Specifica – parte IIB

Leggere Sez. 5.5.4, 5.5.4.1 Ghezzi et al.

Reti di Petri

- Adatte alla specifica di sistemi concorrenti:
  - Formalismo rigoroso e non ambiguo
  - Rappresentazione grafica relativamente semplice e intuitiva.

Reti di Petri

- Estensione degli automi a stati finiti, ma
- stato e transizione di stato sono concetti distribuiti:
  - Lo stato è dato da più stati parziali e indipendenti
  - Una transizione modifica solo una parte dello stato globale del sistema
  - Transizioni indipendenti possono avere luogo contemporaneamente

Reti di Petri

- Adatte alla rappresentazione di sistemi asincroni (in cui gli eventi non accadono a frequenze prestabilite)
- Nella versione più semplice, definiscono solo un ordinamento parziale fra eventi, astraendo dalla nozione di tempo.

Definizione: rete

- Una rete N è una quadrupla

$$N = (P,T,F,W)$$

dove

  - P è detto *insieme dei posti*,
  - T è detto *insieme delle transizioni*
  - F è detta *relazione del flusso*
  - W è detta *funzione peso*

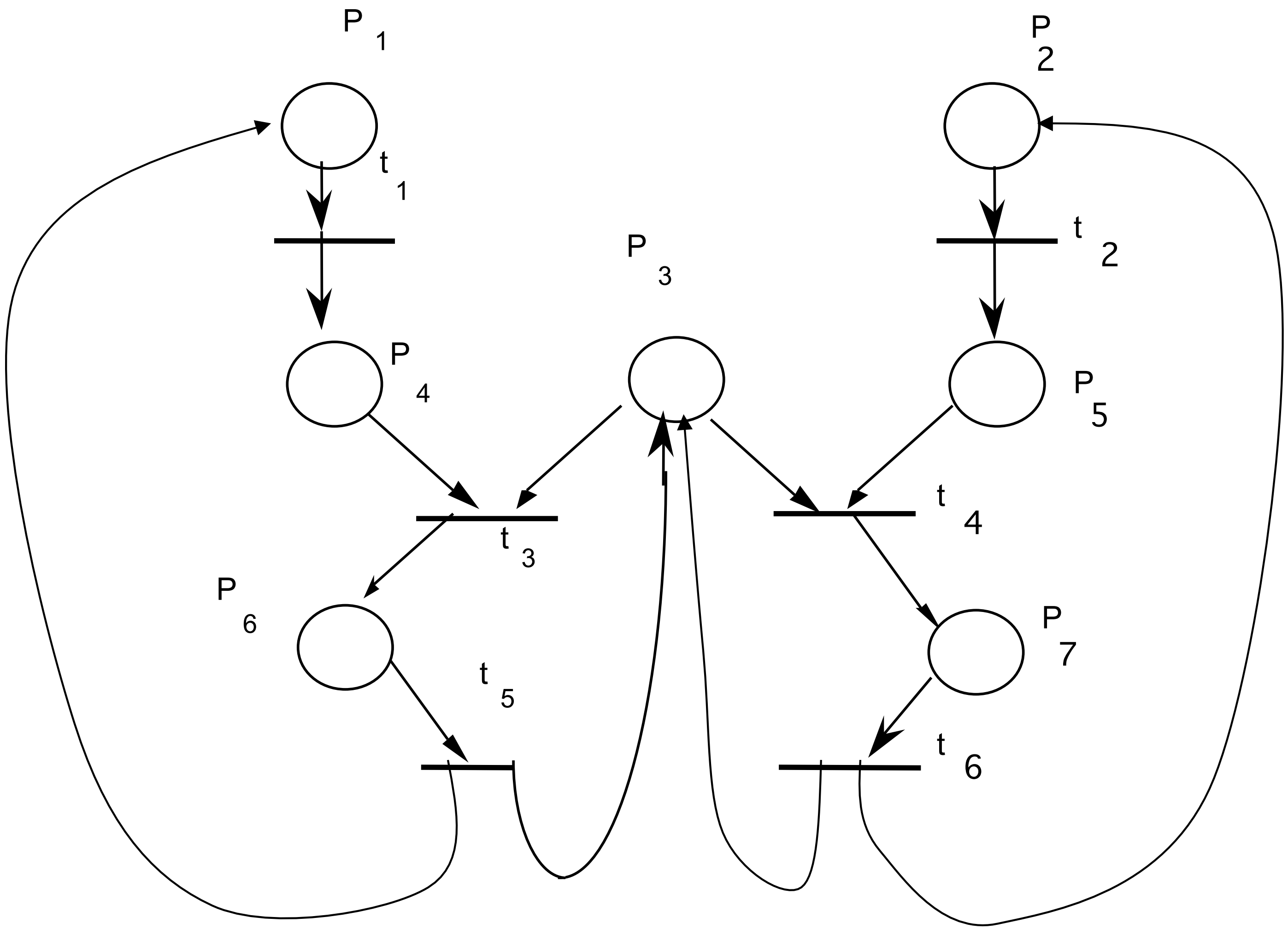
Rappresentazione grafica

- ○ Posti
- — Transizioni
- $\longrightarrow$ Flusso: freccia da A e B se e solo se $(A,B) \in F$
- $\longleftrightarrow$ Può sostituire due frecce

Definizione: rete

- Condizioni:
  - $P \cup T \neq \emptyset$
  - $P \cap T = \emptyset$
  - $F \subseteq (P \times T) \cup (T \times P)$
  - $W: F \to N \setminus \{0\}$ (valore di default 1)
- Marcatura (rappresenta lo stato):
  - $M: P \to N$, assegna a ogni posto un numero naturale di *token* (o *gettoni*).

Esempio



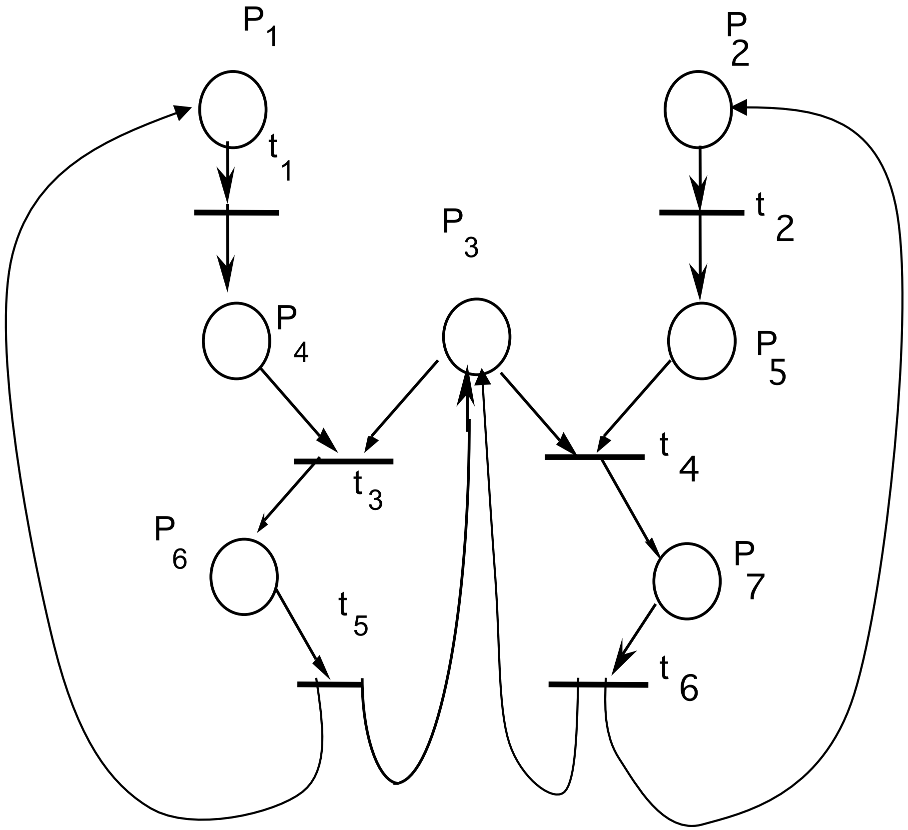
Definizioni

- Un posto p è un *posto di input* di una transizione t se e solo se $(p,t) \in F$ (cioè se una freccia va da p a t).

  $Pre(t) = \{p \mid (p,t) \in F\}$
- Un posto p è un *posto di output* di una transizione t se e solo se $(t,p) \in F$ (cioè se una freccia va da t a p).

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

Esempio

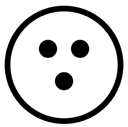


- $Pre(t_3) = \{p_3,p_4\}$
- $Post(t_3) = \{p_6\}$
- $Pre(t_5) = \{p_6\}$
- $Post(t_5) = \{p_1,p_3\}$

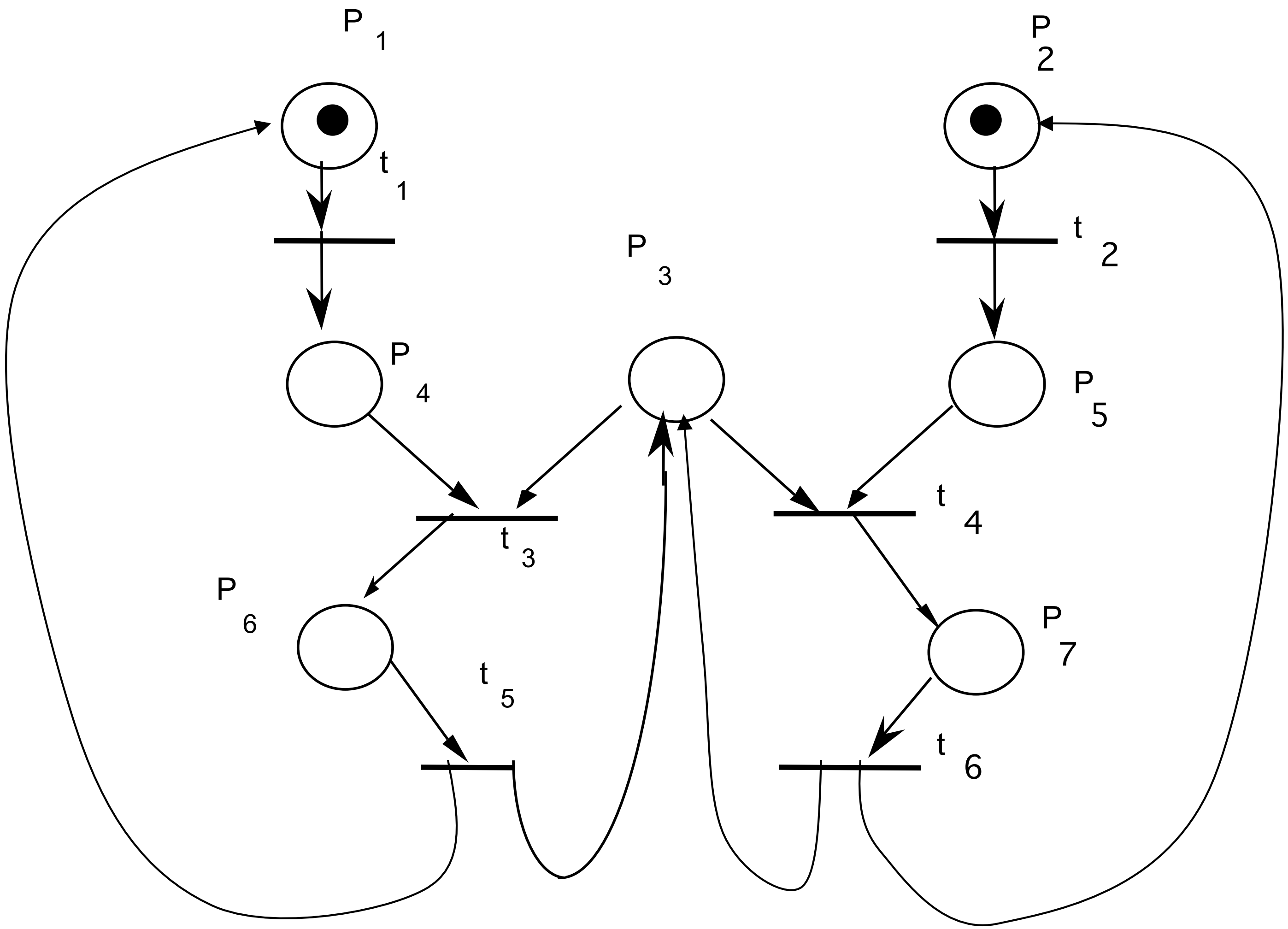
Marcatura iniziale

- Per stabilire l’evoluzione di una Rete di Petri è necessario fornire una *marcatura iniziale* M_0
- M_0 rappresenta l’insieme degli stati parziali, ossia lo stato globale, in cui la rete si trova all’inizio della sua evoluzione

Rappresentazione grafica

- ○ Posti
- — Transizioni
- → Flusso: freccia da A e B se e solo se (A,B) $\in$ F
- ↔ Può sostituire due frecce
-  Marcatura (M(p) = 3)

Esempio



Transizione abilitata

- Una transizione *t* è *abilitata* se e solo se ognuno dei suoi posti di input è marcato con un numero di token maggiore o uguale al peso dell'elemento di flusso che lo unisce a *t*, cioè se e solo se

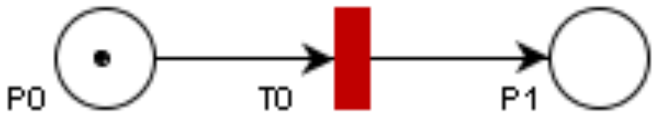
$$\forall\ p \in Pre(t)\ \ M(p) \geq W((p,t))$$

Scatto di una transizione

- Una transizione abilitata può *scattare*.
- Lo scatto di una transizione *t*
  - rimuove da ognuno dei suoi posti *p* di input un numero di token pari al peso dell'elemento di flusso che unisce *p* a *t*
  - Aggiunge a ognuno dei suoi posti *q* di output un numero di token pari al peso dell'elemento di flusso che unisce *t* a *q*.
- Se più transizioni sono abilitate, può scattare una qualunque di esse (*non determinismo*).

Esempio

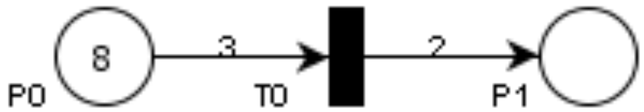
Marcatura iniziale



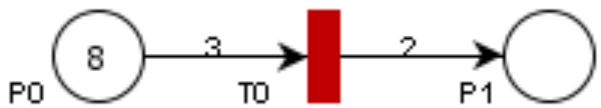
T0 è abilitata

T0 è scattata

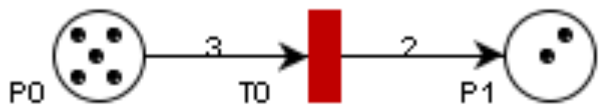
Esempio



Marcatura iniziale

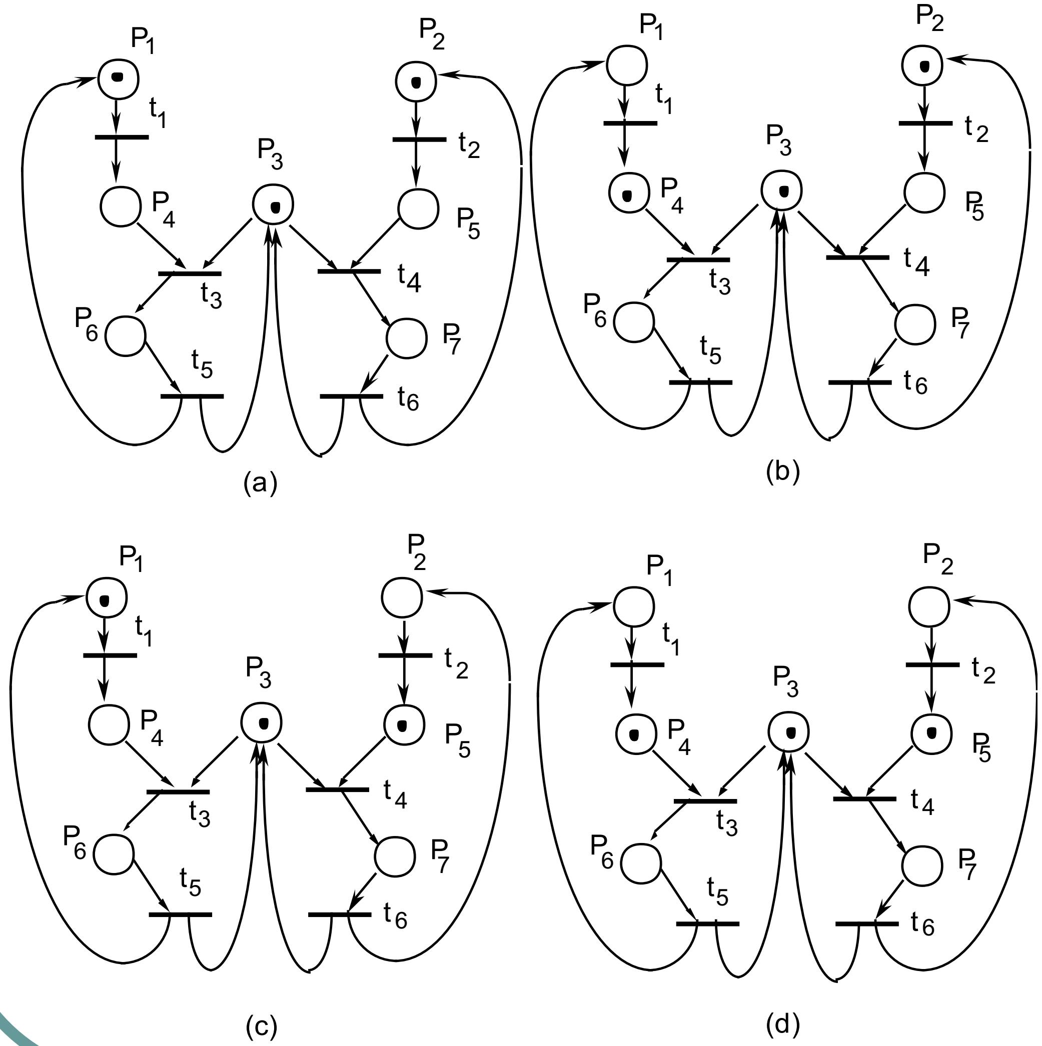


T0 è abilitata



T0 è scattata

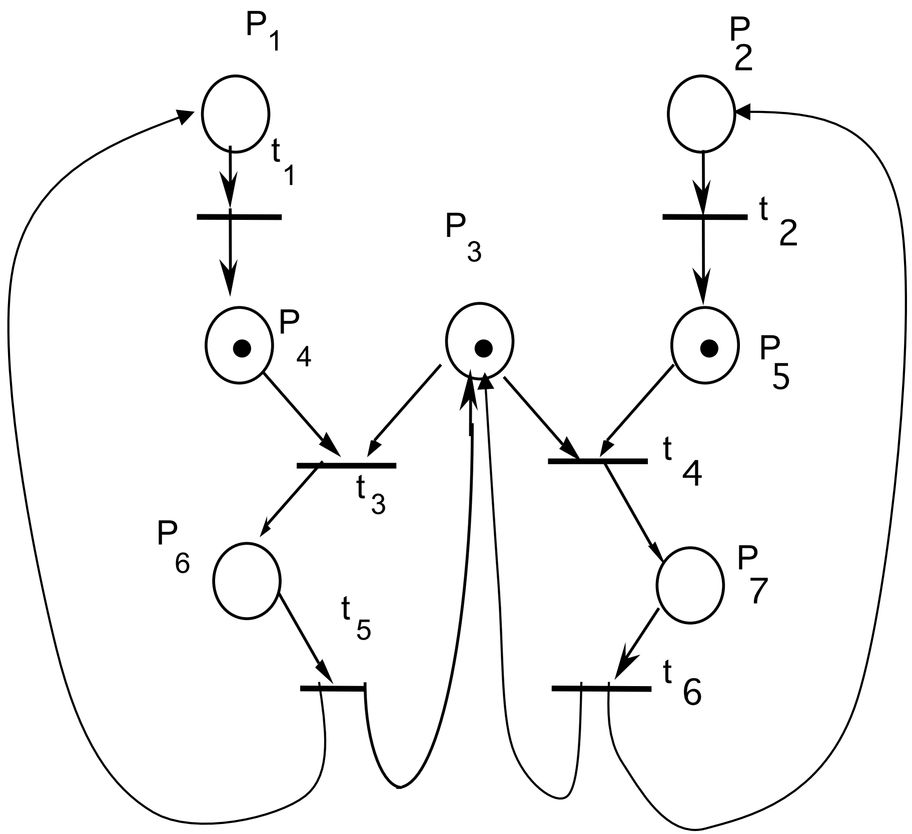
Esempio



- Dalla marcatura iniziale (a) si può passare a:
  - (b) se scatta t_1
  - (c) se scatta t_2

  (una non esclude l'altra)
- Da (b) o da (c) si può passare a (d) o ad altri stati, es. se da (b) scatta t_3.

Esempio

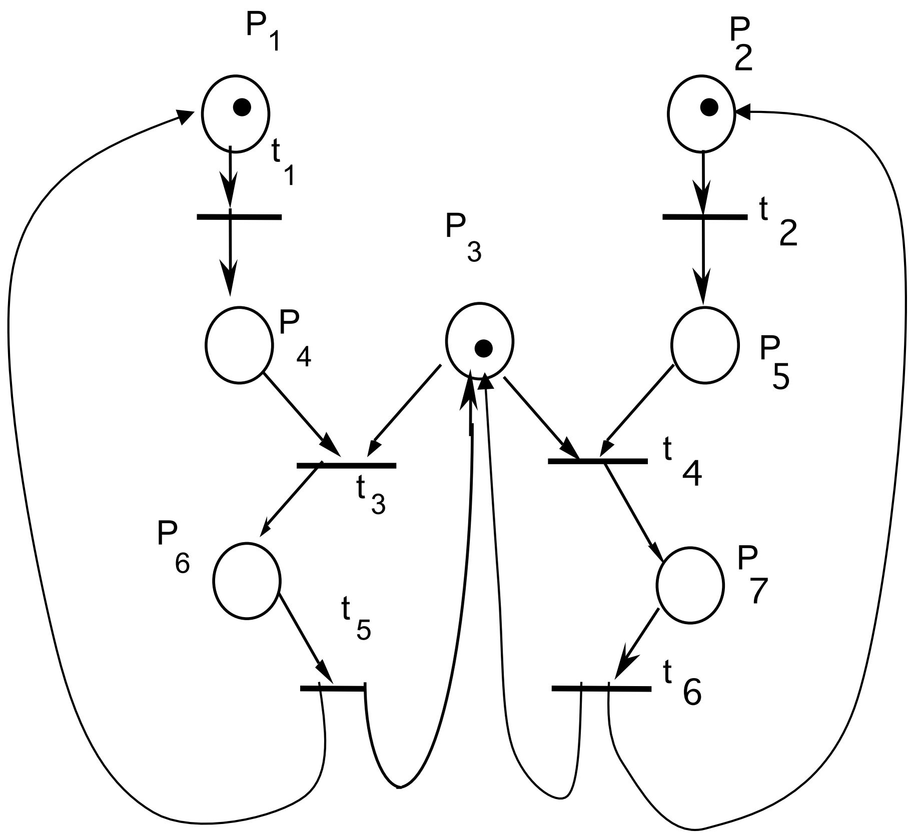


- In questo stato, t_3 e t_4 sono mutuamente esclusive (ognuna delle due, scattando, disabilita l'altra)
- Conflitto strutturale (un posto $p \in Pre(t_i)$ e $p \in Pre(t_j)$)
- Conflitto effettivo, marcatura insufficiente a far scattare sia t_i sia t_j

Sequenza di scatti

- Definizione: data una certa marcatura di una rete, una *sequenza di scatti* $t_1 t_2 \ldots t_N$ è abilitata se:
  - $N=1$ e t_1 è abilitata, oppure
  - $N>1$, t_1 è abilitata e, data la marcatura risultante dallo scatto di t_1, $t_2 \ldots$, t_{N-1} si ha che t_N è abilitata

Esempio sequenza di scatti



- Sono abilitate in questa marcatura le sequenze di scatti:
  - $t_1\ t_2\ t_3\ t_5$
  - $t_1\ t_2\ t_4\ t_6$

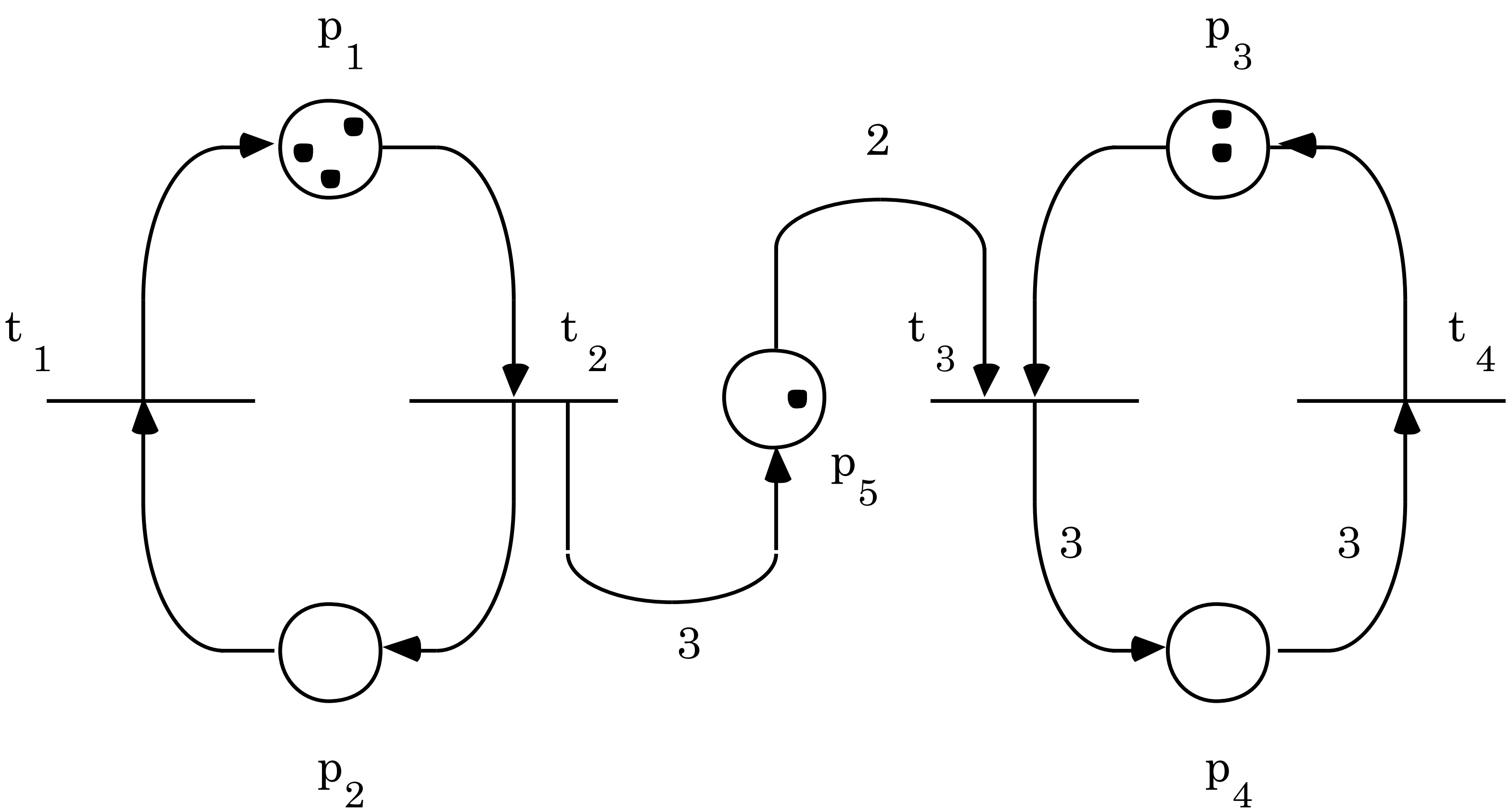
Entrambe riportano la rete nella marcatura in figura

Editor/simulatore di reti di Petri

- Platform Independent Petri Net Editor (PIPE)
- http://pipe2.sourceforge.net

- Petri Net World
- http://www.informatik.uni-hamburg.de/TGI/PetriNets/
- Da cui sono linkati tools e applet java

Rappresentazione matriciale



Matrici I e O, una riga per ogni posto, una colonna per ogni transizione

Rappresentazione matriciale

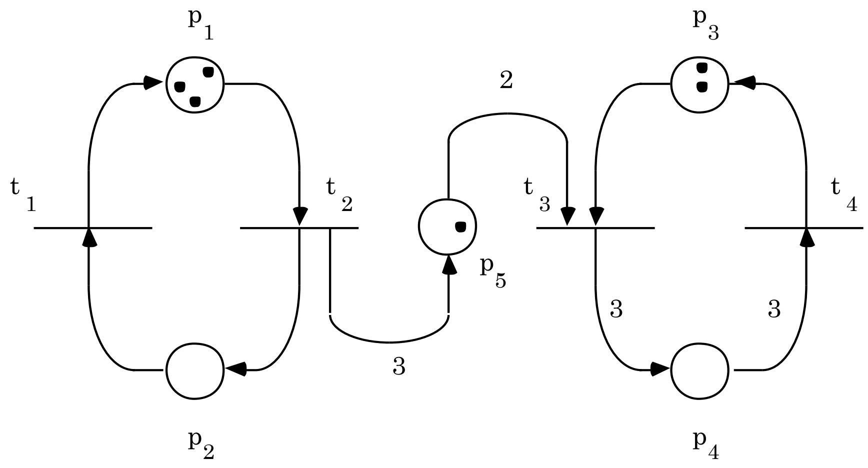
$$O = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 3 & 0 \\ 0 & 3 & 0 & 0 \end{vmatrix} \qquad I = \begin{vmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 3 \\ 0 & 0 & 2 & 0 \end{vmatrix}$$

$$C = O - I = \begin{vmatrix} 1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 3 & -3 \\ 0 & 3 & -2 & 0 \end{vmatrix}$$

$$m = \begin{vmatrix} 3 \\ 0 \\ 2 \\ 0 \\ 1 \end{vmatrix} \qquad \text{Scatta t2:} \qquad m' = \begin{vmatrix} 2 \\ 1 \\ 2 \\ 0 \\ 4 \end{vmatrix}$$

Abilitazione di una transizione

- L'abilitazione di una transizione è determinata in base alla matrice I (matrice degli ingressi) e al vettore marcatura m
- Se $m>=I_j$ è abilitata t_j
- Nell'esempio t_2 è l'unica transizione abilitata



Rappresentazione matriciale

$$O = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 3 & 0 \\ 0 & 3 & 0 & 0 \end{vmatrix} \qquad I = \begin{vmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 3 \\ 0 & 0 & 2 & 0 \end{vmatrix}$$

$$C = O - I = \begin{vmatrix} 1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 3 & -3 \\ 0 & 3 & -2 & 0 \end{vmatrix}$$

$$m = \begin{vmatrix} 3 \\ 0 \\ 2 \\ 0 \\ 1 \end{vmatrix} \qquad \text{Scatta t2:} \qquad m' = \begin{vmatrix} 2 \\ 1 \\ 2 \\ 0 \\ 4 \end{vmatrix}$$

Rappresentazione matriciale

- E’ possibile stabile l’abilitazione di una transizione in base alla sola matrice C (matrice di incidenza) nelle cosiddette ***reti pure***
- Una rete pura è una rete P/T in cui

$$\forall t \in T, Pre(t) \cap Post(t) = \emptyset$$

(si esclude che un posto sia d'ingresso e di uscita a una medesima transizione)

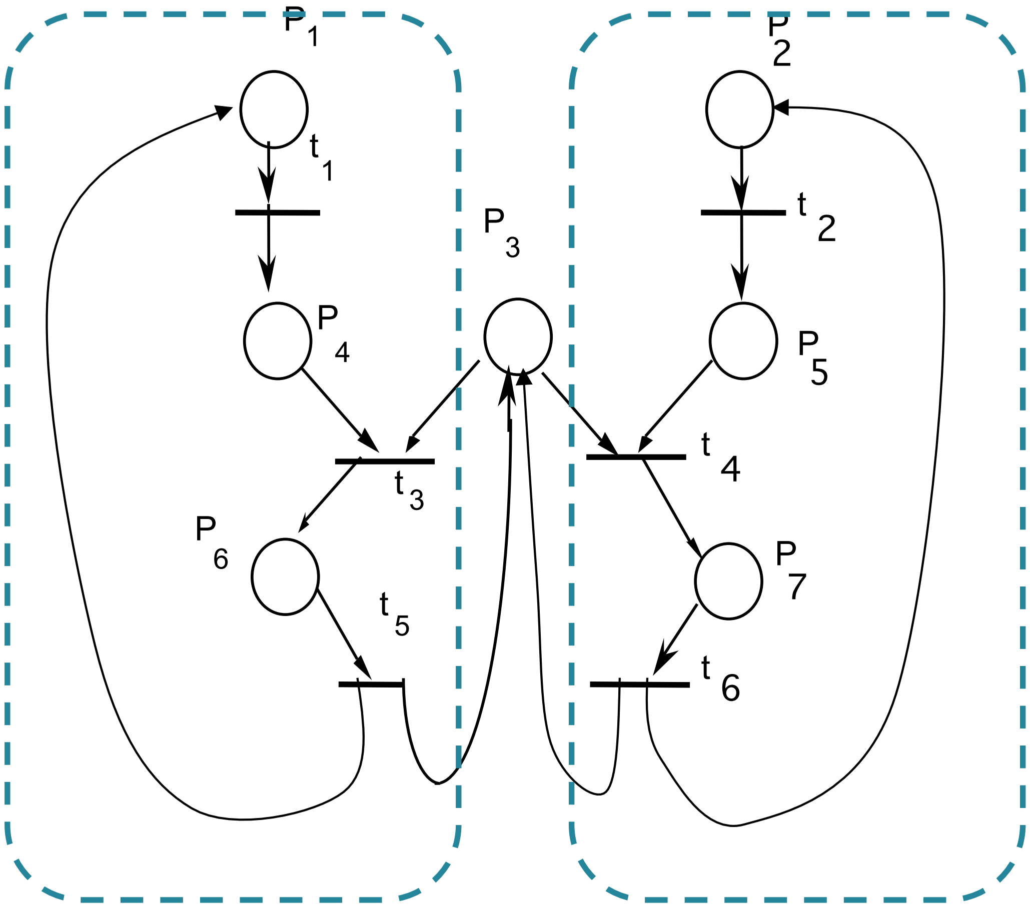
- Data una sequenza di scatti ammissibile, per queste reti:

$$m' = m + C\,s$$

Modellazione di sistemi concorrenti con reti di Petri

- Le marcature rappresentano lo stato in modo distribuito.
- Le transizioni rappresentano eventi.
- L'abilitazione di una transizione rappresenta il rispetto delle condizioni necessarie affinché l'evento corrispondente avvenga.

Esempio



- Due serie di attività indipendenti (t_1, t_3, t_5 e t_2, t_4, t_6)
- Accedono in modo mutuamente esclusivo a una risorsa comune (p_3).

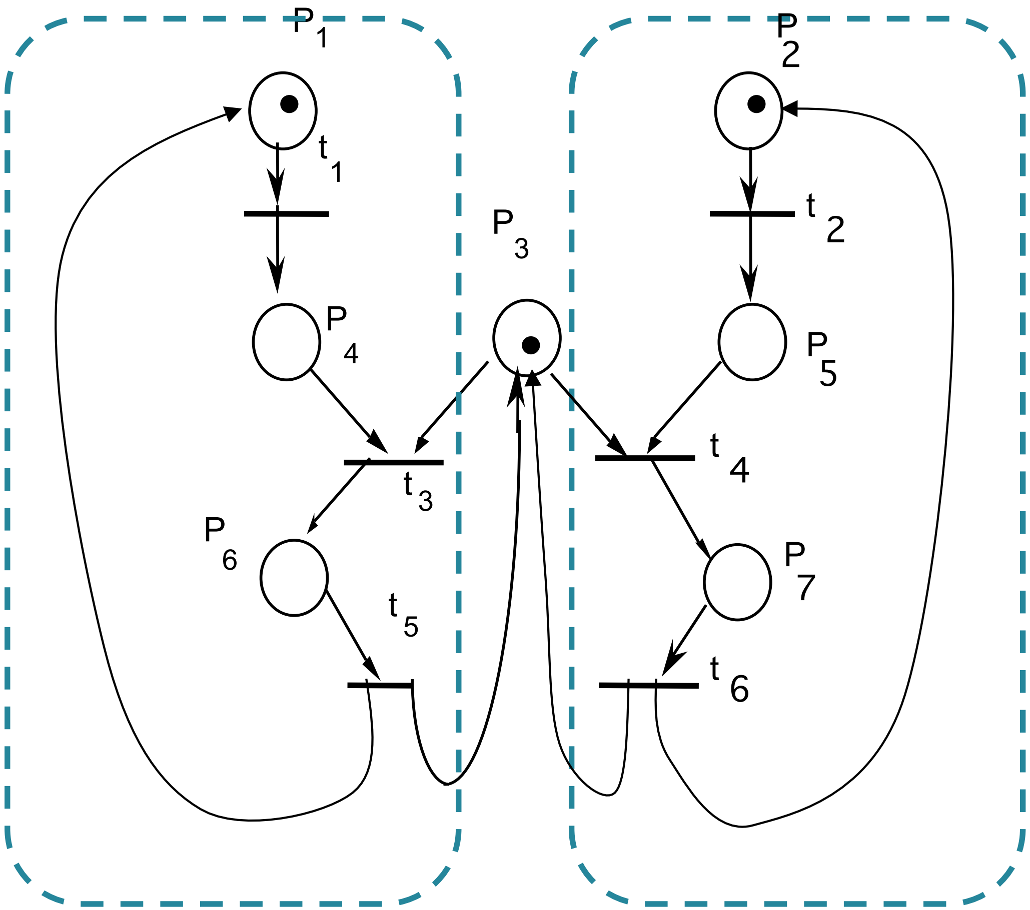
Concorrenza

- Le transizioni abilitate possono scattare indipendentemente l'una dall'altra
- Condizioni comuni:
  - *Concorrenza*: due transizioni sono abilitate, e lo scatto di una non disabilita l'altra.
  - *Conflitto*: due transizioni sono abilitate, ma lo scatto di una disabilita l'altra (esempio: accesso a una risorsa condivisa).

Conflitto

- Le reti di Petri non prevedono meccanismi per risolvere automaticamente i conflitti
- Di due transizioni in conflitto, può scattare sempre la stessa
- L'attività che necessita dell'altra transizione è soggetta a *starvation*.

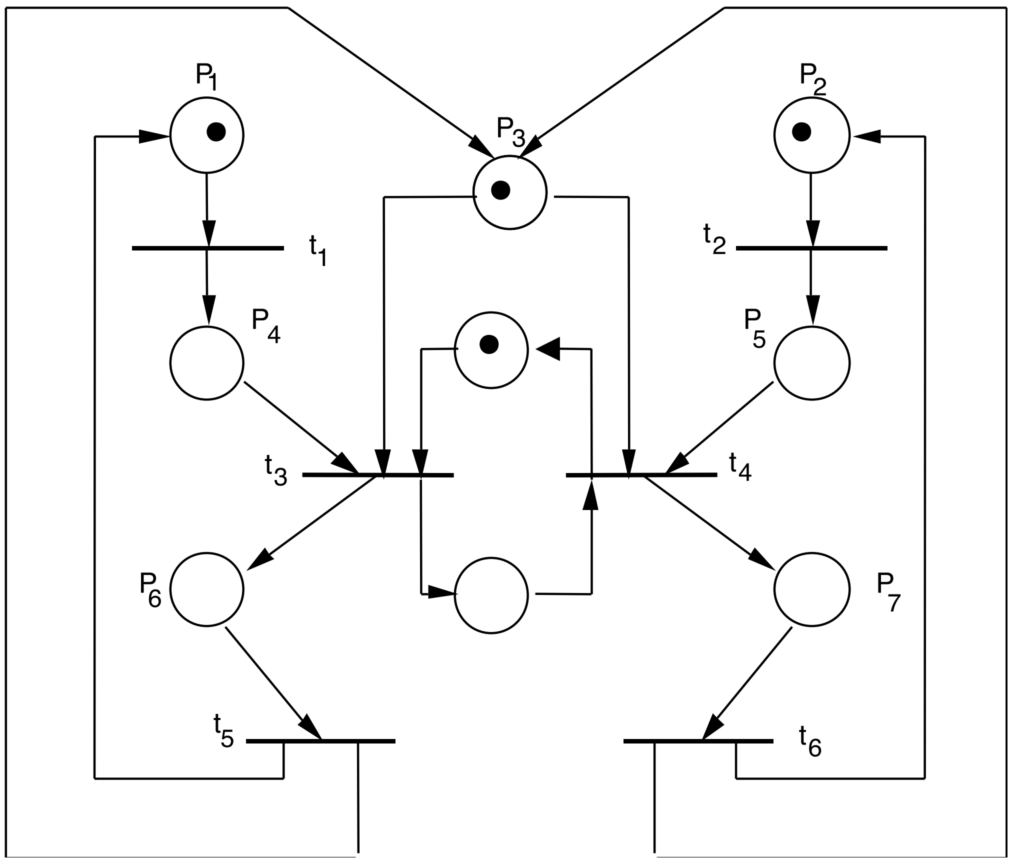
Esempio



- Due serie di attività indipendenti (t_1, t_3, t_5 e t_2, t_4, t_6)
- Può scattare sempre e solo la prima sequenza (*starvation*)

Per evitare la starvation

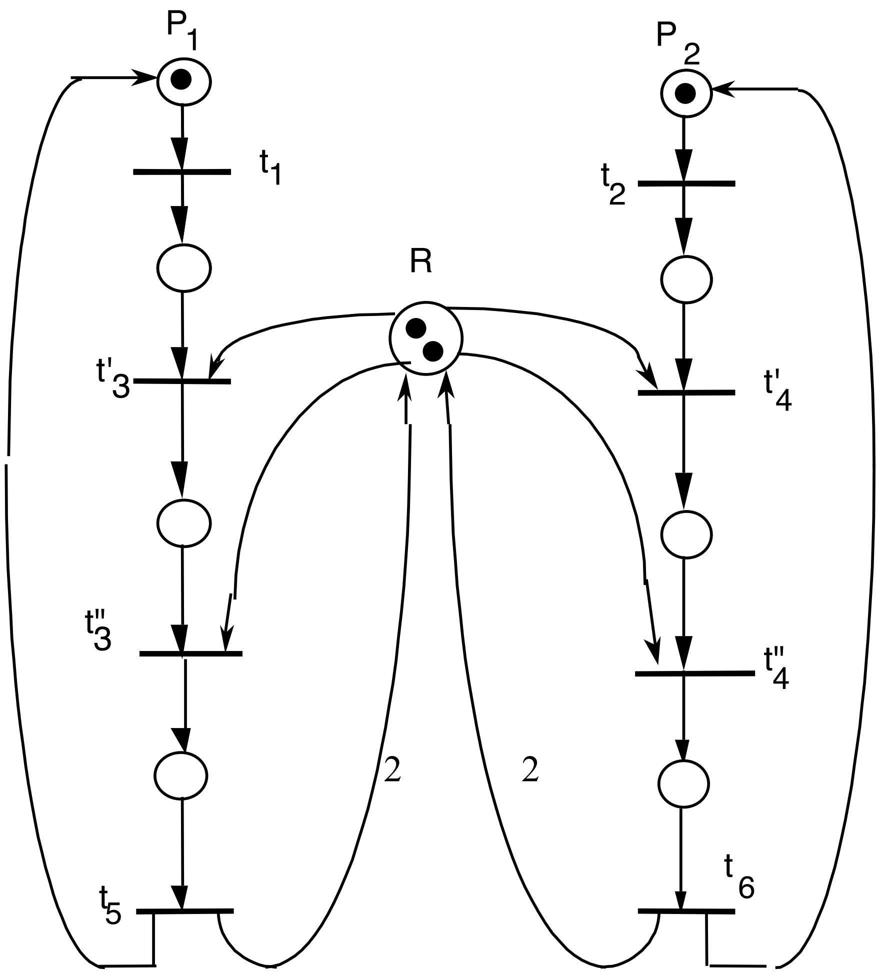
- Si può imporre l'alternanza:



Deadlock

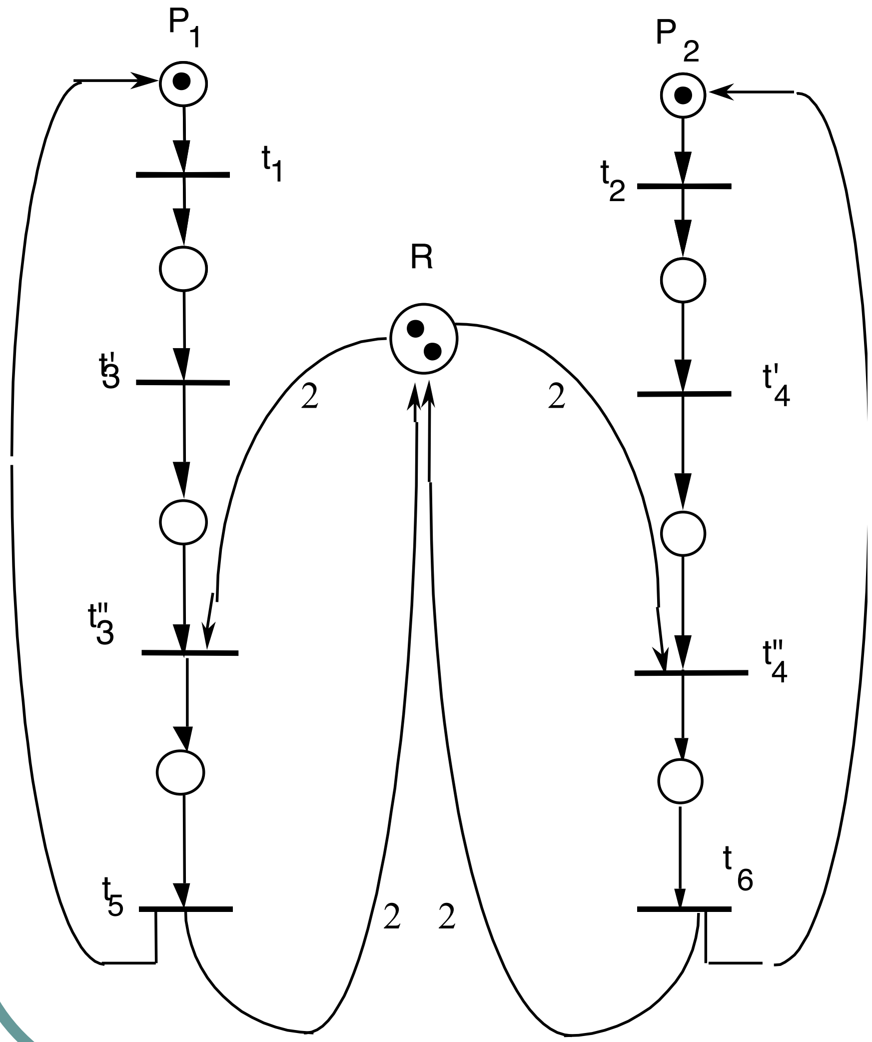
- Un sistema è in *deadlock* se è in uno stato dal quale è impossibilitato a evolvere
- Condizione da evitare, ma difficile da prevedere
- L'analisi delle reti di Petri permette di individuare i *deadlock* di un sistema.

Esempio



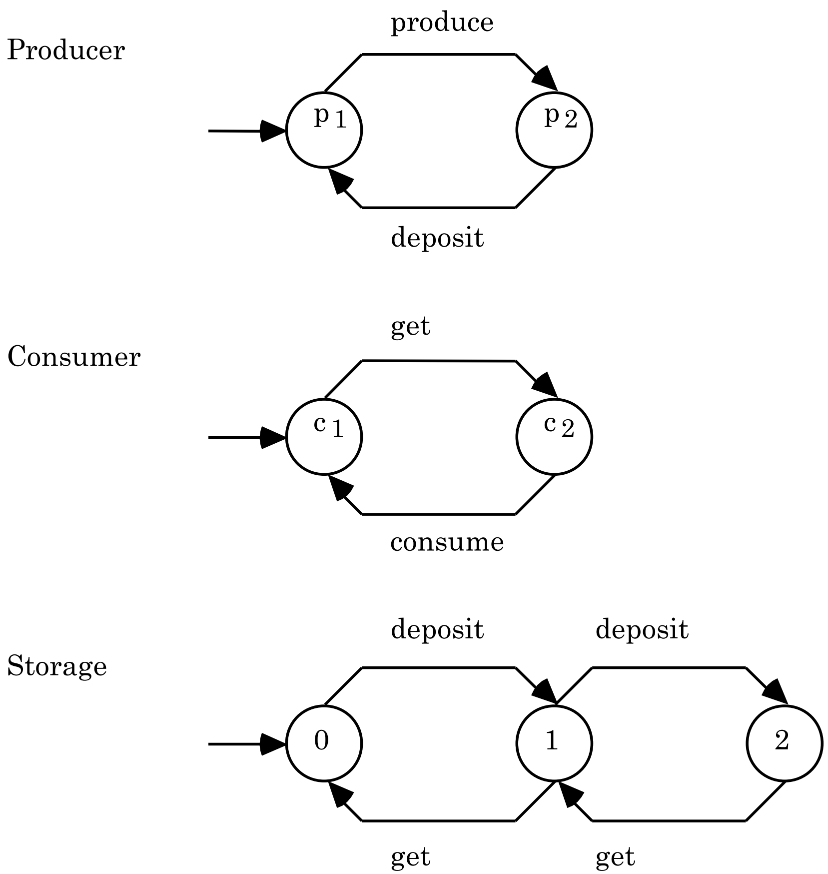
- Attività concorrenti che necessitano di due copie di una risorsa.
- Caso "normale": t_1, t_3', t_3'', t_5
- Caso che porta a deadlock: t_1, t_3', t_2, t_4'

Per evitare il deadlock



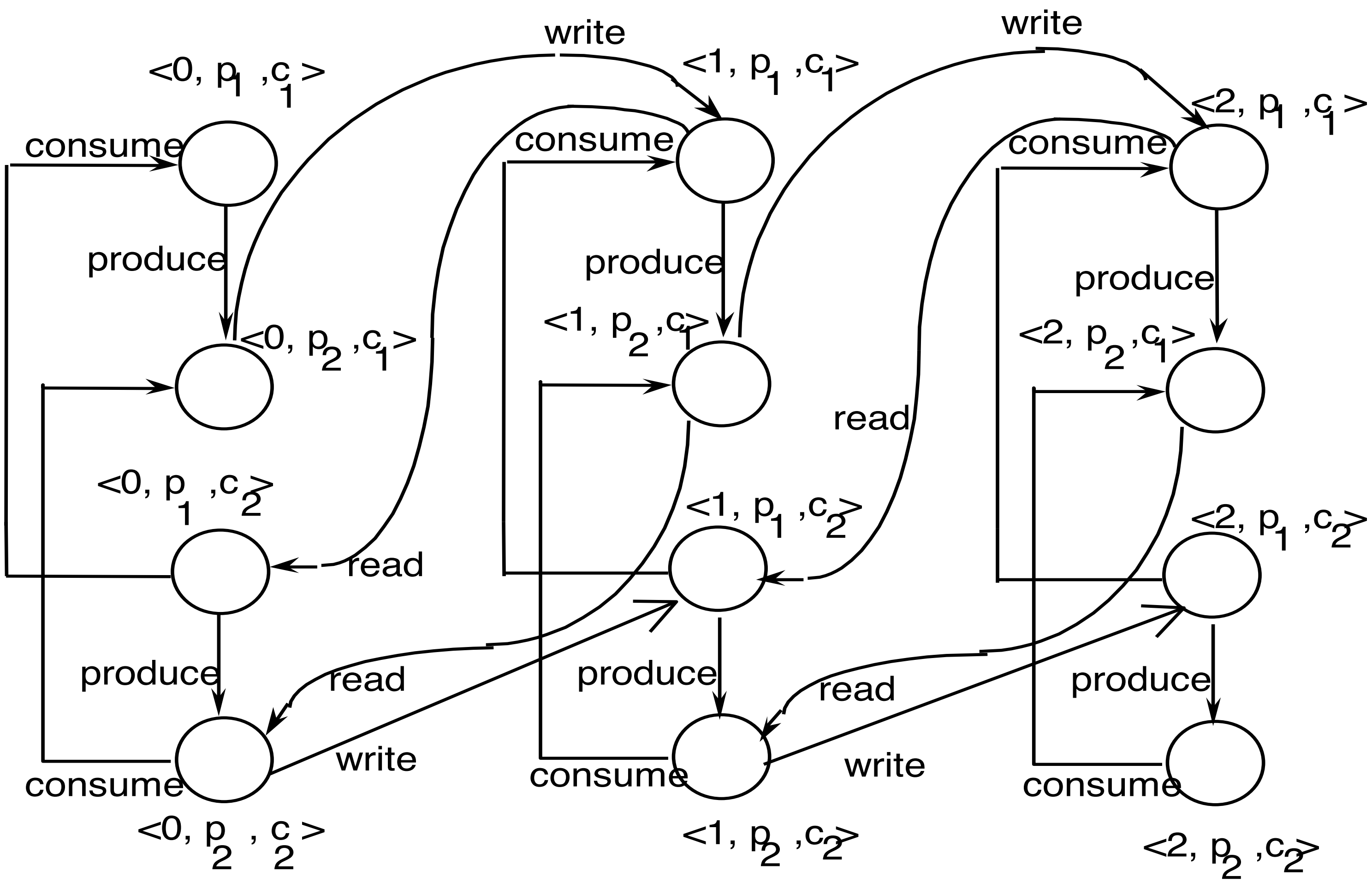
- Le due copie della risorsa vengono prelevate con una sola transizione
- In questa rete non è però risolto il problema della *starvation*

FSA: produttore-consumatore

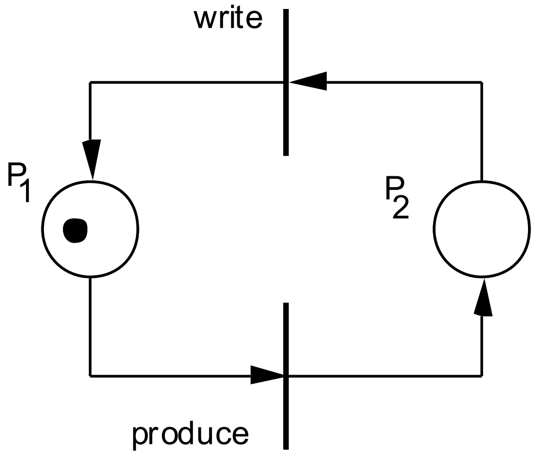


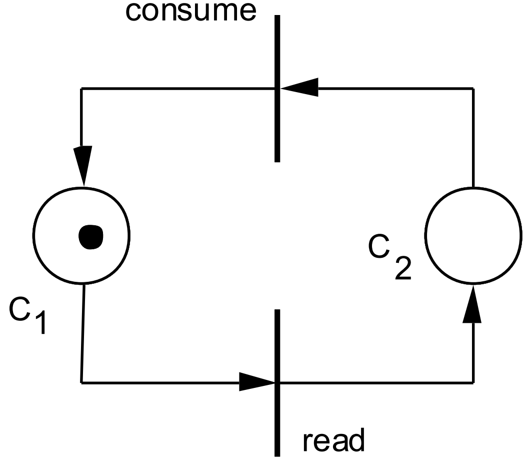
- Deposito con due spazi
- Per modellare l'interazione occorre disegnare un automa unico.
- Lo spazio degli stati è $Q_p \times Q_c \times Q_s$

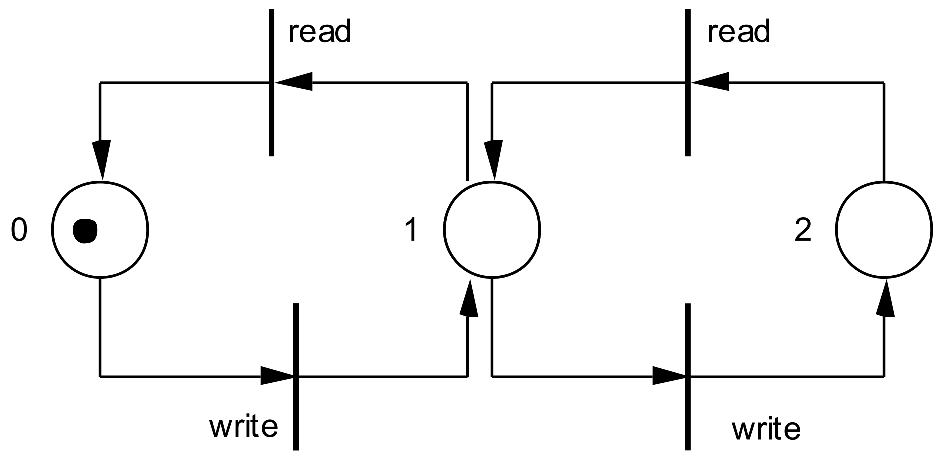
Automa complessivo (DSA)



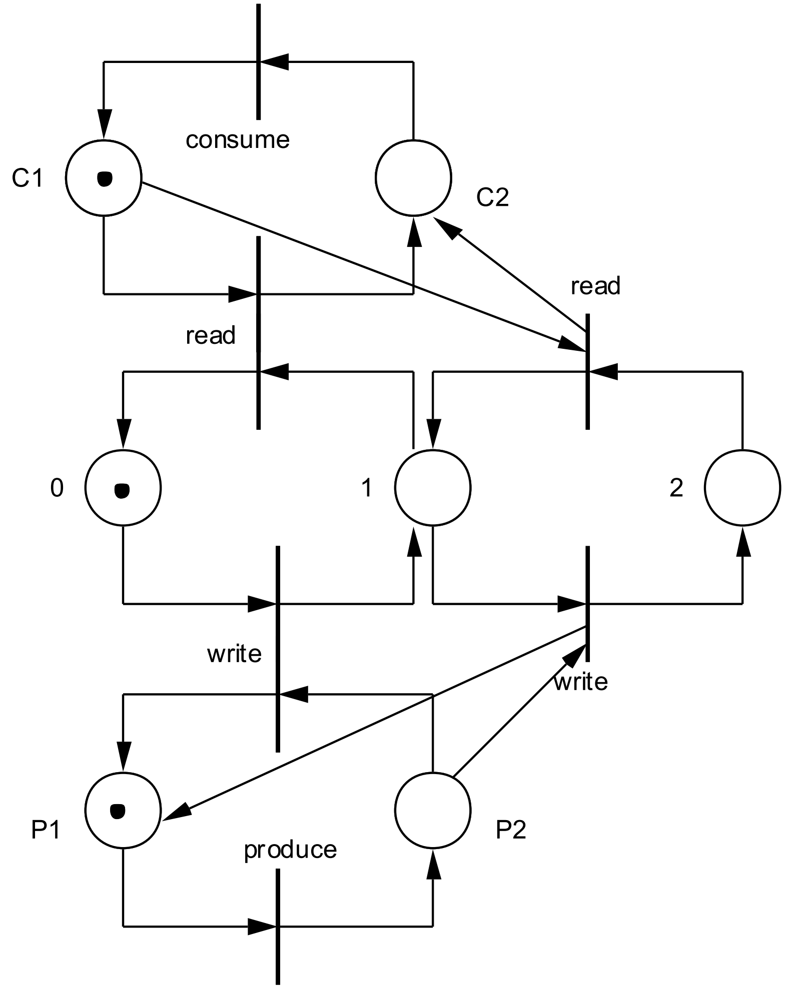
Esempio produttore / consumatore: reti separate







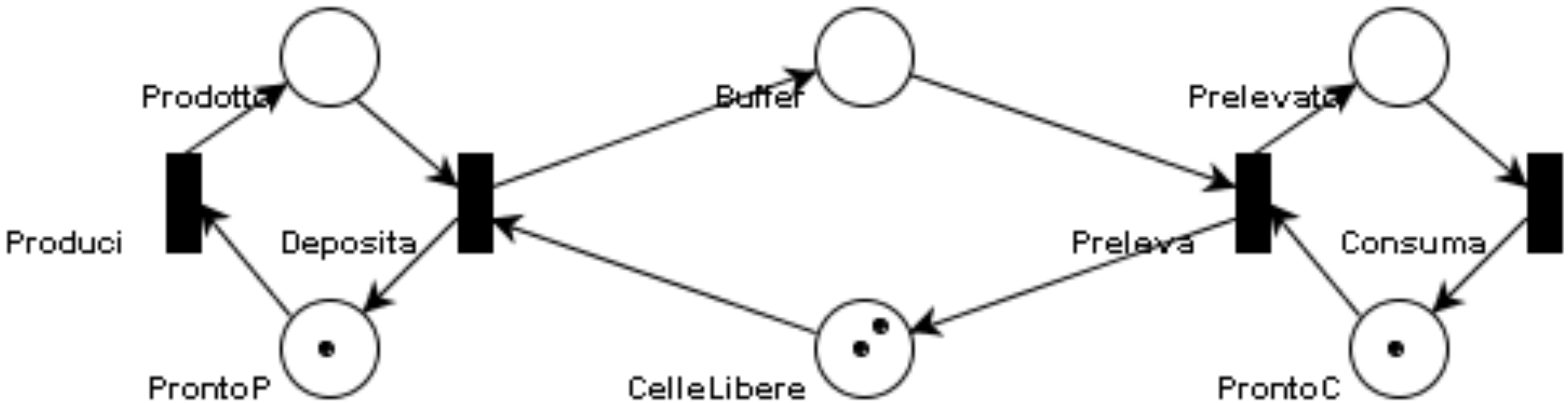
Esempio produttore / consumatore: rete unica



Posti a capacità limitata

- E' possibile stabilire la capacità massima di un posto, cioè il numero massimo di token che può contenere.
- Non risultano abilitate le transizioni che porterebbero un posto a superare la sua capacità massima.
- Si può simulare questo comportamento con un "posto complementare".

Esempio



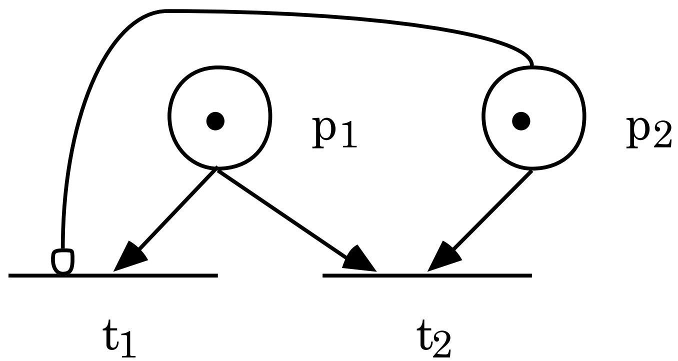
Produttore

Buffer a capacità 2

Consumatore

Archi inibitori

- La presenza di un token in un posto di input disabilita la transizione



- t_1 è disabilitata, t_2 è abilitata

Proprietà di reti di Petri: raggiungibilità

- Una marcatura M' è raggiungibile da una marcatura M se esiste una sequenza di scatti che a partire da M produca M'
- Serve a verificare se si possono raggiungere stati indesiderati (ad esempio situazioni di *deadlock*)

Proprietà di reti di Petri: limitatezza

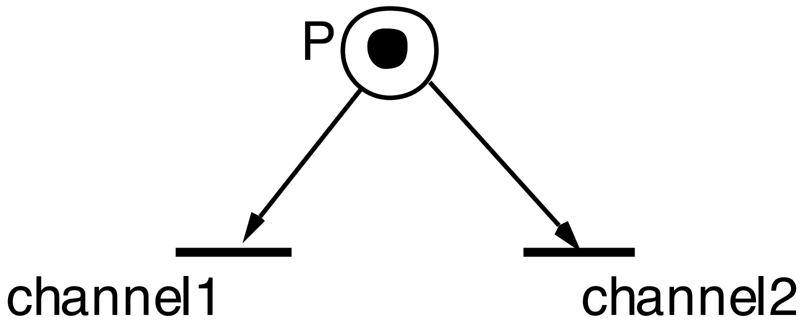
- Un posto si dice k-limitato se in qualunque marcatura raggiungibile non ha più di k token
- Una rete si dice limitata se tutti i suoi posti sono k-limitati per qualche k
- Una rete limitata è equivalente a un automa a stati finiti
- Qualsiasi rete può essere resa limitata con l'aggiunta di posti opportuni

Proprietà di reti di Petri: liveness

- Una transizione t è *live* se per ogni marcatura M raggiungibile, esiste una marcatura M' raggiungibile da M in cui t è abilitata
- Se almeno una transizione è *live*, non può esserci *deadlock* (la rete è *live*)
- *Liveness property*

Limitazioni

- Es: vogliamo che un messaggio sia spedito attraverso uno di due canali:
  - $channel_1$ se il messaggio è ben formato
  - $channel_2$ altrimenti



- Ma i token non hanno valore

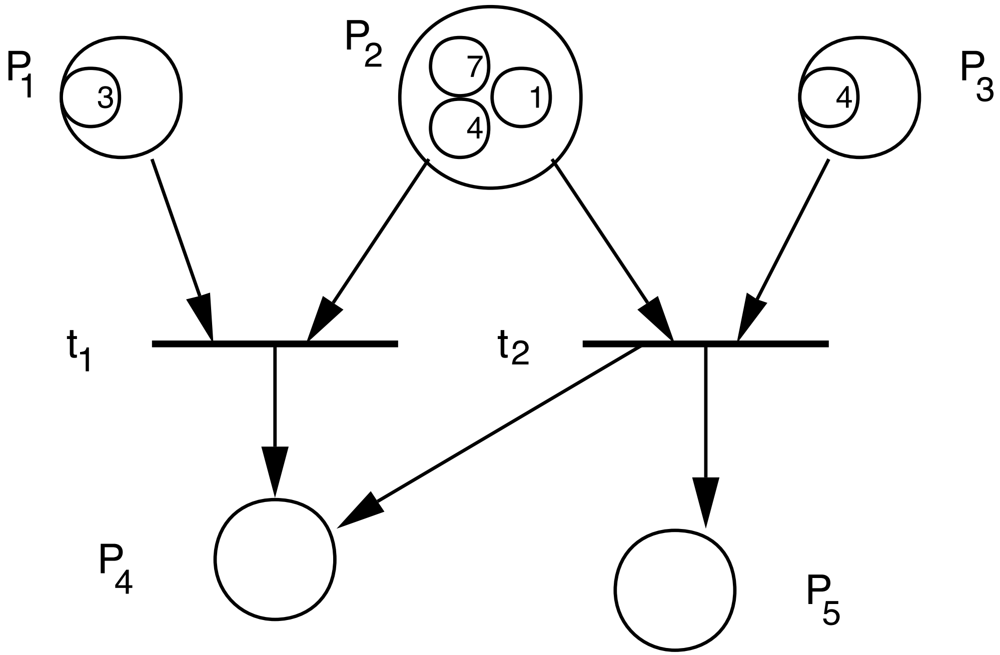
Estensione: assegnare valori ai token

- Alle transizioni sono associati predicati e funzioni
- I predicati, definiti sui valori dei token, determinano l'abilitazione delle transizioni
- Le funzioni determinano i valori dei token nei posti di output

Estensione: assegnare valori ai token

- Una transizione con N posti d'ingresso è abilitata se esistono N token, uno per ogni posto, i cui valori rispettano il predicato associato (*tupla pronta*).
- Possono esserci più tuple pronte
- La tupla pronta selezionata viene rimossa
- Si aggiungono token nei posti di output, con valori secondo le funzioni associate.

Esempio



P2 > P1
e
P4 := P2 + P1
associati con t1

P3 = P2 e
P4 := P3 – P2 e
P5 := P2 + P3
associati con t2

Lo scatto di t1 con <3,7> porta un token di valore 10 in P4. t2 allora può scattare con <4, 4> (inserisce token di valore 0 in P4 e e di valore 8 in P5)

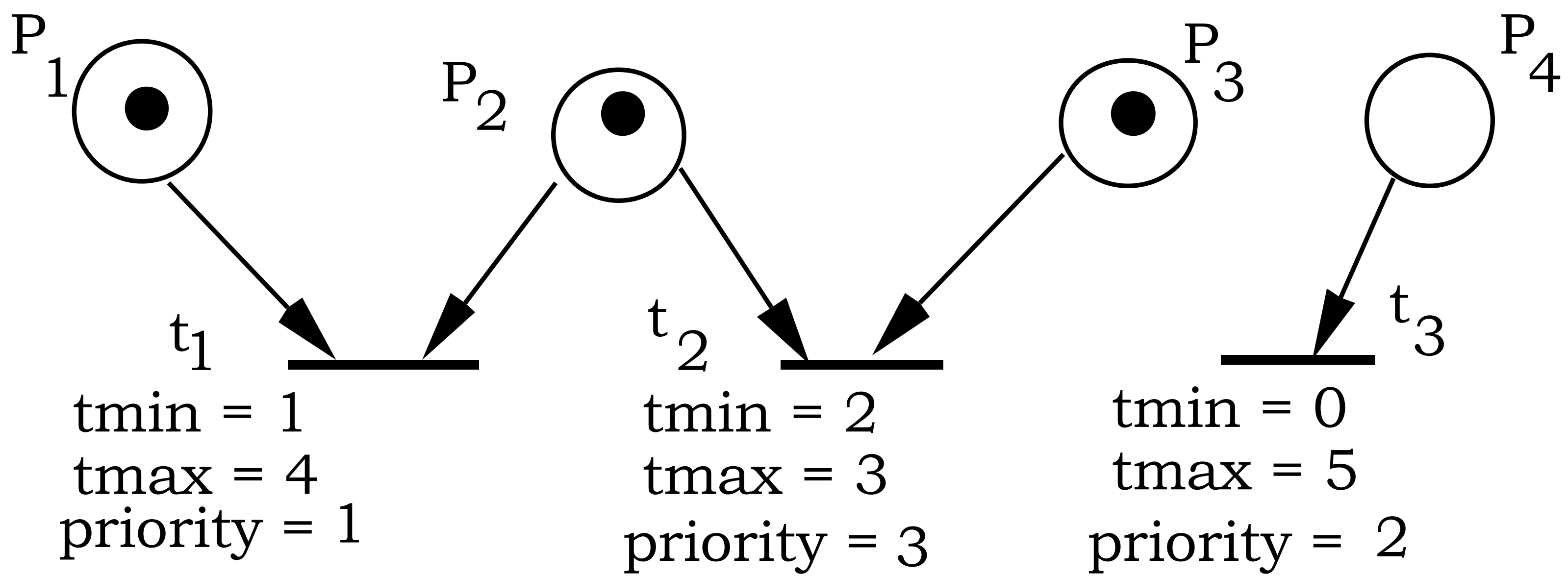
Reti con priorità

- Funzione $pri : T \rightarrow N$
- che assegna a ogni transizione un valore intero di priorità
- Fra più transizioni abilitate, scatta quella (o una di quelle) con priorità più alta
- Permette di esprimere politiche di scheduling (statiche; dinamiche se i token hanno un valore)

Reti temporizzate

- A ogni transizione associate due costanti $<t_{min}, t_{max}>$
- t_0 nella marcatura iniziale
- Una transizione abilitata deve scattare fra t_{min} e t_{max} dopo l'istante di abilitazione.
- Se fra t_{min} e t_{max} viene disabilitata prima di scattare, non scatta.

Esempio: rete temporizzata con priorità



t_1 può scattare tra 1 e 4. Se non scatta prima di 2 non può scattare perchè t_2 ha priorità più alta

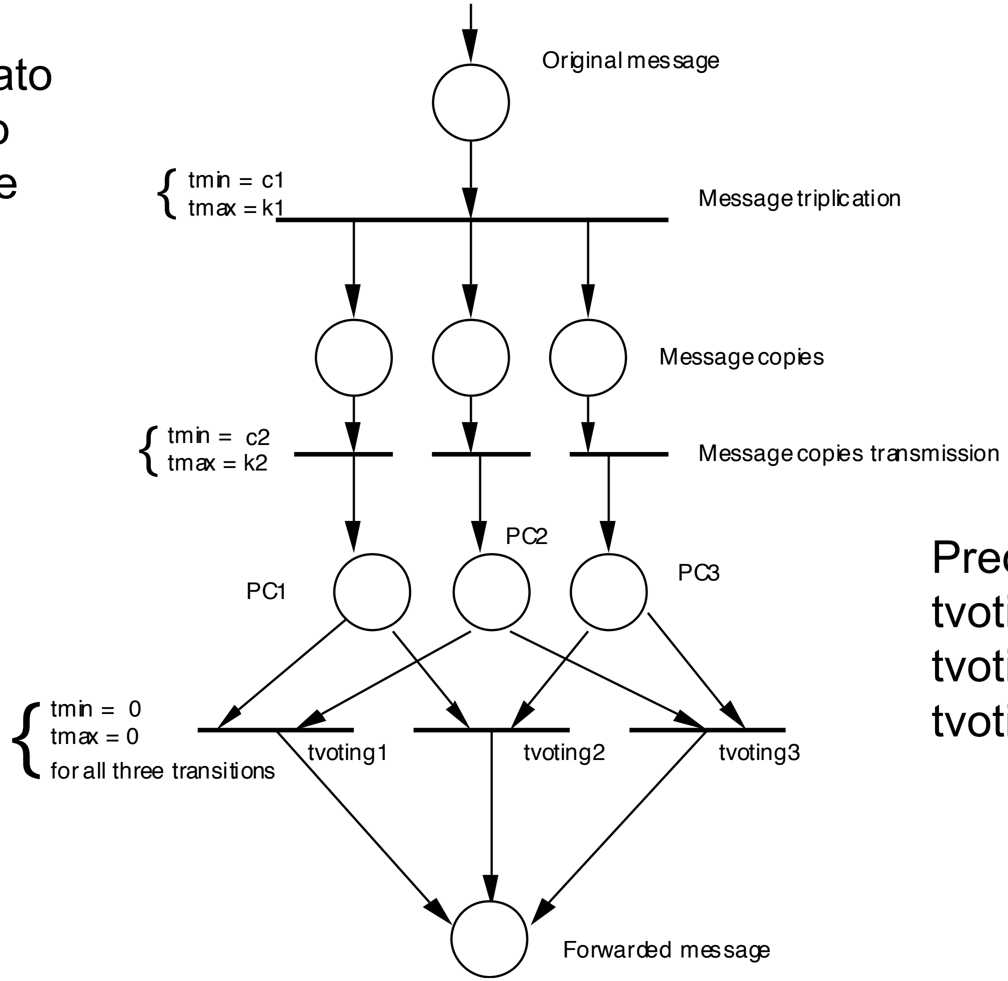
se a t=1 appare un gettone in P_4 per 1<=t<2 possono scattare t_1 e t_3 ma t_1 non può scattare prima di t_3 perché ha priorità più bassa

Esempio: replicazione e selezione di messaggi – 0

- Il messaggio dovrà essere triplicato. Le tre copie dovranno essere spedite attraverso tre canali fisici diversi. Il ricevitore dovrà accettare il messaggio sulla base di una politica di votazione "due su tre".

Esempio: replicazione e selezione di messaggi – 1

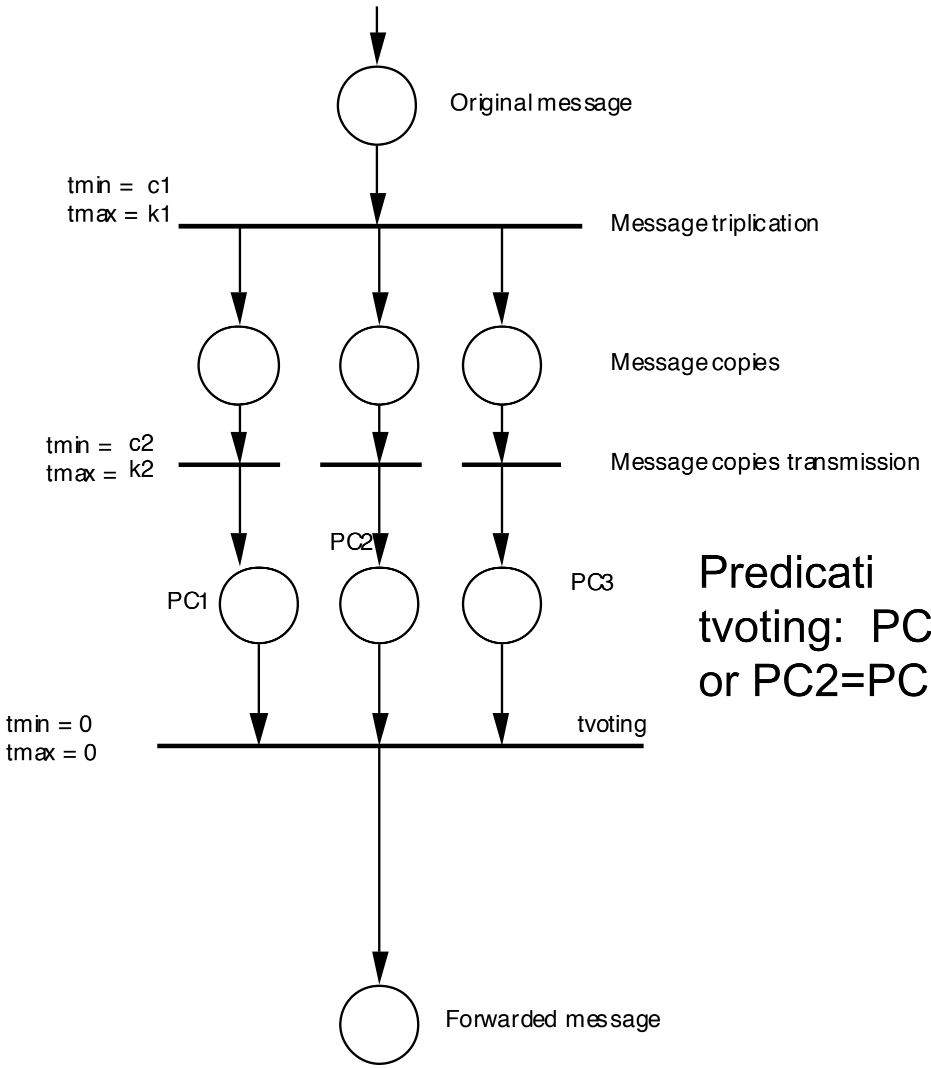
messaggio inoltrato non appena sono state ricevute due copie identiche



Predicati
tvoting1: PC1=PC2
tvoting2: PC1=PC3
tvoting3: PC2=PC3

Esempio: replicazione e selezione di messaggi – 2

messaggio inoltrato solo quando sono state ricevute le tre copie, e almeno due sono identiche

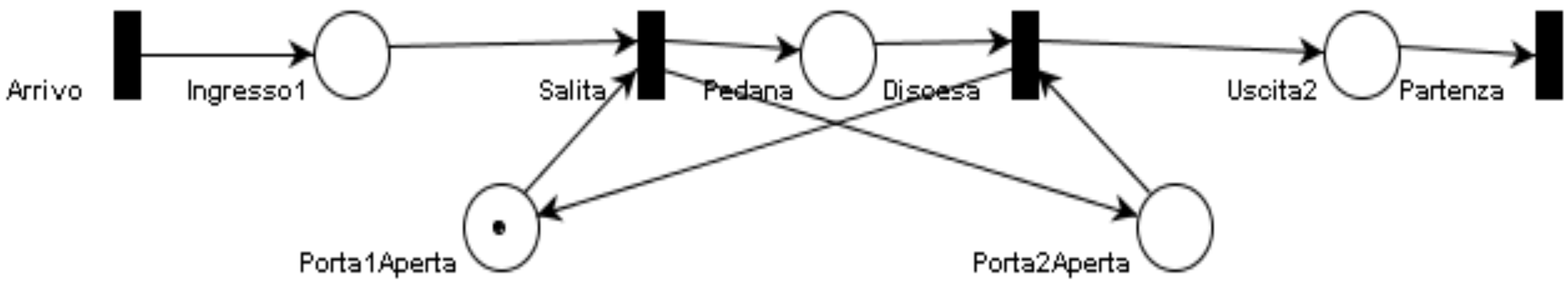


Predicati
tvoting: PC1=PC2 or PC1=PC3 or PC2=PC3

Esercizio (*da un esame …*)

- Modellare con una Rete di Petri un sistema di controllo degli accessi costituito da una pedana e due porte. La pedana si trova tra le due porte. La prima porta è solitamente aperta e la seconda chiusa. Quando la pedana conta una persona, la prima porta si chiude e si apre la seconda porta.

Soluzione

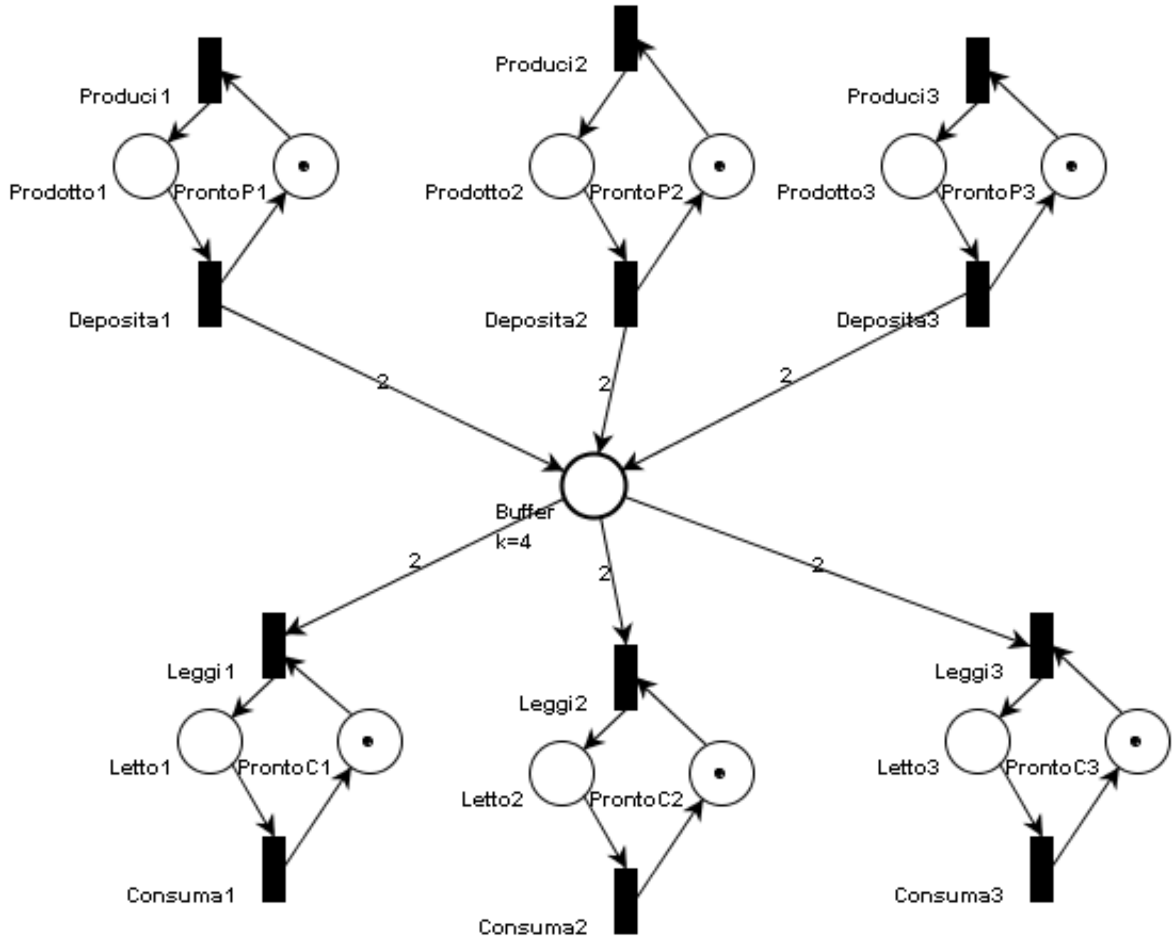


Arrivo
Ingresso1
Salita
Pedana
Discesa
Uscita2
Partenza
Porta1Aperta
Porta2Aperta

Esercizio

Modellare con una rete di Petri un sistema costituito da tre produttori, tre consumatori e un buffer di quattro celle *(realizzato come coda FIFO)*. Ogni messaggio letto o scritto è pari a due celle del buffer. I produttori scrivono sul buffer (non pieno) o producono il messaggio, i consumatori leggono (dal buffer contenente almeno un messaggio) o elaborano il messaggio letto.

Soluzione



Produci1
Produci2
Produci3
Prodotto1
ProntoP1
Prodotto2
ProntoP2
Prodotto3
ProntoP3
Deposita1
Deposita2
Deposita3
2
2
2
Buffer
k=4
2
2
2
Leggi1
Leggi2
Leggi3
Letto1
ProntoC1
Letto2
ProntoC2
Letto3
ProntoC3
Consuma1
Consuma2
Consuma3

Soluzione

- Senza imporre un limite di capacità del posto che rappresenta il buffer: