Sintesi di reti di Petri da sistemi di transizione

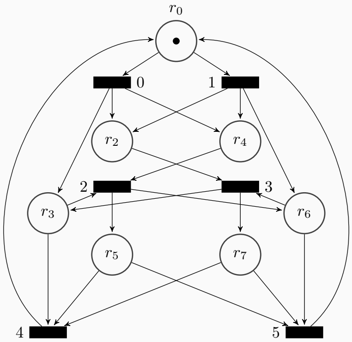
Viktor Teren, Tiziano Villa

Università degli Studi di Verona

Rete di Petri

Una rete di Petri è una sestupla $N = (P, T, A, w, x_0)$ dove

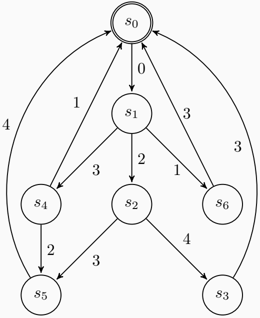
- P è un insieme finito di posti
- T è un insieme finito di transizioni
- A è un insieme di archi, $A \subseteq (P \times T) \cup (T \times P)$
- w è una funzione di peso, $w : A \to \mathbb{N}$
- $\vec{x}_0$ è un vettore della marcatura iniziale, $\vec{x}_0 \in \mathbb{N}^{|P|}$



Sistema di transizione

Un sistema di transizione TS è definito come una quadrupla $TS = (S, E, T, s_0)$ dove

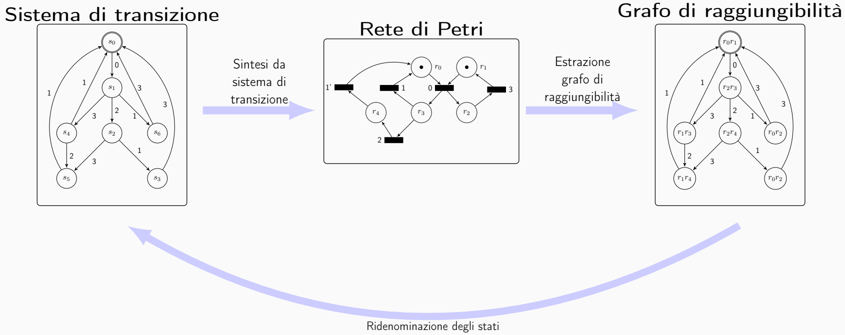
- S è un insieme non vuoto di stati
- E è un insieme di eventi
- $T \subseteq S \times E \times S$
  è una relazione di transizione
- s_0 è lo stato iniziale



Supponiamo che il sistema
di transizione soddisfi le seguenti proprietà:

- Non deve avere autoanelli: $\forall\ (s,e,s') \in TS \neq s'$
- Ogni evento ha almeno un'occorrenza: $\forall e \in E : \exists (s, e, s') \in T$
- Ogni stato è raggiungibile dallo stato iniziale: $\forall s \in S : s_0 \rightarrow^* s$

Flussi di sintesi tra Reti di Petri e sistemi di transizione



Da una rete di Petri limitata si può derivare un sistema di transizione (il suo grafo di raggiungibilità), che descrive il medesimo comportamento (equivalenza comportamentale).

Qui affrontiamo il problema inverso: dato un sistema di transizione, derivare una rete di Petri equivalente.

Motivazioni

Perchè convertire un sistema di transizione in una rete di Petri?

- Modellazione intuitiva di sistemi concorrenti
- Controllo e validazione di sistemi altamente concorrenti grazie alle dimensioni limitate rispetto al sistema di transizione
- ...

Nozioni importanti

Regione

Dato un sistema di transizione $TS = (S, E, T, s_0)$, definiamo una regione come un insieme di stati $r \subseteq S$ tali che valgano le seguenti proprietà per ogni evento $e \in E$:

1. $enter(e, r) \rightarrow \neg in(e, r) \wedge \neg out(e, r) \wedge \neg exit(e, r)$
2. $exit(e, r) \rightarrow \neg in(e, r) \wedge \neg out(e, r) \wedge \neg enter(e, r)$

dove

$$\left.\begin{array}{l} in(e, r) \equiv \exists (s, e, s') \in T : s, s' \in r \\ out(e, r) \equiv \exists (s, e, s') \in T : s, s' \notin r \end{array}\right\} no\_cross$$

$$enter(e, r) \equiv \exists (s, e, s') \in T : s \notin r \wedge s' \in r$$

$$exit(e, r) \equiv \exists (s, e, s') \in T : s \in r \wedge s' \notin r$$

Una regione é pertanto un sottoinsieme di stati per cui tutte le transizioni etichettate con lo stesso evento rispettano la medesima proprietà ($enter/exit/no\_cross$).

Regione

Una regione r è una **pre-regione** di un evento e se esiste una transizione etichettata con e che esce da r. Una regione r è una **post-regione** di un evento e se esiste una transizione etichettata con e che entra in r.

L'insieme di tutte le pre-regioni e post-regioni di e è denotato con ${}^{\circ}e$ ed e° rispettivamente. Per definizione se $r \in {}^{\circ}e$ ($r \in e^{\circ}$) tutte le transizioni etichettate con e escono da r (entrano in r).

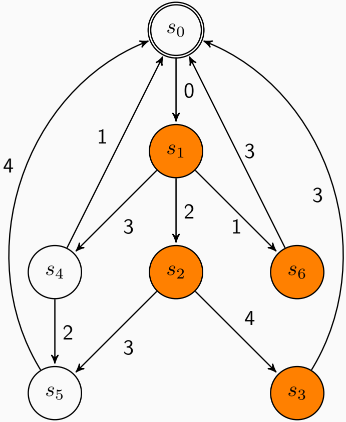
Una regione r' si dice una sottoregione di una regione r se $r' \subset r$.

Una regione r è una regione minimale se nessuna altra regione r' è una sottoregione di r.

Se r e r' sono due regioni con r' sottoregione di r, allora $r - r'$ e' una regione.

Ogni regione si può rappresentare come un'unione di regioni minimali disgiunte.

Esempio: regione

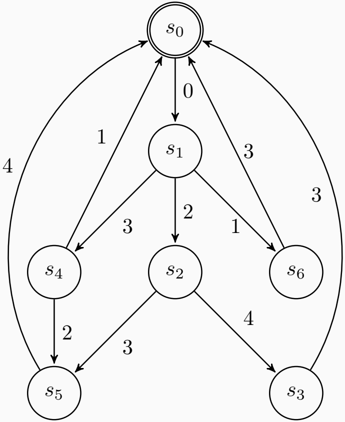


Evento	Proprietà
0	enter
1	no_cross
2	no_cross
3	exit
4	no_cross

$\{s_1, s_2, s_3, s_6\}$ è una regione in quanto tutte le transizioni di ogni evento presentano la medesima proprietà

Regione di eccitazione (ER)

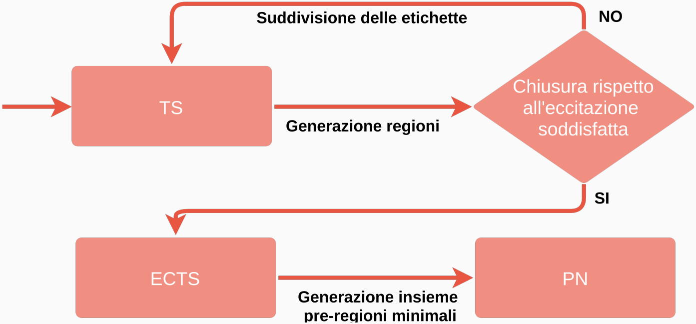
Una **regione di eccitazione** per l'evento e (denotata con ER(e)) è un insieme massimo di stati S tale che per ogni $s \in S$ esiste una transizione $s \xrightarrow{\text{e}}$.



Evento	Regione di eccitazione
0	$\{s_0\}$
1	$\{s_1, s_4\}$
2	$\{s_1, s_4\}$
3	$\{s_1, s_2, s_3, s_6\}$
4	$\{s_2, s_5\}$

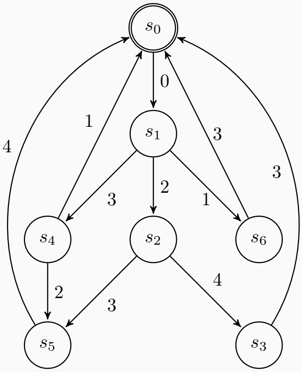
Flusso dell'algoritmo di sintesi

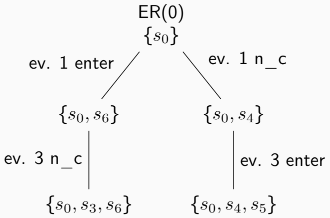
Algoritmo di sintesi



Generazione delle regioni: albero di espansione

A partire dalle regioni di eccitazione per ciascun evento si calcolano gli alberi di espansione attraverso l'aggiunta di nuovi stati, rimuovendo violazioni provocate dagli eventi che impediscono all'insieme di soddisfare tutte le proprietà di una regione.





ER(1)
$\{s_1, s_4\}$

ER(2)
$\{s_1, s_4\}$

ER(3)
$\{s_1, s_2, s_3, s_6\}$

ER(4)
$\{s_2, s_5\}$

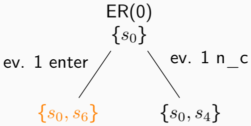
Esempio albero di espansione dell'evento 0

ER(0)
$\{s_0\}$

ER(0) non è una regione dell'evento 1 che presenta sia archi con proprietà *enter* che *no_cross*, di conseguenza l'espansione può essere effettuata in due direzioni distinte:

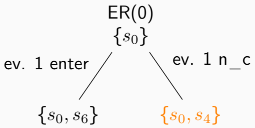
- Aggiungere lo stato s_6 per fare in modo che tutte le etichette abbiano la proprietà *enter*
- Aggiungere lo stato s_4 per fare in modo che tutte le etichette abbiano la proprietà *no_cross*

Esempio albero di espansione dell'evento 0



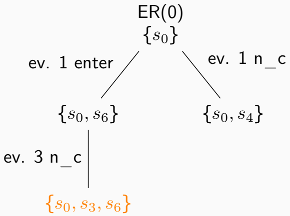
- A causa dell'evento 3 l'insieme $\{s_0, s_6\}$ non è una regione e deve essere espanso ulteriormente.
- A causa dell'evento 3 l'insieme $\{s_0, s_4\}$ non è una regione e deve essere espanso ulteriormente.

Esempio albero di espansione dell'evento 0



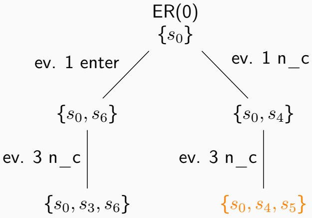
- A causa dell'evento 3 l'insieme $\{s_0, s_6\}$ non è una regione e deve essere espanso ulteriormente.
- A causa dell'evento 3 l'insieme $\{s_0, s_4\}$ non è una regione e deve essere espanso ulteriormente.

Esempio albero di espansione dell'evento 0



L'insieme $\{s_0, s_3, s_6\}$ è una regione.

Esempio albero di espansione dell'evento 0

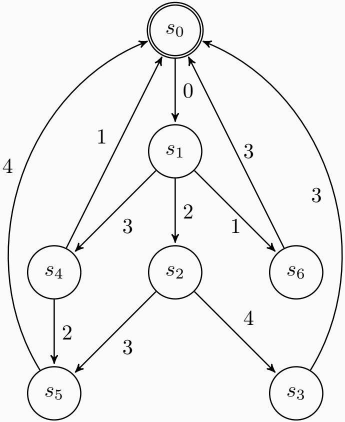


L'insieme $\{s_0, s_4, s_5\}$ è una regione.

La scelta dell'ordine degli eventi in violazione nella costruzione dell'albero di espansione può produrre alberi di dimensioni diverse

Generazione delle regioni

Avendo creato gli alberi di espansione, dalle foglie si ricavano le seguenti regioni:



Regione	Stati della regione
r_0	$\{s_0, s_3, s_6\}$
r_1	$\{s_0, s_4, s_5\}$
r_2	$\{s_1, s_2, s_3, s_6\}$
r_3	$\{s_1, s_4\}$
r_4	$\{s_2, s_5\}$

Generazione insieme di pre-regioni minimali

Tra tutte le regioni consideriamo solo le regioni non banali: escludiamo la regione contenente tutti gli stati e la regione vuota.

A partire dalle regioni non banali si calcola l'insieme R_{TS} di pre-regioni minimali.

Una pre-regione é *minimale* se non esiste un'altra regione *r'* che è una sottoregione di *r*.

N.B.: In un sistema di transizione fortemente connesso ogni regione non banale è anche una pre-regione.

Proprietà di chiusura rispetto all'eccitazione

Una volta calcolato un insieme di pre-regioni minimali, bisogna controllare se il sistema di transizioni è chiuso rispetto all'eccitazione, verificando le seguenti proprietà:

- Chiusura rispetto all'eccitazione: $\forall a$: $\bigcap_{r \in {}^\circ a} r = ER(a)$
- Efficacia dell'evento: $\forall a$: ${}^\circ a \neq \emptyset$

Nell'esempio corrente per gli eventi 1, 2, 3 e 4 gli alberi di espansione corrispondono all'ER dell'evento stesso. Per l'evento 0 invece: $\{s_0, s_3, s_6\} \cap \{s_0, s_4, s_5\} = \{s_0\} = ER(0)$.

La proprietà di chiusura rispetto all'eccitazione è soddisfatta.

Se un sistema di transizione soddisfa la chiusura rispetto all'eccitazione allora si può ottenere una rete di Petri equivalente con una sola transizione per ogni evento del sistema di transizione.

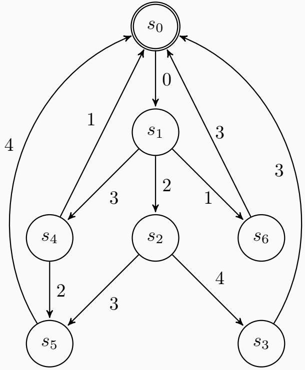
Conversione in rete di Petri: algoritmo

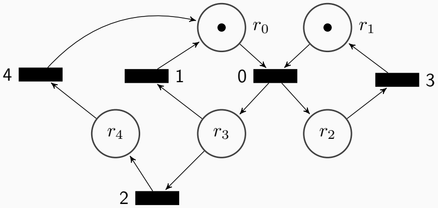
Una volta ottenuto l'insieme delle pre-regioni minimali R_{TS} è possibile ricavare una rete di Petri (unica) detta *saturata* (dato che ad ogni regione corrisponde un posto). Si dimostra che basta considerare le regioni minimali e ottenere una rete di Petri (unica) *saturata minimale*.

Algoritmo:

- per ogni evento $e \in E$ generare nella rete di Petri una transizione etichettata con e;
- Per ogni regione (minimale) $r_i \in R_{TS}$ generare un posto r_i;
- Il posto r_i contiene un gettone nella marcatura iniziale se e solo se la regione corrispondente r_i contiene lo stato iniziale del sistema di transizione;
- Il flusso è il seguente: $e \in r_i\bullet$ se e solo se r_i è una pre-regione di e ed $e \in \bullet r_i$ se e solo se r_i è una post-regione di e.

Conversione in rete di Petri: esempio





Regione	Stati della regione
r_0	$\{s_0, s_3, s_6\}$
r_1	$\{s_0, s_4, s_5\}$
r_2	$\{s_1, s_2, s_3, s_6\}$
r_3	$\{s_1, s_4\}$
r_4	$\{s_2, s_5\}$

Evento	Pre-regioni	Post-regioni
0	r_0, r_1	r_2, r_3
1	r_3	r_0
2	r_3	r_4
3	r_2	r_1
4	r_4	r_0

Equivalenza e minimalità della rete di Petri generata

Il grafo di raggiungibilità della rete di Petri costruita come sopra genera le medesime tracce del sistema di transizione originale (che verifica la chiusura), ma non e' isomorfo ad esso. Possiamo dire che corrisponde a una versione minimizzata del sistema di transizione originale. L'algoritmo di sintesi della rete di Petri esegue una sorta di minimizzazione del sistema di transizione originale.

La rete di Petri risultante può contenere posti ridondanti eliminabili (mediante algoritmi accennati in seguito) per generare: una rete di Petri *irridondante rispetto ai posti*, e una rete di Petri *minima rispetto ai posti*, mantenendo sempre l'equivalenza tra il grafo di raggiungibilità e il sistema di transizione che rispetta la chiusura.

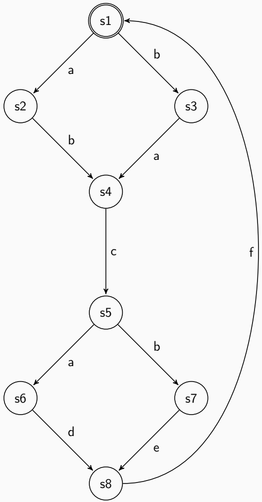
La chiusura rispetto all'eccitazione è necessaria e sufficiente

Che cosa succede se la chiusura non è soddisfatta ?

Se si sintetizza la rete di Petri a partire da un sistema di transizione dove la chiusura non è soddisfatta, allora il linguaggio della rete di Petri sintetizzata contiene strettamente il linguaggio del sistema di transizione originale.

Consideriamo il seguente esempio.

Esempio: flusso con la chiusura non soddisfatta



Stati della regione	Nome
s1, s2, s3, s4	R1
s1, s2, s5, s6	R2
s1, s3, s5, s7	R3
s2, s4, s6	R4
s3, s4, s7	R5
s5, s6, s7	R6
s8	R7

La chiusura è necessaria e sufficiente

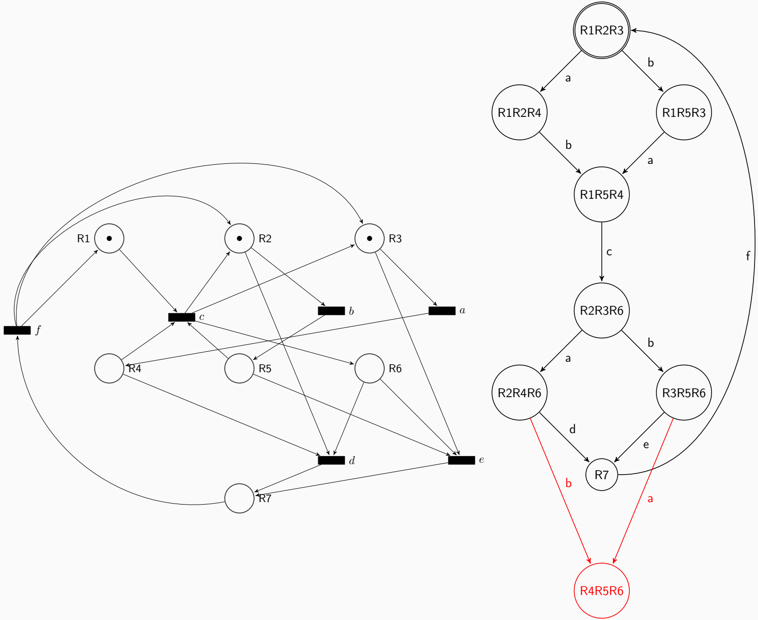
La chiusura non è soddisfatta ad esempio per l'evento a, dato che l'unica pre-regione minimale di a è $R3 = \{s1, s3, s5, s7\}$, che contiene strettamente $ER(a) = \{s1, s3, s5\}$.

Consideriamo lo stato $s7$ che appartiene a $R3$ ma non a $ER(a)$. Esso corrisponde a una marcatura nella rete di Petri ($R3R5R6$) in cui tutte le pre-regioni di a sono marcate (in questo caso la sola pre-regione $R3$), per cui l'evento a è abilitato nel sistema di transizione generato dalla rete di Petri, mentre a non è abilitato nello stato corrispondente del sistema di transizione originale (poiché $s7$ non appartiene a $ER(a)$).

Ne consegue ad esempio che il linguaggio della rete di Petri include la stringa $abcba$ che non è inclusa nel linguaggio del sistema di transizione originale.

Anche l'evento b non soddisfa la condizione di chiusura: ad esempio la stringa $abcab$ appartiene al linguaggio della rete di Petri ma non è inclusa nel linguaggio del sistema di transizione originale.

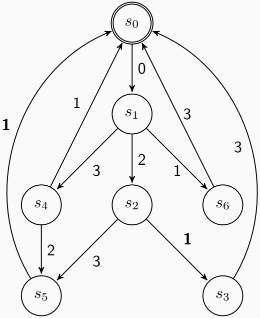
Esempio: flusso con la chiusura non soddisfatta

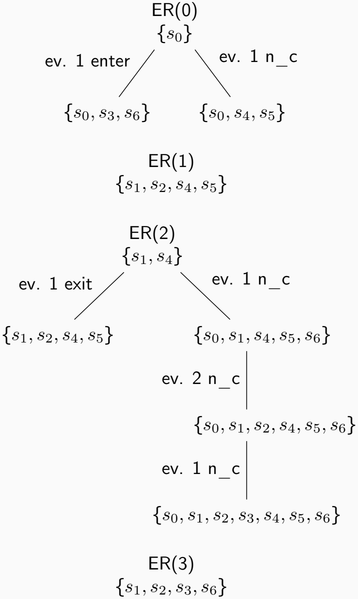


Suddivisione delle etichette

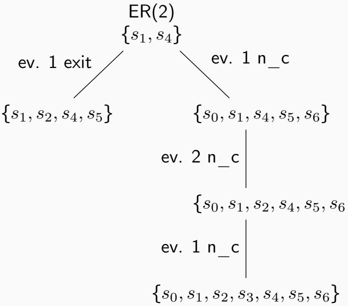
Se la condizione di chiusura rispetto all'eccitazione non è soddisfatta deve essere effettuata la suddivisione delle etichette ("label splitting") per rendere il sistema di transizione chiuso rispetto all'eccitazione (ECTS).

Vediamo una variante dell'esempio iniziale:





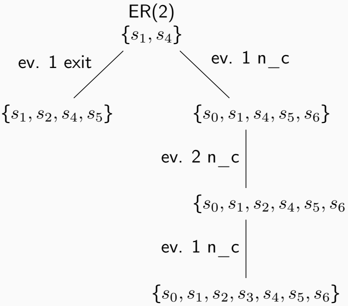
Suddivisione delle etichette



Nell'albero di espansione dell'evento 2 si può notare che:

- l'unica pre-regione non banale per l'evento 2 risulta essere $\{s_1, s_2, s_4, s_5\}$
- la proprietà di chiusura rispetto all'eccitazione non è soddisfatta, infatti $\forall$a: $\bigcap_{r \in {}^\circ a} r = ER(a)$ non vale per l'evento 2, poiché $\bigcap_{r \in {}^\circ 2} r = \{s_1, s_2, s_4, s_5\} \neq ER(2) = \{s_1, s_4\}$

Suddivisione delle etichette



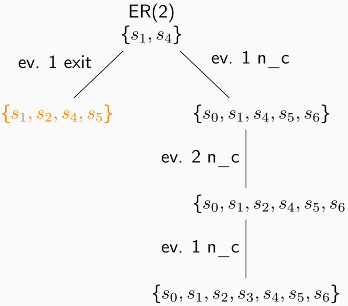
La strategia prevede di scegliere un insieme di stati candidati a diventare regione tra quelli che formavano gli alberi di espansione nell'esplorazione delle regioni per ogni evento.

Sono considerati solo insiemi S' tali per cui:

$$\boxed{\text{ER(a)} \subseteq S' \subset \bigcap_{r \in {}^\circ a} r}$$

Da questi insiemi di stati si seleziona quello con il minor numero di eventi che violano il fatto di essere una regione. Se più insiemi di stati hanno lo stesso numero di eventi in violazione, si seleziona il più piccolo.

Suddivisione delle etichette



La strategia prevede di scegliere un insieme di stati candidati a diventare regione tra quelli che formavano gli alberi di espansione nell'esplorazione delle regioni per ogni evento.

Sono considerati solo insiemi S' tali per cui:

$$\boxed{\text{ER(a)} \subseteq S' \subset \bigcap_{r \in {}^\circ a} r}$$

Da questi insiemi di stati si seleziona quello con il minor numero di eventi che violano il fatto di essere una regione. Se più insiemi di stati hanno lo stesso numero di eventi in violazione, si seleziona il più piccolo.

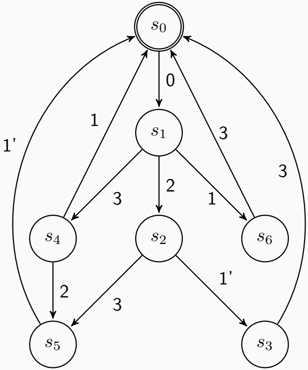
Suddivisione delle etichette

Gl'insiemi di stati selezionati sono forzati a diventare una regione. Come? Suddividendo le etichette di quegli eventi che non verificano le condizioni per essere una regione.

Questa strategia garantisce che la nuova intersezione delle pre-regioni è più vicina a $ER(a)$.

N.B.: Nel calcolo dell'intersezione vengono utilizzate tutte le regioni presenti sulle foglie dell'albero di espansione (in questo caso si utilizza anche la regione banale $\{s_0, s_1, s_2, s_3, s_4, s_5, s_6\}$ che non influisce sul risultato finale).

Suddivisione delle etichette



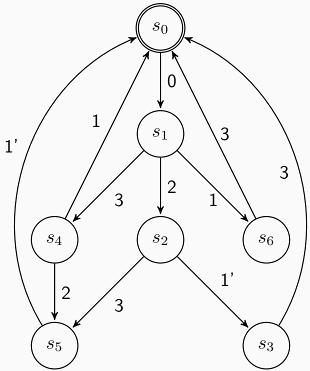
S' contiene i seguenti insiemi di stati con i relativi eventi in violazione:

Insieme di stati	Eventi in violazione
$\{s_1, s_4\}$	1
$\{s_1, s_2, s_4\}$	1, 2, 3
$\{s_1, s_4, s_5\}$	1, 2, 3

L'insieme $\{s_1, s_4\}$ viene scelto come regione candidata essendo l'insieme più piccolo col minor numero di eventi in violazione: l'evento 1 che viola la condizione *no_cross*, per cui si suddivide l'etichetta 1 nelle etichette 1 e $1'$.

La suddivisione delle etichette per un evento può portare modifiche agli alberi di espansione degli altri eventi, per questo motivo si continua iterativamente nel calcolo delle regioni fino a convergenza, cioè finché non si arriva ad avere una ECTS.

Suddivisione delle etichette



La suddivisione è stata effettuata in modo da consentire all'evento 1 di soddisfare la condizione *exit* e all'evento 1' la condizione *no_cross* rispetto alla regione $\{s_1, s_4\}$.

Regioni generate dopo la suddivisione delle etichette

Dopo aver effettuato la suddivisione delle etichette è necessario calcolare nuovamente le regioni non banali che sono:

Regione	Stati della regione
r_0	$\{s_0, s_3, s_6\}$
r_1	$\{s_0, s_4, s_5\}$
r_2	$\{s_1, s_2, s_3, s_6\}$
r_3	$\{s_1, s_4\}$
r_4	$\{s_2, s_5\}$

La proprietà di chiusura rispetto all'eccitazione è soddisfatta.

Ottimizzazioni

Minizzazione delle regioni

Una rete saturata minimale può essere ridondante, cioè si possono eliminare dei posti mantenendo l'equivalenza tra grafo di raggiungibilità e sistema di transizione.

Analogia con la minimizzazione logica:

- Rete saturata $\rightarrow$ Insieme di tutti gl'implicanti di una funzione logica
- Rete saturata minimale $\rightarrow$ Insieme di tutti gl'implicanti primi di una funzione logica
- (Obiettivo:) Rete irridondante con regioni minimali $\rightarrow$ Copertura irridondante d'implicanti primi

Caveat: ci sono esempi di sistemi di transizione con chiusura in cui qualsiasi rete di Petri minima richiede l'uso di almeno una regione che non sia minimale.

Reti irridondanti e reti minime

Una rete di Petri è **irridondante rispetto ai posti** se non si può rimuovere alcun posto senza perdere l'equivalenza del grafo di raggiungibilità.

Una rete di Petri è **minima rispetto ai posti** se non esiste una rete equivalente con il medesimo insieme di transizione e meno posti.

Un insieme di regioni R si dice ridondante se esiste una regione $r \in R$ tale che $R \setminus \{r\}$ soddisfa ancora la chiusura rispetto all'eccitazione.

Un insieme di regioni R si dice minimo se è **irridondante** e nessun altro insieme irridondante di regioni contiene meno regioni.

Se si applica l'algoritmo di sintesi a partire da un insieme di regioni R irridondante o minimo si ottiene una rete di Petri irridondante o minima.

Reti irridondanti e reti minime

Si può sempre ottenere una rete di Petri irridondante rispetto ai posti a partire da un insieme irridondante di regioni minimali (le quali assolvono il ruolo degl'implicanti primi nella minimizzazione logica).

Tuttavia non è detto che ci sia sempre un insieme minimo di regioni costituito solo da regioni minimali. Esiste un esempio in cui c'è un insieme (unico) irridondante di regioni minimali in cui la rete minima rispetto ai posti si ottiene dalla rete irridondante unendo due regioni minimali in una regione non minimale.

Si dimostra che si può sempre ottenere una rete con un numero minimo di posti da un insieme irridondante di regioni minimali mediante l'unione di pre-regioni minimali in pre-regioni non più minimali.

Regioni essenziali

Una regione r è **essenziale** se ci sono uno stato s e un evento e tali che $r \in {}^\circ e, s \notin r$ e per tutti gli $r' \in {}^\circ e, r' \neq r$ si ha $s \in r'$.

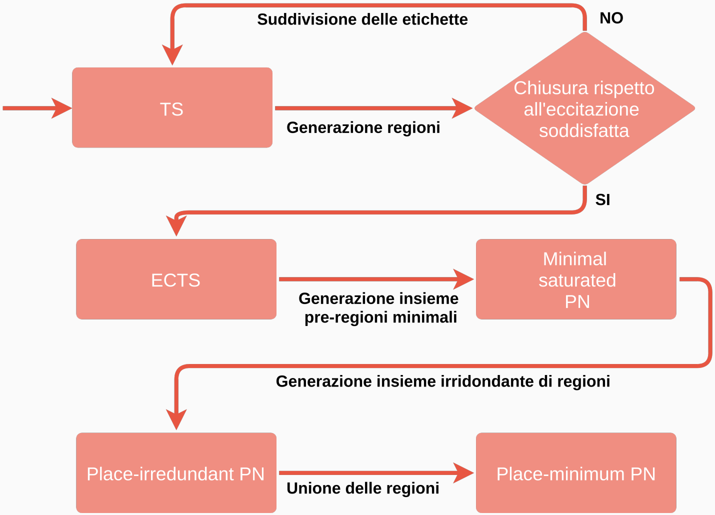
Intuitivamente: una regione è essenziale se è l'unica regione che rimuove dall'intersezione delle pre-regioni uno stato s in cui l'evento e non è abilitato. Per verificare se una regione è essenziale è sufficiente verificare se l'eliminazione di tale regione viola la proprietà di chiusura rispetto all'eccitazione per uno o più eventi.

Generazione insieme minimo di regioni

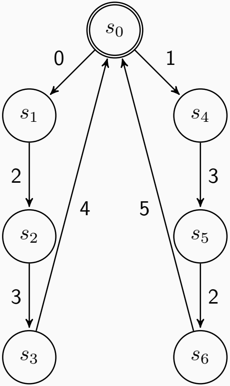
Ha lo scopo di trovare l'insieme di costo minimo di regioni non essenziali, le quali combinate con quelle essenziali riescono a soddisfare la proprietà di chiusura rispetto all'eccitazione.

Per trovare l'insieme irridondante delle regioni, è sufficiente utilizzare un algoritmo vorace per la ricerca dell'insieme di costo minimo che soddisfa la proprietà di chiusura rispetto all'eccitazione. Consideriamo unitario il costo di ogni regione, avendo come scopo la minimizzazione del numero di posti della rete di Petri finale (è possibile modificare la funzione di costo ad esempio per minimizzare il numero di archi). Dato che l'aggiunta di una regione implica l'incremento del costo, tale proprietà consente di effettuare dei tagli nello spazio di ricerca e inoltre garantisce che la soluzione trovata non è soltanto **irridondante** ma è anche **minima**.

Flusso completo dell'algoritmo di sintesi



Generazione insieme minimo di regioni: esempio



Dal sistema di transizione in figura si ricavano le seguenti regioni non banali:
$\{s_0\}$, $\{s_1, s_2, s_3\}$, $\{s_1, s_2, s_4\}$, $\{s_1, s_3, s_5\}$, $\{s_1, s_4, s_5\}$, $\{s_2, s_3, s_6\}$, $\{s_2, s_4, s_6\}$, $\{s_3, s_5, s_6\}$, $\{s_4, s_5, s_6\}$

Di queste regioni quelle essenziali sono:
$\{s_0\}$, $\{s_1, s_2, s_4\}$, $\{s_1, s_3, s_5\}$, $\{s_1, s_4, s_5\}$, $\{s_2, s_4, s_6\}$

Le regioni non essenziali sono:
$\{s_2, s_3, s_6\}$, $\{s_3, s_5, s_6\}$, $\{s_4, s_5, s_6\}$, $\{s_1, s_2, s_3\}$.

Cercando una combinazione di regioni di costo minimo si trova l'insieme composto da due regioni: $\{s_2, s_3, s_6\}$ e $\{s_3, s_5, s_6\}$.

Insieme delle regioni di costo minimo risultante: $\{s_0\}$, $\{s_1, s_2, s_4\}$, $\{s_1, s_3, s_5\}$, $\{s_1, s_4, s_5\}$, $\{s_2, s_4, s_6\}$, $\{s_2, s_3, s_6\}$, $\{s_3, s_5, s_6\}$

Unione delle pre-regioni minimali

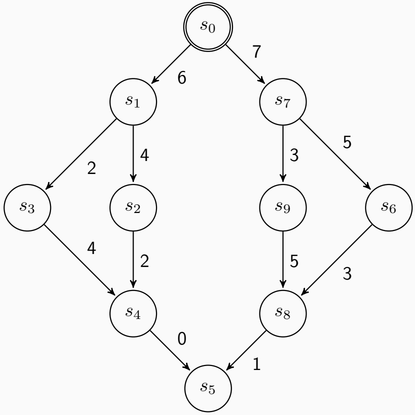
Per generare una rete di Petri ***minima rispetto ai posti*** è necessario eseguire almeno una volta l'unione tra due pre-regioni.

L'algoritmo di unione consiste in una ricerca esaustiva di coppie di pre-regioni **disgiunte**, controllando se unendole la proprietà di chiusura rispetto all'eccitazione rimane soddisfatta. Trovata una coppia che soddisfa tale proprietà si interrompe la ricerca e si sostituisce, al posto delle due pre-regioni originali, la nuova regione data dalla loro unione. S'interrompe la ricerca per il fatto che nella maggior parte dei casi non si può effettuare più di un'unione sulla stessa rete. Si tratta di un compromesso tra tempo di calcolo e qualità della soluzione.

Unione delle pre-regioni minimali: esempio

L'insieme delle regioni che si ricavano dal sistema di transizione è il seguente: $\{\mathbf{s_0}\}$, $\{\mathbf{s_1}, \mathbf{s_2}\}$, $\{\mathbf{s_1}, \mathbf{s_3}\}$, $\{\mathbf{s_2}, \mathbf{s_4}\}$, $\{\mathbf{s_3}, \mathbf{s_4}\}$, $\{\mathbf{s_5}\}$, $\{\mathbf{s_6}, \mathbf{s_7}\}$, $\{\mathbf{s_6}, \mathbf{s_8}\}$, $\{\mathbf{s_7}, \mathbf{s_9}\}$, $\{\mathbf{s_8}, \mathbf{s_9}\}$.

Si può notare che tutte le regioni sono essenziali e la proprietà di chiusura rispetto all'eccitazione è soddisfatta. Nonostante ciò è possibile ridurre il numero di regioni effettuando l'unione tra $\{\mathbf{s_2}, \mathbf{s_4}\}$ e $\{\mathbf{s_6}, \mathbf{s_8}\}$.



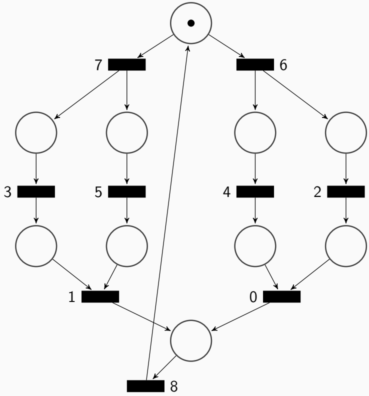
Unione delle pre-regioni minimali: esempio

Per capire meglio perché è possibile unire $\{\mathbf{s_2}, \mathbf{s_4}\}$ e $\{\mathbf{s_6}, \mathbf{s_8}\}$ esaminiamo l'insieme delle pre-regioni per ciascun evento.

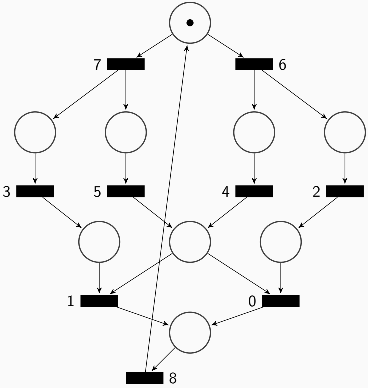
Evento	Pre-regioni	ER
0	$\{s_2, s_4\}$, $\{s_3, s_4\}$	$\{s_4\}$
1	$\{s_6, s_8\}$, $\{s_8, s_9\}$	$\{s_8\}$
2	$\{s_1, s_2\}$	$\{s_1, s_2\}$
3	$\{s_6, s_7\}$	$\{s_6, s_7\}$
4	$\{s_1, s_3\}$	$\{s_1, s_3\}$
5	$\{s_7, s_9\}$	$\{s_7, s_9\}$
6	$\{s_0\}$	$\{s_0\}$
7	$\{s_0\}$	$\{s_0\}$
8	$\{s_5\}$	$\{s_5\}$

Si può notare che l'unione delle regioni $\{s_2, s_4\}$ e $\{s_6, s_8\}$ potrebbe violare la proprietà di chiusura rispetto all'eccitazione soltanto per gli eventi 0 e 1. Sostituendo le regioni originali con l'unione si può verificare che tale proprietà rimane soddisfatta.

Unione delle pre-regioni minimali: esempio



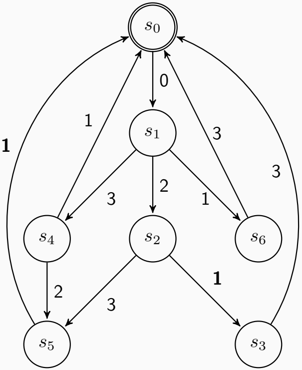
Rete di Petri risultante senza l'unione delle pre-regioni



Rete di Petri risultante con l'unione delle pre-regioni

Esempi

Esempio completo 1: generazione delle regioni



ER(0)
$\{s_0\}$

ev. 1 enter: $\{s_0, s_3, s_6\}$

ev. 1 n_c: $\{s_0, s_4, s_5\}$

ER(1)
$\{s_1, s_2, s_4, s_5\}$

ER(2)
$\{s_1, s_4\}$

ev. 1 exit: $\{s_1, s_2, s_4, s_5\}$

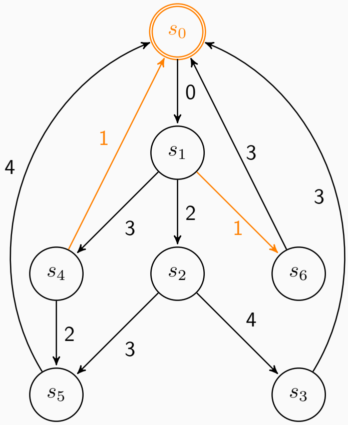
ev. 1 n_c: $\{s_0, s_1, s_4, s_5, s_6\}$

ev. 2 n_c: $\{s_0, s_1, s_2, s_4, s_5, s_6\}$

ev. 1 n_c: $\{s_0, s_1, s_2, s_3, s_4, s_5, s_6\}$

ER(3)
$\{s_1, s_2, s_3, s_6\}$

Esempio completo 1: albero di espansione dell'evento 2

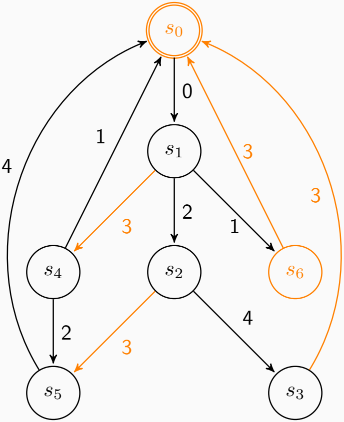


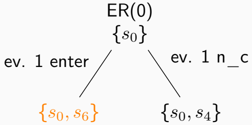
ER(0)
$\{s_0\}$

ER(2) non è una regione a causa dell'evento 1 che presenta sia archi con proprietà *exit* che *no_ cross*, di conseguenza l'espansione può essere effettuata in due direzioni distinte:

- Aggiungere gli stati s_2 ed s_5 per fare in modo che tutte le etichette abbiano la proprietà *exit*
- Aggiungere gli stati s_0, s_5 ed s_6 per fare in modo che tutte le etichette abbiano la proprietà *no_ cross*

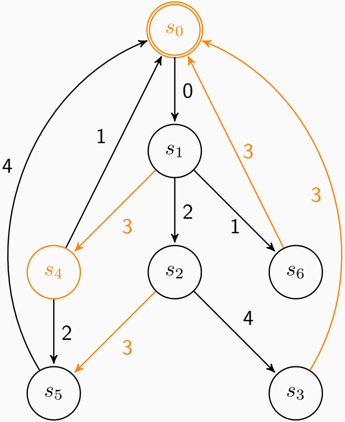
Esempio completo 1: albero di espansione dell'evento 2

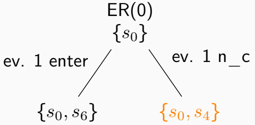




- $\{s_1, s_2, s_4, s_5\}$ è una regione.
- A causa dell'evento 2 l'insieme $\{s_0, s_1, s_4, s_5, s_6\}$ non è una regione e deve essere espanso ulteriormente.

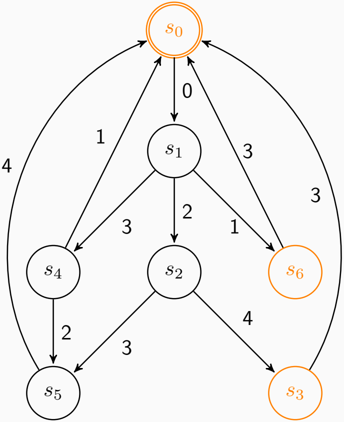
Esempio completo 1: albero di espansione dell'evento 2

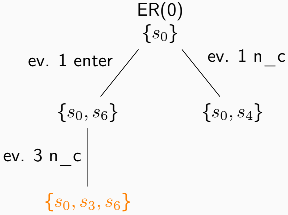




L'insieme $\{s_0, s_1, s_2, s_4, s_5, s_6\}$ non è una regione, la violazione è di nuovo causata dall'evento 1.

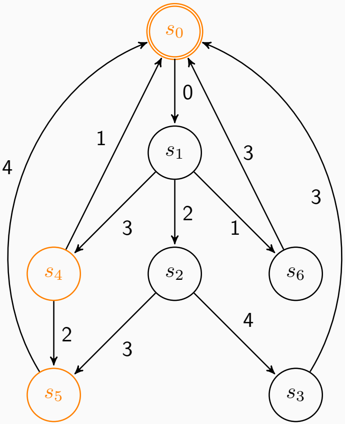
Esempio completo 1: albero di espansione dell'evento 2

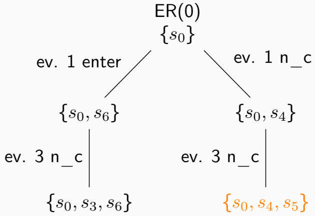




L'insieme banale contenente tutti gli stati sicuramente è una regione (tutte le etichette hanno la medesima proprietà *no_cross*), non ci sono ulteriori espansioni da fare

Esempio completo 1: albero di espansione dell'evento 2





Esempio completo 1: regioni

Regioni non banali ricavate dal sistema di transizione:

Regione	Stati della regione
r_0	$\{s_0, s_3, s_6\}$
r_1	$\{s_0, s_4, s_6\}$
r_2	$\{s_1, s_2, s_3, s_6\}$
r_3	$\{s_1, s_2, s_4, s_5\}$

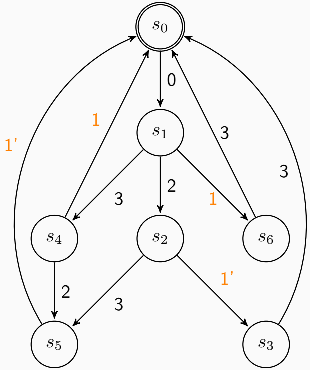
La proprietà di chiusura rispetto all'eccitazione non è soddisfatta a causa dell'albero di espansione dell'evento 2: $\mathbf{r_3} = \{\mathbf{s_1}, \mathbf{s_2}, \mathbf{s_4}, \mathbf{s_5}\} \neq \mathbf{ER}(\mathbf{2})$ (da notare che la regione $\{s_0, s_1, s_2, s_3, s_4, s_5, s_6\}$ presente nell'albero di espansione dell'evento 2 non si considera in quanto banale)

Esempio completo 1: suddivisione delle etichette

Per la suddivisione delle etichette si considera l'albero di espansione dell'evento 2. Utilizzando la formula ER(a)$\subseteq S' \subset \bigcap_{r \in {}^\circ a} r$, nel nostro caso si ha $S' = ER(2) = \{s_1, s_4\}$

Facendo riferimento all'albero di espansione dell'evento 2, l'insieme di stati $\{s_1, s_4\}$ non è una regione in quanto la proprietà non è soddisfatta per l'evento 1, sul quale sarà effettuata la suddivisione delle etichette per consentire a S' di diventare una regione.

Esempio completo 1: suddivisione delle etichette



La suddivisione è stata effettuata in modo da consentire all'evento 1 di soddisfare la condizione *exit* e all'evento 1' la condizione *no_cross* rispetto alla regione $\{s_1, s_4\}$.

Esempio completo 1: generazione delle regioni

Dopo aver effettuato la suddivisione delle etichette è necessario calcolare nuovamente le regioni non banali, che sono le seguenti:

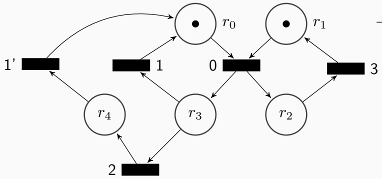
Regione	Stati della regione
r_0	$\{s_0, s_3, s_6\}$
r_1	$\{s_0, s_4, s_5\}$
r_2	$\{s_1, s_2, s_3, s_6\}$
r_3	$\{s_1, s_4\}$
r_4	$\{s_2, s_5\}$
r_5	$\{s_0, s_1, s_3, s_4, s_6\}$

In questo caso la proprietà di chiusura rispetto all'eccitazione è soddisfatta.

La regione $r_5 = \{s_0, s_1, s_3, s_4, s_6\}$ non essendo minimale in quanto contenente la regione $r_3 = \{s_1, s_4\}$ sarà rimossa dall'insieme delle regioni.

Esempio completo 1: rete di Petri

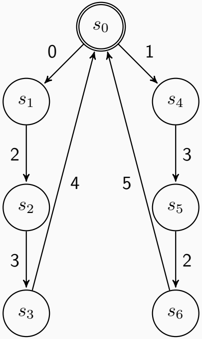
Dato che tutte le regioni dell'insieme di regioni minimali sono essenziali non si effettua il calcolo dell'insieme irridondante di regioni, inoltre non è possibile effettuare alcuna unione tra due regioni. Calcolando per quale evento una regione è una pre-regione oppure post-regione si effettua la trasformazione in rete di Petri.

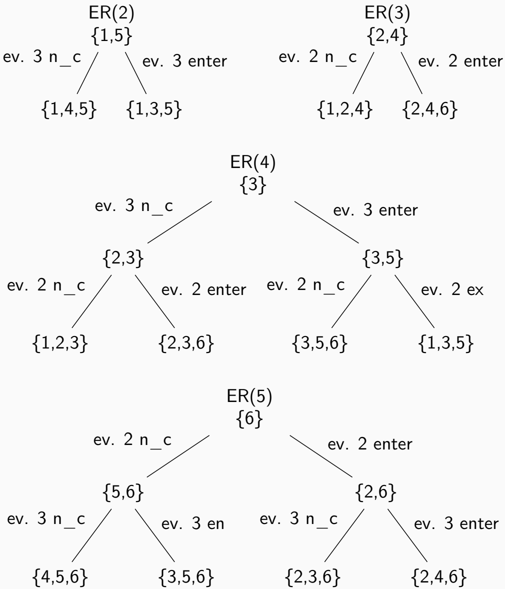


Evento	Pre-regioni
0	r_0, r_1
1	r_3
1'	r_4
2	r_3
3	r_2

Evento	Post-regioni
0	r_2, r_3
1	r_0
1'	r_0
2	r_4
3	r_1

Esempio completo 2: generazione delle regioni





Per gli eventi 0 e 1 ER = $\{0\}$ rappresenta già una regione.

Esempio completo 2: regioni

Le regioni non banali generate sono le seguenti:

Regione	Stati della regione
r_0	$\{s_0\}$
r_1	$\{s_1, s_2, s_3\}$
r_2	$\{s_1, s_2, s_4\}$
r_3	$\{s_1, s_3, s_5\}$
r_4	$\{s_1, s_4, s_5\}$
r_5	$\{s_2, s_3, s_6\}$
r_6	$\{s_2, s_4, s_6\}$
r_7	$\{s_3, s_5, s_6\}$
r_8	$\{s_4, s_5, s_6\}$

La proprietà di chiusura rispetto all'eccitazione è soddisfatta.

Tutte le regioni sono minimali.

Le regioni essenziali sono le seguenti: r_0, r_2, r_3, r_4, r_6.

Esempio completo 2: ricerca insieme minimo di regioni

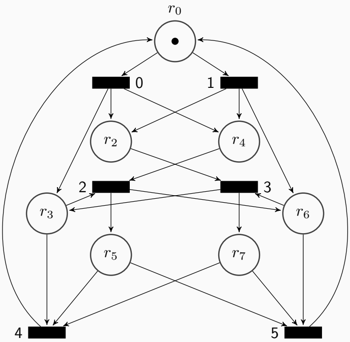
Le regioni non essenziali sono: r_1, r_5, r_7, r_8.

Nessuna regione tra quelle non essenziali può soddisfare singolarmente la proprietà di chiusura rispetto all'eccitazione (insieme con le regioni essenziali, sempre presenti nella soluzione). Con l'algoritmo vorace è possibile trovare l'insieme $\{r_5, r_7\}$, sufficiente per soddisfare la chiusura rispetto all'eccitazione. Le restanti regioni r_1 e r_8 sono eliminate.

Esempio completo 2: pre-regioni e post-regioni

Poiché in questo caso non è possibile effettuare alcuna unione tra regioni (nonostante ci siano molte coppie di regioni disgiunte nessuna unione riuscirebbe a preservare la proprietà di chiusura rispetto all'eccitazione) si passa alla creazione delle pre-regioni e delle post-regioni.

Evento	Pre-regioni	Post-regioni
0	r_0	r_2, r_3, r_4
1	r_0	r_2, r_4, r_6
2	r_3, r_4	r_5, r_6
3	r_2, r_6	r_3, r_7
4	r_3, r_5, r_7	r_0
5	r_5, r_6, r_7	r_0



Bibliografia

- J. Cortadella, M. Kishinevsky, A. Kondratyev, L. Lavagno. A. Yakovlev
  Logic Synthesis of Asynchronous Controllers and Interfaces
  Springer Verlag 2002
- J. Cortadella, M. Kishinevsky, L. Lavagno. A. Yakovlev
  Deriving Petri nets from finite transition systems
  IEEE Transactions on Computers, Vol. 47, N. 8, pp. 859-882, August 1998
- J. Carmona, J. Cortadella, M. Kishinevsky
  A region-based algorithm for discovering Petri nets from event logs
  BPM 2008, LNCS 5240, p. 358-373, Springer Verlag 2008