

Manuale d'uso per il Toolbox Matlab per "unfolding"

19 ottobre 2005

Indice

1	Funzione unfolding_n	4
1.1	Sintassi	4
1.2	Esempio di funzionamento	5
2	Funzione order1	7
2.1	Sintassi	7
2.2	Esempio di funzionamento	8
3	Funzione order2	9
3.1	Sintassi	9
3.2	Esempio di funzionamento	10
4	Funzione McMillan	11
4.1	Sintassi	11
4.2	Esempio di funzionamento	12
5	Funzioni unfolding_graph, graph1, graph2, graphMcM	13
5.1	Sintassi	13
5.2	Esempio di funzionamento	14

<i>INDICE</i>	2
6 Funzione reach_u	15
6.1 Sintassi	15
6.2 Esempio di funzionamento	16
7 Funzione Filosofi	17
7.1 Sintassi	17
7.2 Esempio di funzionamento	17
A Figure	18

Prefazione

Questo manuale non è di tipo esaustivo e ha solo lo scopo di descrivere il funzionamento pratico di ogni singola funzione del pacchetto Matlab per il dispiegamento (unfolding). Per una trattazione più estesa è consigliabile leggere *Analisi di reti posto/transizione mediante ordine parziale: sviluppo di un Toolbox Matlab per unfolding* di Caterina Carboni e Federica Lamon.

Capitolo 1

Funzione unfolding_n

Supponiamo di avere una rete di Petri sana e ordinaria $\langle N, M_0 \rangle$ e di voler determinare il suo dispiegamento $\tilde{N}(M_0)$. Non preoccupiamoci per ora di quale prefisso finito scegliere; costruiremo il dispiegamento fermandolo ad un desiderato livello di profondità (o numero di livelli).

1.1 Sintassi

L'interfaccia della funzione è la seguente:

```
[Pre_u, Post_u, p_tier, t_tier, Pexp, Texp]=unfolding_n(Pre, Post, M0, f)
```

I parametri in ingresso sono dunque:

- `Pre`: la matrice di pre-incidenza della rete $\langle N, M_0 \rangle$.
- `Post`: la matrice di post-incidenza della rete $\langle N, M_0 \rangle$.
- `M0`: la marcatura iniziale della rete $\langle N, M_0 \rangle$.
- `f`: livello desiderato di profondità della rete di dispiegamento.

Le uscite saranno invece:

- `Pre_u`: la matrice di pre-incidenza della rete $\tilde{N}(M_0)$.
- `Post_u`: la matrice di post-incidenza della rete $\tilde{N}(M_0)$.

- p_tier : quest'uscita è strutturata come un array di celle: nella i -esima cella sono indicate le etichette dei posti del $(i - 1)$ -esimo livello della rete di dispiegamento che rendono conto della corrispondenza con i posti appartenenti alla rete iniziale (si noti che gli indici in Matlab partono da uno mentre in questo lavoro la prima fila di posti è stata convenzionalmente indicata come livello 0).
- t_tier : anche quest'uscita è strutturata come un array di celle: nella i -esima cella sono indicate le etichette delle transizioni del $(i - 1)$ -esimo livello della rete di dispiegamento che rendono conto della corrispondenza con le transizioni appartenenti alla rete iniziale.
- P_{exp} : il suo utilizzo è assai simile al P_{exp} presentato nella procedura proposta nel Capitolo 3. E' un vettore al cui interno si trovano tutti i posti della rete di dispiegamento ognuno etichettato con il nome del corrispondente posto della rete di partenza.
- T_{exp} : Analogamente al P_{exp} si tratta di un vettore al cui interno si trovano tutte le transizioni della rete di dispiegamento ognuna etichettata con il nome della corrispondente transizione della rete di partenza.

Ad ogni nodo della rete di dispiegamento sono associate due numerazioni: la prima considera i posti del dispiegamento in ordine crescente come se questi appartenessero a una rete di Petri indipendente e sono quelli che consentono di generare le matrici Pre_u e $Post_u$; la seconda, definita tramite le uscite P_{exp} e T_{exp} , tiene invece conto della numerazione originale ed assolve alla funzione di etichettatura $\ell : (\tilde{P} \rightarrow P) \cup (\tilde{T} \rightarrow T)$ descritta nel precedente capitolo. La corrispondenza tra le due è sempre definita. Gli indici dei vettori P_{exp} , T_{exp} rappresentano infatti la numerazione del dispiegamento mentre il contenuto delle rispettive celle mostra a quale nodo della rete originale corrisponde quel nodo della rete di dispiegamento. Per evitare l'ambiguità, causata dal fatto che a più posti della rete di dispiegamento corrisponde lo stesso posto della rete originale, è stata introdotta in tutte le rappresentazioni grafiche una particolare numerazione per distinguere i vari posti che avrebbero altrimenti la stessa etichetta. Ad esempio, nella rete di dispiegamento in Appendice in Figura A.2 i posti 5, 5.1, 5.2, 5.3, 5.4 etc. corrisponderanno allo stesso posto p_5 della rete di partenza rappresentata in appendice in Figura A.1 (questa numerazione vale per tutte le reti di dispiegamento)

1.2 Esempio di funzionamento

Vengono proposti dei demo (le reti a cui si riferiscono sono riportate in Appendice A) eseguibili direttamente sul prompt dei comandi di Matlab digitando uno dei seguenti comandi:

```
>>demo1_unfolding_n (per la rete in Figura A.1)
```

>>demo2_unfolding_n (per la rete in Figura A.6)

Capitolo 2

Funzione order1

Tale funzione costruisce il primo ordine di dispiegamento di una rete posto/transizione. Per riuscire ad ottenere tale prefisso è stata presa la funzione `unfolding_n` così come poc' anzi presentata e ad essa è stata aggiunta una funzione di controllo che consenta di capire se la transizione corrente è di cut-off (e quindi la procedura deve essere fermata) o se invece non lo è (e dunque si può andare avanti nella costruzione del dispiegamento).

2.1 Sintassi

La funzione che costruisce il primo ordine di dispiegamento ha le seguente interfaccia:

```
[Pre_u, Post_u, p_tier, t_tier, Pexp, Texp, cut_off1, mat_conc, U] =  
=order1 (Pre, Post, M0)
```

I parametri in ingresso sono :

- `Pre`: la matrice di pre-incidenza della rete $\langle N, M_0 \rangle$
- `Post`: la matrice di post-incidenza della rete $\langle N, M_0 \rangle$
- `M0`: la marcatura iniziale della rete $\langle N, M_0 \rangle$

Le uscite saranno invece:

- `Pre_u`: la matrice di pre-incidenza della rete $\tilde{N}(M_0)$.
- `Post_u`: la matrice di post-incidenza dell rete $\tilde{N}(M_0)$.

- `p_tier`: quest'uscita è strutturata come un array di celle: nella i -esima cella sono indicate le etichette dei posti del $(i - 1)$ -esimo livello della rete di dispiegamento che rendono conto della corrispondenza con i posti appartenenti alla rete iniziale (si noti che gli indici in Matlab partono da uno mentre in questo lavoro la prima fila di posti è stata convenzionalmente indicata come livello 0).
- `t_tier`: anche quest'uscita è strutturata come un array di celle: nella i -esima cella sono indicate le etichette delle transizioni del $(i - 1)$ -esimo livello della rete di dispiegamento che rendono conto della corrispondenza con le transizioni appartenenti alla rete iniziale.
- `Pexp`: il suo utilizzo è assai simile al `Pexp` presentato nella procedura proposta nel Capitolo 3. E' un vettore al cui interno si trovano tutti i posti della rete di dispiegamento ognuno etichettato con il nome del corrispondente posto della rete di partenza.
- `Texp`: Analogamente al `Pexp` si tratta di un vettore al cui interno si trovano tutte le transizioni della rete di dispiegamento ognuna etichettata con il nome della corrispondente transizione della rete di partenza.
- `cut_off1`, che contiene le transizioni di cut off del primo ordine di dispiegamento. Esse seguono una numerazione progressiva, non tenendo conto cioè della corrispondenza con la rete originale.
- `U`, che contiene tutte le uscite già elencate con il vantaggio di contenerle tutte in un unico array di celle e di renderle quindi disponibili sotto un'unica variabile nel momento in cui si utilizza il dispiegamento per altre analisi.

2.2 Esempio di funzionamento

Vengono proposti dei demo (le reti a cui si riferiscono sono riportate in Appendice A) eseguibili direttamente sul prompt dei comandi di Matlab digitando uno dei seguenti comandi:

```
>>demo1_order1 (per la rete in Figura A.1)
```

```
>>demo2_order1 (per la rete in Figura A.6)
```

```
>>demo3_order1 (per la rete in Figura A.4)
```

Capitolo 3

Funzione order2

Tale funzione costruisce il secondo ordine di dispiegamento di una rete posto/transizione.

3.1 Sintassi

L'interfaccia della funzione è la seguente:

```
[Pre_u, Post_u, p_tier, t_tier, Pexp, Texp, cut_off1, cut_off2] =  
=order2 (Pre, Post, M0)
```

I parametri in ingresso sono :

- Pre: la matrice di pre-incidenza della rete $\langle N, M_0 \rangle$
- Post: la matrice di post-incidenza della rete $\langle N, M_0 \rangle$
- M0: la marcatura iniziale della rete $\langle N, M_0 \rangle$

Le uscite saranno invece:

- Pre_u: la matrice di pre-incidenza della rete $\tilde{N}(M_0)$.
- Post_u: la matrice di post-incidenza dell rete $\tilde{N}(M_0)$.
- p_tier: quest'uscita è strutturata come un array di celle: nella *i-esima* cella sono indicate le etichette dei posti del *(i - 1)-esimo* livello della rete di dispiegamento che rendono conto della corrispondenza con i posti appartenenti alla rete iniziale (si noti

che gli indici in Matlab partono da uno mentre in questo lavoro la prima fila di posti è stata convenzionalmente indicata come livello 0).

- `t_tier`: anche quest'uscita è strutturata come un array di celle: nella i -esima cella sono indicate le etichette delle transizioni del $(i - 1)$ -esimo livello della rete di dispiegamento che rendono conto della corrispondenza con le transizioni appartenenti alla rete iniziale.
- `Pexp`: il suo utilizzo è assai simile al `Pexp` presentato nella procedura proposta nel Capitolo 3. E' un vettore al cui interno si trovano tutti i posti della rete di dispiegamento ognuno etichettato con il nome del corrispondente posto della rete di partenza.
- `Texp`: Analogamente al `Pexp` si tratta di un vettore al cui interno si trovano tutte le transizioni della rete di dispiegamento ognuna etichettata con il nome della corrispondente transizione della rete di partenza.
- `cut_off1`, che contiene le transizioni di cut off del primo ordine di dispiegamento. Esse seguono una numerazione progressiva, non tenendo conto cioè della corrispondenza con la rete originale.
- `cut_off2` che rappresenta il vettore contenente le transizioni di cut-off del secondo ordine di dispiegamento.

3.2 Esempio di funzionamento

Vengono proposti alcuni demo (le reti a cui si riferiscono sono riportate in Appendice A) eseguibili direttamente sul prompt dei comandi di Matlab digitando il seguente comando:

```
>>demo1_order2 (per la rete in Figura A.1)
```

```
>>demo2_order2 (per la rete in Figura A.6)
```

Capitolo 4

Funzione McMillan

Tale funzione costruisce il prefisso finito di McMillan di una rete posto/transizione.

4.1 Sintassi

L'interfaccia della funzione è:

```
[Pre_u, Post_u, p_tier, t_tier, Pexp, Texp, cut_off, cut_off_ord]=  
=McMillan (Pre, Post, M0)
```

I parametri in ingresso sono :

- Pre: la matrice di pre-incidenza della rete $\langle N, M_0 \rangle$
- Post: la matrice di post-incidenza della rete $\langle N, M_0 \rangle$
- M0: la marcatura iniziale della rete $\langle N, M_0 \rangle$

Le uscite saranno invece:

- Pre_u: la matrice di pre-incidenza della rete $\tilde{N}(M_0)$.
- Post_u: la matrice di post-incidenza dell rete $\tilde{N}(M_0)$.
- p_tier: quest'uscita è strutturata come un array di celle: nella *i-esima* cella sono indicate le etichette dei posti del *(i - 1)-esimo* livello della rete di dispiegamento che rendono conto della corrispondenza con i posti appartenenti alla rete iniziale (si noti

che gli indici in Matlab partono da uno mentre in questo lavoro la prima fila di posti è stata convenzionalmente indicata come livello 0).

- `t_tier`: anche quest'uscita è strutturata come un array di celle: nella i -esima cella sono indicate le etichette delle transizioni del $(i - 1)$ -esimo livello della rete di dispiegamento che rendono conto della corrispondenza con le transizioni appartenenti alla rete iniziale.
- `Pexp`: il suo utilizzo è assai simile al `Pexp` presentato nella procedura proposta nel Capitolo 3. E' un vettore al cui interno si trovano tutti i posti della rete di dispiegamento ognuno etichettato con il nome del corrispondente posto della rete di partenza.
- `Texp`: Analogamente al `Pexp` si tratta di un vettore al cui interno si trovano tutte le transizioni della rete di dispiegamento ognuna etichettata con il nome della corrispondente transizione della rete di partenza.
- `cut_off` con cui si indica il vettore contenente le transizioni di cut off secondo McMillan.
- `cut_off_ord` con cui si indica il vettore contenente le transizioni di cut off secondo McMillan numerate in ordine crescente.

4.2 Esempio di funzionamento

Vengono proposti alcuni demo (le reti a cui si riferiscono sono riportate in Appendice A) eseguibili direttamente sul prompt dei comandi di Matlab digitando il seguente comando:

```
>>demo1_McMillan (per la rete in Figura A.3)
```

```
>>demo2_McMillan (per la rete in Figura A.1)
```

Capitolo 5

Funzioni `unfolding_graph`, `graph1`, `graph2`, `graphMcM`

Tali funzioni costituiscono un ausilio grafico per disegnare la rete di dispiegamento dei vari prefissi finiti.

5.1 Sintassi

L'interfaccia della funzione che permette di disegnare il generico dispiegamento è:

```
unfolding_graph (Pre, Post, M0, f)
```

L'interfaccia della funzione che permette di disegnare il primo ordine di dispiegamento è:

```
graph1 (Pre, Post, M0)
```

L'interfaccia della funzione che permette di disegnare il secondo ordine di dispiegamento è:

```
graph2 (Pre, Post, M0)
```

L'interfaccia della funzione che permette di disegnare il prefisso finito di McMillan è:

```
graphMcM (Pre, Post, M0)
```

I parametri in ingresso sono:

- `Pre`: la matrice di pre-incidenza della rete $\langle N, M_0 \rangle$
- `Post`: la matrice di post-incidenza della rete $\langle N, M_0 \rangle$

- M_0 : la marcatura iniziale della rete $\langle N, M_0 \rangle$
- f : il numero di *tier* a cui si desidera fermare il dispiegamento.

Tutte le funzioni contengono al loro interno le funzioni descritte in precedenza che costruiscono il generico dispiegamento, il primo e il secondo ordine di dispiegamento e il prefisso finito di McMillan tramite le quali è possibile visualizzare lo schema che permetterà di disegnare le reti corrispondenti.

5.2 Esempio di funzionamento

Vengono proposti alcuni demo (le reti a cui si riferiscono sono riportate in Appendice A), uno per ogni funzione descritta precedentemente, eseguibili direttamente sul prompt dei comandi di Matlab digitando uno dei seguenti comandi:

```
>>demo_unfolding_graph (per la rete in Figura A.6)
```

```
>>demo_graph1 (per la rete in Figura A.5)
```

```
>>demo_graph2 (per la rete in Figura A.4)
```

```
>>demo_graphMcM (per la rete in Figura A.4)
```

Capitolo 6

Funzione `reach_u`

Per integrare i programmi finora presentati che, come già detto, erano focalizzati a generare il dispiegamento, è stato sviluppato un programma il quale, lavorando sui risultati ottenuti, riesce ad estrapolare da essi alcune importanti informazioni sul comportamento della rete.

La funzione che verrà analizzata è `reach_u` e può essere utilizzata sostanzialmente sotto quattro differenti modalità. Il suo principale compito è quello di riconoscere se una certa marcatura in ingresso appartiene o meno allo spazio di stato della rete. Oltre a ciò, questa funzione determina anche a quali posti del dispiegamento questa marcatura corrisponde e qual è la sequenza di transizioni che devono scattare perché possa essere raggiunta.

6.1 Sintassi

Nel caso più generale l'interfaccia è:

```
[ret, p_corr, X]=reach_u(M0, M, U, opt)
```

Gli ingressi sono:

- `M0`: la marcatura iniziale.
- `M`: la generica marcatura di cui si vuole analizzare la raggiungibilità.
- `U`: l'uscita complessiva della funzione `order1`.
- `opt`: può assumere un valore tra 1 e 3 e serve per scegliere sotto quale modalità si vuole utilizzare il programma. In particolare:

- (a) Se `opt` non compare tra gli ingressi la funzione riconosce solo se la marcatura è raggiungibile.
- (b) Se `opt=1` la funzione fornisce in uscita anche tutti i posti del dispiegamento corrispondenti alla marcatura `M`.
- (c) Se `opt=2` la funzione fornisce anche una terza uscita che indica qual è la sequenza di scatto più breve che conduce alla marcatura `M`.
- (d) Se `opt=3` la terza uscita della funzione fornisce tutte le sequenze di scatto che generano `M`.

Le uscite sono invece:

- ² `ret` : vale 1 se la marcatura `M` è raggiungibile, 0 altrimenti.
- ² `p_corr` : è una matrice nelle cui righe ci sono i posti del dispiegamento che corrispondono alla marcatura `M` (`p_corr` contiene i posti corrispondenti a `M` nella numerazione progressiva. Per facilitare l'interpretazione dei risultati, il programma stampa su schermo anche i rispettivi posti con la numerazione della rete di partenza).
- ² `X`: è la terza uscita che cambia signi cato a seconda del valore di `opt`:
 - (a) Se `opt=2` `X` contiene la più piccola sequenza di scatto che conduce alla marcatura `M`.
 - (b) Se `opt=3` `X` fornisce tutte le sequenze di scatto che generano `M`.

6.2 Esempio di funzionamento

Vengono proposti alcuni demo (le reti a cui si riferiscono sono riportate in Appendice A) eseguibili direttamente sul prompt dei comandi di Matlab digitando il seguente comando:

```
>>demo1_reach_u (per la rete in Figura A.5)
```

```
>>demo2_reach_u (per la rete in Figura A.1)
```


Figura A.4: Rete P/T

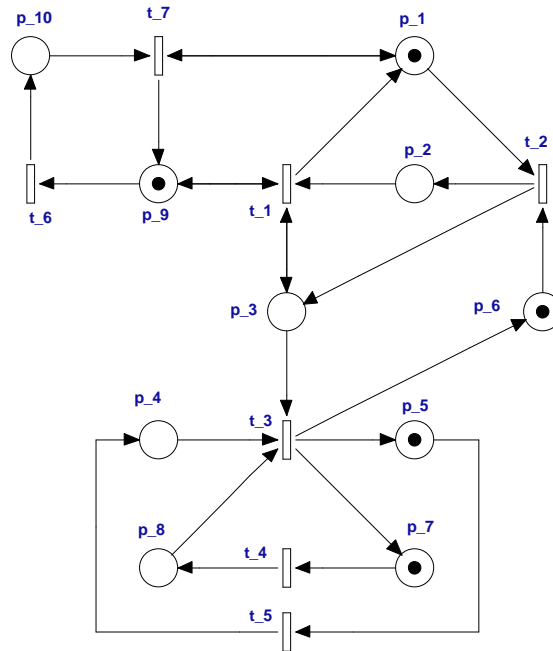


Figura A.5: Rete P/T

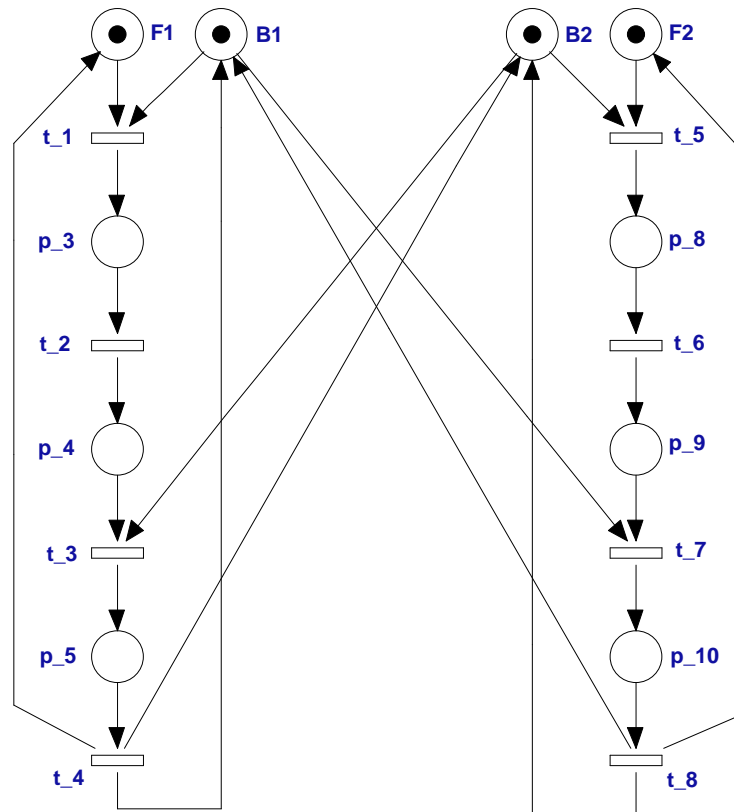


Figura A.6: Rete P/T rappresentante due filosofi con 4 stati ciascuno

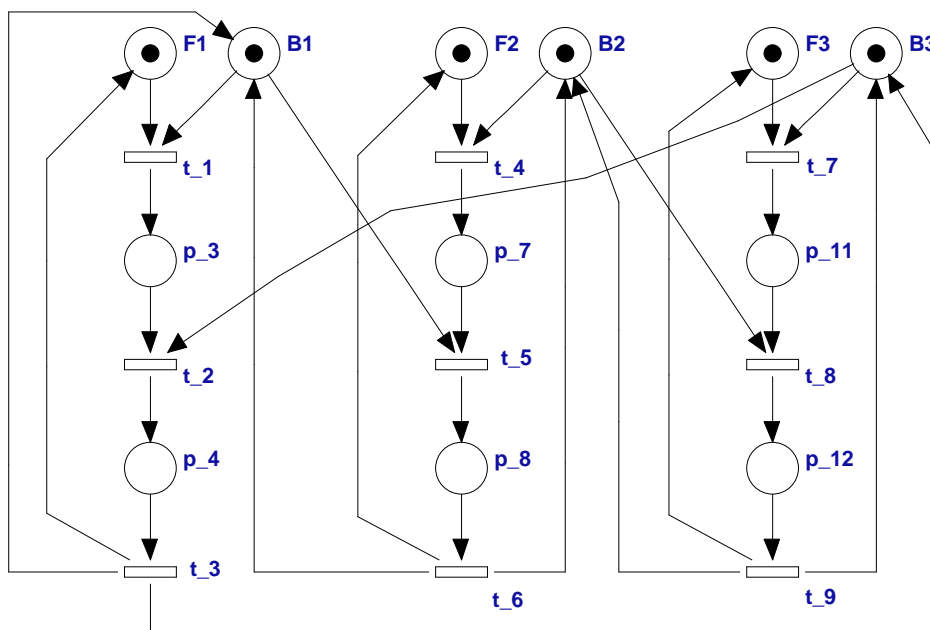


Figura A.7: Rete P/T rappresentante tre filosofi con 3 stati ciascuno