

# Relazione Esercizio Rete WN (F)

Francesco Galla', francesco.galla@edu.unito.it

## 1 Rete F

Rete F: Master di tre tipi distinti (seppur di uguale struttura) che chiameremo  $m_1$ ,  $m_2$  e  $m_3$  e due slave, di distinto tipo ID1 e ID2, ma di uguale struttura (usare quella indicata come “tipo 1”). I master di tipo  $m_1$  e  $m_3$  richiedono servizio solo agli slave di tipo ID1 e i master di tipo  $m_2$  richiedono servizio solo agli slave di tipo ID2. Vi siano  $N$  master per ognuno dei tre tipi e  $R_1$  slave per il tipo ID1 e  $R_2$  per il tipo ID2.

### 1.1 La rete di Petri

La Figura 1 rappresenta la rete di Petri WN del secondo esercizio (rete F). Il master è modellato dai posti `M_Richieste`, `M_Attesa`, `M_Risultati` e dalle transizioni `Dispatch`, `M_Return`, `Ottieni_Res`. Lo slave di tipo 1 è modellato dai posti `S1_Attesa`, `Child1/Child2`, `Fine_C1/Fine_C2`, `S1_Fine` e dalle transizioni `Fork`, `C1_Processa/C2_Processa`, `Join`, `S1_Return`, `S1_Send_Res`. Per collegare le strutture di master e slave sono stati utilizzati due buffer: uno in ingresso, identificato dal posto `Buffer`, uno in uscita identificato dal posto `S1_Send_Res`.

### 1.2 I colori

Questa rete utilizza due classi di colori: una classe di colori chiamata *Master*, utilizzata per distinguere i tipi di master  $m_1, m_2, m_3$ , e una classe chiamata *Slave*, utilizzata per distinguere i tipi di slave  $ID_1, ID_2$ . Le due classi sono divise in due sottoclassi ciascuna per permettere la differenziazione dei tipi di slave. L'utilizzo di un domain  $MxS = (Master)x(Slave)$  permette di identificare le richieste in base ai tipi corrispondenti di *Master* e *Slave*. Questo dominio è utilizzato principalmente per i posti della struttura *Slave*, per evitare di perdere traccia delle richieste una volta che viene chiamata la transizione `Fork`.

### 1.3 Fork con Guardia

Sulla transizione `Fork` è stata inserita una guardia che permette lo scatto solo in caso il master e lo slave rispettino i requisiti dell'esercizio, nello specifico:

1. Se il master appartiene alla sottoclasse  $Master_1$  (vale per  $m_1, m_3$ ) allora lo slave deve appartenere alla sottoclasse  $Slave_1$  ( $ID_1$ ).

- Se il master appartiene alla sottoclasse  $Master_2$  (vale per  $m_2$ ) allora lo slave deve appartenere alla sottoclasse  $Slave_2$  ( $ID_2$ ).

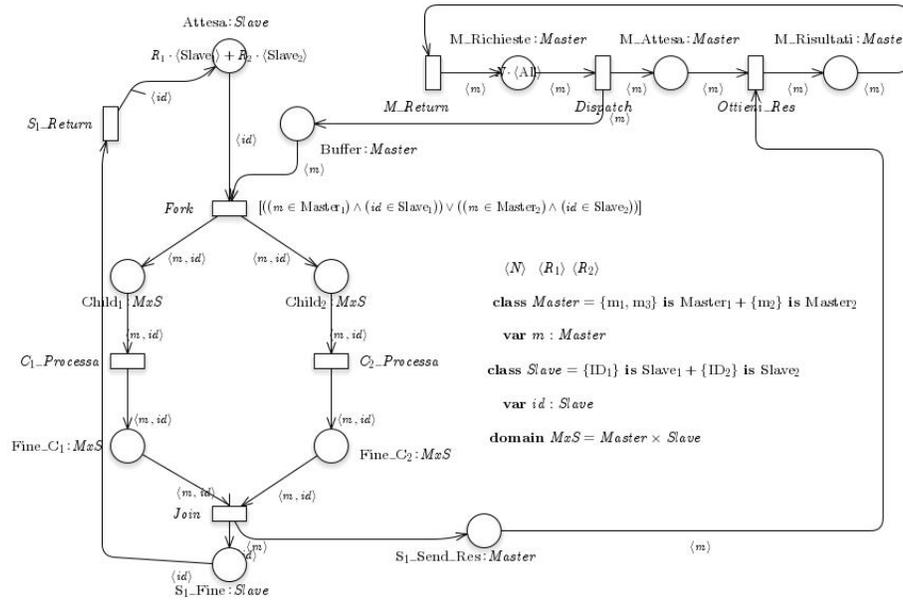


Figure 1: Modello WN della rete F

## 1.4 I risultati

Le tabelle elencano la dimensione dello spazio degli stati al variare del numero di master e di slave (marcatura iniziale). Il numero di markings aumenta in maniera simile sia nel caso del RG sia nel caso del SRG, considerando che l'incremento di  $R_1, R_2$  causa un aumento del numero di marking di un fattore di circa 3, mentre l'incremento di  $N$  causa un aumento di un fattore 30. Il SRG permette di ridurre il numero di markings a circa la metà rispetto al corrispettivo RG, seppure la dimensione dello spazio degli stati aumenti in maniera coerente con il RG. L'SRG permette inoltre di ottenere tempi di calcolo leggermente minori rispetto all'RG.

## 1.5 Tipologie di Master

Aumentando le tipologie di master non è stato sempre possibile preservare la proporzione tra le due sottoclassi, per cui si è scelto di mantenere la seguente relazione:

$$|Master_1| > |Master_2|$$

Effetto della marcatura iniziale sul numero di markings			
N	R1/R2	Markings RG	Markings SRG
1	1	768	432
1	2	2560	1440
1	3	5376	3024
2	1	18720	9720
2	2	97696	50481
2	3	267120	137376
3	1	192000	97600
3	2	1309440	663520

Table 1: Variazione dello spazio degli stati.

I valori di  $N, R_1, R_2$  sono costanti e fissati a 1, per evidenziare l'effetto dell'incremento delle tipologie di master sulla rete. Le tipologie di Slave sono fissate a 2.

Si nota come la crescita delle tipologie di master abbia portato ad un incremento maggiore (circa di un fattore 6) nel caso del RG, mentre il SRG ha un incremento in proporzione minore (circa di un fattore 2).

Effetto delle tipologie di master sul numero di markings		
Tipologie Master	Markings RG	Markings SRG
3	768	432
4	3840	960
5	20480	2880
6	98304	5400
7	481520	12000

Table 2: Variazione dello spazio degli stati.

## 1.6 Tipologie di Slave

Aumentando le tipologie di Slave si è cercato di mantenere la proporzione  $|Slave_1| == |Slave_2|$  quando possibile (ossia quando erano presenti un numero pari di slave).

Anche in questo caso si è scelto di mantenere i valori  $N == R_1 == R_2 == 1$  e  $|Master| = 3$ , cambiando solo le tipologie di Slave per evidenziarne l'effetto sullo spazio degli stati.

Come visto analizzando la variazione delle tipologie di master, il SRG si dimostra molto più compatto rispetto al RG, mantenendo uno spazio degli stati notevolmente ridotto. Si nota infatti come con più di 5 tipologie di master il rapporto tra RG e SRG è di circa un'ordine di grandezza.

Si nota inoltre come l'aumento delle tipologie di slave non abbia influenzato lo spazio degli stati quanto l'aumento delle tipologie di master.

Effetto delle tipologie di slave sul numero di markings		
Tipologie Slave	Markings RG	Markings SRG
2	768	432
3	2688	864
4	7168	1440
5	22528	2160
6	56320	3024
7	163840	4032
8	393216	5184
9	1081344	6480

Table 3: Variazione dello spazio degli stati.

## 1.7 Considerazioni sulla Join

La rete utilizza due classi di colori e un color domain che è dato dal prodotto vettoriale delle due classi, permettendo così una differenziazione precisa delle richieste da parte dello slave, che le utilizza come una tuple  $\langle m, id \rangle$ .

Per ottenere una Join tra due processi che non sono stati creati dalla stessa Fork, bisogna che siano presenti nei posti Child1, Child2 due sottoprocessi (per posto) che abbiano:

1. Lo stesso tipo di master  $m_i$
2. Lo stesso tipo di slave  $ID_i$

Questo si verifica solo nel caso in cui vi siano più master e slave dello stesso tipo, ovvero con  $N > 1$  e  $R_1 > 1 || R_2 > 1$ . In tutti gli altri casi, le Join sono eseguite solo tra due sottoprocessi figli della stessa Fork.