MODELLI DELLA CONCORRENZA

Lucia Pomello

Le reti di Petri: i sistemi elementari

Lezioni del 17 e 21 novembre 2022

Corso di Laurea Magistrale in Informatica
Dipartimento di informatica, sistemistica e comunicazione
Universitá degli studi di Milano–Bicocca

Modelli di sistemi concorrenti e distribuiti

Algebre di processi

-*Calculus of Communicating Systems*, R. Milner, 1980;
-*Communicating Sequential Processes*, C.A.R. Hoare, 1978
- FSP (Finite State Processes),...

modellano un sistema concorrente/distribuito come costituito dalla composizione in parallelo di processi sequenziali che interagiscono tra loro in modo sincrono (*handshaking*).

Automi a stati finiti

-*modelli di reti neurali* (W. McCulloch e W. Pitts 1943)
-*progettazione di circuiti asincroni* (D. A. Huffman e G.H. Mealy , 1954 e 55);
-*macchina a stati finiti* (E. F. Moore 1956)
-*riconoscitori di linguaggi regolari* (S. Kleene, Espr. Reg. 1951)
-*modelli di protocolli di comunicazione*.....

→ **Sistemi di transizioni etichettati**
-*utilizzati per la semantica a "interleaving" delle algebre di Processi (es:CCS)...*

Reti di Petri (C.A. Petri 1962)

....

Teoria Generale delle reti di Petri

Carl Adam Petri (1926 - 2010)
PhD thesis *Kommunikation mit Automaten*(Communication with Automata) 1962

obiettivi:
"sviluppo di una teoria matematica fondata sui principi della fisica moderna (teoria della relatività e fisica quantistica), che sia una **teoria dei sistemi** in grado di descrivere il **flusso di informazione** e permetta di analizzare **sistemi con organizzazione complessa**"

Teoria generale dei sistemi che processano informazione basata su:
- comunicazione;
- sincronizzazione;
- flusso di informazione;
- relazione di concorrenza (indipendenza causale);
- ...

Sviluppo e applicazione della teoria delle reti di Petri

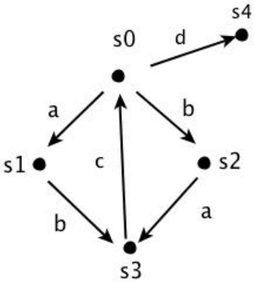
A partire dall'inizio degli anni '70 la teoria è stata sviluppata:

- dal punto di vista dell'espressività del linguaggio (diverse classi di reti);
- tecniche formali di analisi e di verifica di proprietà del modello (algebra lineare; teoria dei grafi);
- notevole utilizzo in diversi ambiti applicativi:
  - specifica di protocolli di comunicazione;
  - disegno di circuiti asincroni;
  - algoritmi e programmi concorrenti / paralleli;
  - modelli di sistemi organizzativi (es: workflow,...),
  - di sistemi software/database distribuiti,
  - di sistemi di controllo anche industriali;
  - modelli di sistemi biologici; di reazioni chimiche;
  - valutazioni delle prestazioni (reti stocastiche, reti temporizzate);
  - ...

I Sistemi di Transizioni Etichettati - LTS

$$A = (S, E, T, s_0)$$

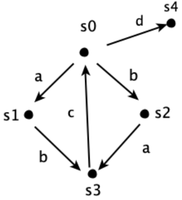
- S insieme degli **stati** (globali)
- E insieme delle **azioni**
- $T \subseteq S \times E \times S$ insieme di **archi etichettati** (*transizioni*)
- s_0 **stato iniziale** (opzionale)



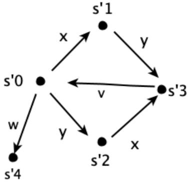
isomorfismo tra LTS

Due sistemi di transizioni etichettati (S_1, E_1, T_1, s_{01}) e (S_2, E_2, T_2, s_{02}) sono **isomorfi** ((S_1, E_1, T_1, s_{01}) **iso** (S_2, E_2, T_2, s_{02})) sse esistono due mappe biunivoche $\alpha : S_1 \to S_2$ e $\beta : E_1 \to E_2$ tali che:

- $\alpha(s_{01}) = s_{02}$
- $\forall s, s' \in S_1, \forall a \in E_1$: $(s, a, s') \in T_1 \Leftrightarrow (\alpha(s), \beta(a), \alpha(s')) \in T_2$



iso



- $\alpha(s_i) = s'_i$, $i = 0, ..., 4$
- $\beta(a) = x$; $\beta(b) = y$; $\beta(c) = v$; $\beta(d) = w$.

Aspetti critici dei Sistemi di transizioni

- nei sistemi distribuiti lo stato globale non è osservabile
- nella realtà fisica non esiste un sistema di riferimento temporale unico
- la simulazione sequenziale non deterministica (semantica a "interleaving") dei sistemi distribuiti è una forzatura, non ne rappresenta le caratteristiche reali del comportamento

Esempio: il produttore e il consumatore

Algebre di processi

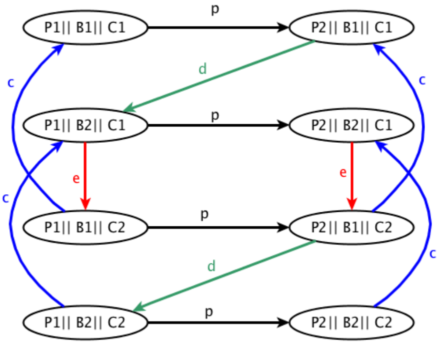
$S = (P1 \mid B1 \mid C1)_{\setminus\{deposita, estrae\}}$

il **produttore**: $P1 = produce.P2$ e $P2 = \overline{deposita}.P1$

il **consumatore**: $C1 = estrae.C2$ e $C2 = consuma.C1$

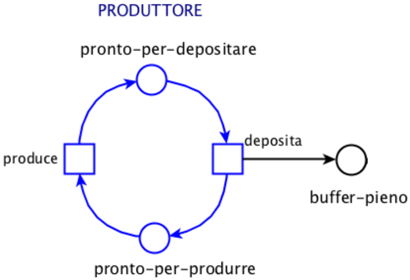
il **buffer**: $B1 = deposita.B2$ e $B2 = \overline{estrae}.B1$

Il comportamento del sistema S può essere modellato dal seguente **sistema di transizioni etichettato**:



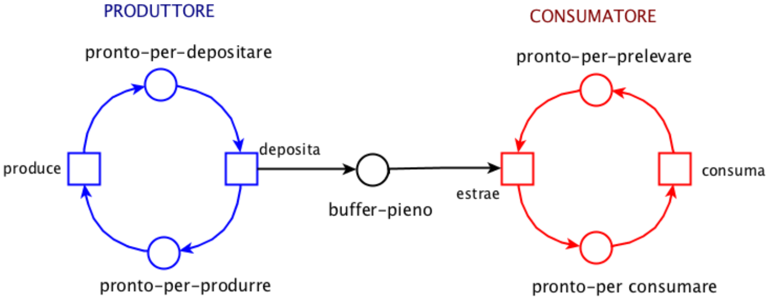
Esempio: il produttore e il consumatore

Una soluzione con le reti di Petri - i sistemi elementari



Esempio: il produttore e il consumatore

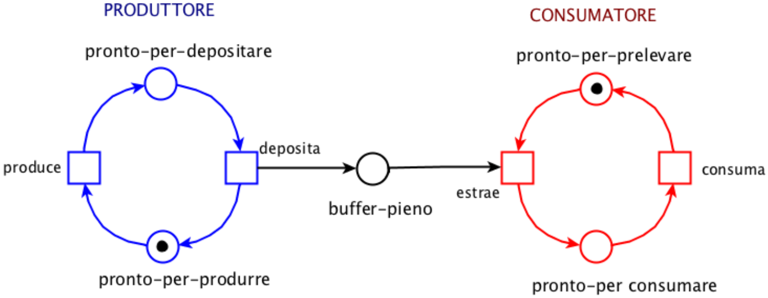
Una soluzione con le reti di Petri - i sistemi elementari



Esempio: il produttore e il consumatore

Una soluzione con le reti di Petri - i sistemi elementari

lo **stato** è definito da una collezione di *stati locali* (le condizioni vere)

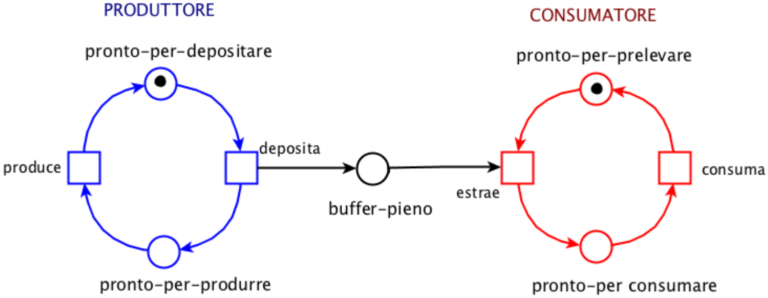


l'evento *produce* è l'unico *abilitato*

Esempio: il produttore e il consumatore

Una soluzione con le reti di Petri - i sistemi elementari

la simulazione (*il gioco delle marche*)

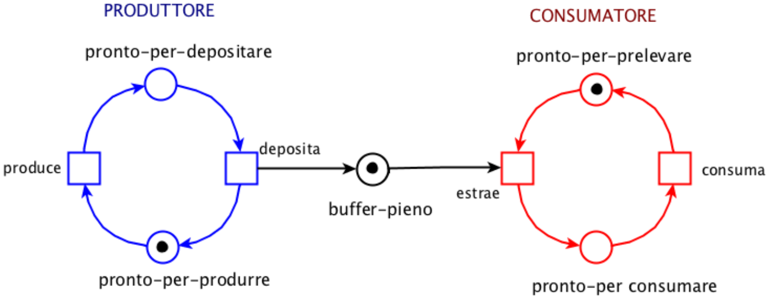


deposita è l'unico evento *abilitato*

Esempio: il produttore e il consumatore

Una soluzione con le reti di Petri - i sistemi elementari

la simulazione (*il gioco delle marche*)

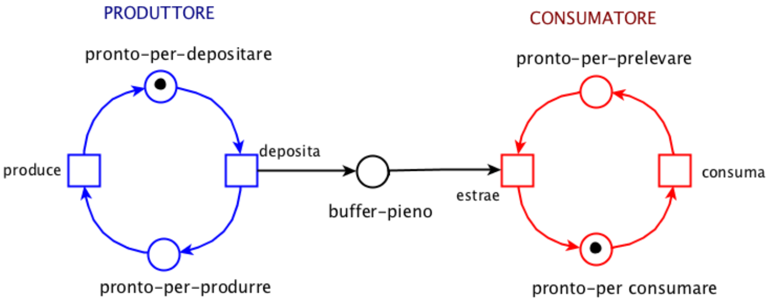


produce e *preleva* sono entrambi abilitati,
possono occorrere in modo **concorrente**.

Esempio: il produttore e il consumatore

Una soluzione con le reti di Petri - i sistemi elementari

la simulazione (*il gioco delle marche*)

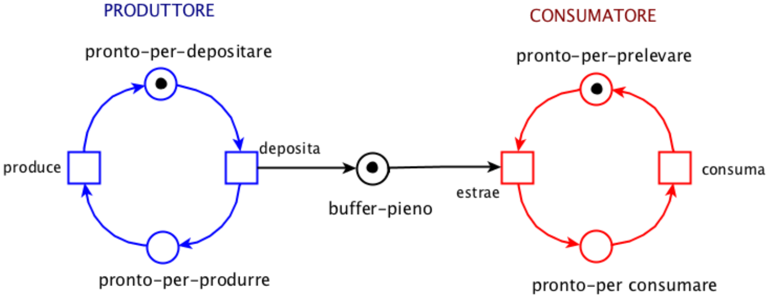


in questa configurazione, *deposita* e *consuma* sono entrambi abilitati, possono occorrere in modo **concorrente**.

Esempio: il produttore e il consumatore

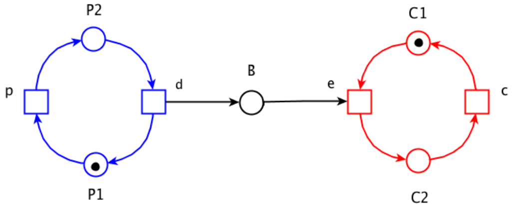
Una soluzione con le reti di Petri - i sistemi elementari

la simulazione (*il gioco delle marche*)

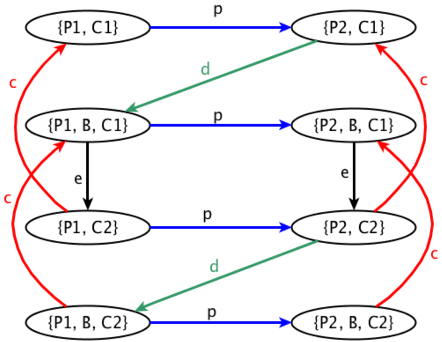


in questa configurazione, *deposita* **non** è abilitato: la post-condizione *buffer-pieno* è **vera**;
deposita può occorrere **solo dopo** *estrae* (**dipendenza causale**).

Esempio: il produttore e il consumatore



Comportamento come **sistema di transizioni etichettato**



I sistemi elementari: le reti elementari

$N = (B, E, F)$ é una **rete** se e solo se:

- B insieme finito di **condizioni** (**stati locali**, proposizioni/proprietá vere o false) rappresentate da $\bigcirc$

- E insieme finito di **eventi** (trasformazioni locali di stato, **transizioni locali**) rappresentati da $\square$

tali che : $B \cap E = \emptyset$ e $B \cup E \neq \emptyset$

- $F \subseteq (B \times E) \ \cup \ (E \times B)$ **relazione di flusso** rappresentata da $\longrightarrow$
tale che : $dom(F) \cup ran(F) = B \cup E$ (no elementi isolati)

Sia $x \in X, \ X = B \cup E$,

${}^\bullet x = \{y \in X : (y, x) \in F\}$ *pre-elementi* di x (precondizioni o pre-eventi)
$x^\bullet = \{y \in X : (x, y) \in F\}$ *post-elementi* di x (postcondizioni o post-eventi)

Sia $A \subseteq B \cup E$, ${}^\bullet A = \cup_{x \in A} {}^\bullet x$ e $A^\bullet = \cup_{x \in A} x^\bullet$

località e dualità tra stati e transizioni

I sistemi elementari

La rete $N = (B, E, F)$ descrive la *struttura statica del sistema*, il comportamento é definito attraverso le nozioni di *caso* e di *regola di scatto* (o di transizione).

Un **caso** (o *configuarazione*) é un insieme di condizioni $c \subseteq B$ che rappresentano l'insieme di condizioni vere in una certa configurazione del sistema, un *insieme di stati locali* che collettivamente individuano lo "*stato globale*" del sistema.

$\bigcirc$ condizione falsa $\qquad$ $\odot$ condizione vera

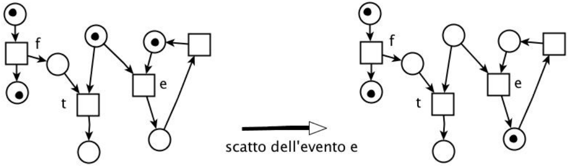
la regola di scatto

Sia $N = (B, E, F)$ una rete elementare e $c \subseteq B$.
L'evento $e \in E$ é **abilitato** (puó occorrere) in c, denotato $c[e >$, sse

$$^\bullet e \subseteq c \text{ and } e^\bullet \cap c = \emptyset$$

Se $c[e >$, allora quando e **occorrere** in c genera un nuovo caso c', denotato $c[e > c'$: $$c' = (c - {}^\bullet e) \cup e^\bullet$$

principio di estensionalitá *(il cambiamento di stato é locale):*
"un evento é completamente caratterizzato dai cambiamenti che produce negli stati locali, tali cambiamenti sono indipendenti dalla particolare configurazione in cui l'evento occorre"

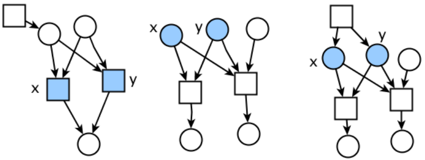


l'evento e é l'unico abilitato: le sue precondizioni sono vere, le postcondizioni false
lo scatto di e rende le precondizioni *false* e le postcondizioni *vere*,
le altre condizioni rimangono inalterate.

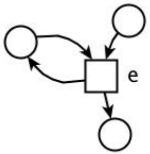
reti semplici, reti pure

Sia $N = (B, E, F)$ una rete elementare

- N è **semplice** sse
$$\forall x, y \in B \cup E,\ {}^\bullet x = {}^\bullet y \wedge x^\bullet = y^\bullet \Rightarrow x = y$$
- N è **pura** sse $\forall e \in E : {}^\bullet e \cap e^\bullet = \emptyset$



reti non semplici



rete non pura

eventi indipendenti, eventi concorrenti

Sia $N = (B, E, F)$ una rete elementare, $U \subseteq E$ e $c, c_1, c_2 \subseteq B$.

- U è un **insieme di eventi indipendenti** sse

$$\forall e_1, e_2 \in U : \; e_1 \neq e_2 \;\Rightarrow\; ({}^\bullet e_1 \cup e_1{}^\bullet) \cap ({}^\bullet e_2 \cup e_2{}^\bullet) = \emptyset$$

- U è un **passo abilitato** (insieme di eventi concorrenti) in c $(c[U >)$ sse

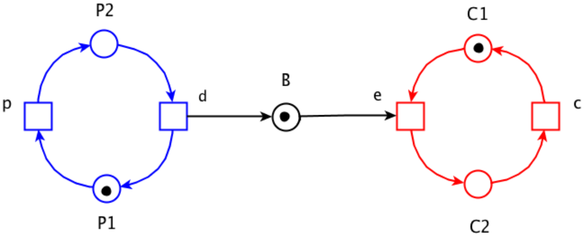
$$U \text{ insieme di eventi indipendenti} \quad \wedge \quad \forall e \in U : c[e >$$

- U è un **passo da** c_1 **a** c_2, $(c_1[U > c_2)$ sse

$$c_1[U > \quad \wedge \quad c_2 = (c_1 - {}^\bullet U) \cup U^\bullet$$

Esempio

Dato il sistema Σ che modella il produttore e consumatore



- $\{p, e\}$, $\{p, c\}$, $\{d, c\}$ sono esempi di insiemi di eventi indipendenti
- $\{p, e\}$ è un passo abilitato in $\{P_1, B, C_1\}$
- $\{P_1, B, C_1\}\ [\{p, e\} > \{P_2, C_2\}$

Sistemi elementari

Un **sistema elementare** $\Sigma = (B, E, F; c_{in})$ è definito da una *rete* $N = (B, E, F)$ e da $c_{in} \subseteq B$ un *caso iniziale*.

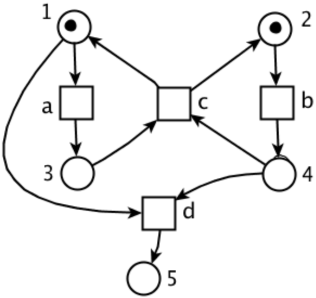
L'insieme dei **casi raggiungibili** (C_Σ) del sistema elementare $\Sigma = (B, E, F; c_{in})$ è il più piccolo sottoinsieme di 2^B tale che:

- $c_{in} \in C_\Sigma$
- se $c \in C_\Sigma, U \subseteq E, c' \subseteq B$ sono tali che: $c[U > c'$, allora $c' \in C_\Sigma$

U_Σ è l'**insieme dei passi** di Σ: $U_\Sigma = \{U \subseteq E \mid \exists c, c' \in C_\Sigma : \; c[U > c'\}$

Esercizio

Dato il seguente sistema elementare Σ



- identificare l'insieme dei casi raggiungibili C_Σ;
- identificare l'insieme dei passi di Σ, U_Σ.

Il comportamento dei sistemi elementari

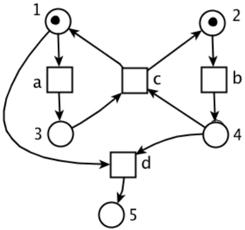
Sia $\Sigma = (B, E, F; c_{in})$ un sistema elementare, $c_i \in C_\Sigma$, $e_i \in E$, $U_i \subseteq E$

- comportamento sequenziale - sequenze di occorrenze / di eventi ("interleaving", simulazione sequenziale non deterministica)
  $c_{in}[e_1 > c_1[e_2 > \ldots [e_n > c_n$ oppure $c_{in}[e_1 e_2 \ldots e_n > c_n$
- comportamento non sequenziale - sequenze di passi ("step semantics")
  $c_{in}[U_1 > c_1[U_2 > \ldots [U_n > c_n$ oppure $c_{in}[U_1 U_2 \ldots U_n > c_n$
- comportamento non sequenziale - processi non sequenziali ("partial order semantics" - "true concurrency")
  ...

Si considerano sia sequenze finite che sequenze infinite (di eventi o di passi)

Esempio

Dato il seguente sistema elementare Σ



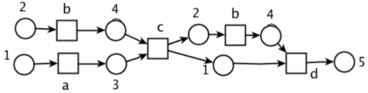
- una possibile sequenza di occorrenze di eventi:

$\{1,2\}[a > \{3,2\}[b > \{3,4\}[c > \{1,2\}[b > \{1,4\}[d > \{5\};$

- una possibile sequenza di passi:

$\{1,2\}[\{a,b\} > \{3,4\}[\{c\} > \{1,2\}[\{b\} > \{1,4\}.$

- un processo non sequenziale di Σ:

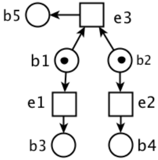


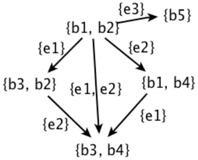
grafo dei casi raggiungibili

Il comportamento di un sistema elementare $\Sigma = (B, E, F; c_{in})$ può essere rappresentato dal suo grafo dei casi.

Il **grafo dei casi** di Σ è il *sistema di transizioni etichettato* $CG_\Sigma = (C_\Sigma, U_\Sigma, A, c_{in})$ dove:

- C_Σ è l'insieme dei nodi del grafo (gli stati globali),
- U_Σ è l'alfabeto,
- A è l'insieme di archi etichettati:
  $A = \{(c, U, c') \mid c, c' \in C_\Sigma,\ U \in U_\Sigma,\ c[U > c'\}$,

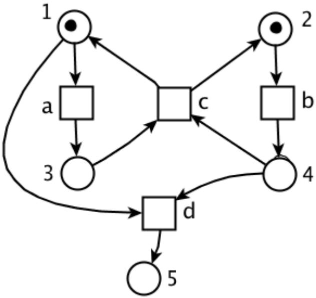




un sistema Σ e il suo grafo dei casi CG_Σ

Esercizio

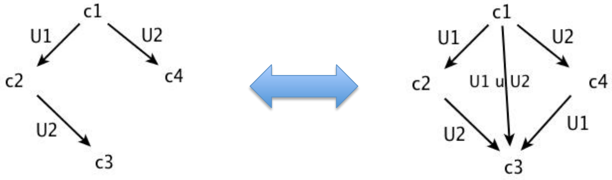
Dato il sistema elementare Σ



- costruire il grafo dei casi C_Σ.

dyamond property

Sia $\Sigma = (B, E, F; c_{in})$ un sistema elementare, $CG_\Sigma = (C_\Sigma, U_\Sigma, A, c_{in})$ il suo grafi dei casi, $U_1, U_2 \in U_\Sigma$: $U_1 \cap U_2 = \emptyset$, $U_1 \neq \emptyset$, $U_2 \neq \emptyset$, e $c_1, c_2, c_3, c_4 \in C_\Sigma$, allora vale:

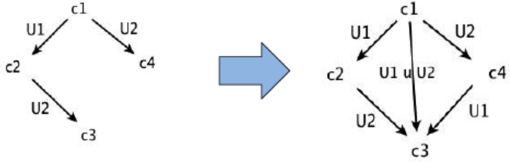


dyamond property

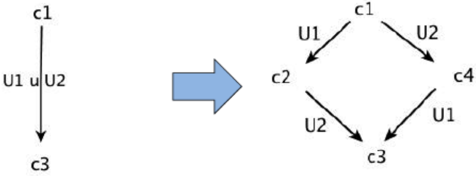
PIU' PRECISAMENTE:

Sia $\Sigma = (B, E, F; c_{in})$ un sistema elementare, $CG_\Sigma = (C_\Sigma, U_\Sigma, A, c_{in})$ il suo grafi dei casi, $U_1, U_2 \in U_\Sigma$: $U_1 \cap U_2 = \emptyset$, $U_1 \neq \emptyset$, $U_2 \neq \emptyset$, e $c_1, c_2, c_3, c_4 \in C_\Sigma$, allora vale:

1)



2)



dyamond property (prova 1)

Consideriamo il caso in cui U_i è un singolo evento e_i, $i = 1, 2$.

Siano $c_1, c_2 \in C_\Sigma$ e $e_1, e_2 \in E$ tali che: $c_1[e_1 > c_2[e_2 >$ e $c_1[e_2 >$,

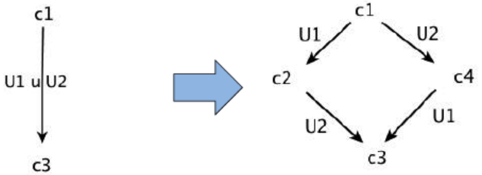
dobbiamo allora dimostrare che: $({}^\bullet e_1 \cup e_1{}^\bullet) \cap ({}^\bullet e_2 \cup e_2{}^\bullet) = \emptyset$.

Da $c_1[e_1 >$ e $c_1[e_2 >$ segue che: ${}^\bullet e_1 \cap e_2{}^\bullet = \emptyset$ e ${}^\bullet e_2 \cap e_1{}^\bullet = \emptyset$,
infatti se e_1 e e_2 sono entrambi abilitati in c_1, le loro pre-condizioni sono vere e le post-condizioni false, e quindi non è possibile che una condizione sia contemporaneamente pre-condizione di e_1 (vera) e anche post-condizione di e_2 (falsa), e viceversa.

Da $c_1[e_1 > c_2[e_2 >$ segue: ${}^\bullet e_1 \cap {}^\bullet e_2 = \emptyset$ e $e_1{}^\bullet \cap e_2{}^\bullet = \emptyset$,
in c_2, infatti, le pre-condizioni di e_1 sono false mentre le pre-condizioni di e_2 sono vere e quindi e_1 e e_2 non possono avere pre-condizioni in comune; inoltre sempre in c_2 le post-condizioni di e_1 sono vere, mentre quelle di e_2 sono false, e quindi e_1 e e_2 non possono avere post-condizioni in comune.

La tesi segue quindi immediatamente.

dyamond property (prova 2)



Supponiamo che $U_1 \cup U_2 \in U_\Sigma$ e che $U_1 \cap U_2 = \emptyset$, $U_1 \neq \emptyset$, $U_2 \neq \emptyset$.

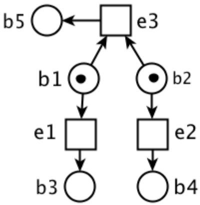
Allora se $c_1[U_1 \cup U_2 > c_3$ sicuramente $c_1[U_1 >$ e $c_1[U_2 >$

e anche: $c_1[U_1 > c_2[U_2 > c_3$ e $c_1[U_2 > c_4[U_1 > c_3$

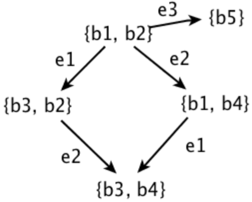
La tesi segue quindi immediatamente.

grafo dei casi sequenziale

Il **grafo dei casi sequenziale** di $\Sigma = (B, E, F; c_{in})$ è
$SCG_\Sigma = (C_\Sigma, E, A, c_{in})$ dove:
$A = \{(c, e, c') \mid c, c' \in C_\Sigma,\ \ e \in E:\ \ c[e > c'\}$



Σ



SCG_Σ

Per la "diamond property", nei sistemi elementari il grafo dei casi e il grafo dei casi sequenziale sono "sintatticamente equivalenti" (possono essere ricavati l'uno dall'altro).

Questo implica il fatto che due sistemi elementari hanno grafi dei casi isomorfi se e solo se hanno grafi dei casi sequenziale isomorfi.

Isomorfismo tra Sistemi di Transizioni Etichettati

Siano $A_1 = (S_1, E_1, T_1, s_{01})$ e $A_2 = (S_2, E_2, T_2, s_{02})$ sistemi di transizioni etichettati, e
$\alpha : S_1 \rightarrow S_2$ e $\beta : E_1 \rightarrow E_2$ siano mappe **biunivoche**, allora

$$< \alpha, \beta > : \ A_1 = (S_1, E_1, T_1, s_{01}) \ \rightarrow \ A_2 = (S_2, E_2, T_2, s_{02})$$

è un **isomorfismo** se e solo se:

- $\alpha(s_{01}) = s_{02}$
- $\forall s, s' \in S_1, \forall e \in E_1 : \ (s, e, s') \in T_1 \Longleftrightarrow (\alpha(s), \beta(e), \alpha(s')) \in T_2$

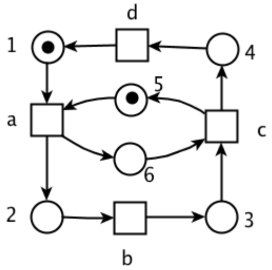
equivalenza basata sui grafi dei casi

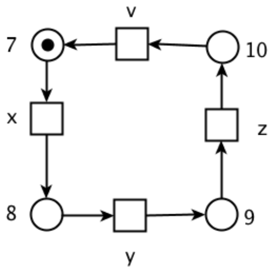
Due sistemi Σ_1 e Σ_2 sono **equivalenti** se e solo se hanno grafi dei casi sequenziali (e quindi anche grafi dei casi) **isomorfi**.

Esercizio

Dati i seguenti sistemi elementari

Σ_1 e Σ_2





verificare se hanno grafo dei casi isomorfi.

il problema della sintesi

Dato un sistema di transizioni etichettato $A = (S, \mathbf{E}, T, s_0)$,

stabilire se esiste un sistema elementare $\Sigma = (B, \mathbf{E}, F; c_{in})$ tale che:

il suo grafo dei casi SCG_Σ sia **isomorfo** ad A.

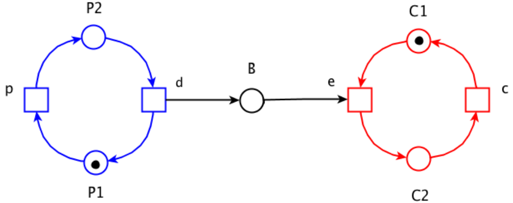
e, in caso affermativo, costruire Σ.

Il problema è stato risolto $\rightarrow$ Teoria delle regioni

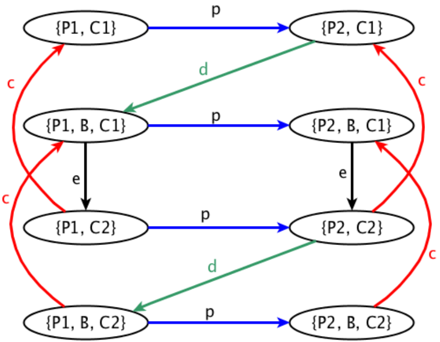
(A dovrà tra l'altro soddisfare la dyamond property)

Esempio: il produttore e il consumatore

un **sistema elementare**



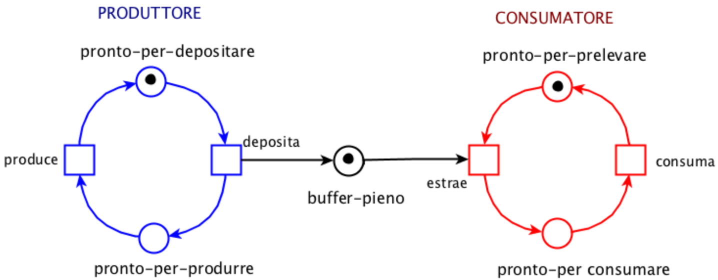
il suo **grafo dei casi sequenziale**



contatto

Sia $\Sigma = (B, E, F; c_{in})$ un sistema elementare, $e \in E$, $c \in C_\Sigma$;
(e, c) è un **contatto** sse:

$$^\bullet e \subseteq c \;\wedge\; e^\bullet \cap c \neq \emptyset$$



sistemi elementari senza contatti

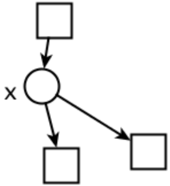
Un sistema elementare $\Sigma = (B, E, F; c_{in})$ è **senza contatti** sse

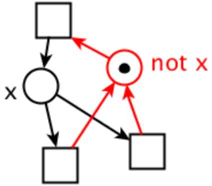
$$\forall e \in E, \forall c \in C_\Sigma \quad {}^\bullet e \subseteq c \Rightarrow e^\bullet \cap c = \emptyset$$

E' possibile trasformare un sistema elementare Σ con contatti in un sistema elementare Σ' che sia senza contatti, senza modificarne il comportamento **?**

SI': aggiungendo a Σ il **complemento** di ogni condizione si ottiene un sistema Σ' con grafo dei casi isomorfo a quello di Σ.

CG_Σ isomorfo a $CG_{\Sigma'}$





sistemi elementari senza contatti (regola di scatto)

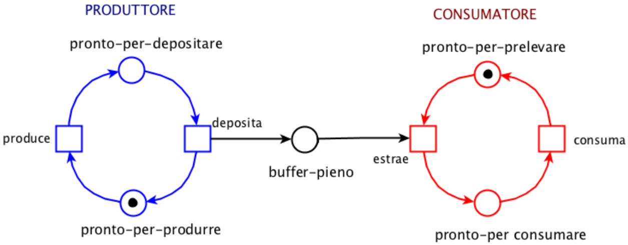
Un sistema elementare $\Sigma = (B, E, F; c_{in})$ è **senza contatti** sse

$$\forall e \in E, \forall c \in C_\Sigma \quad {}^\bullet e \subseteq c \Rightarrow e^\bullet \cap c = \emptyset$$

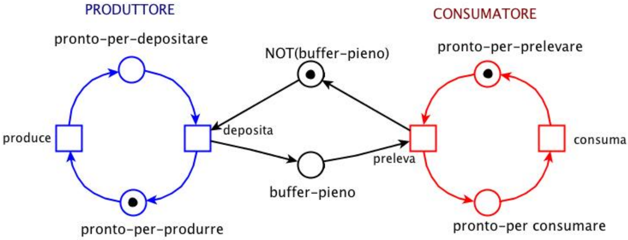
Quindi se un sistema elementare Σ è *senza contatti* allora per verificare che un evento e sia *abilitato* in un caso raggiungibile c è sufficiente verificare che le precondizioni di e siano vere:

$$c[e > \quad \text{sse} \quad {}^\bullet e \subseteq c$$

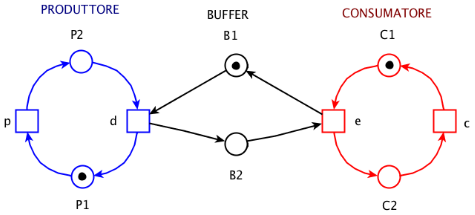
il complemento di una condizione

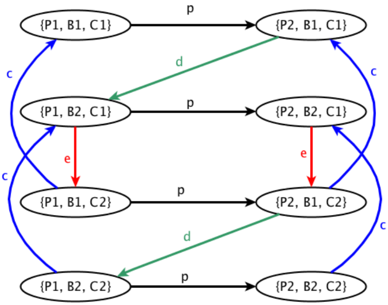


aggiungendo la condizione complemento di *buffer-pieno*, *NOT(buffer-pieno)*, si ottiene



il sistema "produttore e consumatore" senza contatti e il suo grafo dei casi sequenziale



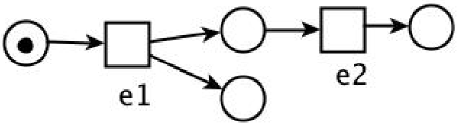


situazioni fondamentali: sequenza

Sia $\Sigma = (B, E, F; c_{in})$ un sistema elementare, $c \in C_\Sigma$, $e_1.e_2 \in E$

- e_1 ed e_2 sono in **sequenza** in c sse

$$c[e_1 > \ \wedge \ \neg c[e_2 > \ \wedge \ c[e_1 e_2 > \ \ (c[e_1 > c'[e_2 >)$$

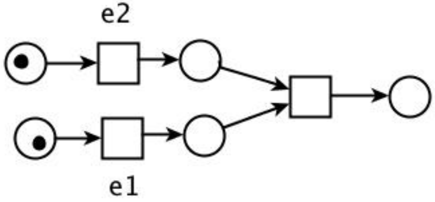


(c'è una relazione di **dipendenza causale** tra e_1 ed e_2)

situazioni fondamentali: concorrenza

Sia $\Sigma = (B, E, F; c_{in})$ un sistema elementare, $c \in C_\Sigma$, $e_1, e_2 \in E$

- e_1 e e_2 sono **concorrenti** in c sse $c[\{e_1, e_2\} >$

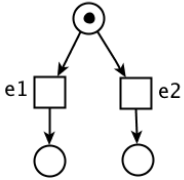


(in modo equivalente, sse
e_1 e e_2 sono *indipendenti* ed entrambi *abilitati* in c)

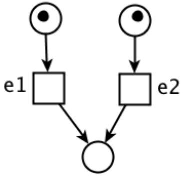
situazioni fondamentali: conflitto

Sia $\Sigma = (B, E, F; c_{in})$ un sistema elementare, $c \in C_\Sigma$, $e_1 . e_2 \in E$

- e_1 e e_2 sono in **conflitto** in c sse $c[e_1 > \ \wedge \ c[e_2 > \ \wedge \ \neg c[\{e_1, e_2\} >$ (sono entrambi abilitati ma l'occorrenza di uno disabilita l'altro)



in avanti: "forward"



all'indietro: "backward"

non è specificato se scatterà e_1 o e_2

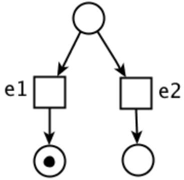
non determinismo

situazioni fondamentali: conflitto

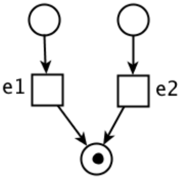
Sia $\Sigma = (B, E, F; c_{in})$ un sistema elementare, $c \in C_\Sigma$, $e_1.e_2 \in E$

- e_1 e e_2 sono in **conflitto** in c sse $c[e_1 > \ \wedge \ c[e_2 > \ \wedge \ \neg c[\{e_1, e_2\} >$ (sono entrambi abilitati ma l'occorrenza di uno disabilita l'altro)

consideriamo adesso queste due configurazioni



in avanti: "forward"



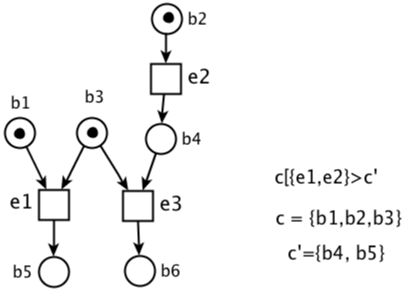
all'indietro: "backward"

cosa è successo?

l'ambiente fornisce/riceve 'informazione' sulla soluzione del conflitto

situazioni fondamentali: confusione1

una 'mistura' di concorrenza e conflitto



nell'esecuzione di $c[\{e_1, e_2\} > c'$ è stato risolto un conflitto?

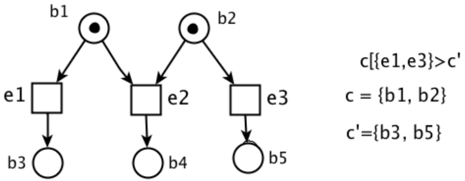
Ci sono due possibilità, entrambe amissibili:
1) occorre prima e_1 senza essere in conflitto
2) occorre prima e_2 e poi il conflitto tra e_1 e e_3 viene risolto a favore di e_1

$\Rightarrow$ CONFUSIONE

non è possibile stabilire oggettivamente se è stato risolto un conflitto

situazioni fondamentali: confusione2

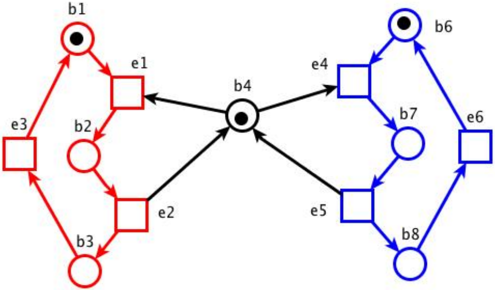
un altro esempio di confusione



nell'esecuzione di $c[\{e_1, e_3\} > c'$ non è possibile stabilire se è stato risolto un conflitto tra e_1 ed e_2 oppure tra e_3 ed e_2.

Chi ha deciso? Chi ha la responsabilità di decidere?
In situazioni di confusione non è chiaramente specificato questo aspetto

esempio: mutua esclusione



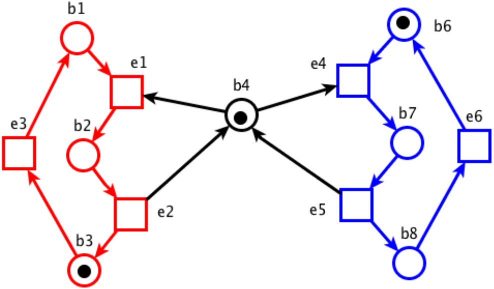
b_4 risorsa libera
b_2, b_7 risorsa in uso
e_1, e_4 acquisizione della risorsa
e_2, e_5 rilascio della risorsa

in nessun caso raggiungibile le condizioni b_2 e b_7 sono entrambe marcate.

e_1 e e_4 sono in **conflitto**

Esercizio: Calcolare il grafo dei casi raggiungibili

esempio: mutua esclusione



b_4 risorsa libera
b_2, b_7 risorsa in uso
e_1, e_4 acquisizione della risorsa
e_2, e_5 rilascio della risorsa

in nessun caso raggiungibile le condizioni b_2 e b_7 sono entrambe marcate.

(e_1 e e_4 sono in **conflitto** in $\{b_1, b_4, b_6\}$)

$\{b_3, b_4, b_6\}[\{e_3, e_4\} > \{b_1, b_7\}$ è stato risolto il conflitto ?

confusione

Sottorete

Siano $N = (B, E, F)$ e $N_1 = (B_1, E_1, F_1)$ due reti elementari

- $N_1 = (B_1, E_1, F_1)$ è **sottorete** di $N = (B, E, F)$ sse:

- $B_1 \subseteq B$;
- $E_1 \subseteq E$;
- $F_1 = F \cap [(B_1 \times E_1) \cup (E_1 \times B_1)]$.

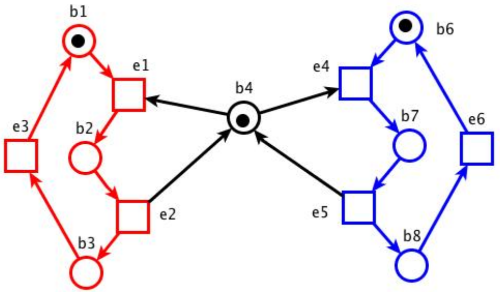
- $N_1 = (B_1, E_1, F_1)$ è la **sottorete generata da** B_1 sse:

- $B_1 \subseteq B$;
- $E_1 = {}^\bullet B_1 \cup B_1{}^\bullet$;
- $F_1 = F \cap [(B_1 \times E_1) \cup (E_1 \times B_1)]$.

- $N_1 = (B_1, E_1, F_1)$ è la **sottorete generata da** E_1 sse:

- $B_1 = {}^\bullet E_1 \cup E_1{}^\bullet$;
- $E_1 \subseteq E$;
- $F_1 = F \cap [(B_1 \times E_1) \cup (E_1 \times B_1)]$.

esempio: sottorete della mutua esclusione



- $N' = (\{b_1, b_2, b_3\}, \{e_1, e_2, e_3\}, F')$ è la sottorete generata dal sottoinsieme di condizioni $B' = \{b_1, b_2, b_3\}$;
- $N' = (\{b_6, b_7, b_8\}, \{e_4, e_5, e_6\}, F')$ è la sottorete generata dal sottoinsieme di condizioni $B' = \{b_6, b_7, b_8\}$;
- $N' = (\{b_2, b_4, b_7\}, \{e_1, e_2, e_4, e_5\}, F')$ è la sottorete generata dal sottoinsieme di condizioni $B' = \{b_2, b_4, b_7\}$;
- $N' = (\{b_1, b_2, b_3, b_4\}, \{e_1, e_2, e_3\}, F')$ è la sottorete generata dal sottoinsieme di eventi $E' = \{e_1, e_2, e_3\}$.

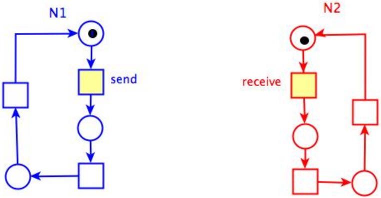
Operazioni di composizione per reti di Petri

$$N = (B, E, F, c_0)$$

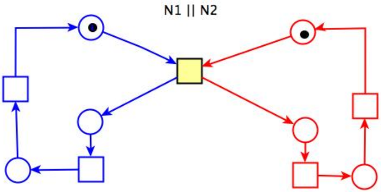
principali metodi di composizione

- sincrona
- asincrona
- mista

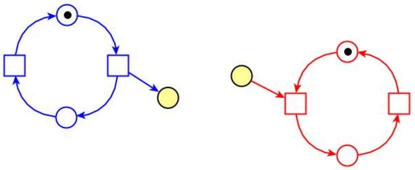
Composizione sincrona ||



identificazione di transizioni (sincronizzazione)



Composizione Asincrona



identificazione di posti (canali di comunicazione)