Introduction to Data Management CSE 344

Lecture 10: Datalog

Datalog

- Very friendly notation for queries
- Initially designed for *recursive* queries
- Some companies offer datalog implementation for data analytics, e.g. LogicBlox
- We discuss only *recursion-free* or *non-recursive* datalog, and add negation

Datalog

How to try out datalog quickly:

- Download DLV from http://www.dbai.tuwien.ac.at/proj/dlv/
- Run DLV on this file:

```
parent(william, john).
parent(john, james).
parent(james, bill).
parent(sue, bill).
parent(james, carol).
parent(sue, carol).

male(john).
male(james).
female(sue).
male(bill).
female(carol).

grandparent(X, Y) :- parent(X, Z), parent(Z, Y).
father(X, Y) :- parent(X, Y), male(X).
mother(X, Y) :- parent(X, Y), female(X).
brother(X, Y) :- parent(P, X), parent(P, Y), male(X), X != Y.
sister(X, Y) :- parent(P, X), parent(P, Y), female(X), X != Y.
```


Datalog: Facts and Rules

Facts = tuples in the database

```
Actor(344759,'Douglas', 'Fowley').
Casts(344759, 29851).
Casts(355713, 29000).
Movie(7909, 'A Night in Armour', 1910).
Movie(29000, 'Arizona', 1940).
Movie(29445, 'Ave Maria', 1940).
```

Rules = queries

```
Q1(y) :- Movie(x,y,z), z='1940'.
```

Find Movies made in 1940

Datalog: Facts and Rules

Facts = tuples in the database

```
Actor(344759,'Douglas', 'Fowley').
Casts(344759, 29851).
Casts(355713, 29000).
Movie(7909, 'A Night in Armour', 1910).
Movie(29000, 'Arizona', 1940).
Movie(29445, 'Ave Maria', 1940).
```

Rules = queries

```
Q1(y) :- Movie(x,y,z), z='1940'.

Q2(f, l) :- Actor(z,f,l), Casts(z,x),
            Movie(x,y,'1940').
```

Find Actors who acted in Movies made in 1940

Datalog: Facts and Rules

Facts = tuples in the database

```
Actor(344759,'Douglas', 'Fowley').
Casts(344759, 29851).
Casts(355713, 29000).
Movie(7909, 'A Night in Armour', 1910).
Movie(29000, 'Arizona', 1940).
Movie(29445, 'Ave Maria', 1940).
```

Rules = queries

```
Q1(y) :- Movie(x,y,z), z='1940'.

Q2(f, l) :- Actor(z,f,l), Casts(z,x),
            Movie(x,y,'1940').

Q3(f,l) :- Actor(z,f,l), Casts(z,x1), Movie(x1,y1,1910),
           Casts(z,x2), Movie(x2,y2,1940)
```

Find Actors who acted in a Movie in 1940 and in one in 1910

Datalog: Facts and Rules

Facts = tuples in the database

```
Actor(344759,'Douglas', 'Fowley').
Casts(344759, 29851).
Casts(355713, 29000).
Movie(7909, 'A Night in Armour', 1910).
Movie(29000, 'Arizona', 1940).
Movie(29445, 'Ave Maria', 1940).
```

Rules = queries

```
Q1(y) :- Movie(x,y,z), z='1940'.
```

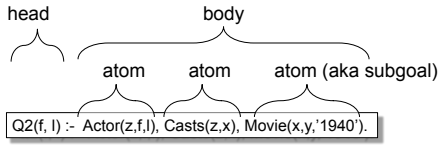
```
Q2(f, l) :- Actor(z,f,l), Casts(z,x),
            Movie(x,y,'1940').
```

```
Q3(f,l) :- Actor(z,f,l), Casts(z,x1), Movie(x1,y1,1910),
           Casts(z,x2), Movie(x2,y2,1940)
```

Extensional Database Predicates = EDB = Actor, Casts, Movie

Intensional Database Predicates = IDB = Q1, Q2, Q3

Datalog: Terminology



f, l = head variables

x,y,z = existential variables

More Datalog Terminology

```
Q(args) :- R1(args), R2(args), ....
```

- Ri(argsi) is also called a relational atom
- Ri(argsi) evaluates to true when relation Ri contains the tuple described by argsi.
  - Example: Actor(344759,'Douglas', 'Fowley') is true
- In addition to relational atoms, we can also have arithmetic atoms
  - Example: z='1940'.

Semantics

- Meaning of a datalog rule = a logical statement !

```
Q1(y) :- Movie(x,y,z), z='1940'.
```

- Means:
  - $\forall x.\ \forall y.\ \forall z.$ (Movie(x,y,z) and z='1940' $\Rightarrow$ Q1(y))
  - and Q1 is the smallest relation that has this property
- Note: logically equivalent to:
  - $\forall\ y.\ (\exists x.\ \exists\ z.$ Movie(x,y,z) and z='1940' $\Rightarrow$ Q1(y))
  - That's why vars not in head are called "existential variables".

Datalog program

A datalog program is a collection of one or more rules

Each rule expresses the idea that from certain combinations of tuples in certain relations, we may infer that some other tuple must be in some other relation or in the query answer

Exampe: Find all actors with Bacon number ≤ 2

```
B0(x) :- Actor(x,'Kevin', 'Bacon')
B1(x) :- Actor(x,f,l), Casts(x,z), Casts(y,z), B0(y)
B2(x) :- Actor(x,f,l), Casts(x,z), Casts(y,z), B1(y)
Q4(x) :- B1(x)
Q4(x) :- B2(x)
```

Note: Q4 is the *union* of B1 and B2

Non-recursive Datalog

- In datalog, rules can be recursive
  - Path(x, y) :- Edge(x, y).
  - Path(x, y) :- Path(x, z), Edge (z, y).
- We focus only on non-recursive datalog

Datalog with negation

Find all actors with Bacon number ≥ 2

```
B0(x) :- Actor(x,'Kevin', 'Bacon')
B1(x) :- Actor(x,f,l), Casts(x,z), Casts(y,z), B0(y)
Q6(x) :- Actor(x,f,l), not B1(x), not B0(x)
```


Safe Datalog Rules

Here are ***unsafe*** datalog rules. What's "unsafe" about them ?

```
U1(x,y) :- Movie(x,z,1994), y>1910
```

```
U2(x)  :- Movie(x,z,1994), not Casts(u,x)
```

A datalog rule is ***safe*** if every variable appears in some positive relational atom

Datalog v.s. Relational Algebra

- Every expression in the basic relational algebra can be expressed as a Datalog query
- But operations in the extended relational algebra (grouping, aggregation, and sorting) have no corresponding features in the version of datalog that we discussed today
- Similarly, datalog can express recursion, which relational algebra cannot

Examples

Schema for our examples
R(A,B,C)
S(D,E,F)
T(G,H)

Examples

Union R(A,B,C) U S(D,E,F)

U(x,y,z) :- R(x,y,z)
U(x,y,z) :- S(x,y,z)

Examples

Intersection

I(x,y,z) :- R(x,y,z), S(x,y,z)

Examples

Selection: $\sigma_{x>100 \text{ and } y='some\ string'}(R)$
L(x,y,z) :- R(x,y,z), x > 100, y='some string'

Selection x>100 **or** y='some string'
L(x,y,z) :- R(x,y,z), x > 100
L(x,y,z) :- R(x,y,z), y='some string'

Examples

Equi-join: $R \bowtie_{R.A=S.D \text{ and } R.B=S.E} S$

J(x,y,z,q) :- R(x,y,z), S(x,y,q)

Examples

Projection

P(x) :- R(x,y,z)

Examples

To express difference, we add negation

D(x,y,z) :- R(x,y,z) NOT S(x,y,z)

More Examples

R(A,B,C)
S(D,E,F)
T(G,H)

Translate: $\Pi_A(\sigma_{B=3}(R))$
A(a) :- R(a,3,_)
Underscore used to denote an "anonymous variable", a variable that appears only once.

More Examples

R(A,B,C)
S(D,E,F)
T(G,H)

Translate: $\Pi_A(\sigma_{B=3}(R) \bowtie_{R.A=S.D} \sigma_{E=5}(S))$
A(a) :- R(a,3,_), S(a,5,_)