Datalog

Susan B. Davidson
CIS 700: Advanced Topics in Databases
MW 1:30-3
Towne 309

http://www.cis.upenn.edu/~susan/cis700/homepage.html
References

• Textbooks
  • Ramakrishnan and Gehrke, Ch 24
  • Ullman (“Principles of Database and Knowledge-Base Systems: Vol 1”), Ch 3
  • Abiteboul, Hull and Vianu (“Foundations of Databases”), Ch. 12, 13:1-3, 15:1-3
  • Phokion Kolaitis’ [tutorial] on database theory at Simon’s
Homework for next week (1/24)

• Read and write a summary on one of the following two papers:
  • Joe Hellerstein, “The Declarative Imperative,” SIGMOD Record 2010
  • Afrati and Ullman, “Transitive Closure and Recursive Datalog Implemented on Clusters” EDBT2012

• What is a summary (print and bring to class)?
  • Short paragraph describing paper
  • 1-3 “strengths”, 1-3 “weaknesses”
  • At least one question you have about the paper.
What is Datalog?

• Logic-based data model designed for recursive queries.
  • “Prolog for Databases”

• Introduced by Chandra and Harel in 1982 and has been widely studied by the research community.

• Modern implementations: commercial (LogicBlox, Datomic), networking (Overlog), programming languages,...

• SQL:1999 and subsequent versions of the SQL standard provide support for linear Datalog.

• We will cover the syntax, semantics, and how to evaluate
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
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
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')
Q2(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 Movies in 1940 and in one in 1910.
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')
Q2(f, l):- Actor(z, f, l), Casts(z, x1), Movie(x1, y1, '1910'), Casts(z, x1), Movie(x1, y1, '1910')

Extensional Database Predicates = EDB (Actor, Casts, Movie)
Intensional Database Predicates = IDB (Q1, Q2, Q3)
Terminology

The head is true if all the subgoals are true.

f, l = head variables
x, y, z = existential variables

“The head is true if all the subgoals are true.”
Safe Datalog Rule

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

Q10(d,ba):- Likes(d,be), Serves(ba,be), not Freq(d,ba)

What is this query asking?
A Datalog rule is **safe** if every variable appears in some positive relational atom.

Here are some **unsafe** Datalog rules. What is “unsafe” about them?

U1(x,y):- Movie(x,z,’1940’), y>‘1910’

Q1(x):- Movie(x,z,’1940’), not Casts(u,x)
Some examples

• Write queries for the following
  • Names of all beers.
  • Names of all beers that Chris likes.
  • Drinkers who frequent at least one bar that serves a beer they like.
  • Drinkers who frequent no bars.
  • Drinkers for whom every bar that they frequent serves at least one beer that they like (and they frequent at least one bar).
  • Drinkers for whom no bar that they frequent serves a beer that they like (and they frequent at least one bar).
The Bachelor problem

Suppose we have an EDB relation married(x,y) and want to calculate the bachelors.

Is this correct?

```
bachelor(Y) :- NOT married(X,Y)
```
The Bachelor problem

Suppose we have an EDB relation married(x,y) and want to calculate the bachelors.

Is this correct?

\[
\text{bachelor}(y) :- \text{NOT married}(x,y)
\]

Is this correct?

\[
\text{bachelor}(y) :- \text{person}(x), \text{NOT married}(x,y)
\]
Datalog versus SQL

• Non-recursive Datalog with negation is a cleaned-up core of SQL
  • Unions of conjunctive queries
  • Forms the core of query optimization, what we know how to reason over easily
• You can translate easily between non-recursive Datalog with negation and SQL.
  • Take the join of the nonnegated, relational subgoals and select/delete from there.
Next time: evaluating Datalog+