Recursive Queries and Datalog

Refer to Jeff Ullman's slides at:
http://infolab.stanford.edu/~ullman/cs345notes/cs345-1.ppt

An example from the past

Return all grandparents and their grand children

ancestor / descendant

Actors

Name	Age	Addr	Parent
PC	36	Mumbai	Madhu
AH	81	LA	Muriel
BN	69	LA	Catherine
AB	42	Mumbai	Jaya
Jaya	63	Mumbai	Indira

Actors_1

Name	Age	Addr	Parent
PC	36	Mumbai	Madhu
AH	81	LA	Muriel
BN	69	LA	Catherine
AB	42	Mumbai	Jaya
Jaya	63	Mumbai	Indira

```
SELECT *
FROM Actors AS Actors1, Actors AS Actors2
WHERE Actors1.Parent = Actors2.Name
```

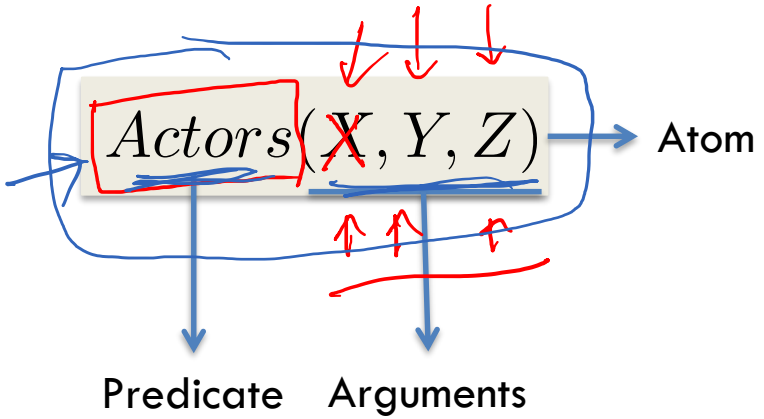
Datalog

- "A logic for relations"
- Consists of if-then rules
- Equivalent to relational algebra in its non-recursive form
- Can form the basis for writing recursive queries in SQL

Predicates

Actors

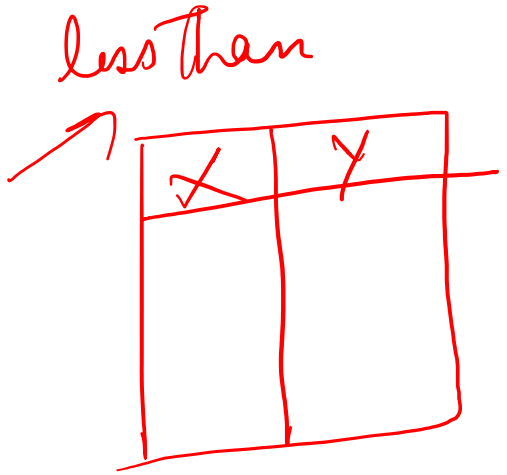
Name	Age	Addr
Anthony Hopkins	81	LA
Bill Nighy	69	LA
Abhishek Bachchan	42	Mumbai

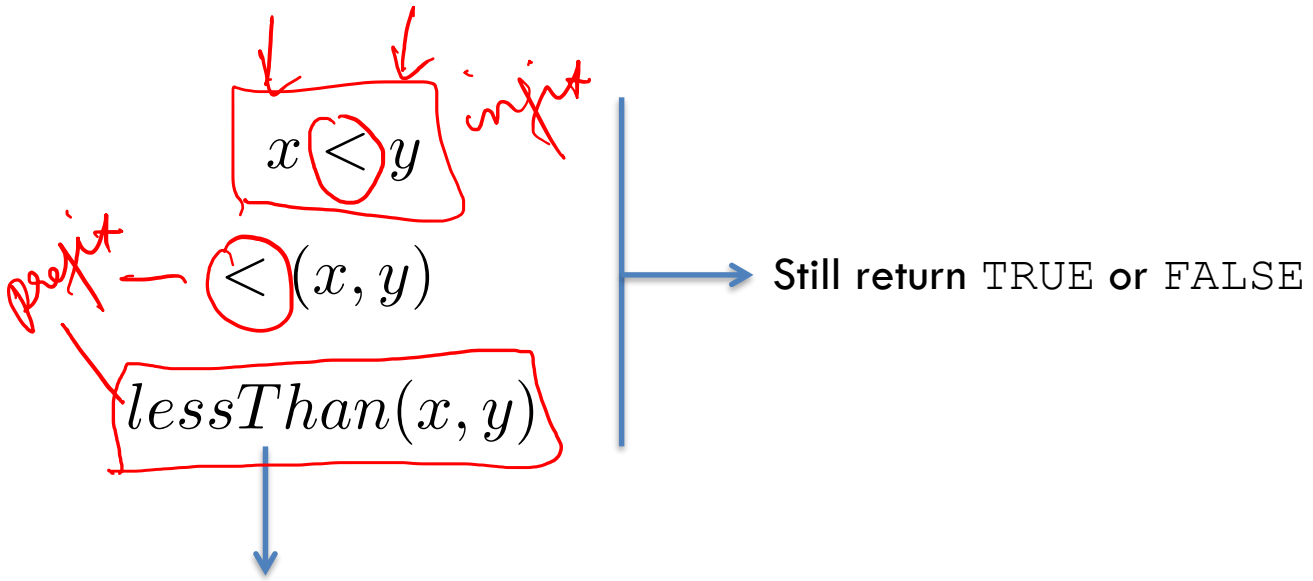


Predicates return `TRUE` or `FALSE`

```
Actors ('Anthony Hopkins', 81, 'LA')
Actors ('Bill Nighy', 49, 'LA')
Actors ('Abhishek Bachchan', Y, Z)
```

Arithmetic Atoms





Still return `TRUE` or `FALSE`

Can be regarded as a relation consisting of two numbers

Datalog Rules

predicates

$$P(X_1, X_2, \ldots) \leftarrow P_1(Y, \ldots) \& P_2(Z, \ldots) \& \ldots$$

Head

Subgoal

Body

If each subgoal is true, then the head is true

$P(X_1, X_2, \ldots) \leftarrow P_1(Y, \ldots)$ AND $P_2(Z, \ldots)$ AND ...

Datalog queries (1/3)

Actors

Name	Age	Addr
Priyanka Chopra	36	Mumbai
Anthony Hopkins	81	LA
Bill Nighy	68	LA
Abhishek Bachchan	42	Mumbai

Return all actors whose age is more than 35.

$$\pi_{Name}(\sigma_{Age>35}(Actors))$$

$$Over35(x) \leftarrow Actors(x, y, z) \text{AND } y > 35$$

Datalog queries (2/3)

Actors

Name	Age	Addr
PC	36	Mumbai
TC	55	LA
BN	69	LA
AB	42	Mumbai

Movies

Name	Year	Title
PC	2011	Don-II
TC	2011	MI-IV
BN	2009	Valkyrie
AB	2010	Raavan

Return all information about actors and their movies

$$Actors \bowtie_{A.Name=M.Name} Movies$$

$$Result(n, a, d, y, t) \leftarrow Actors(n, a, d) \text{ AND } Movies(n, y, t)$$

Actors (PC, 36, M) AND Movies (PC, 2011, Don-II)

Result (PC, 36, ABC, 1999, AB) Actors (PC, 36, ABC) AND Movies (PC, 1999, AB)

Datalog queries (3/3)

Actors

Name	Age	Addr
Priyanka Chopra	36	Mumbai
Anthony Hopkins	81	LA
Bill Nighy	69	LA
Abhishek Bachchan	42	Mumbai

Movies

Name	Year	Title
Priyanka Chopra	2011	Don-II
Anthony Hopkins	2011	Thor: R
Bill Nighy	2009	Valkyrie
Abhishek Bachchan	2010	Raavan

Return the names of actors below the age of 50 who have acted in a movie in 2011

$$\Pi_{Name}(\sigma_{Age<50 \text{ AND } Year=2011}(Actors \bowtie_{A.Name=M.Name} Movies))$$

$$Result(n) \leftarrow Actors(n, a, d) \text{ AND } Movies(n, y, t) \text{ AND } a < 50 \text{ AND } y = 2011$$

Other operators

R

A	B
…	…

S

X	Y
…	…

$$Result(a,b) \leftarrow R(a,b) \text{ AND } S(a,b)$$

$$Result(a,b) \leftarrow R(a,b) \text{ AND NOT } S(a,b)$$

$$Result(a,b) \leftarrow R(a,b)$$

$$Result(a,b) \leftarrow S(a,b)$$

Res(a,b) ← R(a,b) OR S(a,b)

Predicate types

- Extensional predicates (EDB)
  - Relations corresponding to predicates exist and are stored
- Intensional predicates (IDB)
  - Relations are virtual views and are temporary

$$Result(n) \leftarrow Actors(n, a, d) \text{ AND } Movies(n, y, t) \text{ AND } a < 50 \text{ AND } y = 2011$$

$$JoinResult(n, a, y) \leftarrow Actors(n, a, d) \text{ AND } Movies(n, y, t) \leftarrow \text{Join}$$

$$Result(n) \leftarrow JoinResult(n, a, y) \text{ AND } a < 50 \text{ AND } y = 2011 \leftarrow \text{Selection/projection}$$

Safe Rules

Actors

Name	Age	Addr
Priyanka Chopra	36	Mumbai
Anthony Hopkins	81	LA
Bill Nighy	68	LA
Abhishek Bachchan	42	Mumbai

$$Result(n, a, d) \leftarrow \text{NOT } Actors(n, a, d)$$

$Result(y) \leftarrow < (y, 35)$

Every variable that appears in a rule *must appear* in a non-negated, relational subgoal

Fix domain as finite

Recursive Rules

Edge

from	to

Finding paths: cannot be expressed in relational algebra

$$Path(f,t) \leftarrow Edge(f,t)$$

$$Path(f,t) \leftarrow Edge(f,s) \text{ AND } Path(s,t)$$

$$Path(f,t) \leftarrow Edge(f,t)$$

$$Path(f,t) \leftarrow Path(f,s) \text{ AND } Path(s,t)$$

Evaluating Recursive Queries (1/2)

Edge

from	to
Delhi	Bangalore
Bangalore	Chennai
Chennai	Pune

$$Path(f,t) \leftarrow Edge(f,t)$$

$$Path(f,t) \leftarrow Edge(f,s) \text{ AND } Path(s,t)$$

The only IDB

Path (f,t): 0

from	to

Path (f,t): 1

from	to
Delhi	Bangalore
Bangalore	Chennai
Chennai	Pune

Path (f,t): 2

from	to
Delhi	Bangalore
Bangalore	Chennai
Chennai	Pune
Delhi	Chennai
Bangalore	Pune

Path (f,t): 3

from	to
Delhi	Bangalore
Bangalore	Chennai
Chennai	Pune
Delhi	Chennai
Bangalore	Pune
Delhi	Pune

Evaluating Recursive Queries (2/2)

Least fixedpoint evaluation

```
All IDBs are initially empty
WHILE (changes to at least one IDB) DO
   FOR (each IDB predicate p) DO
      evaluate p using current values of all relations
```

Evaluating Non-recursive Queries

- Evaluation involves computing IDBs.
- Construct graph
  - Each IDB is a node
  - Directed edge between P and Q if Q occurs in the body of P
- Compute IDBs in topological sort order

Recursive Datalog to SQL

Basis

$$Path(f,t) \leftarrow Edge(f,t)$$

$$Path(f,t) \leftarrow Edge(f,s) \text{ AND } Path(s,t)$$

```
WITH RECURSIVE Path (f,t) AS
   (SELECT from, to
    FROM Edge
   UNION
    SELECT Edge.from, Path.t
    FROM Path, Edge
    WHERE Edge.to = Path.from)
SELECT * FROM Path
```

Negation and Recursion

R

ATTR
0

$$P(x) \leftarrow R(x) \text{ AND NOT } Q(x)$$

$$Q(x) \leftarrow R(x) \text{ AND NOT } P(x)$$

Least fixedpoint is not unique

$$P = (0), Q = \phi$$

$$Q = (0), P = \phi$$

Is 0 in P or Q ?

Stratified Negation

- Disallow negation in mutually recursive rules
- Construct graph
  - Each IDB is a node
  - Directed edge between P and Q if Q occurs in the body of P
  - *Label with* '–' if Q is negated
- Stratum of predicate P is the largest number of labeled edges on a path from P
- Evaluate the query from the lowest stratum

HW: Stratified negation and SQL