

Contents

1 Rete A

M master identici e S slave identici di tipo 1.

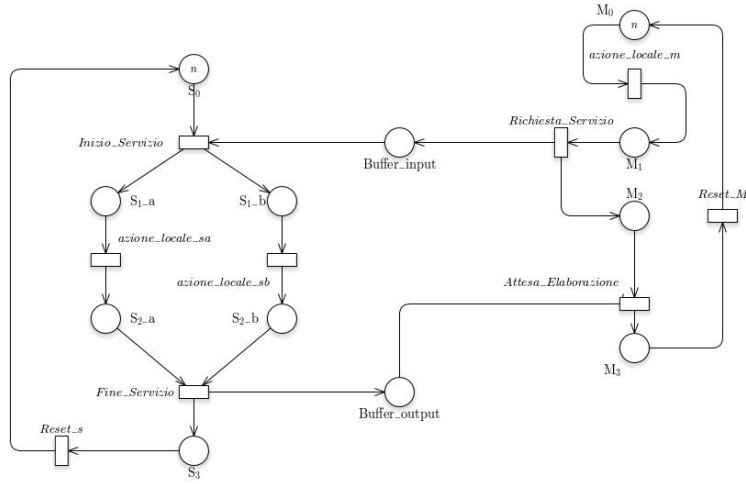


Figure 1: Modello della reteA

La figura rappresenta la rete di Petri P/T dell'esercizio A. Il master è modellato dai posti M_0, M_1, M_2, M_3 e dalle transizioni $Azione_{Locale}$, $Richiesta_{Servizio}$, $Attesa_{Elaborazione}$ e $Reset_M$. Lo slave è modellato dai posti $S_0, S_{1a}, S_{1b}, S_{2a}, S_{2b}$ e S_3 e dalle transizioni $Inizio_{Servizio}$, $Azione_{LocaleSa}$, $Azione_{LocaleSb}$, $Fine_{Servizio}$ e $Reset_s$. La richiesta del servizio verso lo slave è gestita attraverso due buffer, posti $Buffer_{Input}$ e posto $Buffer_{Output}$.

1.1 Risultati

Nella tabella vengono mostrate il numero di archi e di nodi al variare dei parametri M e S . Le cifre sono indicative dell'aumentare della dimensione dello spazio degli stati proporzionalmente al numero di marcature.

master, slaves	Nodi	Archi
1, 1	14	19
2, 2	94	222
3, 3	426	334
4, 4	1500	5610
5, 5	4422	18720
6, 6	11418	52998
7, 7	26598	132594
8, 8	57057	301158
9, 9	114400	632775
10, 10	216788	1246960
11, 11	391612	2328612
12, 12	678912	4153916
13, 13	1135668	7123272
14, 14	1841100	11802420
15, 15	2903124	18973020

1.2 Considerazioni su Fork/Join

Il modello non garantisce che avvenga il join di due processi dello stesso padre quando la marcatura degli slave e' maggiore di 2. Si puo' garantire che avvenga il join di due processi forcati dallo stesso padre nei seguenti modi:

- attraverso differenti strutture slaves
- usando reti WN

1.3 Riduzione

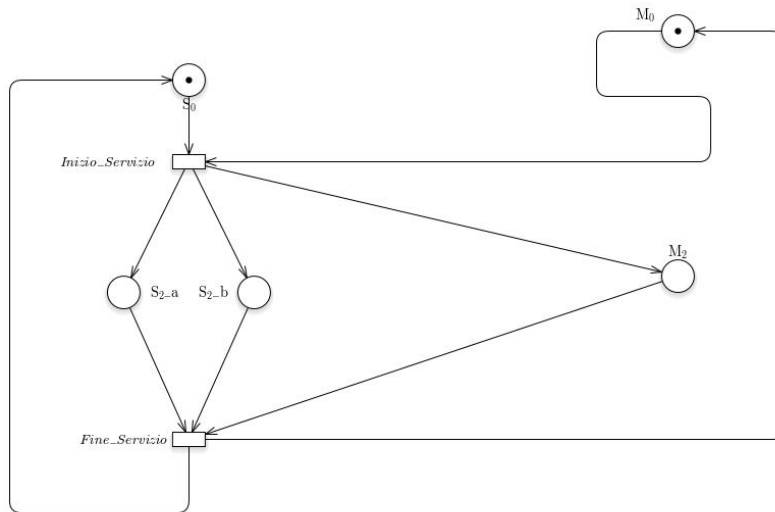
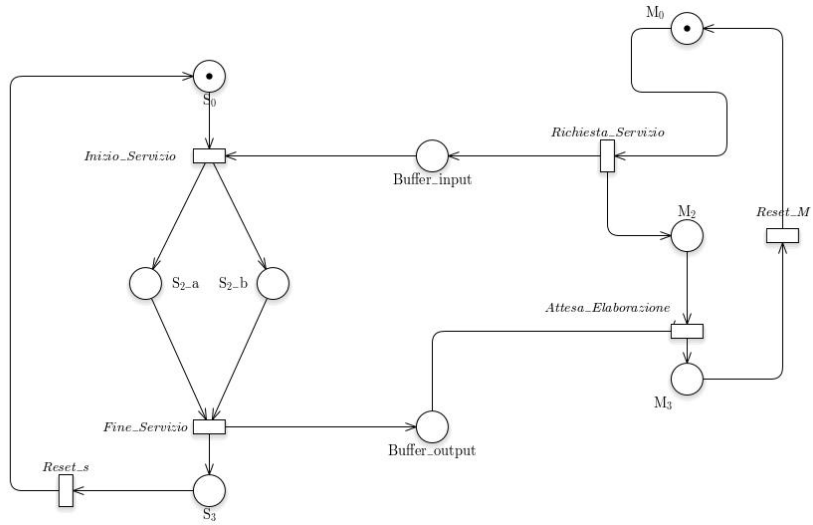
Una rete di petri puo' essere ridotta usando le seguenti tecniche:

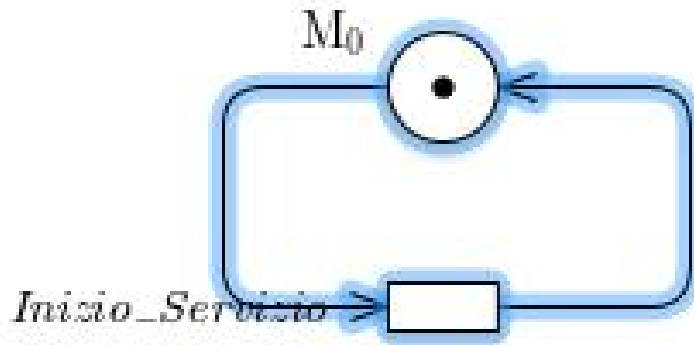
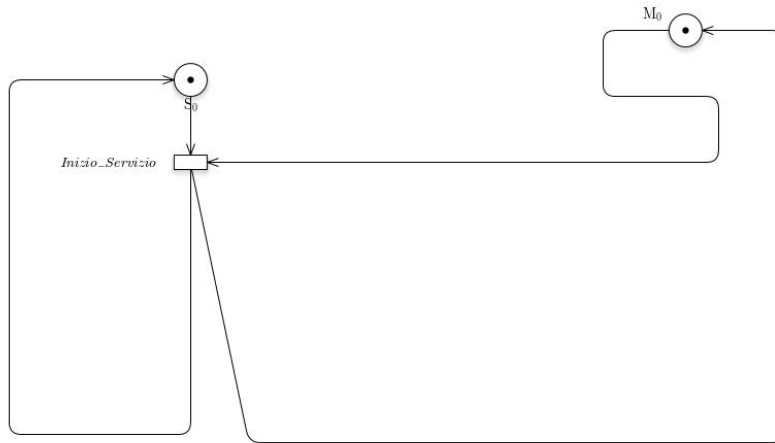
- fusione
- eliminazione
- rimozione dei loop

Nelle figure vengono mostrate alcune fasi di riduzione della rete in analisi; in ordine sono stati applicati:

- fusione di alcuni posti
- fusione di alcune transizioni

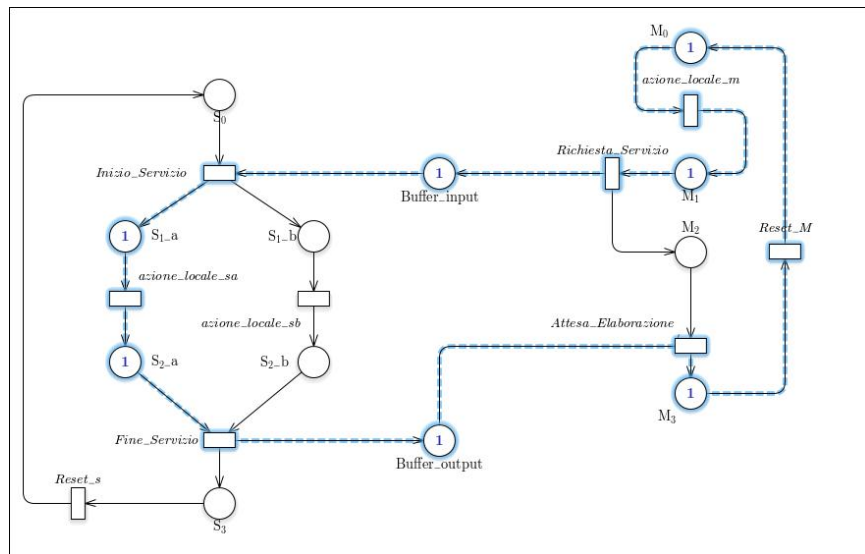
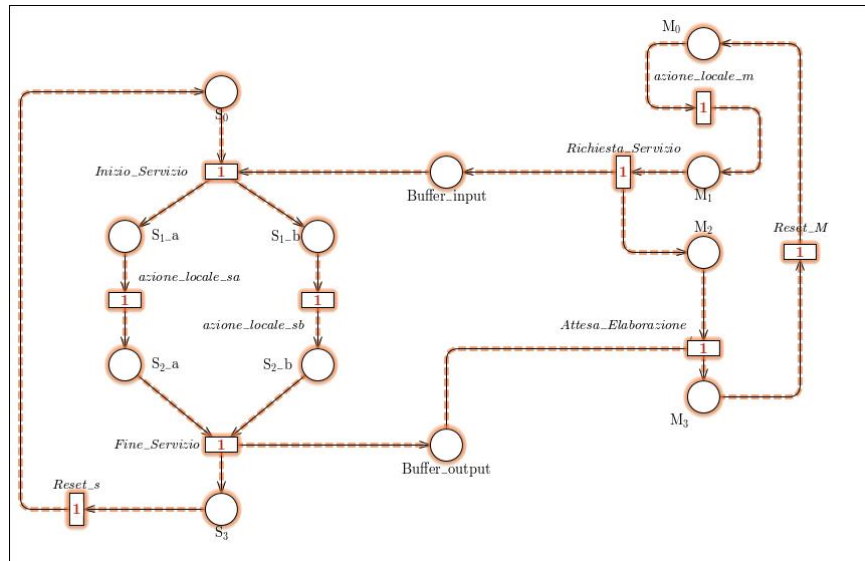
- eliminazione di alcuni posti
- eliminazione di alcune transizioni
- riduzione di self loop





1.4 TODO P e T invarianti

Tramite GreatSPN possiamo calcolare gli T- e P- semiflussi



Gli P-invarianti sono i seguenti:

$$S_0 + S_{1_a} + S_{2_a} + S_3$$

$$S_0 + S_{1_b} + S_{2_b} + S_3$$

$$M_0 + M_1 + M_2 + M_3$$

$$S_{1_a} + S_{2_a} + Buffer_output + Buffer_input + M_0 + M_1 + M_3$$

$$S_{1_b} + S_{2_b} + Buffer_output + Buffer_input + M_0 + M_1 + M_3$$

Il T-invariante e' il seguente:

$$Inizio_{servizio} + azione_{locale}_a + azione_{locale}_b + Fine_{servizio} + Reset_s + azione_{locale}_m + Richiesta_{servizio}$$

Dato che la reteA e' interamente coperta dagli P-semiflussi, possiamo affermare che la rete sia bounded. [] Deadlock [] Liveness

2 Rete B

M master identici, uno slave di tipo 1 e uno slave di tipo 1 scelti liberamente dai master.

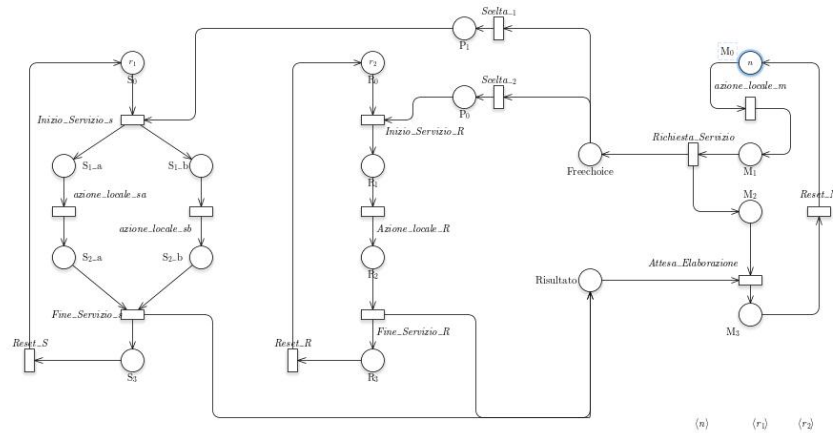


Figure 2: Modello della reteB

La figura rappresenta la rete di Petri P/T dell'esercizio B. Il master è modellato dai posti M0, M1, M2, M3 e dalle transizioni Azione_{Locale}, Richiesta_{Servizio}, Attesa_{Elaborazione} e Reset_M. Lo slave di tipo 1 è modellato dai posti S0, S1_a, S1_b, S2_a, S2_b e S3 e dalle transizioni Inizio_{Servizio}, Azione_{LocaleSa}, Azione_{LocaleSb}, Fine_{Servizio} e Reset_s. Lo slave di tipo 2 è modellato dai posti R0, R1_a, R1_b, R2_a, R2_b e R3 e dalle transizioni Inizio_{ServizioR}, Azione_{LocaleR}, Fine_{Servizio} e Reset_R. La richiesta del servizio verso lo slave scelto e' gestita attraverso due buffer, posti FreeChoice e Risultato.

2.1 Risultati

master, slaves	Stati	Archi
1, 2	40	76
2, 2	204	544
3, 2	728	2400
4, 2	2072	7896
5, 2	5040	21336
6, 2	10920	50064
7, 2	21648	105648
8, 2	39996	205260
9, 2	69784	373252
10, 2	116116	642928

Parametrizzando anche il numero di slaves:

master, slaves	Stati	Archi
1, 2	40	76
2, 2	204	544
4, 4	7265	32674
6, 6	113464	664234
8, 8	1073226	7405654
10, 10	7212128	55762000

2.2 Considerazioni su Fork/Join

Lo slave di tipo 1 processa una sola richiesta alla volta. Il master in attesa del risultato (M2) potrebbe ricevere il risultato di un lavoro richiesto da un altro master.

2.3 TODO P e T invarianti

Tramite GreatSPN possiamo calcolare gli T- e P- semiflussi

Dato che la reteB e' interamente coperta dagli P-semiflussi, possiamo affermare che la rete sia bounded. [] Deadlock [] Liveness

3 Rete C

Due master identici, uno slave di tipo 1 e uno slave di tipo 1 scelti liberamente dai master.

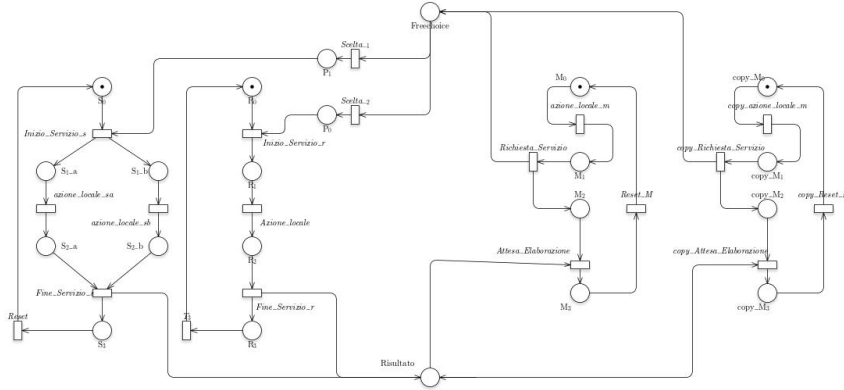
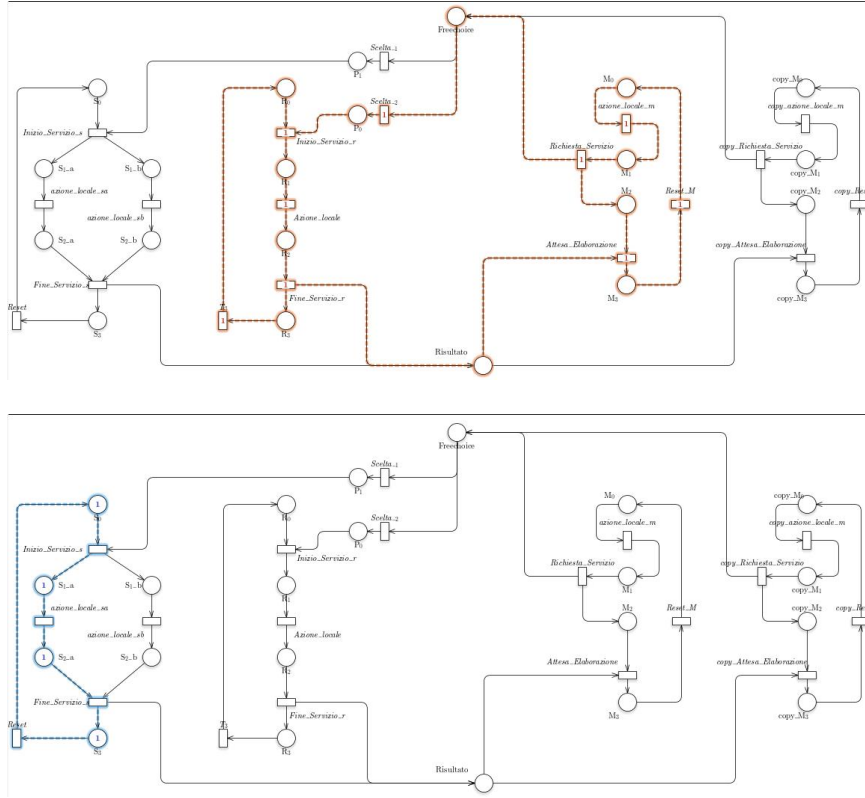


Figure 3: Modello della reteC

La figura rappresenta la rete di Petri P/T dell'esercizio C. Il master è modellato dai posti M0, M1, M2, M3 e dalle transizioni AzioneLocale, RichiestaServizio, AttesaElaborazione e ResetM. Lo slave di tipo 1 è modellato dai posti S0, S1a, S1b, S2a, S2b e S3 e dalle transizioni InizioServizio, AzioneLocaleSa, AzioneLocaleSb, FineServizio e ResetS (il secondo master e' una copia del primo). Lo slave di tipo 2 è modellato dai posti R0, R1a, R1b, R2a, R2b e R3 e dalle transizioni InizioServizioR, AzioneLocaleR, FineServizio e ResetR. La richiesta del servizio verso lo slave scelto e' gestita attraverso due buffer, posti FreeChoice e Risultato.

3.1 TODO P e T invarianti

Tramite GreatSPN possiamo calcolare gli T- e P- semiflussi



Gli P-invarianti sono i seguenti:

- $S_0 + S1_a + S2_a + S3$
- $S_0 + S1_b + S2_b + S3$
- $R_0 + R1 + R2 + R3$
- $M_0 + M1 + M2 + M3$
- $copyM_0 + copyM_1 + copyM_2 + copyM_3$
- $S1_a + S2_a + R1 + R2 + M_0 + M1 + M3 + Freechoice + P_0 + P1 + Risultato + copyM_0 + copyM_1 + copyM_3$
- $S1_b + S2_b + R1 + R2 + M_0 + M1 + M3 + Freechoice + P_0 + P1 + Risultato + copyM_0 + copyM_1 + copyM_3$

Gli T-invarianti sono i seguenti:

- $\text{Inizio}_{\text{Servizio}_r} + \text{Azione}_{\text{Locale}} + \text{Fine}_{\text{Servizio}_r} + T3 + \text{azione}_{\text{locale}_m} + \text{Richiesta}_{\text{Servizio}} + \text{Attesa}_{\text{Elaborazione}} + \text{Reset}_M + \text{Scelta}_1$
- $\text{Inizio}_{\text{Servizio}_s} + \text{Azione}_{\text{Localesa}} + \text{Azione}_{\text{Localesb}} + \text{Fine}_{\text{Servizio}_s} + T3 + \text{azione}_{\text{locale}_m} + \text{Richiesta}_{\text{Servizio}} + \text{Attesa}_{\text{Elaborazione}} + \text{Reset}_M + \text{Scelta}_1$
- $\text{Inizio}_{\text{Servizio}_r} + \text{Azione}_{\text{Locale}} + \text{Fine}_{\text{Servizio}_r} + T3 + \text{Scelta}_2 + \text{copy}_{\text{azione}_{\text{locale}_m}} + \text{copy}_{\text{Richiesta}_{\text{Servizio}}} + \text{copy}_{\text{Attesa}_{\text{Elaborazione}}} + \text{copy}_{\text{Reset}_m}$
- $\text{Inizio}_{\text{Servizio}_s} + \text{Azione}_{\text{Localesa}} + \text{Azione}_{\text{Localesb}} + \text{Fine}_{\text{Servizio}_s} + \text{Reset} + \text{Scelta}_1 + \text{copy}_{\text{azione}_{\text{locale}_m}} + \text{copy}_{\text{Richiesta}_{\text{Servizio}}} + \text{copy}_{\text{Attesa}_{\text{Elaborazione}}} + \text{copy}_{\text{Reset}_m}$

[] Deadlock [] Liveness

4 Rete D

Due master identici, uno slave di tipo 1 e uno slave di tipo 1 scelti associati ciascuno ad un master diverso.

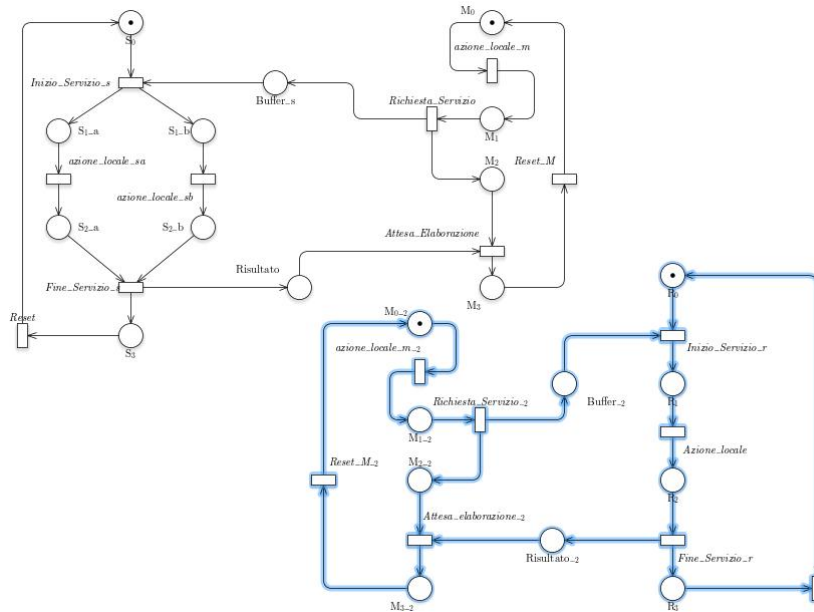
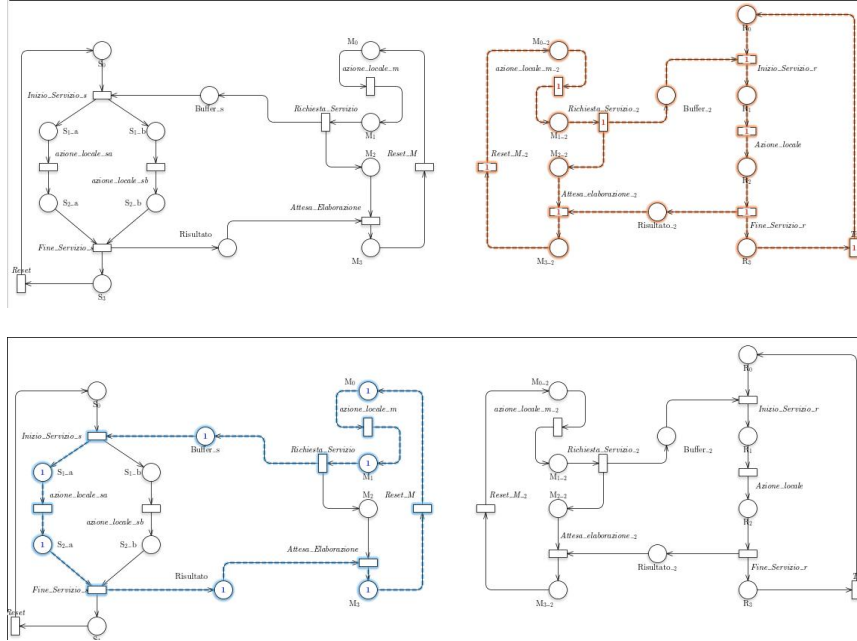


Figure 4: Modello della reteD

4.1 P e T invarianti

Tramite GreatSPN possiamo calcolare gli T- e P- semiflussi



Gli P-invarianti sono i seguenti:

- $S_0 + S1_a + S2_a + S3$
- $S_0 + S1_b + S2_b + S3$
- $R_0 + R1 + R2 + R3$
- $M_0 + M1 + M2 + M3$
- $S1_a + S2_a + M_0 + M1 + M3 + Buffer_s + Risultato$
- $S1_b + S2_b + M_0 + M1 + M3 + Buffer_s + Risultato$
- $M_{0_2} + M_{1_2} + M_{3_2}$
- $R1 + R2 + M_{0_2} + M_{1_2} + M_{3_2} + Buffer_2 + Risultato_2$

Gli T-invarianti sono i seguenti:

- $Inizio_{Servizio_s} + azione_{locale_s} + azione_{locale_s} + Fine_{Servizio_s} + Reset + azione_{locale_m} + Richiesta_{Servizio} + Attesa_{Elaborazione} + Reset_m$

- $\text{Inizio}_{\text{Servizio}_r} + \text{Azione}_{\text{locale}} + \text{Fine}_{\text{Servizio}_r} + T3 \text{ azione}_{\text{locale}m2} + \text{Richiesta}_{\text{Servizio}2} + \text{Attesa}_{\text{Elaborazione}2} + \text{Reset}_{m2}$

4.2 Decision Diagram

L'efficacia dei decision diagram sulla generazione dello stato degli spazi dipende fortemente dall'ordine delle variabili. Di seguito vengono mostrati i decision diagram usando per le assegnazioni i seguenti algoritmi:

- Sloan: un algoritmo di riduzione della banda di matrici sparse con una buona performance
- (advanced) Cuthill-McKee: un altro algoritmo di riduzione della banda di matrici sparse
- Tovchigrechko e Noack: due algoritmo appositamente ideati per le reti di Petri, anch'essi con una buona performance
- P-chaining: un algoritmo che sfrutta le informazioni strutturali della rete ma ha una bassa performance
- Gradient-P
- Gibbs-Poole-Stockmeier: un altro algoritmo matriciale che nella rete in analisi ha restituito il risultato peggiore

