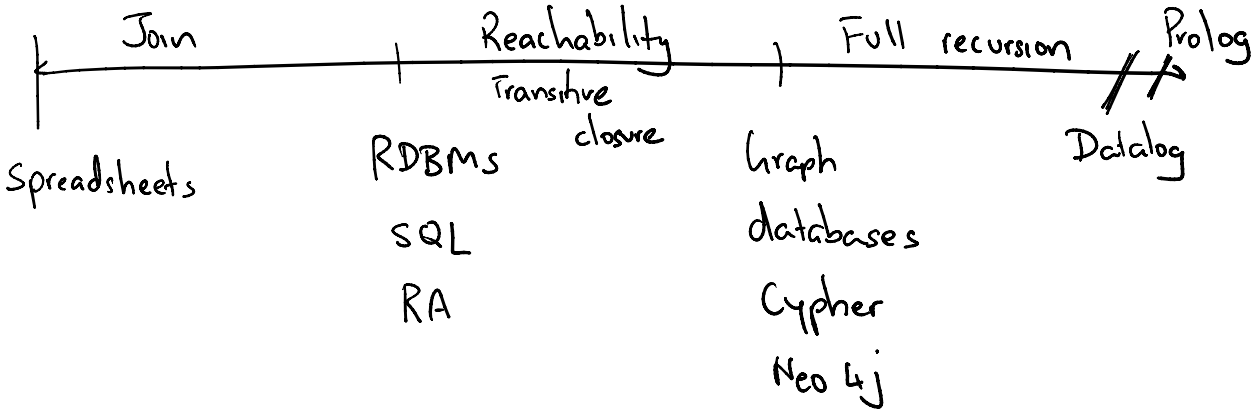
Lecture 26

Tuesday, November 23, 2021 11:54 AM



Question: How do we express deeper forms of recursion?

Example (Non-recursive)

Cousins: Two people whose parents are siblings

Two people who share the same grandparent

Input

Parent

c_1	c_2
x	y

⟹ "x is a parent of y"

Output

Cousin

c_1	c_2
x	y

"x is a cousin of y"

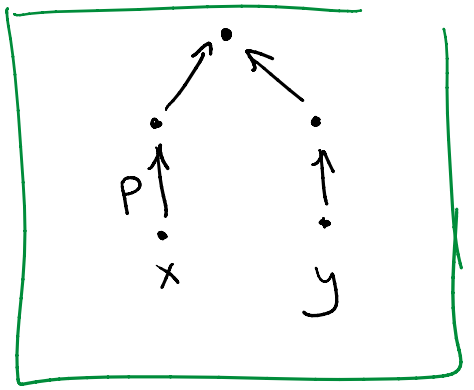
Computing Cousins using RA / SQL

select $(P_1 \cdot c_1, P_2 \cdot c_2)$

from Parent as P_1

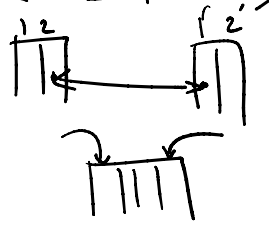
Parent as P_2

where $P_1 \cdot c_2 = P_2 \cdot c_1$



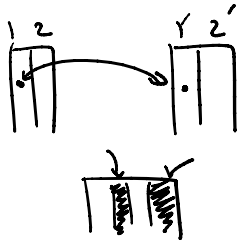
Basic graph pattern

$\pi_{1,4}(P \underset{2=1'}{\bowtie} P)$

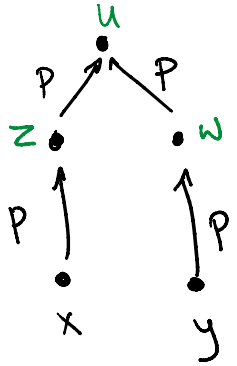


GP

$\pi_{2,4}(GP \underset{1=1'}{\bowtie} GP)$



Cousins.



For all People x, y, z, w, u,

If we have

Parent (x z) & Parent (z u) &

Parent (y w) & Parent (w u)

then we have Cousins (x y).

, | then we have Cousins(x y).

$\forall x,y,z,w,u$

Cousins(x, y) :- P(x, z), P(z, w), P(y, w), P(w, u).

↑ Implicit

This is our first Datalog query.

"clause" "rule".

Head ← "Output"?

Body ← "Input" (?)

How could we write recursive Datalog queries?

Just use the same relation in the head & body!

E(x,y)

P(x,y)

P(x, y) :- E(x, y). ← "or": Just write multiple rules

P(x, z) :- E(x, y), P(y, z).

$$\frac{E(x,y) \quad P(y,z)}{P(x,z)}$$

$P(x,z) :- E(x,y), P(y,z)$.

and

recursion

Theorem: Non-recursive Datalog $=$ Relational algebra positive

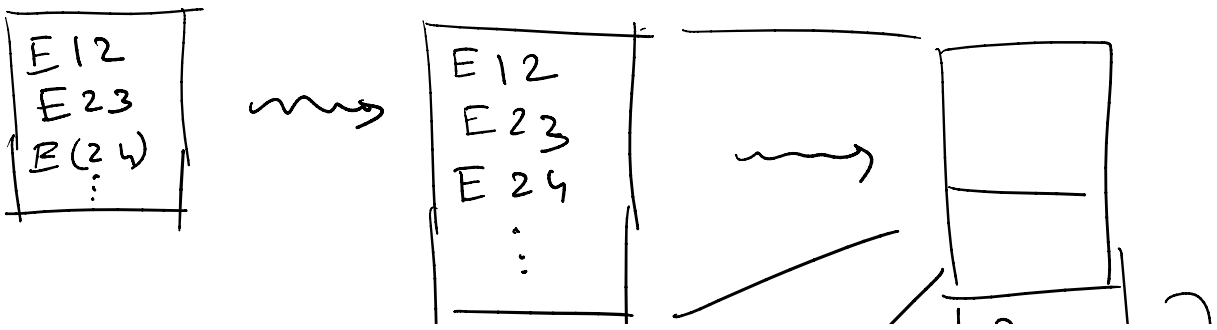
$=$: In expressive power

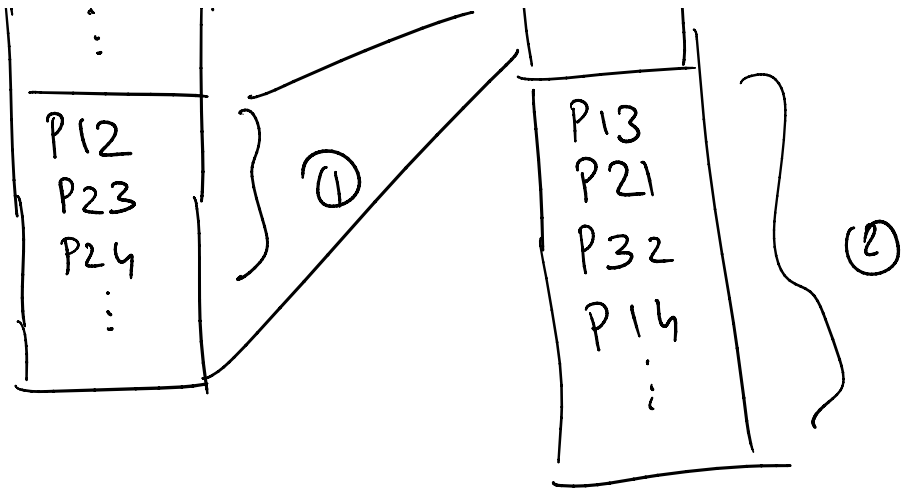
Recursive Datalog

What does a recursive Datalog query "mean"?

$P(x,y) :- E(x,y)$ ... ①

$P(x,z) :- E(x,y), P(y,z)$ ... ②




P12
P23
P24
①
P13
P21
P32
P14
②