

Reti di Petri - Introduzione

Premessa (1)

- Una Rete di Petri (Petri Net - PN) è
 - un **modello** astratto e formale
 - per la rappresentazione del **comportamento dinamico** di sistemi **discreti**
 - che esibiscono attività **asincrone e concorrenti**

Premessa (2)

- Molto usate nella modellizzazione di
 - Sistemi concorrenti
 - Interazione tra sistemi diversi, compresa utente-computer
 - Protocolli di comunicazione
 - Workflow
 - Sistemi complessi
 - ...

Notazione (1)

- Concettualmente una PN è costituita
 - da un insieme di elementi, detti **posti**, che rappresentano i possibili stati del sistema
 - da un insieme di **transizioni**, che rappresentano gli eventi che quando si verificano determinano cambiamenti di stato
 - da un insieme di elementi, detti **token**, la cui presenza/assenza/numerosità/tipo/... permette l'attivazione delle transizioni

Notazione (2)

- Le PN sono rappresentate come grafi
 - i cui nodi sono
 - i posti, raffigurati con dei cerchi
 - le transizioni, raffigurate con dei rettangoli
 - i cui archi sono i link che collegano posti e transizioni
- I token sono rappresentati come pallini all'interno dei posti

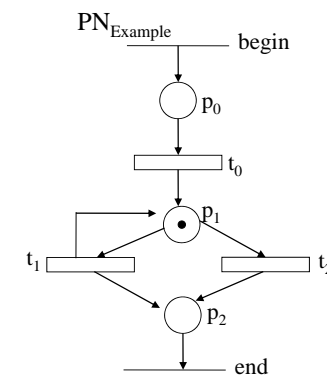
Notazione (3)

- **Attenzione:** a differenza di altri formalismi, le transizioni di stato **NON** sono raffigurate come i link tra gli stati
- Gli archi di input collegano posti con transizioni
- Gli archi di output collegano transizioni con posti

Notazione (4)

- In un dato istante uno o più posti possono essere **marcati**
 - La marcatura (**token**) indica che determinati eventi relativi a quel posto si sono compiuti
 - Una marcatura è indicata da un pallino all'interno di un posto

Esempio (1)



Esempio (2)

- PN_{Example} è il **nome** della rete
- I **posti** sono p_0, p_1, p_2 , di cui
 - p_0 è l'unico **posto iniziale**
 - p_1 ha una marca
 - p_2 è l'unico **posto finale**
- Le **transizioni** sono t_0, t_1, t_2

Informalmente (1)

- Un posto rappresenta una **locazione** in cui risiedono i token
 - un posto può essere considerato come una memoria (temporanea o persistente) di dati
 - in ogni istante, un posto può essere visto come un contenitore di un numero (variabile o costante) di token

Informalmente (2)

- Il comportamento dinamico di una PN è **controllato** (**gestito, determinato, ...**) dal movimento e dalla propagazione dei token
- La disponibilità di token nei posti che precedono una transizione
 - permette l'attivazione della transizione
 - determina la disponibilità di token nei posti che seguono la transizione, così da permettere la successiva propagazione

Computazione (1)

- Lo stato corrente del sistema è dato dal numero (e, quando ha senso parlarne, anche dal tipo/colore) di token presenti in ogni posto
- Le transizioni rappresentano le componenti attive del sistema
 - modellizzano le attività computazionali che sono svolte quando scatta la transizione

Computazione (2)

- L'esecuzione di una transizione determina un cambiamento nello stato del sistema
 - cambiamento delle marcature
- Le transizioni possono scattare solo quando sono **abilitate**

Abilitazione

- Una transizione di stato è **abilitata** solo se tutti i posti che precedono quella transizione hanno un opportuno grado di marcatura
 - le precondizioni per l'esecuzione di quella transizione sono soddisfatte

PN: Definizioni

- $PN = \langle \text{Name}, P, T, E, P', Q, M \rangle$, dove
 - Name è il nome della rete
 - $P = \{p_0, p_1, \dots, p_n\}$ insieme finito, non vuoto di posti
 - $T = \{t_0, t_1, \dots, t_m\}$ insieme finito non vuoto di transizioni
 - $E \subseteq (P \times T) \cup (T \times P)$ insieme di archi orientati che vanno da posti a transizioni o da transizioni a posti
 - $P' \subseteq P$ insieme finito non vuoto di posti iniziali
 - $Q \subseteq P$ è insieme finito non vuoto di posti finali
 - $M : P \rightarrow \{0, 1\}$ è la funzione di marcatura che associa ad ogni posto di P il valore 0 o 1
 - $\forall p_i$ se $M(p_i)=1$ allora p_i si dice **marcato**, altrimenti **non marcato**

Preset/Postset

- $\forall t_i$ il suo **preset** è l'insieme di posti che la precedono
 - È indicato con $\text{Pr}(t_i)$
 - Se $\forall p_j \in \text{Pr}(t_i) M(p_j)=1$ allora diciamo che $\text{Pr}(t_i)$ è **marcato**
- $\forall t_i$ il suo **postset** è l'insieme di posti che la seguono
 - È indicato con $\text{Post}(t_i)$

Attivazione

- Una transizione è **attivabile** se il suo preset è marcato
- Quando una transizione t_i viene **attivata** allora
 - I posti del suo preset **perdono** una marca
 - $\forall p_j \in \text{Pr}(t_i) M(p_j)=0$
 - I posti del suo postset ne **acquisiscono** una
 - $\forall p_h \in \text{Post}(t_i) M(p_h)=1$

Esecuzione di una PN

- Eseguire una PN significa stabilire uno o più cammini che legano uno o più elementi di P' a un elemento di Q
- È necessario marcare tutti i posti in P'
- Vengono poi attivate in sequenza tutte le transizioni attivabili lungo un cammino
- L'esecuzione termina quando un elemento in Q è marcato

Costrutto di sequenza

- La sequenza in una PN è definita dalle transizioni che permettono di passare da un posto nel Preset a un posto in Postset

Costrutto di iterazione

- Il costrutto di iterazione in una PN è realizzato con uno più archi che collegano una transizione a uno o più posti del suo preset
 - Ogni volta che la transizione viene attivata, il suo preset viene nuovamente marcato, e quindi la transizione è nuovamente attivabile

Esempio (1)

- Un sistema portuale prevede
 - un molo, dove attraccano le navi
 - un rimorchiatore, che scorta l'ingresso/uscita delle navi
- Le navi arrivano all'esterno di un porto con una frequenza di 80 minuti
- Una nave è scortata dall'esterno al molo da un rimorchiatore
 - tempo richiesto: 30 min.
 - solo dopo l'attracco della nave al molo il rimorchiatore diventa disponibile per altro

Esempio (2)

- Le operazioni di scarico della nave richiedono 60 min
- Dopo aver completato lo scarico, se il rimorchiatore è disponibile scorta la nave all'esterno
 - tempo richiesto 60 min
 - il molo diventa disponibile per altre navi

Esempio (3)

