11– Chiusura di un insieme di attributi

Prof.ssa Maria De Marsico
demarsico@di.uniroma1.it

SAPIENZA
UNIVERSITÀ DI ROMA

Parte del materiale è tratto dal materiale preparato dalla prof.ssa Moscarini per il canale A-L

Cosa vogliamo ottenere

- Quando si decompone uno schema di relazione *R* su cui è definito un insieme di dipendenze funzionali *F*, **oltre** ad ottenere schemi in 3NF occorre
  1) **preservare le dipendenze**
  2) poter **ricostruire tramite join** tutta e sola l'informazione originaria.
- Le dipendenze funzionali che si vogliono preservare sono **tutte** quelle che sono soddisfatte **da ogni istanza legale di *R***, cioè le dipendenze funzionali in F^+.
- Quindi si è interessati a calcolare F^+ …
- E sappiamo come farlo … ma …

Cosa vogliamo ottenere

- …calcolare F^+ richiede tempo **esponenziale** in $|R|$.

- Ricordiamo che se $X \rightarrow Y \in F^+$, per le regole della decomposizione e dell'unione, si ha che $X \rightarrow Z \in F^+$, per ogni $Z \subseteq Y$; pertanto … il calcolo di $|F^+|$ è **esponenziale** in $|R|$.

- **Fortunatamente per i nostri scopi è sufficiente avere un metodo per decidere se una dipendenza funzionale $X \rightarrow Y$ appartiene ad F^+ *(cioè alla chiusura di un insieme di dipendenze)***

- Ciò può essere fatto **calcolando X^+ e verificando se $Y \subseteq X^+$.**
- Infatti ricordiamo il lemma : **$X \rightarrow Y \in F^A$ se e solo se $Y \subseteq X^+$…**
- ***… e il teorema che dimostra che F^A = F^+***

Utilità della chiusura X^+

- Vedremo che il calcolo di X^+ è utile in diversi casi
- Verificare le condizioni perché un insieme di attributi sia chiave di uno schema
- Verificare se una decomposizione preserva le dipendenze funzionali dello schema originario
- ...

Come si calcola X^+

• Per il calcolo della chiusura dell'insieme di attributi *X*, denotata con X^+ , possiamo usare il seguente algoritmo.

Algoritmo «Calcolo di X $^+$»

X può essere un **singolo attributo**

Input uno schema di relazione *R*, un insieme *F* di dipendenze funzionali su *R*, un sottoinsieme *X* di *R*;

Output la chiusura di *X* rispetto ad *F* (**restituita nella variabile *Z***);

begin

$Z:=X;$

$S:=\{A / Y \rightarrow V \in \mathbf{F} \wedge A \in V \wedge Y \subseteq Z\}$

while $S \not\subset Z$

do

begin

$Z:=Z \cup S;$

$S:=\{A / Y \rightarrow V \in \mathbf{F} \wedge A \in$

end

end

Si inseriscono in S i **singoli** attributi che compongono le parti destre di dipendenze **in F** la cui parte sinistra **è contenuta in Z** (in pratica decomponendo le parti destre).

All'inizio Z è proprio X, quindi inseriamo gli attributi che sono **determinati funzionalmente** da X; una volta che questi sono entrati in Z, da questi ne aggiungiamo altri (per **transitività**).

Possiamo «numerare» gli insiemi Z successivi

$Z^{(i)}$ è l'insieme ottenuto dopo la i-esima iterazione del **while**

Come si calcola X^+

- Per il calcolo della chiusura dell'insieme di attributi X, denotata con X^+ , possiamo usare il seguente algoritmo.

Algoritmo «Calcolo di X $^+$»

X può essere un **singolo attributo**

Input uno schema di relazione R, un insieme F di dipendenze funzionali su R, un sottoinsieme X di R;

Output la chiusura di X rispetto ad F (**restituita nella variabile *Z***);

begin

$Z:=X;$

$S:=\{A / Y \rightarrow V \in \mathbf{F} \wedge A \in V \wedge Y \subseteq Z\}$

while $S \not\subset Z$

do

begin

$Z:=Z \cup S;$

$S:=\{A / Y \rightarrow V \in \mathbf{F} \wedge A \in$

end

end

All'iterazione i+1 aggiungiamo in S i **singoli** attributi che compongono le parti destre di dipendenze ***in F*** la cui parte sinistra **è contenuta in $Z^{(i)}$** cioè $S:=\{A / Y \rightarrow V \in \mathbf{F} \wedge A \in V \wedge Y \subseteq \mathbf{Z}^{(i)}\}$

Alla fine di ogni iterazione aggiungiamo qualcosa a Z, **ma non eliminiamo MAI** nessun attributo

L'algoritmo si ferma quando il **nuovo** insieme S che otteniamo è (**già**) contenuto nell'insieme Z, cioè quando **non possiamo** aggiungere **nuovi** attributi alla chiusura transitiva di X.

Vediamo che stiamo usando implicitamente F^A

F= {AB->C, B->D, AD->E, CE->H}
R=ABCDEHL

Vogliamo calcolare la chiusura di AB

Z=AB
S={C, D} AB->C in F , per inserire D AB->B (RIF) + B->D (in F) = AB->D (TRANS)

S ha qualcosa in più?
Z={A,B,C,D}
S={C,D,E } per inserire E AB->B (RIF) + B->D (in F) + AB->AD (AUM) + AD->E (in F) = AB->E (TRANS)

S ha qualcosa in più?
Z= ABCDE
S= CDEH per inserire H AB->C (in F) + AB->AD (AUM di B->D in F) + AD->E (in F) + AB->E (TRANS) + AB->CE (UNIONE) + CE->H (in F) = AB -> H (TRANS)

S ha qualcosa in più?

Z= ABCDEH
S=CDEH

Estendere la variabile Z da cui prendiamo i determinanti nei cicli while successivi equivale ad applicare gli assiomi di Armstrong

S ha qualcosa in più?
STOP

L'algoritmo è corretto

SAPIENZA
Università di Roma
Dipartimento di Informatica

Teorema: L'Algoritmo **«Calcolo di X $^+$»** calcola correttamente la chiusura di un insieme di attributi X rispetto ad un insieme F di dipendenze funzionali.

Dim. Indichiamo con $Z^{(0)}$ il valore iniziale di Z ($Z^{(0)}=X$) e con $Z^{(i)}$ ed $S^{(i)}$, $i \geq 1$, i valori di Z ed S **dopo** l'*i*-esima esecuzione del corpo del ciclo; è facile vedere che $\boldsymbol{Z^{(i)} \subseteq Z^{(i+1)}}$, per ogni i.

Ricordiamo:
*In $Z^{(i)}$ ci sono gli attributi aggiunti a Z **fino** alla i-esima iterazione*
Alla fine di ogni iterazione aggiungiamo qualcosa a Z, **ma non eliminiamo MAI** nessun attributo

Sia j tale che $S(j) \subseteq Z(j)$ (cioè, $Z(j)$ è il valore di Z quando l'algoritmo **termina**); proveremo che:

$\boldsymbol{A \in Z^{(j)}}$ **<u>se e solo</u> se** $\boldsymbol{A \in X^+}$**.**

Teorema

SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

- **Parte solo se.** Mostreremo per **induzione** su *i* che $Z^{(i)} \subseteq X^+$, per **ogni *i*. (**e quindi, in particolare $Z^{(f)} \subseteq X^+$.)

riflessività

Base dell'induzione: i=0. Poiché $\boldsymbol{Z^{(0)} = X}$ e $\boldsymbol{X \subseteq X^+}$, si ha $\boldsymbol{Z^{(0)} \subseteq X^+}$.

Induzione: i>0. Per **l'ipotesi induttiva $Z^{(i-1)} \subseteq X^+$**.

Sia ***A*** un attributo in $\boldsymbol{Z^{(i)} - Z^{(i-1)}}$;

È stato aggiunto **proprio** durante la i-esima iterazione perché **non era** in $\boldsymbol{Z^{(i-1)}}$

deve esistere una dipendenza $\boldsymbol{Y \to V \in F}$ tale che $\boldsymbol{Y \subseteq Z^{(i-1)}}$ e $\boldsymbol{A \in V}$. Poiché $\boldsymbol{Y \subseteq Z^{(i-1)}}$, **per l'ipotesi induttiva si ha che $Y \subseteq X^+$;** pertanto, per il **Lemma**, $\boldsymbol{X \to Y \in F^A}$. Poiché $\boldsymbol{X \to Y \in F^A}$ e $\boldsymbol{Y \to V \in F}$, per l'assioma della **transitività** si ha $\boldsymbol{X \to V \in F^A}$ e quindi, per il **Lemma** , $\boldsymbol{V \subseteq X^+}$. Pertanto, **per ogni $A \in Z^{(i)} - Z^{(i-1)}$** si ha $\boldsymbol{A \in X^+}$. Da ciò segue, per l'ipotesi induttiva, che $\boldsymbol{Z^{(i)} \subseteq X^+}$.

Gli attributi in $\boldsymbol{Z^{(i-1)}}$ *ci sono per ipotesi induttiva e abbiamo mostrato che ci vanno anche quelli inseriti in Z* ***all'i-esima iterazione*** *del ciclo*

Teorema

SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

- **Parte se.** Sia ***A* un attributo in *X*$^+$** e sia *j* tale che $S^{(j)} \subseteq Z^{(j)}$ (cioè, $Z^{(j)}$ è il valore di *Z* quando l'algoritmo si termina). **Mostreremo che $A \in Z^{(j)}$.**

  Poiché $A \in X^+$, si ha $X \to A \in F^+$ (per il Teorema); pertanto $X \to A$ deve essere soddisfatta da **ogni** istanza legale di *R*. Si consideri la seguente istanza *r* di *R*:

$$\overbrace{\quad\quad\quad\quad\quad\quad}^{Z^{(j)}} \quad \overbrace{\quad\quad\quad\quad\quad\quad}^{R-\,Z^{(j)}}$$

r

1	1	…	1	1	1	…	1
1	1	…	1	0	0	…	0

- **Mostriamo prima che *r* è un'istanza legale di *R*.**

Teorema

SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

	$\overbrace{\qquad\qquad\qquad\qquad}^{Z^{(j)}}$				$\overbrace{\qquad\qquad\qquad\qquad}^{R- Z^{(j)}}$			
r	1	1	…	1	1	1	…	1
	1	1	…	1	0	0	…	0

•**Mostriamo prima che *r* è un'istanza legale di *R*.**

Infatti, se,

per assurdo,

esistesse in *F* **una dipendenza funzionale $V \to W$ non soddisfatta da *r***,

si **dovrebbe** avere $V \subseteq Z^{(j)}$ (i valori delle due tuple **sono uguali SOLO** in quel sottoinsieme di R), <u>e **serve che siano uguali** per poter dire che la dipendenza NON E' soddisfatta</u>)

e $W \cap (R- Z^{(j)}) \neq \varnothing$; (i valori delle due tuple <u>**sono diversi SOLO** in quel sottoinsieme di R</u>) **ma**, in tal caso, si avrebbe $S^{(j)} \not\subset Z^{(j)}$ **(contraddizione**).

PERCHE'?

Teorema

$Z^{(j)}$ | **R-** $Z^{(j)}$

r	**1**	**1**	**…**	**1**	**1**	**1**	**…**	**1**
	1	**1**	**…**	**1**	**0**	**0**	**…**	**0**

Abbiamo assunto che j è il valore per cui $S^{(j)} \subseteq Z^{(j)}$ (all'iterazione j **non abbiamo aggiunto NIENTE** di nuovo, e quindi da lì in poi non potremo farlo neppure continuando).
Se fosse $V \subseteq Z^{(j)}$ e $W \cap (R- Z^{(j)}) \neq \emptyset$ ***avrei <u>qualche elemento </u>di W <u>non è ancora </u>in $Z^{(j)}$; applicando l'algoritmo alla iterazione j+1 potrei ancora raccogliere in S questi NUOVI elementi tramite V→W e poi inserirli in $Z^{(j+1)}$*** **. L'algoritmo però si ferma <u>solo quando non è più </u>possibile inserire <u>nuovi</u> elementi in Z, che significa che da quel punto in poi continueremmo ad ottenere <u>sempre</u> lo stesso insieme S che è stato già aggiunto a Z, e quindi siamo ad una contraddizione**

Teorema

- **Parte se (continua)**

	$Z^{(j)}$				$R - Z^{(j)}$			
r	**1**	**1**	**…**	**1**	**1**	**1**	**…**	**1**
	1	**1**	**…**	**1**	**0**	**0**	**…**	**0**

•Poiché *r* è un'istanza legale di *R* **deve** soddisfare ***X→A*; (che è in F $^{+}$)** (se esistono due tuple uguali su X devono essere uguali anche su A). Esistono due tuple uguali su X? **CERTO!**

$X = Z^{(0)} \subseteq Z^{(j)}$, e le due tuple sono uguali su TUTTI gli attributi in $Z^{(j)}$; ma allora le due tuple DEVONO essere uguali anche su A, che, allora, **deve essere in $Z^{(j)}$.**

Osservazione: proprietà dell'insieme vuoto

- Prima di tutto va sottolineato che la notazione $\{\varnothing\}$ indica **l'insieme che contiene l'insieme vuoto (insieme di insiemi)** e **non va** pertanto confusa con il semplice **insieme vuoto** $\varnothing$
- l'insieme vuoto è un sottoinsieme di ogni insieme *A*:

  $\forall A : A \supseteq \varnothing$
- l'unione di un qualunque insieme *A* con l'insieme vuoto è *A*:

  $\forall A : A \cup \varnothing = A$
- l'intersezione di un qualunque insieme *A* con l'insieme vuoto è l'insieme vuoto: $\forall A : A \cap \varnothing = \varnothing$
- il prodotto cartesiano di un qualunque insieme *A* con l'insieme vuoto è l'insieme vuoto: $\forall A : A \times \varnothing = \varnothing$
- l'unico sottoinsieme dell'insieme vuoto è l'insieme vuoto stesso
- il numero di elementi dell'insieme vuoto (vale a dire la sua cardinalità) è zero; l'insieme vuoto è quindi finito: $|\varnothing| = 0$

Esempio

- Dato lo schema di relazione R = (A, B, C, D, E, H) e il seguente insieme di dipendenze funzionali su R

F = { AB→ CD, EH→ D, D→ H }

calcolare le chiusure degli insiemi A, D e AB

R = (A, B, C, D, E, H) F = { AB→ CD, EH→ D, D→ H }

begin

*Z:=**A**;*

$S:=\{L \mid Y \rightarrow V \in \mathbf{F} \wedge L \in V \wedge Y \subseteq A\} = \emptyset$ (A da solo non determina alcun altro attributo)

controllo while (*S⊄Z ?*): $\emptyset \subset$ A **quindi non** entriamo nella prima iterazione del while

end

A $^{+}$**= *A***

R = (A, B, C, D, E, H)   F = { AB→ CD, EH→ D, D→ H }

begin

$Z := \mathbf{D};$

$S := \{L \mid Y \rightarrow V \in \mathbf{F} \wedge L \in V \wedge Y \subseteq D\}$= H (per la dipendenza D→ H)

controllo while (*S⊄Z ?*): ***H*⊄** D **quindi** entriamo nella prima iterazione del while

begin (prima iterazione del while)

$Z := Z \cup S = D \cup H = \mathbf{DH}$

$S := \{L \mid Y \rightarrow V \in \mathbf{F} \wedge L \in V \wedge Y \subseteq DH\}$ = H (per la dipendenza D→ H)

end

controllo while: H⊂ DH (non abbiamo aggiunto niente di nuovo)

usciamo dal while

end

$\mathbf{D^{+} = DH}$

R = (A, B, C, D, E, H)   F = { AB→ CD, EH→ D, D→ H }

begin

$Z:=\mathbf{AB};$

$S:=\{A \mid Y\rightarrow V \in \mathbf{F} \wedge A \in V \wedge Y \subseteq AB\}$= CD (per la dipendenza AB→ CD)=CD

controllo while (*$S \not\subset Z$?*): ***CD*** $\not\subset$ AB **quindi** entriamo nella prima iterazione del while

begin (prima iterazione del while)

$Z:=Z \cup S = AB \cup CD = \mathbf{ABCD}$

$S:=\{A \mid Y\rightarrow V \in \mathbf{F} \wedge A \in V \wedge Y \subseteq ABCD\}$ = {C, D (per la dipendenza AB→ CD), <u>H</u> (per la dipendenza D→ H)}=CDH

end

controllo while: *CDH* $\not\subset$ ABCD (abbiamo aggiunto H, quindi entriamo nella seconda iterazione)

begin (seconda iterazione del while)

$Z:=Z \cup S = ABCD \cup CDH = \mathbf{ABCDH}$

$S:=\{A \mid Y\rightarrow V \in \mathbf{F} \wedge A \in V \wedge Y \subseteq ABCDH\}$ = {C, D (per la dipendenza AB→ CD), H (per la dipendenza D→ H)}=CDH

end

controllo while: *CDH* $\subset$ ABCDH (non abbiamo aggiunto nulla di nuovo)

usciamo dal while

end

$\mathbf{AB^{+} = ABCDH}$

Esercizio

SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

- Dato lo schema di relazione R = (A, B, C, D, E, H, I) e il seguente insieme di dipendenze funzionali su R

F = { A→ E, AB→ CD, EH→ I, D→ H }

calcolare la chiusura dell'insieme AB

- La successione delle Z e delle S sarà

Z= AB

S= CDE (per A→ E, AB→ CD)

Ciclo while

Iter. 1 parte con Z= ABCDE, calcola S= CDE<u>H</u> (A→ E, AB→ CD, <u>D→ H</u>)

Iter. 2 parte con Z= ABCDEH, calcola S= CDEH<u>I</u> (..., D→ H, <u>EH→I</u>)

Iter. 3 parte con Z= ABCDEHI, calcola S= CDEHI (...)

CDEH I $\subset$ ABCDEHI quindi usciamo dal ciclo while

AB^+=ABCDEHI AB determina funzionalmente tutto lo schema ...