Database

Raffaele Daniele Facendola, Matteo Guarnerio, Francesco Gusmeroli

Novembre 16, 2011

Contents

1 Algebra relazionale

Siano x ed y due tabelle, definiamo le operazioni dell'algebra relazionale come:
*NB: Due tabelle si dicono **compatibili** se hanno lo stesso schema.+*
NB: La notazione x., indica l'insieme di **tutti** gli attributi di x.*

- **FONDAMENTALI**
  - **Selezione**: $\sigma_{a=value}(x)$, restituisce lo stesso schema di x e tutte le tuple di x che hanno attributo a = value. E' possibile usare diversi predicati di comparazione (=, $\neq$, <, >) connessi tra di loro mediante le operazioni booleane.
  - **Proiezione:** $\Pi_{a,b,...}(x)$**,** restituisce solo gli attributi a, b, ecc.. della tabella x e tutte le tuple appartenenti ad essa.
  - **Unione**: $x \cup y$, x è compatibile con y, restituisce lo stesso schema di x in cui sono presenti tutte le tuple di x e tutte le tuple di y (senza le eventuali ripetizioni).
  - **Differenza:** x-y, x è compatibile con y, restituisce lo stesso schema di x in cui sono presente tutte le tuple di x che **non** appartengono ad y.
  - **Prodotto cartesiano:** $x \times y$, restituisce uno schema composto da tutti gli attributi di x e di y e tutte le tuple che sono combinazione di ciascuna tupla di x con ciascuna tupla di y.
- **DERIVATE**
  - **Intersezione:** $x \cap y$, derivata da $x-(x-y)$, restituisce lo stesso schema di x (compatibile con y) e tutte le tuple che fanno parte sia di x che di y.
  - **Join:** $x \bowtie_{condizioni} y$, derivato da $\sigma_{condizioni}(x \times y)$, restituisce le combinazioni di tuple di x ed y che soddisfano le "condizioni".
    - **Equi-join**: se le condizioni ammettono solo confronti di uguaglianza
    - **Join-naturale**: equijoin di tutti gli attributi omonimi di x ed y (si elimina la colonna ripetuta e si omettono le condizioni).
    - **Semi-join**: $x \rhd\!< y$, derivata da $\Pi_{x.*}(x \bowtie_{condizioni} y)$, equivalente al join di cui vengono visualizzati solo gli attributi di x.
  - **Divisione:** $x \div y$**,** derivata da $\Pi_{x-y}x - \Pi_{x-y}((\Pi_{x-y}x \times y) - x)$, equivalente all'operazione inversa della join. Seleziona le tuple di x che contengono **tutte** le tuple di y.
- **NON ALGEBRICHE**
  - **Assegnamento:** $nome = op$ serve per nominare come "nome" il risultato di un operazione "op".
  - **Ridenominazione:** $\rho_{vecchio \leftarrow nuovo}x$ serve per rinominare un attributo di x "vecchio" come "nuovo"

E' possibile scrivere le espressioni di algebra relazionale usando la **notazione ad albero,** in cui la radice è rappresentata dall'operazione e i figli (massimo 2) rappresentano gli argomenti (tabelle o operazioni).

1.1 Ottimizzazione

- **Eliminazione dei prodotti cartesiani:** $\sigma_p(x \times y) \equiv x \bowtie_p y$
- **Push della selezione rispetto al join:** $\sigma_p(x \bowtie_c y) \equiv (\sigma_{px}x) \bowtie_c (\sigma_{py}y)$ dove px sono tutti i predicati di p che riguardano gli attributi di x, e py quelli che riguardano quelli di y.
- **Push della proiezione rispetto al join**: $\Pi_l(x \bowtie_c y) \equiv \Pi_L((\Pi_{Lx \cup Jx}x) \bowtie_c (\Pi_{Ly \cup Jy}y))$ dove "c" agisce sugli attributi di x "Jx" e di y "Jy" e Lx = L - SCHEMA(y), Ly = L - SCHEMA(x).
- **Atomizzazione delle selezioni:** $\sigma_{P,Q}x \equiv \sigma_P(\sigma_Q x)$
- **Idempotenza della proiezione:** $\Pi_L x \equiv \Pi_L \Pi_T x$ con $L \subseteq T$
- **Push della selezione rispetto all'unione:** $\sigma_P(x \cup y) \equiv (\sigma_P x) \cup (\sigma_P y)$
- **Push della selezione rispetto alla differenza:** $\sigma_P(x - y) \equiv (\sigma_P x) - (\sigma_P y) \equiv (\sigma_P x) - y$ (è possibile omettere la selezione su y in quanto i suoi attributi in ogni caso non fanno parte del risultato dell'espressione).
- **Push della proiezione rispetto all'unione**: $\Pi_L(x \cup y) \equiv (\Pi_L x) \cup (\Pi_L y)$
- **Distributività di join rispetto all'unione:** $(x \cup y) \bowtie (z \cup w) \equiv (x \bowtie z) \cup (x \bowtie w) \cup (y \bowtie z) \cup (y \bowtie w)$

2 Calcolo relazionale - TRC

Un'interrogazone è della forma $\{t \mid p(t)\}$ ovvero tutte le tuple t che tali per cui $p(t)$ è vera, dove $p(t)$ è una formula costruita tramite **atomi**.

Tra gli atomi riconosciamo:

- tutte le tuple t appartenenti ad un insieme r: $t \in r$
- tutte le operazioni di confronto tra tuple t_1 in un insieme A_1 con tuple t_2 in un insieme A_2: $t_1\{<, >, =, \neq\}t_2$
- tutte le operazion di confrontro tra tuple e costanti.

Definiamo **formule ben formate** (WFF):

- Un atomo è una WFF
- Se p è una WFF allora anche (p) e $\neg p$
- Se p_1 e p_2 sono WFF, lo sono anche $p_1 \wedge p_2$, $p_1 \vee p_2$ e $p_1 \Rightarrow p_2$
- Se p è una wff in cui s è una variabile, lo sono anche $\exists s \in r(p(s))$ e $\forall s \in r(p(s))$

Una formula si dice **unsafe** se restituisce un insieme infinito:

- $\{t \mid t \notin r\}$

2.1 Equivalenza tra algebra relazionale e calcolo relazionale

- **Selezione**: $\sigma_{A=1} = \{t | \exists t_1 \in r(t_1[A] = 1) \wedge t = t_1\}$
- **Proiezione:** $\Pi_{AC} r = \{t | \exists t_1 \in (t[A, C] = t_1[A, C])\}$
- **Prodotto cartesiano:** $r(A, B, C) \times s(D, E, F) = \{t | \exists t_1 \in r, \exists t_2 \in s(t[A, C, B] = t_1[A, B, C] \vee t[D, E, F] = t_2[D, E, F])\}$
- **Unione:** $r \cup s = \{t | (\exists t_1 \in r, t = t_1) \vee (\exists t_2 \in s, t = t_2)\}$
- **Differenza**: $r - s = \{t | \exists t_1 \in r(t = t_1) \wedge \neg(\exists t_2 \in s, t = t_2)\}$
- **Esempio di join**: $R(A, C) \bowtie_{A=B} S(B, D) \Longleftrightarrow \{t | \exists t_1 \in r, \exists t_2 \in s(t[A, C] = t_1[A, C] \vee t[B, D] = t_2[B, D] \vee t_1[A] = t_2[B])\}$

3 Datalog

Un programma datalog è un insieme di regole

Ogni regola è composta da una testa (head o LHS) e da un corpo (body o RHS): $p : -p_1, p_2, ..., p_n$, dove p è detto *letterale*, ovvero una funzione con alcuni parametri $f(x, y, ..., z)$

Per questioni di sicurezza tutte le variabili che compaiono in LHS **devono** comparire in RHS.

LHS è vero se RHS è vero, in tal caso la regola è vera. Il corpo è vero se per ogni letterale presente in esso tutte le variabili sono *unificabili*, ovvero è possibile trovare una combinazione di costanti che, sostituite ai parametri, rendono vero il letterale.

Qualora il valore assunto da una variabile fosse ininfluente ai fini della valutazione di un letterale, è possibile sostituire al valore di un parametro il simbolo "_" (**don't care**)

3.1 Tuple

Una n-upla in datalog si dichiara nel seguente modo: $NomeTupla("valore1","valore2",...,"valoren")$. La precedente è anche detto **fatto di base** (o *letterale ground*).

3.2 Query

Le query in datalog si esprimono mediante i cosiddetti **goal** della forma: $? - p_1, p_2, ..., p_n$. Se nel corpo non compaiono variabili allora il goal restituisce "vero" o "falso" se tutti i letterali contenuti in esso sono veri o falsi, altrimenti la query restituisce l'insieme di tutte le combinazioni di variabili che rendono vera la regola.

3.2.1 Esempio

Sia data la regola: $Fratello(X,Y) : -Genitori(Z,X), Genitori(Z,Y), X \neq Y$ "X è fratello di Y **se** Z è genitore di X **e** Z è genitore di Y **e** X è diverso da Y".

La query $? : -Fratello("Carlo", X)$ restituisce la lista di X che rende vero il letterale Fratello("Carlo",X).

3.3 Equivalenza di operazioni

3.3.1 Selezione

E' possibile effettuare una selezione aggiungendo ai letterali di una regola l'operazione di confronto tra una variabile contenuta nel corpo ed un valore costante (o un'altra variabile).

Esempio: Testa(X,Y):-Letterale(X), X < Y (selezione di tutte le tuple con X < Y)

3.3.2 Proiezione

Una proiezione consiste semplicemente nell'elencare nella testa della regola l'insieme degli attributi che si vuole "proiettare".

Esempio: Testa(X,Y,Z):-... (proiezione degli attributi X, Y, Z)

3.3.3 Join

Per effettuare una join tra due o più letterali è sufficiente utilizzare una stessa variabile come parametro dei parametri coinvolti nella join.

*Esempio: Testa(X,Y):-Letterale1(X,Y), Letterale2(X,_) (join tra Letterale1 e Letterale2 sull'attributo X. **nb:** il simbolo don't care indica che del Letterale2 ci interessano tutte le tuple a prescindere dai valori assunti dal secondo valore).*

3.3.4 Unione

Per effettuare l'unione insiemistica del tipo$P = R \cup S$ è necessario ricorrere a più regole con la stessa testa:

- P(X,Y):-R(X,Y)
- P(X,Y):-S(X,Y)

3.3.5 Differenza

Per effettuare la differenza insiemistica $P = R - S$, invece si usa una regola del tipo: $P(X,Y) : -R(X,Y), \neg S(X,Y)$

3.3.6 Negazione

E' possibile usare la versione negata dei letterali mediante l'operatore $\neg$ ma bisogna stare attenti perchè il risultato potrebbe essere indefinito.

Esempio: $q(x) : -\neg p(x)$, $? : -q(x)$ *è indefinito in quanto tutto ciò che non appartiene a p(x) è l'insieme di tutte le tuple con arità 1 che non sono uguali a p(x) (insieme infinito!).*

3.4 Query ricorsive

Una query si dice ricorsiva se all'interno del corpo è presente il letterale della testa:
$Antenato(X,Y) : -Antenato(X,Z), Genitore(Z,Y)$
Inoltre vi è bisogno di una regola che evita che la ricorsione non termini:
$Antenato(X,Y) : -Genitore(X,Y)$