Basi di Dati
La teoria della normalizzazione

Corso B

La normalizzazione

- L'argomento si inquadra nella buona progettazione di una base dati
- In laboratorio si è vista la progettazione E-R, seguita dal passaggio di progettazione logica che permette di ottenere lo schema logico relazionale
- Entro certi principi lo schema è valido

Linea di progettazione alternativa

- Presenteremo ora una linea alternativa di progettazione basata sulla teoria della normalizzazione

- Inizialmente conviene "dimenticare" l'approccio E-R → progettazione logica

- L'approccio che tratteremo è un **approccio teorico** tutto interno al **modello relazionale**

Approccio teorico

1. In qualche modo (non ci interessa come) abbiamo ottenuto uno schema logico di una base dati relazionale

2. Si effettua un'analisi dello schema secondo le metodologie della normalizzazione

3. L'analisi consentirà si trasformare lo schema iniziale della base di dati in uno schema normalizzato

Normalizzazione vs. E-R

I due approcci non sono in antitesi, ma complementari

La normalizzazione permette anzi di capire meglio il funzionamento dell'E-R e della progettazione logica

L'argomento verrà affrontato nel dettaglio alla fine della trattazione sulla teoria della normalizzazione

Obiettivi

Gli obiettivi generali della buona progettazione sono:

- esprimibilità delle informazioni
- efficienza
- leggibilità degli schemi

Esempio

Immaginiamo una relazione ESAMI

ESAMI(MATR, NomeS, IndirizzoS, CAPS, CodiceFiscaleS, DataNascitaS, Corso, Voto, DataEsame, CodProf, NomeProf, Qualifica, TipoUfficio)

Useremo delle abbreviazioni

ESAMI(MATR, NS, IS, CAP, CF, DN, Co, Vo, DE, CP, NP, Q, TU)

Dal punto di vista del modello relazionale è una relazione valida

Criticità delle relazioni

Istanziamo la relazione con dei dati ragionevoli

MATR	**NS**	**IS**	**CAP**	**CF**	**DN**	**Co**	**Vo**	**DE**	**CP**	**NP**	**Q**	**TU**
341	Piero	TO	101	PX	1990	BD	27	1/4/15	P1	Pensa	Ric	ExLab
343	Giorgio	NO	102	GY	1991	Prog	21	2/5/15	B1	Bono	ProfA	Piano1
341	Piero	TO	101	PX	1990	Prog	25	3/5/15	B1	Bono	ProfA	Piano1
343	Giorgio	NO	102	GY	1991	BD	18	6/4/15	P1	Pensa	Ric	ExLab
…	…	…	…	…	…	…	…	…	…	…	…	…

Supponiamo che ESAMI sia l'unica relazione che descrive il sistema informativo, essa contiene infatti:

- Anagrafica studenti
- Anagrafica docenti
- Esami superati

Criticità

Criticità di esprimibilità

- Anomalie di inserzione
- Anomalie di cancellazione

Sono criticità derivanti dal sistema informativo

Anomalie di inserzione

- Un nuovo studente intende immatricolarsi
- Lo studente, ovviamente, non ha dato nessun esame
- Ho bisogno di inserire una nuova tupla

<u>MATR</u>	NS	IS	CAP	CF	DN	<u>Co</u>	Vo	DE	CP	NP	Q	TU
341	Piero	TO	101	PX	1990	BD	27	1/4/15	P1	Pensa	Ric	ExLab
343	Giorgio	NO	102	GY	1991	Prog	21	2/5/15	B1	Bono	ProfA	Piano1
341	Piero	TO	101	PX	1990	Prog	25	3/5/15	B1	Bono	ProfA	Piano1
343	Giorgio	NO	102	GY	1991	BD	18	6/4/15	P1	Pensa	Ric	ExLab
444	**Laura**	**TO**	**101**	**LZ**	**1992**	**NULL**	**NULL**	**NULL**	**NULL**	**NULL**	**NULL**	**NULL**

- Una parte della tupla non sarà valorizzata

Anomalie di inserzione

Problema: non posso inserire la nuova tupla, dato che Corso è parte della chiave principale e non può avere valori nulli

MATR	NS	IS	CAP	CF	DN	Co	Vo	DE	CP	NP	Q	TU
341	Piero	TO	101	PX	1990	BD	27	1/4/15	P1	Pensa	Ric	ExLab
343	Giorgio	NO	102	GY	1991	Prog	21	2/5/15	B1	Bono	ProfA	Piano1
341	Piero	TO	101	PX	1990	Prog	25	3/5/15	B1	Bono	ProfA	Piano1
343	Giorgio	NO	102	GY	1991	BD	18	6/4/15	P1	Pensa	Ric	ExLab
444	**Laura**	**TO**	**101**	**LZ**	**1992**	**NULL**	**NULL**	**NULL**	**NULL**	**NULL**	**NULL**	**NULL**

L'anomalia consiste nell'incapacità di inserire una informazione **concettualmente significativa** proveniente dal sistema informativo

Anomalie di inserzione

- Viene assunto un nuovo docente
- Il docente non ha ancora firmato alcun esame
- Voglio inserirlo nel DB

MATR	**NS**	**IS**	**CAP**	**CF**	**DN**	**Co**	**Vo**	**DE**	**CP**	**NP**	**Q**	**TU**
341	Piero	TO	101	PX	1990	BD	27	1/4/15	P1	Pensa	Ric	ExLab
343	Giorgio	NO	102	GY	1991	Prog	21	2/5/15	B1	Bono	ProfA	Piano1
341	Piero	TO	101	PX	1990	Prog	25	3/5/15	B1	Bono	ProfA	Piano1
343	Giorgio	NO	102	GY	1991	BD	18	6/4/15	P1	Pensa	Ric	ExLab
NULL	**NULL**	**NULL**	**NULL**	**NULL**	**NULL**	**NULL**	**NULL**	**NULL**	**R1**	**Rossi**	**Ric**	**Piano2**

Non posso inserirlo perché non posso valorizzare la chiave

Anomalia di cancellazione

- Alcune informazioni contenute nella base dati non posso essere cancellate

- Ad esempio, se un docente va in pensione, cancellando ogni riferimento ad esso, elimino anche delle informazioni importanti relative ai suoi esami

- Anomalia di cancellazione: impossibilità di cancellare informazioni significative dal punto di vista del sistema informativo

Anomalia di update

Criticità di efficienza

- Anomalia di update che implica inconsistenza potenziale

Immaginiamo di aggiornare l'indirizzo di recapito dello studente 341 che ha registrato già molti esami

Dal punto di vista concettuale, se voglio una base di dati consistente, la modifica deve interessare tutte le tuple in cui appare lo studente 341

Anomalia di update

Immaginiamo ora che il docente *P1* cambi ufficio

Se vogliamo una basi di dati consistente la modifica va apportata a tutte le tuple corrispondenti agli esami firmati da *P1*

Se anche una sola tupla non viene modificata (update incompleto) lo stato della base di dati diventa concettualmente inconsistente

Inoltre la modifica interessa un numero enorme di tuple (un docente a fine carriera avrà firmato migliaia di esami)

Considerazione generale

Le relazioni, quindi, possono dare origine ad anomalie di tre forme

- Anomalia di inserzione (esprimibilità)
- Anomalia di cancellazione (esprimibilità)
- Anomalia di update (esprimibilità ed efficienza)

Per quanto riguarda la leggibilità abbiamo problemi di

- sinonimie
- omonimie

Azioni necessarie

- Minimizzazione delle anomalie, ovvero trasformare la relazione in modo da eliminare il più possibile le anomalie

- Per farlo, occorre procedere alla **normalizzazione**

- Per sinonimie e omonimie occorre invece seguire principi di **standardizzazione**

Osservazione

- Abbiamo considerato una relazione più o meno significativa dal punto di vista di un sistema informativo noto
- In base alla comune conoscenza del sistema informativo abbiamo materializzato esempi di presenza di anomalie
- Siamo ancora su un piano puramente intuitivo
- Cercheremo ora di matematizzare queste intuizioni, cioè presenteremo la teoria formale della normalizzazione

Approccio scientifico al problema

- Percezione del problema
  - Obiettivi di buona progettazione
- Tassonomia dei fatti salienti
  - Anomalie di inserzione, cancellazione, update
- Ricerca di una formalizzazione adeguata
  - **Dipendenze funzionali**
- Messa a punto di una teoria ben fondata
- Conseguenze della teoria
- Applicazioni della teoria

Formalizzazione del problema

Questa tematica fa emergere una classe di vincoli di base di dati molto interessanti ai nostri fini

Tali vincoli sono le **dipendenze funzionali**

Dipendenza funzionale

Nel sistema informativo "segreteria studenti", alcuni attributi caratterizzano le cosiddette **molecole informative**, esempi

- Studenti caratterizzati da *MATR, NomeS, IndirizzoS…*
- Docenti caratterizzati da *CodProf, NomeProf, Qualifica…*

Mi serve una formalizzazione dell'idea di molecola informativa rimanendo nel modello relazionale (usando cioè attributi e valori)

Dipendenza funzionale

Astrattamente, dal punto di vista del sistema informativo, con la Matricola dello studente do automaticamente una caratterizzazione dello studente

C'è cioè una correlazione tra la matricola dello studente e la sua descrizione in termini di attributi

MATR → NomeS, IndirizzoS, CAPS, CodiceFiscaleS, DataNascitaS

Dobbiamo ora formalizzare questa correlazione all'interno del modello relazionale

Dipendenza funzionale

- Immaginiamo uno studente X tra i migliaia presenti
- Immaginiamo la tupla *t* dello studente X associata ad un esame
- Immaginiamo un'altra tupla *s* relativa ad un altro esame sostenuto dallo stesso studente X
- Ci aspettiamo che nella tupla s la matricola sia abbinata allo stesso Nome, Indirizzo, CAP, Codice Fiscale e Data di Nascita di *t*
- **Da qui il vincolo!**

Dipendenza funzionale

Ogni volta che vengono prese in esame tuple distinte della mia relazione esami e le tuple coincidono sul valore della matricola, impongo al DBMS che anche tutti gli attributi caratterizzanti lo studente coincidano

Dipendenza funzionale (definizione)

Data una relazione *r(A)*, un sottinsieme *X* di attributi di *A* ($X \subseteq A$), un altro sottinsieme *Y* di attributi di *A* ($Y \subseteq A$) il **vincolo di dipendenza funzionale** $X \rightarrow Y$ (*X determina Y*) è soddisfatto se e solo se

$$\forall t_1, t_2 \in r\ (t_1[X]=t_2[X] \Rightarrow t_1[Y]=t_2[Y])$$

Esempi di vincoli

ESAMI(<u>MATR</u>, NS, IS, CAP, CF, DN, <u>Co</u>, Vo, DE, CP, NP, Q, TU)

Dipendenze funzionali:

- MATR → NS, IS, CAP, CF, DN
- CF → MATR
- IS → CAP (nell'ipotesi di indirizzo completo)
- MATR, Co → Vo, DE, CP, NP
- CP → NP, Q
- Q → TU

Immaginiamo un vincolo per cui ogni docente è titolare di un solo corso

- CP → Co (dipendenza funzionale che useremo in seguito)

Progettazione e dipendenze funzionali

- Un buon lavoro di progettazione richiede un'analisi attenta della realtà

- Bisogna collezionare, accanto ad ogni relazione, tutte le dipendenze funzionali suggerite dal sistema informativo

- Per ogni relazione *r* abbiamo un **insieme *F*** di dipendenze funzionali

Insieme di dipendenze funzionali

Il nostro insieme F conterrà:

{ MATR → NS, IS, CAP, CF, DN;

CF → MATR;

IS → CAP;

MATR, Co → Vo, DE, CP, NP;

CP → NP, Q;

Q → TU;

CP → Co }

Formulazioni alternative

Ci sono tantissime formulazioni alternative anche equivalenti delle dipendenze funzionali

Un progettista vede:

- MATR → NS, IS, CAP, CF, DN
- CF → MATR
- IS → CAP

Un altro progettista vede:

- MATR → NS, IS
- MATR → CF, DN
- CF → MATR, NS, DN
- IS → CAP

Formulazioni alternative

I due progettisti hanno letto la realtà in modo diverso?

Sì, i due insiemi sono sintatticamente diversi

I due progettisti hanno letto la realtà in modo equivalente?

Bisogna applicare la definizione formale di vincolo di dipendenza funzionale

Equivalenza delle dipendenze funzionali

Primo insieme F'

f'_1: MATR $\rightarrow$ NS, IS, CAP, CF, DN

Secondo insieme F'' (reintroduco il CAP per semplicità)

f''_1: MATR $\rightarrow$ NS, IS, CAP

f''_2: MATR $\rightarrow$ CF, DN

Le due formulazioni sono equivalenti se, quando la base di dati rispetta F'', allora rispetta anche F' e viceversa

Verifica dell'equivalenza

Ritorniamo alla definizione di vincolo

$f''_1 \wedge f''_2$:

$\forall t_1,t_2 \in r\ (t_1[MATR]=t_2[MATR] \Rightarrow t_1[\{NS,IS,CAP\}]=t_2[\{NS,IS,CAP\}])$
$\wedge\ (t_1[MATR]=t_2[MATR] \Rightarrow t_1[\{CF,DN\}]=t_2[\{CF,DN\}])$

quindi

$\forall t_1,t_2 \in r\ (t_1[MATR]=t_2[MATR] \Rightarrow$
$t_1[\{NS,IS,CAP,CF,DN\}]=t_2[\{NS,IS,CAP, CF,DN\}])$

che è la definizione di f'_1:MATR → NS, IS, CAP, CF, DN

Verifica dell'equivalenza

Ma vale anche il viceversa

$f'_1: \forall t_1, t_2 \in r\ (t_1[MATR]=t_2[MATR] \Rightarrow$

$t_1[\{NS,IS,CAP,CF,DN\}]=t_2[\{NS,IS,CAP, CF,DN\}])$

quindi, a maggior ragione

$\forall t_1, t_2 \in r\ (t_1[MATR]=t_2[MATR] \Rightarrow t_1[\{NS,IS,CAP\}]=t_2[\{NS,IS,CAP\}])$

che è la definizione di f''_1: MATR → NS, IS, CAP

Verifica dell'equivalenza

Ma vale anche il viceversa

$f'_1: \forall t_1,t_2 \in r\ (t_1[MATR]=t_2[MATR] \Rightarrow$

$t_1[\{NS,IS,CAP,CF,DN\}]=t_2[\{NS,IS,CAP, CF,DN\}])$

quindi, a maggior ragione

$\forall t_1,t_2 \in r\ (t_1[MATR]=t_2[MATR] \Rightarrow t_1[\{CF,DN\}]=t_2[\{CF,DN\}])$

che è la definizione di f''_2: MATR → CF, DN

Quindi il rispetto di f'_1 implica il rispetto di f''_1 e f''_2

Significato di equivalenza

Due basi di dati, una progettata con i vincoli F', un'altra progettata con i vincoli F'', fanno evolvere lo stato della base dati esattamente nello stesso modo

Altro esempio

Insieme di vincoli F'

f'_1: MATR → NS, IS, CF, DN

f'_2: IS → CAP

Insieme di vincoli F''

f''_1: MATR → NS, IS, CAP, CF, DN

f''_2: IS → CAP

- Sono due formulazioni equivalenti?

Verifica

$f'_1: \forall t_1, t_2 \in r\ (t_1[\mathbf{MATR}]=t_2[\mathbf{MATR}] \Rightarrow t_1[\{NS,\mathbf{IS},CF,DN\}]=t_2[\{NS,\mathbf{IS},CF,DN\}])$

$f'_2: \forall t_1, t_2 \in r\ (t_1[\mathbf{IS}]=t_2[\mathbf{IS}] \Rightarrow t_1[CAP]=t_2[CAP])$

Quindi

$\forall t_1, t_2 \in r\ (t_1[\mathbf{MATR}]=t_2[\mathbf{MATR}] \Rightarrow t_1[CAP]=t_2[CAP])$

Quindi

$$f''_1: \forall t_1, t_2 \in r\ (t_1[MATR]=t_2[MATR] \Rightarrow t_1[\{NS,IS,CAP,CF,DN\}]=t_2[\{NS,IS,CAP,CF,DN\}])$$

Verifica (viceversa)

$f''_1: \forall t_1, t_2 \in r\ (t_1[MATR]=t_2[MATR] \Rightarrow$
$t_1[\{NS,IS,CAP,CF,DN\}]=t_2[\{NS,IS,CAP,CF,DN\}])$

Quindi, *a fortiori*

$f'_1: \forall t_1, t_2 \in r\ (t_1[MATR]=t_2[MATR] \Rightarrow$
$t_1[\{NS,IS,CF,DN\}]=t_2[\{NS,IS,CF,DN\}])$

e

$f'_2: \forall t_1, t_2 \in r\ (t_1[IS]=t_2[IS] \Rightarrow t_1[CAP]=t_2[CAP])$

Teoria di Armstrong

- Utilizzare la sola definizione di vincolo di dipendenza funzionale per verificare le equivalenze può risultare molto pesante
- Conviene quindi usare la Teoria di Armstrong delle dipendenze funzionali
- Armstrong è riuscito a costruire una teoria assiomatica con la quale caratterizza la dipendenza funzionale, ovvero l'oggetto "$\rightarrow$"

Assiomi della Teoria di Armstrong

- Assioma di riflessività

$$\textit{se } Y \subseteq X, \textit{ allora } X \rightarrow Y$$

- Assioma di unione

$$\textit{se } X \rightarrow Y \textit{ e } X \rightarrow Z, \textit{ allora } X \rightarrow YZ$$

$$\textit{dove } YZ = Y \cup Z$$

- Assioma di transitività

$$\textit{se } X \rightarrow Y \textit{ e } Y \rightarrow Z, \textit{ allora } X \rightarrow Z$$

Vincolo dipendenza funzionale

Il vincolo della dipendenza funzionale è un modello (in senso matematico) della Teoria di Armstrong

Un modello calza su una teoria (*fitta* la teoria) quando gli assiomi della teoria sono rispettati dal modello

Teoria e modello:

- La teoria di Armstrong è la teoria della "$\rightarrow$" (assiomi di riflessività, unione, transitività)
- Il modello sono i vincoli di dipendenza funzionale, cioè $\forall t_1, t_2 \in r \ (t_1[X]=t_2[X] \Rightarrow t_1[Y]=t_2[Y])$

Assioma della riflessività

Assioma di riflessività

$$\textit{se } Y \subseteq X, \textit{ allora } X \rightarrow Y$$

Considero un insieme qualsiasi di attributi (esempio: X={CF,Co,Q}) e ne prendo un sottinsieme (esempio: Y={Co,Q}) sicuramente c'è una dipendenza funzionale {CF,Co,Q} → {Co,Q}

Ma cosa significa nel modello?

Assioma della riflessività

Nel modello dei vincoli la proprietà diventa:

$\forall t_1, t_2 \in r\ (t_1[\{CF,Co,Q\}]=t_2[\{CF,Co,Q\}] \Rightarrow t_1[\{Co,Q\}]=t_2[\{Co,Q\}])$

Questo assioma quindi introduce un vincolo che non ha nessuna utilità nel modello

Assioma dell'unione

Assioma di unione

se $X \rightarrow Y$ *e* $X \rightarrow Z$*, allora* $X \rightarrow YZ$

Vedere esempio fatto in precedenza

Assioma di transitività

Assioma di transitività

se X → Y e Y → Z, allora X → Z

Vedere esempio fatto in precedenza

Conseguenza

La Teoria di Armstrong, dal punto di vista del modello dei vincoli delle dipendenze funzionali è una teoria corretta: posso quindi applicare i teoremi della teoria di Armstrong al modello dei vincoli di dipendenza funzionale

Teorema dell'espansione

Data una dipendenza funzionale $X \rightarrow Y$ e un insieme di attributi W, allora $WX \rightarrow WY$

Esempio: considero la dipendenza funzionale $Q \rightarrow TU$ e l'attributo CF, quindi $CF,Q \rightarrow CF,TU$

Esercizio: applicare la definizione di vincolo per verificare la validità dell'esempio

Teorema dell'espansione

Ipotesi: data X → Y, dato W

Tesi: WX → WY

Dimostrazione (con la teoria di Armstrong)

1. Partiamo da *WX*
2. Scelgo un sottoinsieme: *W*
3. Per l'assioma di **riflessività**: *WX* → *W*
4. Per l'assioma di **riflessività**: *WX* → *X*
5. Sappiamo che *X* → *Y* (**ipotesi**)
6. Allora *WX* → *Y* per **transitività**
7. Ma se *WX* → *Y* e *WX* → *W*, per l'assioma **dell'unione**: *WX* → *WY*

[CVD]

Teorema di decomposizione

Data una dipendenza funzionale $X \rightarrow YZ$,

allora $X \rightarrow Y$ e $X \rightarrow Z$

Si veda l'esempio fatto in precedenza

Teorema di decomposizione

Ipotesi: dato $X \rightarrow YZ$

Tesi: $X \rightarrow Y$ e $X \rightarrow Z$

Dimostrazione

1. Partiamo da $X \rightarrow YZ$
2. Per l'assioma di **riflessività**: $YZ \rightarrow Y$
3. Per l'assioma di **transitività**: $X \rightarrow Y$
4. Per l'assioma di **riflessività**: $YZ \rightarrow Z$
5. Per l'assioma di **transitività**: $X \rightarrow Z$
6. Quindi, se $X \rightarrow YZ$, allora $X \rightarrow Y$ e $X \rightarrow Z$

[CVD]

Teorema di pseudo-transitività

Date $X \rightarrow Y$, $WY \rightarrow Z$, allora $WX \rightarrow Z$

Dimostrazione

1. Partiamo da $X \rightarrow Y$
2. Per il teorema di **espansione** $WX \rightarrow WY$
3. Per l'assioma di **transitività**, $WX \rightarrow WY$ e $WY \rightarrow Z$, allora $WX \rightarrow Z$

[CVD]

Teorema del prodotto

Date le dipendenze funzionali $X \rightarrow Y$ e $W \rightarrow Z$, allora vale $XW \rightarrow YZ$

Dimostrazione

1. Partiamo da XW
2. Per l'assioma di **riflessività** $XW \rightarrow X$
3. Ma è data $X \rightarrow Y$, quindi per l'assioma di **transitività** $XW \rightarrow Y$
4. Per l'assioma di **riflessività** vale anche $XW \rightarrow W$
5. Ma è data $W \rightarrow Z$, quindi per l'assioma di **transitività** $XW \rightarrow Z$
6. Per l'assioma **dell'unione** abbiamo quindi $XW \rightarrow YZ$

[CVD]

Esercizio

Porre come assioma la proprietà dell'espansione e dimostrare l'unione

Attributi estranei

Immaginiamo che un progettista, anziché vedere la dipendenza funzionale seguente:

- MATR, Co → Vo, DE, CP

veda le dipendenze funzionali seguenti:

- MATR,NS,Co → Vo,DE,CP
- MATR → NS

Le altre dipendenze rimangono tali e quali

Attributi estranei

- MATR,NS,Co → Vo,DE,CP
- MATR → NS

Si capisce che si può semplificare il primo vincolo eliminando NS, visto che è determinato da MATR (secondo vincolo)

L'attributo NS viene detto **attributo estraneo**

Attributi estranei

- MATR,NS,Co $\rightarrow$ Vo,DE,CP
- MATR $\rightarrow$ NS

L'attributo NS viene detto **attributo estraneo**

Si può dimostrare formalmente utilizzando gli assiomi e teoremi di Armstrong che da queste dipendenze si può estrarre la dipendenza:

MATR, Co $\rightarrow$ Vo,DE,CP

Attributi estranei

Dimostrazione

- Partiamo dall'insieme di attributi *MATR,Co*
- Per riflessività: *MATR,Co* → *MATR*
- Considero ora *MATR* → *NS*
- Per transitività posso concludere che *MATR,Co* → *NS*
- Per il teorema dell'espansione, con *W=MATR,Co*(*):
  MATR,Co → *MATR,Co,NS* (è un'unione di attributi)
- Ma *MATR,Co,NS* → *Vo,DE,CP*
- Per transitività: *MATR,Co* → *Vo,DE,CP*

[CVD]

* W può essere un insieme qualsiasi

Attributi estranei

Posso quindi sostituire il vincolo di dipendenza

$$MATR,NS,Co \rightarrow Vo,DE,CP$$

con

$$MATR, Co \rightarrow Vo, DE, CP \ ?$$

Devo prima dimostrare che da MATR,Co $\rightarrow$ Vo,DE,CP discende MATR,NS,Co $\rightarrow$ Vo,DE,CP

Attributi estranei

Date *MATR,Co* → *Vo,DE,CP* e *MATR* → *NS*, allora *MATR,NS,Co* → *Vo,DE,CP*

Dimostrazione

1. Per riflessività: MATR,NS,Co → MATR,Co
2. Ma so che *MATR,Co* → *Vo,DE,CP*, quindi per transitività: *MATR,NS,Co* → *Vo,DE,CP*

I due sistemi di vincoli fanno evolvere lo stato della base di dati nello stesso modo

Attributo estraneo: formalizzazione

Abbiamo le seguenti dipendenze funzionali F

- $ABCD \rightarrow E$
- $B \rightarrow C$ (data o dedotta dalle dipendenze date, ad esempio per transitività)

allora C è un attributo estraneo e si può cancellare, la dipendenza è allora ABD → E

Attributo estraneo: dimostrazione

1. Partiamo da *ABD* (antecedente della dipendenza funzionale in cui *C* è stato eliminato)
2. Per riflessività: $ABD \rightarrow B$
3. $B \rightarrow C$ è una dipendenza funzionale valida o derivata, quindi, per transitività $ABD \rightarrow C$
4. Per il teorema di espansione: $ABD \rightarrow ABCD$
5. Ma $ABCD \rightarrow E$, quindi per transitività $ABD \rightarrow E$

[CVD]

Attributo estraneo: dimostrazione

Dato il sistema di dipendenze F':

- $ABD \rightarrow E$
- $B \rightarrow C$ (data o derivata)

posso dedurre il sistema F (con l'attributo estraneo)

Dimostrazione

1. Per riflessività: $ABCD \rightarrow ABD$
2. Ho $ABD \rightarrow E$, quindi, per transitività: $ABCD \rightarrow E$

[CVD]

Attributo estraneo

- Posso quindi sostituire il sistema di dipendenze F con il sistema F', essendo questi perfettamente equivalenti

- Tra due sistemi equivalenti, preferisco quello più semplice (senza attributi estranei)

Chiusura di un insieme F

La **chiusura** di un insieme di dipendenze funzionali F è un insieme di dipendenze funzionali F^+ tali che ogni dipendenza funzionale f^+ dell'insieme F^+ sia derivabile da F

(l'insieme F^+ è finito)

Equivalenza

La chiusura ci dà lo strumento formale per caratterizzare dal punto di vista teorico il concetto di equivalenza

Immaginiamo di avere un progettista P1 che individua le dipendenze F e un altro progettista P2 che individua le dipendenze G, dove G è sintatticamente diverso da F

P1 e P2 stanno dicendo la stessa cosa? Ovvero, F e G sono equivalenti?

Definizione di equivalenza

$F \equiv G$ (F è equivalente a G) se e solo se $F^+ = G^+$

Abbiamo ricondotto l'equivalenza ad un'uguaglianza insiemistica: se tutte le dipendenze derivate da *F* sono uguali a tutte le dipendenze derivate da *G*, le due basi di dati evolvono allo stesso modo

Equivalenza

- Risulta però complesso costruire F^+ e G^+ per verificare l'equivalenza

- Fortunatamente, esiste una proprietà che porta al medesimo risultato

Definizione di equivalenza

$F \equiv G$ (F è equivalente a G) se e solo se $F \vdash G$ *(G* è deducibile da *F)* *e* $G \vdash F$ *(F* è deducibile da *G)*

- $F \vdash G$*:* presa una qualsiasi dipendenza g di G, g è deducibile da F
- $G \vdash F$*:* presa una qualsiasi dipendenza f di F, f è deducibile da G

Equivalenza delle definizioni

Abbiamo quindi due definizioni di equivalenza

$F \equiv G$ (F è equivalente a G) se e solo se $F^+ = G^+$

e

$F \equiv G$ (F è equivalente a G) se e solo se $F \vdash G$ *(G* è deducibile da *F) e* $G \vdash F$ *(F* è deducibile da *G)*

Dobbiamo dimostrare che se $F \vdash G$ *(G* è deducibile da *F) e* $G \vdash F$ *(F* è deducibile da *G),* allora $F^+ = G^+$

Definizioni equivalenti

Ricordiamo che l'uguaglianza insiemistica $F^+ = G^+$ significa che $F^+ \subseteq G^+$ *e* $G^+ \subseteq F^+$

Dimostrazione di $G^+ \subseteq F^+$

1. Consideriamo una dipendenza $g^+ \in G^+$
2. g^+ deducibile da G, quindi $G \vdash g^+$
3. Per le proprietà di deduzione (transitive), se $F \vdash G$ *e* $G \vdash g^+$, allora $F \vdash g^+$, cioè $g^+ \in F^+$
4. Abbiamo quindi dimostrato che, preso un elemento qualsiasi $g^+ \in G^+$, vale anche $g^+ \in F^+$, quindi $G^+ \subseteq F^+$

Definizioni equivalenti

Dimostrazione di $F^+ \subseteq G^+$

1. Consideriamo una dipendenza $f^+ \in F^+$
2. f^+ deducibile da F, quindi $F \vdash f^+$
3. Per le proprietà di deduzione (transitive), se $G \vdash F$ *e* $F \vdash f^+$, allora $G \vdash f^+$, cioè $f^+ \in G^+$
4. Abbiamo quindi dimostrato che, preso un elemento qualsiasi $f^+ \in F^+$, vale anche $f^+ \in G^+$, quindi $F^+ \subseteq G^+$

Valgono quindi sia $F^+ \subseteq G^+$ che $G^+ \subseteq F^+$, ovvero $F^+ = G^+$

[CVD]

Esempio

F (Progettista 1)	G (Progettista 2)
MATR → NS, IS, CAP, CF, DN CF → MATR IS → CAP MATR, Co → Vo, DE, CP, NP CP → NP, Q Q → TU CP → Co	MATR → NS,IS MATR → DN,CF IS → CAP MATR,Co → Vo,DE MATR,Co,NS → CP CP → NP,Q Q → TU CP → Co

Da *F* è immediatamente deducibile *G*, da *G* è immediatamente deducibile *F* (basta applicare gli assiomi e teoremi di Armstrong)

Quindi, posso verificare l'equivalenza di *F* e *G* senza chiamare in causa F^+ e G^+

Chiusura di un insieme di attributi

Dato un insieme di attributi A, su cui è definito l'insieme di dipendenze funzionali F, dato un sottoinsieme $X \subseteq A$, la chiusura X^+ di X è definita come

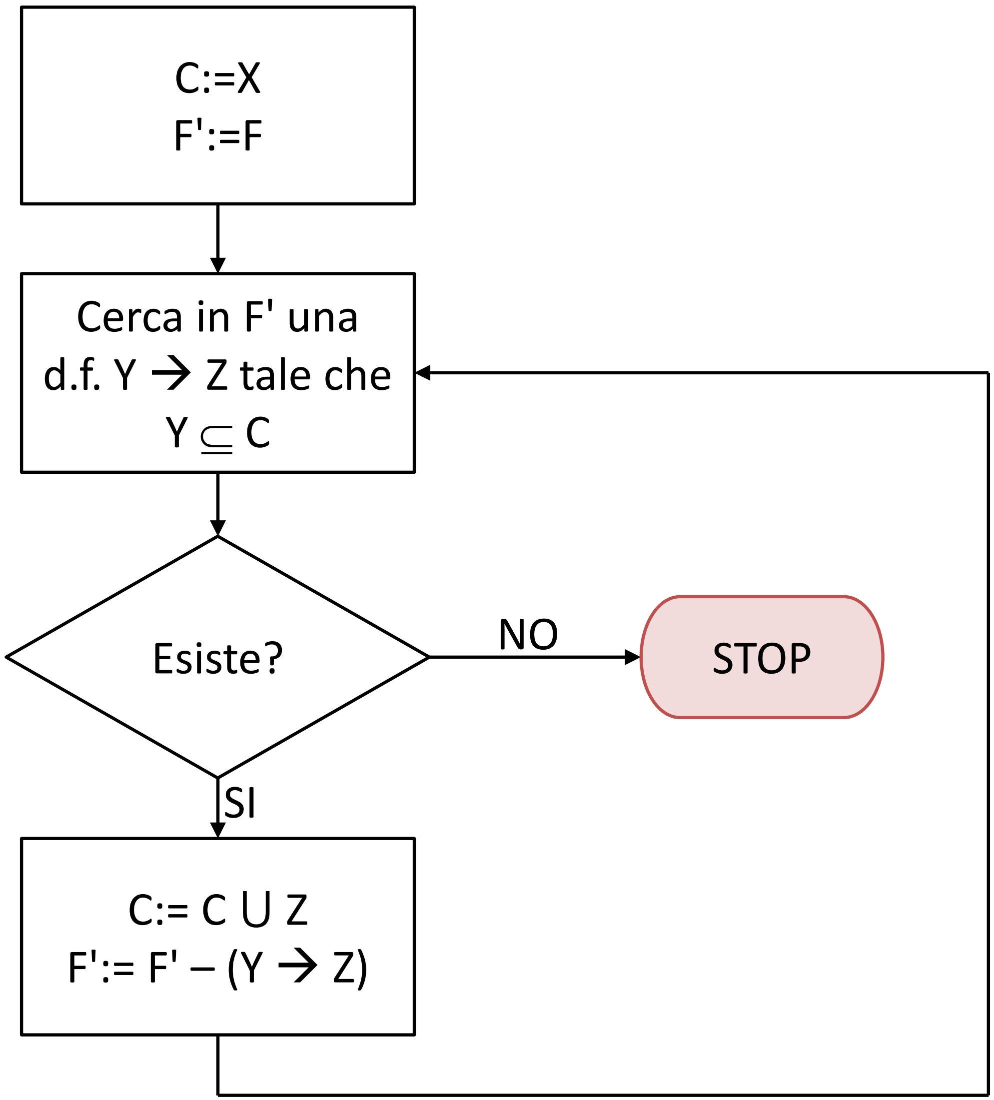
$$X^+ = \{A_i \mid F \vdash X \rightarrow A_i\}$$

Esempio

- Considero $X=\{MATR\}$ e ne calcolo la chiusura seguendo la definizione
- Guardo le dipendenze F
- Nella chiusura di *MATR* c'è sicuramente *MATR* per riflessività (quindi $MATR \in X^+$)
- Da $MATR \rightarrow NS, IS, CAP, CF, DN$, deduco che $\{NS, IS, CAP, CF, DN\} \subseteq X^+$
- La chiusura di $X=\{MATR\}$ è quindi

$$X^+ = \{MATR, NS, IS, CAP, CF, DN\}$$

Algoritmo per il calcolo di X^+

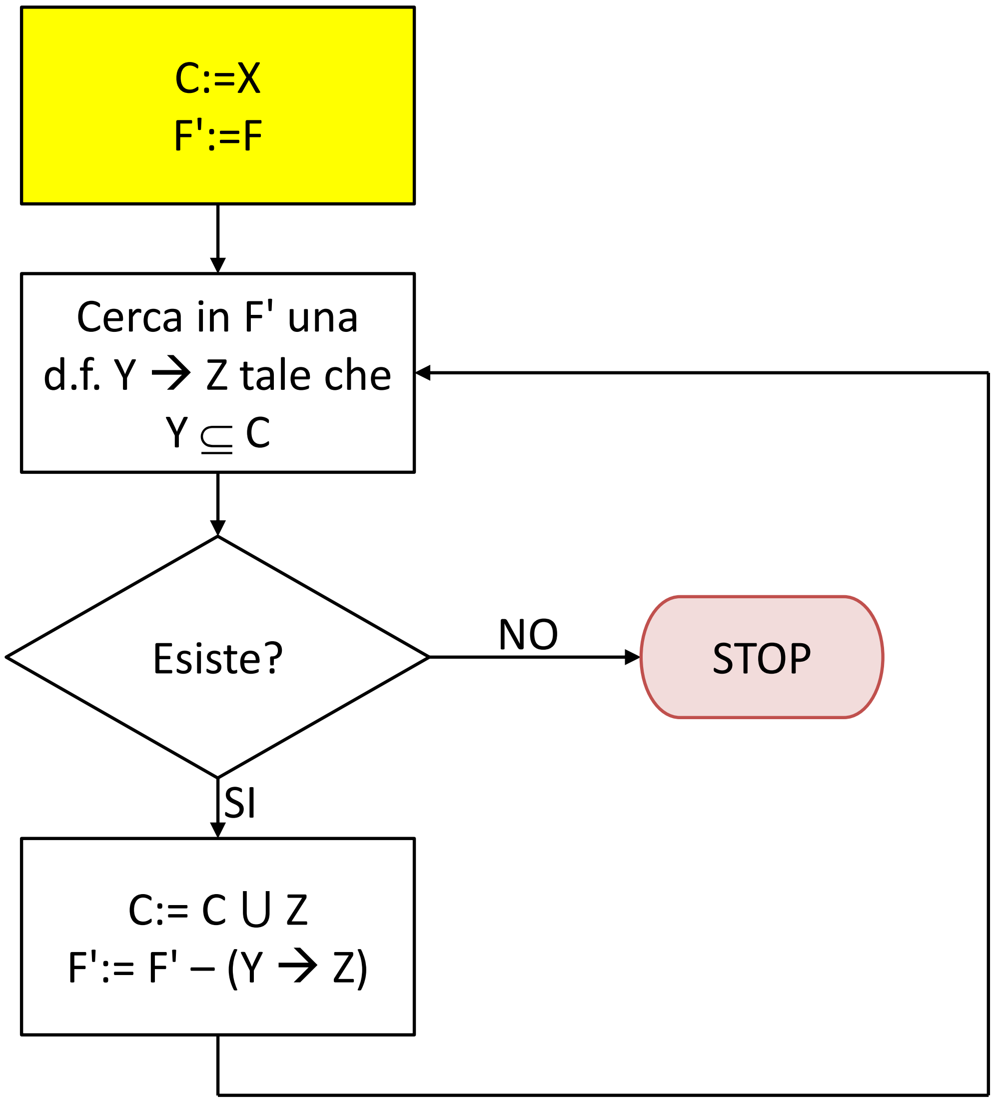


Esempio

- Parto con X = {CP}

F
MATR → NS, IS, CAP, CF, DN
CF → MATR
IS → CAP
MATR, Co → Vo, DE, CP, NP
CP → NP, Q
Q → TU
CP → Co

Esempio

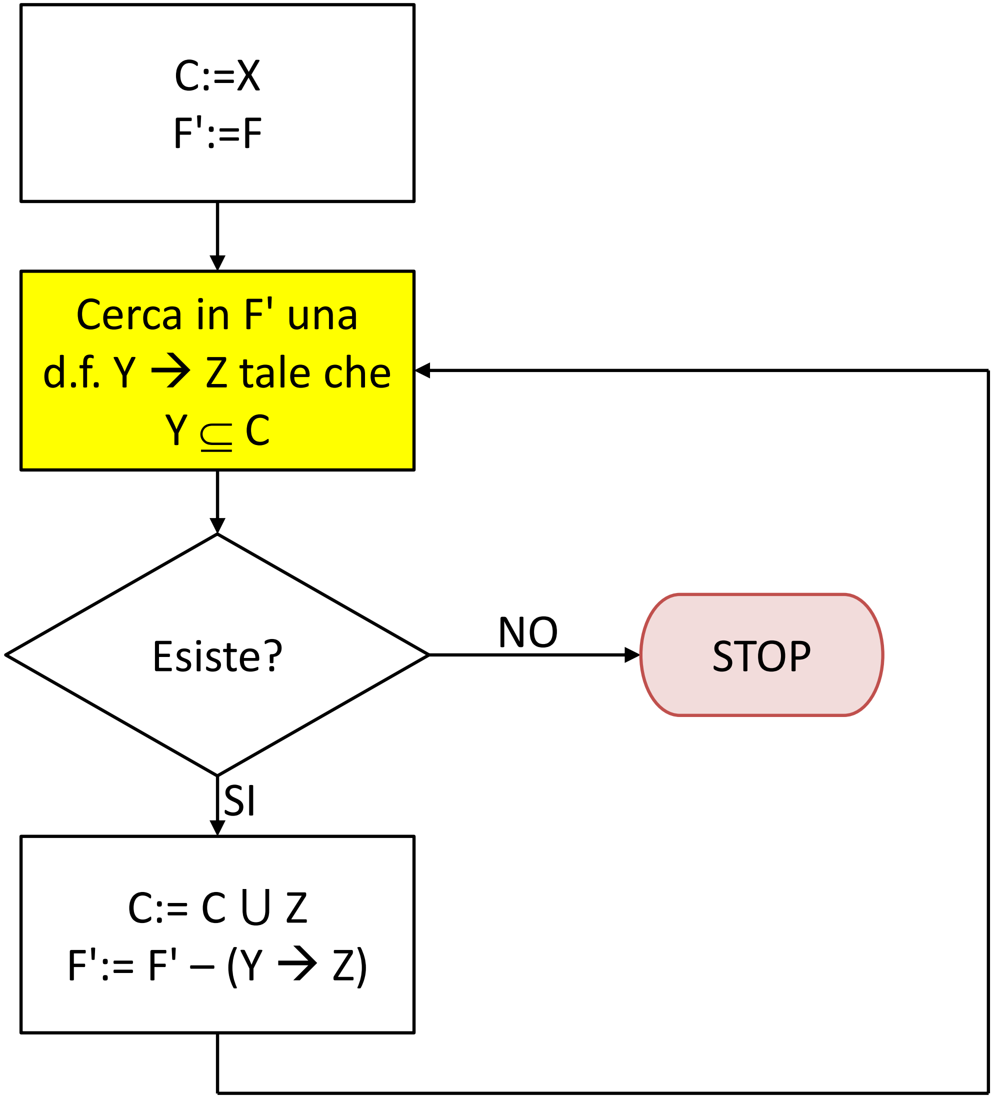

C:=X
F':=F
Cerca in F' una
d.f. Y → Z tale che
Y ⊆ C
Esiste?
NO
STOP
SI
C:= C U Z
F':= F' – (Y → Z)

Esempio

- Parto con X = {CP}
- C = {CP} e F' = F

F'
MATR → NS, IS, CAP, CF, DN
CF → MATR
IS → CAP
MATR, Co → Vo, DE, CP, NP
CP → NP, Q
Q → TU
CP → Co

Esempio



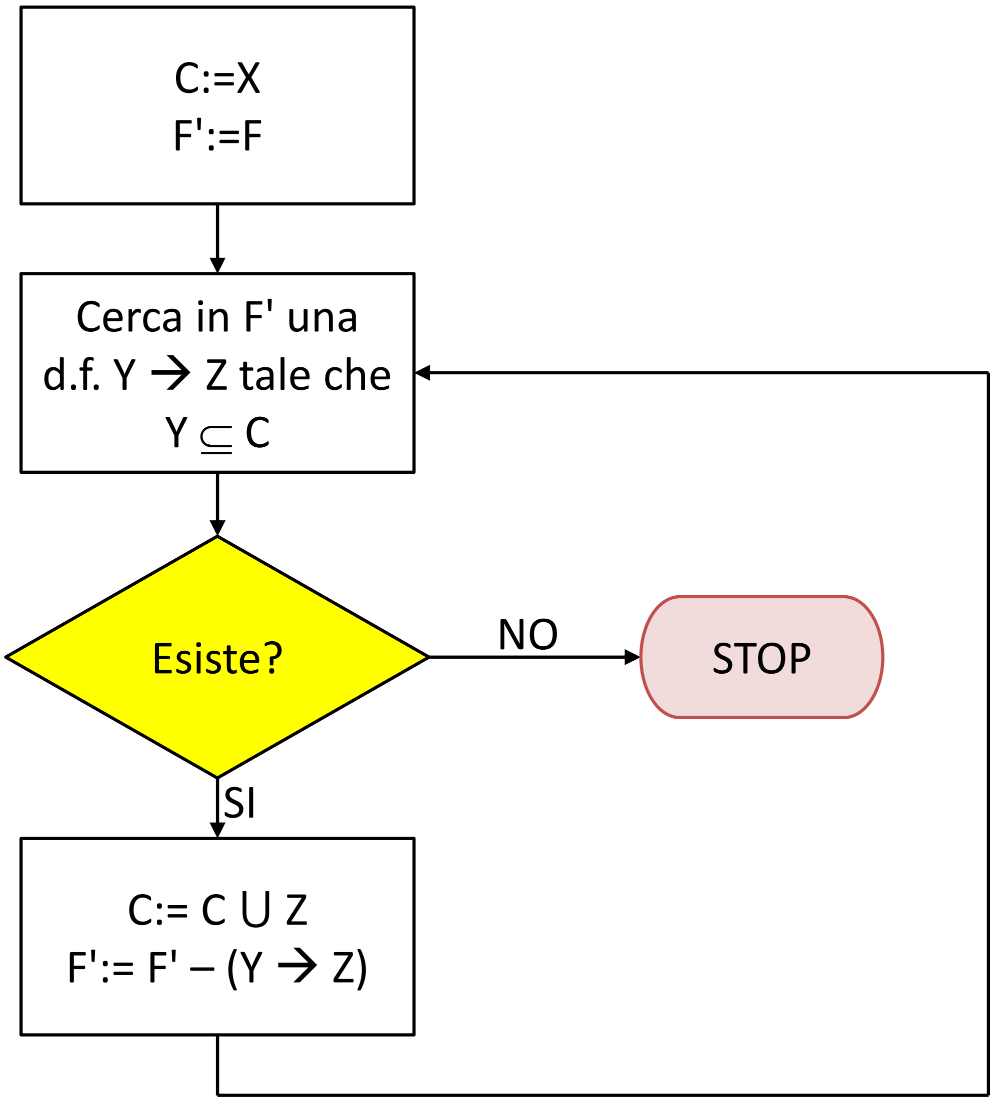
C:=X
F':=F
Cerca in F' una
d.f. Y → Z tale che
Y ⊆ C
Esiste?
NO
STOP
SI
C:= C U Z
F':= F' – (Y → Z)

Esempio

- Parto con X = {CP}
- C = {CP} e F' = F
- Cerco una dipendenza funzionale in F' con antecedente X

F'
MATR → NS, IS, CAP, CF, DN
CF → MATR
IS → CAP
MATR, Co → Vo, DE, CP, NP
CP → NP, Q
Q → TU
CP → Co

Esempio

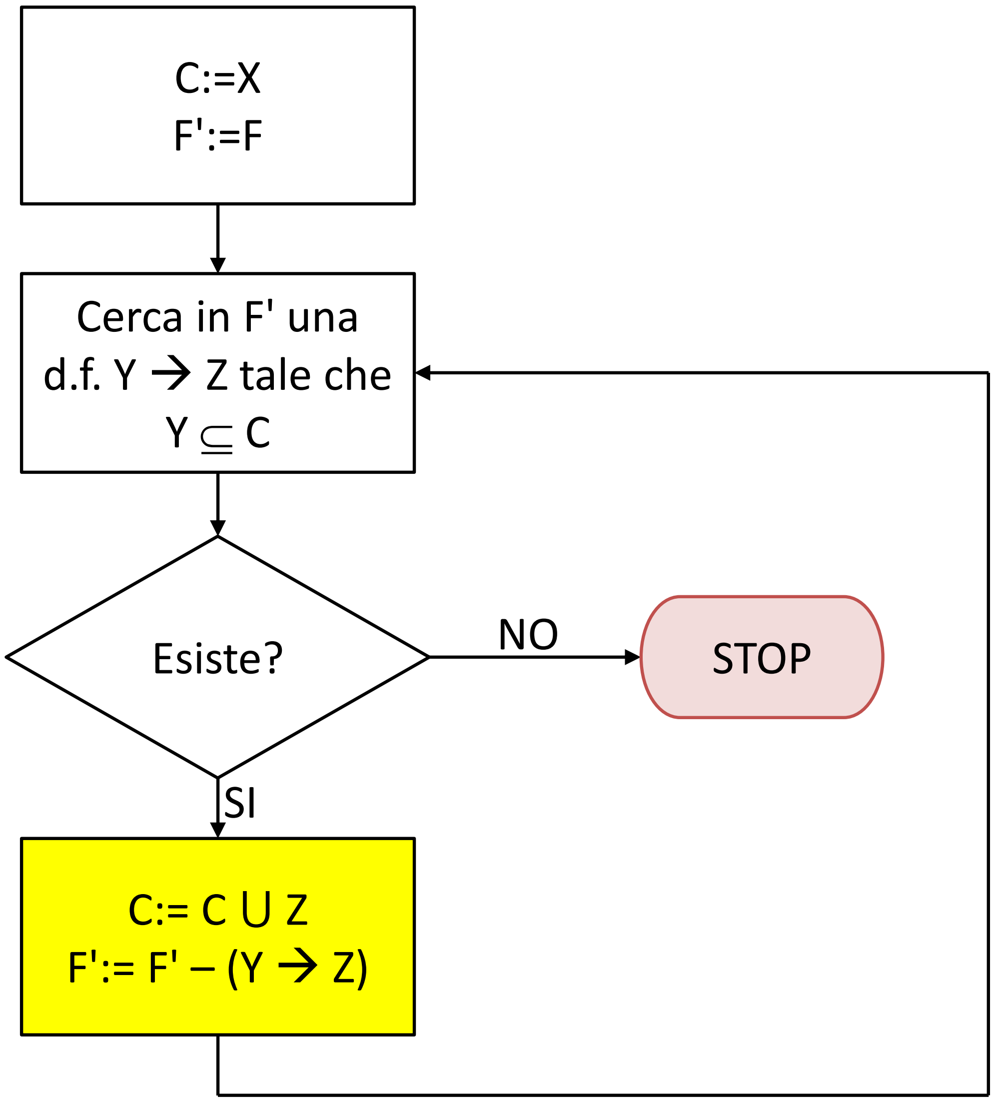


Esempio

- Parto con X = {CP}
- C = {CP} e F' = F
- Cerco una dipendenza funzionale in F' con antecedente X
- Esiste? Sì: CP → NP, Q

F'
MATR → NS, IS, CAP, CF, DN
CF → MATR
IS → CAP
MATR, Co → Vo, DE, CP, NP
* CP → NP, Q
Q → TU
CP → Co

Esempio

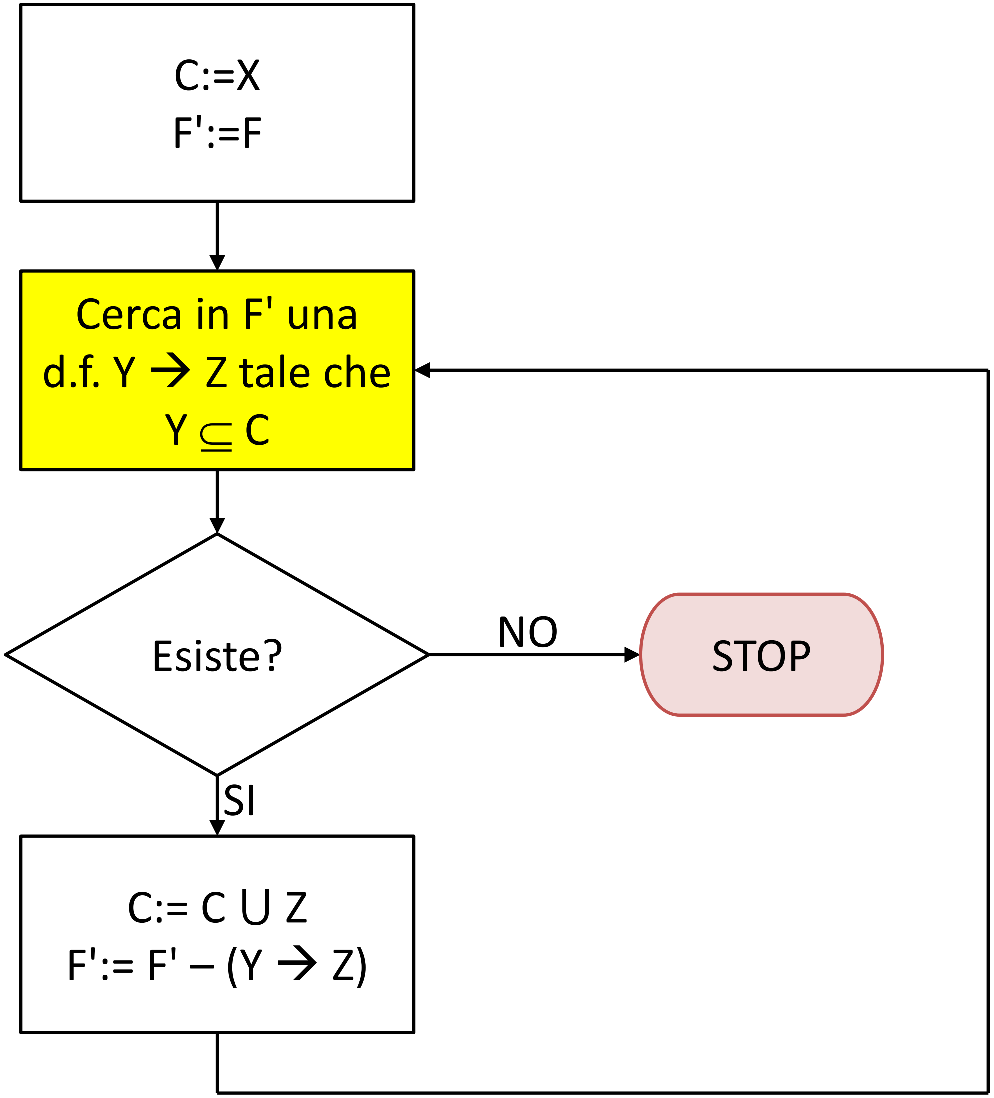


Esempio

- Parto con X = {CP}
- C = {CP} e F' = F
- Cerco una dipendenza funzionale in F' con antecedente X
- Esiste? Sì: CP → NP, Q
- C = {CP} ∪ {NP,Q} = {CP,NP,Q}

F'
MATR → NS, IS, CAP, CF, DN
CF → MATR
IS → CAP
MATR, Co → Vo, DE, CP, NP
~~CP → NP, Q~~
Q → TU
CP → Co

Esempio

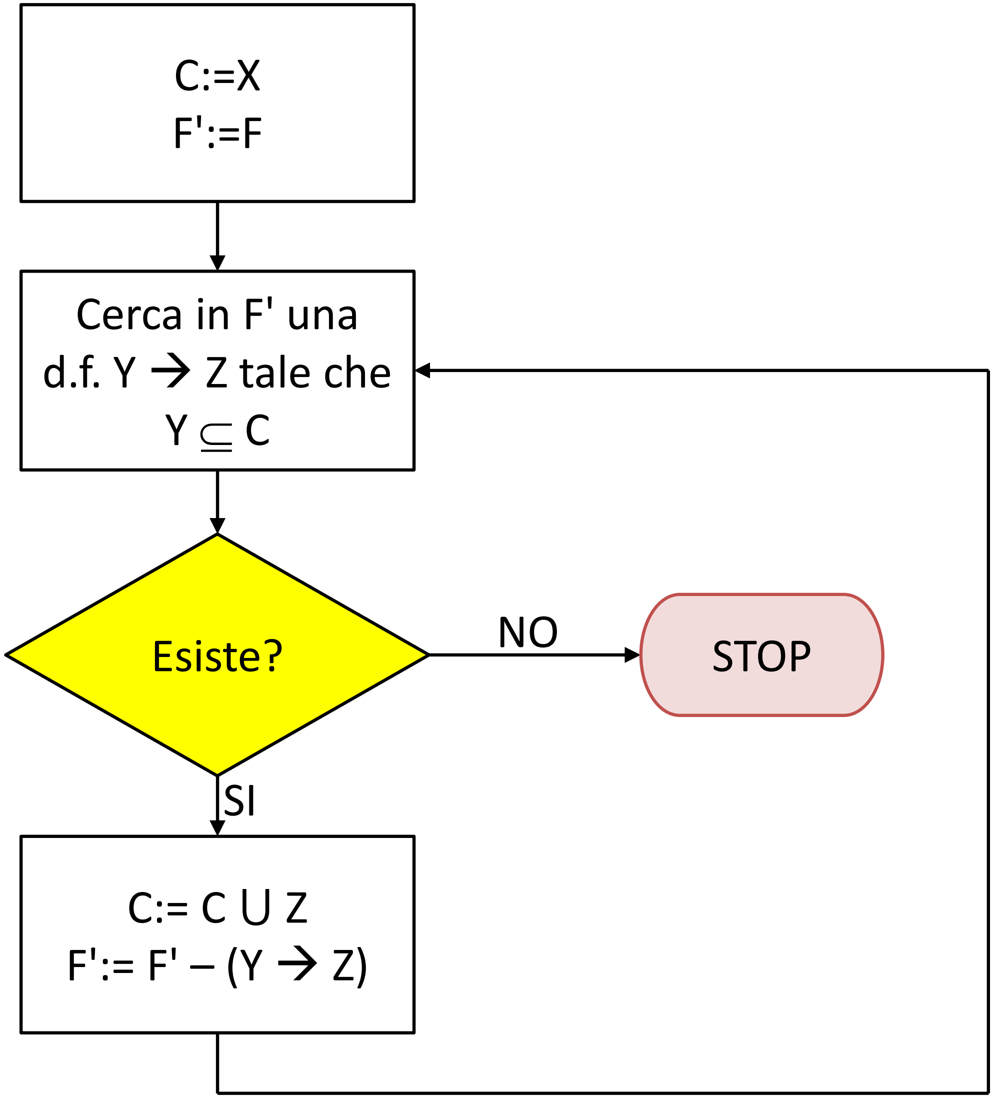


Esempio

- Parto con X = {CP}
- C = {CP} e F' = F
- Cerco una dipendenza funzionale in F' con antecedente in {CP}
- Esiste? Sì: CP → NP, Q
- C = {CP} ∪ {NP,Q} = {CP,NP,Q}
- Cerco una d.f. con antecedente in {CP,NP,Q}

F'
MATR → NS, IS, CAP, CF, DN
CF → MATR
IS → CAP
MATR, Co → Vo, DE, CP, NP
~~CP → NP, Q~~
Q → TU
CP → Co

Esempio

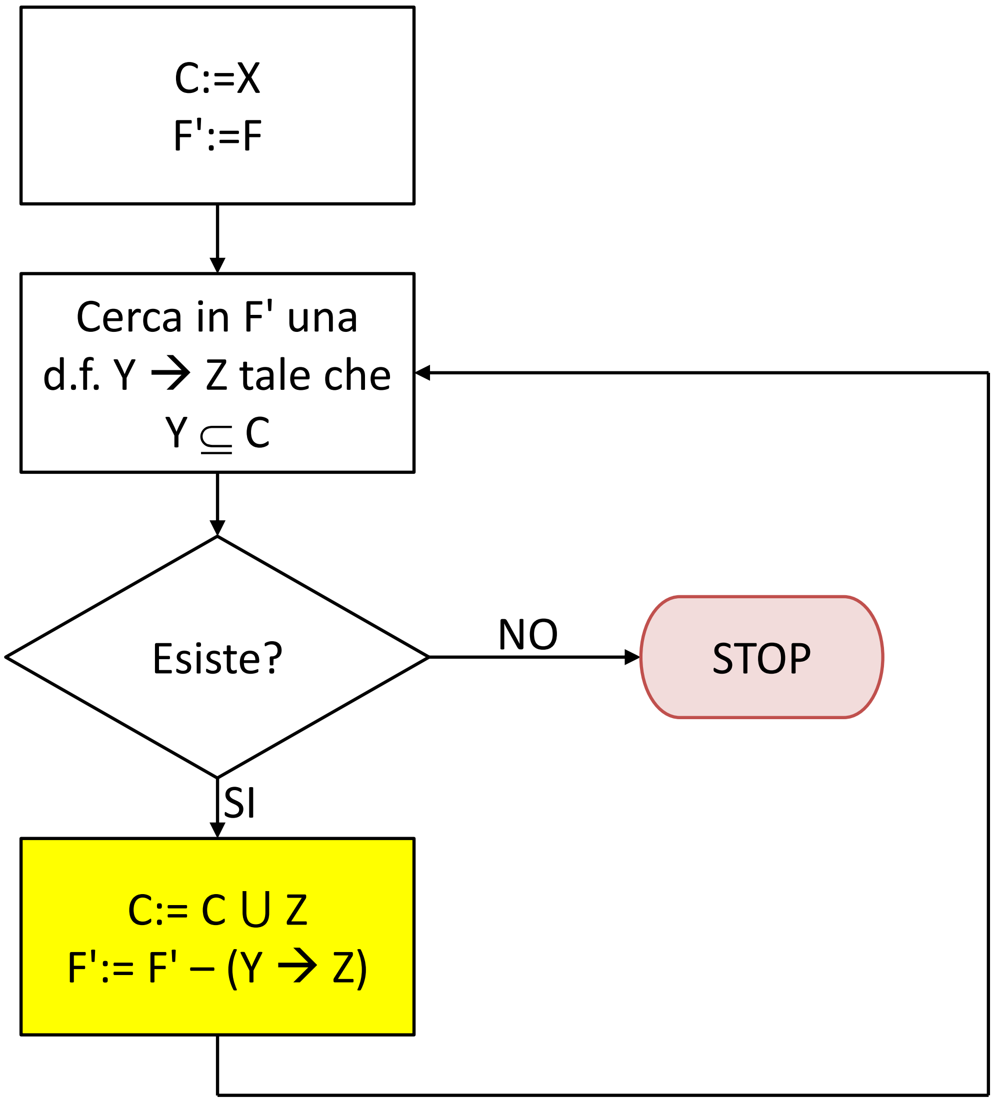


Esempio

- Parto con X = {CP}
- C = {CP} e F' = F
- Cerco una dipendenza funzionale in F' con antecedente in {CP}
- Esiste? Sì: CP → NP, Q
- C = {CP} ∪ {NP,Q} = {CP,NP,Q}
- Cerco una d.f. con antecedente in {CP,NP,Q}
- Esiste? Sì: Q → TU

F'
MATR → NS, IS, CAP, CF, DN
CF → MATR
IS → CAP
MATR, Co → Vo, DE, CP, NP
~~CP → NP, Q~~
*Q → TU
CP → Co

Esempio

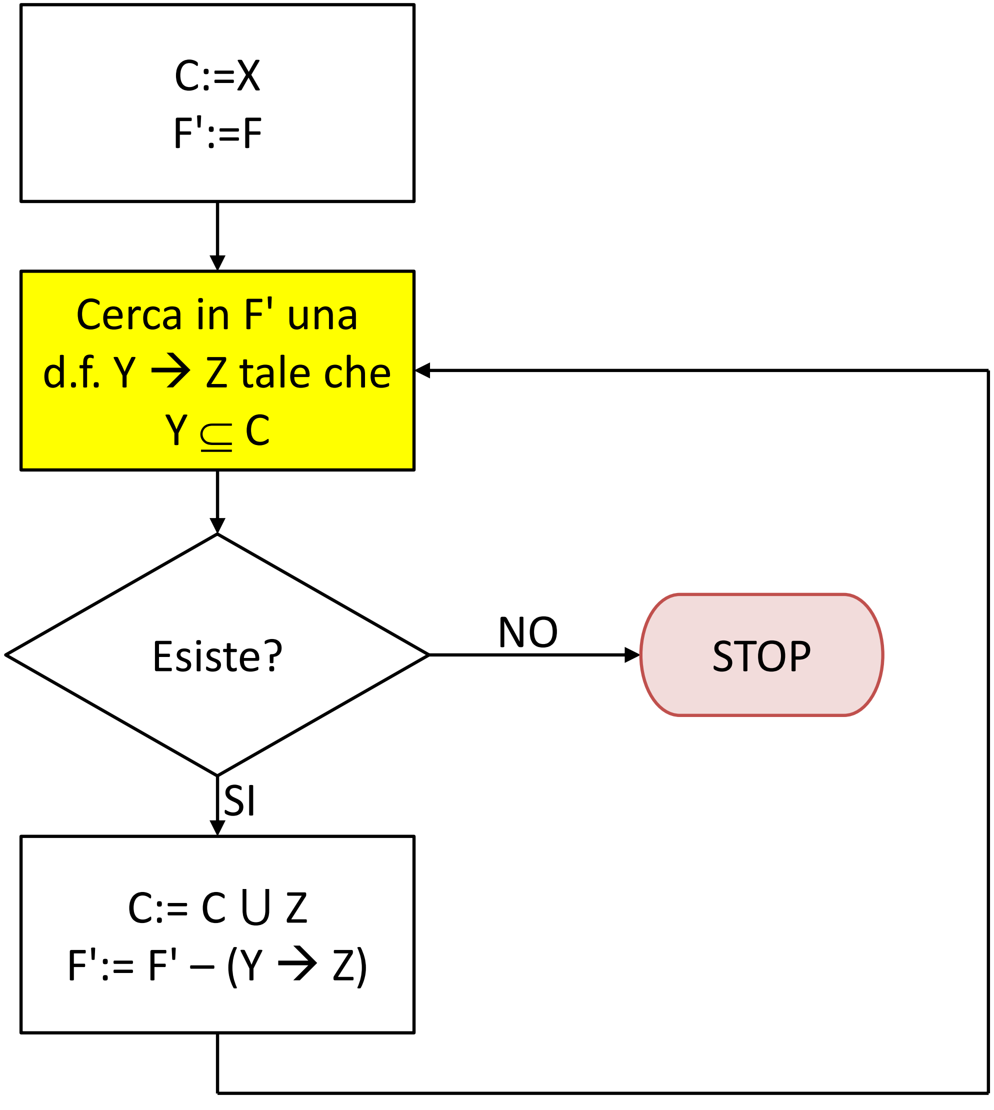


Esempio

- Parto con X = {CP}
- C = {CP} e F' = F
- Cerco una dipendenza funzionale in F' con antecedente in {CP}
- Esiste? Sì: CP → NP, Q
- C = {CP} ∪ {NP,Q} = {CP,NP,Q}
- Cerco una d.f. con antecedente in {CP,NP,Q}
- Esiste? Sì: Q → TU
- C = {CP,NP,Q} ∪ TU = {CP,NP,Q,TU}

F'
MATR → NS, IS, CAP, CF, DN
CF → MATR
IS → CAP
MATR, Co → Vo, DE, CP, NP
~~CP → NP, Q~~
~~Q → TU~~
CP → Co

Esempio

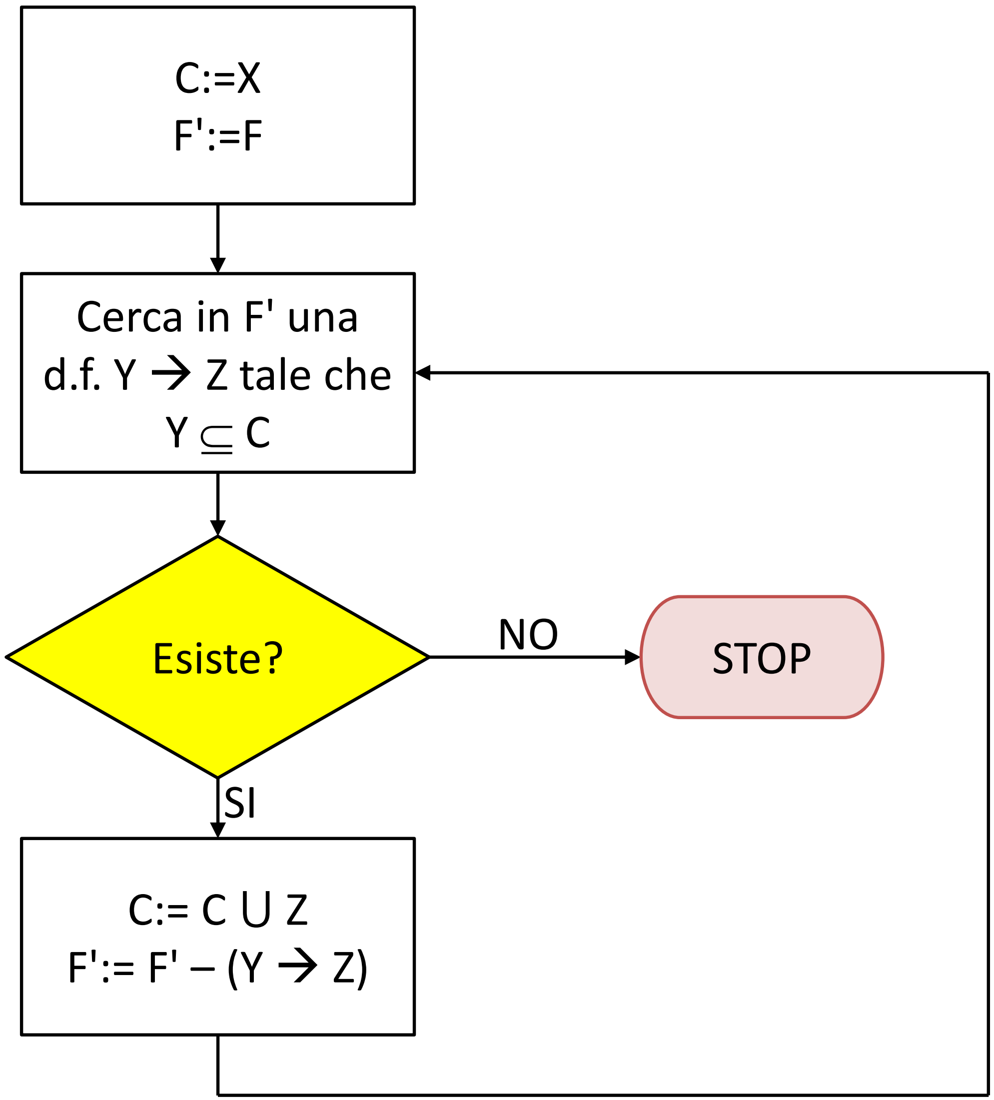

C:=X
F':=F
Cerca in F' una d.f. Y → Z tale che Y ⊆ C
Esiste?
NO
STOP
SI
C:= C U Z
F':= F' – (Y → Z)

Esempio

- Parto con X = {CP}
- C = {CP} e F' = F
- Cerco una dipendenza funzionale in F' con antecedente in {CP}
- Esiste? Sì: CP → NP, Q
- C = {CP} ∪ {NP,Q} = {CP,NP,Q}
- Cerco una d.f. con antecedente in {CP,NP,Q}
- Esiste? Sì: Q → TU
- C = {CP,NP,Q} ∪ TU = {CP,NP,Q,TU}
- Cerco una d.f. con antecedente in {CP,NP,Q,TU}

F'
MATR → NS, IS, CAP, CF, DN
CF → MATR
IS → CAP
MATR, Co → Vo, DE, CP, NP
~~CP → NP, Q~~
~~Q → TU~~
CP → Co

Esempio

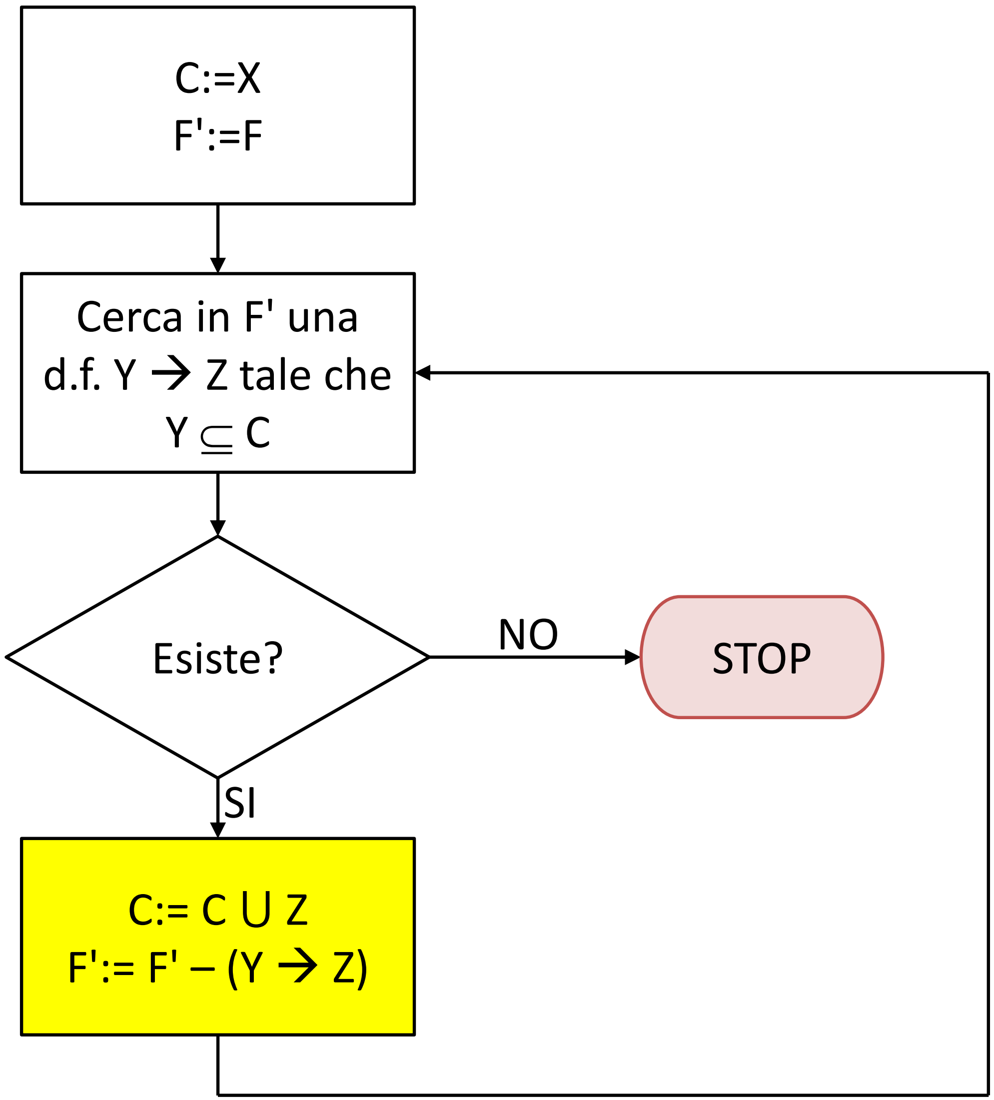


Esempio

- Parto con X = {CP}
- C = {CP} e F' = F
- Cerco una dipendenza funzionale
  in F' con antecedente in {CP}
- Esiste? Sì: CP → NP, Q
- C = {CP} ∪ {NP,Q} = {CP,NP,Q}
- Cerco una d.f. con antecedente in {CP,NP,Q}
- Esiste? Sì: Q → TU
- C = {CP,NP,Q} ∪ TU = {CP,NP,Q,TU}
- Cerco una d.f. con antecedente in {CP,NP,Q,TU}
- Esiste? Sì: CP → Co

F'
MATR → NS, IS, CAP, CF, DN
CF → MATR
IS → CAP
MATR, Co → Vo, DE, CP, NP
~~CP → NP, Q~~
~~Q → TU~~
CP → Co

Esempio

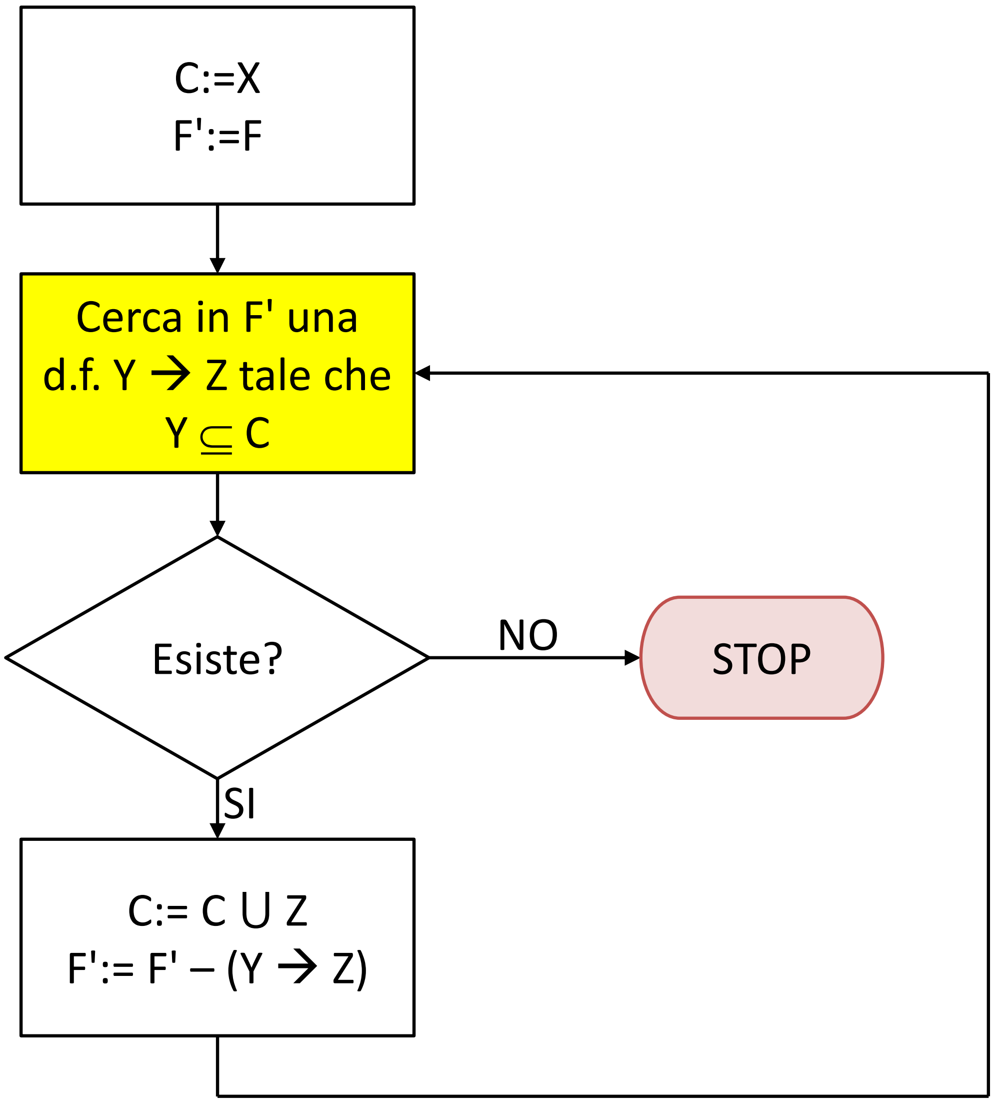


Esempio

- Parto con X = {CP}
- C = {CP} e F' = F
- Cerco una dipendenza funzionale in F' con antecedente in {CP}
- Esiste? Sì: CP → NP, Q
- C = {CP} ∪ {NP,Q} = {CP,NP,Q}
- Cerco una d.f. con antecedente in {CP,NP,Q}
- Esiste? Sì: Q → TU
- C = {CP,NP,Q} ∪ TU = {CP,NP,Q,TU}
- Cerco una d.f. con antecedente in {CP,NP,Q,TU}
- Esiste? Sì: CP → Co
- C = {CP,NP,Q,TU} ∪ Co = {CP,NP,Q,TU,Co}

F'
MATR → NS, IS, CAP, CF, DN
CF → MATR
IS → CAP
MATR, Co → Vo, DE, CP, NP
~~CP → NP, Q~~
~~Q → TU~~
~~CP → Co~~

Esempio

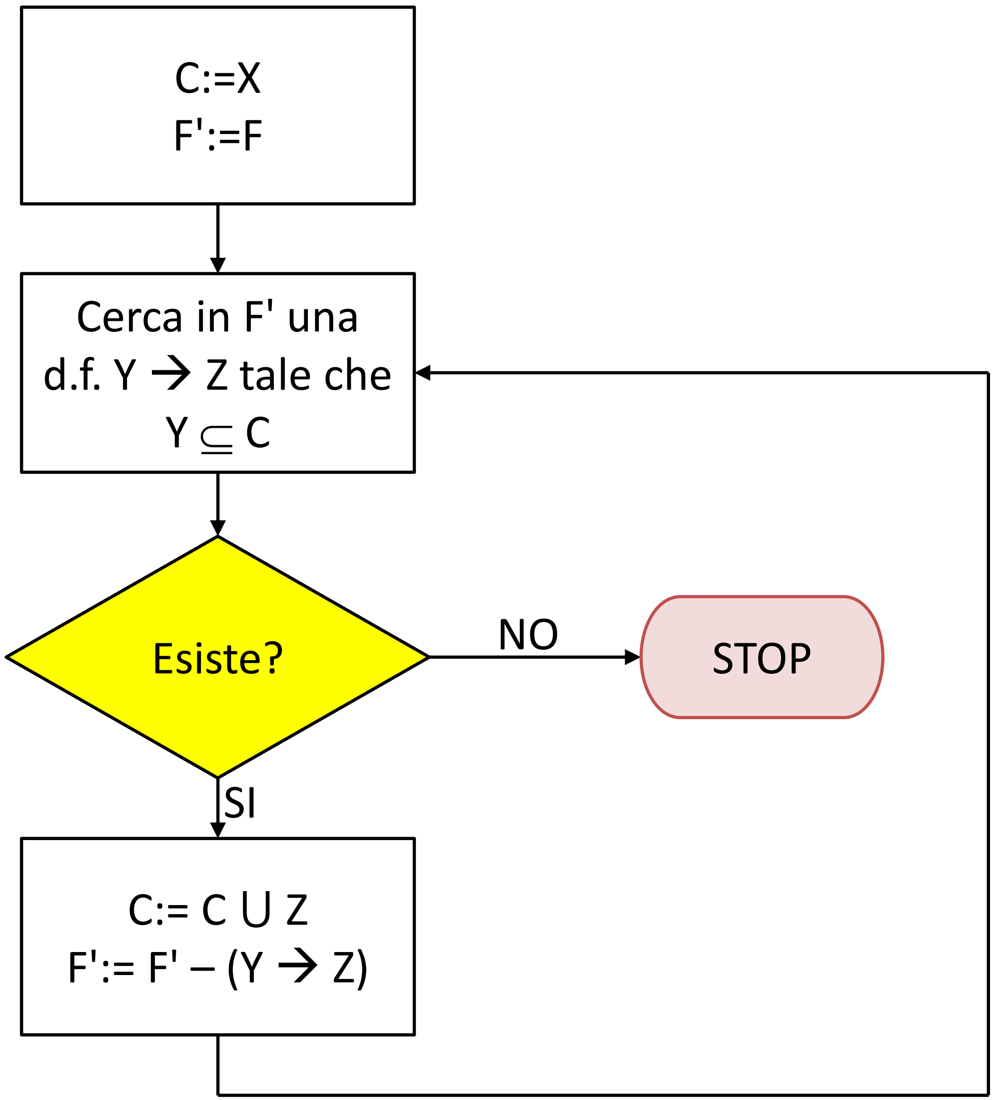

C:=X
F':=F
Cerca in F' una
d.f. Y → Z tale che
Y ⊆ C
Esiste?
NO
STOP
SI
C:= C U Z
F':= F' – (Y → Z)

Esempio

F'
MATR → NS, IS, CAP, CF, DN
CF → MATR
IS → CAP
MATR, Co → Vo, DE, CP, NP
~~CP → NP, Q~~
~~Q → TU~~
~~CP → Co~~

- Parto con X = {CP}
- C = {CP} e F' = F
- Cerco una dipendenza funzionale in F' con antecedente in {CP}
- Esiste? Sì: CP → NP, Q
- C = {CP} ∪ {NP,Q} = {CP,NP,Q}
- Cerco una d.f. con antecedente in {CP,NP,Q}
- Esiste? Sì: Q → TU
- C = {CP,NP,Q} ∪ TU = {CP,NP,Q,TU}
- Cerco una d.f. con antecedente in {CP,NP,Q,TU}
- Esiste? Sì: CP → Co
- C = {CP,NP,Q,TU} ∪ Co = {CP,NP,Q,TU,Co}
- Cerco una d.f. con antecedente in {CP,NP,Q,TU,Co}

Esempio

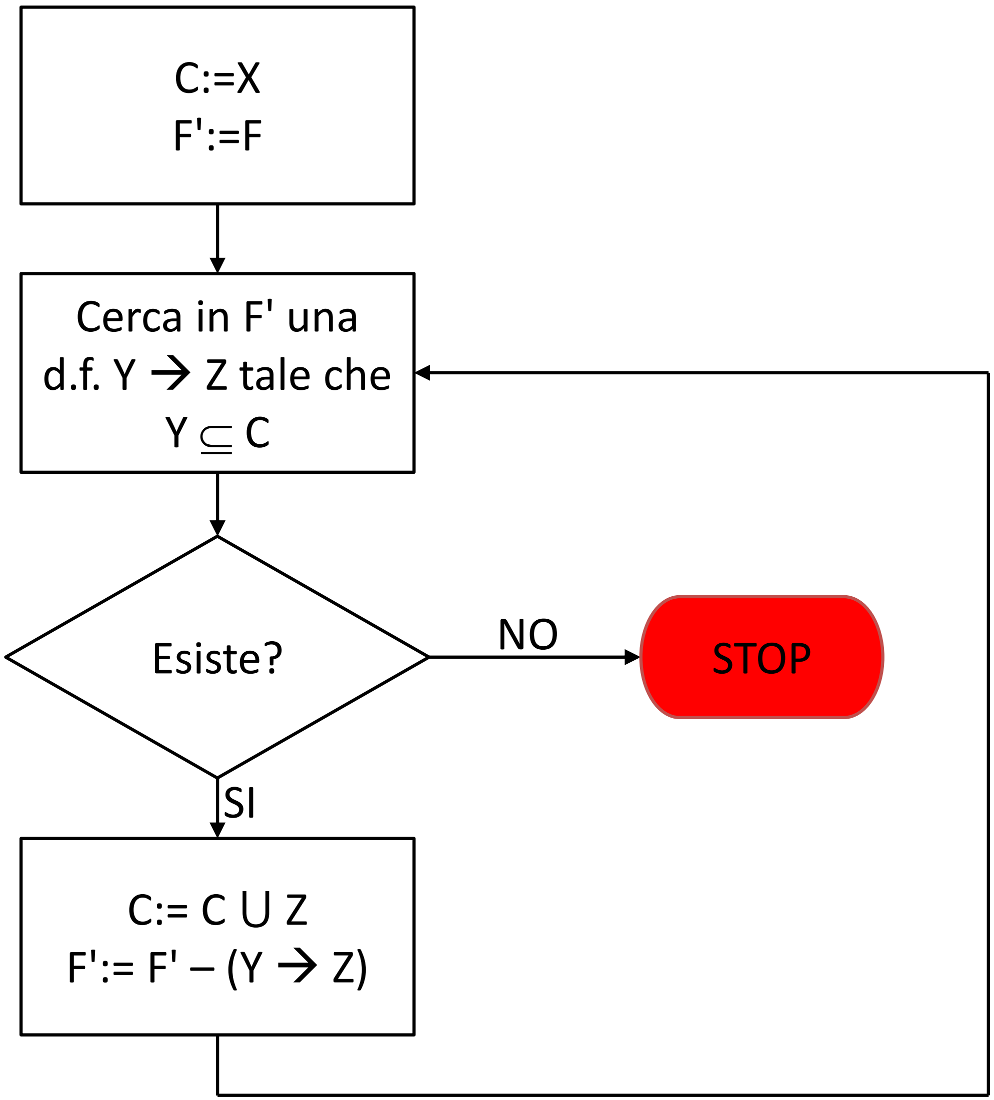


Esempio

F'
MATR → NS, IS, CAP, CF, DN
CF → MATR
IS → CAP
MATR, Co → Vo, DE, CP, NP
~~CP → NP, Q~~
~~Q → TU~~
~~CP → Co~~

- Parto con X = {CP}
- C = {CP} e F' = F
- Cerco una dipendenza funzionale in F' con antecedente in {CP}
- Esiste? Sì: CP → NP, Q
- C = {CP} ∪ {NP,Q} = {CP,NP,Q}
- Cerco una d.f. con antecedente in {CP,NP,Q}
- Esiste? Sì: Q → TU
- C = {CP,NP,Q} ∪ TU = {CP,NP,Q,TU}
- Cerco una d.f. con antecedente in {CP,NP,Q,TU}
- Esiste? Sì: CP → Co
- C = {CP,NP,Q,TU} ∪ Co = {CP,NP,Q,TU,Co}
- Cerco una d.f. con antecedente in {CP,NP,Q,TU,Co}
- Esiste? **No!**

Esempio



Esempio

F'
MATR → NS, IS, CAP, CF, DN
CF → MATR
→ Vo, DE, CP, NP

- Parto con X = {CP}
- C = {CP} e F' = F
- Cerco una d
  in F' con ant
- Esiste? Sì: C
- C = {CP} U {
- Cerco una d
- Esiste? Sì: Q
- C = {CP,NP,Q
- Cerco una d.f. con antecedente in {CP,NP,Q,TU}
- Esiste? Sì: CP → Co
- C = {CP,NP,Q,TU} U Co = {CP,NP,Q,TU,Co}
- Cerco una d.f. con antecedente in {CP,NP,Q,TU,Co}
- Esiste? **No!**

$$X^+ = \{CP,NP,Q,TU,Co\}$$

Complessità dell'algoritmo

La complessità dell'algoritmo è **polinomiale** rispetto al numero di dipendenze funzionali e al numero di attributi

Proprietà dell'algoritmo

Tutte le proprietà si basano sulla deduzione logica, le problematiche sono quindi ricondotte a metodiche deduttive

L'algoritmo rende meccaniche le deduzioni, infatti si verifica la proprietà seguente:

$$F \vdash X \rightarrow Y \text{ se e solo se } Y \subseteq X^+$$

Dimostrazione

- Supponiamo di voler verificare che $X \rightarrow Y$
- Possiamo semplicemente applicare l'algoritmo con su X per calcolare X^+
- Se $Y \subseteq X^+$, $X \rightarrow Y$ è una dipendenza funzionale deducibile dall'insieme di dipendenze date per definizione

- Il viceversa è anche vero: se $X \rightarrow Y$ è deducibile da F, allora $Y \subseteq X^+$

Applicazione

Per verificare la dipendenza $MATR \rightarrow CAP$, basta applicare l'algoritmo con $X=\{MATR\}$ e verificare che $\{CAP\}$ sia contenuto nell'insieme X^+ prodotto in uscita dall'algoritmo

Otteniamo $X^+=\{MATR, NS, IS, CAP, CF, DN\}$, quindi possiamo concludere che $MATR \rightarrow CAP$ è una dipendenza valida nel sistema delle dipendenze F

Applicazione

Verificiamo ora MATR → Vo

Applichiamo l'algoritmo di chiusura su X={MATR} e otteniamo $X^+ = \{MATR, NS, IS, CAP, CF, DN\}$

Vo non è incluso in X^+ quindi *MATR* → *Vo* non è deducibile da *F*

Nota

Alcuni testi riformulano la proprietà

$$F \vdash X \rightarrow Y \text{ se e solo se } Y \subseteq X^+$$

nel seguente modo

$$(X \rightarrow Y) \in F^+ \text{ se e solo se } Y \subseteq X^+_F$$

chiamandola **proprietà di membership**

dove la chiusura X^+ su un insieme F è indicata come X^+_F