Calcolo Relazionale

Calcolo Relazionale

- Famiglia di **linguaggi dichiarativi** basati sul **calcolo dei predicati del primo ordine**
- Diverse versioni:
  - calcolo **relazionale sui domini**
    - **calcolo sui domini**, in breve
  - calcolo **su n-uple con dichiarazione di *range***
    - **calcolo sulle tuple**, in breve
    - base per il linguaggio SQL

Assunzioni

- I **simboli di predicato** corrispondono alle **relazioni** presenti nella base di dati (più alcuni predicati standard quali uguaglianza e diseguaglianza)
  - Non compaiono simboli di funzione
- Nel calcolo relazionale vengono utilizzate prevalentemente **formule aperte**, cioè formule con variabili libere, il cui valore di verità dipende dai valori assegnati alle variabili libere
  - Il risultato di un'interrogazione (formula aperta) è costituito dalle tuple di valori che, sostituiti alle variabili libere, la rendono vera
- In coerenza con quanto fatto in algebra relazionale (attributi con nome), utilizzeremo una **notazione non posizionale**

Calcolo sui domini

- **Sintassi**: le **espressioni** hanno la forma:

$$\{A_1 : x_1, \ldots, A_k : x_k | f\}$$

- dove:
    - $A_1, \ldots, A_k$ sono attributi distinti (possono anche non comparire nello schema della base di dati)
    - $x_1, \ldots, x_k$ sono variabili (che assumiamo essere distinte, nonostante non sia strettamente necessario)
    - $A_1 : x_1, \ldots, A_k : x_k$ è chiamata ***target list*** (lista degli obiettivi), e descrive il risultato
    - f è una **formula** costruita a partire da formule atomiche utilizzando eventualmente i connettivi Booleani e quantificatori $\exists x$ e $\forall x$, con x variabile

Formule atomiche

- $R(A_1 : x_1, \ldots, A_p : x_p)$, dove $R(A_1, \ldots, A_p)$ è uno **schema di relazione** e $x_1, \ldots, x_p$ sono variabili
  - Interpretabile come $[x_1, \ldots, x_p] \in R$
- x_i OP x_j
  - dove x_i e yx_j sono variabili e OP è un **operatore di confronto** $<\,,\, > =\,,\, \leq\,,\, \geq\,,\, \neq$
- x_i OP c oppure c OP x_i
  - dove c è una costante (nel dominio A_i di x_i)

Formule

- La formule atomiche sono formule
- Se f è una formula, allora anche $\neg f$ lo è
- Se f_1 e f_2 sono formule, allora anche $f_1 \wedge f_2$ lo è
- Se f_1 e f_2 sono formule, allora anche $f_1 \vee f_2$ lo è
- Se f è una formula e x una variabile, allora anche $\exists x(f)$ e $\forall x(f)$ sono formule, dove $\exists$ e $\forall$ sono **quantificatori**
- Per convenienza, useremo laddove necessario le parentesi
- Per convenienza, raggrupperemo le variabili usate nei quantificatori con la medesima formula; per esempio $\exists x(\exists y(f)) \equiv \exists x, y(f)$

Calcolo sui domini

- Il **valore di verità di una formula** è definito nel modo seguente (assumiamo per semplicità che tutti gli attributi abbiano lo stesso dominio):
  - una formula atomica $R(A_1 : x_1, \ldots, A_p : x_p)$ è vera sui valori di $x_1, \ldots, x_p$ che costituiscono una n-upla di R
  - una formula atomica $x\theta y$ $(x\theta c)$ è vera sui valori x e y che rendono vera la condizione θ
  - il valore di verità di $\wedge$, $\vee$ e $\neg$ è definito nel modo usuale
  - una formula della forma $\exists x(f)$ (rispettivamente, $\forall x(f)$) è vera se esiste almeno un elemento del dominio che (rispettivamente, ogni elemento del dominio), sostituito ad x, rende vera f

Calcolo sui domini

- Un'espressione del calcolo sui domini:

$$\{A_1 : x_1, \ldots, A_k : x_k \,|\, f\}$$

- può essere "interpretata" come una **formula logica** del tipo

$$\{x_1, \ldots, x_k \,|\, f(x_1, \ldots, x_k)\}$$

- dove
  - $x_1, \ldots, x_k$ sono **variabili** o **costanti**
  - $f(x_1, \ldots, x_k)$ è un **predicato** che può essere VERO o FALSO
- **Semantica**: il risultato di un'espressione del calcolo sui domini è una relazione su $A_1, \ldots, A_k$ che contiene n-uple di tutti i possibili valori per $x_1, \ldots, x_k$ che rendono vero il predicato $f(x_1, \ldots, x_k)$ rispetto a un'istanza di base di dati a cui l'espressione è applicata

Base di dati per gli esempi

- Impiegato(Matr, Nome, Età, Stipendio)
- Supervisione(Matr, Capo)

Esempio

- Trovare matricola e nome di tutti gli impiegati che guadagnano più di 40

$$\{\text{Matr: } m, \text{ Nome: } n \mid \\ \text{Impiegati(Matr: } m, \text{ Nome: } n, \text{ Età: } e, \text{ Stipendio: } s) \land s > 40\}$$

- La formula "Impiegati(Matr: m, Nome: n, Età: e, Stipendio: s)" ci assicura che le variabili m, n, e, s assumano valori che compaiono nelle n-uple di Impiegati
- La formula "$s > 40$" ci assicura che la variabile s, che assume valori nel dominio Stipendio, assuma solo valori maggiori di 40
- Come risultati ci interessano solo gli attributi matricola e nome

Esempio

- Trovare la matricola degli impiegati con un capo di nome “Luca”

- Ci interessa la matricola m degli impiegati...

  {Matr: m | Impiegati(Matr: m, Nome: n, Età: e, Stipendio: s) ...

- ... per i quali esiste una n-upla in Supervisione...

  ... $\wedge \exists m', c'$(Supervisione(Matr: m', Capo: c') ...

- ...con la stessa matricola e con attributo Capo con valore “Luca”

  ... $\wedge\, m = m' \wedge c' =$ "Luca")}

- Mettendo tutto insieme:

  {Matr: m | Impiegati(Matr: m, Nome: n, Età: e, Stipendio: s) $\wedge$
  $\exists m', c'$(Supervisione(Matr: m', Capo: c') $\wedge\, m = m' \wedge c' =$ "Luca")}

Esempio

- Trovare i nomi dei capi che supervisionano almeno due impiegati

- Procediamo passo passo:

$$\{\text{Capo: } c \mid \text{Supervisione(Matr: } m\text{, Capo: } c) \ldots$$

$$\ldots\ \wedge \exists m', c'(\text{Supervisione(Matr: } m'\text{, Capo: } c') \ldots$$

$$\ldots\ \wedge c = c' \wedge m \neq m')\}$$

Esempio

- Trovare matricola e nome dei capi i cui impiegati guadagnano più di 40

{Matr: c, Nome: n |

Impiegati(Matr: c, Nome: n, Età: e, Stipendio: s) $\wedge$

$\forall m', n', e', s'($

Impiegati(Matr: m', Nome: n', Età: e', Stipendio: s') $\wedge$

Supervisione(Capo: c, Impiegato: m') $\wedge$

$s' > 40$

$)$

}

Calcolo sui domini: discussione

- **Pregi**:
  - Dichiaratività
- **Difetti**:
  - Verbosità (tante variabili!)
  - Possibilità di scrivere espressioni senza senso (**dipendenti dal dominio**)
    - $\{A : x, B : y \mid R(A : x) \land y = y\}$
      - Nel risultato compaiono tuple per qualsiasi valore del dominio di B
      - Se il dominio di B è infinito, il risultato è infinito
      - Se il dominio di B cambia, il risultato cambia (**dipendenza dal dominio**)
    - $\{A : x \mid \neg R(A : x)\}$
      - Nel risultato compaiono tuple per qualsiasi valore del dominio di A che non compaiono in R
  - Nell'algebra tutte le espressioni hanno un senso (**indipendenza dal dominio**)

Indipendenza dal dominio

- **Un'espressione** di un linguaggio di interrogazione si dice **indipendente dal dominio** se il suo risultato, su ciascuna istanza della base di dati, non varia al variare del dominio rispetto al quale l'espressione viene valutata (purché ogni dominio contenga almeno i valori presenti nell'istanza e nell'espressione)
- **Un linguaggio** si dice **indipendente dal dominio** se tali sono tutte le sue espressioni
- Il **calcolo sui domini** non è indipendente dal dominio
- L'**algebra relazionale** è indipendente dal dominio
  - Costruisce i risultati a partire dalle relazioni presenti nella base di dati, senza far mai riferimento ai domini degli attributi: i valori che compaiono nei risultati sono tutti presenti nell'istanza cui l'espressione viene applicata

Calcolo sulle tuple

- Il **calcolo su domini** presenta anche lo svantaggio di richiedere **numerose variabili**, spesso una per ciascun attributo di ciascuna relazione coinvolta (lo stesso con i quantificatori)
- Dal calcolo su domini al **calcolo su tuple**: **le variabili denotano tuple, non singoli valori**
- **Una variabile** per ciascuna **relazione** coinvolta
- Occorre associare una struttura (insieme degli attributi della relazione) a ciascuna variabile che consenta di individuare le singole componenti delle tuple

Calcolo sulle tuple

- Le **espressioni** hanno la forma:

$$\{T|L|f\}$$

- dove:
  - T è una ***target list*** (obiettivi dell'interrogazione)
  - L è una ***range list***
  - f è una **formula**

Target List

- Notazione:
  - x è una variabile
  - X è un insieme di attributi di una relazione
  - Y e Z sono sottoinsiemi di attributi di X di pari lunghezza
- T è una lista di elementi del tipo
  - $Y : x\,.\,Z$
    - Vogliamo solo gli attributi Z della variabile x, che assumerà valori definiti in L (*range list*), e li chiameremo Y
  - $x\,.\,Z \equiv Z : x\,.\,Z$
    - Vogliamo solo gli attributi Z della variabile x, che assumerà valori definiti in L (*range list*), e non li rinomineremo
  - $x\,.\,* \equiv X : x\,.\,X$
    - Vogliamo tutti gli attributi della variabile x, che assumerà valori definiti in L (*range list*)

Range List

- L è una lista che contiene, senza ripetizioni, tutte le variabili della *target list*, con la relazione associata da cui sono prelevati i valori assunti dalla variabile
- In altre parole $L \equiv x_1(R_1), \ldots, x_k(R_k)$
  - x_i è una variabile
  - R_i è una relazione
- L è una **dichiarazione di range**: specifica l'insieme dei valori che possono essere assegnati alle variabili
  - Non occorrono più condizioni atomiche che vincolano una tupla ad appartenere ad una relazione.

Formule Atomiche

- Notazione:
  - x_i indica una variabile, c indica una costante
  - A_i indica un attributo, R indica una relazione
  - OP è un operatore di confronto $<\,,>\,=\,,\leq\,,\geq\,,\neq$
- Formule atomiche:
  - $x_i\,.\,A_i$ OP $x_j\,.\,A_j$
  - $x_i\,.\,A_i$ OP c
  - c OP $x_i\,.\,A_i$

Formule

- La formule atomiche sono formule
- Se f è una formula, allora anche $\neg f$ lo è
- Se f_1 e f_2 sono formule, allora anche $f_1 \wedge f_2$ lo è
- Se f_1 e f_2 sono formule, allora anche $f_1 \vee f_2$ lo è
- Se f è una formula e x una variabile che indica una n-upla su R, allora anche $\exists x(R)(f)$ e $\forall x(R)(f)$ sono formule, dove $\exists$ e $\forall$ sono **quantificatori**
  - Notare che anche i **quantificatori** contengono ora delle **dichiarazioni di range**
  - $\exists x(R)(f)$ significa "esiste nella relazione R una n-upla x che soddisfa la formula f"

Esempio

- Trovare matricola, nome, età e stipendio degli impiegati che guadagnano più di 40

$$\{i\,.\,* \mid i(\text{Impiegati}) \mid i\,.\,\text{Stipendio} \; > 40\}$$

Esempio

- Trovare matricola e nome degli impiegati che guadagnano più di 40

$$\{i\,.\,(\text{Matr, Nome}) \mid i(\text{Impiegati}) \mid i\,.\,\text{Stipendio} \; > 40\}$$

Esempio

- Trovare matricola e nome dei capi i cui impiegati guadagnano più di 40

$$
\begin{gathered}
\{\text{Matr, Nome} : i'.(\text{Matr, Nome}) \mid \\
i'(\text{Impiegati}),\ s(\text{Supervisione}),\ i(\text{Impiegati}) \mid \\
i'.\text{Matr} = s.\text{Capo} \\
\wedge\, s.\text{Impiegato} = i.\text{Matr} \\
\wedge\, i.\text{Stipendio} > 40\}
\end{gathered}
$$

Calcolo sulle n-uple: discussione

- Nel calcolo sulle n-uple le variabili rappresentano tuple quindi si ha **minore verbosità**
- **Alcune interrogazioni** importanti **non si possono esprimere**, in particolare le unioni: $R_1(AB) \cup R_2(AB)$
  - Ogni variabile nel risultato ha un solo *range*, mentre vorremmo n-uple sia della prima relazione che della seconda
  - Intersezione e differenza sono esprimibili
- Per questa ragione SQL (che è basato su questo calcolo) prevede un **operatore esplicito di unione**, ma non tutte le versioni prevedono intersezione e differenza

Calcolo e algebra: limiti

- **Calcolo** e **algebra** sono sostanzialmente **equivalenti**:
  - per ogni espressione del calcolo relazionale che sia indipendente dal dominio esiste un'espressione nell'algebra relazione equivalente a essa
  - per ogni espressione dell'algebra relazionale esiste un'espressione del calcolo relazionale equivalente a essa (e quindi indipendente dal dominio)
- Ci sono però **interrogazioni** interessanti non **esprimibili**:
  - calcolo di **valori derivati**: possiamo solo **estrarre valori**, non calcolarne di nuovi:
    - a livello di n-upla o di singolo valore (conversioni somme, differenze, etc.)
    - su insiemi di n-uple (somme, medie, etc.)
  - interrogazioni **inerentemente ricorsive**, come la **chiusura transitiva**