Esercizio Reti di Petri Produttori e Consumatori

Lorenzo Dentis

December 14, 2022

1 Primo setting: 1 produttore, 1 consumatore, 1 buffer a N posizioni

1.1 Primo setting:

La Petri Net in Figura 1 mostra una basilare rete "producer-consumer" divisibile in 3 sezioni.

- Buffer: il buffer ad accesso casuale ad N posizioni è implementato tramite due posti, *buffer* e *buffer_capacity*. Il primo svolge il ruolo di limite minimo, quanti prodotti sono presenti all'interno del buffer; il secondo fa da limite massimo, impedendo che il produttore inserisca più di N elementi.
- Producer: Il produttore genera l'elemtento e lo inserisce nel buffer tramite la transizione *place*, tale transizione non è abilitata se non sono presenti token nel posto *buffer_capacity*, cioè il buffer è pieno
- Consumer: Il consumatore preleva un elemento dal buffer (solo se è presente un token nel posto *buffer*) e lo "consuma".

Il reachability graph mostra un totale di 8 markings quando $N = 1$, il numero di markings aumenta linearmente rispetto al valore N.

$$markings = 4 * (N + 1)$$

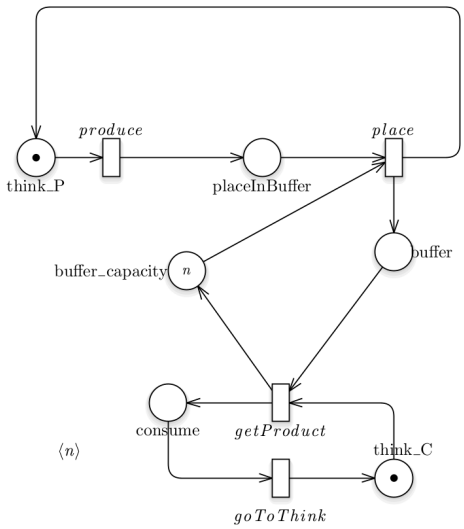

Figure 1: Setting 1

1.2 Rete di petri colorata

La rappresentazione tramite rete di petri colorata si rivela praticamente identica al caso precendente, a parte la differente notazione le figure 1 e 2 sono identiche.

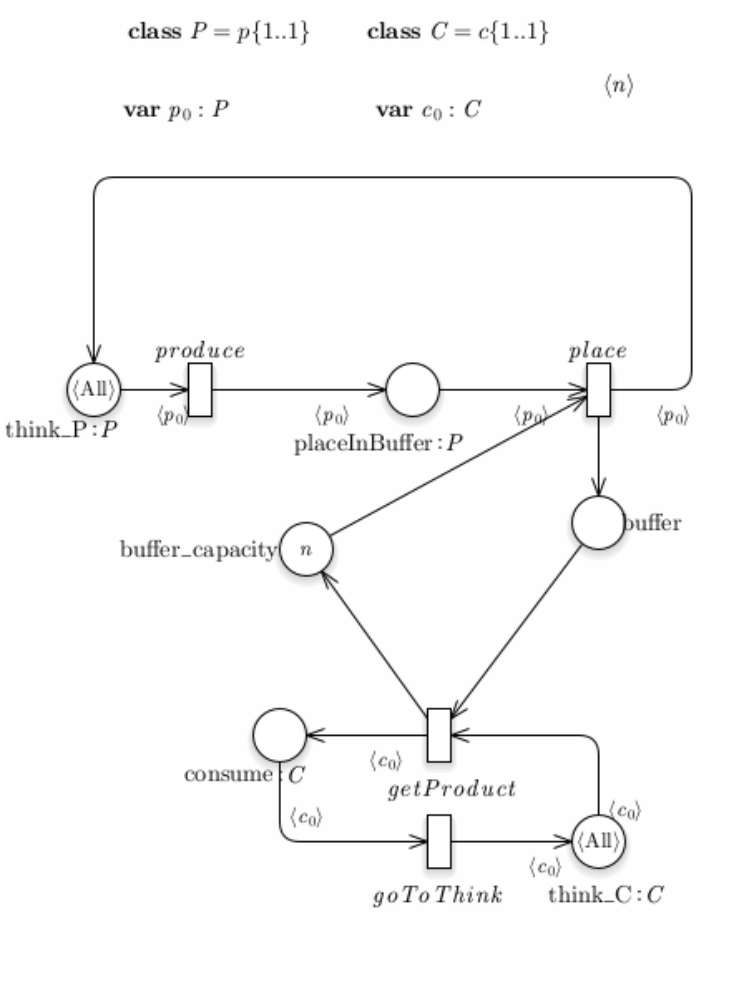

Figure 2: Setting 1 CPN

Il reachability graph mostra un totale di 8 markings quando $N = 1$, il numero di markings aumenta linearmente rispetto al valore N. Perfettamente in linea con quanto rileato nel caso precendente.

$$markings = 4 * (N + 1)$$

Non vi è alcuna differenza tra il RG ed il SRG, essendoci un singolo token nel consumatore ed un singolo token nel produttore non viene generata alcun marking "simbolico". Una eventuale rete unfolded sarebbe identica a questa rete.

2 Secondo setting: 1 produttore, 2 consumatori, 1 buffer a N posizioni

2.1 Secondo setting: scalatura di marcatura

La soluzione con la scalatura di marcatura richiede di modificare la marcatura del posto *Think_C*, portando il numero di token a due.
L'implementazione è triviale ed il numero di markings mantiene una relazione di linearità rispetto ad N. Per quanto la costante di proporzionalità sia superiore rispetto al caso precedente non vi è un eccessivo incremento di markings anche con valori di N alti. Ad esempio con $N = 1$ il reachability graph presenta 12 markings mentre con $N = 30$ ne sono presenti 182.
La figura 3 mostra la rete di Petri risultante, praticametente identica alla rete tratta nella sezione precendente (setting 1).

$$markings = 6 * (N + 1)$$

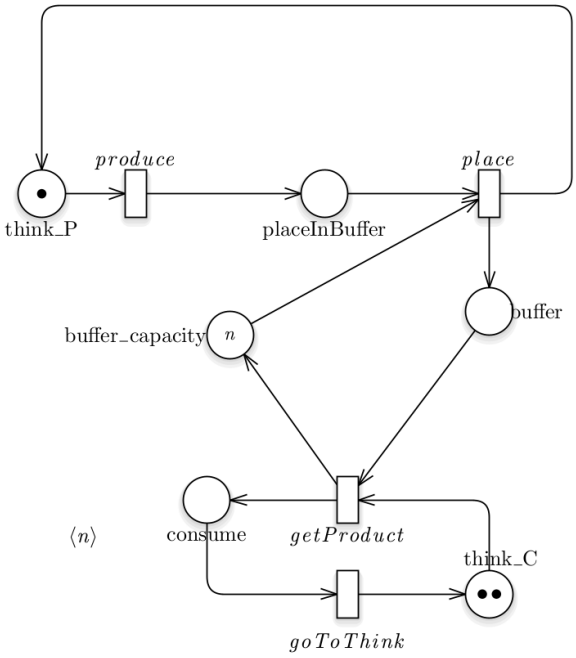

Figure 3: Setting 2 con scalatura di marking

2.2 Secondo setting: replicazione di sottoreti

La soluzione tramite la replicazione della sottorete consumatore presenta una implementazione più complessa e più "scomoda", la rete risultate per quanto meno chiara è ancora di facile comprensione.
Il reachability graph è più esteso, con un solo posto nel buffer si ottengono 16 markings e, a differenza dell secondo setting (sezione 2.1), la costante di proporzionalità è 8. Questo porta ad un totale di 248 markings, contro i 182 del caso precedente, quando $N = 30$

$$markings = 8 * (N + 1)$$

.

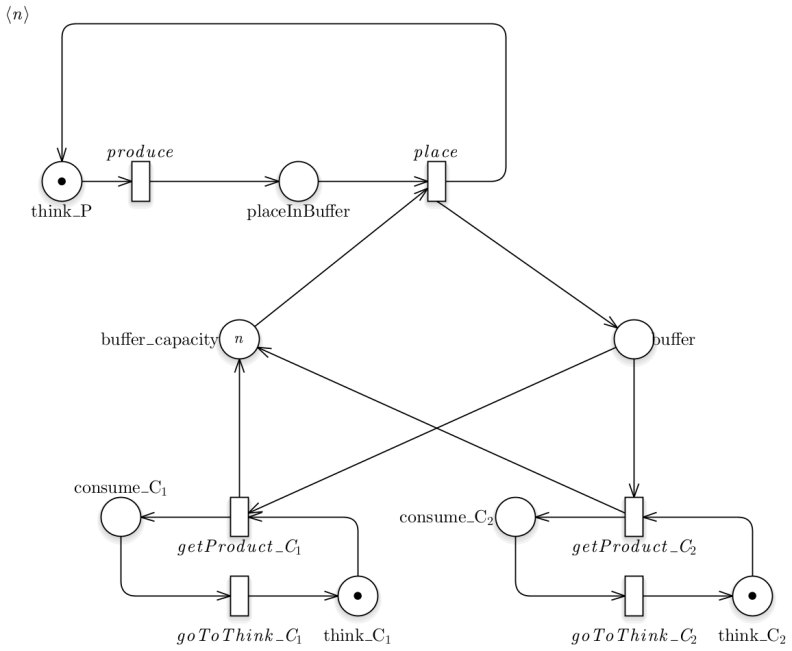

Figure 4: Setting 2 tramite replicazione

2.3 Secondo setting: rete di petri colorata

Anche questa implementazione si presenta sostanzialmente identica all sezione 2.1. La marcatura iniziale del posto *Think_C* viene solamente incrementata di uno e la relazione di linearità del numero di markings del reachability graph rispetto al numero di posti nel buffer viene rispettata.
Infatti come ci si aspetterebbe il numero di markings del reachability graph simbolico è identico al numero di marking ottenuto tramite scalatura di marking La sostanziale differenza, che è anche il vantaggio che fornisce l'uso delle Reti

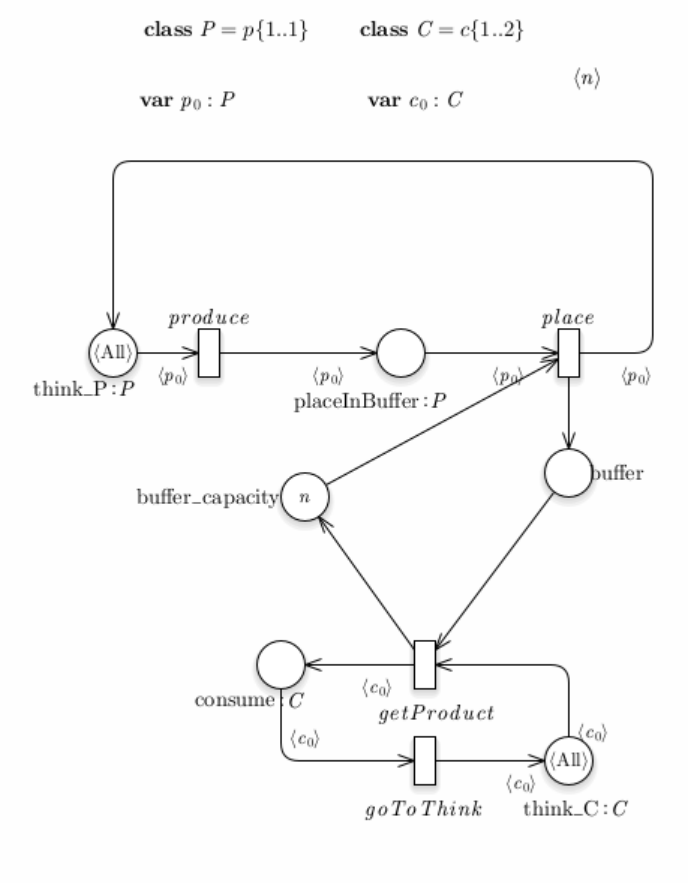

Figure 5: Setting 2 tramite rete di petri colorata

di Petri colorate, sta nella possibilità di distinguere i due consumatori.
I consumatori, che nella sezione 2.1 erano indistinguibili, ora sono identificati come $< c_1 >$ e $< c_2 >$. Questo senza però dover ricorrere alla scomoda costruzione della sezione 2.2. La dimensione del Reachability Graph è aumentata di conseguenza,con due soli posti nel buffer sono presenti 24 markings, con 3 posti 32 markings e con 4 posti 40. Esattamente come nel caso 2.2

$$markings = 8 * (N + 1)$$

Questo poichè grazie alla colorazione dei posti è possibile trattare ogni consumatore come se fosse una rete differente, portando il numero di stati del reachability graph allo stesso valore ottenuto separando fisicamente le due reti. Facendo l'unfolding di questa rete si ottiene la rete ottenuta tramite replicazione di sottoretti.

3 Terzo setting: P produttori, C consumatori,1 buffer a N posizioni

3.1 Terzo setting: scalatura di marcatura

Anche questo setting è di semplice implementazione, come si può vedere in Figura 6.Si tratta solamente di inserire dei parametri al posto del marking di Producer e Consumer.

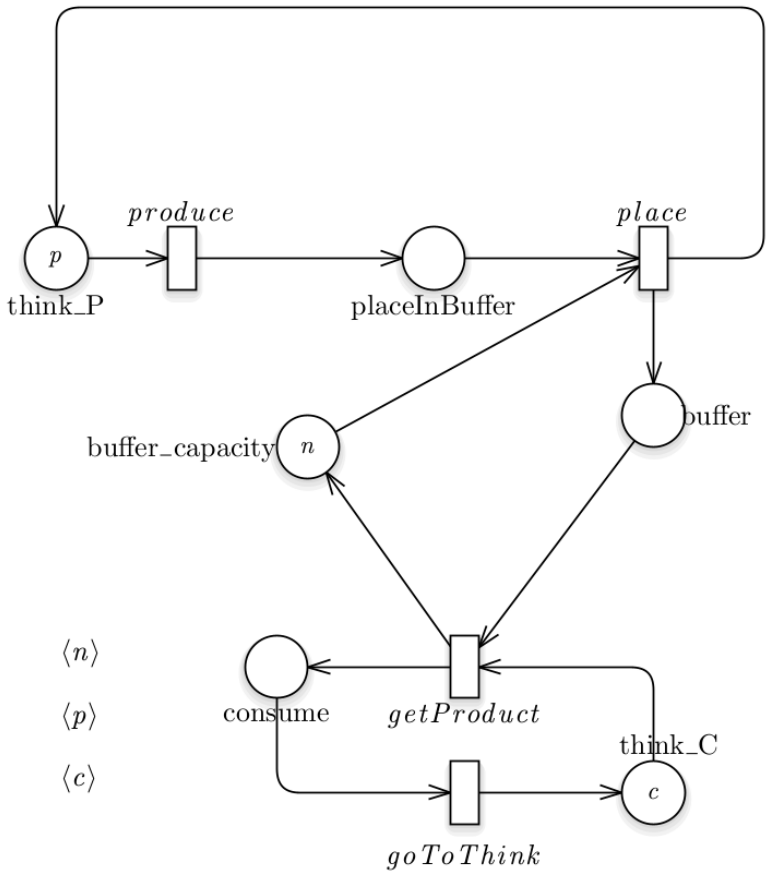

Figure 6: Setting 3 tramite la scalatura di marcatura

Calcolare come varia esattamente il numero di markings rispetto ai valori P, C ed N è complesso, ma al crescere di uno qualsiasi dei parametri l'aumento del numero di markings è comunque limitato, ad esempio impostando un buffer di 4 posti, 4 consumatori e 4 produttori vengono generati 125 markings

3.2 Terzo setting: replicazione di sottoreti

La metodologia di replicazione delle sottoreti è inadatta a implementare un sistema a P produttori e C consumatori, in quanto se il numero di produttori e consumatori non è noto è impossibile generare manualmente il numero corretto di sottoreti.
Una implementazione teorica può essere vista in Figura 7, ma è appunto solo teorica. Dato che non ci sono **realmente** P produttori e C consumatori non si possono usare i tool di analisi di GreatSPN.

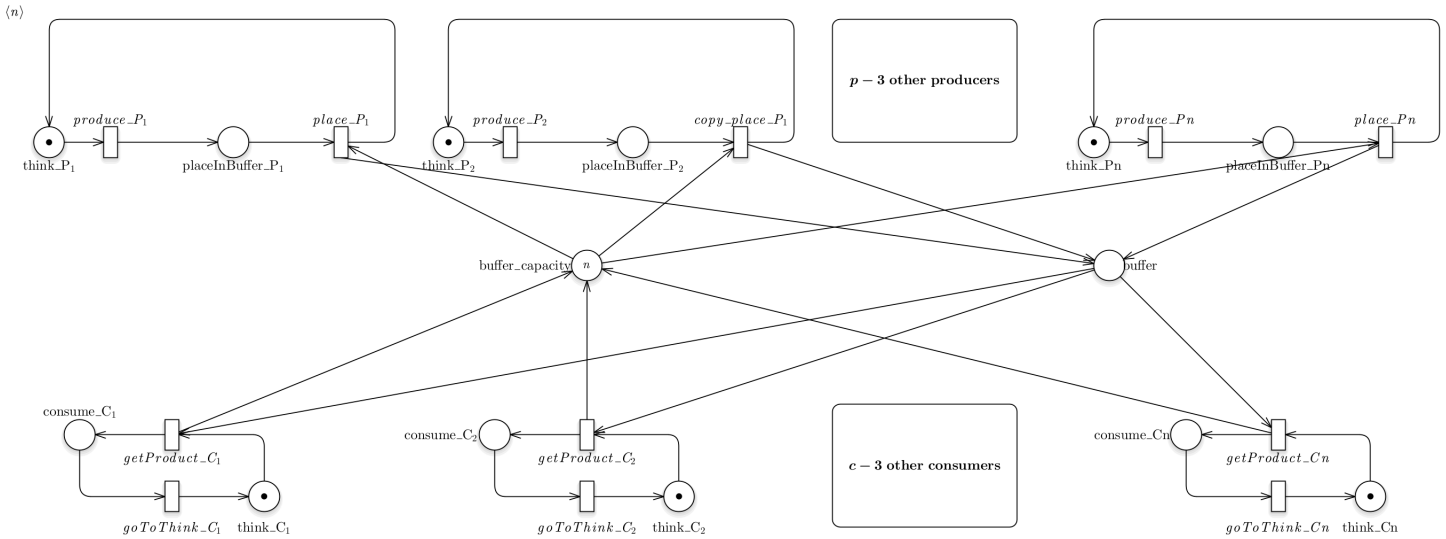

Figure 7: Setting 3 tramite la scalatura di marcatura

Comunque volendo fare un'analisi della dimensione del Reachability graph possiamo prendere l'esempio in figura 6 e confrontarlo con la stessa situazione ma con la scalatura di marcatura, quindi 3 Producer, 3 Consumers e N posti nel buffer. In questo caso la differenza è notevole, il numero di markings aumenta molto utilizzando il metodo di replicazioni e cresce molto più velocemente, i risultati possono essere osservati in figura 8.

n	1	2	3	4	5	6	7	8	9	10
marking	32	48	64	80	96	112	128	144	160	176
ripetizione	128	192	256	320	384	448	512	576	640	704

Figure 8: Setting 3 tramite la scalatura di marcatura

Da questi dati si può concludere che la strategia di replicazione di sottoreti non è vantagggiosa nel dal punto di vista implementativo, ne dal punto di vista della dimensione del reachability graph.

3.3 Terzo setting: rete di Petri colorata

Come nel caso precendete (sezione 2), l'implentazione tramite rete di Petri colorata combina il vantaggio implementativo della scalatura tramite incremento di marcatura con la distinguibilità della replicazione di sottoreti.
Tutti i consumatori sono identificati da un valore $< c_i >$ $con\ i \in [1, C]$ ed i produttori dal valore $< p_i >$ $con\ i \in [1, P]$ ed è quindi possibile seguirne lo sviluppo e le scelte. Questo pur mantenendo la raffigurazione chiara e pulita della strategia si scalatura tramite incremento di marking.

Anche in questo caso il numero di marking distinti del reachability graph segue perfettamente il trend della replicazione di sottoreti (vedere figura 8).
Il *Symbolic Reachability Graph* è identico al *reachability graph* della rete generata tramite scalatura di marking, mentre il *reachability graph* si comporta esattamente come quello che si otterrebbe della rete ottenuta tramite replicazione.
Anche in questo caso effettuare unfolding di questa rete colorata porterebbe alla rete della sezione 3.2.

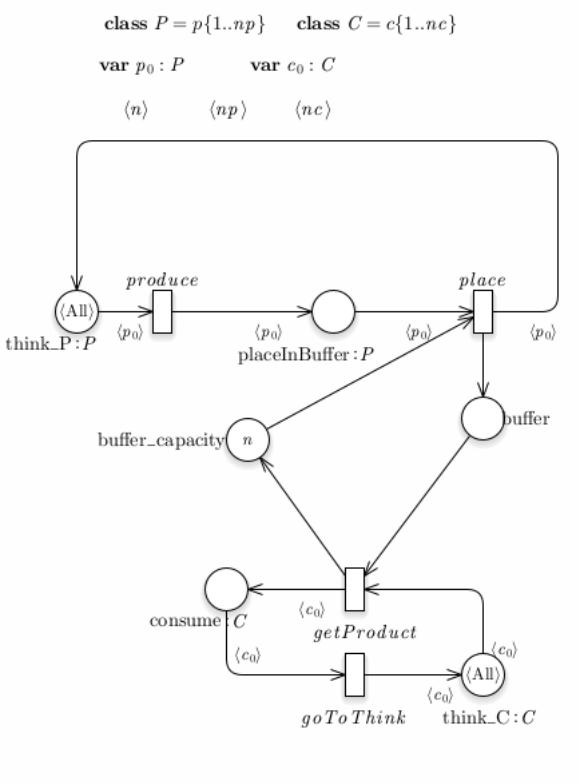

Figure 9: Setting 3 rete di Petri colorata