

## Datalog

## 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):

- mancano i simboli di funzione
- modello di valutazione pienamente dichiarativo

2

## Base di dati d'esempio per Datalog

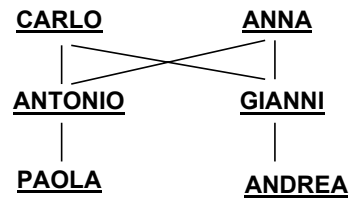
### GENITORE

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

3



4

## Regole Datalog

- Ogni regola è composta da una testa (head o LHS) e da un corpo (body o RHS)  
ES:  $P :- P_1, P_2, \dots, P_n$
- Ogni  $P$  rappresenta un predicato (chiamato letterale), così composto:
  - nome
  - argomenti:
    - costanti
    - variabili
    - simbolo "don't care" (non nella testa)
 ES:  $\text{persona}(X, \_ , "M")$

5

## Regole Datalog

Hanno l'interpretazione che la testa è vera se il corpo è vero

Esempi di regole

$\text{Padre}(X, Y) :- \text{Persona}(X, \_ , 'M'), \text{Genitore}(X, Y)$   
 $\text{Madre}(X, Y) :- \text{Persona}(X, \_ , 'F'), \text{Genitore}(X, Y)$

Le variabili del LHS devono apparire nel RHS

6

## Fatti Datalog

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

*Genitore*("Carlo", "Antonio").

7

## Query in Datalog

- Esempio di query (detta *goal*)  
?- *Padre*("Carlo", X)
- Valutazione: si cerca una regola che abbia per testa il predicato *Padre* e si cercano valori per la variabile X (*unificazione*)  
Si ottiene  $X = \{\text{"Antonio"} \text{ "Gianni"}\}$
- Un goal senza variabili restituisce *True* o *False*
  - ?- *Padre*("Carlo", "Antonio")  $\Rightarrow$  *True*
  - ?- *Padre*("Carlo", "Andrea")  $\Rightarrow$  *False*

8

## Altre regole

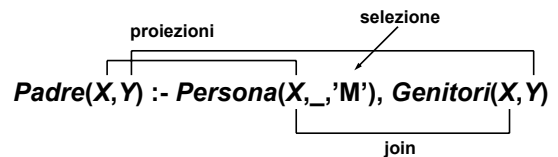
*Nonno*(X,Z) :- *Padre*(X,Y), *Genitore*(Y,Z)

*Fratello*(X,Y) :- *Genitore*(Z,X), *Genitore*(Z,Y),  $X \neq Y$

*Zio*(X,Y) :- *Persona*(X,\_, 'M'), *Fratello*(X,Z),  
*Genitore*(Z,Y)

9

## Corrispondenza tra Datalog e l'algebra relazionale



- Espressione corrispondente in algebra relazionale:

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

10

## 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  $EDB \cap IDB = \emptyset$

11

## Negazione in Datalog

- Alcuni letterali del corpo possono essere negati  
es: *Padre*(X,Y) :- *Genitore*(X,Y), not *Madre*(X,Y)
- L'uso della negazione in Datalog aumenta il potere espressivo del linguaggio
- Anche l'uso della negazione richiede cautela:
  - $q(X) :- \neg p(X)$
  - $p(0).$
  - ?-  $q(X)$  produce un risultato infinito!

12

## 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;  $P = R - S$ :  
 $P(X, Y) :- R(X, Y), \neg S(X, Y)$

13

## 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

14

## Query ricorsive

- Una query ricorsiva presenta il letterale della testa all'interno del corpo della regola:
- Antenato(X, Y) :- Genitore(X, Y) ← start  
Antenato(X, Y) :- Antenato(X, Z), Genitore(Z, Y) ← ricorsione

15

## Meccanismo di valutazione

- Si può immaginare che venga seguito il seguente processo iterativo:  
 $ANTENATO^0 \Leftarrow GENITORI$   
 $ANTENATO^1 \Leftarrow (\Pi_{1,4} ANTENATO^0 \triangleright_{2=1} GENITORI) \cup ANTENATO^0$   
 $ANTENATO^2 \Leftarrow (\Pi_{1,4} ANTENATO^1 \triangleright_{2=1} GENITORI) \cup ANTENATO^1$   
 ...  
 fino a che  $ANTENATO^n$  risulta pari a  $ANTENATO^{n-1}$  (punto fisso)

16

## Valutazione delle query ricorsive

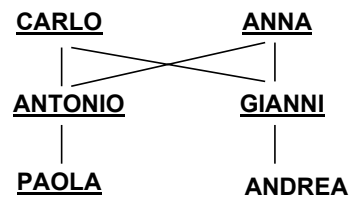
- Nella base dati d'esempio si ottiene il risultato illustrato a fianco

Nuovi elementi rispetto a GENITORE

ANTENATO

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

17



18

## 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:

$n(X) :- n(X - 1)$

$n(0) .$

?-  $n(X)$  produce un risultato infinito!

19

## Regole corrette

• La negazione deve essere *safe*

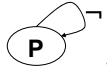
- tutte le variabili di un letterale negato devono comparire in un letterale positivo del corpo della regola

$S(X, Y) :- R(X), \neg Q(Y)$  non va bene

• La negazione deve essere *stratificata*

- sinteticamente, non ci devono essere cicli di dipendenza tra letterali negati

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



20

## Riepilogo della terminologia Datalog

Modello  
relazionale  
standard

Datalog

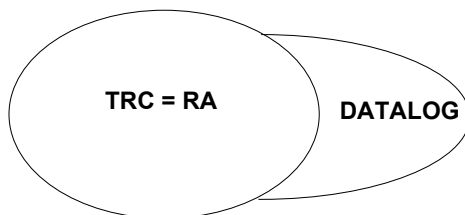
Relazione	Predicato
Attributo	Argomento
Tupla	Fatto
Query	Regole
Interrogazione	Goal

21

## Sintesi sui poteri espressivi

- $TRC = RA$ , con TRC safe e non dipendente dal dominio
- Datalog più espressivo di TRC e RA (a causa della ricorsione), con espressioni Datalog che siano corrette, e quindi:
  - safe
  - stratificate
  - che non producano infiniti simboli.

22



23

## In che modo si differenziano i linguaggi formali?

- L'algebra è un linguaggio PROCEDURALE: si dice esattamente in che modo valutare le interrogazioni (e si può ottimizzare)
- TRC è un linguaggio DICHIARATIVO: si dice cosa si vuole ottenere ma non come ottenerlo.
- DATALOG è pure dichiarativo

24

## Perchè vi abbiamo raccontato tutto ciò?

- SQL assomiglia al calcolo relazionale
- L'algebra assomiglia a quanto viene effettivamente valutato da un DBMS
- Datalog avvicina le basi di dati al mondo della logica e della deduzioni