

Basi di Dati

prof. Letizia Tanca

**Altri linguaggi formali di
interrogazione per
il Modello Relazionale dei Dati**

Classificazione dei Linguaggi

LINGUAGGI **FORMALI**

- **Algebra relazionale**
- **Calcolo relazionale**
 - **Delle tuple**
 - **Dei domini**
- **Datalog**

Calcolo Relazionale

- e' una famiglia di linguaggi
 - formali
 - dichiarativi (non procedurali)
 - per formulare interrogazioni
- due tipi principali
 - calcolo relazionale delle **tuple** (TRC)
 - calcolo relazionale dei **domini** (DRC)
- ha un potere espressivo **pari a quello dell'algebra relazionale**

Definizione Formale del TRC

forma standard

- $\{ t \mid p(t) \}$ (tutte le tuple t per le quali e' vera $p(t)$)
- $p(t)$ e' una formula costruita tramite atomi

ATOMI

- $t \in r$
- $t_1[A_1] \text{ COMP } t_2[A_2]$
- $t[A] \text{ COMP } k$

COMPOSIZIONE DI FORMULE BEN FORMATE (WFF)

- un atomo e' una wff
- se p e' una wff, lo sono anche $\neg p$ e (p)
- se $p1$ e $p2$ sono wff, lo sono anche $p1 \wedge p2$, $p1 \vee p2$, $p1 \Rightarrow p2$
- se p e' una wff in cui s e' una variabile, lo sono anche $\exists s \in r (p(s))$ e $\forall s \in r (p(s))$

Proprieta' del TRC

- $p1 \wedge p2 \equiv \neg(\neg p1 \vee \neg p2)$ (DE MORGAN)
- $\forall t \in r (p(t)) \equiv \neg \exists t \in r (\neg p(t))$
- $p1 \Rightarrow p2 \equiv \neg p1 \vee p2$
- **formule unsafe:**
 - $\{ t \mid t \notin r \}$ FORNISCE UN RISULTATO INFINITO!!!
- **si restringe la formulazione del calcolo delle formule ai domini attivi:**
 - **le tuple che soddisfano una formula possono essere composte solamente da valori che compaiono**
 - **esplicitamente nella formula**
 - **in tuple di relazioni menzionate nella formula**

Esempio di TRC

**NOMI DEGLI STUDENTI CHE HANNO PRESO 30 IN
"matematica"**

$\{ t \mid \exists t1 \in \text{studente}, \exists t2 \in \text{esame}, \exists t3 \in \text{corso}$
 $(t[\text{NOME}] = t1[\text{NOME}] \wedge$
 $t1[\text{MATR}] = t2[\text{MATR}] \wedge$
 $t2[\text{C-CORSO}] = t3[\text{C-CORSO}] \wedge$
 $t2[\text{VOTO}] = 30 \wedge$
 $t3[\text{TITOLO}] = \text{"matematica"}) \}$

Esempio di TRC

**MATRICOLA DEGLI STUDENTI CHE HANNO SOSTENUTO
“matematica” MA NON “basi di dati”**

$$\{ t \mid \exists t1 \in esame, \exists t2 \in corso \\ (t[MATR] = t1[MATR] \wedge \\ t1[C-CORSO] = t2[C-CORSO] \wedge \\ t2 [TITOLO] = \text{“matematica”}) \wedge \\ \neg(\exists t3 \in esame, \exists t4 \in corso \\ (t3[MATR] = t[MATR] \wedge \\ t3[C-CORSO] = t4[C-CORSO] \wedge \\ t4[TITOLO] = \text{“basi di dati”})) \}$$

...e ora proviamo con i presidenti...

Basi di Dati

7

Esempio di schema di base di dati

PRESIDENTI (NOME-P, DATA-N, DATA-M, PARTITO, STATO, NOME-M)
CONGRESSI (# CONGRESSO, %S-REP, %C-REP, %S-DEM, %C-DEM)
AMMINISTRAZIONI (# AMMIN, DATA-IN, VICE-PRES, DATA-N-VP, NOME-P, DATA-N-P)
ELEZIONI (ANNO, VOTI-PRES, NOME-P, DATA-N-PRES, NOME-PERD, DATA-N-PERD, VOTI-PERD)
STATI (STATO, POPOLAZ, # AMMIN.)
PRESID-CONGR (NOME-P, DATA-N, # CONGR)

Basi di Dati

8

Interrogazioni

- *Trovare l'anno di nascita del presidente J.F. Kennedy*
- *Trovare gli anni in cui è stato eletto un presidente repubblicano proveniente dall'Illinois*
- *Trovare i numeri di congressi presieduti dal presidente eletto nel 1955*
- *Trovare i perdenti delle elezioni vinte da qualche presidente di nome Roosevelt*
- *Trovare i nomi delle mogli dei presidenti provenienti dalla California eletti dopo il 1960*
- *Trovare le persone che sono state presidenti OPPURE vicepresidenti in amministrazioni inaugurate dopo il 1880*
- *Trovare le persone che dopo il 1880 sono state almeno una volta presidenti E almeno una volta vicepresidenti*
- *Trovare le persone che sono state presidenti MA MAI vicepresidenti in amministrazioni inaugurate dopo il 1880*

Equivalenza tra Algebra e TRC

FACCIAMO VEDERE CHE SI POSSONO REALIZZARE I 5 OPERATORI FONDAMENTALI (in realtà la prova è più complicata di così...)

- **SELEZIONE**

$$\sigma_{A=1} r \Leftrightarrow \{ t \mid \exists t1 \in r (t1[A] = 1) \wedge t=t1 \}$$

- **PROIEZIONE**

$$\Pi_{AC} r \Leftrightarrow \{ t \mid \exists t1 \in r (t[A,C] = t1[A,C]) \}$$

- **PRODOTTO CARTESIANO**

$$r(A,B,C) \times s(D,E,F) \Leftrightarrow \{ t \mid \exists t1 \in r, \exists t2 \in s (t[A,B,C] = t1[A,B,C] \wedge t[D,E,F] = t2[D,E,F]) \}$$

Equivalenza tra Algebra e TRC

- **UNIONE**

$$r \cup s \Leftrightarrow \{ t \mid \exists t1 \in r, \exists t2 \in s (t=t1 \vee t=t2) \}$$

- **DIFFERENZA**

$$r - s \Leftrightarrow \{ t \mid \exists t1 \in r (t=t1) \wedge \neg (\exists t2 \in s, t=t2) \}$$

- **ESEMPIO DI JOIN**

$$R(A,C) \bowtie_{A=B} S(B,D) \Leftrightarrow \{ t \mid \exists t1 \in r, \exists t2 \in s \\ (t[A,C] = t1[A,C] \wedge t[B,D] = t2[B,D] \wedge t1[A]=t2[B]) \}$$

Linguaggi di query basati sulla programmazione logica

- La programmazione logica è un paradigma di programmazione basato su regole
- Linguaggio di riferimento: Prolog (1970)
- **Datalog**: “Prolog per basi di dati” (1984)
- Differenze principali rispetto a Prolog (per chi lo conosce):
 - niente simboli di funzione
 - modello di valutazione non procedurale (non SLD Resolution)

Regole Datalog

- Ogni regola è composta da una testa (head o LHS) e da un corpo (body o RHS)

$$p \text{ :- } p1, p2, \dots pn$$

- Ogni P rappresenta un predicato (chiamato letterale), così composto:
 - nome
 - argomenti:
 - **costanti**
 - **variabili**
 - **simbolo “don’t care” (_) (non nella testa)**
- Le variabili del LHS devono apparire nel RHS

Alcune regole

- $Dipendente(X,Y) \text{ :- } Persona(X), Assunto(X,Y,D), Ditta(Y), D < "9.11.2005"$.
- $Parte_di(X,Z) \text{ :- } Prodotto(X), Prodotto(Y), Prodotto(Z), Parte_di(X,Y), Parte_di(Y,Z)$.
- $Ospite(X,Y) \text{ :- } Persona(X), Luogo_di_Soggiorno(Y), Dorme(X,Y)$.

Base di dati d'esempio per Datalog

GENITORI

Genitore	Figlio
Carlo	Antonio
Carlo	Gianni
Anna	Antonio
Anna	Gianni
Gianni	Andrea
Antonio	Paola

PERSONA

Nome	Età	Sesso
Carlo	65	M
Antonio	40	M
Anna	60	F
Gianni	43	M
Andrea	22	M
Paola	20	F

Basi di Dati

15

Regole Datalog

- Ciascuna tupla corrisponde a un **fatto** (letterale *ground*). Ad esempio:

Genitori("Carlo","Antonio").

rappresenta la prima tupla della relazione
GENITORI

- Esempio di regola:

Padre(X,Y):- Persona(X,_, 'M'), Genitori(X,Y).

Basi di Dati

16

Unificazione

- **Interpretazione** della regola:
 - *LHS è vero se RHS è vero*
 - *RHS è vero se tutte le variabili presenti nei letterali sono unificabili, ovvero sostituibili, con valori costanti che rendono vero il letterale*
- **Persona(X , $_$, $'M'$)**: le unificazioni possibili per X sono:
{“Carlo”, “Antonio”, “Gianni”, “Andrea”}
- **Genitori(X , Y)**: le unificazioni possibili per la coppia X , Y sono: { (“Carlo”, “Antonio”), etc ...} (**l'intero contenuto di GENITORI**)

Basi di Dati

17

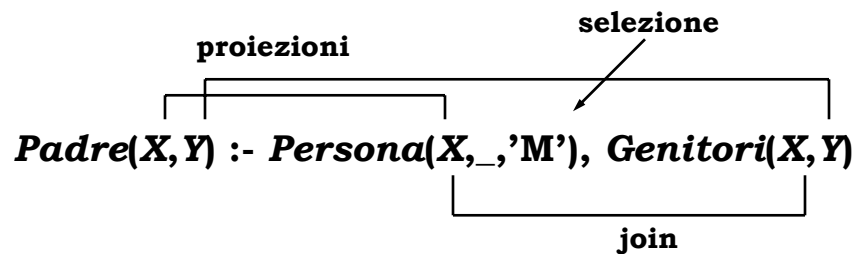
Interpretazione delle regole Datalog

- Le sostituzioni per X che vanno bene sono
{ “Carlo”, “Gianni”, “Antonio” }
- Per le variabili X e Y si ottengono le seguenti coppie di sostituzioni
{ (“Carlo”, “Antonio”), (“Carlo”, “Gianni”), (“Gianni”, “Andrea”), (“Antonio”, “Paola”) }

Basi di Dati

18

Corrispondenza tra Datalog e l'algebra relazionale



- Espressione corrispondente in algebra relazionale:

$$\text{PADRE} = \Pi_{1,5} \sigma_{3 = 'M'} (\text{PERSONA} \bowtie_{1=1} \text{GENITORI})$$

Database estensionale e intensionale

- Database estensionale (EDB): insieme delle tabelle presenti nel DB
- Database intensionale (IDB): insieme dei predicati che sono a sinistra in una regola

➤ ***è la conoscenza dedotta a partire da EDB***

- Normalmente si impone $\text{EDB} \cap \text{IDB} = \emptyset$

Query in Datalog

- Si esprimono come **“goal”** :
?- Padre(“Carlo”,X)
 - Valutazione: si cerca una regola che definisca *Padre* e una sostituzione che unifichi con la variabile *X*
 - Si ottiene $X = \{\text{“Antonio”, “Gianni”}\}$
- NOTA:** Un goal senza variabili restituisce *True* o *False*
- ?- Padre(“Carlo”, “Antonio”) \Rightarrow *True*
 - ?- Padre(“Carlo”, “Andrea”) \Rightarrow *False*

Basi di Dati

21

Altre regole sulla stessa BD

- $Madre(X, Y) :- Persona(X, _ , 'F'), Genitori(X, Y).$
- $Nonno(X, Z) :- Genitori(X, Y), Genitori(Y, Z).$
- $Fratello(X, Y) :- Genitori(Z, X), Genitori(Z, Y), X \neq Y.$
- $Zio(X, Y) :- Persona(X, _ , 'M'), Fratello(X, Z), Genitori(Z, Y).$

Basi di Dati

22

Negazione in Datalog

- Alcuni letterali del corpo possono essere negati
 - L'uso della negazione in Datalog aumenta il potere espressivo del linguaggio
 - Anche l'uso della negazione richiede cautela:
 - $q(\mathbf{X}) :- \neg p(\mathbf{X}).$
 - $p(\mathbf{0}).$ e tutti gli altri sono "non P"???
 - ?- $q(\mathbf{X})$ produce un risultato infinito!
- Ancora una volta si rischiano interrogazioni UNSAFE

Basi di Dati

23

Potere espressivo di Datalog

- Datalog senza negazione permette di rappresentare gli operatori $\{\sigma, \Pi, \times, \cup\}$
- $\{\sigma, \Pi, \times\}$ già visti
- Per l'unione si usano più regole con la stessa testa; $P = R \cup S$:
 - $P(X, Y) :- R(X, Y).$
 - $P(X, Y) :- S(X, Y).$
- Per la differenza serve il *not*; $P = R - S$:
 - $P(X, Y) :- R(X, Y), \neg S(X, Y)$
- La negazione VA TENUTA SOTTO CONTROLLO

Basi di Dati

24

Esempio di schema di base di dati

PRESIDENTI (NOME-P, DATA-N, DATA-M, PARTITO, STATO, NOME-M)
CONGRESSI (# CONGRESSO, %S-REP, %C-REP, %S-DEM, %C-DEM)
AMMINISTRAZIONI (# AMMIN, DATA-IN, VICE-PRES, DATA-N-VP, NOME-P, DATA-N-P)
ELEZIONI (ANNO, VOTI-PRES, NOME-P, DATA-N-PRES, NOME-PERD, DATA-N-PERD, VOTI-PERD)
STATI (STATO, POPOLAZ, # AMMIN.)
PRESID-CONGR (NOME-P, DATA-N, # CONGR)

Interrogazioni

- *Trovare l'anno di nascita del presidente J.F. Kennedy*
- *Trovare gli anni in cui è stato eletto un presidente repubblicano proveniente dall'Illinois*
- *Trovare i numeri di congressi presieduti dal presidente eletto nel 1955*
- *Trovare i perdenti delle elezioni vinte da qualche presidente di nome Roosevelt*
- *Trovare i nomi delle mogli dei presidenti provenienti dalla California eletti dopo il 1960*
- *Trovare le persone che sono state presidenti OPPURE vicepresidenti in amministrazioni inaugurate dopo il 1880*
- *Trovare le persone che sono state presidenti E ANCHE vicepresidenti in qualche amministrazione inaugurata dopo il 1880*
- *Trovare le persone che sono state presidenti MA MAI vicepresidenti in amministrazioni inaugurate dopo il 1880*

Query ricorsive

- Datalog con negazione ha quindi un potere espressivo almeno pari all'algebra relazionale
- In effetti ha un potere superiore, perché permette l'espressione di query ricorsive
- Una query ricorsiva (immediata) presenta il letterale della testa all'interno del corpo della regola:

Antenato(X, Y) :- Genitore(X, Y).

Antenato(X, Y) :- Antenato(X, Z), Genitore(Z, Y).

Basi di Dati

27

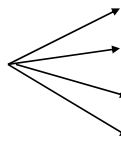
Valutazione delle query ricorsive

- Nella base dati d'esempio si ottiene il risultato illustrato a fianco

ANTENATO

Carlo	Antonio
Carlo	Gianni
Anna	Antonio
Anna	Gianni
Gianni	Andrea
Antonio	Paola
Carlo	Andrea
Carlo	Paola
Anna	Andrea
Anna	Paola

**Nuovi elementi
rispetto a
GENITORE**



Basi di Dati

28

Meccanismo di valutazione

Si può immaginare che venga seguito il seguente processo iterativo:

$$\begin{aligned} \text{ANTENATO}^0 &\Leftarrow \text{GENITORE} \\ \text{ANTENATO}^1 &\Leftarrow (\Pi_{1,4} \text{ANTENATO}^0 \triangleright \triangleleft_{2=1} \text{GENITORE}) \cup \\ &\quad \text{ANTENATO}^0 \\ \text{ANTENATO}^2 &\Leftarrow (\Pi_{1,4} \text{ANTENATO}^1 \triangleright \triangleleft_{2=1} \text{GENITORE}) \cup \\ &\quad \text{ANTENATO}^1 \\ &\dots \end{aligned}$$

fino a che ANTENATO^n risulta pari a
 ANTENATO^{n-1} (**punto fisso**)

Basi di Dati

29

Predicati e funzioni

- Nella definizione delle regole si possono usare predicati speciali
 - operatori di confronto:
= (unificazione), \neq , $<$, \leq , $>$, \geq
 - funzioni aritmetiche:
 $+$, $-$, $*$, \div
- Bisogna fare molta attenzione:
 $n(\mathbf{X}) :- n(\mathbf{X} - 1).$
 $n(0).$
?- $n(\mathbf{X})$ produce un risultato infinito!

Basi di Dati

30

Regole corrette

Le regole devono essere *ben formate*:

- La negazione deve essere *safe*:
tutte le variabili di un letterale negato devono comparire in un letterale positivo del corpo della regola: es. **$S(X) :- \neg R(X)$** non va bene
- La negazione deve essere *stratificata*:
in sintesi, non ci devono essere cicli di dipendenza tra letterali negati. Ad es.:

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

$R(Y) :- T(X, Y), \neg P(Y)$ non va bene

Basi di Dati

31

Usi di Datalog

Datalog costituisce un linguaggio per la definizione di viste e di vincoli

Esempi di vincoli:

$incorrectdb() :- Madre(X,Z), Madre(Y,Z), X \neq Y.$

$incorrectdb() :- Genitori(X,Z), \neg Persona(X,_,_).$

$incorrectdb() :- Antenato(X,X).$

Basi di Dati

32

Riepilogo della terminologia Datalog

Relazione	→	Predicato
Attributo	→	Argomento
Tupla	→	Fatto
Vista	→	Regola
Interrogazione	→	Goal

Altri programmi logici

- **Cugini della stessa generazione**
 - $sgc(X, X)$.
 - $sgc(X, Y) :- genitore(X, Z), sgc(Z, W), genitore(Y, W)$.
- **Supervisori e subordinati**
 - $supervisore(X, Y) :- dirige(X, Y)$.
 - $supervisore(X, Y) :- dirige(X, Z), supervisore(Z, Y)$.
 - $subordinato(X, Y) :- supervisore(Y, X)$.
 - $incorrectdb(X) :- supervisore(X, X)$.
 - $incorrectdb(X) :- dirige(X, Y), dirige(Z, Y), X \neq Z$.
- **Generazione della distinta base**
 - $componente(X, Y) :- parte_di(X, Y)$.
 - $componente(X, Y) :- parte_di(X, Z), componente(Z, Y)$.

Esercizi (A)

CLIENTE(CodiceFiscale, Cognome, Nome, DataNascita, Sesso, Nazionalità)

ALBERGO(PartitaIVA, NomeAlbergo, Località, NomeProprietario)

SERVIZI OFFERTI(PIVAAlbergo, NomeServizio)

PRENOTAZIONE(PIVAAlbergo, CFCliente, Datainizio, NCamera, Tariffagiornaliera, Datafine)

1. Estrarre il nome e il cognome dei clienti che non hanno mai prenotato camere nell'albergo "Del Sole" di Torino.
2. Estrarre la partita IVA di ogni albergo il cui proprietario ha anche un albergo in una diversa località.
3. Estrarre i clienti che hanno prenotato un albergo di Torino per un periodo superiore a un mese oppure hanno prenotato un albergo che offre la sauna in una città qualsiasi.

Basi di Dati

35

Esercizi (B)

LABORATORIO(Numero, NumeroPosti)

PRENOTAZIONE(NumeroLab, Data, OraInizio, OraFine, DocTitoloCorso, Esercitatore)

DOCENTE(NomeDocente, Dipartimento, Posizione, Materia, #prenot)

1. Trovare i laboratori che sono stati prenotati da un docente ordinario di "Basi di dati", ma che non hanno alcuna prenotazione relativa ad un docente del "Dipartimento di Ingegneria dei Sistemi".
2. Trovare il nome e il dipartimento del docente che ha effettuato il maggior numero di prenotazioni
3. Trovare le informazioni dei laboratori che non sono mai stati prenotati né da un "professore ordinario" né da un esercitatore del dipartimento "DEI"

Basi di Dati

36