3 2 4 <u>6</u>	4 5 3 2
↘↘↘↘↘5 <u>3 4 6</u>	OLD	OLD
↘↘↘↘↘↘5 <u>4</u>	3 2 4 <u>6</u>	4 5 2 3
↘↘↘↘↘↘↘5 <u>4 6</u>	OLD	OLD

P6-minimi	•S	S•
6	<u>5</u>	6
↘6 <u>5</u>	5 3	6 4 5
↘↘6 <u>5 3</u>	5 3 2	6 4 5 3
↘↘↘6 <u>5 3 4</u>	OLD	OLD
↘↘↘↘6 <u>5 3 2</u>	5 3 2 1	6 4 5 3 2
↘↘↘↘↘6 <u>5 3 2 1</u>	5 3 2 1 6	6 4 5 3 2 1

Trasmettitore e ricevitore (21)



Sifoni	P-minimo per	Minimo?	Pericoloso?
P1 P2	P1	SI	NO (P1)
P1 P4 P5	P1 P2 P4	SI	SI
P1 P2 P3	P3	NO	---
P3 P4 P5 P6	P3 P6	NO	---
P1 P3 P4 P5	P3	NO	---
P4 P5 P6	P4 P5	SI	NO (P1)
P1 P2 P3 P5	P5	NO	---
P1 P2 P3 P5 P6	P6	NO	---

CONCLUSIONE:

Alla fine il problema della deadlock sta nel sifone {P1, P4, P5}.

In questo semplice caso, avremmo potuto identificare questo sifone semplicemente dalle deadlock (i.e. l'insieme degli stati smarcati nelle infinite deadlock).

Trasmettitore e ricevitore (22)



8. Controllo supervisivo PI-based

Impedire svuotamento di $\{P1, P4, P5\}$

$$m_1 + m_4 + m_5 \geq 1$$

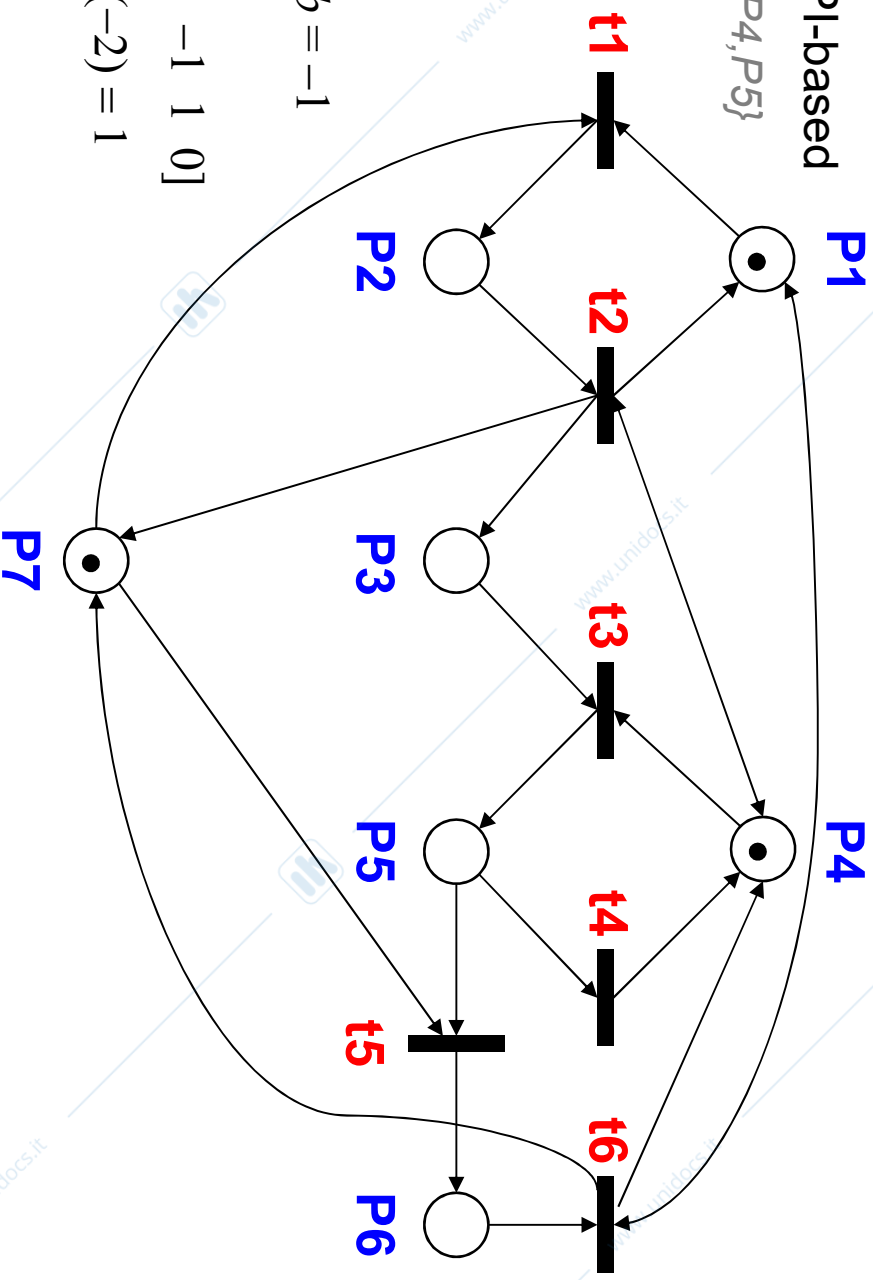
$$\downarrow$$

$$-m_1 - m_4 - m_5 \leq -1$$

$$L = [-1 \ 0 \ 0 \ -1 \ -1 \ 0] \quad b = -1$$

$$C_c = -L \cdot C_p = [-1 \ 1 \ 0 \ 0 \ -1 \ 1 \ 0]$$

$$M_{c0} = b - L \cdot M_{p0} = -1 - (-2) = 1$$



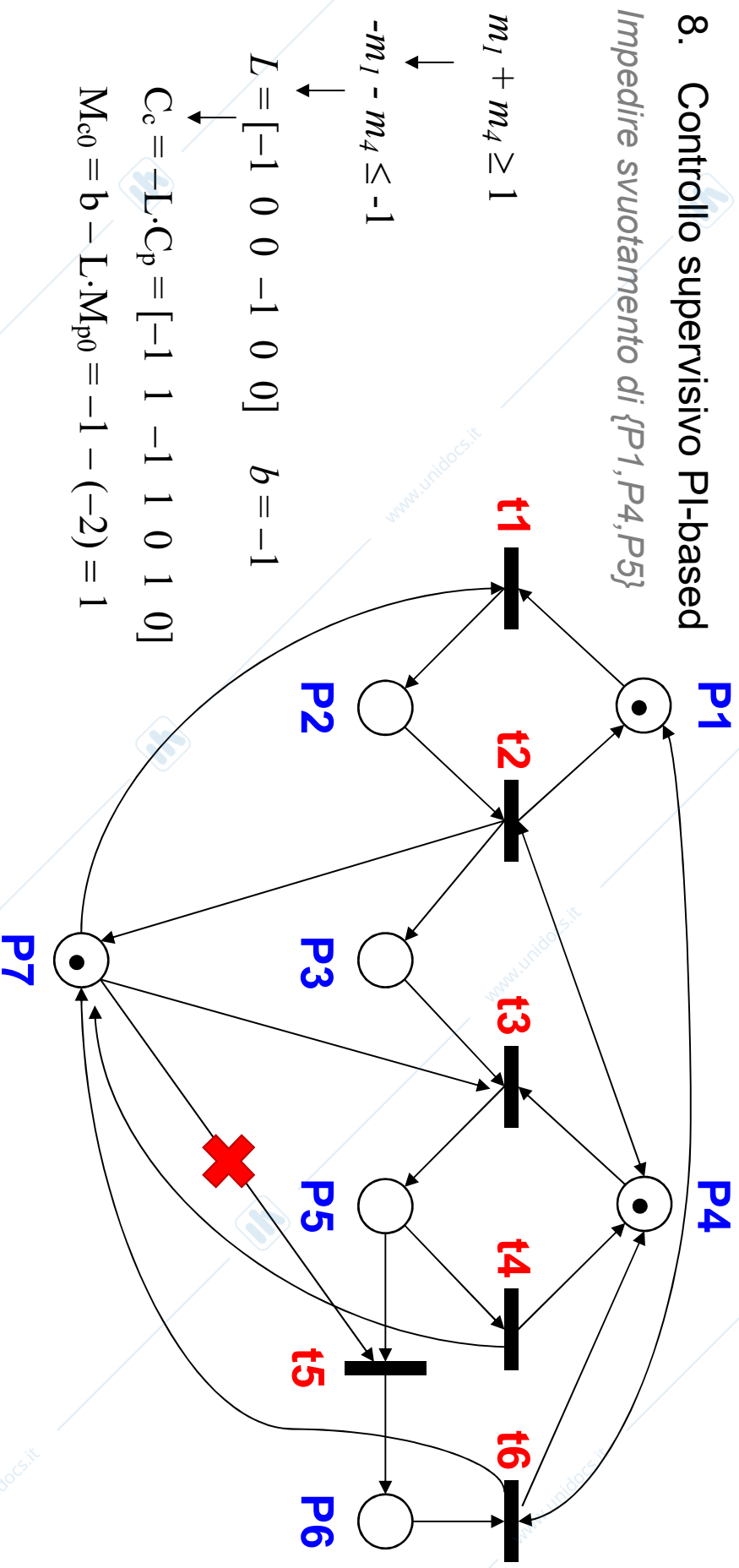
E se considerassimo t5 come non controllabile?

Trasmittitore e ricevitore (23)



8. Controllo supervisivo PI-based

Impedire svuotamento di $\{P1, P4, P5\}$



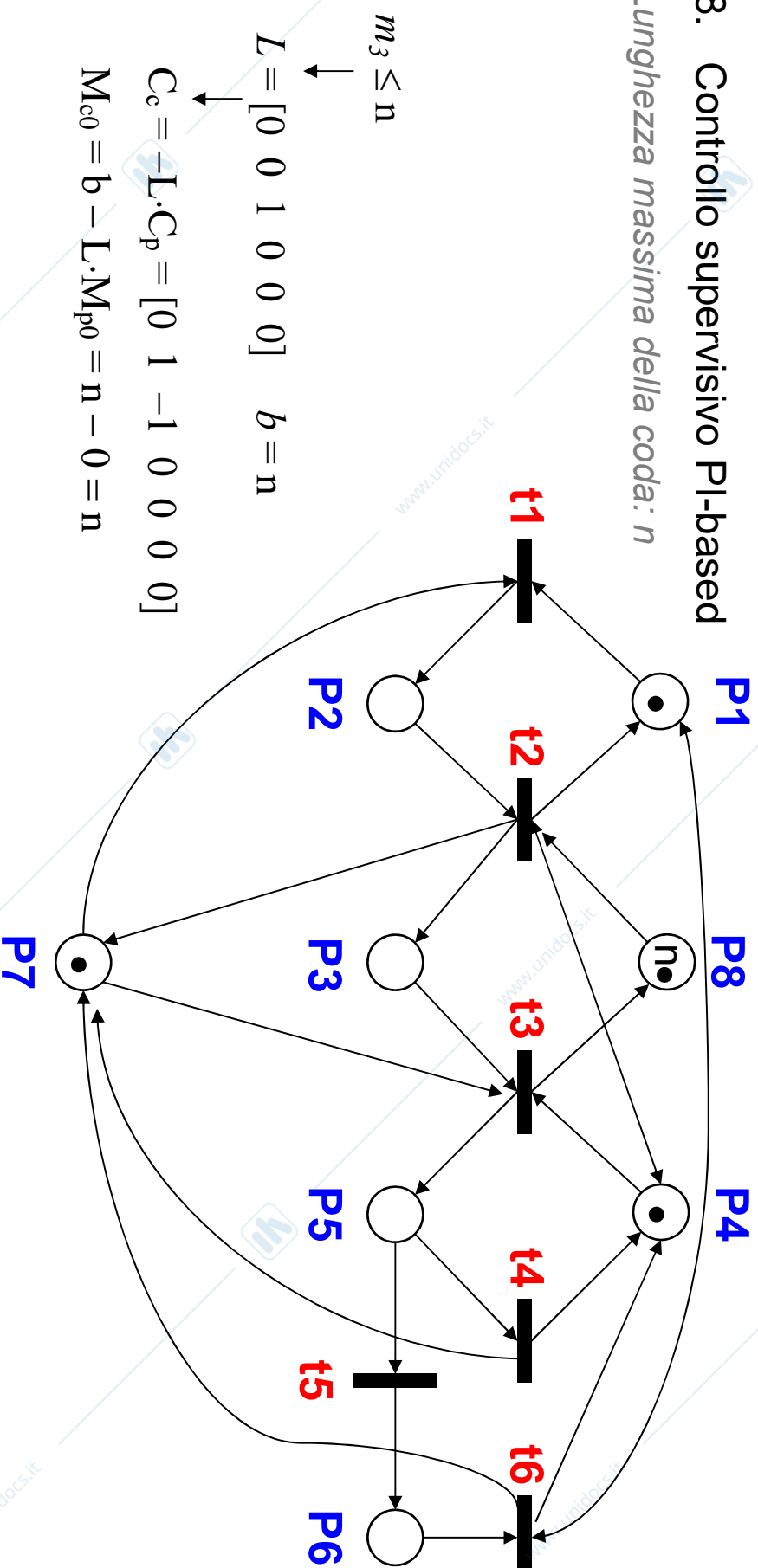
Oppure riformuliamo il vincolo!

Trasmettitore e ricevitore (24)



8. Controllo supervisivo PI-based

Lunghezza massima della coda: n



$$m_3 \leq n$$

$$L = [0 \ 0 \ 1 \ 0 \ 0 \ 0] \quad b = n$$

$$C_c = -L \cdot C_p = [0 \ 1 \ -1 \ 0 \ 0 \ 0]$$

$$M_{c0} = b - L \cdot M_{p0} = n - 0 = n$$

E se volessimo limitare la coda dei messaggi del ricevitore?

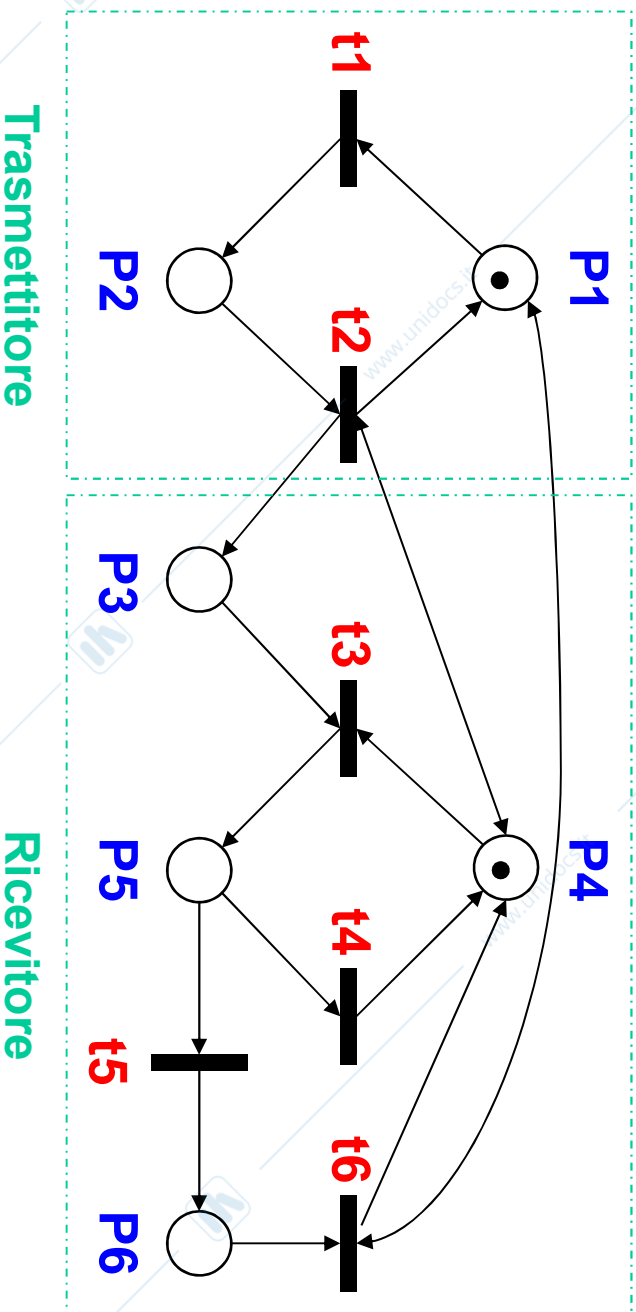
Otteniamo semplicemente la classica struttura «buffer» a 2 stati

Trasmittitore e ricevitore (25)



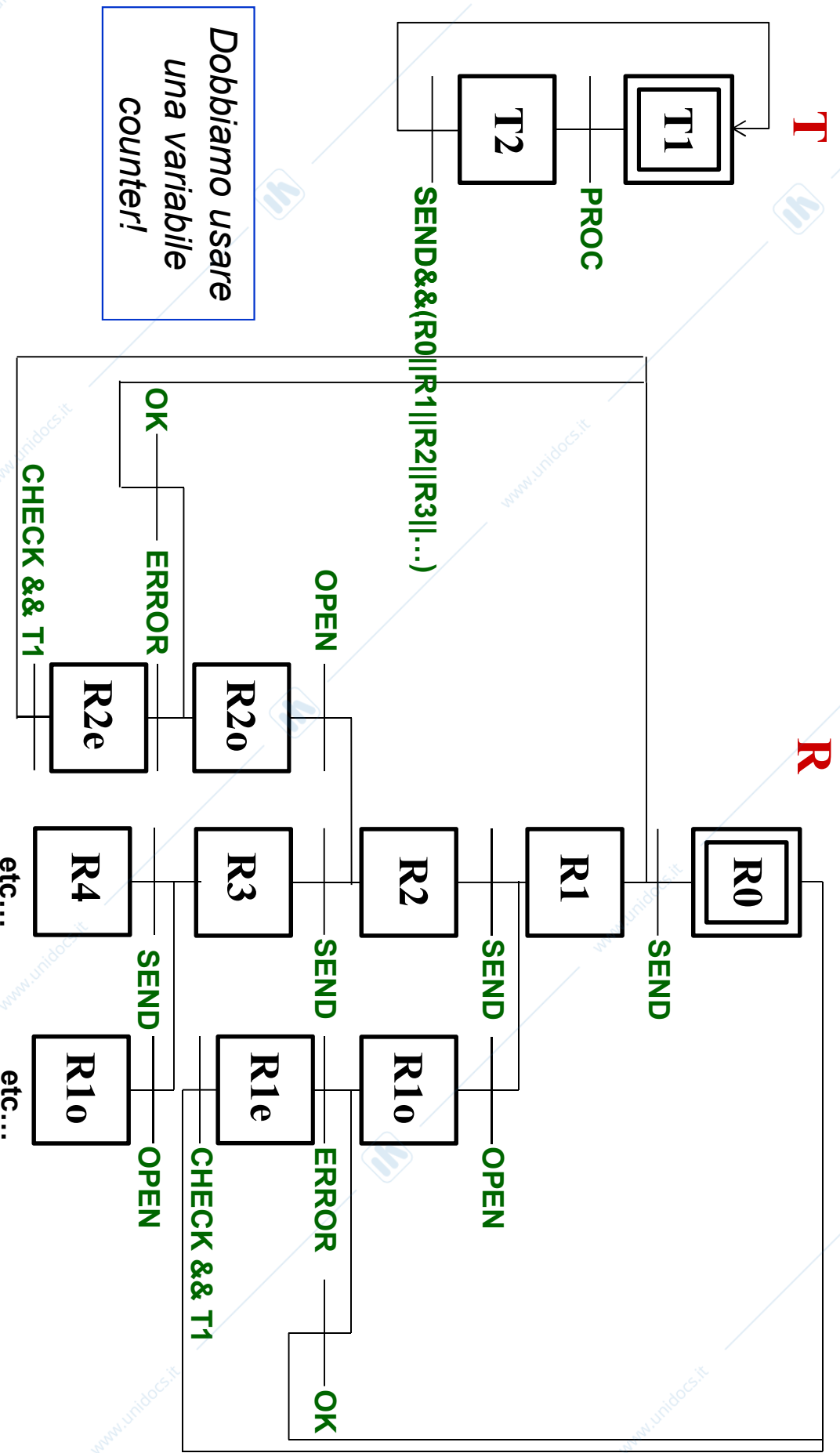
SFC

INTERPRETAZIONE
A RISORSE



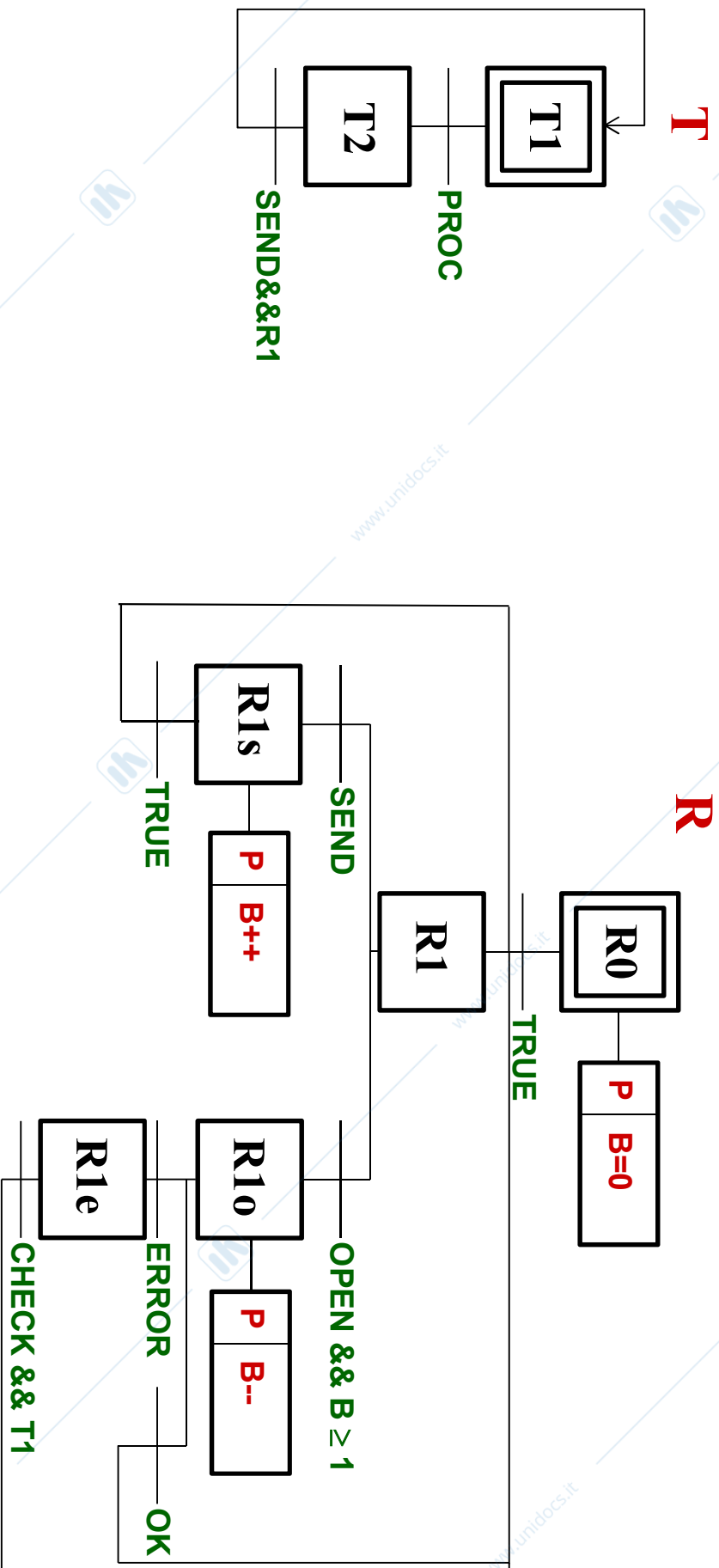
Non esistono scelte giuste nella definizione delle risorse e dei loro confini, solo scelte comode.

Trasmittitore e ricevitore (26)



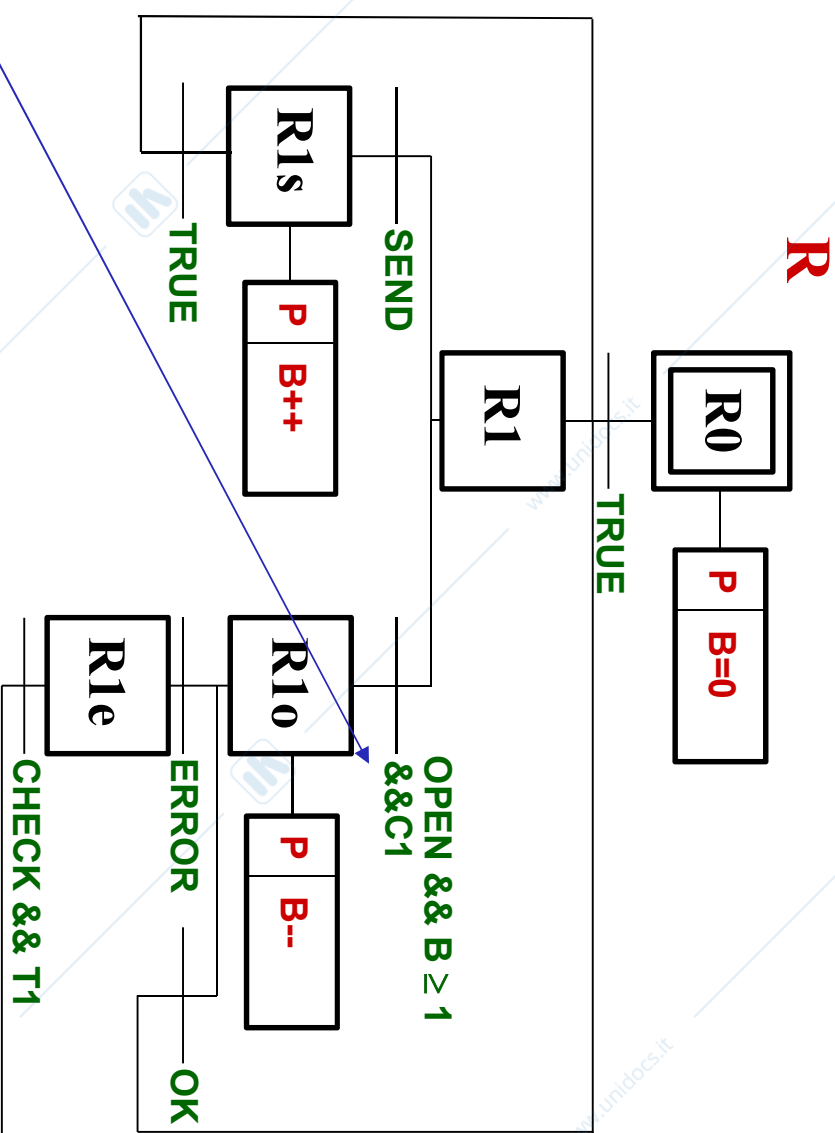
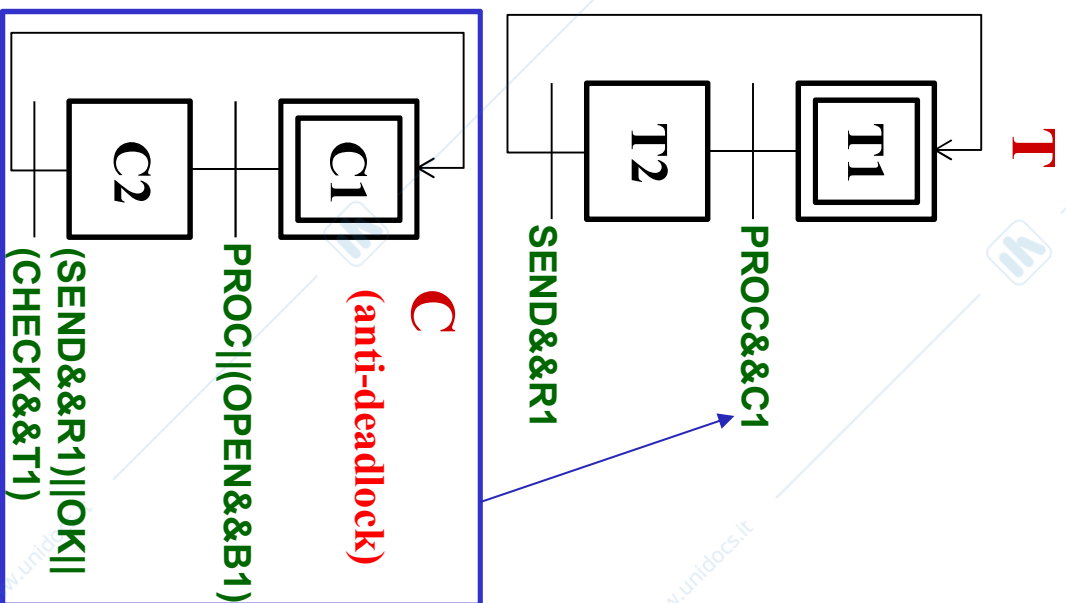
Dobbiamo usare una variabile counter!

Trasmittitore e ricevitore (27)



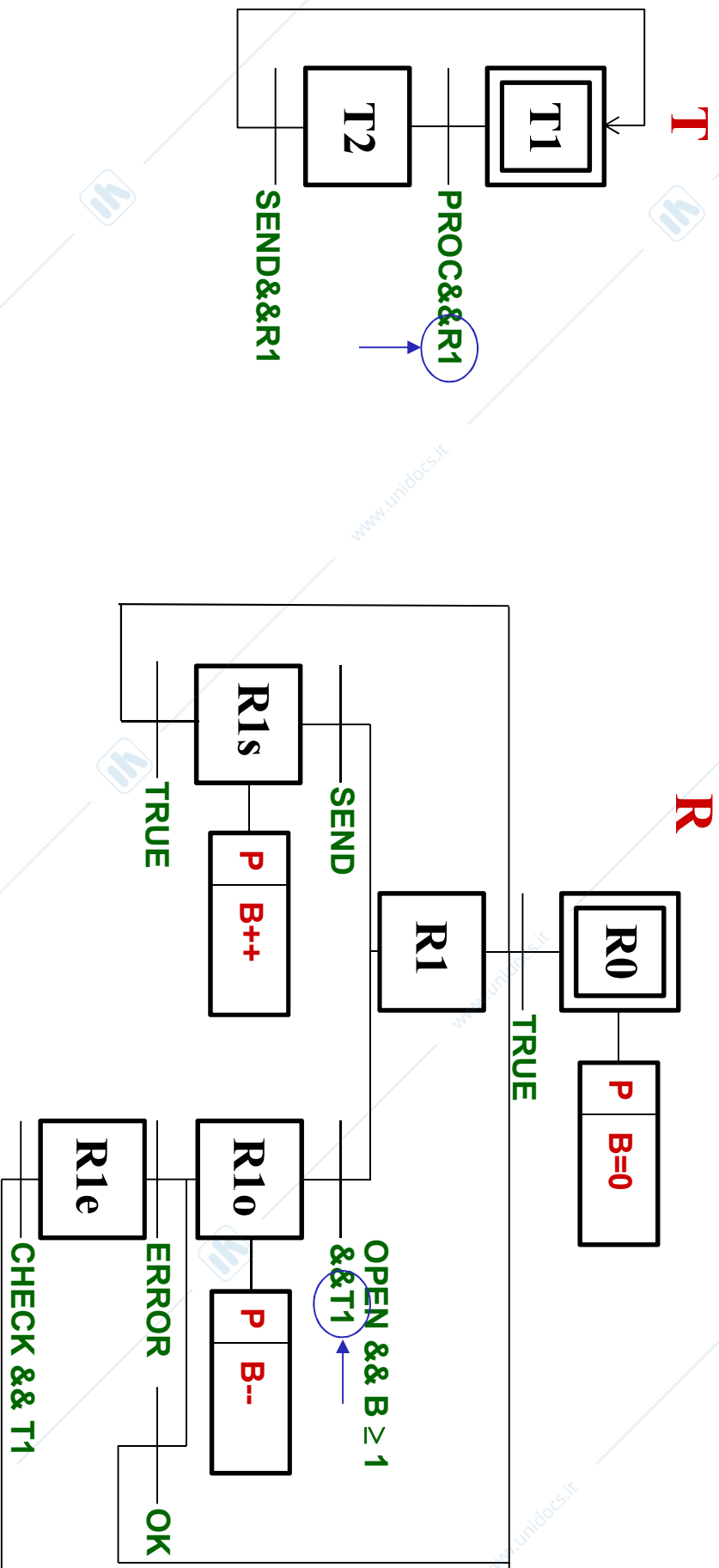
E se volessimo introdurre i controlli (anti-deadlock & limite buffer)?

Trasmittitore e ricevitore (28)

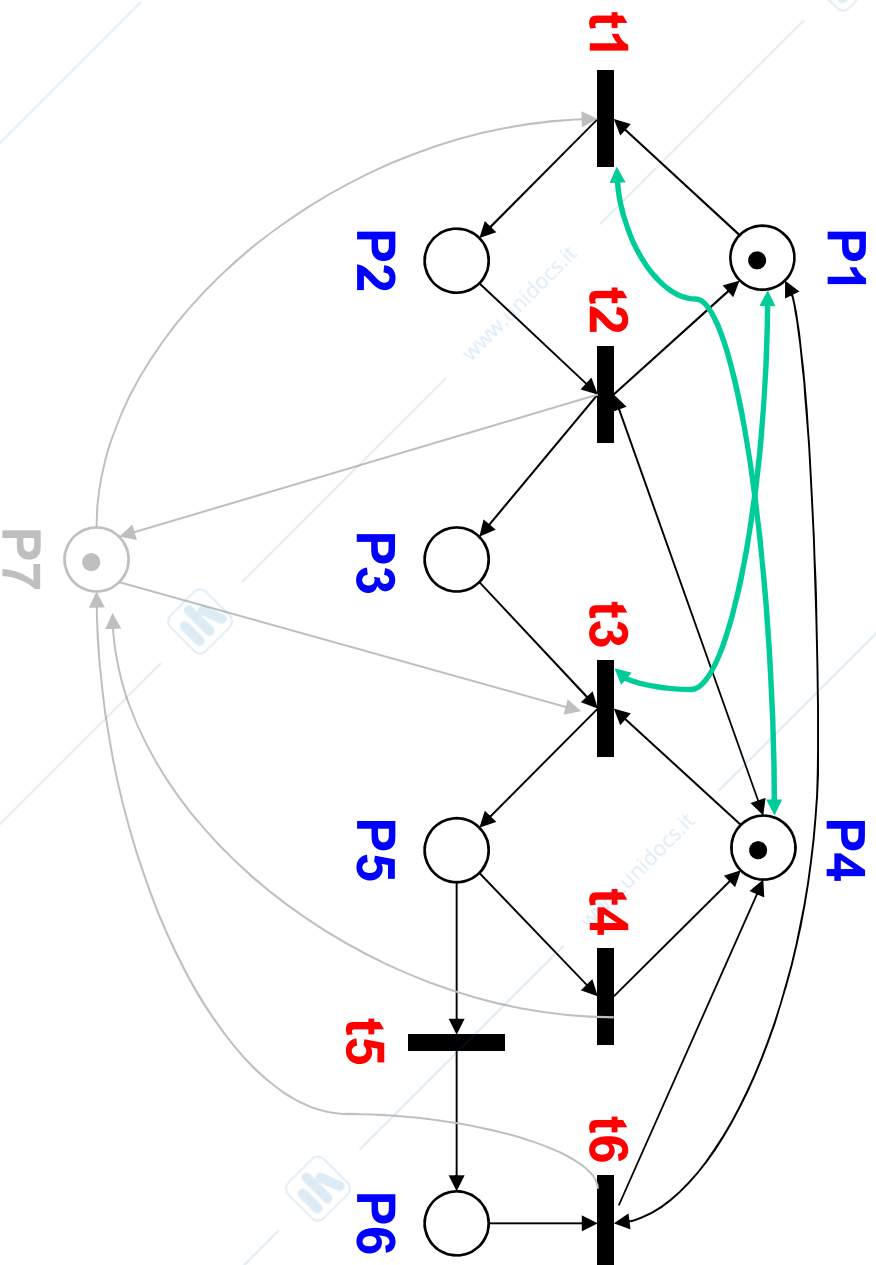


Forse possiamo evitare di aggiungere il modello del controllore...

Trasmittitore e ricevitore (29)

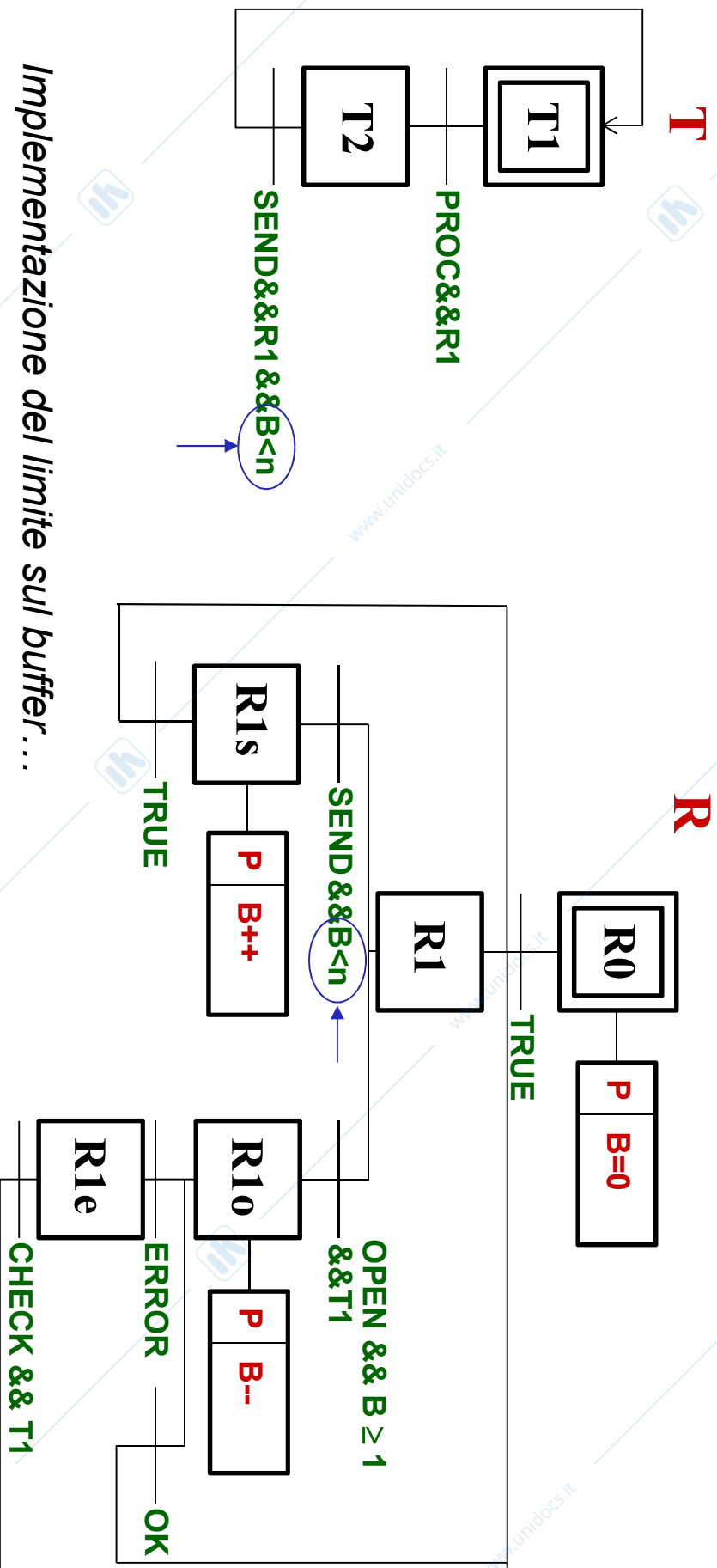


Trasmettitore e ricevitore (30)



Chiaramente, avremmo potuto fare lo stesso per la Rete di Petri!

Trasmittitore e ricevitore (31)



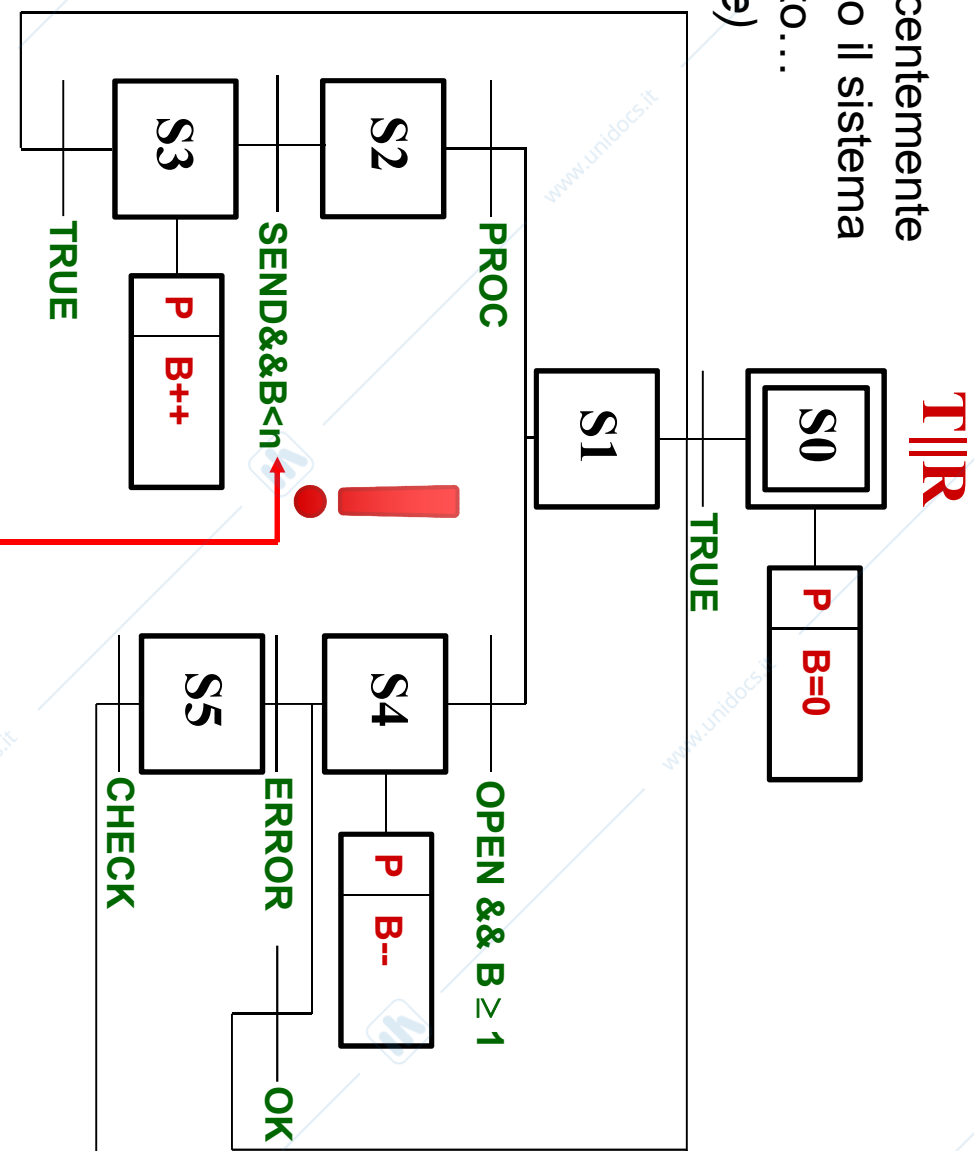
Implementazione del limite sul buffer...

Considerando tutte queste interazioni tra le transizioni dei due modelli, è probabile che un *modello centralizzato* (i.e. non la modellazione a risorse usata ora) fosse la scelta migliore...

Trasmittitore e ricevitore (32)

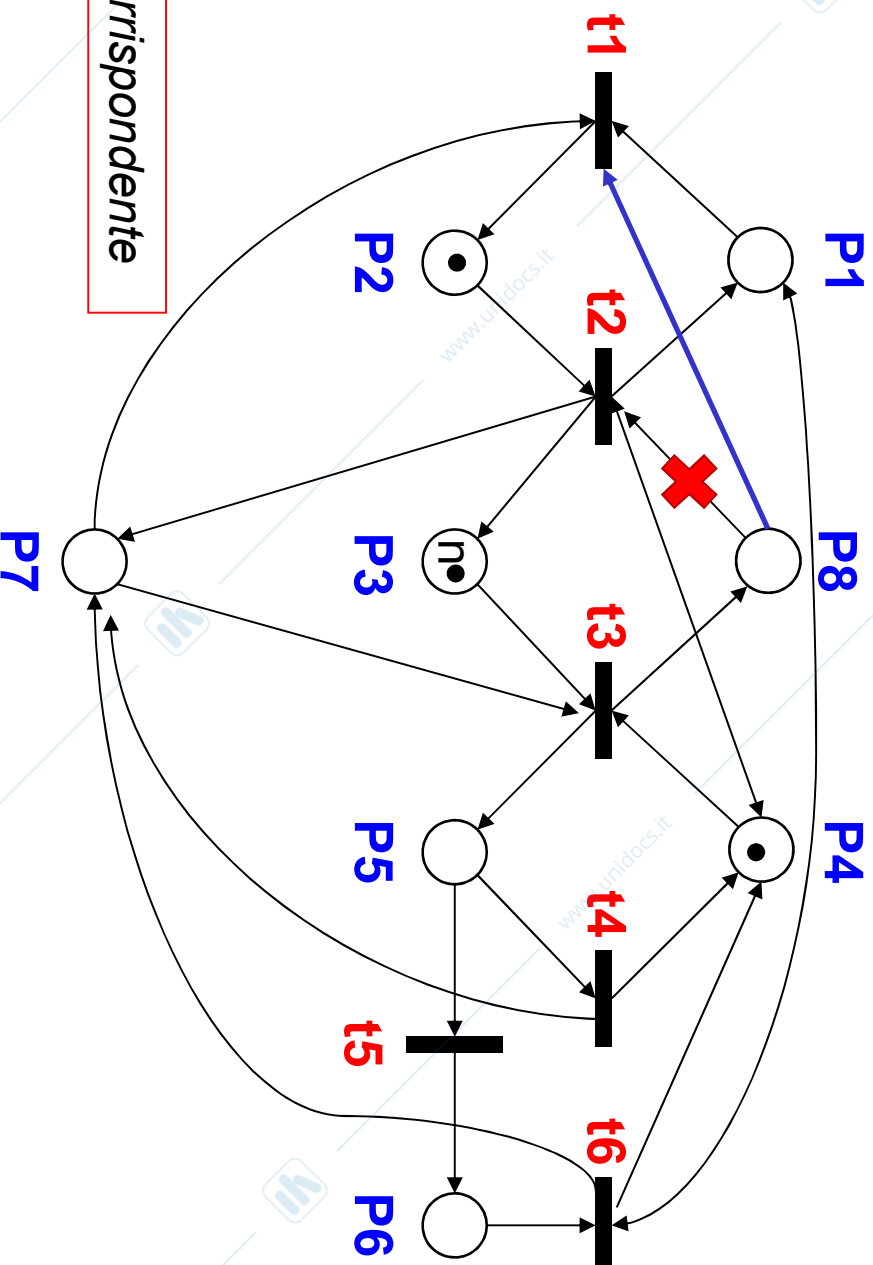


Rappresentabile decentemente solo se consideriamo il sistema anti-deadlock inserito...
(provare per credere)



Grazie ad SFC, notiamo un comportamento potenzialmente MOLTO dannoso..!

Trasmettitore e ricevitore (33)



Deadlock corrispondente

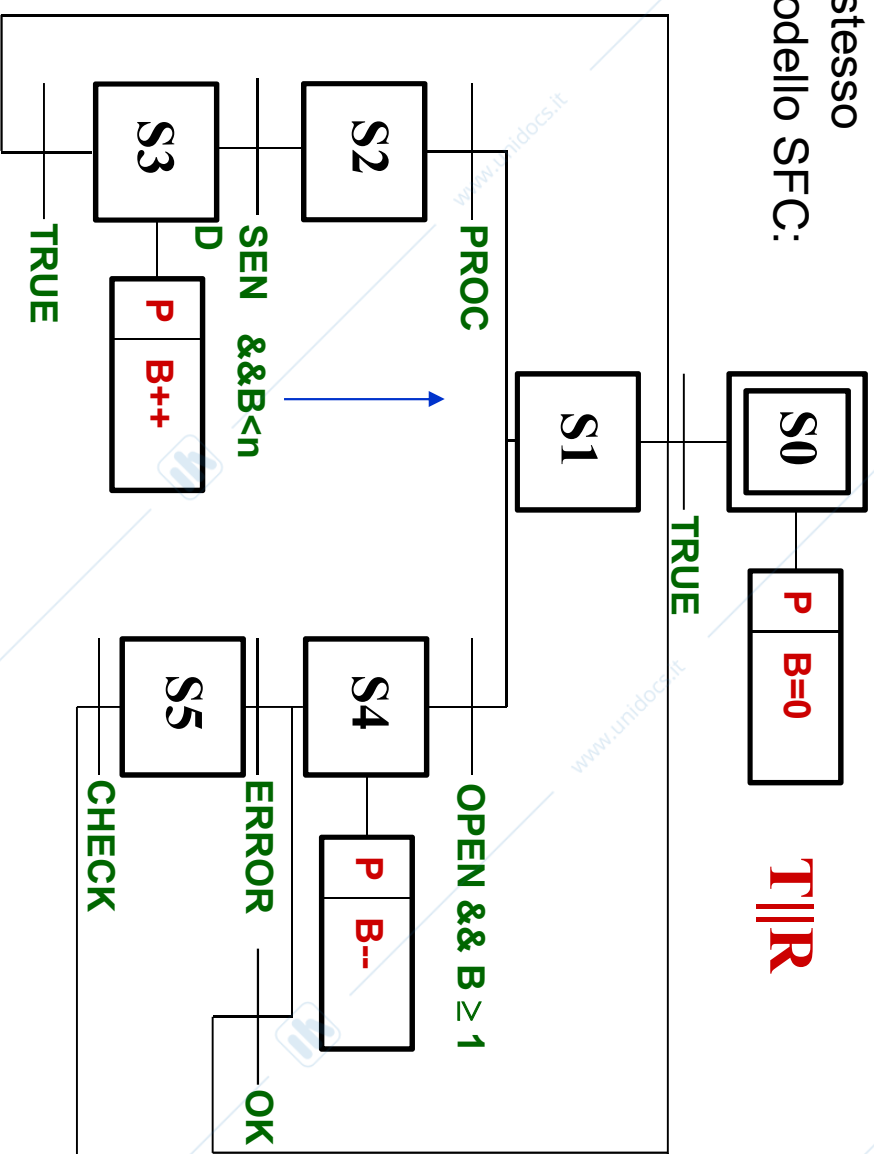
In effetti $\{P1, P5, P6, P7, P8\}$ è un sifone! Dovremmo imporre un altro controllo supervisivo P1-based.

Oppure...

Trasmittitore e ricevitore (34)



Rappresentiamo lo stesso cambiamento nel modello SFC:




NOTA FINALE: in questo esempio SFC ci siamo permessi di ignorare la condizione di mutua esclusività delle transizioni nelle strutture OR.

Prof. Luca Ferrarini

Paragone Rdp - SFC



Rdp

- Analisi formale di proprietà rilevanti per l'evoluzione del sistema (principalmente, deadlocks) 

SFC

- Deadlock non visibili, serve simulare il sistema per verificare comportamenti non desiderati

- Ottima rappresentazione di sistemi composti da elementi (semi-)indipendenti
- Per sistemi composti da elementi molto inter-dipendenti:
 - Rdp: Grafo di raggiungibilità
 - SFC: Modello completo (invece del sistema di modellazione «a risorse»)

- Difficile valutare «ad occhio» la qualità dell'evoluzione del sistema: meglio ricorrere – se possibile – al grafo di raggiungibilità
- Evoluzione del sistema visivamente intuitiva, specialmente per sistemi modellati tramite un unico modello SFC → analisi più naturale 

Trasmittitore e ricevitore (35)

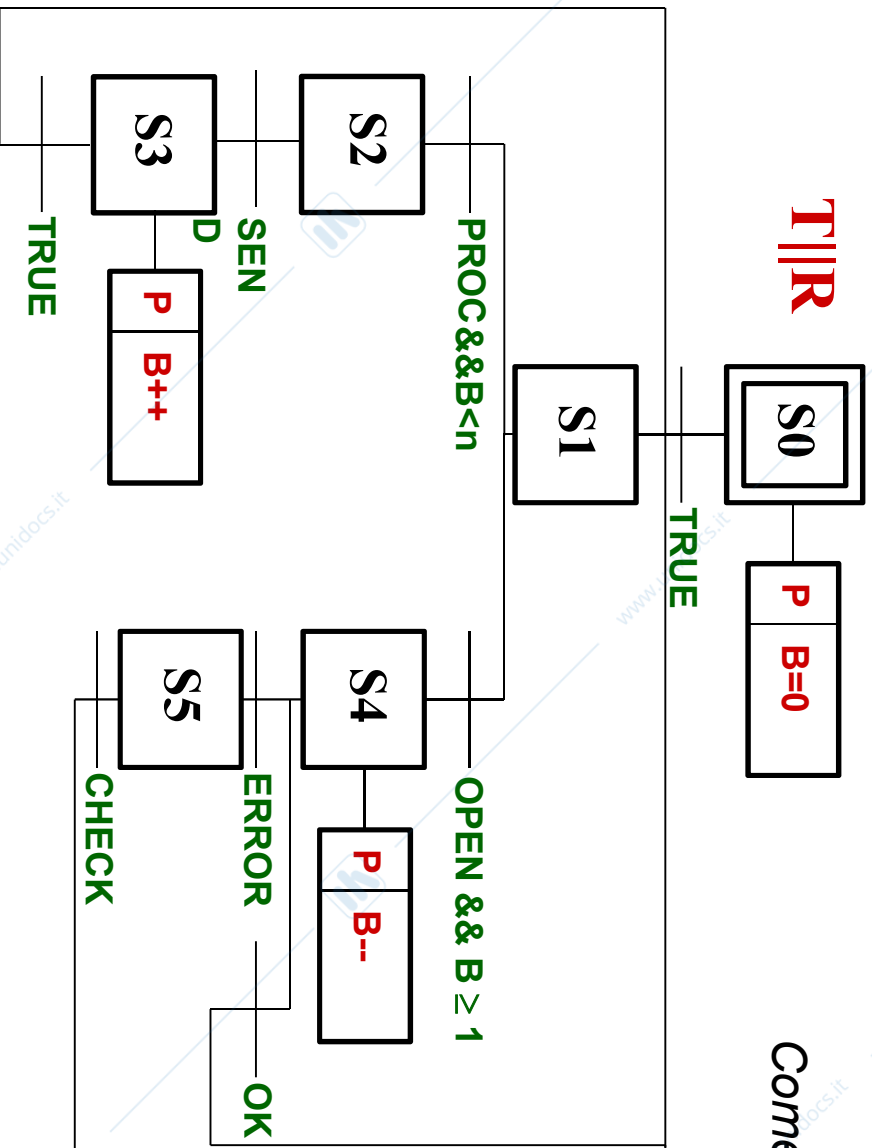


LAD

TRADUZIONE DAL
MODELLO SFC

T||R

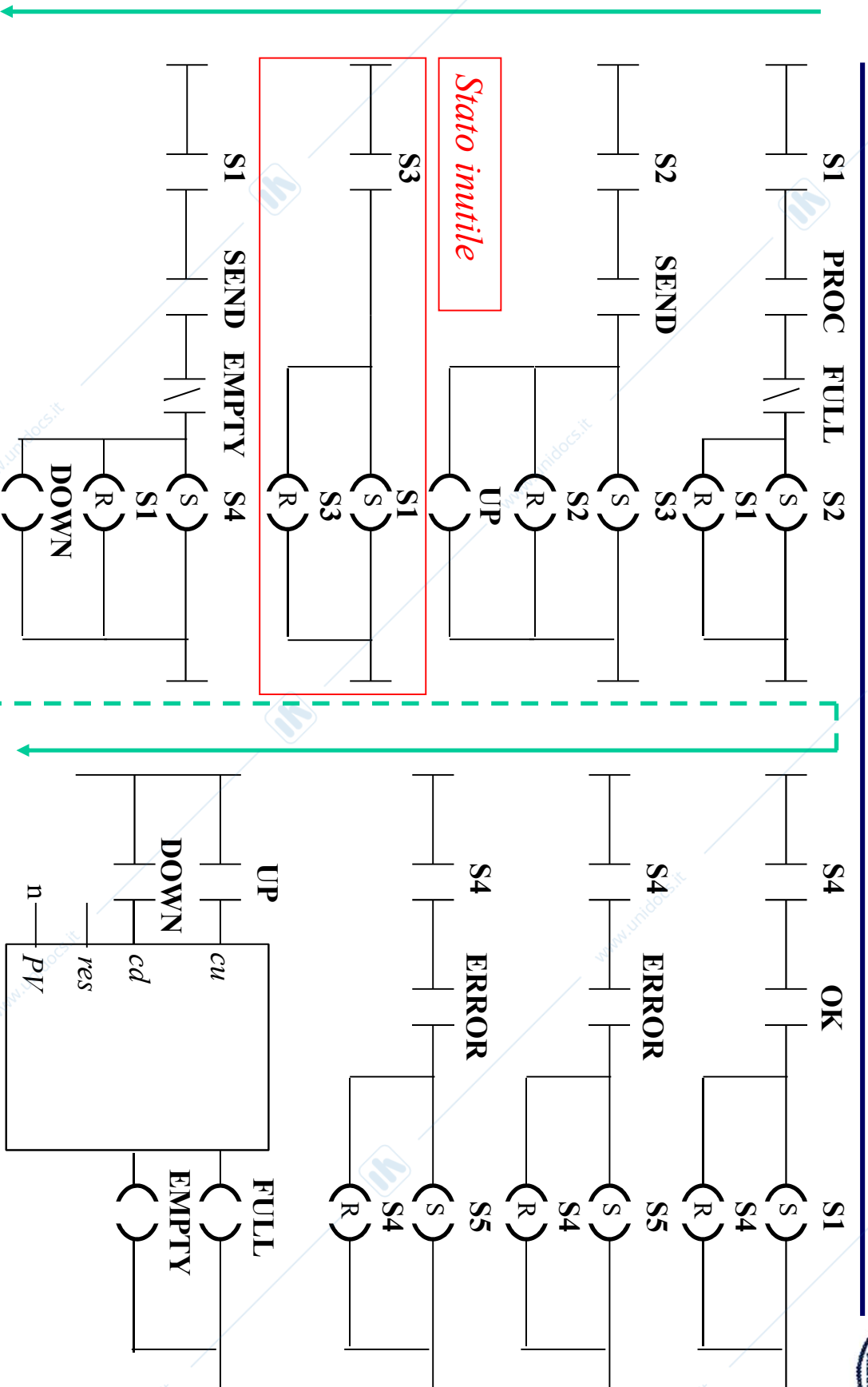
Come facciamo per il counter «B»?



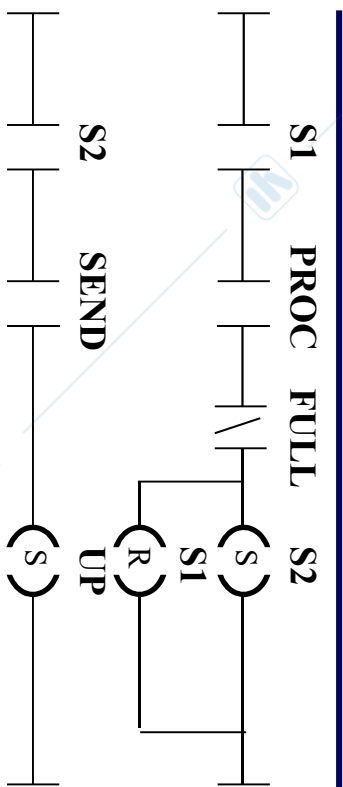
Esistono anche i contatori up-down

CTRUD	
CU	QU
CD	QD
R	
LD	
PV	CV

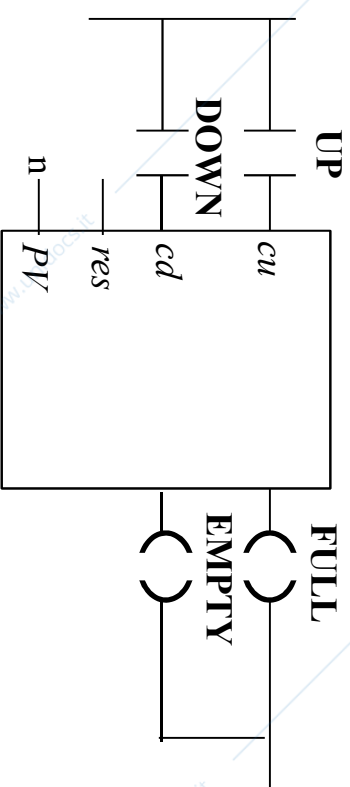
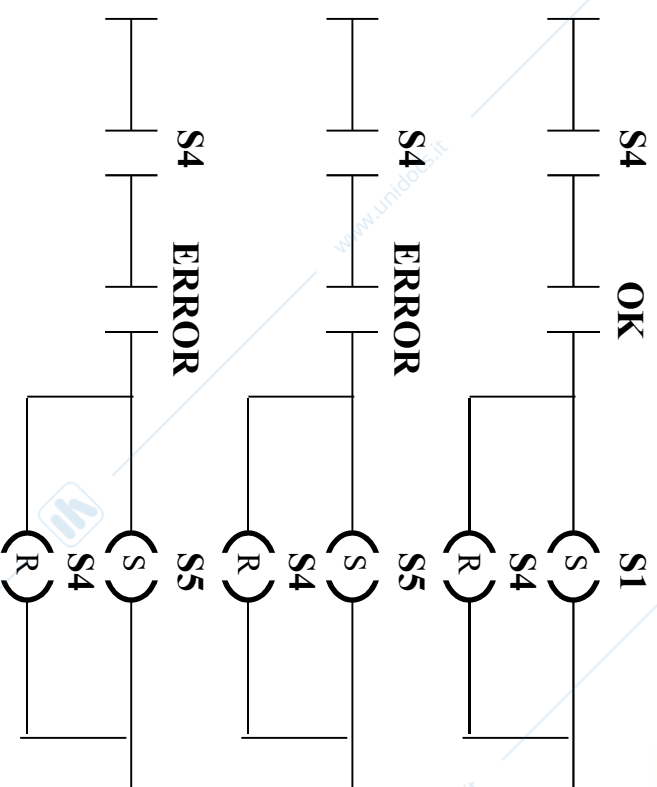
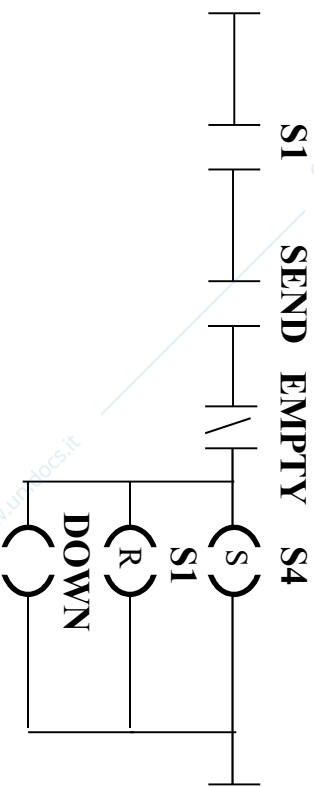
Trasmittitore e ricevitore (36)



Trasmittitore e ricevitore (37)



NOTA:
L'implementazione LAD ci ha imposto una logica sequenziale
→ La mutua esclusività degli OR nell'SFC è quindi necessariamente affrontata in LAD



Conclusioni LAD



LAD

- Il linguaggio LAD è il più comunemente usato nell'implementazione di logiche di controllo discrete nel mondo PLC
- La logica LAD impone la *sequenzialità* delle istruzioni: qualsiasi forma di indecisione lasciata nella fase di modellazione:
 - Rdp: abilitazione di diverse transizioni, ma solo una scatta;
 - SFC: mancanza della mutua esclusività nelle strutture OR-DIVERGENZA oppure esecuzione «in parallelo» di più modelli SFC (tipico della modellazione «a risorse»)
- ... viene necessariamente risolta secondo lo specifico ordine in cui vengono implementate le istruzioni LAD.