Capitolo 5

NORMALIZZAZIONE DI SCHEMI RELAZIONALI

Soluzione degli esercizi

1. Dimostriamo che $F \vdash X \to YZ$ implica $F \vdash X \to Y$. $F \vdash YZ \to Y$ per riflessività, e la tesi segue per transitività. $F \vdash X \to YZ \Rightarrow F \vdash X \to Z$ è analogo.

   Dimostriamo che se $F \vdash X \to Y$ (1) e $F \vdash WY \to Z$ (2), allora $F \vdash XW \to Z$. Per arricchimento, da (1) segue $F \vdash XW \to YW$ (3); la tesi segue per transitività da (3) e (2).

   Dimostriamo che se $F \vdash X \to YZ$ (1) e $F \vdash Z \to BW$ (2), allora $F \vdash X \to YZB$. Per decomposizione, da (1) segue $F \vdash X \to Z$ (3); per transitività da (3) e (2) segue $F \vdash X \to BW$ (4); per unione da (1) e (4) segue $F \vdash X \to YZBW$ (5); infine la tesi segue per decomposizione da (5).
2. Sia $F = \{A \to B, C \to B, D \to ABC, AC \to D\}$. Il calcolo di C^+ ed A^+ mostra che non sono presenti attributi estranei. La dipendenza $D \to B$ è la sola dipendenza ridondante. Lo schema in forma canonica è quindi: $G = \{A \to B, C \to B, D \to A, D \to C, AC \to D\}$.
3. Consideriamo un qualunque Y tale che $A \notin Y$. Esaminando il funzionamento dell'algoritmo di chiusura lenta, si osserva che A non può appartenere ad Y^+, per cui $A \notin Y$ implica che Y non può essere superchiave, per cui Y non può neppure essere una chiave.
4. $F = \{AB \to CDE, AC \to BDE, B \to C, C \to B, C \to D, B \to E\}$

   (a) Una copertura canonica per F è data da:

   $$G = \{B \to C, B \to D, B \to E, C \to B\}.$$

   (b) Ogni chiave deve contenere A, poiché A non è implicata da altri attributi. Quindi, dato che $AB^+ = T$, AB è chiave. Dato che $AC^+ = T$, AC è chiave. Calcolando AD^+, AE^+ e ADE^+, si verifica che non ci sono altre chiavi.

   (c) Lo schema non è in 3NF a causa delle dipendenze $C \to D$, $B \to E$, poiché D ed E non sono attributi primi e C e B non sono chiavi.

   (d) Applicando l'algoritmo di sintesi a G otteniamo la seguente decomposizione, che non è comunque la sola possibile: $\{BCDE, AB(o\ AC)\}$. Si noti che AB (o AC) va aggiunto perché altrimenti lo schema risultante non conterrebbe alcuna relazione che contenga una chiave dello schema originale (e inoltre andrebbe perduto l'attributo A).

5. (a): Per definizione, se uno schema relazionale è in BCNF allora per ogni $X \to A \in F^+$ non banale X è una superchiave, per cui lo schema è in 3NF. (b) equivale ad (a) per contrapposizione.
6. Dobbiamo proiettare $F = \{AB \to C, AC \to B, AD \to E, B \to D, BC \to A, E \to G\}$ su
   (a) $X = ABC$;
   (b) $X = ABCD$;
   (c) $X = ABCEG$;
   (d) $X = ABCH$;
   (e) $X = ABCDEGH$.

   Sia K l'insieme degli attributi che non appaiono a destra di nessuna dipendenza e sia U l'insieme degli attributi che non appaiono a sinistra di nessuna dipendenza; in questo caso $K = H$ e $U = GH$. Per proiettare le dipendenze su X calcoliamo Y^+ per ogni sottoinsieme stretto non vuoto di $X - U$, e, se $Y^+ \neq Y$, aggiungiamo $Y \to (X \cap Y^+ - Y)$ alle dipendenze della proiezione. Consideriamo per prima cosa i sottoinsiemi più piccoli e, ogni volta che scopriamo che $A \in Y^+$, ignoriamo tutti i soprainsiemi di YA, dato che A sarebbe estraneo nella dipendenza; indichiamo (*) sotto per indicare gli insiemi ignorati per questo motivo. In particolare, se troviamo un Y' tale che $Y'^+ = (X - K)$, allora non consideriamo più nessun soprainsieme di Y'.
   Tra i singoletti, gli unici con chiusura non banale sono $B^+ = BD$ ed $E^+ = EG$.
   Passando alle coppie abbiamo: $AB^+ = ABCDEG$, $AC^+ = ACBDEG$, $AD^+ = ADEG$, $AE^+ = AEG(**)$, $BC^+ = BCDAEG$, $BD^+ = (*)$, $BE^+ = BEDG(**)$, $CD^+ = CD(**)$, $CE^+ = CEG(**)$, $DE^+ = DEG(**)$. Le chiusure segnate con (**) non sono interessanti perché banali o perché il determinante può essere diviso in due sottoinsiemi Y' ed Y'' tali che $(Y' \cup Y'')^+ = Y'^+ \cup Y''^+$. Dato che AB^+, AC^+ e BC^+ contengono $X - K$, è inutile considerarne i soprainsiemi. Quindi, le sole terne da considerare sono ADE, BDE e CDE, utili nel caso 6e. $ADE^+ = (*)$, $BDE^+ = (*)$, $CDE^+ = CDEG(**)$. Possiamo ora calcolare una copertura per le proiezioni, considerando, per ciascun $Y \subset X$ con $(X \cap Y^+ - Y) \neq \emptyset$, la dipendenza $Y \to (X \cap Y^+ - Y)$. Consideriamo solo gli Y la cui chiusura sia "interessante" come sopra specificato, ovvero consideriamo solo: $B^+ = BD$, $E^+ = EG$, $AB^+ = ABCDEG$, $AC^+ = ACBDEG$, $AD^+ = ADEG$, $BC^+ = BCDAEG$.

   (a) $X = ABC$: $\{AB \to C, AC \to B, BC \to A\}$;
   (b) $X = ABCD$: $\{B \to D, AB \to CD, AC \to BD, BC \to AD\}$;
   (c) $X = ABCEG$: $\{E \to G, AB \to CEG, AC \to BEG, BC \to AEG\}$;
   (d) $X = ABCH$: $\{AB \to C, AC \to B, BC \to A\}$;
   (e) $X = ABCDEGH$: $\{B \to D, E \to G, AB \to CDEG, AC \to BDEG,$

$BC \to ADEG, AD \to EG\}$.

Lo coperture così ottenute non sono tutte canoniche. Le coperture canoniche corrispondenti sono:

(a) $X = ABC$: $\{AB \to C, AC \to B, BC \to A\}$;
(b) $X = ABCD$: $\{B \to D, AB \to C, AC \to B, BC \to A\}$;
(c) $X = ABCEG$: $\{E \to G, AB \to C, AC \to B, BC \to A, BC \to E\}$;
(d) $X = ABCH$: $\{AB \to C, AC \to B, BC \to A\}$;
(e) $X = ABCDEGH$: $\{B \to D, E \to G, AB \to C, AC \to B, BC \to A, AD \to E\}$.

Osserviamo che, se $Y_A \to A$ è una dipendenza non banale in F^+, allora $BC \subseteq Y_A$, ed analogamente $Y_B \to B$ e $Y_C \to C$ implicano $AC \subseteq Y_B$ e $AB \subseteq Y_C$, per cui ogni chiave di uno $X \supseteq ABC$ deve includere AB oppure AC oppure BC. D'altra parte, la chiusura di AB, AC, e BC, contiene tutti gli attributi tranne H. Da questo possiamo dedurre l'insieme delle chiavi e degli attributi primi dei cinque schemi.

(a) $X = ABC$: $\{AB \to C, AC \to B, BC \to A\}$;
chiavi: $\{AB, AC, BC\}$;
primi: $\{ABC\}$.
(b) $X = ABCD$: $\{B \to D, AB \to C, AC \to B, BC \to A\}$;
chiavi: $\{AB, AC, BC\}$;
primi: $\{ABC\}$.
(c) $X = ABCEG$: $\{E \to G, AB \to C, AC \to B, BC \to A, BC \to E\}$;
chiavi: $\{AB, AC, BC\}$;
primi: $\{ABC\}$.
(d) $X = ABCH$: $\{AB \to C, AC \to B, BC \to A\}$;
chiavi: $\{ABH, ACH, BCH\}$;
primi: $\{ABCH\}$.
(e) $X = ABCDEGH$: $\{B \to D, E \to G, AB \to C, AC \to B, BC \to A, AD \to E\}$;
chiavi: $\{ABH, ACH, BCH\}$;
primi: $\{ABCH\}$.

A questo punto è facile indicare per ogni schema la sua forma normale.

(a) $X = ABC$: $\{AB \to C, AC \to B, BC \to A\}$;
chiavi: $\{AB, AC, BC\}$;
primi: $\{ABC\}$, BCNF e quindi 3NF.
(b) $X = ABCD$: $\{B \to D, AB \to C, AC \to B, BC \to A\}$;
chiavi: $\{AB, AC, BC\}$;
primi: $\{ABC\}$, $B \to D$ viola le due forme normali.
(c) $X = ABCEG$: $\{E \to G, AB \to C, AC \to B, BC \to A, BC \to E\}$;
chiavi: $\{AB, AC, BC\}$;
primi: $\{ABC\}$, $E \to G$ viola le due forme normali.
(d) $X = ABCH$: $\{AB \to C, AC \to B, BC \to A\}$;
chiavi: $\{ABH, ACH, BCH\}$;
primi: $\{ABCH\}$, $AB \to C$ viola le due forme normali.

(e) $X = ABCDEGH$: $\{B \rightarrow D, E \rightarrow G, AB \rightarrow C, AC \rightarrow B, BC \rightarrow A, AD \rightarrow E\}$;
chiavi: $\{ABH, ACH, BCH\}$;
primi: $\{ABCH\}$, $B \rightarrow D$ viola le due forme normali.

7. Assumendo che $A \rightarrow C$ sia l'unico vincolo sullo schema, la decomposizione $\{R_1(A, B), R_2(B, C)\}$ non preserva i dati, per cui il confronto tra i due schemi non è molto significativo. Comunque, uno schema $R(A, B, C)$ dovrebbe semplificare la gestione di un vincolo $A \rightarrow C$ rispetto ad uno schema $\{R_1(A, B), R_2(B, C)\}$, dato che nel secondo caso, per verificare se l'inserimento di un'ennupla in R_1 o in R_2 viola il vincolo, è necessario accedere alle due relazioni R_1 ed R_2, anziché alla sola relazione R.
8. Le regole date producono sempre uno schema in BCNF, a patto che gli attributi di ciascuna collezione dello schema ad oggetti rispettino la BCNF, il che in genere avviene per uno schema ben disegnato.
9. La nozione di 4NF, non definita nel testo, formalizza la necessità di rappresentare gli attributi multivalore separatamente dagli altri attributi per evitare ridondanza dei dati. Così, dato lo schema di partenza $I\langle\{CF, C, Tel, NomF\}, \{CF \rightarrow C\}\rangle$, la decomposizione $\{I(CF, C), I2(CF, Tel, NomF)\}$ è in 3NF (perchè la chiave di $I2$ è $\{CF, Tel, NomF\}$), ma la realazione $I2$ è molto ridondante. In questo caso l'eliminazione della ridondanza avviene con la decomposizione: $\{I(CF, C), ITel(CF, Tel), INomF (CF, NomF)\}$ che è in 4NF.
10. Consideriamo per prima cosa le dipendenze con a sinistra un singoletto. Abbiamo: $A \rightarrow B$, $C \rightarrow B$, $D \rightarrow ABC$. Tra i determinanti con due attributi, è inutile considerare quelli che contengono D, poiché D è chiave da solo, né quelli che contengono B, poiché B è costante. Rimane la dipendenza $AC \rightarrow BD$. Portando in forma canonica, otteniamo: $\{A \rightarrow B, C \rightarrow B, D \rightarrow A, D \rightarrow C, AC \rightarrow D\}$.
11. Schema: $I(N, L, S), \{N \rightarrow LS, L \rightarrow S\}$. Portandolo in forma canonica, abbiamo: $\{N \rightarrow L, L \rightarrow S\}$, chiavi: $\{N\}$.

    (a) Lo schema non è in 3NF né, quindi, in BCNF, per la dipendenza $L \rightarrow S$.
    (b) Algoritmo di sintesi: $R1(NL), \{N \rightarrow L\}$; $R2(LS), \{L \rightarrow S\}$. Non c'è bisogno di aggiungere schemi poiché NL è già una superchiave.
    (c) Algoritmo di analisi: poiché $L \rightarrow S$ viola la 4NF, decompongo in $R1(LS), \{L \rightarrow S\}$ ed $R1(LN), \{N \rightarrow L\}$, i due schemi sono in 4NF, per cui ho finito. La decomposizione ha preservato le dipendenze.

12. Dal testo ricaviamo le seguenti dipendenze, che sono già in forma normale:

    (1) NInventario → Modello, NSerie, Costo
    (2) Modello → Descrizione
    (3) Modello, NSerie → NInventario
    (4) NInventario → Responsabile
    (5) Responsabile → Telefono

    Le chiavi dello schema sono { NInventario; (Modello, NSerie) }. Lo schema presenta numerose anomalie, testimoniate dalle dipendenze (2), (4) e (5). Applicando l'algoritmo di sintesi, otteniamo il seguente schema, che rispetta sia la 3NF che la BCNF:

Apparecchiature(NInventario, Modello, NSerie, Costo, Responsabile),
Modelli(Modello, Descrizione),
Responsabili(Responsabile, Telefono).

13. Le colonne dello schema attuale sono le seguenti, di cui riportiamo un'abbreviazione: SiglaCorso (SiglaC), Tipologia (Tipo), NomeInsegnante (NomeI), IndirizzoInsegnante (IndI), NomeAllievo (NomeA), TelefonoAllievo (TelA), VersatoFinora (Vers).

    (a) Dipendenze funzionali:

        i. SiglaC → Tipo, NomeI
        ii. NomeI → IndI
        iii. NomeA → TelA
        iv. NomeA, SiglaC → Vers

    (b) Le dipendenza sono già in forma canonica. La sola chiave è la coppia *(NomeA, SiglaC)*.

    (c) Applicando l'algoritmo di sintesi otteniamo lo schema:
    Corsi(SiglaC, Tipo, NomeI),
    Insegnanti(NomeI, indI),
    Allievi(NomeA, TelA),
    SoldiVersati(NomeA, SiglaC, Vers),
    che è anche in BCNF.

    (d) Considerando le dipendenze nell'ordine in cui sono elencate, otteniamo la seguente decomposizione:
    R(SiglaC, Tipo, NomeI, IndI, NomeA, TelA, Vers),
    {*SiglaC → (Tipo, NomeI), NomeI → IndI, NomeA → TelA, (NomeA, SiglaC) → Vers*} ⇒
    R1(SiglaC, Tipo, NomeI, IndI), {*SiglaC → (Tipo, NomeI), NomeI → IndI*},
    R2(SiglaC, NomeA, TelA, Vers), {*NomeA → TelA, (NomeA, SiglaC) → Vers*} ⇒
    R11(NomeI, IndI), {*NomeI → IndI*},
    R12(NomeI, SiglaC, Tipo), {*SiglaC → (Tipo, NomeI)*},
    R2(SiglaC, NomeA, TelA, Vers), {*NomeA → TelA, (NomeA, SiglaC) → Vers*} ⇒
    R11(NomeI, IndI), {*NomeI → IndI*},
    R12(NomeI, SiglaC, Tipo), {*SiglaC → Tipo, NomeI*},
    R21(NomeA, TelA), {*NomeA → TelA*},
    R22(NomeA, SiglaC, Vers), {*NomeA, SiglaC → Vers*}.

    Questa decomposizione è uguale a quella ottenuta usando l'algorimo di sintesi, e conserva anche le dipendenze.

14. (a) Problema: dato lo schema di relazione $R\langle T, F\rangle$, $A \in T$, A è primo?
    Discussione: Un algoritmo per valutare la primalità di un attributo A consiste nel generare tutti i sottoinsiemi di T che contengono A e verificare se uno di essi sia una chiave. Questo algoritmo ha complessità

esponenziale rispetto al numero di attributi, poiché il numero di sottoinsiemi di un insieme di a elementi è 2^a, e verificare se un sottoinsieme di un insieme di a attributi sia una chiave, rispetto ad un insieme di p dipendenze, ha complessità polinomiale $O(ap)$. Non sono noti algoritmi polinomiali.

(b) Problema: Dati due insiemi di dipendenze F e G, $F \equiv G$?
Discussione: Il problema ammette un algoritmo polinomiale. È sufficiente verificare, per ogni $X \rightarrow Y \in F$ che si abbia $Y \subseteq X^+_G$, e che per ogni $X \rightarrow Y \in G$ si abbia che $Y \subseteq X^+_F$. Dato che la chiusura e l'inclusione possono essere valutate in tempo $O(ap)$, la complessità dell'algoritmo è quindi $O(ap^2)$.

(c) Problema: Dato lo schema di relazione $R\langle T, F\rangle$, e $X \subseteq T$, X è una superchiave?
Discussione: È sufficiente verificare se $T = X^+_F$, con complessità $O(ap)$.

(d) Dato lo schema di relazione $R\langle T, F\rangle$, e $X \subseteq T$, X è una chiave?
Discussione: È sufficiente verificare che si abbia $X^+_F = T$ e che, per ogni $A \in X$, si abbia $(X - A)^+_F \neq T$. La complessità è quindi $O(a^2 p)$.

15. (a) Trovare una chiave di $R\langle T, F\rangle$: si può utilizzare il seguente algoritmo, dove $A[1..n]$ enumera gli attributi in T:

```
input: A[1..n], F;
K := [A[1], ..., A[n]];
for i in 1..n
    if chiudi((K − A[i]),F) = T then K := K − A[i];
end
return(K)
```

È facile dimostrare le seguenti invarianti:

– sia K_i il valore di K dopo il ciclo i-esimo; per ogni i, si ha che $K_i \rightarrow T$;
– per ogni $j \leq i$, se l'attributo $A[j]$ appartiene a K_i, allora $A[j]$ non è estraneo in $K_i \rightarrow T$.

Ponendo $i = n$, abbiamo quindi che K_n è una superchiave senza attributi estranei, ovvero è una chiave.

(b) Trovare tutte le chiavi di $R\langle T, F\rangle$: in generale, il numero di chiavi di uno schema può crescere in modo esponenziale con le dimensioni dello schema stesso, per cui nessun algoritmo che le enumeri può avere una complessità meno che esponenziale rispetto alle dimensioni dello schema.

(c) $F - \{X \rightarrow Y\} \equiv F$: basta verificare se $X^+_{F-\{X\rightarrow Y\}} \supseteq Y$, con costo $O(ap)$.

16. (a) Test 3NF: Un modo per determinare se uno schema $R\langle T, F\rangle$ sia in 3NF consiste nel portare lo schema in una forma canonica G e poi nel verificare se, per ogni dipendenza in $X \rightarrow A$ in G, se X non è una chiave, allora A è primo. Per portare F in forma canonica è sufficiente:

i. dividere le dipendenze in modo che ogni membro destro sia composto di un solo attributo;

   ii. eliminare, da ogni membro sinistro, gli attributi ridondanti;
   iii. eliminare le dipendenze ridondanti.

   Tutte queste operazioni si possono effettuare in tempo polinomiale.

   Data poi una dipendenza $X \rightarrow A$ in G, verificare se X sia chiave si può fare in tempo polinomiale, poiché significa verificare se $X^+ = T$; tuttavia, quando X non è chiave, verificare se A sia primo usando l'algoritmo sopra descritto richiede un tempo esponenziale. Quindi questo algoritmo ha complessità esponenziale. Non sono noti algoritmi polinomiali.

(b) Test BCNF: Questo problema ha complessità polinomiale; basta utilizzare lo stesso algoritmo visto al punto precedente per portare F in forma canonica G. Per ogni $X \rightarrow A$ in G bisogna poi verificare se X sia una chiave, e questa operazione è polinomiale.

(c) Proiezione delle dipendenze: per ogni $Y \subset X$ non vuoto si calcoli Y_F^+, e si generi in questo modo l'insieme $G = \{Y \rightarrow A \mid Y \subset X,\ A \in ((Y_F^+ - Y) \cap X)\}$. Questa operazione ha una complessità esponenziale rispetto alla dimensione di X. Si porti poi G in forma canonica, con un costo polinomiale rispetto alla dimensione di G (che, nel caso pessimo, è esponenziale rispetto alla dimensione di X).

   In pratica, si possono adottare gli accorgimenti descritti in 6 per ridurre la quantità di sottoinsiemi di X da considerare, ma questi non bastano a rendere la complessità di questo algoritmo meno che esponenziale. Non sono noti algoritmi polinomiali.

(d) Test BCNF di sottoschema: Per risolvere questo problema si può operare come segue: prima si proietta F su X, operando come descritto sopra, e poi si verifica se $R\langle X, \Pi_X F\rangle$ è in BCNF. Questo algoritmo è esponenziale, dato che questo è il costo della fase di proiezione.

(e) Test copertura canonica: L'algoritmo più semplice verifica le seguenti condizioni:

   i. Tutti i membri destri sono formati da un solo attributo: costo $O(p)$
   ii. Per ogni $X \rightarrow A \in F$, per ogni $B \in X$, $A \notin (X - B)^+$: costo $p \times a \times O(ap) = O(a^2 p^2)$
   iii. Per ogni $X \rightarrow A \in F$, $A \in X^+$, dove la chiusura è calcolata rispetto ad $F - \{X \rightarrow A\}$: costo $p \times O(ap) = O(ap^2)$

   Costo totale dell'algoritmo: $O(a^2 p^2)$.

17. Se un'istanza di relazione r soddisfa una dipendenza $X \rightarrow Y$, allora $\forall t_1, t_2 \in r.\ t_1[X] = t_2[X] \rightarrow t_1[Y] = t_2[Y]$. Da $s \subseteq r$ segue quindi che anche $\forall t_1, t_2 \in s.\ t_1[X] = t_2[X] \rightarrow t_1[Y] = t_2[Y]$, per cui s soddisfa $X \rightarrow Y$.
18. Ragionando come in 17, si dimostra che $r \cap s$ soddisfa $X \rightarrow Y$. Invece $r \cup s$ potrebbe non soddisfare $X \rightarrow Y$. Si considerino le istanze di relazione $\{(A = 1, B = 1)\}$ e $\{(A = 1, B = 2)\}$, con schema $R(AB)$. Ciascuna delle deu soddisfa $A \rightarrow B$, tuttavia la loro unione $\{(A = 1, B = 1), (A = 1, B = 2)\}$ non soddisfa la dipendenza.
19. Sia Y la sola chiave di $R\langle T, F\rangle$, e assumiamo, per assurdo, che $R\langle T, F\rangle$ sia in 3NF ma non in BCNF. Ne segue che esiste una dipendenza non banale $X \rightarrow A$ tale che X non è superchiave ma $A \in Y$. Da $X \rightarrow A$ segue che

$XY - A$ determina Y, per cui $XY - A$ è una superchiave, per cui contiene una chiave la quale non contiene A. Quindi $R\langle T, F\rangle$ ha almeno due chiavi, contraddicendo l'ipotesi.

20. Dato uno schema $R\langle T, F\rangle$ vogliamo dimostrare che, se per ogni $X \rightarrow Y \in F$ non banale X è una superchiave, allora per ogni $X \rightarrow Y \in F^+$ non banale X è una superchiave (l'implicazione inversa è immediata dato che $F \subseteq F^+$). A tale scopo basta osservare che F^+ deriva da F tramite l'applicazione degli assiomi di Armstrong, e questi, partendo da uno schema che soddisfa la condizione sopra specificata, producono ancora solo dipendenze che o sono banali o hanno a sinistra una superchiave. Infatti, la riflessività aggiunge solo dipendenze banali. L'arricchimento ricava una dipendenza con determinante XW a partire da una con determinante X; se la dipendenza ottenuta è non banale allora anche quella originale era non banale, per cui X è una superchiave, per cui XW è una superchiave. Si consideri infine la derivazione di $X \rightarrow Z$ a partire da $X \rightarrow Y$ e $Y \rightarrow Z$ per transitività. Se $X \rightarrow Y$ e $Y \rightarrow Z$ sono entrambe banali, allora anche $X \rightarrow Z$ è banale. Se almeno una è non banale allora almeno una tra X ed Y è superchiave, e quindi, dato che X determina Y, X è superchiave.
21. Diciamo che uno schema $R\langle T, F\rangle$ soddisfa la condizione (p) se per ogni $X \rightarrow A_1, \ldots, A_n \in F$ e per ogni i, si ha che o vale X superchiave (p^1_X) o vale $A_i \in X$ $(p^2_{A_i,X})$, o vale A_i *primo* $(p^3_{A_i})$. Nel seguito, abbreviamo spesso (p^1_X), $(p^2_{A_i,X})$ e $(p^3_{A_i})$ a (p^1), (p^2) e (p^3). Vogliamo dimostrare che, se $R\langle T, F\rangle$ soddisfa (p), allora $R\langle T, F\rangle$ è in 3NF (l'implicazione inversa è banale).

   A tale scopo basta osservare che F^+ deriva da F tramite l'applicazione degli assiomi di Armostrong, e questi, partendo da uno schema che soddisfa (p), producono ancora solo dipendenze che soddisfano (p). Infatti, la riflessività aggiunge solo dipendenze che soddisfano (p^2). L'arricchimento ricava $XW \rightarrow A_1, \ldots, A_n, W$ a partire da $X \rightarrow A_1, \ldots, A_n$; se valeva (p^1_X), allora vale (p^1_{XW}) per la nuova dipendenza. Altrimenti, ciascuno degli attributi A_i continua a soddisfare (p^2) o (p^3) come in $X \rightarrow A_1, \ldots, A_n$, e tutti gli attributi in W soddisfano (p^2), dato che appartengono a XW.

   Si consideri infine la derivazione di $X \rightarrow A_1, \ldots, A_n$ a partire da $X \rightarrow Y$ e $Y \rightarrow A_1, \ldots, A_n$ per transitività, e si consideri un A_i. Se X è superchiave, allora (p^1_X) vale, e abbiamo finito. Se invece X non è superchiave, allora neppure Y è superchiave, e quindi, dato che $Y \rightarrow A_i$ soddisfa (p), o A_i soddisfa $(p^3_{A_i})$, oppure A_i non soddisfa $(p^3_{A_i})$ e $A_i \in Y$. In quest'ultimo caso, dato che $X \rightarrow Y$ soddisfa (p), X non è superchiave, A_i non soddisfa (p^3), e $A_i \in Y$, allora A_i deve soddisfare $(p^2_{A_i,X})$, c.v.d.