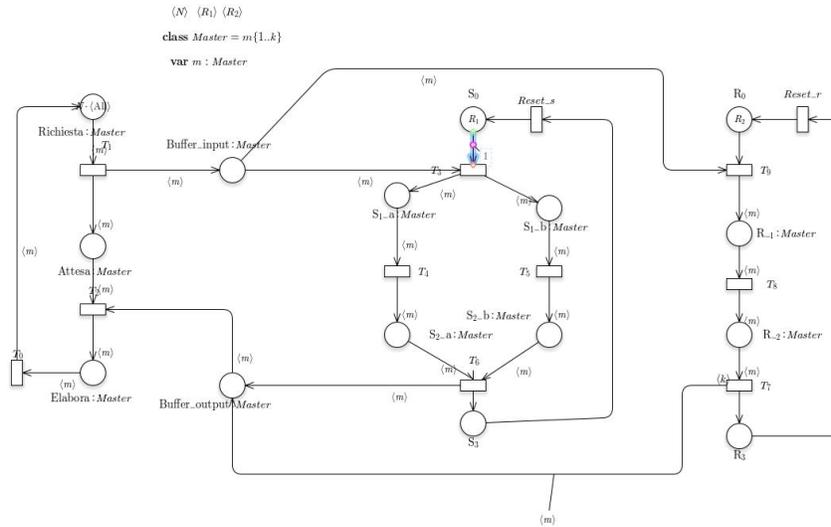


# Contents

## 1 Rete E



I master sono modellati dai posti Richiesta, Attesa e Elabora. Lo slave di tipo 1 e' modellato dai posti S0, S1<sub>a</sub>, S2<sub>a</sub>, S1<sub>b</sub>, S3. Lo slave di tipo 2 e' modellato dai posti R0, R1, R2 e R3. La richiesta agli slave e' modellata attraverso due posti: Buffer<sub>Input</sub>, Buffer<sub>Output</sub>.

La classe di colori utilizzata e' Master, che permette di distinguere i tre master  $m_1$ ,  $m_2$ ,  $m_3$ , mentre il color domain e'  $m$  e viene utilizzato in tutte le transizioni dove c'e' necessita' di distinguere l'origine del marking.

### 1.1 Reachability Graph

Le seguenti tabelle elencano la dimensione dello spazio degli stati al variare del numero di master (N) e di slave (S<sub>n</sub> e R<sub>n</sub>).

N	Sn, Rn	Marking RG	Marking SRG
1	1	1024	232
1	2	4080	888
1	3	9280	2032
1	4	16480	3616
2	1	28480	5240
2	2	201708	35874
2	3	678240	119488
3	1	310400	54080
3	2	3151680	538380

L'utilizzo del Symbolic Reachability Graph permette di ridurre il numero di marking di un fattore  $>4$ .

## 1.2 Reachability Graph al variare delle classi di colore

k	Marking RG	Marking SRG
3	1024	232
4	5632	460
5	29696	804
6	151552	1288
7	753664	1936
8	3670016	2772
9	17563648	3820
10	82837504	5104

La tabella mostra l'aumentare dei marking del Reachability Graph all'aumentare delle classi di colore. Benché la crescita dello spazio degli stati sia lineare in entrambi i casi, notiamo come lo spazio del SRG sia notevolmente più contenuto dello spazio del RG. Inoltre, il rapporto marking RG/SRG (dello stato degli spazi) aumenta proporzionalmente al valore di k.

A conferma di quanto espresso, notiamo che anche con valori di marcatura iniziale maggiore per il master e gli slave, notiamo lo stesso effetto sul rapporto di marking RG/SRG (limitatamente alle possibilità di calcolo).

N, Sn, Rn	k	Marking RG	Marking SRG
2, 2, 2	3	201708	35874
2, 2, 2	4	4190624	203136
2, 2, 2	5	78523200	875478

### 1.3 Considerazioni su Fork/Join

Non può avvenire un join fra due processi con diverso padre quando:

- $N = 1$ : sullo stesso spazio non può essere presente più di un token con lo stesso colore
- $S_n = 1$ : in questo secondo caso, come nei precedenti esercizi, avviene un solo fork alla volta.

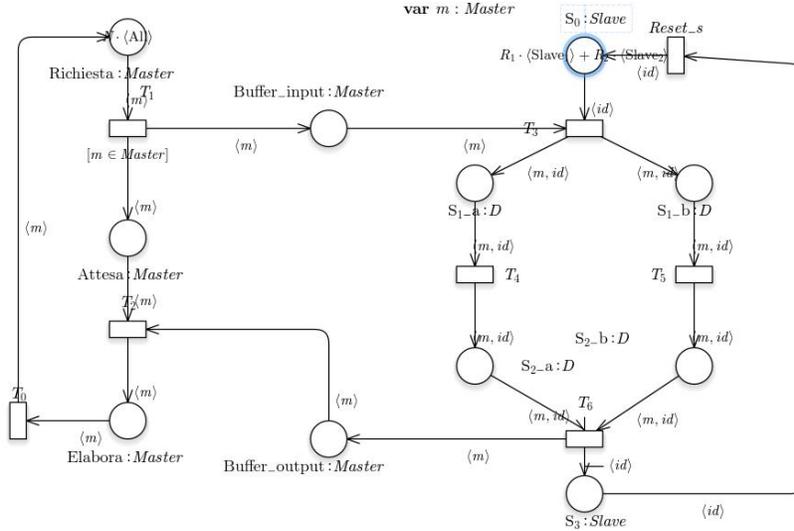
Di seguito viene mostrata un'esecuzione in cui è possibile che avvenga il join di due processi con lo stesso padre:

- con  $N > 1$  consideriamo che nella rete vi siano al momento iniziale due token dello stesso colore "m0"
- entrambi i token vengono inseriti nel  $\text{Buffer}_{\text{input}}$  e in seguito inviati allo slave di tipo 1.
- quando  $S_n > 1$  uno dei due token può eseguire lo scatto attraverso la transizione T3 prima che l'altro token arrivi alla transizione T6 (in cui avviene il join).

## 2 Rete F

$\langle N \rangle \langle n \rangle \langle R_1 \rangle \langle R_2 \rangle$   
**class** *Slave* = {ID<sub>1</sub>} **is** Slave<sub>1</sub> + {ID<sub>2</sub>} **is** Slave<sub>2</sub> **var** *id* : *Slave*  
**class** *Master* = {m<sub>1</sub>, m<sub>3</sub>} **is** M<sub>1</sub> + {m<sub>2</sub>} **is** M<sub>2</sub>  
**domain** *D* = *Master* × *Slave*

**Condizione booleana FORK :**  
 $[(m = m_1 \vee m = m_3) \wedge id = ID_1] \vee (m = m_2 \wedge id = ID_2)$



I master sono modellati dai posti Richiesta, Attesa e Elabora. Gli slave di tipo 1 sono modellato dai posti S0, S1<sub>a</sub>, S2<sub>a</sub>, S1<sub>b</sub> e S3. La richiesta agli slave e' modellata attraverso due posti: BufferInput, Bufferoutput.

Nella rete sono presenti due classi di colore:

- Master, che permette di distinguere i diversi master  $m_1, m_2, m_3$
- Slave, che permette di distinguere i due slaves  $ID_1$  e  $ID_2$

La classe Slave e' divisa in due sottoclassi, Slave1 e Slave2, per facilitare in seguito la parametrizzazione degli slaves. Anche la classe Master e' divisa in due sottoclassi utilizzate nell'espressione booleane della transizione Fork. Il dominio  $D = Master \times Slave$  e' utilizzato per la multiplicity degli archi fra fork e join, in modo da tener traccia delle richieste effettuate.

### 2.1 Reachability Graph

Le seguenti tabelle elencano la dimensione dello spazio degli stati al variare del numero di master (N) e di slave (S<sub>n</sub> e R<sub>n</sub>).

N	R1, R2	Marking RG	Marking SRG
1	1	768	432
1	2	2560	1440
1	3	5376	3024
1	4	5184	9216
2	1	18720	9720
2	2	97696	50481
2	3	267120	137376
3	1	192000	97600
3	2	1309440	663520

Rispetto alla precedente rete, il rapporto stato degli spazi di RG/SRG e' minore.

## 2.2 Reachability Graph al variare delle classi di colore

La seguente tabella mostra l'incremento dello spazio degli stati all'aumentare degli elementi della classe di color Master. Come nella precedente rete, il rapporto marking RG/SRG aumenta proporzionalmente alla cardinalita' della classe.

M1	M2	Marking RG	Marking SRG
2	1	768	432
3	1	3840	960
2	2	4096	1296
3	2	20480	2880
3	3	102400	6400
4	3	491520	12000
4	4	2359296	22500
5	4	11010048	37800
5	5	51380224	63504

Nella seguente tabella viene evidenziato l'incremento dello stato degli spazi all'aumentare degli elementi della classe di colore Slave.

Slave1	Slave2	Marking RG	Marking SRG
1	1	768	432
1	2	2048	720
2	1	2688	864
2	2	7168	1440
2	3	17920	2016
3	2	22528	2160
3	3	56320	3024
3	4	135168	3888
4	3	163840	4032
4	4	393216	5184
4	5	917504	6336
5	4	1081344	6480
5	5	2523136	7920

Notiamo che l'alta cardinalita' della classe Slave abbia avuto un impatto minore sulla dimensione del RG rispetto alla classe master. Il rapporto RG/SRG segue quanto detto finora ed e' il piu' grande fra tutte le reti analizzate.

### 2.3 Considerazioni su Fork/Join

Posso fare le stesse considerazioni che nella rete precedente (rete E), tenendo in considerazione che avendo due diverse tipologie di slave, un join fra due processi con padre diverso avverra' solo quando

$$(R_1 > 1 \vee R_2 > 1) \wedge N > 1$$