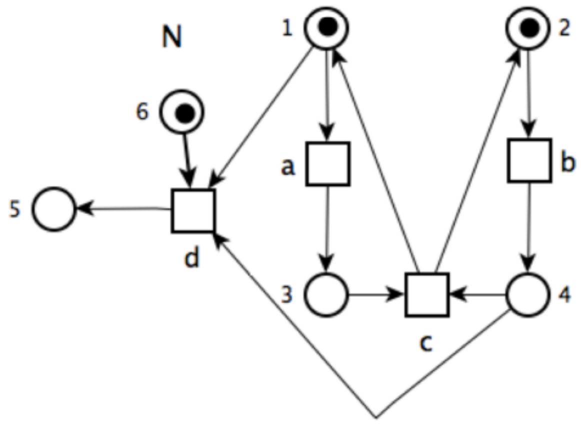
Lezione 11 09/11/2023

Esercizi sui processi

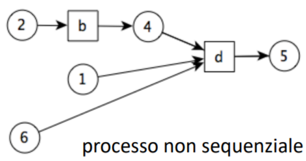
Esercizio 1

Dato il sistema



1) Disegnare un processo non sequenziale del sistema in figura che contenga almeno due occorrenze dell'evento b e una dell'evento d .
Indicate una linea e un taglio che contengano entrambi un'occorrenza dell'evento b .

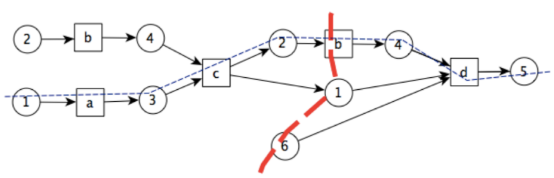
2) Disegnare un processo ramificato che contenga due occorrenze dell'evento d .



processo non sequenziale che non soddisfa la specifica: contiene solo un'occorrenza dell'evento b .

Perché volevamo due occorrenza dell'evento b.

Quindi è un processo non sequenziale.

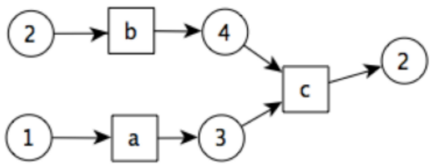


Questa è una possibile soluzione

Prima ho a e b in maniera concorrente, poi, e poi di nuovo ho 2-1-6, ho quindi 2 occorrenze di b come richiesto, e un'occorrenza di d.

Una linea è per esempio quella tratteggiata blu, ma ce ne sono altre come 6-d-5, oppure 2-b-4-c-1-d-5.

Nei tagli devo prendere le condizioni in relazione co, cioè per esempio 6-1-4. Però dobbiamo prenderne uno che contiene b. Devo però appunto prendere anche 6 perchè è in relazione co (indipendente da tutto tranne che d)



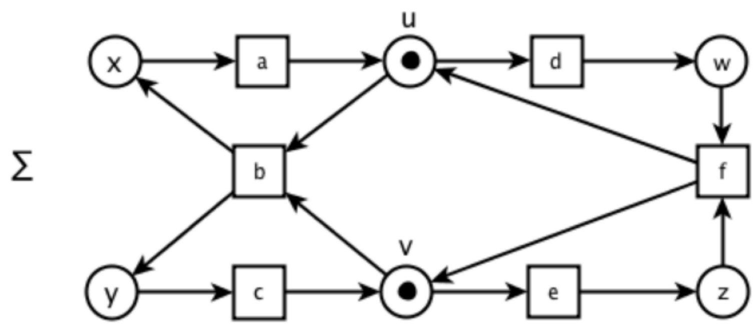
Non è un processo non sequenziale ....

Perché manca la condizione 6 che anche se isolata va registrata perché vera nel caso iniziale.

Manca anche 1 che è la post di c. Devo registrare tutte le pre e post condizioni che abilitano gli eventi.

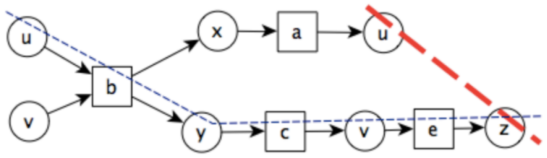
Esercizio 2

Dato il sistema



Disegnare un processo non sequenziale del sistema contenente almeno una occorrenza di z e di c. Indicare una linea e un taglio che contengano entrambi un'occorrenza della condizione z.

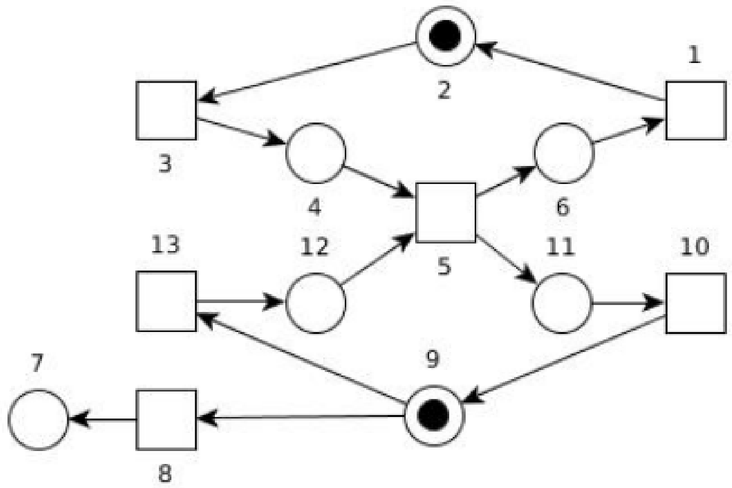
una possibile soluzione



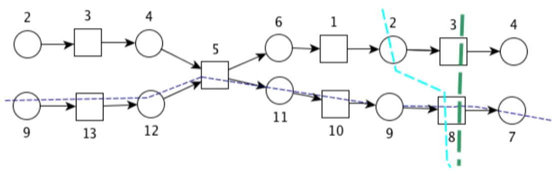
Registro prima b, poi ho a e c in maniera indipendente, poi v e u avendo poi il taglio su u-z.

La linea è quindi un'insieme massimale di eventi allineati.

Esercizio 3



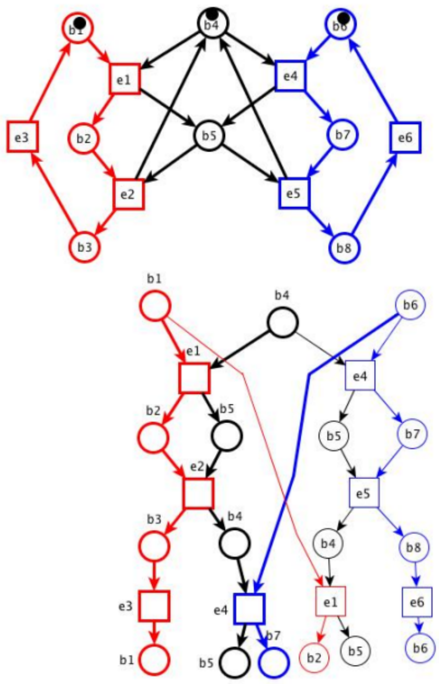
Disegnate un processo non sequenziale che contenga almeno un'occorrenza di ciascun evento; indicate una linea e due tagli contenenti un'occorrenza dell'evento 8.



8 deve scattare dopo altrimenti blockerei il sistema.

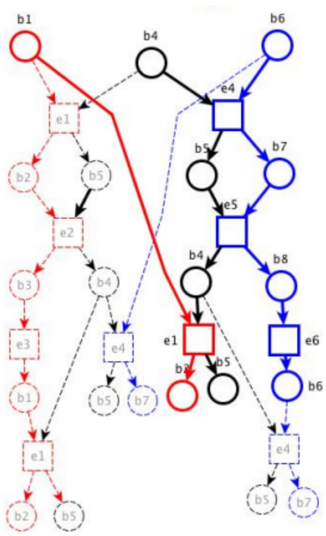
Processi non sequenziali (continuo)

Processo ramificato del sistema "mutua esclusione"



Per esempio può esser eseguito e1 oppure e4.

In questo modello ho che se vado a prendere un insieme di condizioni e vado a prendere il passato, questo è già determinato. Invece quello che succede dopo è ancora aperto, per esempio quando sono in b3-b4-b6 ho segnato e4 ma avrei potuto avere e1. Ho quindi più scelte possibili sul futuro.



Se mi fermo in b1-b4-b6 potrei andare avanti con e1 (come evidenziato) ma potrei andare avanti anche con e4. Quindi se prendo un taglio il passato è determinato ma il futuro non ancora.

Nella figura più processi sono modellati da un’unica rete con conflitti in avanti.

Posso quindi avere conflitti in avanti, ma non all’indietro perché sono già state registrate le scelte.

Dobbiamo quindi definire queste reti senza cicli e con conflitti in avanti.

Reti di occorrenze

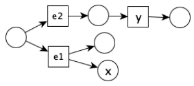
Registrano i processi ramificati, ovvero hanno l’informazione di diversi possibili comportamenti.

$N = (B, E, F)$ è una **rete di occorrenze** sse

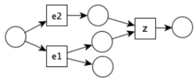
(i) $\forall b \in B : |{}^\bullet b| \leq 1$: CONFLITTI solo in avanti
(ii) $\forall x, y \in B \cup E : (x, y) \in F^+ \Rightarrow (y, x) \notin F^+$: no cicli
(iii) $\forall e \in E : \{x \in B \cup E \mid xF^*e\}$ è finito
(iv) la relazione di *conflitto* $\#$ **non** è riflessiva

Dove la relazione $\# \subseteq X \times X$ $(X = B \cup E)$ è definita come segue:

$$x \# y \text{ sse } \exists e_1, e_2 \in E : {}^\bullet e_1 \cap {}^\bullet e_2 \neq \emptyset \wedge e_1 \leq x \wedge e_2 \leq y.$$



rete di occorrenze



non è una rete di occorrenze
$z \# z$

É ancora possibile associare ad N un ordine parziale $(X, \leq) = (B \cup E, F^*)$

(dopo quello scritto a destra) Per esempio x e y nell'immagine sono in relazione di conflitto. Z però arriva dopo sia e1 che e2, quindi sarebbe come dire che z è in conflitto con se stessa. Z non sarà mai abilitato perché viene eseguita e1 oppure e2, la rete quindi non è una rete di occorrenze.

Quindi la relazione di conflitto non è riflessiva.

Anche qui possiamo avere un ordine parziale tra due eventi e/o condizioni.

(i) Ogni condizione ha al più solo un pre evento (nelle reti causali avevano al più un pre evento e un post evento). Può quindi avere più archi uscenti.

(ii) Non abbiamo cicli

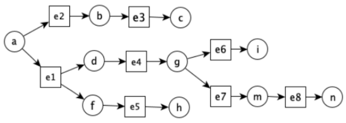
(iii) Siccome queste reti registrano comportamenti effettivi, se registro un evento il passato di questo evento sarà finito, mentre il futuro può essere infinito.

(iv) Siccome abbiamo i conflitti, abbiamo # che è la relazione di conflitto. Prima avevamo per esempio la relazione li che definiva l'esistenza di una sequenza di archi tra x e y, oppure co che definiva l'indipendenza tra x e y, ora abbiamo la relazione di conflitto che è definita come x e y sono in conflitto se esistono due eventi e1 e2 che hanno almeno una pre condizione in comune, con x successivo a e1 e y successivo a e2 (o viceversa).

Quindi x e y sono nel futuro di due eventi che sono in conflitto.

reti di occorrenze, ordini parziali e relazioni ($\#$, **li** e **co**)

$N = (B, E, F)$ rete di occorrenze, $(X, \leq)$ ordine parziale associato, $x, y \in X$
- x **li** y sse $x \leq y$ or $y \leq x$
- $x \# y$ sse (si veda la pag precedente)
- x **co** y sse **not**$(x < y)$ and **not**$(y < x)$ and **not**$(y \# x)$



- esempi di elementi in relazione di conflitto $\#$:
  $e_1 \# e_2$; $e_3 \# e_4$; $c \# e_4$; $c \# h$; $e_6 \# e_8$; ...

conflitto $\#$ è simmetrica, **non** è transitiva

- esempi di elementi in relazione **li**:
  e_1 **li** e_7; e_1 **li** i; a **li** e_3; ...

- esempi di elementi in relazione **co**:
  e_4 **co** e_5; e_5 **co** e_6; e_5 **co** e_7; h **co** e_8; ...

Rimangono le relazioni li e co anche nelle reti di occorrenze.

La relazione di conflitto è simmetrica ma non transitiva. Per esempio e3 è in conflitto con e4, c è in conflitto con e4, ma c ed e3 non sono in conflitto.

Processi ramificati

Sia $\Sigma = (S, T, F, c_{in})$ un sistema elementare **senza contatti** e **finito**.

$< N = (B, E, F); \phi >$ è un processo ramificato di Σ sse
- (B, E, F) è una **rete di occorrenze** (si ammettono condizioni isolate)
- $\phi : B \cup E \to S \cup T$ è una mappa :
  - *(1)* $\phi(B) \subseteq S$, $\phi(E) \subseteq T$
  - *(2')* $\forall e_1, e_2 \in E$: $({}^\bullet e_1 = {}^\bullet e_2 \wedge \phi(e_1) = \phi(e_2)) \Rightarrow e_1 = e_2$
  - *(3')* $\forall e \in E$: la restrizione di ϕ a ${}^\bullet e$ è una biiezione tra ${}^\bullet e$ e ${}^\bullet \phi(e)$ e
    la restrizione di ϕ a $e^\bullet$ è una biiezione tra $e^\bullet$ e $\phi(e)^\bullet$
  - *(4')* la restrizione di ϕ a $Min(N)$ è una biiezione tra $Min(N)$ e c_{in}.

Dobbiamo modificare le condizioni della mappa.

(2') Se ho due eventi che hanno esattamente le stesse pre, e corrispondono alla registrazione dello stesso evento nel sistema, allora devono essere lo stesso evento. Altrimenti avresti e1 con le stesse condizioni di e2, messo come alternativa, ma questo non può essere se e1=e2.

Per esempio nell'esempio precedente (quello blu rosso) i due e4 (in alto a destra e in basso a sinistra) sono mappati sullo stesso evento, ma hanno pre diverse (una è la stessa ma non tutte).

(3') devo registrare tutte le pre e tutte le post di un evento.

processi non sequenziali vs ramificati

$\Sigma = (S, T, F, c_{in})$ sistema elementare **senza contatti** e **finito**.

$< N = (B, E, F); \phi >$ è un processo non sequenziale di Σ sse

* (B, E, F) è una **rete causale** (si ammettono condizioni isolate)

* $\phi : B \cup E \rightarrow S \cup T$ è una mappa :

(1) $\phi(B) \subseteq S$, $\phi(E) \subseteq T$

(2) $\forall x_1, x_2 \in B \cup E : \ \phi(x_1) = \phi(x_2) \ \Rightarrow \ (x_1 \leq x_2)$ or $(x_2 \leq x_1)$

(3) $\forall e \in E : \ \phi(^\bullet e) = {}^\bullet\phi(e)$ and $\phi(e^\bullet) = \phi(e)^\bullet$

(4) $\phi(Min(N)) = c_{in}$.

$< N = (B, E, F); \phi >$ è un processo ramificato di Σ sse

* (B, E, F) è una **rete di occorrenze** (si ammettono condizioni isolate)

* $\phi : B \cup E \rightarrow S \cup T$ è una mappa :

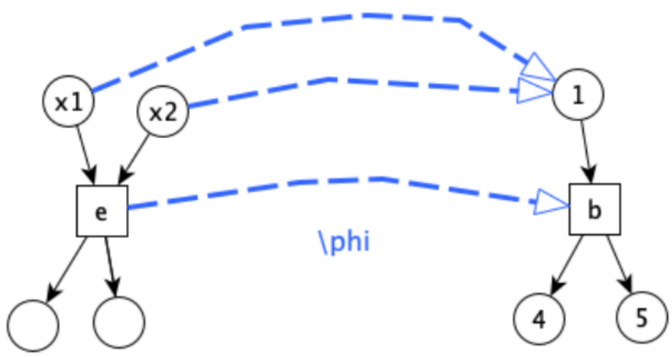
(1) $\phi(B) \subseteq S$, $\phi(E) \subseteq T$

(2') $\forall e_1, e_2 \in E : \ ({}^\bullet e_1 = {}^\bullet e_2 \wedge \phi(e_1) = \phi(e_2) \ ) \ \Rightarrow \ e_1 = e_2$

(3') $\forall e \in E :$ la restrizione di ϕ a ${}^\bullet e$ è una biiezione tra ${}^\bullet e$ e ${}^\bullet\phi(e)$ e
la restrizione di ϕ a $e^\bullet$ è una biiezione tra $e^\bullet$ e $\phi(e)^\bullet$

(4') la restrizione di ϕ a $Min(N)$ è una biiezione tra $Min(N)$ e c_{in}.

$(2) \wedge (3) \Rightarrow (3')$
$(2) \wedge (4) \Rightarrow (4')$



questa ϕ soddisfa (3): $\phi(^\bullet e) = {}^\bullet\phi(e)$

ma **non** (2): $\phi(x_1) = \phi(x_2)$ ma x_1 **co** x_2

e **non** (3'): non c'è corrispondenza biunivoca tra ${}^\bullet e = \{x_1, x_2\}$ e ${}^\bullet(\phi(e)) = \{1\}$

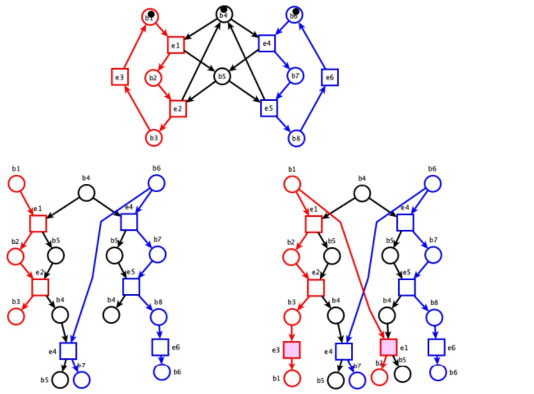
Un processo non sequenziale può essere definito anche richiedendo che ϕ soddisfi: $(1), (2), (3')$ e $(4')$

Se sono tutti e due processi ramificati, uno è prefisso dell’altro se uno è sottorete dell’altro.

prefisso

Sia $\Sigma = (S, T, F, c_{in})$ un sistema elementare *finito* e *senza contatti* e siano $\Pi_1 =< N_1; \phi_1 >, \Pi_2 =< N_2; \phi_2 >$ *processi ramificati* di Σ.

$\Pi_1 =< N_1; \phi_1 >$ è un **prefisso** di $\Pi_2 =< N_2; \phi_2 >$ sse N_1 è una *sottorete* di N_2 e $\phi_{2|N_1} = \phi_1$ (ϕ_2 "ristretto" a N_1 è uguale a ϕ_1).



$\Pi_1 =< N_1; \phi_1 >$ **prefisso** di $\Pi_2 =< N_2; \phi_2 >$

Σ ammette un *unico* processo ramificato che è *massimale* rispetto alla relazione di prefisso tra processi.

Tale processo massimale è chiamato *unfolding* di Σ, denotato $Unf(\Sigma)$.

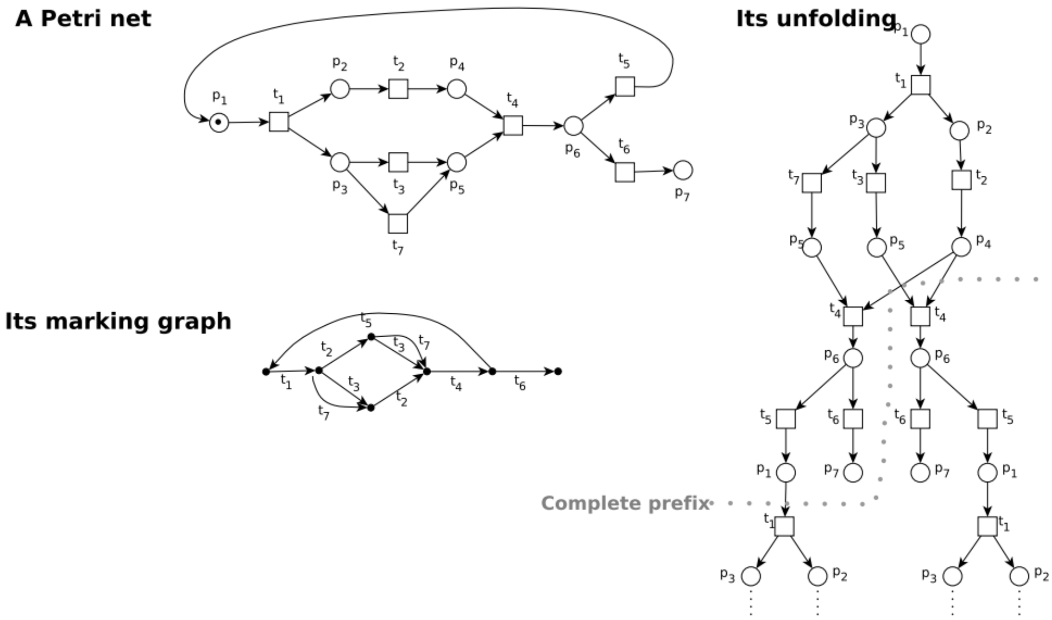
Un processo non sequenziale è un processo ramificato $\Pi =< N; \phi >$ tale che N sia una rete causale (senza conflitti), tale processo è chiamato anche **corsa (run)**.

Inoltre ogni processo non sequenziale di Σ è un prefisso dell’unfolding $Unf(\Sigma)$.

Quindi la corsa è una possibile storia. Il processo ramificato contiene più storie. L’unfolding contiene tutte le storie, e può essere infinito.

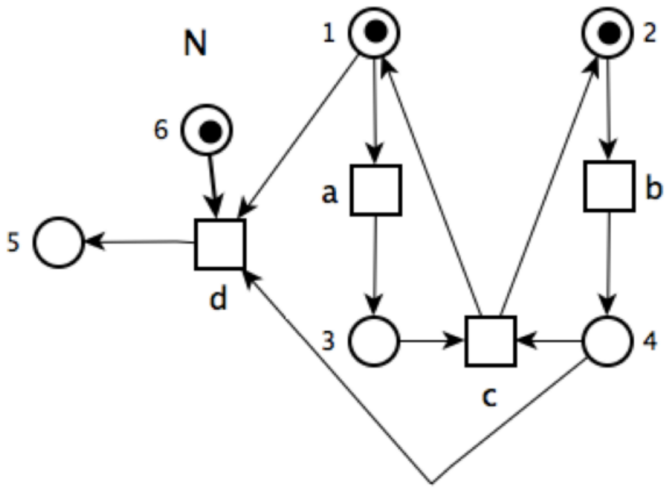
prefisso finito di un unfolding

Petri nets – concurrent and sequential views

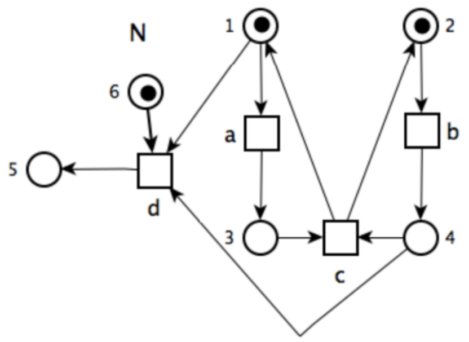


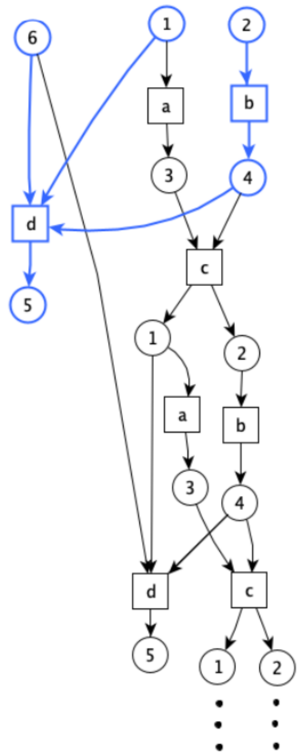
Continuo di un esercizio precedente

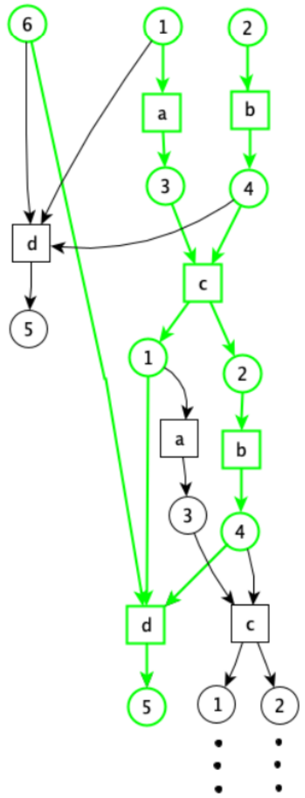
Dato il sistema



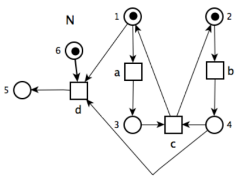
2) Disegnare un processo ramificato che contenga due occorrenze dell'evento d .

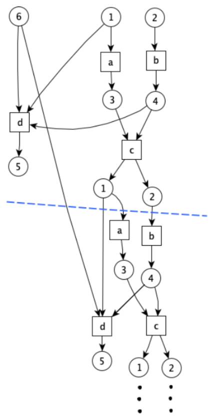




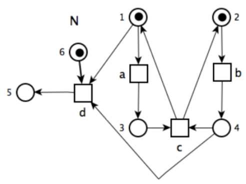


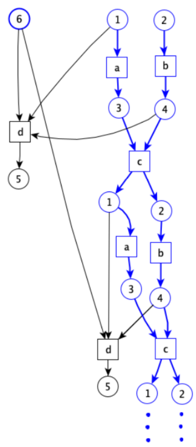
L’unfolding e due prefissi (blu e verde)
che sono anche
processi non sequenziali (finiti)





L’unfolding e un suo prefisso finito





L’unfolding e un prefisso **infinito** (blu)
che è anche
un processo non sequenziale
(notare la condizione isolata 6)