

Decomposizioni che hanno un join senza perdita

Index

- [Introduction](#)
 - [Definizione](#)
 - [Teorema](#)
 - [Digressione](#)
 - [Algoritmo di verifica](#)
 - [Esempi](#)
 - [Teorema](#)
-

Introduction

Quando si decompone uno schema occorre tenere presente (oltre al dover mantenere tutte le dipendenze originali), deve permettere di **ricostruire mediante il join naturale** ogni **istanza legale dello schema originario** (senza aggiunta di tuple estranee)
[Qui](#) qualche esempio in cui ciò non accade

Definizione

Se si decompone uno schema di relazione R si vuole che la decomposizione $\{R_1, R_2, \dots, R_k\}$ ottenuta sia tale che ogni istanza legale r di R sia ricostruibile mediante join naturale (\bowtie) da un'istanza legale $\{r_1, r_2, \dots, r_k\}$ dello schema decomposto $\{R_1, R_2, \dots, R_k\}$

Definizione

Sia R uno schema di relazione. Una decomposizione $\rho = \{R_1, R_2, \dots, R_k\}$ di R ha un **join senza perdita** se per ogni istanza legale r di R si ha

$$r = \pi_{R_1}(r) \bowtie \pi_{R_2}(r) \bowtie \dots \bowtie \pi_{R_k}(r)$$

Teorema

Sia R uno schema di relazione e $\rho = \{R_1, R_2, \dots, R_k\}$ una decomposizione di R . Per ogni istanza legale r di R , indicato con $m_\rho(r) = \pi_{R_1}(r) \bowtie \pi_{R_2}(r) \bowtie \dots \bowtie \pi_{R_k}(r)$ si ha:

- $r \subseteq m_\rho(r)$
- $\pi_{R_i}(m_\rho(r)) = \pi_{R_i}(r)$
- $m_\rho(m_\rho(r)) = m_\rho(r)$

Digressione

Consideriamo la solita istanza legale di $R = ABC$ con l'insieme di dipendenze funzionali $F = \{A \rightarrow B, C \rightarrow B\}$ (non è in 3NF - B non è contenuto in una chiave)

R	A	B	C
r	a1	b1	c1
	a2	b1	c1
	a3	b1	c2

In base alle possibili decomposizioni dello schema, questa istanza si decompone in

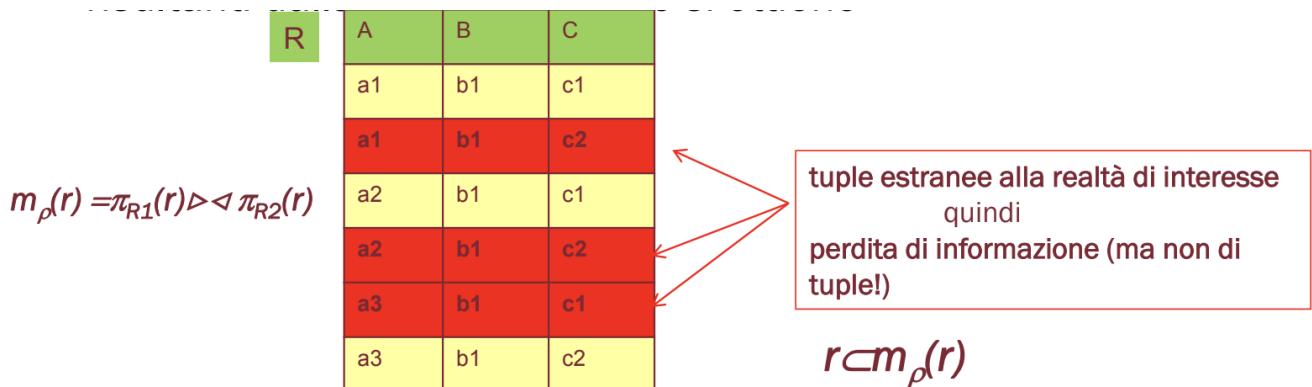
R1	A	B
$\pi_{R_1}(r)$	a1	b1
	a2	b1
	a3	b1

R2	B	C
$\pi_{R_2}(r)$	b1	c1
	b1	c2

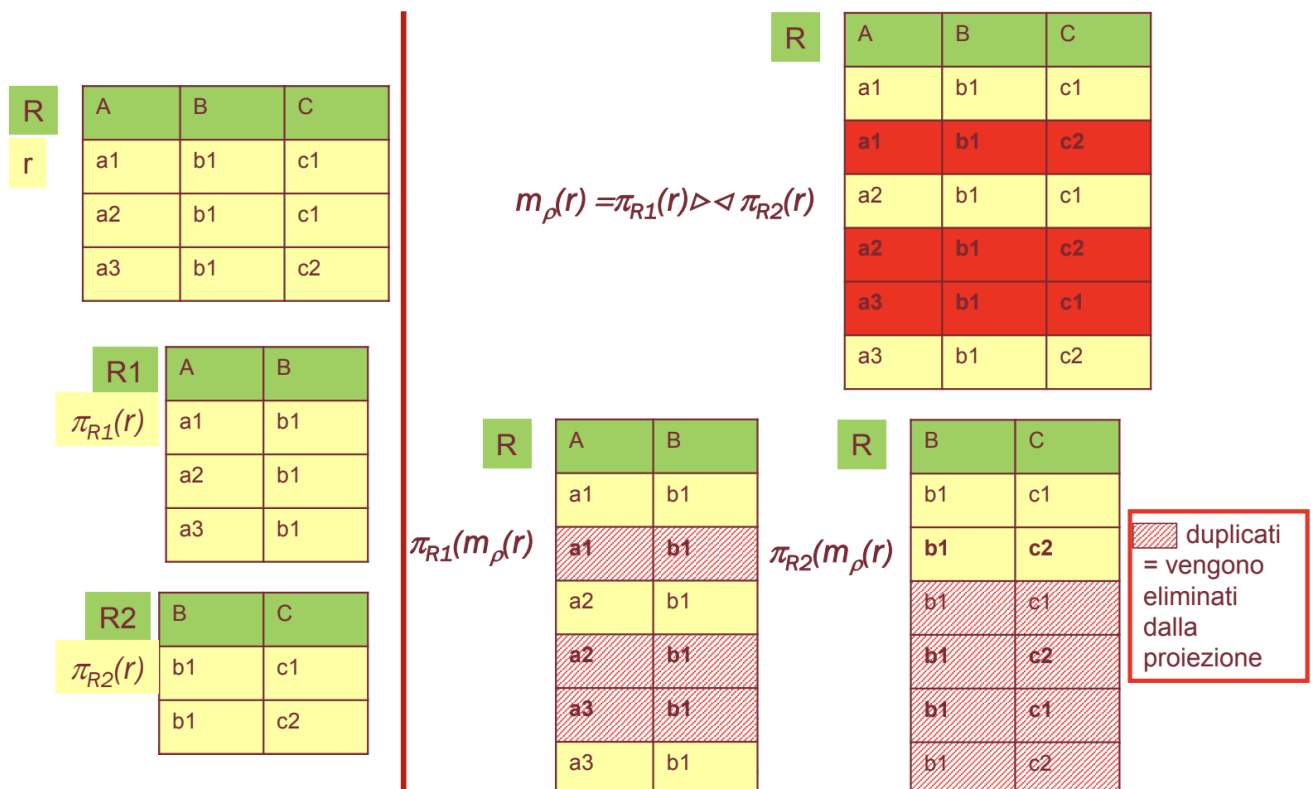
La prima è ottenuta proiettando l'istanza originale su AB . La seconda è ottenuta proiettando l'istanza originale su BC .

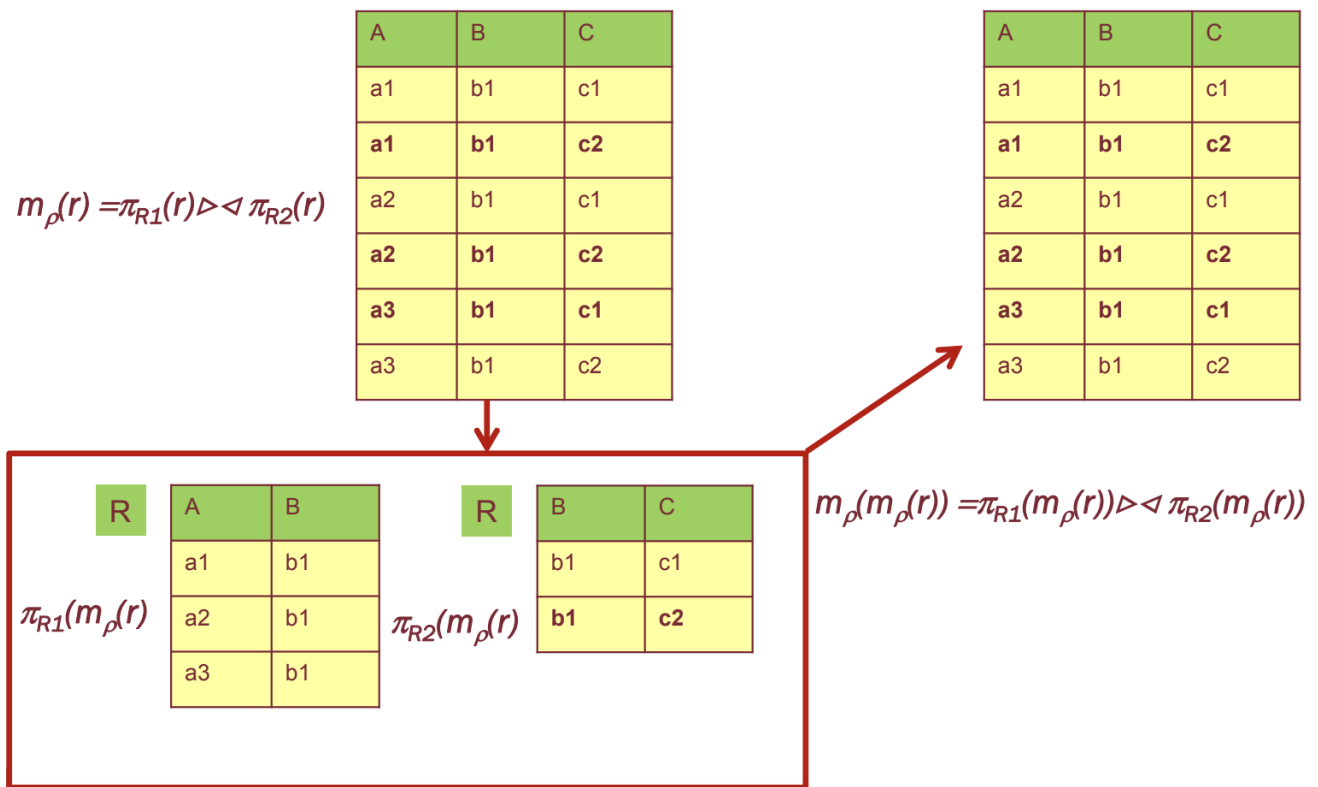
Notare l'eliminazione del duplicato. Forse questa eliminazione ci farà perdere tuple originali? No

Dovrebbe essere possibile ricostruire l'istanza di partenza esattamente tramite join invece se si effettua il join delle due istanze legali risultanti dalla decomposizione si ottiene



Occorre garantire che il join delle istanze risultanti dalla decomposizione non riveli *perdita di informazioni*





Algoritmo di verifica

Il seguente algoritmo permette di verificare se una decomposizione data ha un join senza perdita in tempo polinomiale

Input uno schema di relazione R , un insieme F di dipendenze funzionali su R , una decomposizione $\rho = \{R_1, R_2, \dots, R_k\}$ di R

Output decide se ρ ha un join senza perdita

begin

Costruisci una tabella r nel modo seguente:

r ha $|R|$ colonne e $|\rho|$ righe

all'incrocio dell' i -esima riga e della j -esima colonna metti

il simbolo a_j se l'attributo $A_j \in R_i$

il simbolo b_{ij} altrimenti

repeat

for every $X \rightarrow Y \in F$

do if ci sono due tuple t_1 e t_2 in r tali che $t_1[X] = t_2[X]$ e $t_1[Y] \neq t_2[Y]$

then for every attribute $A_j \in Y$

do if $t_1[A_j] = a_j$

then $t_2[A_j] := t_1[A_j]$

else $t_1[A_j] := t_2[A_j]$

until r ha una riga con tutte 'a' **or** r non è cambiato

if r ha una riga con tutte 'a'

then ρ ha un join senza perdita

else ρ non ha un join senza perdita

Esempi

≡ Esempio 1 >

$$R = (A, B, C, D, E)$$

$$F = \{C \rightarrow D, AB \rightarrow E, D \rightarrow B\}$$

$$\rho = \{AC, ADE, CDE, AD, B\}$$

Verificare se la decomposizione ρ ha un join senza perdita

Cominciamo a costruire la relativa tabella

	A	B	C	D	E
AC	a1	b12	a3	b14	b15
ADE	a1	b22	b23	a4	a5
CDE	b31	b32	a3	a4	a5
AD	a1	b42	b43	a4	b45
B	b51	a2	b53	b54	b55

Info

Per chiarezza applichiamo le dipendenze funzionali nell'ordine e vediamo i cambiamenti che vengono effettuati sulla tabella (ricordiamo che ogni cambiamento corrisponde a fare in modo che venga soddisfatta una dipendenza funzionale, per ottenere alla fine dell'algoritmo una tabella che

rappresenta un'istanza legale dello schema)

Indicheremo col simbolo \rightarrow le modifiche ai valori della tabella e con un apice l'ordine delle sostituzioni quando opportuno

	A	B	C	D	E	F = { C \rightarrow D, AB \rightarrow E, D \rightarrow B }
AC	a1	b12	a3	b14 \rightarrow a4 ⁽¹⁾	b15	
ADE	a1	b22 \rightarrow b12 ⁽²⁾	b23	a4	a5	
CDE	b31	b32 \rightarrow b12 ⁽²⁾	a3	a4	a5	
AD	a1	b42 \rightarrow b12 ⁽²⁾	b43	a4	b45	
B	b51	a2	b53	b54	b55	

- $C \rightarrow D$
 - la prima e la terza riga coincidono sull'attributo $C = a3$, quindi cambiamo $b14$ in $a4$ in modo che la dipendenza funzionale sia soddisfatta (solo le righe che hanno valori uguali in C devono avere valori uguali in D)
- $AB \rightarrow E$
 - non viene utilizzata in questo passo: la dipendenza funzionale è già soddisfatta, in quanto non ci sono (ancora) tuple uguali su AB e diverse su E , quindi non devono essere effettuati cambiamenti
- $D \rightarrow B$
 - nelle prime quattro righe $D = a4$, quindi cambiamo $b22$ in $b12$, $b32$ in $b12$, $b42$ in $b12$ (potevamo scegliere una diversa sostituzione delle b , purché le rendesse tutte uguali)

Abbiamo completato la prima iterazione del for e la tabella è stata modificata quindi continuiamo

	A	B	C	D	E	F = { C \rightarrow D, AB \rightarrow E, D \rightarrow B }
AC	a1	b12	a3	a4	b15 \rightarrow a5	
ADE	a1	b12	b23	a4	a5	
CDE	b31	b12	a3	a4	a5	
AD	a1	b12	b43	a4	b45 \rightarrow a5	
B	b51	a2	b53	b54	b55	

- $C \rightarrow D$
 - non viene utilizzata in questo passo: la dipendenza funzionale è già soddisfatta in quanto non ci sono tuple uguali su C e diverse su D
- $AB \rightarrow E$
 - la prima, la seconda e la quarta riga coincidono sugli attributi $AB = \langle a1, b12 \rangle$, quindi cambiamo $b15$ in $a5$ e $b45$ in $a5$ in modo che

la dipendenza funzionale sia soddisfatta (se le righe hanno valori uguali in AB , devono avere valori uguali in E)

- $D \rightarrow B$
 - non viene utilizzata in questo passo: la dipendenza funzionale è già soddisfatta in quanto non ci sono tuple uguali su A e diverse su B

Abbiamo completato la seconda iterazione del for e la tabella è stata modificata quindi continuiamo

	A	B	C	D	E	F = { C → D, AB → E, D → B }
AC	a1	b12	a3	a4	a5	
ADE	a1	b12	b23	a4	a5	
CDE	b31	b12	a3	a4	a5	
AD	a1	b12	b43	a4	a5	
B	b51	a2	b53	b54	b55	

- $C \rightarrow D$
 - non viene utilizzata in questo passo
- $AB \rightarrow E$
 - non viene utilizzata in questo passo
- $D \rightarrow B$
 - non viene utilizzata in questo passo

La tabella non cambia più e quindi l'algoritmo termina. Ora occorre verificare la presenza della tupla con tutte a

Poiché non c'è una riga con tutte a , il join non è senza perdita

≡ Esempio 2 >

$$R = (A, B, C, D, E, H, I)$$

$$F = \{A \rightarrow B, B \rightarrow AE, DI \rightarrow B, D \rightarrow HI, HI \rightarrow C, C \rightarrow A\}$$

$$\rho = \{ACD, BDEH, CHI\}$$

Verificare se la decomposizione ρ ha un join senza perdita

Cominciamo a costruire la tabella

F = { A → B, B → AE, DI → B, D → HI, HI → C, C → A }							
	A	B	C	D	E	H	I
ACD	a1	b12	a3	a4	b15	b16	b17
BDEH	b21	a2	b23	a4	a5	a6	b27
CHI	b31	b32	a3	b34	b35	a6	a7

$F = \{A \rightarrow B, B \rightarrow AE, DI \rightarrow B, D \rightarrow HI, HI \rightarrow C, C \rightarrow A\}$

	A	B	C	D	E	H	I
ACD	a1	b12	a3	a4	b15	b16→a6 ⁽¹⁾	b17
BDEH	b21→a1 ⁽³⁾	a2	b23→a3 ⁽²⁾	a4	a5	a6	b27→b17 ⁽¹⁾
CHI	b31→a1 ⁽³⁾	b32	a3	b34	b35	a6	a7

- $A \rightarrow B$
 - non si applica a questa iterazione
- $B \rightarrow AE$
 - non si applica a questa iterazione
- $DI \rightarrow B$
 - ci sono due tuple uguali su D ma non su I , non si applica a questa iterazione
- $D \rightarrow HI$
 - la prima e la seconda riga coincidono sull'attributo $D = a4$, quindi cambiamo H e I ma separatamente $b16 \rightarrow a6$ mentre $b27 \rightarrow b17$
- $HI \rightarrow C$
 - ora abbiamo due tuple uguali su HI (la prima e la seconda entrambe con valori $\langle a6, b17 \rangle$) quindi modifichiamo i valori della C nelle stesse tuple - $b23 \rightarrow a3$
- $C \rightarrow A$
 - le tuple sono tutte uguali su C , quindi le facciamo diventare uguali su A , e poiché abbiamo la prima con valore a , diventano tutte a ($b21 \rightarrow a1$, $b31 \rightarrow a1$)

Abbiamo completato la prima iterazione del ciclo for e la tabella è stata modificata quindi continuiamo

$F = \{A \rightarrow B, B \rightarrow AE, DI \rightarrow B, D \rightarrow HI, HI \rightarrow C, C \rightarrow A\}$

	A	B	C	D	E	H	I
ACD	a1	b12→a2 ⁽¹⁾	a3	a4	b15→a5 ⁽²⁾	a6	b17
BDEH	a1	a2	a3	a4	a5	a6	b17
CHI	a1	b32→a2 ⁽¹⁾	a3	b34	b35→a5 ⁽²⁾	a6	a7

- $A \rightarrow B$
 - le tuple sono tutte uguali su A , quindi le facciamo diventare uguali su B e poiché abbiamo la seconda con valore a , diventano tutte a ($b12 \rightarrow a2$, $b32 \rightarrow a2$)
- $B \rightarrow AE$
 - ora tutte le tuple sono uguali su B , quindi devono diventare uguali anche su AE ; su A sono già uguali, la seconda tupla ha una a sull'attributo E , quindi diventeranno tutte a ($b15 \rightarrow a5$, $b35 \rightarrow a5$)

- $DI \rightarrow B$
 - la prima e la seconda tupla sono uguali su $DI = \langle a4, b17 \rangle$, quindi devono diventare uguali su B ma lo sono già
- $D \rightarrow HI$
 - già soddisfatta, nulla da modificare
- $HI \rightarrow C$
 - già soddisfatta, nulla da modificare
- $C \rightarrow A$
 - già soddisfatta, nulla da modificare

Abbiamo completato la seconda iterazione del ciclo for e la tabella è stata modificata quindi continuiamo

$F = \{A \rightarrow B, B \rightarrow AE, DI \rightarrow B, D \rightarrow HI, HI \rightarrow C, C \rightarrow A\}$							
	A	B	C	D	E	H	I
ACD	a1	a2	a3	a4	a5	a6	b17
BDEH	a1	a2	a3	a4	a5	a6	b17
CHI	a1	a2	a3	b34	a5	a6	a7

- $A \rightarrow B$
 - già soddisfatta, nulla da modificare
- $B \rightarrow AE$
 - già soddisfatta, nulla da modificare
- $DI \rightarrow B$
 - già soddisfatta, nulla da modificare
- $D \rightarrow HI$
 - già soddisfatta, nulla da modificare
- $HI \rightarrow C$
 - già soddisfatta, nulla da modificare
- $C \rightarrow A$
 - già soddisfatta, nulla da modificare

Abbiamo completato l'iterazione del for e la tabella non è stata modificata, quindi l'algoritmo termina. Occorre verificare la presenza della tupla con tutte a
Poiché non c'è una riga con tutte a , il join non è senza perdita

≡ Esempio 3 >

$$R = (A, B, C, D, E, G)$$

$$F = \{AB \rightarrow C, DG \rightarrow B, G \rightarrow D, E \rightarrow G\}$$

$$\rho = \{ABD, AEG, BCE\}$$

Verificare se la decomposizione ρ ha un join senza perdita

Cominciamo a costruire la tabella

	A	B	C	D	E	G
ABD	a1	a2	b13	a4	b15	b16
AEG	a1	b22	b23	b24	a5	a6
BCE	b31	a2	a3	b34	a5	b36

F = { AB → C, DG → B, G → D, E → G }						
	A	B	C	D	E	G
ABD	a1	a2	b13	a4	b15	b16
AEG	a1	b22	b23	b24	a5	a6
BCE	b31	a2	a3	b34	a5	b36 → a6 ⁽¹⁾

- $AB \rightarrow C$
 - non si applica a questa iterazione
- $DG \rightarrow B$
 - non si applica a questa iterazione
- $G \rightarrow D$
 - non si applica a questa iterazione
- $E \rightarrow G$
 - la seconda e la terza riga coincidono sull'attributo $E = a5$, quindi
cambiamo G su $b36 \rightarrow a6$

Abbiamo completato la prima iterazione del ciclo for e la tabella è stata modificata quindi continuiamo

F = { AB → C, DG → B, G → D, E → G }						
	A	B	C	D	E	G
ABD	a1	a2	b13	a4	b15	b16
AEG	a1	b22	b23	b24	a5	a6
BCE	b31	a2	a3	b34 → b24 ⁽¹⁾	a5	a6

- $AB \rightarrow C$
 - non si applica a questa iterazione
- $DG \rightarrow B$
 - non si applica a questa iterazione
- $G \rightarrow D$
 - la seconda e la terza riga coincidono sull'attributo $G = a6$, quindi

rendiamo i valori di D in $b34 \rightarrow b24$

- $E \rightarrow G$
 - non si applica a questa iterazione

Abbiamo completato la seconda iterazione del ciclo for e la tabella è stata modificata quindi continuiamo

$F = \{ AB \rightarrow C, DG \rightarrow B, G \rightarrow D, E \rightarrow G \}$						
	A	B	C	D	E	G
ABD	a1	a2	b13	a4	b15	b16
AEG	a1	$b22 \rightarrow a2^{(1)}$	b23	b24	a5	a6
BCE	b31	a2	a3	b24	a5	a6

- $AB \rightarrow C$
 - non si applica a questa iterazione
- $DG \rightarrow B$
 - la seconda e la terza tupla sono uguali $\langle b24, a6 \rangle$, quindi facciamo diventare uguali le tuple su B ; la terza ha il valore a quindi $b22 \rightarrow a2$
- $G \rightarrow D$
 - non si applica a questa iterazione
- $E \rightarrow G$
 - non si applica a questa iterazione

Abbiamo completato la terza iterazione del ciclo for e la tabella è stata modificata quindi continuiamo

$F = \{ AB \rightarrow C, DG \rightarrow B, G \rightarrow D, E \rightarrow G \}$						
	A	B	C	D	E	G
ABD	a1	a2	b13	a4	b15	b16
AEG	a1	a2	$b23 \rightarrow b13^{(1)}$	b24	a5	a6
BCE	b31	a2	a3	b24	a5	a6

- $AB \rightarrow C$
 - prima e seconda tupla uguali su $AB \langle a1, a2 \rangle$ quindi modifichiamo C in $b23 \rightarrow b13$
- $DG \rightarrow B$
 - non si applica a questa iterazione
- $G \rightarrow D$
 - non si applica a questa iterazione
- $E \rightarrow G$
 - non si applica a questa iterazione

Abbiamo completato la quarta iterazione del ciclo for e la tabella è stata modificata quindi continuiamo

$$F = \{ AB \rightarrow C, DG \rightarrow B, G \rightarrow D, E \rightarrow G \}$$

	A	B	C	D	E	G
ABD	a1	a2	b13	a4	b15	b16
AEG	a1	a2	b13	b24	a5	a6
BCE	b31	a2	a3	b24	a5	a6

- $AB \rightarrow C$
 - non si applica a questa iterazione
- $DG \rightarrow B$
 - non si applica a questa iterazione
- $G \rightarrow D$
 - non si applica a questa iterazione
- $E \rightarrow G$
 - non si applica a questa iterazione

Abbiamo completato l'iterazione del for e la tabella non è stata modificata, quindi l'algoritmo termina. Occorre verificare la presenza della tupla con tutte a
Poiché non c'è una riga con tutte a , il join non è senza perdita

≡ Esempio 4 >

$$R = (A, B, C, D, E, H, I)$$

$$F = \{ H \rightarrow B, DI \rightarrow H, D \rightarrow I, B \rightarrow I, B \rightarrow E, E \rightarrow C \}$$

$$\rho = \{ ABDE, CDH, AHI \}$$

Verificare se la decomposizione ρ ha un join senza perdita

Cominciamo a costruire la tabella

	A	B	C	D	E	H	I
ABDE	a1	a2	b13	a4	a5	b16	b17
CDH	b21	b22	a3	a4	b25	a6	b27
AHI	a1	b32	b33	b34	b35	a6	a7

$$F = \{ H \rightarrow B, DI \rightarrow H, D \rightarrow I, B \rightarrow I, B \rightarrow E, E \rightarrow C \}$$

	A	B	C	D	E	H	I
ABDE	a1	a2	b13	a4	a5	b16	b17 → a7 ⁽³⁾
CDH	b21	b22	a3	a4	b25	a6	b27 → b17 ⁽²⁾ b17 → a7 ⁽³⁾
AHI	a1	b32 → b22 ⁽¹⁾	b33 → a3 ⁽⁵⁾	b34	b35 → b25 ⁽⁴⁾	a6	a7

- $H \rightarrow B$
 - seconda e terza tupla uguali su H quindi le modifichiamo su B :
 $b32 \rightarrow b22$
- $DI \rightarrow H$
 - non si applica a questa iterazione
- $D \rightarrow I$
 - prima e seconda tupla uguali su D quindi le modifichiamo su I :
 $b27 \rightarrow b17$
- $B \rightarrow I$
 - seconda e terza tupla ora sono uguali ($b22$) quindi le modifichiamo su I :
 $b17^{(2)} \rightarrow a7$ (notare che alla prossima iterazione riapplicando $D \rightarrow I$ anche sulla prima tupla potremo trasformare $b17 \rightarrow a7$, quindi possiamo anticipare cioè trasformare tutti i valori che sono già uguali tra loro)
- $B \rightarrow E$
 - seconda e terza tupla uguali su B ($b22$) quindi le modifichiamo su E :
 $b35 \rightarrow b25$
- $E \rightarrow C$
 - seconda e terza tupla uguali su E ($b25$) quindi le modifichiamo su C :
 $b33 \rightarrow a3$

Abbiamo completato la prima iterazione del ciclo for e la tabella è stata modificata quindi continuiamo

$F = \{ H \rightarrow B, DI \rightarrow H, D \rightarrow I, B \rightarrow I, B \rightarrow E, E \rightarrow C \}$							
	A	B	C	D	E	H	I
ABDE	a1	a2	b13	a4	a5	$b16 \rightarrow a6^{(1)}$	a7
CDH	b21	b22	a3	a4	b25	a6	a7
AHI	a1	b22	a3	b34	b25	a6	a7

- $H \rightarrow B$
 - non si applica a questa iterazione
- $DI \rightarrow H$
 - prima e seconda tupla uguali su $DI < a4, a7 >$; le modifichiamo su H :
 $b16 \rightarrow a6$
- $D \rightarrow I$
 - non si applica a questa iterazione
- $B \rightarrow I$
 - non si applica a questa iterazione
- $B \rightarrow E$

- non si applica a questa iterazione
- $E \rightarrow C$
 - non si applica a questa iterazione

Abbiamo completato la seconda iterazione del ciclo for e la tabella è stata modificata quindi continuiamo

$F = \{ H \rightarrow B, DI \rightarrow H, D \rightarrow I, B \rightarrow I, B \rightarrow E, E \rightarrow C \}$							
	A	B	C	D	E	H	I
ABDE	a1	a2	b13 \rightarrow a3 ⁽³⁾	a4	a5	a6	a7
CDH	b21	b22 \rightarrow a2 ⁽¹⁾	a3	a4	b25 \rightarrow a5 ⁽²⁾	a6	a7
AHI	a1	b22 \rightarrow a2 ⁽¹⁾	a3	b34	b25 \rightarrow a5 ⁽²⁾	a6	a7

- $H \rightarrow B$
 - tutte le tuple sono uguali su H , quindi diventano uguali anche su B :
 $b22 \rightarrow a2$
- $DI \rightarrow H$
 - non si applica a questa iterazione
- $D \rightarrow I$
 - non si applica a questa iterazione
- $B \rightarrow I$
 - non si applica a questa iterazione
- $B \rightarrow E$
 - tutte le tuple sono uguali su B , quindi diventano uguali anche su E :
 $b25 \rightarrow a5$
- $E \rightarrow C$
 - tutte le tuple sono uguali su E , quindi diventano uguali anche su C :
 $b13 \rightarrow a3$

Abbiamo completato l'iterazione del for e la tabella è stata modificata, ma poiché c'è una riga con tutte a (la prima), possiamo fermarci e il join è senza perdita

Teorema

Sia R uno schema di relazione, F un insieme di dipendenze funzionali su R e $\rho = \{R_1, R_2, \dots, R_k\}$ una decomposizione di R . L'algoritmo di verifica decide correttamente se ρ ha un join senza perdita

Occorre dimostrare che: ρ ha un join senza perdita ($m_\rho(r) = r$ per ogni r legale) **se e solo se** quando l'algoritmo termina la tabella r ha una tupla con tutte $'a'$

Parte solo se

Supponiamo per assurdo che ρ abbia un join senza perdita ($m_\rho(r) = r$) e che quando l'algoritmo termina la tabella r non abbia una tupla con tutte $'a'$.

La tabella r può essere interpretata come un'istanza legale di R (basta sostituire ai simboli $'a'$ e $'b'$ valori presi dai domini dei corrispondenti attributi in modo tale che ad uno stesso simbolo venga sostituito lo stesso valore) in quando l'algoritmo termina quando non ci sono più violazioni delle dipendenze in F .

Poiché nessun simbolo $'a'$ che compare nella tabella costruita inizialmente viene mai modificato dall'algoritmo, per ogni $i, i = 1, \dots, k$, $\pi_{R_i}(r)$ contiene (fin dall'inizio) una tupla con tutte $'a'$ (quella ottenuta proiettando l'istanza r sugli attributi di R_i e precisamente nella riga corrispondente al sottoschema R_i) pertanto $m_\rho(r)$ contiene sicuramente una tupla con tutte $'a'$ e, quindi, $m_\rho(r) \neq r$ (contraddizione)

Parte se

Per uno sketch della prova della parte “se” consultare il testo J. D. Ullman, “Principles of database and knowledge-base systems”, vol. I, Computer Science Press, 1988.