Sintesi Sequenziale Sincrona

Sintesi Comportamentale di Reti Sequenziali Sincrone

Riduzione del numero degli stati per Macchine Non Completamente Specificate

Compatibilità

Versione del 13/01/05 (Ferrandi – Antola)

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. Ad esempio

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

- La riduzione del numero degli stati in macchine non completamente specificata 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.

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_α
  - si ottengono le stesse sequenze d'uscita *ovunque queste siano specificate*
- La compatibilità tra s_i e s_j si indica con: $s_i \vee s_j$

Macchine non completamente specificate: compatibilità

- La compatibilità è una relazione meno forte di quella di *indistinguibilità.* Valgono le proprietà riflessiva e simmetrica ma…
- 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 $s_i \times s_k$. :

s_i - sequenza di uscita:	0	0	-	**1**	-	-	1	. . .
s_j - sequenza di uscita:	0	0	1	-	-	1	1	. . .
s_k - sequenza di uscita:	0	0	-	**0**	-	1	1	. . .

valori d'uscita diversi

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 entrambe le seguenti relazioni:
  1. $\lambda(s_i, i_\alpha) = \lambda(s_j, i_\alpha)$
     - I valori di uscita sono identici se ambedue specificati
     - se uno o entrambi non sono specificati l'uguaglianza si ritiene soddisfatta
  2. $\delta(s_i, i_\alpha) \vee \delta(s_j, i_\alpha)$
     - gli stati prossimi sono compatibili se ambedue specificati
     - se uno o entrambi non sono specificati la compatibilità si ritiene soddisfatta

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

1. $s_i \not\vee s_j$: stati non compatibili
   - Se i simboli d'uscita sono diversi e/o
   - Se gli stati prossimi sono già stati verificati come non compatibili
2. $s_i \vee s_j$: stati compatibili
   - Se i simboli d'uscita sono uguali e
   - Se gli stati prossimi sono già stati verificati come compatibili
3. compatibilità condizionata: insieme di coppie di stati che devono essere compatibili affinchè la coppia in oggetto sia compatibile

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 affinchè 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

Riduzione del numero degli stati: *Esempio*

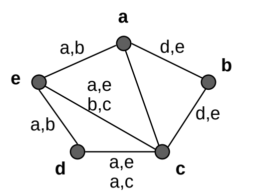
Tabella degli stati

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

Tabella delle implicazioni

b	de			
c	$\vee$	de		
d	x	x	ae ac	
e	ab	~~ad~~	ae bc	ab
	a	**b**	**c**	**d**

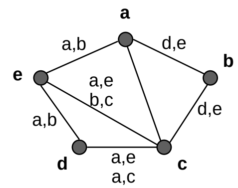
Grafo delle compatibilità



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*
- *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
  - Le classi di massima compatibilità non generano una partizione tra gli stati (non sono disgiunte): uno stato può appartenere a più di una classe
  - Le classi di massima compatibilità sono ovviamente classi di compatibilità prime

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

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

```
{a,b,c} : {(d,e)}
{a,c,e} : {(a,b);(b,c)}
{c,d,e} : {(a,b);(a,e);(a,c);(b,c)}
{a,c}   : ø
{a,e}   : {(a,b)}
{c,d}   : {(a,c);(a,e)}
{c,e}   : {(a,e);(b,c)}
{d,e}   : {(a,b)}
{b}     : ø
{d}     : ø
{e}     : ø
```

classi di massima compatibilità

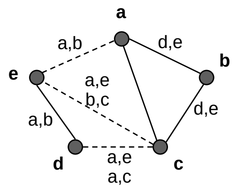
ad esempio: copre un numero di stati inferiore (è contenuta) ma ha anche meno vincoli

ab, **bc**, **a** e **c** non sono classi di compatibilità prime

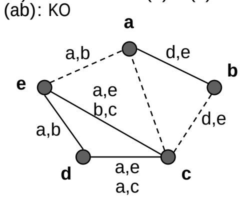
Riduzione del numero degli stati: Insieme chiuso di classi di compatibilità

- *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



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

Riduzione del numero degli stati: *costruzione della tabella degli stati della macchina ridotta*

- Una volta identificata la copertura tramite le classi di compatibilità, la costruzione della tabella degli stati della macchina ridotta avviene nel modo seguente
  - Gli stati della macchina ridotta sono le classi di compatibilità individuate
  - Per ogni classe di compatibilità:
    - se, per almeno uno degli stati della classe, lo stato prossimo è specificato, allora la classe di compatibilità che lo contiene sarà lo stato prossimo della macchina ridotta
      - Poichè l'insieme delle classi che costituiscono la copertura può essere non disgiunto, uno stato della macchina originaria può essere presente in più classi di copertura. Nella costruzione della tabella degli stati della macchina ridotta è arbitrario scegliere la classe cui appartiene
    - se, per almeno uno degli stati originari che costituiscono lo stato prossimo della macchina ridotta, l'uscita è specificata, allora questa uscita sarà l'uscita associata allo stato prossimo nella macchina ridotta
    - in ogni altro caso si mantengono le condizioni non specificate

Tabella degli stati della macchina ridotta

- 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

Riduzione del numero degli stati: *individuazione di un insieme chiuso di classi di compatibilità*

- A causa della mancanza di disgiunzione tra le classi di massima compatibilità, per trovare la macchina compatibile minima (che può anche essere non unica) è necessario ricorrere ad algoritmi di copertura esaustivi
- Si considerano nel seguito tre tecniche, non esaustive, che consentono di identificare un insieme, possibilmente ridotto, e chiuso di classi che copre la macchina data

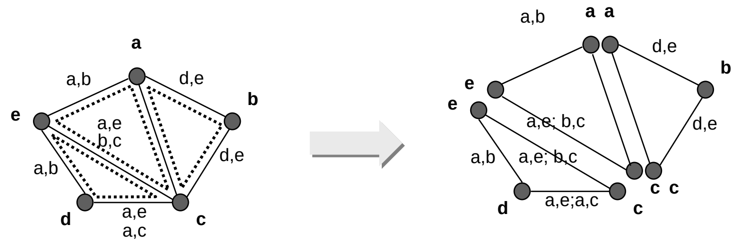
1. Uso diretto delle classi di massima compatibilità
2. Euristica con classi di massima compatibilità (che può generare soluzioni non ammissibili)
3. Euristica con classi di compatibilità prime

Riduzione del numero degli stati:
1) uso diretto delle classi di massima compatibilità

- 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

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



- 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

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

- 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

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

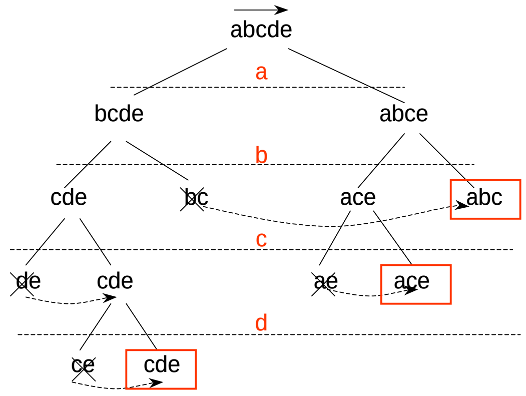
Ricerca delle classi di massima compatibilità
Albero dei compatibili massimi per colonne

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

Classi di compatibilità massima – *Esempio di derivazione dall'albero*

b	de			
c	∨	de		
d	x	x	ae ac	
e	ab	~~ad~~	ae bc	ab
	a	**b**	**c**	**d**



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

Riduzione del numero degli stati:
2) euristica con classi di massima compatibilità

Euristica con classi di massima compatibilità

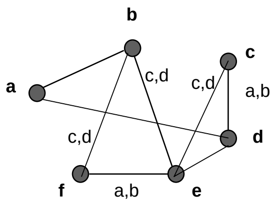
- Ricerca di un insieme chiuso di classi di compatibilità che coprono la macchina a stati non completamente specificata
  - L'algoritmo *greedy* proposto è semplice e lavora sul grafo di compatibilità
  - Parte considerando tutte le classi di massima compatibilità e consente di trovare una copertura della macchina a stati tramite un insieme di classi di compatibilità (non necessariamente tutte massime) di cardinalità non superiore al numero di classi di massima compatibilità
  - La chiusura degli insiemi individuati per la copertura è garantita solo se i vincoli di compatibilità iniziali soddisfano una opportuna espressione logica
  - Lavorando senza la verifica dell'espressione iniziale (versione mostrata), l'algoritmo può individuare soluzioni non ammissibili perché non chiuse

Euristica con classi di massima compatibilità

1. Inizializzare una lista *L1* vuota
2. Finchè il grafo non è vuoto:
   a. Individuare e ordinare le classi di massima compatibilità presenti sul grafo per dimensione
   b. Individuare la classe di compatibilità massima di dimensione massima presente sul grafo
   c. Inserire nella lista *L1* tutti i vincoli presenti nella classe di compatibilità considerata
   d. Eliminare dalla lista *L1* e dal grafo i vincoli soddisfatti dalla classe considerata
   e. Eliminare dal grafo tutti i nodi (ed i relativi archi) appartenenti alla classe di compatibilità considerata che non appartengono a nessun vincolo presente nella lista *L1* e/o nel grafo
3. Le classi così individuate formano un insieme di classi di compatibilità chiuso?
4. Se sì, è stata individuata una soluzione ammissibile. Se no, il procedimento viene ripetuto operando una diversa scelta (iniziale)

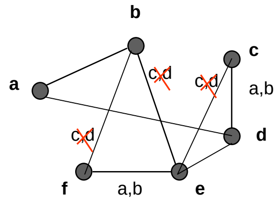
Algoritmo di ricerca - *Esempio*

Grafo di partenza Passo 1

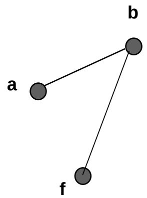


Passo 1

a’)	cde, bfe, ab, ad
b’)	cde
c’)	L1= ab, cd
d’)	L1= ab

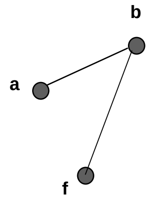


e’)



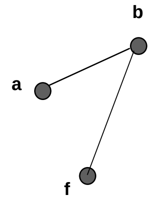
Algoritmo di ricerca – *Esempio (cont.)*

Grafo di partenza Passo 2

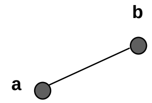


Passo 2

a’)	bf, ab
b’)	bf
c’)	L1= ab
d’)	L1= ab

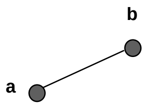


e’)



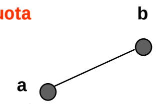
Algoritmo di ricerca – *Esempio (cont.)*

Grafo di partenza Passo 3



Passo 3

a’)	ab
b’)	ab
c’)	L1= ab
d’)	L1= vuota
e’)	grafo vuoto



Copertura individuata

cde, bf, ab

E’ chiusa? Si! Ed è costituita da un insieme di cardinalità inferiore rispetto a quello costituito da tutte le classi di massima compatibilità

Riduzione del numero degli stati: 3) euristica con classi di compatibilità prime

Euristica con classi di compatibilità prime

- Ricerca di un insieme chiuso di classi di compatibilità che coprono la macchina a stati non completamente specificata
  - L’algoritmo *greedy* proposto usa una funzione costo per guidare nella scelta e lavora considerando tutte le classi di compatibilità prime
  - Consente di trovare una copertura della macchina a stati tramite un insieme chiuso di classi di compatibilità prime di cardinalità non superiore al numero di classi di massima compatibilità
  - Per garantire la chiusura degli insiemi individuati per la copertura, l’algoritmo prevede un passo preliminare per la trasformazione dei vincoli di compatibilità definiti dalla tabella della implicazioni

Euristica con classi di compatibilità prime

Funzione di costo (= beneficio nella scelta di una classe):

- Benefici:
  - Numero di stati coperti dalla classe di compatibilità (+)
  - Numero di vincoli risolti dalla scelta della classe in altre classi già scelte(+)
- Costi:
  - Numero di vincoli introdotti dalla scelta della classe di compatibilità (-)

Vincoli: Tramite la tabella degli stati, 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
- 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

Il processo termina quando tutti gli stati sono stati coperti e tutti i vincoli delle classi scelte sono soddisfatti

Euristica con classi di compatibilità prime
passo preliminare: trasformazione vincoli

```
Classi : Vincoli
{a,b,c} : {(d,e)}
{a,c,e} : {(a,b);(b,c)} >> {(a,b,c)}
{c,d,e} : {(a,b);(a,e);(a,c);(b,c)}>>{(a,e);(a,b,c)}
{a,c}   : ø
{a,e}   : {(a,b)}
{c,d}   : {(a,c);(a,e)}
{c,e}   : {(a,e);(b,c)}
{d,e}   : {(a,b)}
{b}     : ø
{d}     : ø
{e}     : ø
```

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

Euristica con classi di compatibilità prime
passo 1: calcolo costi e scelta classe

```
{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}   : ø                     +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}     : ø                     +1+0-0 = +1
{d}     : ø                     +1+0-0 = +1
{e}     : ø                     +1+0-0 = +1
```

Copertura al passo 1:

C = {(a,b,c)}

Euristica con classi di compatibilità prime
passo 1: eliminazione vincoli e classi

- Si eliminano i vincoli risolti dalla classe scelta e quelli risolti dai vincoli della classe scelta
- Si eliminano le classi ricomprese nella classe scelta e nei suoi vincoli

{a,b,c} : {(d,e)}		**scelta al passo 1**
{a,c,e} : ~~{(a,b,c)}~~	+3+0-1 = +2	
{c,d,e} : {(a,e);~~{a,b,c}~~}	+3+0-2 = +1	
~~{a,c} : ø~~	~~+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} : ø~~	~~+1+0-0 = +1~~	
~~{d} : ø~~	~~+1+0-0 = +1~~	
~~{e} : ø~~	~~+1+0-0 = +1~~	

Euristica con classi di compatibilità prime

passo 2: calcolo costi e scelta classe

{a,b,c} : {(d,e)}		**scelta al passo 1**
{a,c,e} : ~~{(a,b,c)}~~		+1+0-0 = +1
{c,d,e} : {(a,e);~~(a,b,c)~~}		+2+0-1 = +1
~~{a,c} : ø~~		
{a,e} : {~~(a,b)~~}		+1+0-0 = +1
{c,d} : {~~(a,c)~~;(a,e)}		+1+0-1 = 0
{c,e} : {(a,e);~~(b,c)~~}		+1+0-1 = 0
{d,e} : {~~(a,b)~~}		+2+**1**-0 = +3
~~{b} : ø~~		
~~{d} : ø~~		
~~{e} : ø~~		

Copertura al passo 2:
C = {(a,b,c); (d,e)}

Euristica con classi di compatibilità prime

passo 2: eliminazione vincoli e classi

{a,b,c} : {~~(d,e)~~}		**scelta al passo 1**
{a,c,e} : ~~{(a,b,c)}~~		+1+0-0 = +1
{c,d,e} : {(a,e);~~(a,b,c)~~}		+2+0-1 = +1
~~{a,c} : ø~~		
{a,e} : {~~(a,b)~~}		+1+0-0 = +1
{c,d} : {~~(a,c)~~;(a,e)}		+1+0-1 = 0
{c,e} : {(a,e);~~(b,c)~~}		+1+0-1 = 0
{d,e} : {~~(a,b)~~}		**scelta al passo 2**
~~{b} : ø~~		
~~{d} : ø~~		
~~{e} : ø~~		

Sono stati coperti tutti gli stati e soddisfatti tutti i vincoli delle classi scelte. La copertura finale è: C = {(a,b,c); (d,e)}

Tabella degli stati della macchina ridotta

- 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