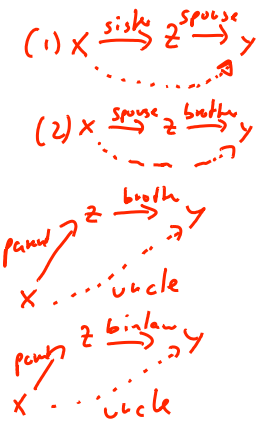
Announcements

- Assignments:
  - Individual HW1 (came out Wed; due next Fri)
  - Group Project(-let) out! (due in 3 weeks)
- Suggested Datalog systems
  - DES Datalog
  - DLV Datalog
  - Later: LogiQL
- Next week:
  - Lectures Mon, **Wed 9am**, Wed 3:10pm
  - Discussion Fri 3:10pm

Living in the family

```
brother_in_law(X,Y) <- sister(X,Z), spouse(Z,Y).
brother_in_law(X,Y) <- spouse(X,Z), brother(Z,Y).
sister_in_law(X,Y)  <- brother(X,Z), spouse(Z,Y).
sister_in_law(X,Y)  <- spouse(X,Z), sister(Z,Y).

uncle(X,Y) <- parent(X,Z), brother(Z,Y).
uncle(X,Y) <- parent(X,Z), brother_in_law(Z,Y).
aunt(X,Y)  <- parent(X,Z), sister(Z,Y).
aunt(X,Y)  <- parent(X,Z), sister_in_law(Z,Y).
```



Datalog Syntax

- A **relational database** is given as a set of **facts**:

```
employee(john, 40000, toys). ...
employee(mary, 65000, cs). ...
...
dept(cs, mary). ...
```

- A **Datalog program** defines **views** by means of **rules** of the form **Head** ← **Body**:

```
boss(Emp,Mgr) ← employee(Emp, Salary, DeptNo), dept(DeptNo,Mgr)
highpaid(Emp) ← employee(Emp, Salary, _), Salary > 60000
```

- **EDB**: extensionally defined relations (facts): `employee/3, dept/2`
- **IDB**: intensionally (i.e., rule-) defined relations (views): `boss/2`
- A **query** is a view with a distinguished **answer**/n relation:

```
answer(Emp,Mgr) ← employee(Emp, Salary, DeptNo), dept(DeptNo,Mgr)
```

Notation:

- **lowercase**: **relation names** (`employee/3, highpaid/1, ...`) and **constants** (aka data values: `john, toys, 50000, ...`)
- **UPPERCASE/Capitalized**: **variables** (`Emp, X, ...`) ("_" means: don't care)

Datalog: Variables are local to a rule

- What about these rules?
  1. p(X,Y) :- q(X,Y,Z).
  2. p(U,V) :- q(U,V,W).

- What is the equivalent RA (SQL) expression for (1) and (2)?

Datalog: Safety

- Safety:
  - every variable occurs positively in the body! (why?)
    - … more precisely, in a positive relational atom.
- In particular,
  - every head variable occurs positively in the body
  - every negated (body) variable occurs positively in the body

↑ the rule is then range-restricted (safe)

Datalog vs Relational Algebra

Relational operations have concise representations! Examples: sel := $\sigma_{X=a \wedge X \neq Y}(p)$

```
sel(X,Y) :- p(X,Y), X=a, not X=Y.   % SELECT some tuples from p(X,Y)

proj(X) :- p(X,Y).                  % PROJECT on the first argument

join(X,Y,Z) :- p(X,Y), q(Y,Z).      % JOIN p(A,B), q(C,D) s.t. B=C

prod(X,Y) :- p(X), q(Y).            % PRODUCT of p(X) and q(Y)

intersect(X) :- p(X), q(X).         % INTERSECTION of p(X), q(X)

diff(X) :- p(X), not q(X).          % SET-DIFFERENCE: p(X) \ q(X)

union(X) :- p(X).                   % UNION of p(X), ...
union(X) :- q(X).                   % ... and q(X)
```

proj := $\Pi_X(p) = \Pi_{\$1}(p)$

join := $p \bowtie_{\$2=\$1} q$

prod := $p \times q$

intersect := $p \cap q$

diff := $p \setminus q$

union := $p \cup q$

Rules have a "logical reading" (i.e., rules are formulas):

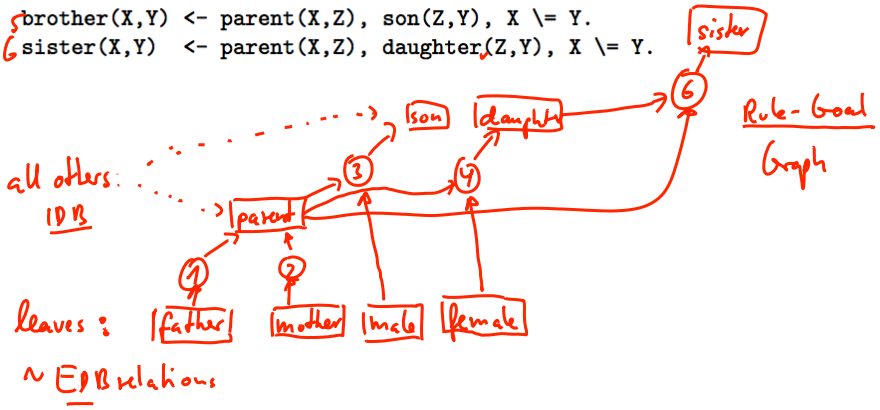
$$\forall X \; (\; \mathtt{diff}(X) \leftarrow \mathtt{p}(X) \wedge \neg \mathtt{q}(X) \;).$$
$$\forall X \; (\; \mathtt{union}(X) \leftarrow \mathtt{p}(X) \vee \mathtt{q}(X) \;).$$

Living in the family: Rule-Goal Graph

```
1 parent(X,Y) <- father(X,Y).
2 parent(X,Y) <- mother(X,Y).

3 son(X,Y)      <- parent(Y,X), male(Y).
4 daughter(X,Y) <- parent(Y,X), female(Y).

5 brother(X,Y) <- parent(X,Z), son(Z,Y), X \= Y.
6 sister(X,Y)  <- parent(X,Z), daughter(Z,Y), X \= Y.
```

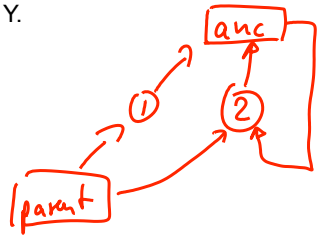


Datalog: Rule-Goal Graph (cont'd)

```
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.

1 ancestor(X, Y) :- parent(X, Y).
2 ancestor(X, Y) :- parent(X, Z), ancestor(Z, Y).
```

anc := parent
Repeat
  anc := parent ⋈ anc
         $2=$1
Until no change



Bottom-up Fixpoint Evaluation