



6

Informatica

CdS in «**Scienze e Tecnologie dei Beni Culturali**» – AA 2014-2015

Mini-sito dell'insegnamento: <http://www.unife.it/scienze/beni.culturali/insegnamenti/informatica>

Prof. Giorgio Poletti

giorgio.poletti@unife.it - <http://docente.unife.it/giorgio.poletti>



UNIVERSITÀ
DEGLI STUDI
DI FERRARA
- EX LABORE FRUCTUS -



Contenuti della lezione

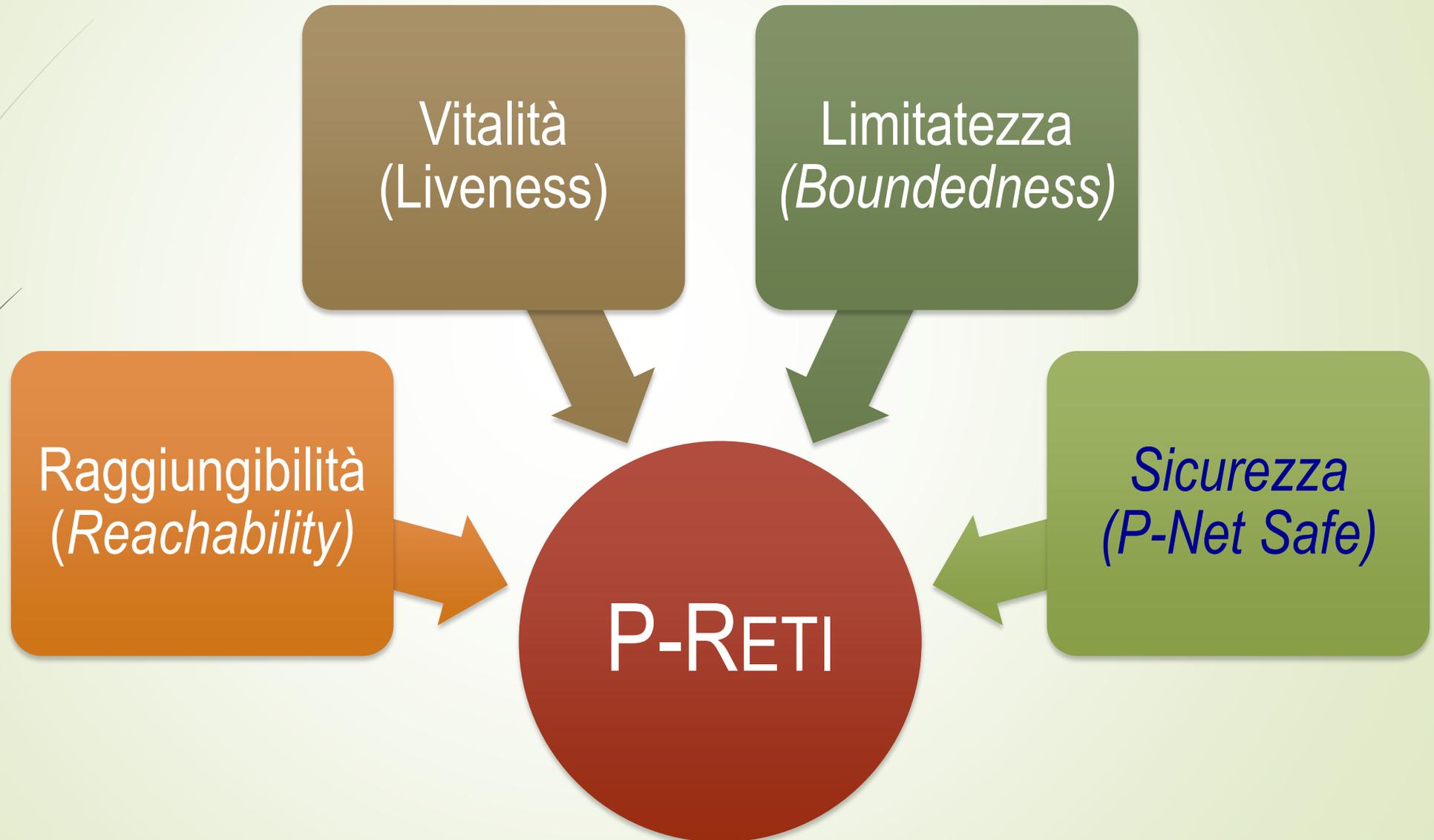
► Reti di Petri

- Caratteristiche delle P-Reti
- Problemi classici
- Schema di risoluzione di un problema con le P-Reti
- Tipologie principali delle P-Reti

► Reti Scale Free

- Note sulle reti a invarianza di scala
- Dalle reti casuali alle reti ad invarianza di scala

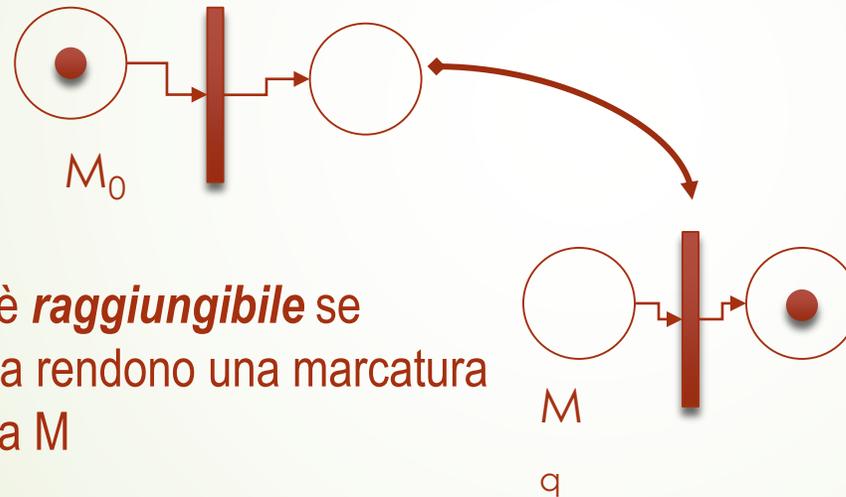
Caratterizzazione delle P-Reti



Raggiungibilità nelle P-Reti

Raggiungibilità
(Reachability)

- Data una **MARCATURA INIZIALE M_0** in una *rete di Petri G* si indica con **$R(G, M_0)$** l'insieme delle MARCATURE RAGGIUNGIBILI a partire da M_0 .



Una Marcatura M_q è **raggiungibile** se esistono scatti che la rendono una marcatura possibile a partire da M

Problema della raggiungibilità

$M_q \in R(G, M_0)$?

La marcatura in esame appartiene all'insieme delle marcature possibili?

- **Sotto quali condizioni M_q è uno stato sbagliato?**
- **Non può e non deve essere raggiungibile.**

Esempio

PORTE APERTE E ASCENSORE NON PRESENTE

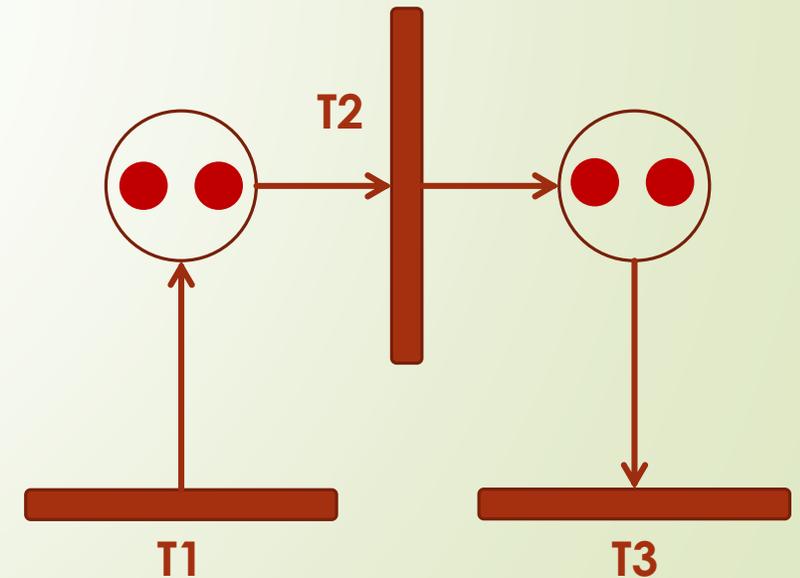
Raggiungibilità nelle P-Reti

Raggiungibilità
(Reachability)

- Data una **MARCATURA INIZIALE M_0** in una *rete di Petri G* si indica con **$R(G, M_0)$** l'insieme delle MARCATURE RAGGIUNGIBILI a partire da M_0 .

GRAFO DI RAGGIUNGIBILITÀ: grafo in cui i nodi sono le possibili marcature e gli archi le transizioni che modificano una marcatura

Buon metodo per trovare gli stati "sbagliati" ovvero che non devono essere raggiunti (...*barriere alzate e treno in passaggio...*)

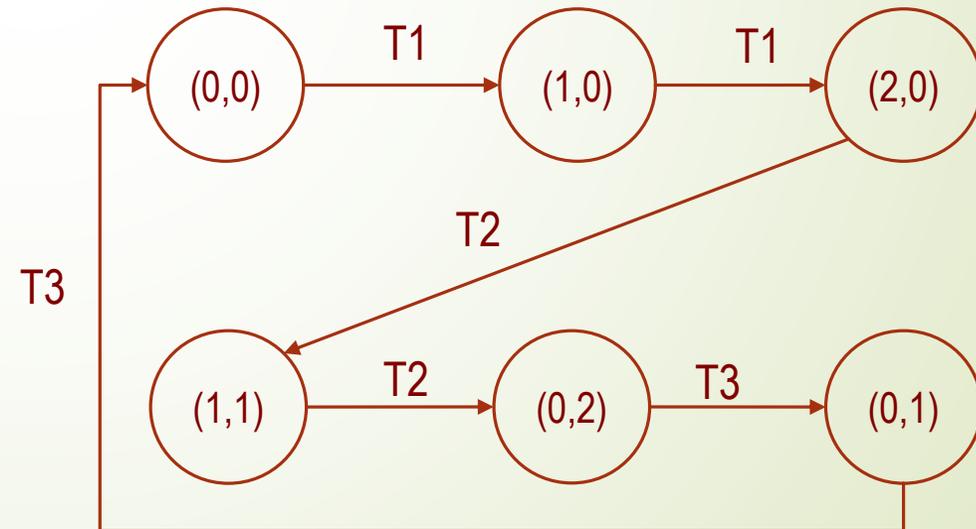
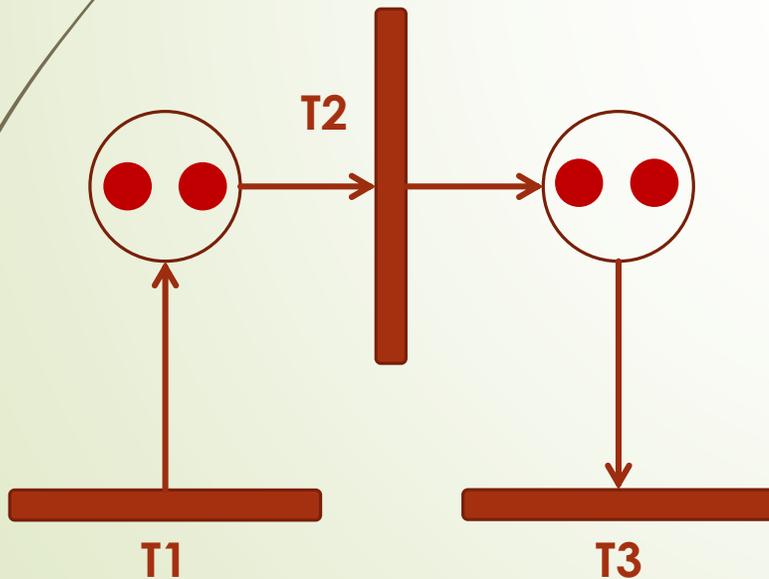


Grafo di raggiungibilità delle P-Reti

Raggiungibilità
(Reachability)

- Data una **MARCATURA INIZIALE** M_0 in una *rete di Petri* G si indica con $R(G, M_0)$ l'insieme delle MARCATURE RAGGIUNGIBILI a partire da M_0 .

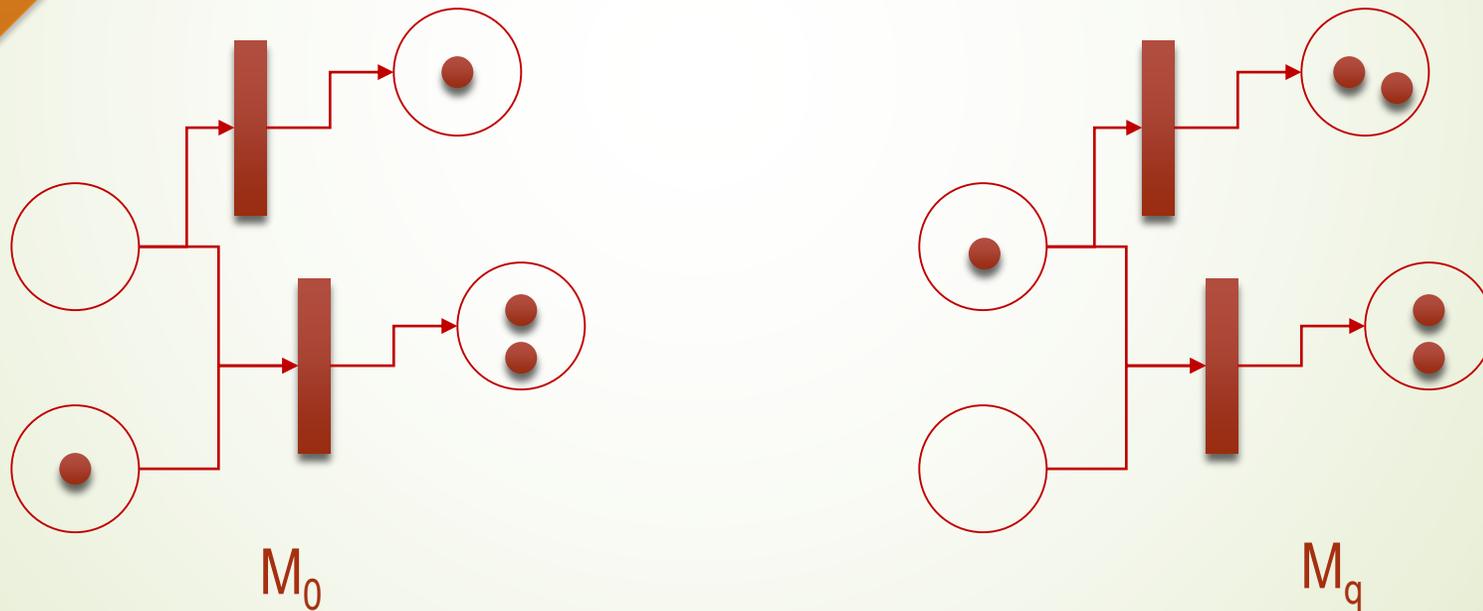
GRAFO DI RAGGIUNGIBILITÀ: grafo in cui i nodi sono le possibili marcature e gli archi le transizioni che modificano una marcatura



Limitatezza nelle P-Reti

Limitatezza
(Boundedness)

- Posto Limitato (k-limitato): k è il numero massimo di token nel posto per una qualsiasi marcatura possibile della rete
- P-Rete LIMITATA se ogni posto è limitato

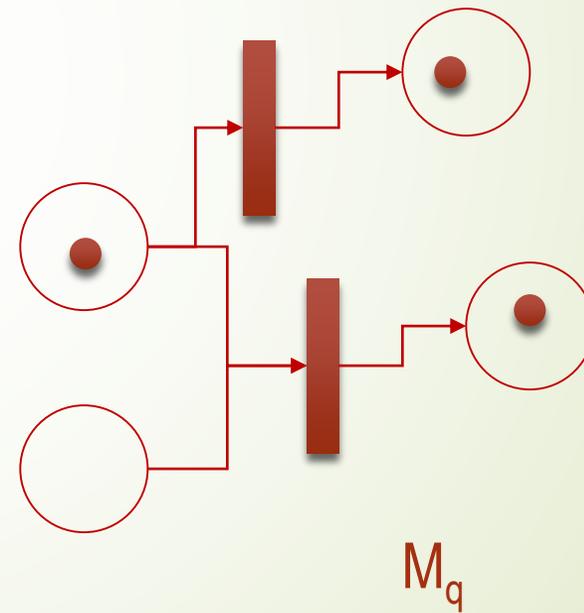
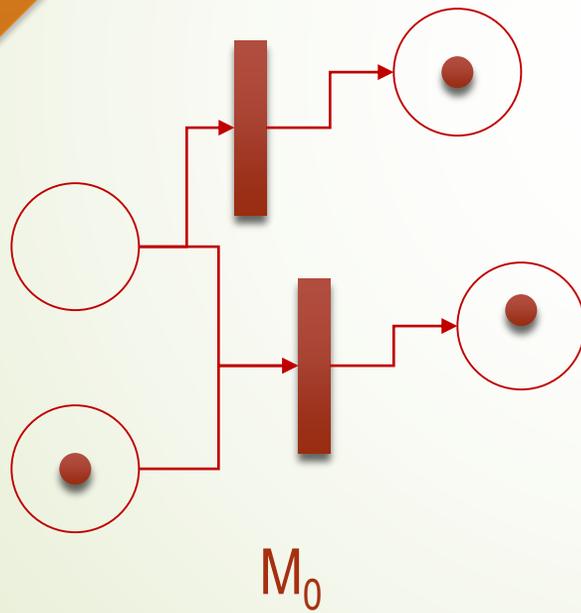


Esempio di P-Rete 2-limitata

Sicurezza nelle P-Reti

Sicurezza
(P-Net Safe)

- Una rete 1-limitata (k-limitata con $K=1$) si dice SICURA



Esempio di P-Rete 1-limitata, P-Net Safe

Vitalità nelle P-Reti

Vitalità
(Liveness)

- **L** è detto LIVELLO DI VITALITÀ e rappresenta l'attivabilità di una transizione **T** in una marcatura raggiungibile

La rete è viva se...

...è sempre possibile far scattare una transizione **T** qualsiasi a seguito di una qualsiasi sequenza di scatti

...detta M_0 marcatura qualsiasi raggiungibile da M_0 ...

P-Rete è κ -LIVE

ogni transizione **T** è κ -Live

Vitalità nelle P-Reti

Vitalità
(Liveness)

- L è detto LIVELLO DI VITALITÀ e rappresenta l'attivabilità di una transizione T in una marcatura raggiungibile

Transizione
Morta

$0 - L_0$
Live

- T non può scattare in nessuna marcatura raggiungibile

$1 - L_1$
Live

- Esiste almeno una marcatura raggiungibile in cui T può scattare

$2 - L_2$
Live

- per ogni numero intero K esiste almeno una marcatura raggiungibile in cui T può scattare K volte

$3 - L_3$
Live

- esiste almeno una marcatura raggiungibile in cui T può scattare infinite volte

Transizione
Viva

$4 - L_4$
Live

- T può scattare in ogni marcatura raggiungibile

Problemi classici per P-Reti



Problema dei 5 filosofi affamati

(dining philosophers problem, Dijkstra)

Schema per problemi di concorrenza e condivisione di risorse



Problema del barbiere che dorme

Schematizza problemi analoghi a quelli di un help desk informatizzato

Problema del barbiere che dorme



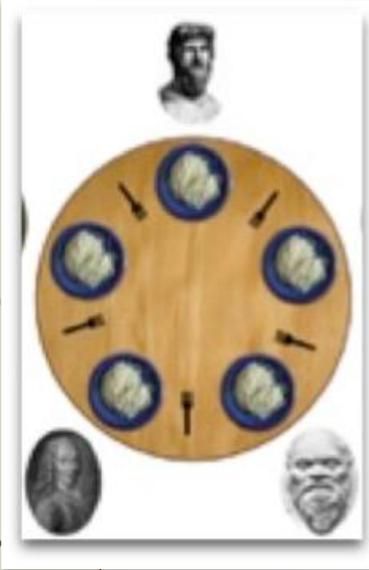
Problema del barbiere che dorme

(Schematizza problemi analoghi a quelli di un help desk informatizzato)

Un barbiere possiede un negozio con una sola sedia da lavoro e un certo numero limitato di posti per attendere. Se non ci sono clienti il barbiere dorme altrimenti, all'arrivo del primo cliente il barbiere si sveglia ed inizia a servirlo. Se dovessero sopraggiungere clienti durante il periodo di attività del barbiere, essi si mettono in attesa sui posti disponibili. Al termine dei posti di attesa, un ulteriore cliente viene scartato.

*Una corretta programmazione concorrente deve far "**dormire**" il barbiere in assenza di clienti, attivare il barbiere sul primo cliente al suo arrivo e mettere in coda tutti i successivi clienti tenendoli inattivi.*

Problema dei 5 filosofi affamati



STARVATION, (*"inedia"*) in informatica lo stato di un processo pronto per essere eseguito ma che non riesce ad ottenere le risorse di cui necessita.

Problema della cena dei 5 filosofi affamati

(Dining philosophers problem, Dijkstra)

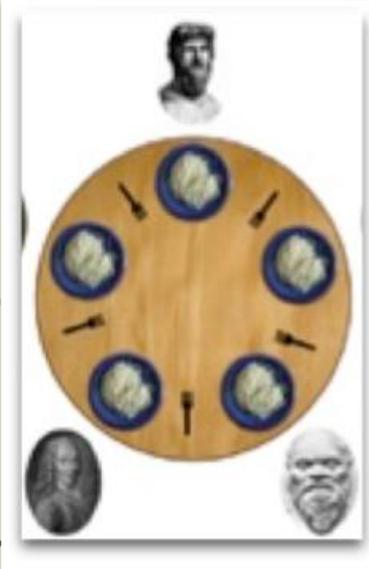
Schematizza problemi problemi di concorrenza e condivisione di risorse

1. Un filosofo può prendere solo le due bacchette che stanno alla sua destra e alla sua sinistra, una per volta, e solo se sono libere, non può sottrarre la risorsa bacchetta ad un altro filosofo che l'ha già acquisita e sta mangiando (no preemption, non c'è predominanza).

2. Finché non riesce a prendere le bacchette, il filosofo deve aspettare affamato. Quando invece è sazio posa le bacchette al loro posto e si mette a pensare per un certo tempo.

*Una corretta programmazione concorrente deve essere in grado di far mangiare alternativamente tutti i filosofi evitando che qualcuno in particolare soffra di **starvation** ed evitando che si verifichino stalli in fase di "acquisizione delle bacchette".*

Problema dei 2 filosofi affamati



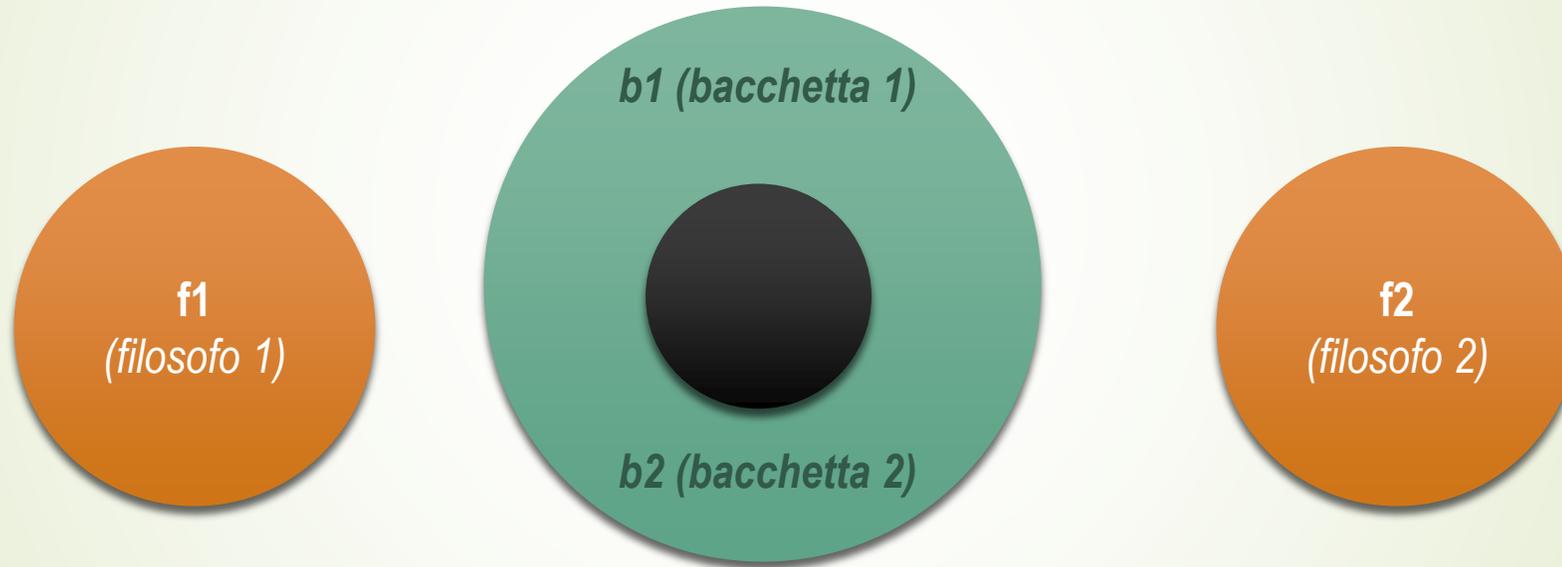
Problema della cena dei 2 filosofi affamati

(Dining philosophers problem, Dijkstra)

- 1. Nietzsche ed Eraclito mangiano spesso assieme*
- 2. Siedono attorno ad un tavolo rotondo e hanno, ognuno, a disposizione un piatto di cibo e due singole bacchette sono collocate ai lati dei loro piatti*
- 3. Sempre, o pensano o mangiano*
- 4. Quando uno dei due comincia ad avere fame cerca di prendere possesso delle bacchette alla sua destra e alla sua sinistra, in ordine arbitrario*
- 5. Qualora riesca a prenderle entrambe, mangia per un po'. Successivamente depone le bacchette e si rimette a pensare*
- 6. Nessuno dei due è in grado di mangiare con una singola bacchetta o con le mani*
- 7. Il problema consiste nel far in modo che entrambi i filosofi riescano a cibarsi e pensare periodicamente*

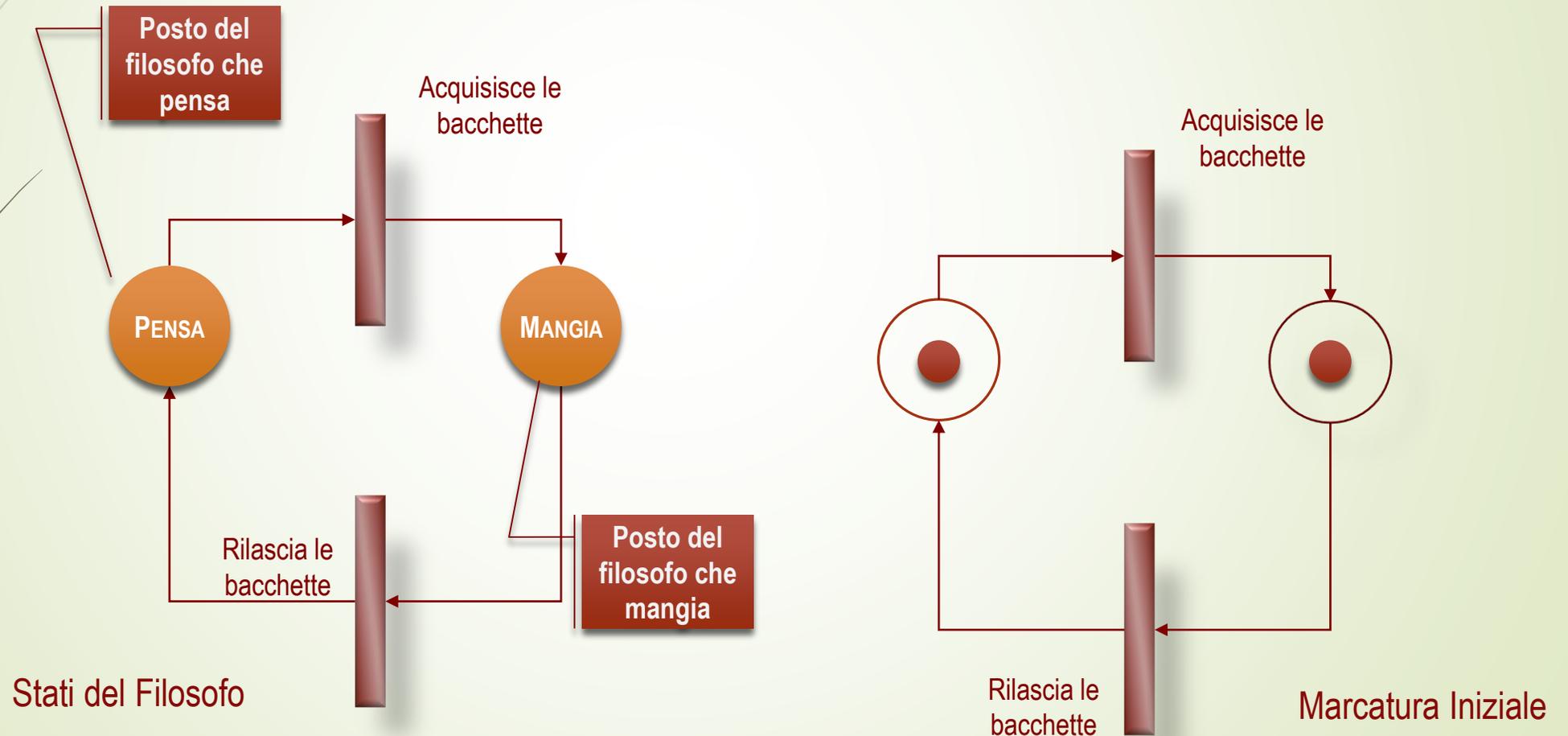
Problema dei 2 filosofi affamati

Il schema



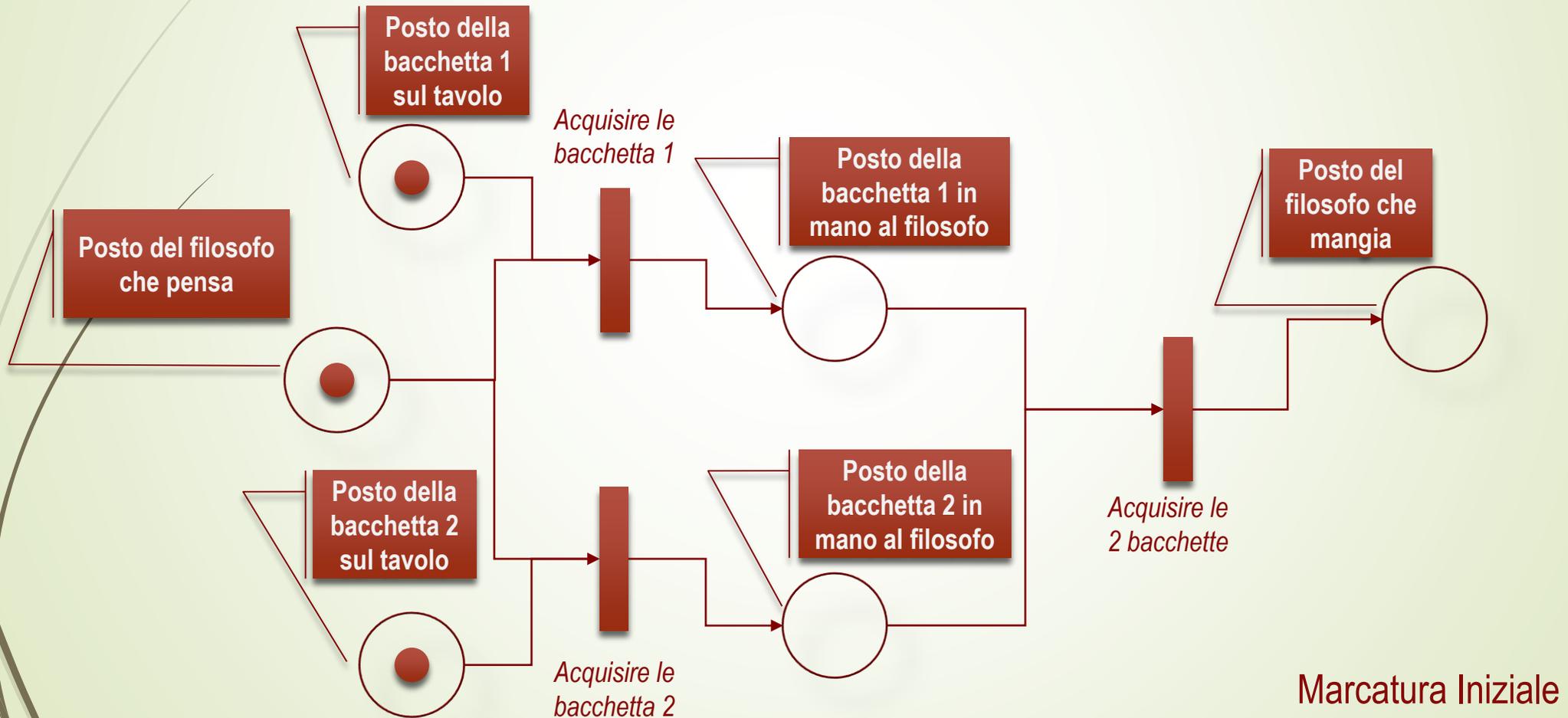
Problema dei 2 filosofi affamati

Il schema, logico per un filosofo



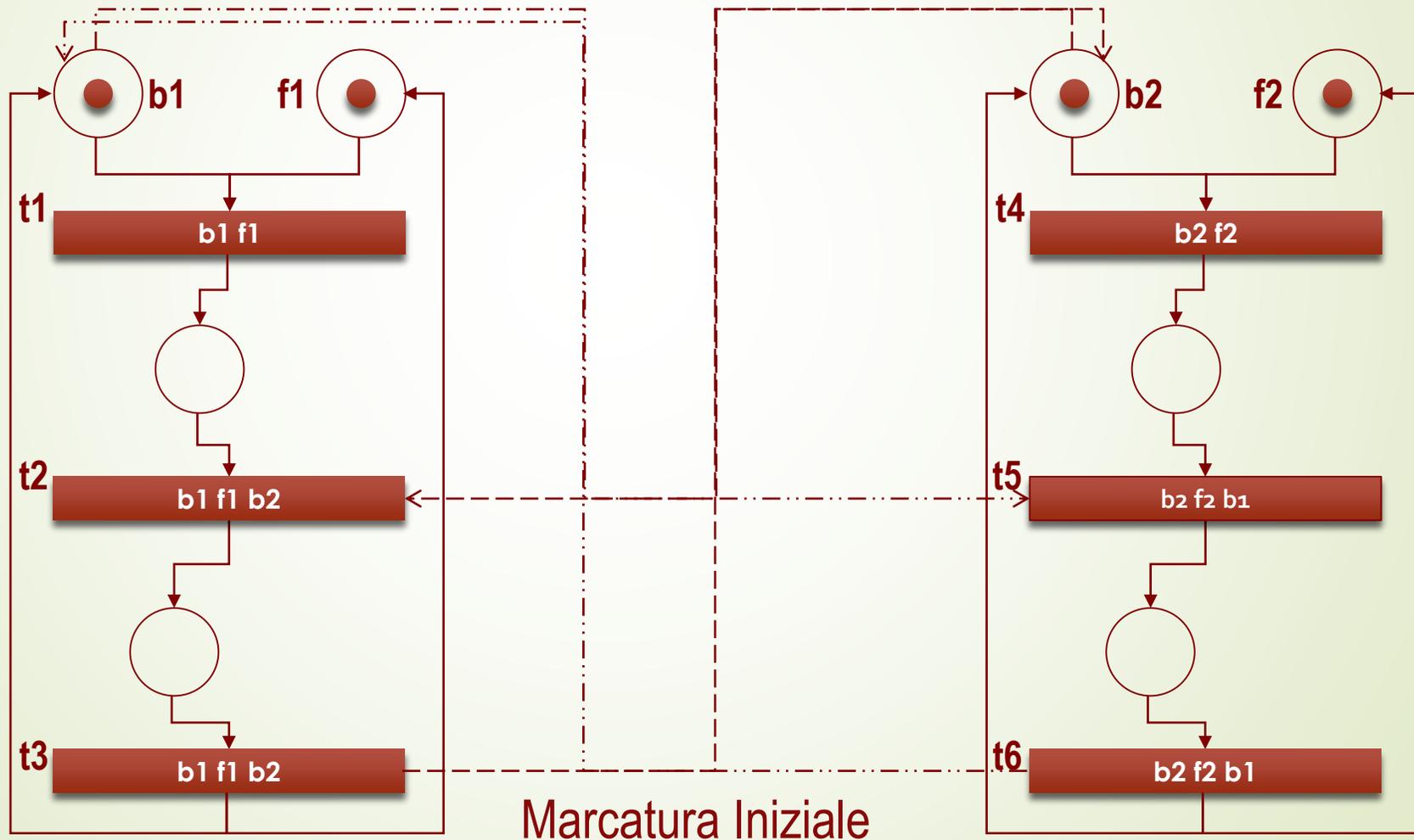
Problema dei 2 filosofi affamati

Il schema, logico per un filosofo



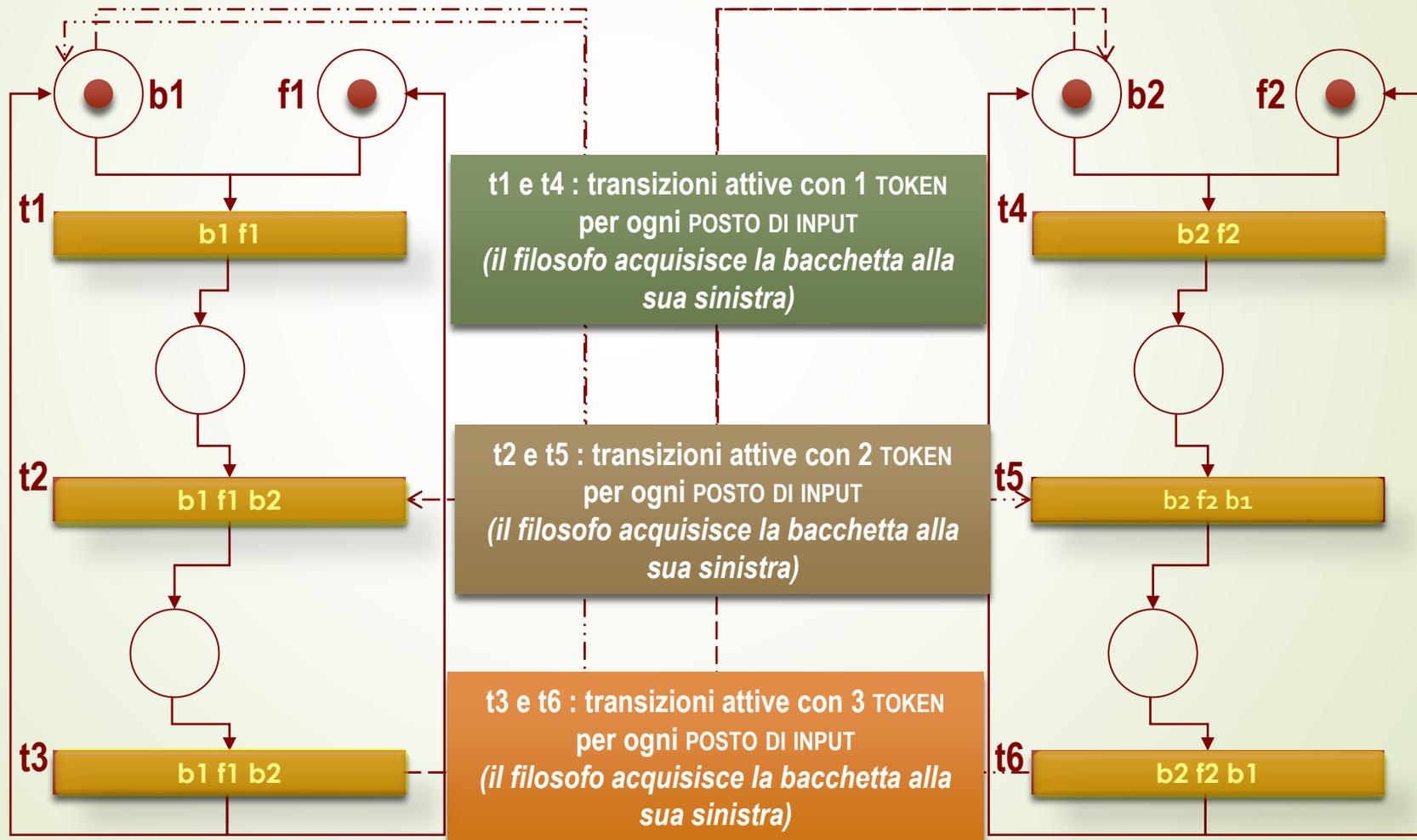
Problema dei 2 filosofi affamati

Schema logico del problema



Problema dei 2 filosofi affamati

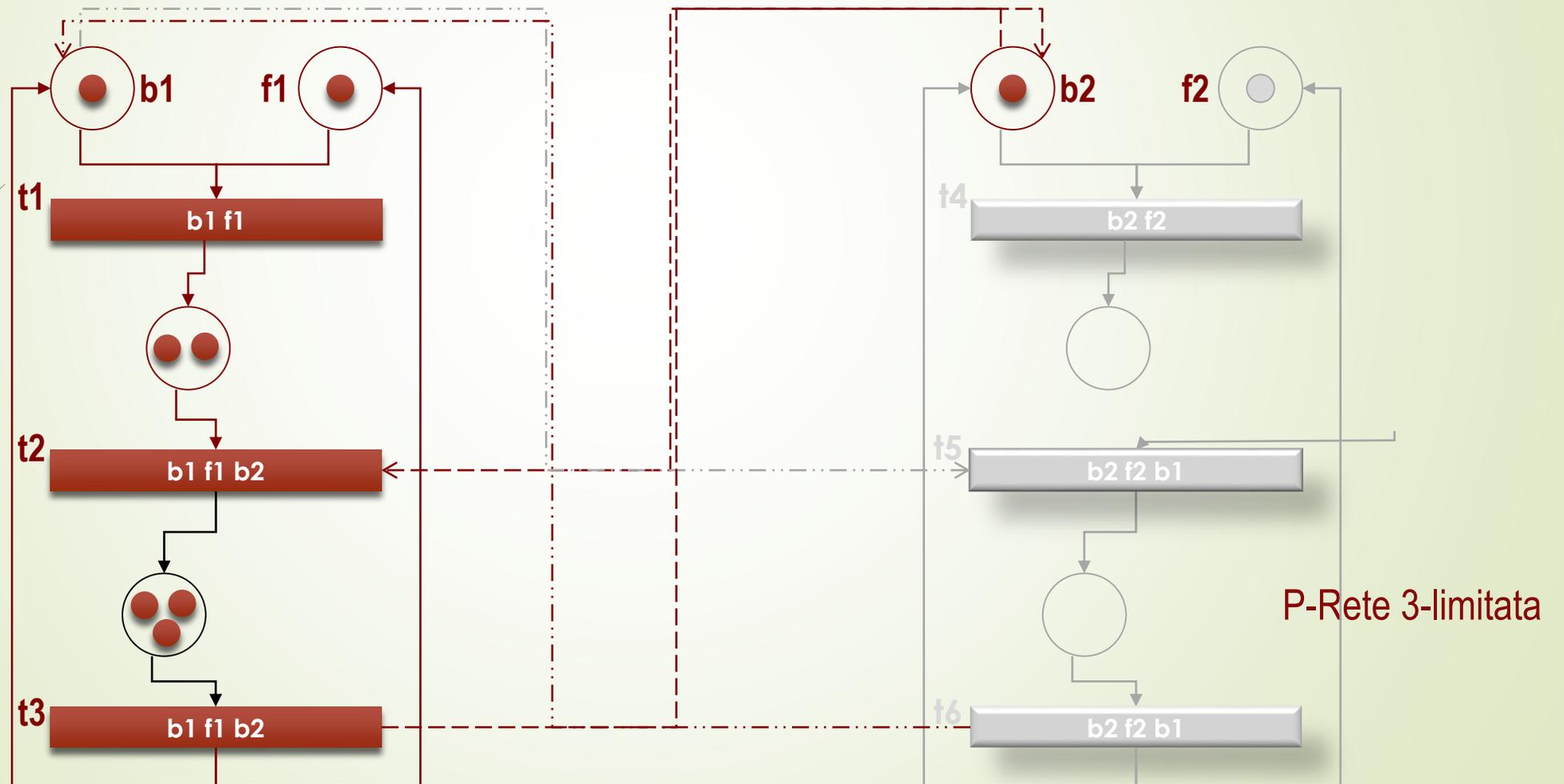
Schema logico del problema



P-Rete 3-limitata

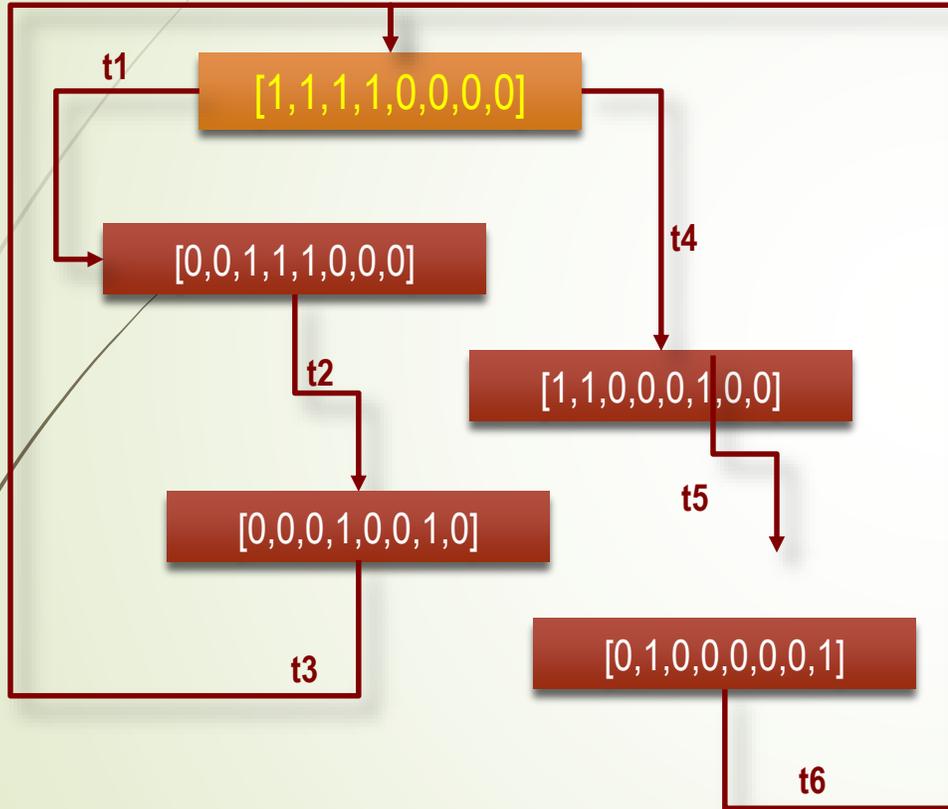
Problema dei 2 filosofi affamati

Schema funzionamento per 1 filosofo



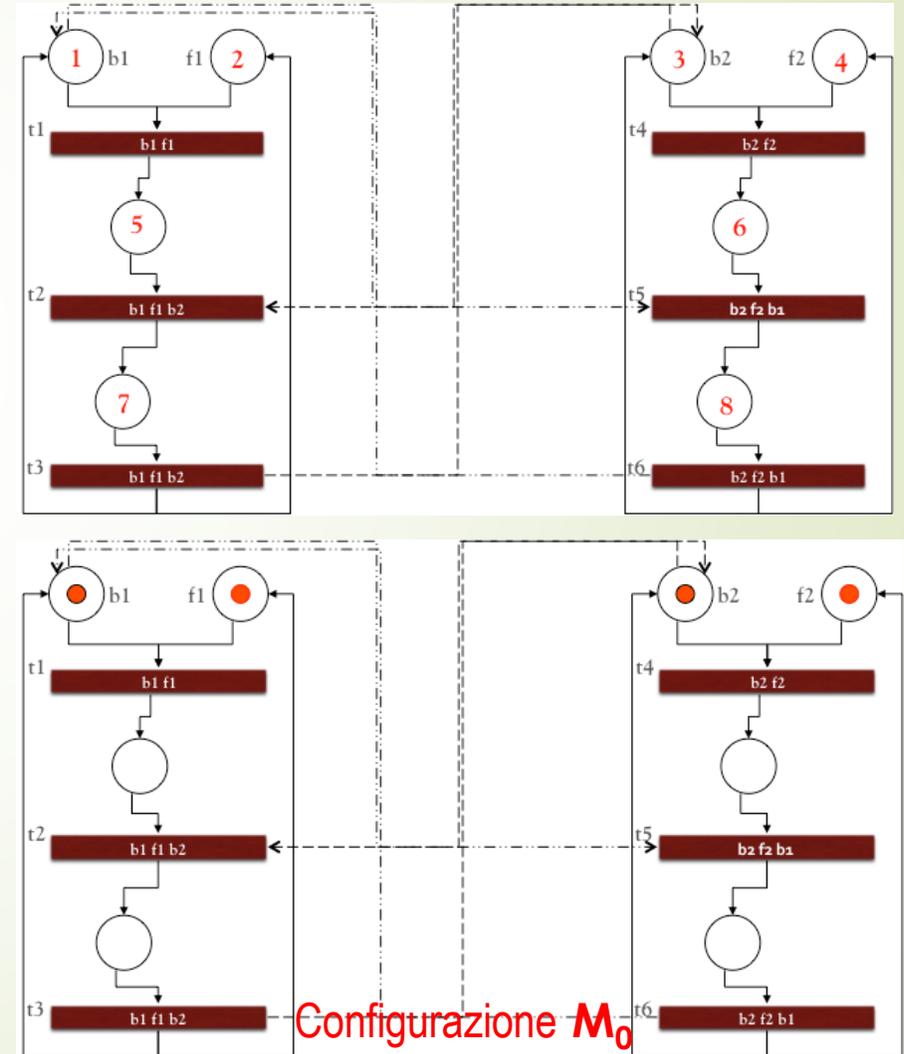
Problema dei 2 filosofi affamati

Grafo di raggiungibilità



MARCATURA, un insieme di 8 valori, in ordine, 1 se ci sono TOKEN 0 se non ci sono nello stato i .

6 marcature possibili



Configurazione M_0

Costrutti nelle P-Reti



Concorrenza



Conflitto



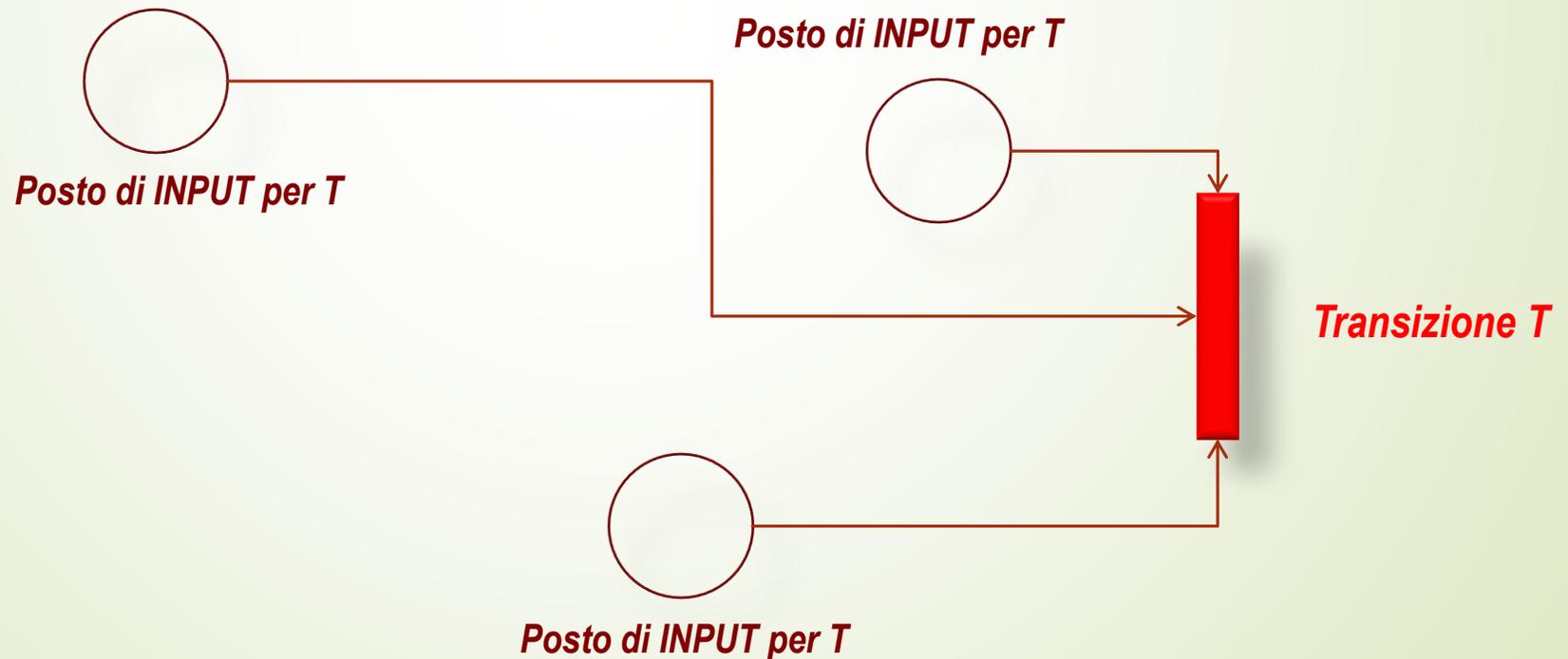
Confusione

Concorrenza nelle P-Reti



Concorrenza

Una TRANSIZIONE ha più POSTI di INPUT



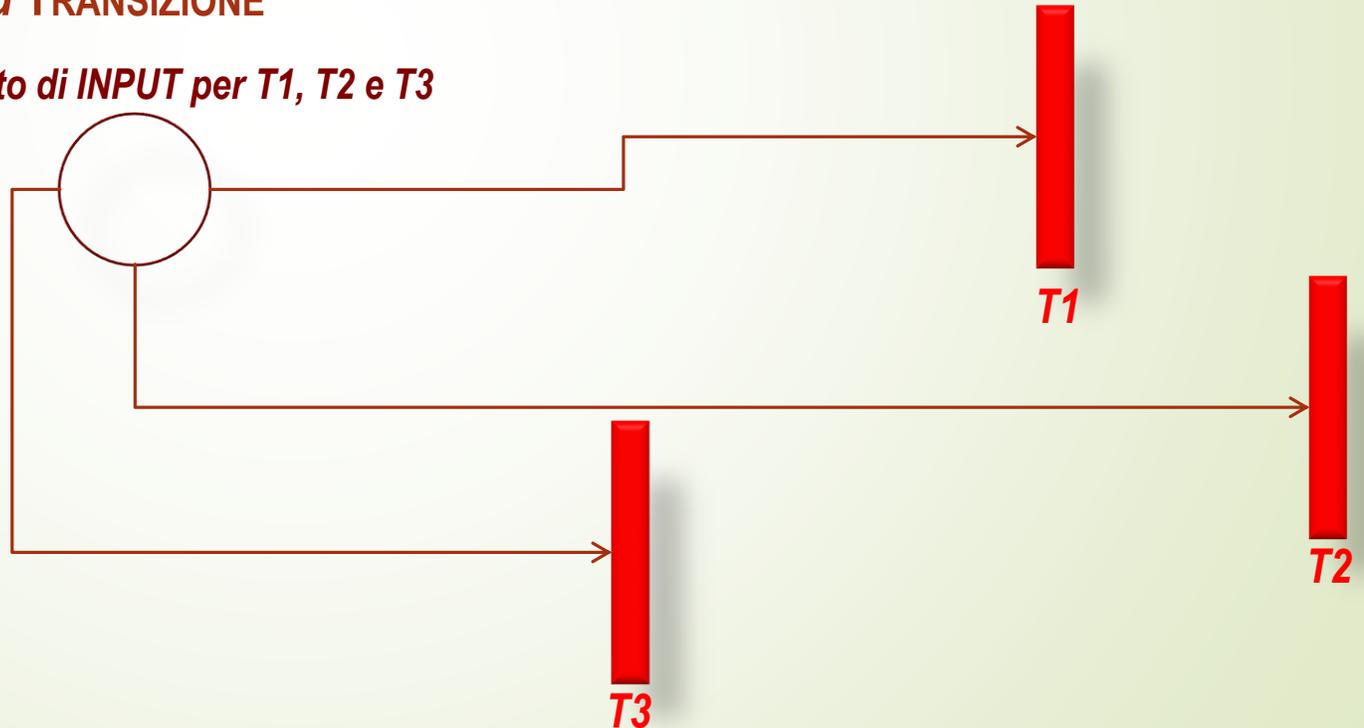
Conflitto nelle P-Reti



Conflitto

Una POSTO è INPUT per più TRANSIZIONE

Posto di INPUT per T1, T2 e T3

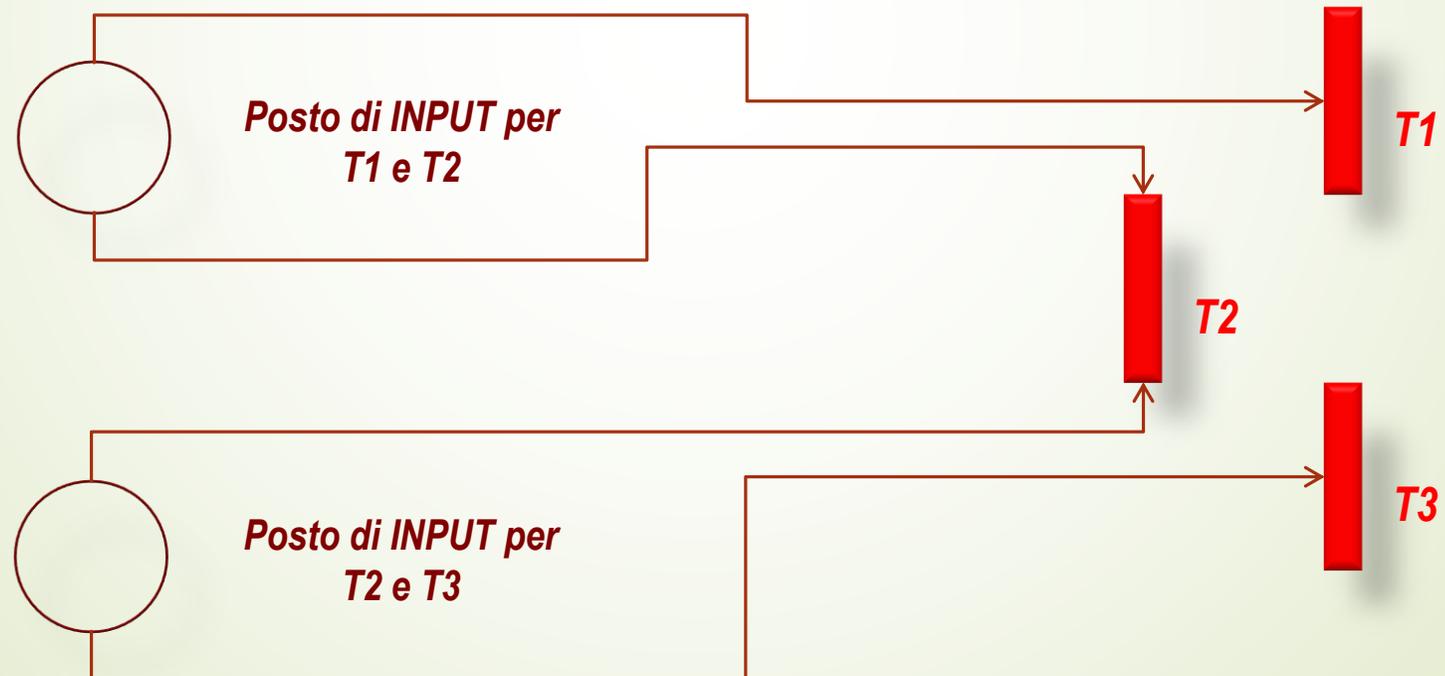


Confusione nelle P-Reti

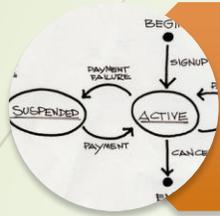


Confusione

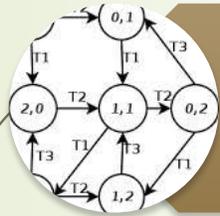
Unione di **CONCORRENZA** e **CONFLITTO**



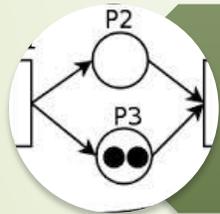
P-Reti: Tipologie notevoli



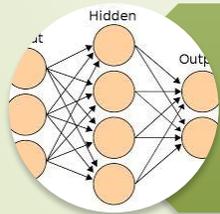
Macchine a Stati Finiti (SM: State Machine)



Grafo Marcato (MG: Marked Graph)



Scelta Libera (FC: Free Choice)



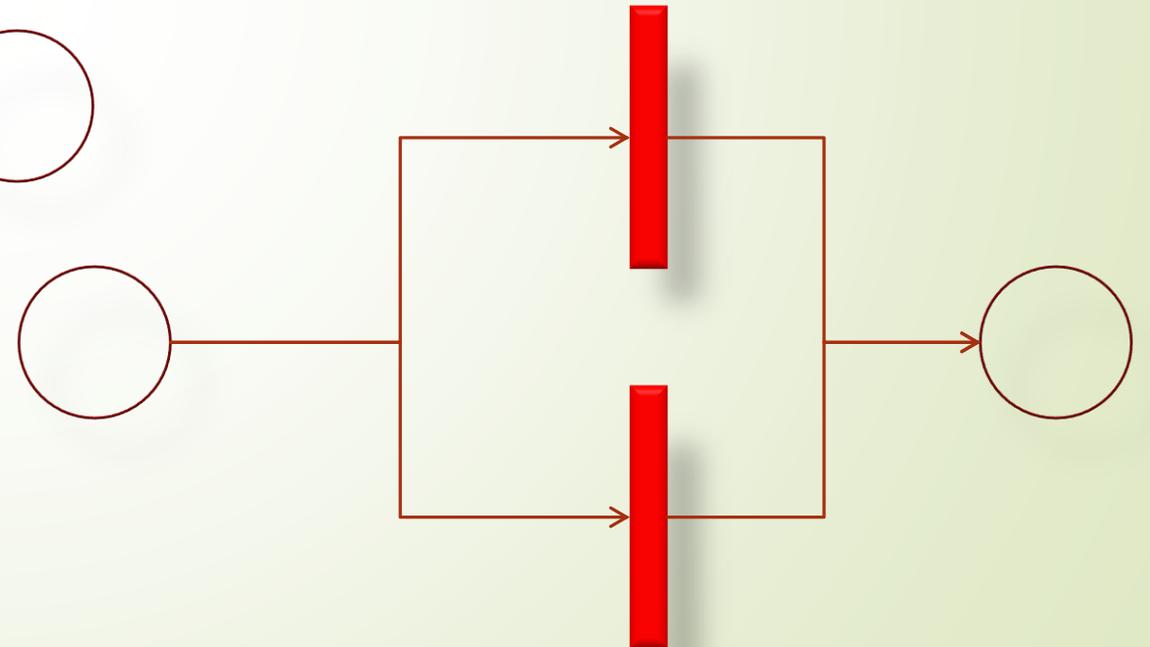
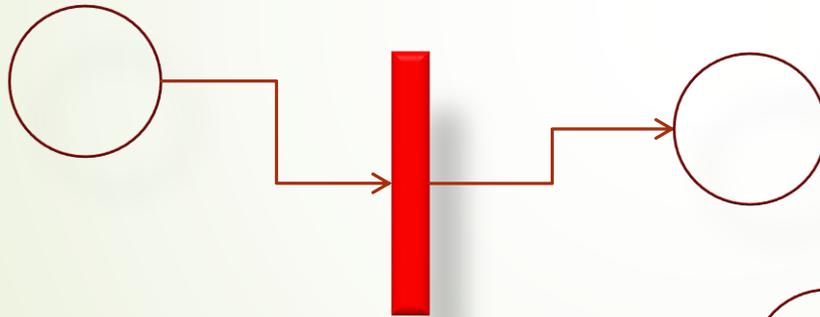
Scelta Asimmetrica (AC: Asymmetric Choice)

P-Reti: Tipologie notevoli



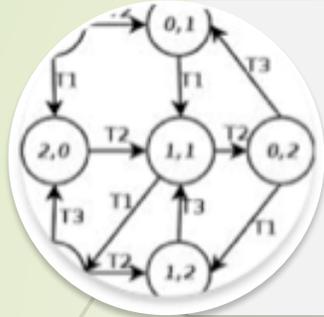
Macchine a Stati Finiti (SM: State Machine)

Ogni **TRANSIZIONI** ha un solo arco in entrata e uno solo in uscita



- Nessuna **CONCORRENZA**
- Nessun **CONFLITTO**

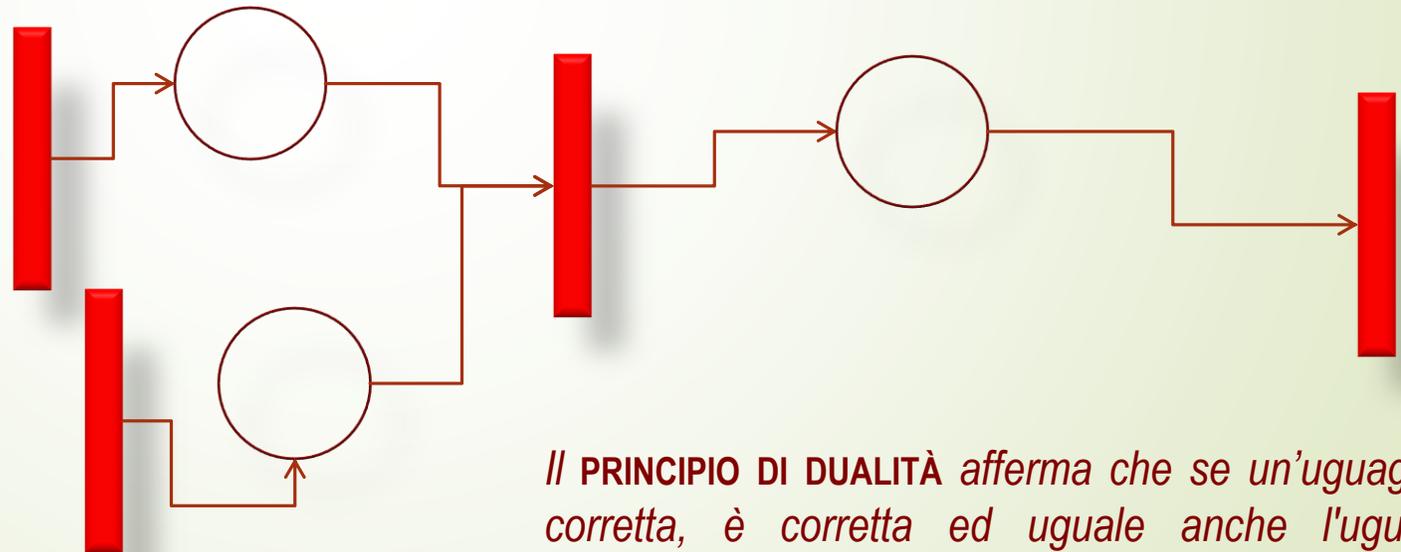
P-Reti: Tipologie notevoli



Grafo Marcato (MG: Marked Graph)

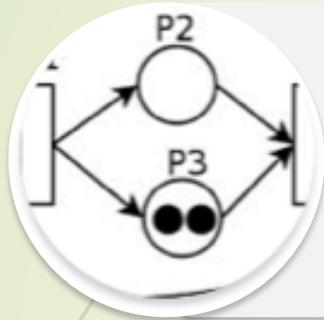
Ogni **POSTO** ha un solo arco di entrata e uno solo di uscita
(Duale della Macchina a Stati Finiti)

- **Possibili** Concorrenze
- **Nessun** Conflitto



Il **PRINCIPIO DI DUALITÀ** afferma che se un'uguaglianza è corretta, è corretta ed uguale anche l'uguaglianza ottenuta sostituendo da tutte e due le parti 1 con 0 e 0 con 1.

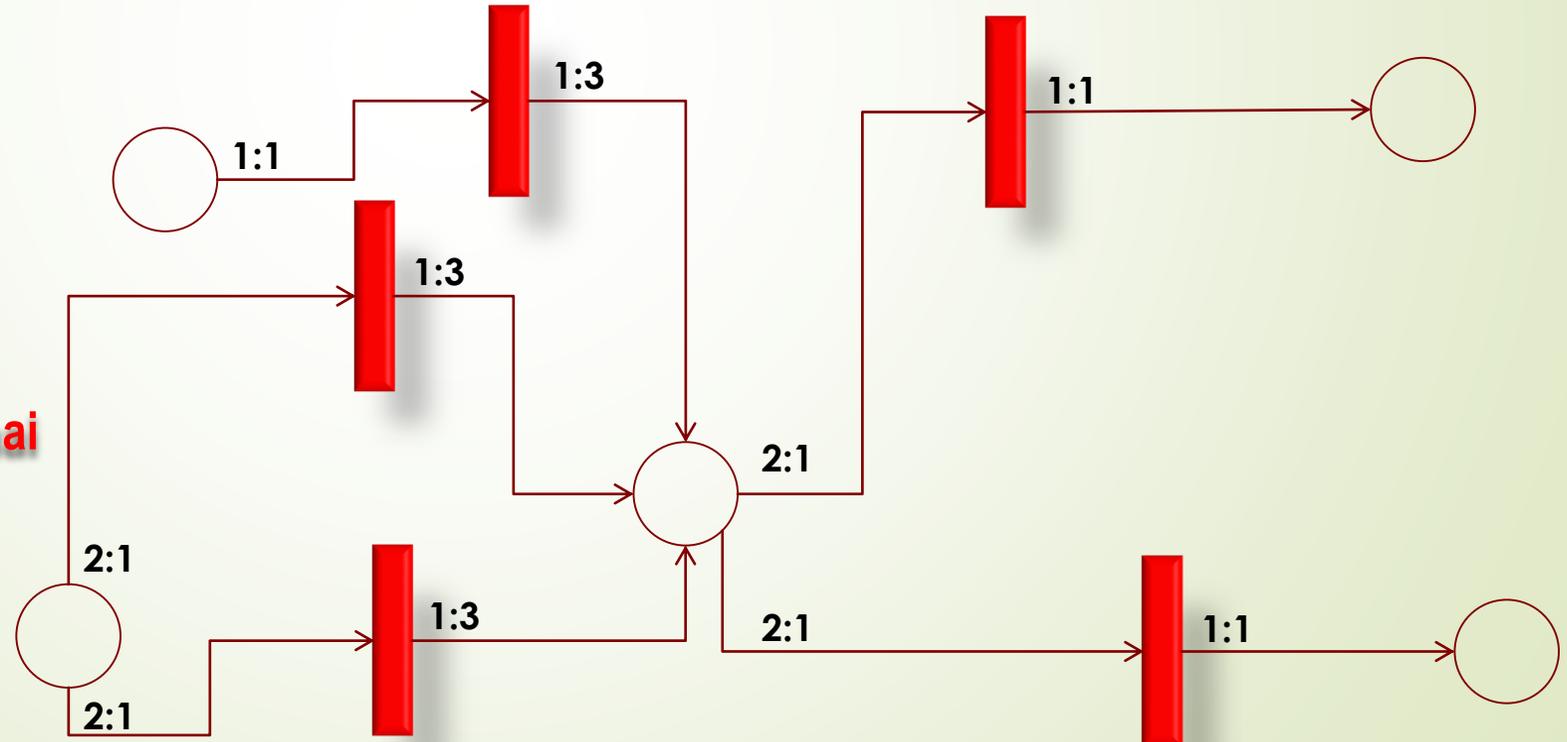
P-Reti: Tipologie notevoli



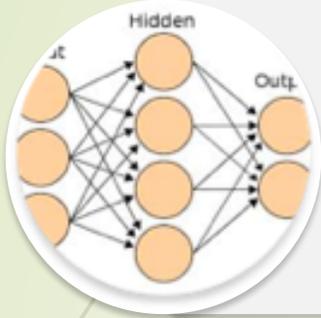
Scelta Libera (FC: Free Choice)

Ogni ARCO è o l'unico che esce da un POSTO o l'unico che entra in una TRANSIZIONE

- Possibili Concorrenze
- Possibili Conflitti
- **Conflitti e Concorrenze mai simultaneamente**



P-Reti: Tipologie notevoli



Scelta Asimmetrica (AC: Asymmetric Choice)

Se 2 Posti (A e B) sono **POSTI di INPUT** per una stessa **TRANSIZIONE**, l'insieme delle **TRANSIZIONI** per cui A è **POSTO di INPUT** contiene le **TRANSIZIONI** per cui B è **POSTO di INPUT**

- Possibili Concorrenze
- Possibili Conflitti
- **Conflitti + Concorrenze (Confusione MAI ASIMMETRICAMENTE)**

