IL PROGETTO DI UNA BASE DI DATI RELAZIONALE

1. La Terza Forma Normale

Immaginiamo di dover progettare una base di dati relazionale contenente i dati degli studenti e dei corsi di un' Università.
La prima soluzione che potrebbe venire in mente è di progettare un'unica relazione

Università(Matr,Nome,Città,Prov,C#,Titolo,Docente, C_laurea,Data,Voto)

in cui una tupla *(m,n,c,p,C,t,D,l,d,v)* rappresenta il fatto che uno studente con matricola *m* e nome *n*, residente nella città *c* che si trova in provincia di *p*, ha sostenuto l'esame del corso, con codice *C* e titolo *t*, tenuto dal docente *D*, del corso di laurea *l* in data *d* riportando il voto *v*.
Adottando questa soluzione si avrebbero un certo numero di inconvenienti che vanno sotto il nome di

- **anomalie di inserimento**: non posso inserire i dati di uno studente se non ha sostenuto almeno un esame;
- **anomalie di cancellazione**: se cancello i dati di un corso (perché il corso è stato disattivato) e c'è uno studente che ha sostenuto solo l'esame relativo a quel corso perdo le informazioni sullo studente;
- **anomalie di aggiornamento**: se devo modificare il docente di un corso devo farlo per ogni tupla in cui compare il corso;
- inoltre si ha
- **ridondanza nella rappresentazione dei dati:** le informazioni anagrafiche di uno studente sono ripetute per ogni esame sostenuto dallo studente.

Queste anomalie sono dovute al fatto che si sono rappresentati in un'unica relazione più concetti: il concetto *Studente*, il concetto *Corso* e il concetto *Esame*. Rappresentando i tre concetti in tre relazioni distinte

Studente(Matr,Nome,Città,Prov)
Corso(C#,Titolo,Docente, C_laurea)
Esame(Matr, C#, Data,Voto)

tali anomalie vengono eliminate. Tuttavia, è possibile riscontrare il permanere di simili anomalie concernenti le città:

- non posso inserire il fatto che una città si trova in una certa provincia se non c'è uno studente che risiede in quella città
- non posso cancellare il fatto che una città sta in una certa provincia senza perdere i dati di tutti gli studenti che risiedono in quella città
- per modificare la provincia in cui si trova una città devo modificare tutte le tuple relative a studenti che risiedono in quella città.

Per eliminare queste ultime anomalie che hanno la stessa origine di quelle viste precedentemente (in un'unica relazione sono stati rappresentati il concetto *Studente* e il concetto *Città*) posso utilizzare lo schema seguente:

Studente(Matr,Nome,Città,Prov)
Comune(Città,Prov)
Corso(C#,Titolo,Docente, C_laurea)
Esame(Matr, C#, Data,Voto).

Come può essere formalizzato questo discorso? Vale a dire: ci sono proprietà formali degli schemi di relazione che ci permettono di dire che uno schema è migliore di un altro?

Se analizziamo le prime anomalie vediamo che sono legate al fatto che mentre nella relazione ci sono dati (*Voto* e *Data*) che sono determinati univocamente da *Matr* e *C#*, i dati di uno studente sono determinati univocamente da *Matr* e i dati di un corso sono determinati univocamente da *C#.*
Se analizziamo le seconde vediamo che sono legate al fatto che la provincia in cui risiede uno studente è determinata univocamente dalla matricola dello studente solo in quanto lo studente risiede in una sola città e una città si trova in una sola provincia.

Il concetto di "determina univocamente" è colto dal concetto formale di dipendenza funzionale.

Definizione 1 Dato uno schema di relazione *R* (che d'ora in avanti assumiamo essere un insieme di attributi) una *dipendenza funzionale* su *R* è una coppia ordinata di sottoinsiemi non vuoti *X* ed *Y* di *R* e viene denotata come $X \rightarrow Y$.

Inoltre, diciamo che un'istanza r di R *soddisfa la dipendenza funzionale* $X \rightarrow Y$ se per ogni coppia di tuple t_1 e t_2 di r si ha che **se** $t_1[X] = t_2[X]$ **allora** $t_1[Y] = t_2[Y]$ *(in altre parole* $t_1[X] = t_2[X] \Rightarrow t_1[Y] = t_2[Y]$ *dove* $\Rightarrow$ ha il consueto significato di "implica"**)**.
Se F è un insieme di dipendenze funzionali su R ed r è un'istanza di R che soddisfa tutte le dipendenze in F, diciamo che r è un'*istanza legale* di R. Infine la *chiusura di F,* denotata con F^+, è l'insieme di dipendenze funzionali che sono soddisfatte da ogni istanza legale di R.

Banalmente si ha che $F \subseteq F^+$.

Il concetto di dipendenza funzionale permette di definire formalmente il concetto di chiave.

Definizione 2 Dato uno schema di relazione R, un insieme di dipendenze funzionali F su R e un sottoinsieme K di R diciamo che K è una *chiave* per R se:
- $K \rightarrow R \in F^+$,
- per ogni sottoinsieme proprio K' di K, $K' \rightarrow R \notin F^+$.

Nel seguito se $A_1,\ldots,A_n$ sono attributi in R e X ed Y sono sottoinsiemi di R, denoteremo:
- con $A_1 \ldots A_n$ l'insieme $\{A_1,\ldots,A_n\}$
- con XY l'insieme $X \cup Y$.

Considerando lo schema di relazione *Università* possiamo osservare che un'istanza di *Università* per rispecchiare la realtà di interesse deve soddisfare le seguenti dipendenze funzionali su Università:
- *Matr C#→Nome Città ProvTitolo Docente C_laurea Data Voto*
- *C# →Titolo Docente C_laurea*
- *Matr→Nome Città Prov*
- *Città→Prov*

Quindi *Matr C#* costituisce una chiave per *Università*. Inoltre, ci sono attributi che "dipendono parzialmente" dalla chiave (ad es., *Titolo Docente C_laurea* dipendono funzionalmente da *C#*) e attributi che "dipendono transitivamente" dalla chiave (ad es., *Prov* dipende funzionalmente da *Matr* in quanto dipende funzionalmente da *Città* che dipende funzionalmente da *Matr*).
Le dipendenze funzionali sullo schema finale saranno invece

Matr→Nome Città Prov	su	*Studente(Matr,Nome,Città,Prov)*
Città→Prov	su	*Comune(Città,Prov)*
C# →Titolo Docente C_laurea	su	*Corso(C#,Titolo,Docente, C_laurea)*
Matr C#→Data Voto	su	*Esame(Matr, C#, Data,Voto).*

e quindi in nessuno schema di relazione ci sono attributi che dipendono parzialmente né transitivamente dalla chiave. Uno schema di relazione in cui non ci sono sono attributi che dipendono parzialmente né transitivamente dalla chiave è detto in terza forma normale (3NF)

Per formalizzare i concetti introdotti, diamo alcune definizioni preliminari.

Definizione 3 Dati uno schema di relazione R e un insieme di dipendenze funzionali F su R diciamo che:
- un attributo A di R è *primo* se appartiene ad una chiave di R
- un sottoinsieme X di R è una *superchiave* se contiene una chiave di R.

Definizione 4 Siano R uno schema di relazione e F un insieme di dipendenze funzionali su R.
- $X \rightarrow A \in F^+$ tale che $A \notin X$ è una *dipendenza parziale* su R se A non è primo ed X è contenuto propriamente in una chiave di R.
- $X \rightarrow A \in F^+$ tale che $A \notin X$ è una *dipendenza transitiva* su R se A non è primo e **per ogni chiave** K di R si ha che X non è contenuto propriamente in K e $K - X \neq \emptyset$.

Definizione 5 Siano R uno schema di relazione e F un insieme di dipendenze funzionali su R. R è *in 3NF* se per ogni dipendenza funzionale $X \rightarrow A \in F^+$ tale che $A \notin X$ si ha che
- A è primo, oppure
- X è una superchiave.

Teorema 1 Siano *R* uno schema di relazione e *F* un insieme di dipendenze funzionali su *R*. Uno schema *R* è in 3NF se e solo se non esistono né dipendenze parziali né dipendenze transitive in *R*.
Dim. *Parte solo se* . Banale
Parte se. Supponiamo per assurdo che *R* non sia in 3NF; in tal caso esiste una dipendenza funzionale $X \rightarrow A$ F^+ con $A \notin X$ t.c. *A* non è primo e *X* non è una superchiave. Poiché *X* non è una superchiave due casi (mutuamente esclusivi) sono possibili:

- o per ogni chiave *K* di *R* *X* non è contenuto propriamente in *K* e $K\text{-}X \neq \emptyset$; in tal caso $X \rightarrow A$ è una dipendenza transitiva su *R* (contraddizione), oppure
- o esiste una chiave *K* di *R* tale che $X \subset K$; in tal caso $X \rightarrow A$ è una dipendenza parziale su *R* (contraddizione).

Abbiamo visto che uno schema in 3NF ha delle buone proprietà che lo rendono preferibile ad uno che non è in 3NF; pertanto, un obbiettivo da tenere presente quando si progetta una base di dati è quello di produrre uno schema in cui ogni relazione sia in 3NF.
Normalmente nella fase di progettazione concettuale si usa il modello Entità-associazione. Nella fase di progettazione concettuale si individuano per l'appunto i concetti che devono essere rappresentati nella base di dati. Se questo lavoro di individuazione è fatto accuratamente lo schema relazionale che può essere derivato in modo automatico con opportune regole, è in 3NF. Se tuttavia dopo tale processo ci trovassimo a produrre uno schema che non è in 3NF dovremmo procedere ad una fase di decomposizione di tale schema in maniera analoga a quella esaminata nell'esempio sui dati di un'Università.

Uno schema che non è in 3NF può essere decomposto in più modi in un insieme di schemi in 3NF. Ad esempio lo schema $R=ABC$ con l'insieme di dipendenze funzionali $F\{A \rightarrow B,\ B \rightarrow C\ \}$(lo schema non è in 3NF per la presenza in F^+ della dipendenza transitiva $B \rightarrow C$) può essere decomposto in:
$R1=AB$ con $\{A \rightarrow B\}$ e
$R2=BC$ con $\{B \rightarrow C\}$
oppure
$R1=AB$ con $\{A \rightarrow B\}$ e
$R2=AC$ con $\{A \rightarrow C\}$
Entrambi gli schemi sono in 3NF. In entrambi i casi dopo la decomposizione le istanze dei due schemi saranno gestite in maniera indipendente. Tuttavia, la seconda soluzione non è soddisfacente. Infatti, consideriamo l'istanza della base di dati costituita dalle due istanze legali di R1 e R2

R1

A	B
a1	b1
a2	b1

R2

A	C
a1	c1
a2	c2

L'istanza di *R* che posso ricostruire da questa (l'unico modo è di ricostruirla facendo un join naturale!)

R

A	B	C
a1	b1	c1
a2	b1	c2

non è un'istanza legale di *R*, in quanto non soddisfa la dipendenza funzionale $B \rightarrow C$. Gli attributi coinvolti "direttamente" come determinante e dipendente nella dipendenza sono in schemi diversi e non è possibile (senza complessi vincoli aggiuntivi) continuare a garantire che la dipendenza venga soddisfatta. ATTENZIONE: vedremo che il fatto che gli attributi delle dipendenze si trovino in schemi diversi non implica automaticamente che le dipendenze non possano essere soddisfatte.

Consideriamo ora lo schema $R=ABC$ con l'insieme di dipendenze funzionali $F\{A \rightarrow B,\ C \rightarrow B\}$ (lo schema non è in 3NF per la presenza in F^+ delle dipendenze parziali $A \rightarrow B$ e $C \rightarrow B$) tale schema può essere decomposto in:
$R1=AB$ con $\{A \rightarrow B\}$ e
$R2=BC$ con $\{C \rightarrow B\}$.
Tale decomposizione preserva sicuramente tutte le dipendenze in F^+ in quanto gli attributi "direttamente" coinvolti nella stessa dipendenza si trovano nello stesso schema. Tuttavia, la decomposizione non è soddisfacente. Infatti, consideriamo l'istanza legale di *R*

R

A	B	C
a1	b1	c1
a2	b1	c2

Una istanza di una relazione è la rappresentazione di una fotografia della realtà di interesse in un certo istante (nel caso particolare nella realtà di interesse sono veri i due fatti *(a1,b1,c1)* e *(a2,b1,c2)* e non altri). Pertanto, quando si decompone uno schema si vuole che ogni sua istanza sia ricostruibile da un'istanza dello schema ottenuto dalla decomposizione. Nell'esempio considerato l'istanza dovrebbe essere ricostruita mediante join naturale da:

R1

A	B
a1	b1
a2	b1

R2

B	C
b1	c1
b1	c2

Viceversa, effettuando il join naturale delle due relazioni si ottiene:

R

A	B	C
a1	b1	c1
a2	b1	c2
a1	b1	c2
a2	b1	c1

In conclusione, quando si decompone uno schema per ottenerne uno in 3NF occorre tenere presenti altri due requisiti dello schema decomposto:

a. deve preservare le dipendenze funzionali che valgono su ogni istanza legale dello schema originario
b. deve permettere di ricostruire mediante join naturale ogni istanza legale dello schema originario.

2. Decomposizioni che preservano le dipendenze

Quando si decompone uno schema di relazione R su cui è definito un insieme di dipendenze funzionali F, 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^+.

Sia R uno schema di relazione e F un insieme di dipendenze funzionali. Sia F^A l'insieme di dipendenze funzionali definito nel modo seguente:

a. se $f \in F$ allora $f \in F^A$
b. se $Y \subseteq X \subseteq R$ allora $X \rightarrow Y \in F^A$ (**assioma della riflessività**)
c. se $X \rightarrow YF^A$ allora $XZ \rightarrow YZ \in F^A$, per ogni $Z \subseteq R$ (**assioma dell'aumento**)
d. se $X \rightarrow Y \in F^A$ e $Y \rightarrow Z \in F^A$ allora $X \rightarrow Z \in F^A$ (**assioma della transitività**)

Dimostreremo che $F^+ = F^A$, cioè che la chiusura di un insieme di dipendenze funzionali F può essere ottenuta a partire da F applicando ricorsivamente gli assiomi della riflessività, dell'aumento e della transitività, conosciuti come *assiomi di Armstrong*:

Nel seguito saranno utili tre regole (che sono conseguenza degli assiomi di Armstrong) che consentono di derivare da dipendenze funzionali in F^A altre dipendenze funzionali in F^A. Le tre regole sono:

- se $X \rightarrow Y \in F^A$ e $X \rightarrow Z \in F^A$ allora $X \rightarrow YZ \in F^A$ (**regola dell'unione**)
- se $X \rightarrow Y \in F^A$ e $Z \subseteq Y$ allora $X \rightarrow Z \in F^A$ (**regola della decomposizione**)
- se $X \rightarrow Y \in F^A$ e $WY \rightarrow Z \in F^A$ allora $WX \rightarrow Z \in F^A$ (**regola della pseudotransitività**).

Il seguente risultato mostra che le tre regole sono corrette.

Teorema 2 Sia F un insieme di dipendenze funzionali. Valgono le seguenti implicazioni:

a. se $X \rightarrow Y \in F^A$ e $X \rightarrow Z \in F^A$ allora $X \rightarrow YZ \in F^A$
b. se $X \rightarrow Y \in F^A$ e $Z \subseteq Y$ allora $X \rightarrow Z \in F^A$
c. se $X \rightarrow Y \in F^A$ e $WY \rightarrow Z \in F^A$ allora $WX \rightarrow Z \in F^A$

Dim.
Prova di a).
Se $X \rightarrow Y \in F^A$**,** per l'assioma dell'aumento si ha $X \rightarrow XY \in F^A$**.** Analogamente**,** se $X \rightarrow Z \in F^A$**,** per l'assioma dell'aumento si ha $XY \rightarrow YZ \in F^A$**.** Quindi, poiché $X \rightarrow XY \in F^A$ e $XY \rightarrow YZ \in F^A$, per l'assioma della transitività si ha $X \rightarrow YZ \in F^A$.
Prova di b).
Se $Z \subseteq Y$ allora, per l'assioma della riflessività, si ha $Y \rightarrow Z \in F^A$. Quindi, poiché $X \rightarrow Y \in F^A$ e $Y \rightarrow Z \in F^A$. per l'assioma della transitività si ha $X \rightarrow Z \in F^A$.
Prova di c).
Se $X \rightarrow Y \in F^A$, per l'assioma dell'aumento si ha $WX \rightarrow WY \in F^A$**.** Quindi, poiché $WX \rightarrow WY \in F^A$ e $WY \rightarrow Z \in F^A$, per l'assioma della transitività si ha $WX \rightarrow Z \in F^A$.

Si osservi che, per la regola dell'unione, se $X \rightarrow Ai \in F^A$, $i=1,\dots,n$, allora $X \rightarrow A_1, A_2,\dots, An \in F^A$ e che, per la regola della decomposizione, se $X \rightarrow A_1, A_2,\dots, A_n \in F^A$ allora $X \rightarrow A_i \in F^A$, $i1,\dots,n$; cioè:

$$XA_1, A_2,\dots, A_n \in F^A \text{ se e solo se } XA_i \in F^A,\ i=1,\dots,n.$$

Pertanto, possiamo limitarci, quando necessario, a considerare solo dipendenze funzionali in cui il membro destro sia un singleton.

Allo scopo di dimostrare che $F^{+}=F^A$, introduciamo il concetto di chiusura di un insieme di attributi rispetto ad un insieme di dipendenze funzionali.

Definizione 6 Siano R uno schema di relazione, F un insieme di dipendenze funzionali su R e X un sottoinsieme di R. La *chiusura* di X rispetto ad F, denotata con X^+_F (o semplicemente X^+, se non sorgono ambiguità) è definito nel modo seguente: $X^+_F=\{A|X \rightarrow A \in F^A\}$.

Lemma 1 Siano R uno schema di relazione ed F un insieme di dipendenze funzionali su R. Si ha che: $X \rightarrow Y \in F^A$ se e solo se $Y \subseteq X^+$.
Dim. Sia $Y=A_1, A_2,\dots, A_n$.
Parte se. Poiché $Y \subseteq X^+$, per ogni i, $i= 1,\dots,n$, si ha che $X \rightarrow Ai \in F^A$. Pertanto, per la regola dell'unione, $X \rightarrow Y \in F^A$.
Parte solo se. Poiché $X \rightarrow Y \in F^A$, per la regola della decomposizione si ha che, per ogni i, $i=1,\dots,n$, $X \rightarrow A_i \in F^A$, cioè $Ai \in X^+$, per ogni i, $i=1,\dots,n$, e, quindi, $Y \subseteq X^+$.

A questo punto siamo in grado di dimostrare che $F^+=F^A$.

Teorema 3 Siano R uno schema di relazione ed F un insieme di dipendenze funzionali su R. Si ha $F^+ = F^A$.
Dim.
$\boldsymbol{F^+ \supseteq F^A}$**.** Sia $X \rightarrow Y$ una dipendenza funzionale in F^A. Dimostriamo che $X \rightarrow Y \in F^+$ per induzione sul numero i di applicazioni di uno degli assiomi di Armstrong.
Base dell'induzione: $i=0$. In tal caso $X \rightarrow Y$ è in F e quindi, banalmente, $X \rightarrow Y$ è in F^+.
Induzione: i. Per l'ipotesi induttiva ogni dipendenza funzionale ottenuta a partire da F applicando gli assiomi di Armstrong un numero di volte minore o uguale a $i-1$ è in F^+. Tre casi si possono presentare

a. $X \rightarrow Y$ è stata ottenuta mediante l'assioma della riflessività, in tal caso $Y \subseteq X$. Sia r un'istanza di R e siano t_1 e t_2 due tuple di r tali che $t_1[X] = t_2[X]$; banalmente si ha $t_1[Y] = t_2[Y]$.
b. $X \rightarrow Y$ è stata ottenuta applicando l'assioma dell'aumento ad una dipendenza funzionale $V \rightarrow W$ in F^A, ottenuta applicando ricorsivamente gli assiomi di Armstrong un numero di volte minore o uguale a $i-1$, tale che $X=VZ$ e $Y=WZ$, per qualche $Z \subseteq R$. Sia r un'istanza legale di R e siano t_1 e t_2 due tuple di r tali che $t_1[X] = t_2[X]$; banalmente si ha che $t_1[V] = t_2[V]$ e $t_1[Z] = t_2[Z]$. Per l'ipotesi induttiva da $t_1[V] = t_2[V]$ segue $t_1[W] = t_2[W]$; da $t_1[W] = t_2[W]$ e $t_1[Z] = t_2[Z]$ segue $t_1[Y] = t_2[Y]$.
c. $X \rightarrow Y$ è stata ottenuta applicando l'assioma della transitività a due dipendenze funzionale $X \rightarrow Z$ e $Z \rightarrow Y$ in F^A, ottenute applicando ricorsivamente gli assiomi di Armstrong un numero di volte minore o uguale a $i-1$. Sia r un'istanza di R e siano t_1 e t_2 due tuple di r tali che $t_1[X] = t_2[X]$. Per l'ipotesi induttiva da $t_1[X] = t_2[X]$ segue $t_1[Z] = t_2[Z]$; da $t_1[Z] = t_2[Z]$, ancora per l'ipotesi induttiva segue $t1[Y] = t_2[Y]$.

$\boldsymbol{F^+ \subseteq F^A}$**.** Supponiamo per assurdo che esista una dipendenza funzionale $X \rightarrow Y \in F^+$ tale che $X \rightarrow Y \notin F^A$. Mostreremo che esiste un'istanza legale di R che non soddisfa XY (che contraddice il fatto che XYF^+).
Consideriamo la seguente istanza r di R:

X^+ R-X^+

|_____________| |__________|

r	1	1	…	1	1	1	…	1
	1	1	…	1	0	0	…	0

Mostriamo che:

a. r è un'istanza legale di R. Sia $V\to W$ una dipendenza funzionale in F e supponiamo per assurdo che non sia soddisfatta da r. In tal caso le due tuple di r devono avere gli stessi valori per V e differenti valori per W; ciò implica che $V\subseteq X^+$ e $W\cap(R\text{-}X^+)\neq\emptyset$. Poiché $V\subseteq X^+$, per il Lemma1, si ha che $X\to V\in F^A$; pertanto, per l'assioma della transitività, $X\to W\in F^A$ e, quindi, per il Lemma 1, $W\subseteq X^+$ (che contraddice $W\cap(R\text{-}X^+)\neq\emptyset$)).

a. In quanto istanza legale, r soddisfa tutte le dipendenze in F^+ *e in particolare* $X\to Y$. Poiché $X\subseteq X^+$ (per l'assioma della riflessività) le due tuple di r coincidono sugli attributi X e quindi, poiché r soddisfa $X\to Y$, devono coincidere anche sugli attributi Y. Questo implica che $Y\subseteq X^+$ e quindi, per il Lemma 1, che $X\to Y \in F^A$.

Quando diciamo che una decomposizione deve preservare le dipendenze intendiamo che deve preservare F^+. Calcolare F^+, tuttavia, richiede tempo esponenziale in F. Infatti se $X\to Y\in F^+$, per le regole della decomposizione e dell'unione, si ha che $X\to Z\in F^+$, per ogni $Z\subseteq Y$; pertanto F^+ è esponenziale nel numero di attributi dello schema R. D'altra parte, per i nostri scopi è sufficiente poter decidere se una dipendenza funzionale $X\to Y$ appartiene ad F^+; ciò può essere fatto calcolando X^+ e verificando se $Y\subseteq X^+$. Il calcolo di X^+ può essere fatto mediante il seguente algoritmo polinomiale.

Algoritmo 1

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 (nella variabile Z);

begin

$Z:=X;$

$S:=\{A|Y\to V\in F \wedge A\in V\wedge Y\subseteq Z\}$

while $S \not\subseteq Z$

  do

  begin

  $Z:=Z\cup S;$

  $S:=\{A|Y\to V\in F \wedge A\in V\wedge Y\subseteq Z\}$

  end

end

Teorema 4 L'Algoritmo 1 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 $Z^{(i)}\subseteq Z^{(i+1)}$, per ogni i. Proveremo che:

esiste i tale che $A\in Z^{(i)}$ se e solo se $A\in X^+$.

Parte solo se. Mostreremo per induzione su i che $Z^{(i)}\subseteq X^+$, per ogni i.

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

Induzione: i>0. Per l'ipotesi induttiva $Z^{(i-1)} \subseteq X^+$. Sia A un attributo in $Z^{(i)}$-$Z^{(i-1)}$; deve esistere una dipendenza $Y\to V \in F$ tale che $Y\subseteq Z^{(i-1)}$ e $A\in V$. Poiché $Y\subseteq Z^{(i-1)}$, per l'ipotesi induttiva si ha che $Y\subseteq X^+$; pertanto, per il Lemma 1, $X\to Y\in F^A$. Poiché $X\to Y\in F^A$ e $Y\to V\in F$, per l'assioma della transitività si ha $X\to V\in F^A$ e quindi, per il Lemma 1, $V\subseteq X^+$. Pertanto, per ogni $A\in Z^{(i)}$-$Z^{(i-1)}$ si ha $A\in X^+$. Da ciò segue, per l'ipotesi induttiva, che $Z^{(i)}\subseteq X^+$.

Parte se. Sia A un attributo in X^+ e sia j tale che $S^{(j)}=Z^{(j)}$ (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 $X\to A$ F^+ (per il Teorema 3); pertanto $X\to A$ deve essere soddisfatta da ogni istanza legale di R. Si consideri la seguente istanza r di R:

|___ $Z^{(j)}$ ___| |_ R- $Z^{(j)}$ _|

r	1	1	…	1	1	1	…	1
	1	1	…	1	0	0	…	0

Mostriamo che r è un'istanza legale di R. Infatti, se, per assurdo, esistesse in F una dipendenza funzionale $V \rightarrow W$ non soddisfatta da r, si dovrebbe avere $V \subseteq Z^{(j)}$ e $W \cap (R- Z^{(j)}) \neq \emptyset$; ma, in tal caso, si avrebbe $S^{(j)} \not\subset Z^{(j)}$ (contraddizione). Poiché r è un'istanza legale di R deve soddisfare $X \rightarrow A$ *che si trova in* F^A *e per il teorema 3 in* F^+; ma, allora, poiché $X=Z^{(0)} \subseteq Z^{(j)}$, A deve essere in $Z^{(j)}$.

Vogliamo ora formalizzare il concetto di decomposizione che "preserva un insieme di dipendenze funzionali". A tal fine, cominciamo con l'introdurre i concetti di decomposizione di uno schema di relazione e di equivalenza tra due insiemi di dipendenze funzionali.

Definizione 7 Sia R uno schema di relazione. Una *decomposizione* di R è una *famiglia* $\rho=\{R_1, R_2,\ldots, R_k\}$ di sottoinsiemi di R che ricopre R ($\cup_{i=1}^k Ri=R$).

Definizione 8 Siano F e G due insiemi di dipendenze funzionali. F e G sono *equivalenti* ($F \equiv G$) se $F^+=G^+$.

Verificare l'equivalenza di due insiemi F e G di dipendenze funzionali richiede dunque che venga verificata l'uguaglianza di F^+ e G^+, cioè che $F^+ \subseteq G^+$ e che $F^+ \supseteq G^+$. Come detto in precedenza, calcolare la chiusura di un insieme di dipendenze funzionali richiede tempo esponenziale. Il seguente lemma ci permette tuttavia di verificare l'equivalenza dei due insiemi di dipendenze funzionali in tempo polinomiale.

Lemma 2 Siano F e G due insiemi di dipendenze funzionali. Se $F \subseteq G^+$ allora $F^+ \subseteq G^+$.
Dim. Sia $f \in F^+-F$. Poiché, per il Teorema 3, f è derivabile da F mediante gli assiomi di Armstrong e ogni dipendenza funzionale in F è derivabile da G mediante gli assiomi di Armstrong, f è derivabile da G mediante gli assiomi di Armstrong.

Definizione 9 Sia R uno schema di relazione, F un insieme di dipendenze funzionali su R e $\rho=\{R_1, R_2,\ldots, R_k\}$ una decomposizione di R.
Diciamo che ρ *preserva* F se $F \equiv \cup_{i=1}^k \pi_{Ri}(F)$, dove $\pi_{Ri}(F) = \{X \rightarrow Y \mid X \rightarrow Y \in F^+ \wedge XY \subseteq R_i\}$.

Verificare se una decomposizione preserva un insieme di dipendenze funzionali F richiede, dunque, che venga verificata l'equivalenza dei due insiemi di dipendenze funzionali F e $G=\cup_{i=1}^k \pi_{Ri}(F)$; poiché, per definizione, $F^+ \supseteq G$ (scritto anche come $G \subseteq F^+$), per il Lemma 2 è sufficiente verificare che $F \subseteq G^+$; ciò può essere fatto con il seguente algoritmo (la cui correttezza è una banale conseguenza del Lemma 1 e del Teorema 3).

Algoritmo 2
Input due un insiemi F e G di dipendenze funzionali su R;
Output la variabile *successo* (al termine avrà valore *true* se $F \subseteq G^+$, *false*, altrimenti)
begin
successo:true;
for every $X \rightarrow Y \in \in F$
do **begin**
calcola X^+_G;
if $Y \not\subset X^+_G$ **then** *successo:false*
end
end

L'Algoritmo 2 richiede che venga calcolato X^+_G; se si volesse utilizzare a tale scopo l'Algoritmo 1 si dovrebbe prima calcolare G, ma, per la definizione di G, ciò richiederebbe il calcolo di F^+ che richiede tempo esponenziale. Il seguente algoritmo, la cui correttezza è dimostrata nel successivo Teorema 5, permette di calcolare X^+_G a partire da F.

Algoritmo 3

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, un sottoinsieme X di R;

Output la chiusura di X rispetto a $G=\cup_{i=1}^{k} \pi_{Ri} (F)$, (nella variabile Z);

begin
Z:=X;
S:=∅;
for *j:1* **to** *k*
  do $S:=S\cup(Z \cap R_j)^+_F \cap R_j$;
while $S \not\subset Z$
  do
  begin
  Z:=Z∪S
  for *j:1* **to** *k*
    do $S:=S\cup(Z \cap R_j)^+_F \cap R_j$;
  end
end

Teorema 5 Sia R uno schema di relazione, F un insieme di dipendenze funzionali su R, $\rho=\{R_1, R_2,\dots, R_k\}$ una decomposizione di R e X un sottoinsieme di R. L'Algoritmo 3 calcola correttamente X^+_G, dove $G=\cup_{i=1}^{k} {}_{Ri} (F)$.
Dim. Indichiamo con $Z^{(0)}$ il valore iniziale di Z ($Z^{(0)}=X$) e con $Z^{(i)}$, $i\geq 1$, il valore di Z dopo l'i-esima esecuzione dell'assegnazione $Z:=Z\cup S$; è facile vedere che $Z^{(i)}\subseteq Z^{(i+1)}$, per ogni i. Sia $Z^{(f)}$ il valore di Z quando l'algoritmo termina; proveremo che:

$$A\in Z^{(f)} \text{ se e solo se } A\in X^+_G$$

Parte solo se. Mostreremo per induzione su i che $Z^{(i)}\subseteq X^+_G$, per ogni i.
Base dell'induzione: i=0. Poiché $Z^{(0)}=X$ e $X\subseteq X^+$, si ha $Z^{(0)} \subseteq X^+_G$.
Induzione: i>0.
Per l'ipotesi induttiva $Z^{(i-1)}\subseteq X^+_G$. Sia A un attributo in $Z^{(i)}-Z^{(i-1)}$; in tal caso deve esistere un indice j tale che $A\in(Z^{(i-1)}\cap R_j)^+_F\cap R_j$. Poiché $A\in(Z^{(i-1)}\cap R_j)^+_F$ si ha $(Z^{(i-1)} \cap R_j)\to A\in F^+$ (per il Teorema 3). Poiché $(Z^{(i-1)} \cap R_j)\to A\in F^+$, $A\in R_j$ e $Z^{(i-1)}\cap R_j\subseteq R_j$ si ha, per la definizione di G, che $(Z^{(i-1)} \cap R_j)\to A \in G$. Poiché per l'ipotesi induttiva si ha che $X\to Z^{(i-1)}\in G^+$, per la regola di decomposizione si ha anche che $X\to(Z^{(i-1)}\cap R_j)\in G^+$ e, quindi, per l'assioma della transitività, che $X\to A\in G^+$, cioè $A\in X^+_G$. Quindi $Z^{(i)}\subseteq X^+_G$.
Parte se. Mostreremo che $X^+G\subseteq Z^{(f)}$, dove $Z^{(f)}$ è il valore di Z quando l'algoritmo termina.
Osservazione:
  (1) presi comunque due insiemi di attributi X e Y e un insieme di dipendenze funzionali F si ha (per la definizione di chiusura di un insieme di attributi):
    se $X\subseteq Y$ allora $X^+_F\subseteq Y^+_F$
Poichè $X=Z^{(0)} \subseteq Z^{(f)}$, da (1) segue che $X^+_G\subseteq (Z^{(f)})^+_G$. Mostreremo che $Z^{(f)}=(Z^{(f)})^+_G$ da cui segue $X^+_G\subseteq Z^{(f)}$.

Supponiamo per assurdo che $Z^{(f)}$ sia il valore finale di Z dell'algoritmo che calcola X^+_G, ma che $Z^{(f)} \neq(Z^{(f)})^+_G$. Consideriamo l'Algoritmo 1 e utilizziamolo per il calcolo di $(Z^{(f)})^+_G$ ($Z^{(f)}$ è un insieme di attributi); per evitare ambiguità con l'algoritmo di cui stiamo dimostrando la correttezza, riscriviamo sostituendo la variabile W che conterrà la chiusura alla variabile Z e la variabile U alla variabile S:

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

Output la chiusura di X rispetto a G (nella variabile W);

begin
W:=X;
$U:=\{A|Y\to V\in G \wedge A\in V \wedge Y\subseteq W\}$
while U non è contenuto in W
  do
  begin
  $W:=W\cup U;$
  $U:=\{A|Y\to V\in G \wedge A\in V \wedge Y\subseteq W\}$
  end
end

Se eseguiamo tale algoritmo fornendo in input l'insieme di attributi $Z^{(f)}$ e l'insieme di dipendenze funzionali *G,* al termine la variabile *W* conterrà $(Z^{(f)})^+_G$.
Se, come abbiamo supposto per assurdo, $Z(f) \neq (Z^{(f)})^+_G$, deve esistere un attributo *B* che appartiene a $U^{(0)}$ *e* non appartiene a $W^{(0)}=Z^{(f)}$ (altrimenti ci saremmo fermati e si avrebbe $Z(f)= (Z^{(f)})^+_G$).
D'altra parte si ha:

$$U^{(0)}=\{A \mid Y\rightarrow V\in G \wedge A\in V \wedge Y\subseteq W^{(0)} \}$$

E quindi, per la definizione di *G*, perché B sia in questo insieme deve esistere *j* tale che:

$$B\in\{A \mid Y\rightarrow V\in F^+ \wedge YV\subseteq R_j \wedge A\in V \wedge Y\subseteq W^{(0)} \}.$$

Da $Y\subseteq W^{(0)} \wedge YV\subseteq R_j$ *segue che :*

$$(a)\ Y\subseteq W^{(0)}\cap R_j = Z^{(f)} \cap R_j$$

Inoltre dal fatto che $Y\rightarrow V\in F^+$ *segue per il Lemma 1 che* $V\subseteq Y^+_F$ e, quindi, per la (a) e per l'osservazione (1) si ha che

$$V\subseteq (Z^{(f)} \cap R_j)^+_F.$$

Infine poichè $YV\subseteq R_j$ *si ha:*

$$V\subseteq (Z^{(f)} \cap R_j)^+_F\cap R_j.$$

Poiché $B\in V$ *si ha che* $B\in(Z^{(f)} \cap Rj)^+_F\cap R_j$ *e quindi* $B\in S^{(f)}$ nell'esecuzione dell'algoritmo che calcola X^+_G. Poiché *B* non appartiene a $Z^{(f)}$ (per l'ipotesi per assurdo), $Z^{(f)}$ non può essere il valore finale di *Z* (contraddizione).

3. Decomposizioni che hanno un join senza perdita

Come si è detto nel paragrafo 1, se si decompone uno schema di relazione *R* che non è in 3NF si vuole che la decomposizione $\rho=\{R_1, R_2,\dots, R_k\}$ 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 $\{R_1, R_2,\dots, R_k\}$. Poiché per ricostruire una tupla *t* di *r* è necessario che $t[R_i]\in r_i$, $i=1,\dots,k$, si deve avere $r_i=\pi_{Ri}(r)$, $i=1,\dots,k$.

Definizione 10 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 $r=\pi_{R1}(r) \bowtie \pi_{R2}(r) \bowtie\dots\bowtie\pi_{Rk}(r)$.

Teorema 6 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 $m_\rho(r)=\pi_{R1}(r) \bowtie \pi_{R2}(r) \bowtie\dots\bowtie\pi_{Rk}(r)$ si ha:

a. $r \subseteq m_\rho(r)$
b. $\pi_{Ri}(m_\rho(r))= \pi_{Ri}(r)$
c. $m_\rho(m_\rho(r)) = m_\rho(r)$.

Dim.
Prova di a). Sia *t* una tupla di *r*. Per ogni *i*, $i=1,\dots,k$, $t[R_i]\in\pi_{Ri}(r)$ e quindi $t\in m\rho(r)$.
Prova di b). Per a) si ha $r\subseteq m(r)$ e, quindi, $\pi_{Ri}(r)\subseteq\pi_{Ri}(m_\rho(r))$. E' sufficiente, pertanto, mostrare che $\pi_{Ri}(r)\supseteq\pi_{Ri}(m_\rho(r))$. Banalmente, per ogni tupla $t\in m\rho(r)$ e per ogni *i*, $i=1,\dots,k$, deve esistere una tupla $t'\in r$ tale $t[R_i]=t'[R_i]$.
Prova di c). Per b) si ha $\pi_{Ri}(m_\rho(r))=\pi_{Ri}(r)$. Pertanto

$$m_\rho(m\rho(r))=\pi_{R1}(m_\rho(r)) \bowtie\pi_{R2}(m_\rho(r)) \bowtie \dots\bowtie\pi_{Rk}(m_\rho(r))=\pi_{R1}(r) \bowtie \pi_{R2}(r) \bowtie \dots\bowtie\pi_{Rk}(r)=m_\rho(r).$$

Il seguente algoritmo permette di decidere in tempo polinomiale se una decomposizione di uno schema di relazione ha un join senza perdita.

Algoritmo 4
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 ha un join senza perdita;
begin
Costruicsi 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
end

Teorema 7 Sia R uno schema di relazione, F un insieme di dipendenze funzionali su R e $\rho=\{R_1, R_2,\ldots, R_k\}$ una decomposizione di R. L'Algoritmo 4 decide correttamente se ha un join senza perdita.
Dim. Occorre dimostrare che:

ha un join senza perdita
se e solo se
quando l'algoritmo termina la tabella *r* ha una tupla con tutte *'a'*.

Dimostreremo la parte "solo se" .
Parte solo se. Supponiamo per assurdo che ρ abbia un join senza perdita 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 quanto 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, $i=1,\ldots,k$, $\pi_{R_i}(r)$ contiene una tupla con tutte *'a'*; pertanto $m_\rho(r)$ contiene una tupla con tutte *'a'* e, quindi, $m_\rho(r) \neq r$ (contraddizione).

Corollario Sia R uno schema di relazione, F un insieme di dipendenze funzionali su R e $\rho=\{R_1, R_2\}$ una decomposizione di R. Se $R_1 \cap R_2 \rightarrow R_1 - R_2$ o $R_1 \cap R2 \rightarrow R2 - R_1$ allora ρ ha un join senza perdita.

4. Decomposizioni in 3NF che conservano le dipendenze funzionali e hanno un join senza perdita

In questo paragrafo mostreremo che dato uno schema di relazione R e un insieme di dipendenze funzionali F su R esiste sempre una decomposizione $\rho=\{R_1, R_2,\ldots, R_k\}$ di R tale che:

a. per ogni i, $i=1,\ldots,k$, R_i è in 3NF
b. ρ preserva F
c. ρ ha un join senza perdita

e che una tale decomposizione può essere calcolata in tempo polinomiale.

A tal fine abbiamo bisogno di introdurre il concetto di copertura minimale di un insieme di dipendenze funzionali.

Definizione, 11 Sia F un insieme di dipendenze funzionali. Una *copertura minimale* di F è un insieme G di dipendenze funzionali equivalente ad F tale che:

1) per ogni dipendenza funzionale in G la parte destra è un singleton, cioè è costituita da un unico attributo (ogni attributo nella parte destra è non ridondante)
2) per nessuna dipendenza funzionale $X \rightarrow A$ in G esiste $X' \subseteq X$ tale che $G \equiv G-\{X \rightarrow A\} \cup \{X' \rightarrow A\}$ (ogni attributo nella parte sinistra è non ridondante)
3) per nessuna dipendenza funzionale $X \rightarrow A$ in G, $G \equiv G-\{X \rightarrow A\}$ (ogni dipendenza è non ridondante).

Per ogni insieme di dipendenze funzionali F esiste una copertura minimale che può essere ottenuta in tempo polinomiale a partire dall'insieme G equivalente ad F in cui per ogni dipendenza funzionale la parte destra è un singleton (G esiste sempre per la regola della decomposizione)

- prima sostituendo ricorsivamente ogni dipendenza funzionale $A_1,\ldots,A_{i-1},A_i,A_{i+1},\ldots,An \rightarrow A$ tale che $G \equiv G-\{A_1,\ldots,A_{i-1},A_i,A_{i+1},\ldots,A_n \rightarrow A\} \cup \{A_1,\ldots,A_{i-1},A_{i+1},\ldots,A_n \rightarrow A\}$ con la dipendenza funzionale $A_1,\ldots,A_{i-1},A_{i+1},\ldots,A_n \rightarrow A$
- e successivamente eliminando ricorsivamente ogni dipendenza funzionale $X \rightarrow A$ tale che $G \equiv G-\{X \rightarrow A\}$.

Il seguente algoritmo, dato uno schema di relazione R e un insieme di dipendenze funzionali F su R, che è una copertura minimale, permette di calcolare in tempo polinomiale una decomposizione $\rho=\{R_1, R_2,\ldots, R_k\}$ di R tale che:

a. per ogni i, $i=1,\ldots,k$, R_i è in 3NF
b. ρ preserva F

Algoritmo 5

Input uno schema di relazione R e un insieme F di dipendenze funzionali su R, che è una copertura minimale;

Output una decomposizione ρ di R che preserva F e tale che per ogni schema di relazione in ρ è in 3NF;

begin
$\rho:=\emptyset;$
$S:=\emptyset;$
for every $A\in R$ tale che A non è coinvolto in nessuna dipendenza funzionale in F **do**
    $S:=S\cup A;$
if $S\neq\emptyset$ **then**
    begin
    $R:=R-S;$
    $\rho:=\rho\cup\{S\}$
    end
if esiste una dipendenza funzionale in F che coinvolge tutti gli attributi in R
    then $\rho:=\rho\cup\{R\}$
    else for every $X\rightarrow A\in F$ **do** $\rho:=\rho\cup\{XA\}$
end

Teorema 8 Sia R uno schema di relazione ed F un insieme di dipendenze funzionali su R, che è una copertura minimale. L'Algoritmo 5 permette di calcolare in tempo polinomiale una decomposizione ρ di R tale che:

- ogni schema di relazione in ρ è in 3NF
- ρ preserva F.

Dim.

***ρ* preserva *F*.** Sia $G=\cup_{i=1}^{k}\pi_{R_i}(F)$. Poiché per ogni dipendenza funzionale $X\rightarrow AF$ si ha che $XA\in\rho$, si ha che $G\supseteq F$ e, quindi $G^+\supseteq F^+$. L'inclusione $G^+\subseteq F^+$ è banalmente verificata in quanto, per definizione, $G\subseteq F^+$.

Ogni schema di relazione in *ρ* è in 3NF.

Ricordiamo che siamo in presenza di una decomposizione, quindi l'insieme F minimale viene proiettato sui sottoschemi ottenuti e implica che nella valutazione della 3NF dobbiamo tenere in considerazione le dipendenze appartenenti a queste proiezioni.

Se $S\in\rho$, ogni attributo in S fa parte della chiave (è primo) e quindi, banalmente, S è in 3NF.

Se $R\in\rho$ esiste una dipendenza funzionale in F che coinvolge tutti gli attributi in R. Poiché F è una copertura minimale tale dipendenza avrà la forma $R-A\rightarrow A$; poiché F è una copertura minimale, non ci può essere una dipendenza funzionale $X\rightarrow A$ in F tale che $X\subseteq R-A$ e, quindi, $R-A$ è chiave in R. Sia YB una qualsiasi dipendenza in F; se $B=A$ allora, poiché F è una copertura minimale, $Y=R-A$ (cioè, Y è una superchiave); se $B\neq A$ allora $B\in R-A$ e quindi B è primo.

Se $XA\in\rho$, poiché F è una copertura minimale, non ci può essere una dipendenza funzionale $X'\rightarrow A$ in F tale che $X'\subseteq X$ e, quindi, X è chiave in XA. Sia YB una qualsiasi dipendenza in F tale che $YB\subseteq XA$; se $B=A$ allora, poiché F è una copertura minimale, $Y=X$ (cioè, Y è una superchiave); se $B\neq A$ allora $B\in X$ e quindi B è primo.

Teorema 9 Sia R uno schema di relazione, F un insieme di dipendenze funzionali su R, che è una copertura minimale e ρ la decomposizione di R prodotta dall'Algoritmo 5. La decomposizione $\sigma=\rho\cup\{K\}$, dove K è una chiave per R, è tale che:

a. ogni schema di relazione in σ è in 3NF
b. σ preserva F
c. σ ha un join senza perdita.

Dim.

***σ* preserva *F*.** Poiché ρ preserva F anche σ preserva F.

Ogni schema di relazione in *σ* è in 3NF. Poiché $\sigma=\rho\cup\{K\}$, è sufficiente verificare che lo schema di relazione K è in 3NF. Infatti, se ci fosse una dipendenza funzionale $X\rightarrow A$ in F^+, con $XA\subseteq K$, A è primo.

σ **ha un join senza perdita.** Supponiamo che l'ordine in cui gli attributi in *R-K* vengono aggiunti a *Z* dall'Algoritmo 1 quando calcola K^+ sia $A_1,A_2,\ldots,A_n$, e supponiamo che per ogni *i*, *i=1,…,n*, l'attributo A_i venga aggiunto a *Z* a causa della presenza in *F* della dipendenza $Yi \rightarrow Ai$ dove:

$$Yi \subseteq Z^{(i-1)}=KA_1A_2\ldots A_{i-1} \subseteq K^+.$$

Per dimostrare che σ ha un join senza perdita mostreremo che quando l'Algoritmo 4 è applicato a σ viene prodotta una tabella che ha una riga con tutte '*a*'.
Senza perdita di generalità, supponiamo che l'Algoritmo 4 esamini le dipendenze funzionali $Y_1 \rightarrow A_1, Y2 \rightarrow A2, \ldots, Yn \rightarrow An$ in questo ordine. Dimostreremo per induzione su *i* che dopo che è stata considerata la dipendenza funzionale $Yi \rightarrow Ai$ nella riga che corrisponde allo schema di relazione *K* c'è una '*a*' in ogni colonna j con $j \leq i$.
Base dell'induzione: i=1. Poiché $Y_1 \subseteq Z^{(0)}=K$, sia nella riga che corrisponde allo schema di relazione Y_1A_1 che in quella che corrisponde allo schema di relazione *K* ci sono tutte '*a*' in corrispondenza agli attributi in Y_1; inoltre nella riga che corrisponde allo schema di relazione Y_1A_1 c'è una '*a*' in corrispondenza ad A_1. Pertanto l'Algoritmo 4 pone una '*a*' in corrispondenza ad A_1 nella riga che corrisponde allo schema di relazione *K*.
Induzione: i>1. Per l'ipotesi induttiva, nella riga che corrisponde allo schema di relazione *K* c'è una '*a*' in corrispondenza ad ogni attributo A_j con $j \leq i-1$. Poiché $Yi \subseteq KA1A2\ldots A_{i-1}$, sia nella riga che corrisponde allo schema di relazione Y_iA_i che in quella che corrisponde allo schema di relazione *K* ci sono tutte '*a*' in corrispondenza agli attributi in Y_i; inoltre nella riga che corrisponde allo schema di relazione Y_iA_i c'è una '*a*' in corrispondenza ad A_i. Pertanto l'Algoritmo 4 pone una '*a*' in corrispondenza ad A_i nella riga che corrisponde allo schema di relazione *K*.

5. La forma normale di Boyce-Codd

Successivamente alla terza forma normale sono state definite altre forme normali per gli schemi di relazione, alcune delle quali sono basate su vincoli (dipendenze multivalore e dipendenze di join) più generali delle dipendenze funzionali. Una forma normale che ancora si basa sul concetto di dipendenza funzionale è la cosidetta forma normale di Boyce-Codd.

Definizione 12 Siano *R* uno schema di relazione e *F* un insieme di dipendenze funzionali su *R*. *R* è *in forma normale di Boyce-Codd* se per ogni dipendenza funzionale $X \rightarrow A \in F^+$ tale che $A \notin X$ si ha che *X* è una superchiave.

Come è evidente dalla definizione, la forma normale di Boyce-Codd è più restrittiva della terza forma normale: ogni schema di relazione in forma normale di Boyce-Codd è in terza forma normale, ma non è vero il viceversa. Infatti, consideriamo il seguente schema di relazione,

Orario_lezioni=Aula Giorno Ora Corso

su cui sono definite le seguenti dipendenze funzionali

Aula Giorno Ora→Corso

(in una certa aula in un certo giorno ad una certa ora viene tenuto un solo corso)

Corso→Aula

(un corso viene tenuto sempre nella stessa aula).
Poiché la chiave di *Orario_lezioni* è *Aula Giorno Ora,* nella dipendenza funzionale *Corso→Aula* la parte sinistra non è una superchiave e, quindi, lo schema di relazione *Orario_lezioni* non è in forma normale di Boyce-Codd pur essendo in terza forma normale. In uno schema che non è in forma normale di Boyce-Codd si presenta ancora il problema della ridondanza nella rappresentazione dei dati (ad esempio in *Orario_lezioni* il fatto che un corso si tiene in una certa aula è ripetuto per ogni ora in cui il corso viene tenuto). Sorge quindi il problema di capire se per la forma normale di Boyce-Codd valgono risultati analoghi a quelli visti nei paragrafi precedenti per la terza forma normale, vale a dire se è valida la seguente affermazione:

"Dato uno schema di relazione *R* e un insieme di dipendenze funzionali *F* su *R* esiste sempre una decomposizione $\rho=\{R_1, R_2,\ldots, R_k\}$ di *R* tale che:

a. per ogni *i*, *i=1,…,k*, R_i è in forma normale di Boyce-Codd
b. ρ preserva *F*
c. ρ ha un join senza perdita"

La risposta a questa domanda è negativa. Infatti, è evidente che qualsiasi decomposizione di *Orario_lezioni* che non contenga lo schema di relazione *Aula Giorno Ora Corso*, non permette di rappresentare la dipendenza funzionale *Aula Giorno Ora→Corso*. Tuttavia è possibile dimostrare che dato uno schema di relazione *R* e un insieme di dipendenze funzionali *F* su *R* esiste sempre una decomposizione $\rho=\{R_1, R_2,\ldots, R_k\}$ di *R* tale che:

a. per ogni *i*, *i=1,…,k*, R_i è in forma normale di Boyce-Codd
b. ha un join senza perdita
e che esiste un algoritmo che produce una tale decomposizione in tempo polinomiale.
Tale algoritmo fornirebbe nel caso dell’esempio esaminato la decomposizione:

{R1=Aula Corso, R2=Giorno Ora Corso}.