



Sintesi Sequenziale Sincrona

Sintesi Comportamentale di Reti Sequenziali Sincrone

Riduzione del numero degli stati per Macchine Non Completamente Specificate

Compatibilità

Versione del 15/01/05



Macchine non completamente specificate

- Sono macchine in cui per alcune configurazioni degli ingressi e dello stato presente non sono specificati gli stati prossimi e/o le configurazioni d'uscita
- La **riduzione del numero degli stati** in macchine non completamente specificate è ricondotta alla individuazione di una **macchina minima che copre** (compatibile con) quella data
- Il metodo di riduzione è simile a quello per macchine completamente specificate ma si basa sulla proprietà di **compatibilità** tra stati, invece che su quella di indistinguibilità.

- 2 -



Macchine non completamente specificate: sequenza di ingresso applicabile e stati compatibili

Data una macchina non completamente specificata:

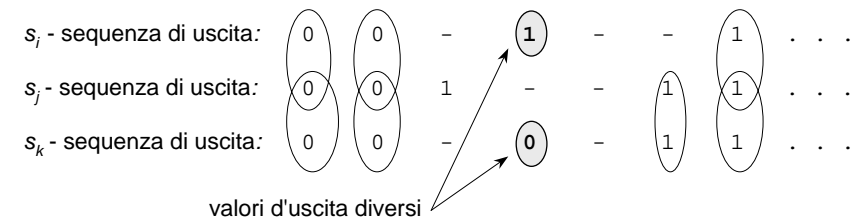
- una **sequenza di ingresso** si dice **applicabile** a partire da uno stato s_i se:
 - la funzione stato prossimo δ è specificata per ogni simbolo d'ingresso della sequenza, tranne al più l'ultimo
- Due stati s_i e s_j di una macchina M si dicono **compatibili** se
 - partendo da s_i e da s_j
 - usando ogni possibile sequenza di ingresso **applicabile** I_a
 - si ottengono le stesse sequenze d'uscita *ovunque queste siano specificate*
- La compatibilità tra s_i e s_j si indica con: $s_i \vee s_j$

- 3 -



Macchine non completamente specificate: compatibilità

- La **compatibilità** è una relazione meno forte di quella di **indistinguibilità**
- **Non vale la proprietà transitiva** cioè se $s_i \vee s_j$ e $s_j \vee s_k$ può non essere $s_i \vee s_k$. Quindi la compatibilità **non è** una relazione di **equivalenza**
- Ad esempio, $s_i \vee s_j$ e $s_j \vee s_k$ ma $s_i \not\vee s_k$:



- 4 -



Riduzione del numero degli stati: stati compatibili

- La **regola di Paull - Unger** è stata estesa per trattare il caso delle macchine non completamente specificate
- Due stati sono **compatibili** se e solo se, per ogni simbolo di ingresso i_α valgono le seguenti relazioni:
 - $\lambda(s_i, i_\alpha) = \lambda(s_j, i_\alpha)$ e
 - I valori di uscita sono identici se ambedue specificati
 - se uno o entrambi non sono specificati l'uguaglianza si ritiene soddisfatta
 - $\delta(s_i, i_\alpha) \vee \delta(s_j, i_\alpha)$
 - gli stati prossimi sono compatibili se ambedue specificati
 - se uno o entrambi non sono specificati la compatibilità si ritiene soddisfatta

- 5 -



Riduzione del numero degli stati: compatibilità e regola di Paull-Unger

- Poiché gli insiemi S e I hanno cardinalità finita, l'analisi di tutte le coppie di stati può portare ad una delle tre condizioni
 - $s_i \nvee s_j$
 - Se i simboli d'uscita sono diversi e/o
 - Se gli stati prossimi sono già stati verificati come non compatibili
 - $s_i \vee s_j$
 - Se i simboli d'uscita sono uguali e
 - Se gli stati prossimi sono già stati verificati come compatibili
 - Insieme di coppie di stati che devono essere compatibili affinché la coppia in oggetto sia compatibile (**compatibilità condizionate**)

- 6 -



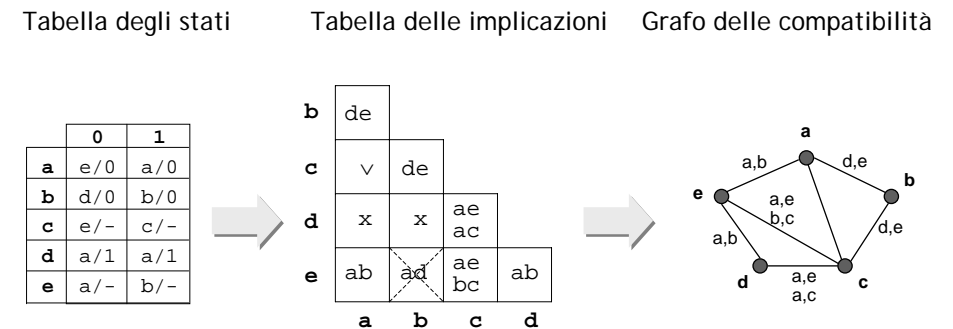
Riduzione del numero degli stati: tabella delle implicazioni

- Le relazioni di compatibilità si identificano con la **Tabella delle Implicazioni** che viene costruita come nel caso della indistinguibilità
- L'analisi della tabella consente di propagare le incompatibilità, ma non di risolvere i vincoli di compatibilità condizionata. Quindi al termine dell'analisi, ogni elemento contiene:
 - Il simbolo di non compatibilità, se gli stati corrispondenti non sono compatibili
 - Il simbolo di compatibilità, se gli stati corrispondenti sono compatibili
 - Le coppie di stati che devono essere compatibili affinché la coppia in oggetto sia compatibile (**vincoli**)
- Poiché la relazione di compatibilità **non è transitiva**, non si può concludere che tutte le **compatibilità** sono soddisfatte. I **vincoli** vanno mantenuti per la costruzione delle **classi di compatibilità**
- Le classi di compatibilità si costruiscono esaminando il **grafo delle compatibilità**, che riporta le compatibilità **condizionate** e quelle incondizionate

- 7 -



Riduzione del numero degli stati: **Esempio**



- 8 -



Riduzione del numero degli stati: classi di compatibilità

Classe di compatibilità:

- Insieme di stati compatibili fra di loro a coppie
- Sul grafo di compatibilità una classe di compatibilità è rappresentata da un **sottografo completo**
- le classi di compatibilità non generano una partizione tra gli stati (**non sono disgiunte**): uno stato può appartenere a più di una classe

Classe di compatibilità prima:

- Classe di compatibilità per la quale non esiste alcuna altra classe di compatibilità che la ricopre e che abbia un insieme di vincoli in essa incluso, o al limite coincidente

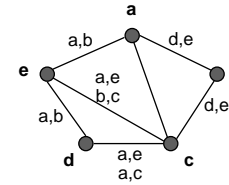
Classe di massima compatibilità:

- Classe di compatibilità non contenuta in alcuna altra classe
- Una classe di massima compatibilità è individuata sul grafo da un **sottografo completo non contenuto in nessun altro sottografo**

- 9 -



Riduzione del numero degli stati: classi di compatibilità - esempio



Classi di compatibilità:

- a, b, c, d, e, ab, ac, ae, bc, ce, cd, de, abc, aec, dec

Classi di massima compatibilità:

- abc, aec, dec

- 10 -



Riduzione del numero degli stati: classi di compatibilità prime - esempio

$\{a, b, c\}$	$\{(d, e)\}$	} classi di massima compatibilità
$\{a, c, e\}$	$\{(a, b); (b, c)\}$	
$\{c, d, e\}$	$\{(a, b); (a, e); (a, c); (b, c)\}$	
$\{a, c\}$	\emptyset	
$\{a, e\}$	$\{(a, b)\}$	ad esempio copre un numero di stati inferiore (è contenuta) ma ha anche meno vincoli
$\{c, d\}$	$\{(a, c); (a, e)\}$	
$\{c, e\}$	$\{(a, e); (b, c)\}$	
$\{d, e\}$	$\{(a, b)\}$	
$\{b\}$	\emptyset	
$\{d\}$	\emptyset	
$\{e\}$	\emptyset	

- 11 -



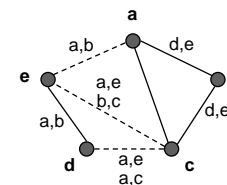
Riduzione del numero degli stati: Insieme chiuso di classi di compatibilità

Insieme chiuso di classi di compatibilità:

- Per ogni classe dell'insieme deve valere la seguente relazione:
 - per ogni simbolo di ingresso, data una classe dell'insieme, e un simbolo di ingresso, l'insieme degli stati futuri relativi è contenuto in una stessa classe (almeno) dell'insieme (cioè tutti i vincoli sono rispettati)

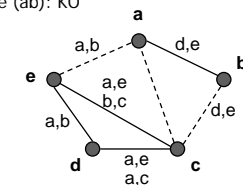
Insieme (abc), (ed): chiuso

- da (abc) con 0 vado in (ed), con 1 in (abc): OK
- da (ed) con 0 vado in (a), con 1 in (ab): OK



Insieme (ab), (ced): NON chiuso

- da (ab) con 0 vado in (ed), con 1 in (ab): OK
- da (ced) con 0 vado in (e) e (a): KO, con 1 in (c) e (ab): KO



- 12 -



Riduzione del numero degli stati: copertura della macchina

- Data una macchina M e il suo insieme di classi di compatibilità, la macchina M' il cui **insieme degli stati** è costituito da un **insieme chiuso delle classi di compatibilità di M** (che include tutti gli stati di M) **copre M**
- Per costruzione, il comportamento di M' è compatibile con quello di M e cioè,
 - Partendo da un qualsiasi stato di M , ne esiste uno in M' tale che
 - Per ogni sequenza di ingresso applicabile a entrambi, le sequenze di uscita sono identiche ogni volta che l'uscita di M è specificata
- Il problema della **minimizzazione del numero di stati** di una macchina non completamente specificata equivale quindi a:
 - Trovare il **più piccolo insieme chiuso di classi di compatibilità** che **coprono tutti gli stati della macchina**

- 13 -



Riduzione del numero degli stati: copertura e minimizzazione

- L'insieme di **tutte le classi di massima compatibilità** è **chiuso** e **copre l'insieme S degli stati**
- Associando un nuovo stato ad una classe di massima compatibilità si ottiene una nuova macchina con un numero di stati:
 - Possibilmente minore di quello della macchina di partenza
 - **Non necessariamente minimo**
- Il numero di classi di massima compatibilità è il **limite superiore al numero degli stati ridotto**
- In genere, **la macchina minima non è unica**. Gli algoritmi esaustivi per identificare la macchina minima partono tutti dall'insieme delle classi di compatibilità massime

- 14 -



Riduzione del numero degli stati: ricerca delle classi di massima compatibilità

- La definizione delle classi di massima compatibilità può avvenire individuando direttamente sul grafo tutti i più grandi sottografi completi
- Esistono diversi algoritmi specifici per l'individuazione di tutte le classi di massima compatibilità che utilizzano la **tabella delle implicazioni** considerando tutte e sole le **incompatibilità**.
- Costruzione della funzione per il test di compatibilità
- **Costruzione, per colonne o per righe, dell'albero dei compatibili massimi**

- 15 -



Ricerca delle classi di massima compatibilità Albero dei compatibili massimi per colonne

Premesse:

- La radice dell'albero è costituita da tutti gli stati della macchina (elencati secondo l'ordine presente nella tabella delle implicazioni)
- Ogni **nodo** è **costituito** da un **elenco di stati possibilmente compatibili**
- Ogni stato della macchina genera un livello nell'albero
- I nodi di un certo livello sono costituiti da un elenco di stati per i quali la compatibilità è già stata verificata per tutti gli stati in elenco corrispondenti ai livelli dell'albero al momento costruito
- Se un nodo è costituito da stati tutti già analizzati, tranne al più l'ultimo, allora l'analisi relativa a quel nodo è terminata e **il nodo è una foglia dell'albero**
- Se un nodo è costituito da un insieme di stati già compresi in un altro nodo dello stesso livello o di un nodo foglia, il nodo può essere eliminato

- 16 -



Ricerca delle classi di massima compatibilità Albero dei compatibili massimi per colonne

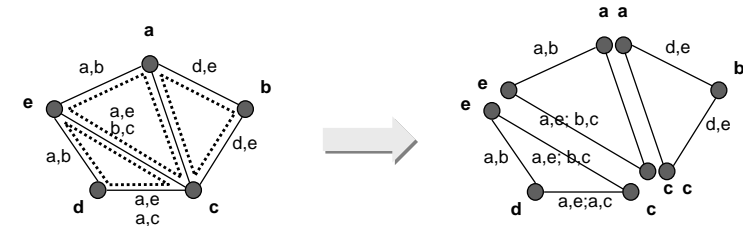
La costruzione dell'albero avviene secondo queste linee guida

- Dalla radice vengono costruiti 2 nuovi nodi, derivanti **dall'esame del primo stato a sinistra** dell'elenco che costituisce la radice stessa
 - Il **nodo a sinistra** è costituito **da tutti gli stati della radice tranne lo stato corrente** (all'inizio il primo stato dell'elenco)
 - Il **nodo a destra** contiene **lo stato in esame, cioè il primo** (quelli precedenti, se esistono) **e tutti i successivi ad esso compatibili** (derivati dalla colonna corrispondente allo stato in esame, nella tabella delle implicazioni che riporta le sole incompatibilità)
- Terminata la generazione dei nodi di un livello, si passa ad esaminare lo **stato successivo** dell'elenco costruendo quindi un **nuovo livello** dell'albero
- Ad ogni livello aggiunto nell'albero si esamina uno stato e si costruiscono due sotto-alberi per ogni nodo già presente, sempre secondo le modalità **sinistra-destra**
- Il procedimento **termina**, quando si sono esaminati **tutti gli stati**, tranne l'ultimo dell'elenco di partenza
- Le **foglie dell'albero** rappresentano i **compatibili massimi**

- 17 -



Classi di compatibilità massima - Esempio di derivazione dal grafo

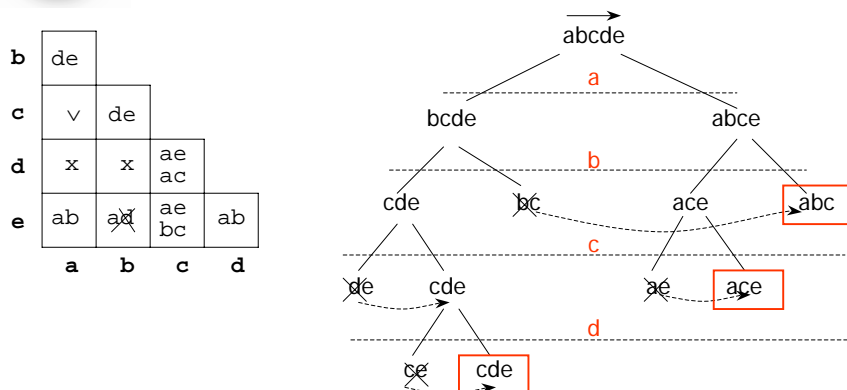


- Classi di massima compatibilità:
 - $\{a, b, c\} : \{(d, e)\}$
 - $\{a, c, e\} : \{(a, b); (b, c)\}$
 - $\{c, d, e\} : \{(a, b); (a, e); (a, c); (b, c)\}$
- Una copertura ammissibile è data dall'insieme delle classi di massima compatibilità: tale copertura non è necessariamente minima

- 18 -



Classi di compatibilità massima - Esempio di derivazione dall'albero



- Classi di massima compatibilità: $\{a, b, c\}$, $\{a, c, e\}$, $\{c, d, e\}$

- 19 -



Riduzione del numero degli stati: ricerca di una copertura minimale

- La mancanza di disgiunzione tra le classi di massima compatibilità non consente di definire metodi esatti per la minimizzazione. Si utilizza un'euristica.
- Ricerca di un insieme chiuso di classi di compatibilità che coprono la macchina a stati non completamente specificata
 - L'algoritmo *greedy* proposto consente di trovare una **copertura della macchina** a stati **tramite un insieme chiuso di classi di compatibilità di cardinalità non superiore al numero di classi di massima compatibilità**

- 20 -



Ricerca copertura chiusa

Funzione di costo:

- Benefici:
 - Numero di stati coperti dalla classe di compatibilità (+)
 - Numero di vincoli risolti dalla scelta della classe di compatibilità (+)
- Costi:
 - Numero di nuovi vincoli introdotti dalla scelta della classe di compatibilità (-)

⇒ **Vincoli:** Le coppie di vincoli vengono "trasformate" in raggruppamenti di stati compatibili, per garantire la chiusura della copertura

Algoritmo:

Partendo dalla lista delle classi di compatibilità prime, si itera il seguente processo:

- Si calcola il valore della funzione di costo per ogni classe di copertura
- Si sceglie una tra le classi a valore maggiore (non con tutti i contributi nulli)
- Si "eliminano" i *vincoli* risolti dipendenti dalla scelta fatta, eliminando sia quelli che non sono più tali perché "coperti" dalla classe scelta, sia quelli coperti dai *vincoli* della classe scelta
- Si "eliminano" le classi completamente coperte dai vincoli della classe scelta perché non interessanti per successive scelte

Il processo termina quando tutti gli stati sono stati coperti e tutti i vincoli sono stati rispettati

- 21 -



Ricerca copertura minima

passo preliminare: trasformazione vincoli

$\{a,b,c\}$: $\{(d,e)\}$	$+3+0-1 = +2$
$\{a,c,e\}$: $\{(a,b);(b,c)\} \gg \{(a,b,c)\}$	$+3+0-1 = +2$
$\{c,d,e\}$: $\{(a,b);(a,e);(a,c);(b,c)\} \gg \{(a,e);(a,b,c)\}$	$+3+0-2 = +1$
$\{a,c\}$: \emptyset	$+2+0-0 = +2$
$\{a,e\}$: $\{(a,b)\}$	$+2+0-1 = +1$
$\{c,d\}$: $\{(a,c);(a,e)\}$	$+2+0-2 = 0$
$\{c,e\}$: $\{(a,e);(b,c)\}$	$+2+0-2 = 0$
$\{d,e\}$: $\{(a,b)\}$	$+2+0-1 = +1$
$\{b\}$: \emptyset	$+1+0-0 = +1$
$\{d\}$: \emptyset	$+1+0-0 = +1$
$\{e\}$: \emptyset	$+1+0-0 = +1$

	0	1
a	e/0	a/0
b	d/0	b/0
c	e/-	c/-
d	a/1	a/1
e	a/-	b/-

- 22 -



Ricerca copertura minima

passo 1

$\{a,b,c\}$: $\{(d,e)\}$	$+3+0-1 = +2$
$\{a,c,e\}$: $\{(a,b,c)\}$	$+3+0-1 = +2$
$\{c,d,e\}$: $\{(a,e);(a,b,c)\}$	$+3+0-2 = +1$
$\{a,c\}$: \emptyset	$+2+0-0 = +2$
$\{a,e\}$: $\{(a,b)\}$	$+2+0-1 = +1$
$\{c,d\}$: $\{(a,c);(a,e)\}$	$+2+0-2 = 0$
$\{c,e\}$: $\{(a,e);(b,c)\}$	$+2+0-2 = 0$
$\{d,e\}$: $\{(a,b)\}$	$+2+0-1 = +1$
$\{b\}$: \emptyset	$+1+0-0 = +1$
$\{d\}$: \emptyset	$+1+0-0 = +1$
$\{e\}$: \emptyset	$+1+0-0 = +1$

- 23 -



Ricerca copertura minima

effetti del passo 1 e passo 2

$\{a,b,c\} : \{(d,e)\}$	scelta al passo 1	$C = \{\{a,b,c\}\}$
$\{a,c,e\} : \{\overline{\{a,b,c\}}\}$	$+1+0-0 = +1$	
$\{c,d,e\} : \{(a,e); \overline{\{a,b,c\}}\}$	$+2+0-1 = +1$	
$\{a,c\} : \emptyset$...	
$\{a,e\} : \{\overline{\{a,b\}}\}$	$+1+0-0 = +1$	
$\{c,d\} : \{\overline{\{a,c\}}; (a,e)\}$	$+1+0-1 = 0$	
$\{c,e\} : \{(a,e); \overline{\{b,c\}}\}$	$+1+0-1 = 0$	
$\{d,e\} : \{\overline{\{a,b\}}\}$	$+2+1-0 = +3$	
$\{b\} : \emptyset$...	
$\{d\} : \emptyset$	$+1+0-0 = +1$	
$\{e\} : \emptyset$	$+1+0-0 = +1$	

- 24 -



Ricerca copertura minima effetti del passo 2 -- fine

$\{a, b, c\} : \{\overline{d, e}\}$ scelta al passo 1
 $\{a, c, e\} : \{\overline{a, b, c}\}$
 $\{c, d, e\} : \{(a, e); \overline{a, b, c}\}$
 $\{a, e\} : \{\overline{a, b}\}$
 $\{c, d\} : \{\overline{a, c}\}; (a, e)\}$
 $\{c, e\} : \{(a, e); \overline{b, c}\}$
 $\{d, e\} : \{\overline{a, b}\}$ scelta al passo 2
 $\overline{d} : \emptyset$
 $\overline{e} : \emptyset$

termina: tutti gli stati sono stati coperti e tutti i vincoli sono soddisfatti

$C = \{\{a, b, c\}; \{d, e\}\}$



Tabella degli stati della macchina ridotta Esempio

- Sulla base di:
 - Tabella degli stati della macchina iniziale
 - Insieme chiuso delle classi di compatibilità
- Si determina la nuova tabella degli stati corrispondente alla macchina ridotta

Tabella degli stati

	0	1
a	e/0	a/0
b	d/0	b/0
c	e/-	c/-
d	a/1	a/1
e	a/-	b/-



$s0 = \{a, b, c\}$
 $s1 = \{d, e\}$



Tabella degli stati ridotta

	0	1
s0	s1/0	s0/0
s1	s0/1	s0/1