Introduction to Data Management CSE 344

Section 5: DataLog

TA: Yuyin Sun

Announcements

- Homework 4 due Friday, 11pm
- Midterm Monday!

Section Plan

- Review DataLog
  - Review basic concepts
- Exercise
  - Writing DataLog queries

DataLog

- A query language
- Its syntax is very similar to predicate logic
- It is also very similar to logical programming (Prolog)

DataLog Syntax

- Atom
  - **Relational atom**: if R is a relation name of arity n, and t_1, …, t_n are variables or constants, then R(t_1, …, t_n) is a relational atom
  - **Arithmetic atom**: comparison between two arithmetic expressions, for example x > 5, x >= y, etc

DataLog Syntax (cont')

- Rule
  - R_1 :- R_2, …, R_n
  - n >= 1
  - R_1 is a relational atom, head of the rule
  - R_2, …, R_n are relational atom or arithmetic atoms, possible preceded by NOT
  - R_2, …, R_n are called subgoals, body of the rule

DataLog Example

- Suppose we have a relation Person over schema (Name, Age, Address, Telephone). Then the following DataLog rule will define a relation which contains names of people aged over 18:
  - Adult(x) :- Person(x, y, z, u), y >= 18

Datalog: Facts

Facts = tuples in the database

```
Person( 'Aaron', 34, CSE491).
Person( 'Daniel', 23, CSE491).
Person( 'Thomas', 36, CSE491).
Person( 'Dieter', 49, CSE564).
Person( 'Connor', 21, CSE491).
Person( 'Tanner', 24, CSE491).
```

DataLog Queries: Rules

Person(Name, Age, Address, Telephone)

Facts = tuples in the database

```
Person( 'Aaron', 34, CSE491).
Person( 'Daniel', 23, CSE491).
Person( 'Thomas', 36, CSE491).
Person( 'Dieter', 49, CSE564).
Person( 'Connor', 21, CSE491).
Person( 'Tanner', 24, CSE491).
```

Rules = select interesting facts

```
Age30( 'Aaron', 34).
Age30( 'Thomas', 36).
Age30( 'Dieter', 49).
```

```
Age30(a, b) :- Person(a, b, c), b > 30
```

Actor(id, fname, lname)
Casts(pid, mid)
Movie(id, name, year)

Datalog: Facts and Rules

Facts = tuples in the database

Rules = queries

```
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).
```

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

Find Movies made in 1940

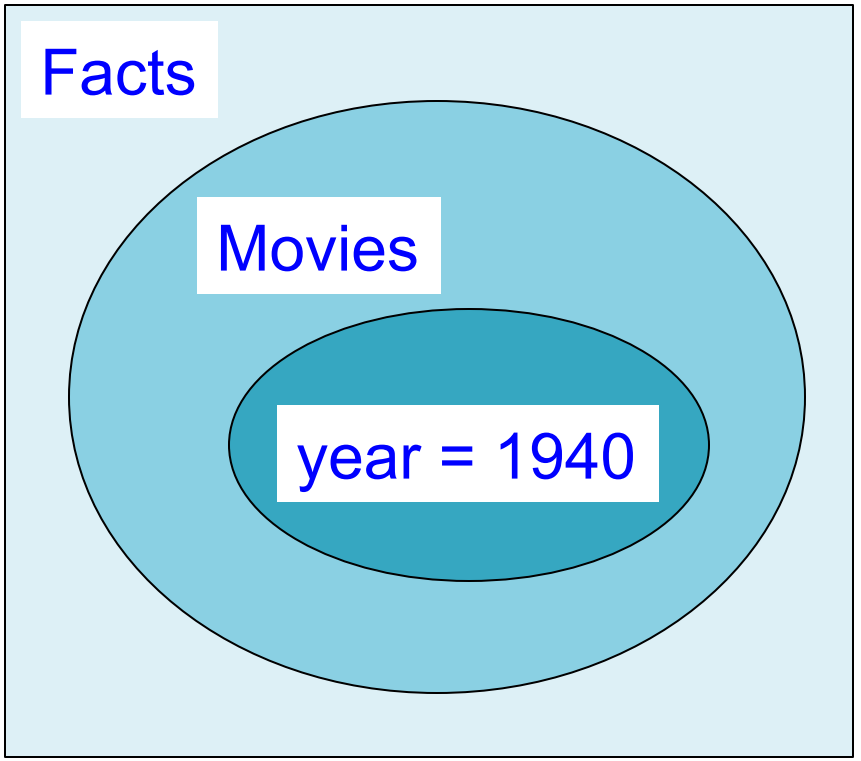
Actor(id, fname, lname)
Casts(pid, mid)
Movie(id, name, year)

Meaning of DataLog

Rules = queries

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

Find Movies made in 1940



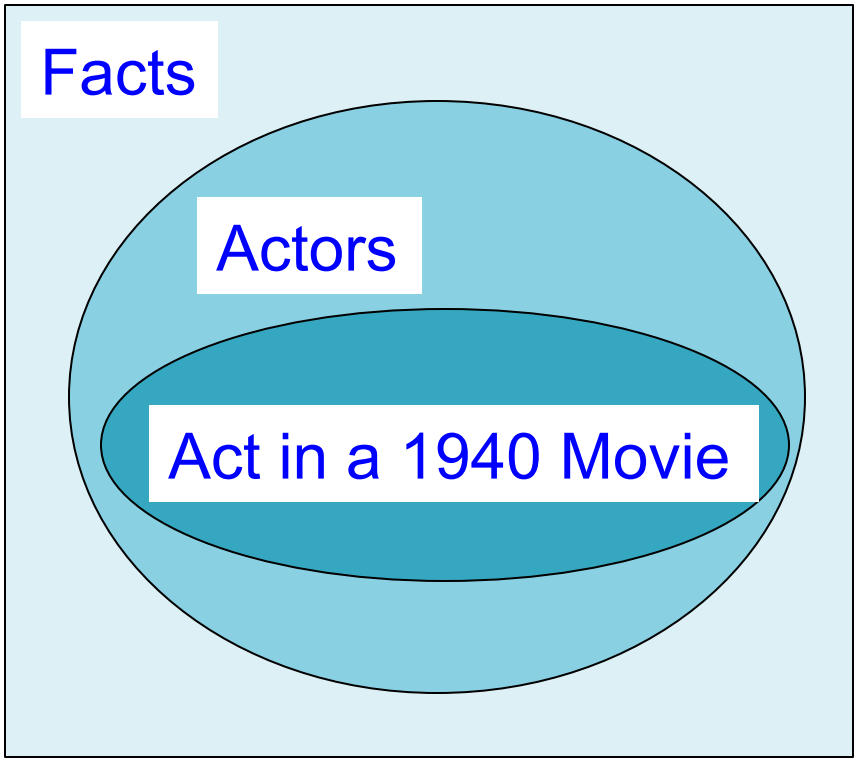
Actor(id, fname, lname)
Casts(pid, mid)
Movie(id, name, year)

More Examples

Rules = queries

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

Find Actors who acted in Movies made in 1940



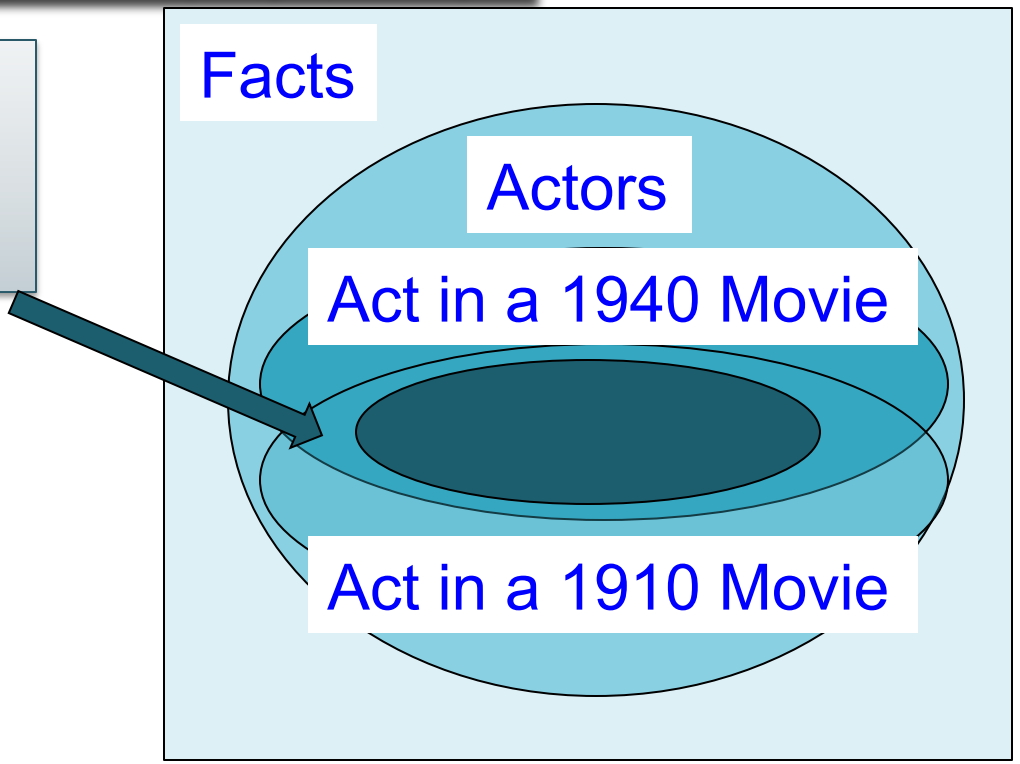
Actor(id, fname, lname)
Casts(pid, mid)
Movie(id, name, year)

More Examples

Rules = queries

```
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

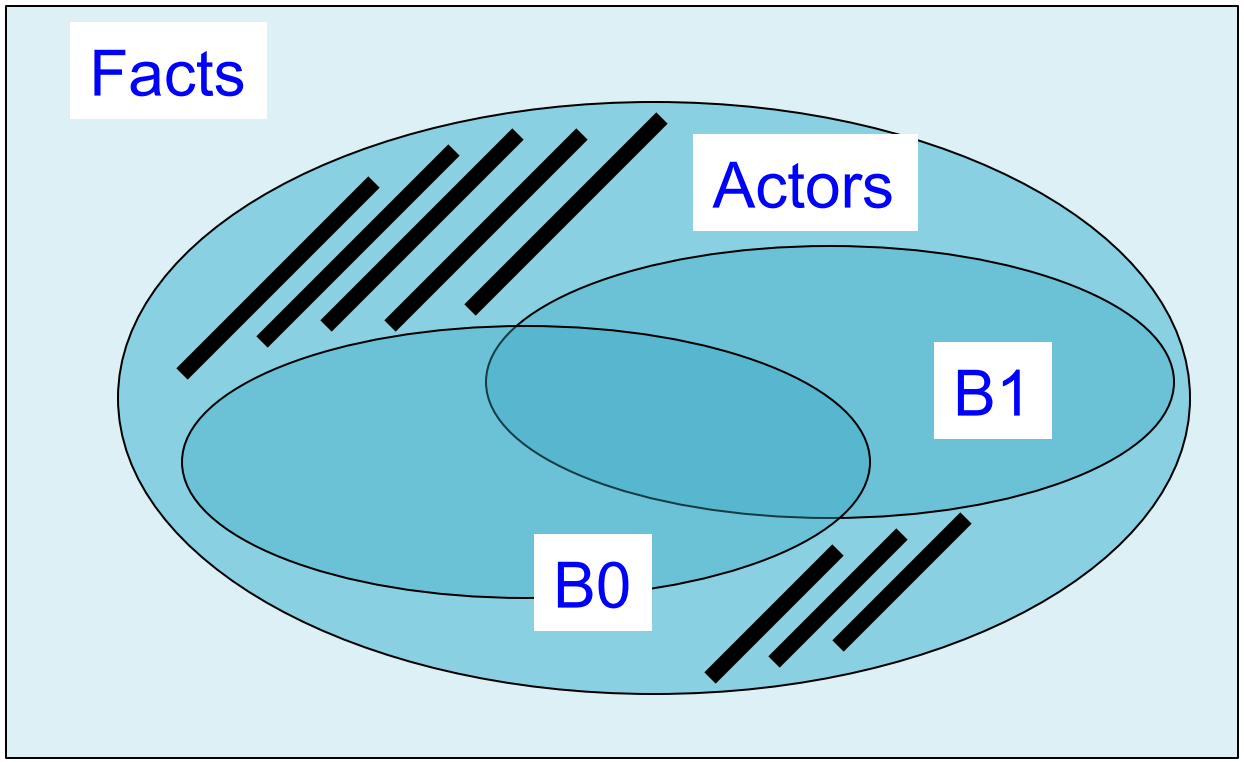


Actor(id, fname, lname)
Casts(pid, mid)
Movie(id, name, year)

Negation Example

Find all actors who do not have a 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)
```



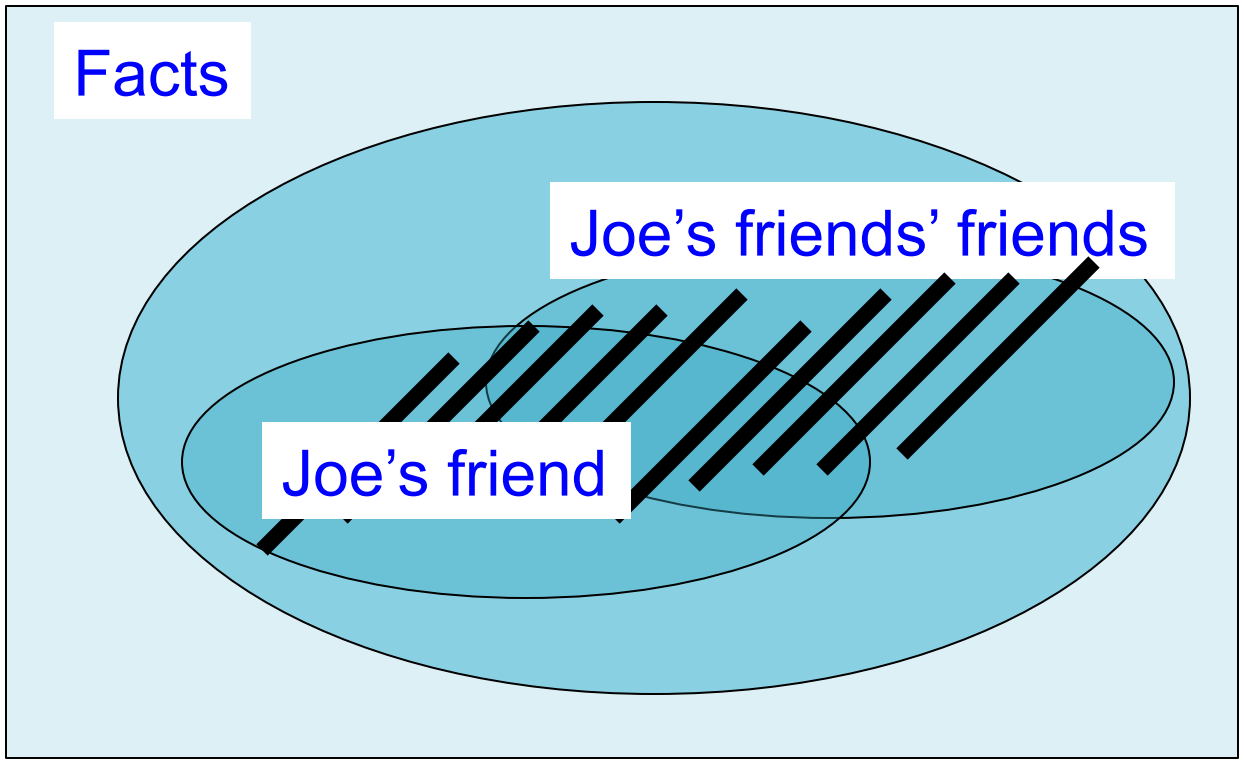
Friend(name1, name2)

Enemy(name1, name2)

More Examples

Find Joe's friends, and Joe's friends of friends.

```
A(x) :- Friend('Joe', x)
A(x) :- Friend('Joe', z), Friend(z, x)
```



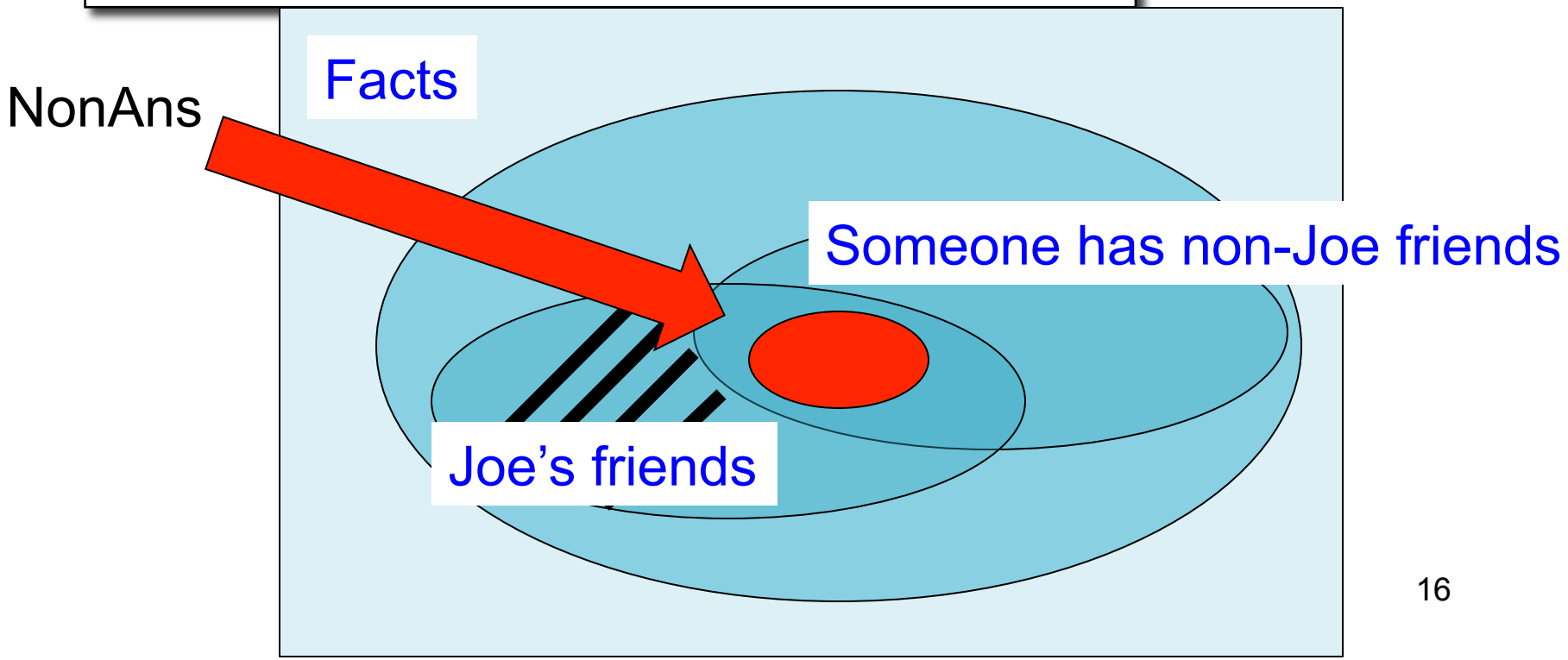
Friend(name1, name2)

Enemy(name1, name2)

More Examples

Find all of Joe's friends who do not have any friends except for Joe:

```
JoeFriends(x) :- Friend('Joe',x)
NonAns(x) :- JoeFriends(x), Friend(x,y), y != 'Joe'
A(x) :- JoeFriends(x), NOT NonAns(x)
```



Friend(name1, name2)

Enemy(name1, name2)

More Examples

Find all people such that all their enemies' enemies are their friends

- Q: if someone doesn't have any enemies nor friends, do we want them in the answer?
- A: Yes!

```
Everyone(x) :- Friend(x,y)
Everyone(x) :- Friend(y,x)
Everyone(x) :- Enemy(x,y)
Everyone(x) :- Enemy(y,x)
NonAns(x) :- Enemy(x,y),Enemy(y,z), NOT Friend(x,z)
A(x) :- Everyone(x), NOT NonAns(x)
```

DataLog

- Person(Name, Age, Address, Telephone)
- people aged over 18:
  - Adult(x) :- Person(x, y, z, u), y >= 18
  - Adult(a) :- Person(a, b, c, d), b >= 18
  - Adult(x) :- Person(x, y, _, _), y >= 18

z, u are dummy variables, can be replaced by underscores