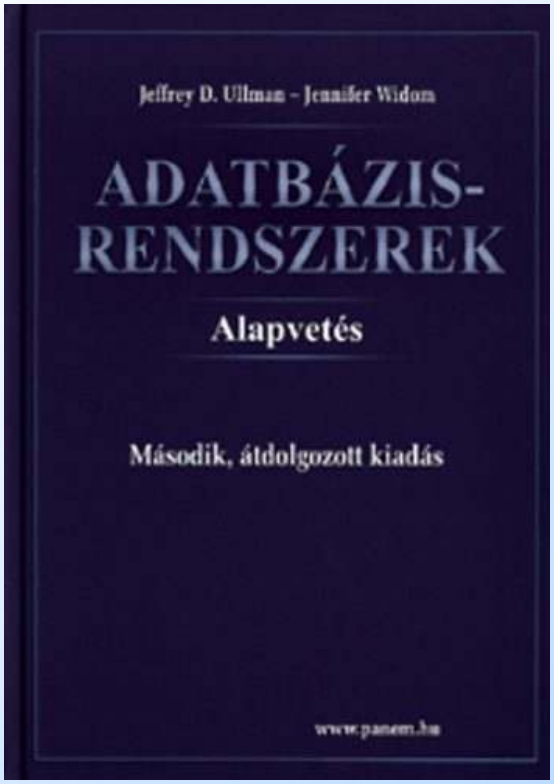
Datalog -2: Rekurzió



Ullman-Widom: Adatbázisrendszerek

Alapvetés

Második, átdolgozott kiadás, Panem, 2009

10.2. Rekurzió Datalogban és SQL-99-ben

(Jeffrey D. Ullman, 2007)

Datalog

- Datalog program – véges sok Datalog szabály megadásából áll
- Extenzionális, intenzionális predikátumok
- Datalog program kiértékelése
- Rekurzió megadása, ha nem használunk negációt

Expressive Power of Datalog

- Without recursion, Datalog can express all and only the queries of core relational algebra.
  - The same as SQL select-from-where, without aggregation and grouping.
- But with recursion, Datalog can express more than these languages.

Recursive Example

- EDB: Par(c,p) = *p* is a parent of *c*.
- Generalized cousins: people with common ancestors one or more generations back:

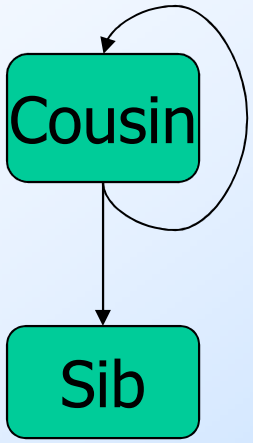
```
Sib(x,y) <- Par(x,p) AND Par(y,p) AND x<>y
Cousin(x,y) <- Sib(x,y)
Cousin(x,y) <- Par(x,xp) AND Par(y,yp)
                AND Cousin(xp,yp)
```

Rekurzió definíciója

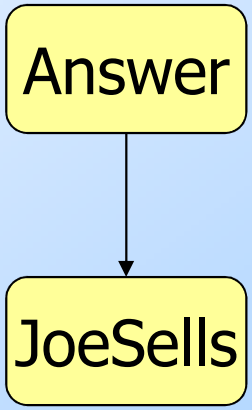
- Megelőzési gráf pontjai P1, ... Pn

  élek Pi $\rightarrow$ Pj-be, ha van olyan szabály, amelynek feje Pj és törzsében szerepel Pi
- Form a *dependency graph* whose nodes = IDB predicates.
- Arc *X* -> *Y* if and only if there is a rule with *X* in the head and *Y* in the body.
- Cycle = recursion; no cycle = no recursion

Example: Dependency Graphs



Recursive

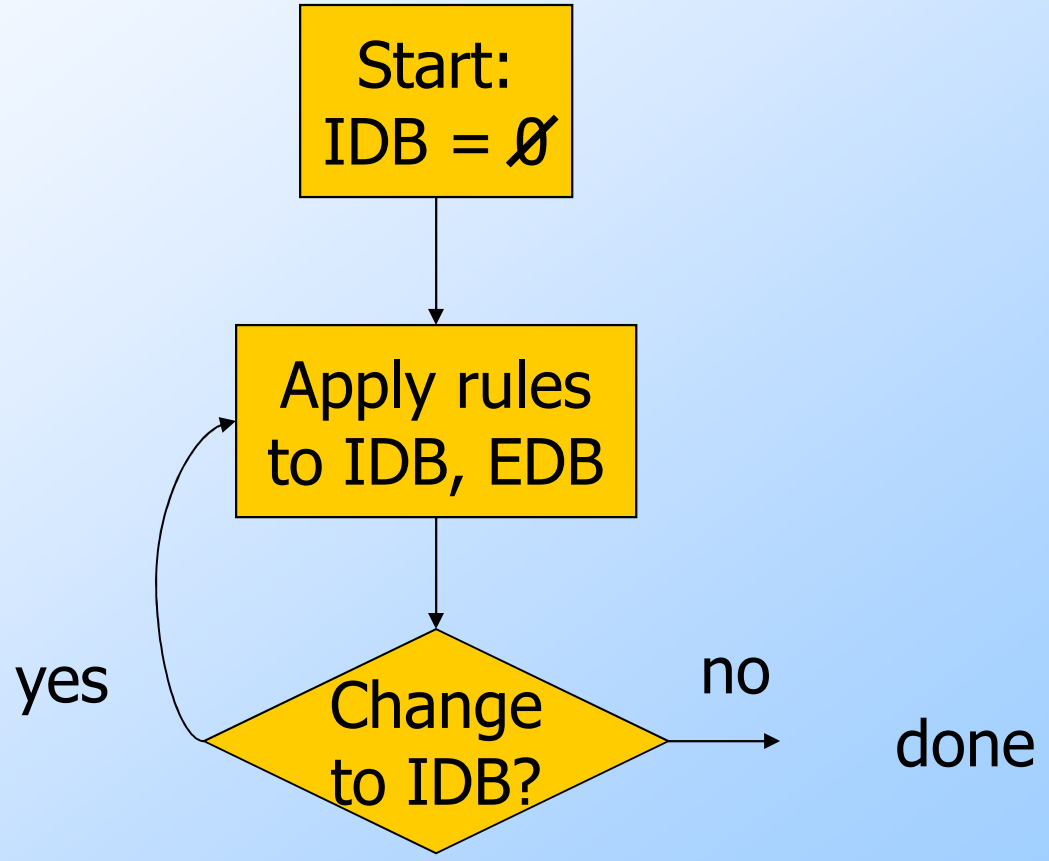


Nonrecursive

Evaluating Recursive Rules

- The following works when there is no negation:
  1. Start by assuming all IDB relations are empty.
  2. Repeatedly evaluate the rules using the EDB and the previous IDB, to get a new IDB.
  3. End when no change to IDB.

The "Naïve" Evaluation Algorithm



Seminaive Evaluation

- Since the EDB never changes, on each round we only get new IDB tuples if we use at least one IDB tuple that was obtained on the previous round.
- Saves work; lets us avoid rediscovering *most* known facts.
  - A fact could still be derived in a second way.

SQL-99 Recursion

- Datalog recursion has inspired the addition of recursion to the SQL-99 standard. (IBM DB2 implementálta)
- Tricky, because SQL allows negation grouping-and-aggregation, which interact with recursion in strange ways.
- Megszorítások a rekurzióra:

  monoton lekérdezések

  csak lineáris rekurzió szerepel

Form of SQL Recursive Queries

WITH
[RECURSIVE] <relnév1> AS (SELECT…)
[RECURSIVE] <relnév2> AS (SELECT…)
…
[RECURSIVE] <relnévk> AS (SELECT…)
 SELECT … FROM listán <relnévi>-k

Example: SQL Recursion – (1)

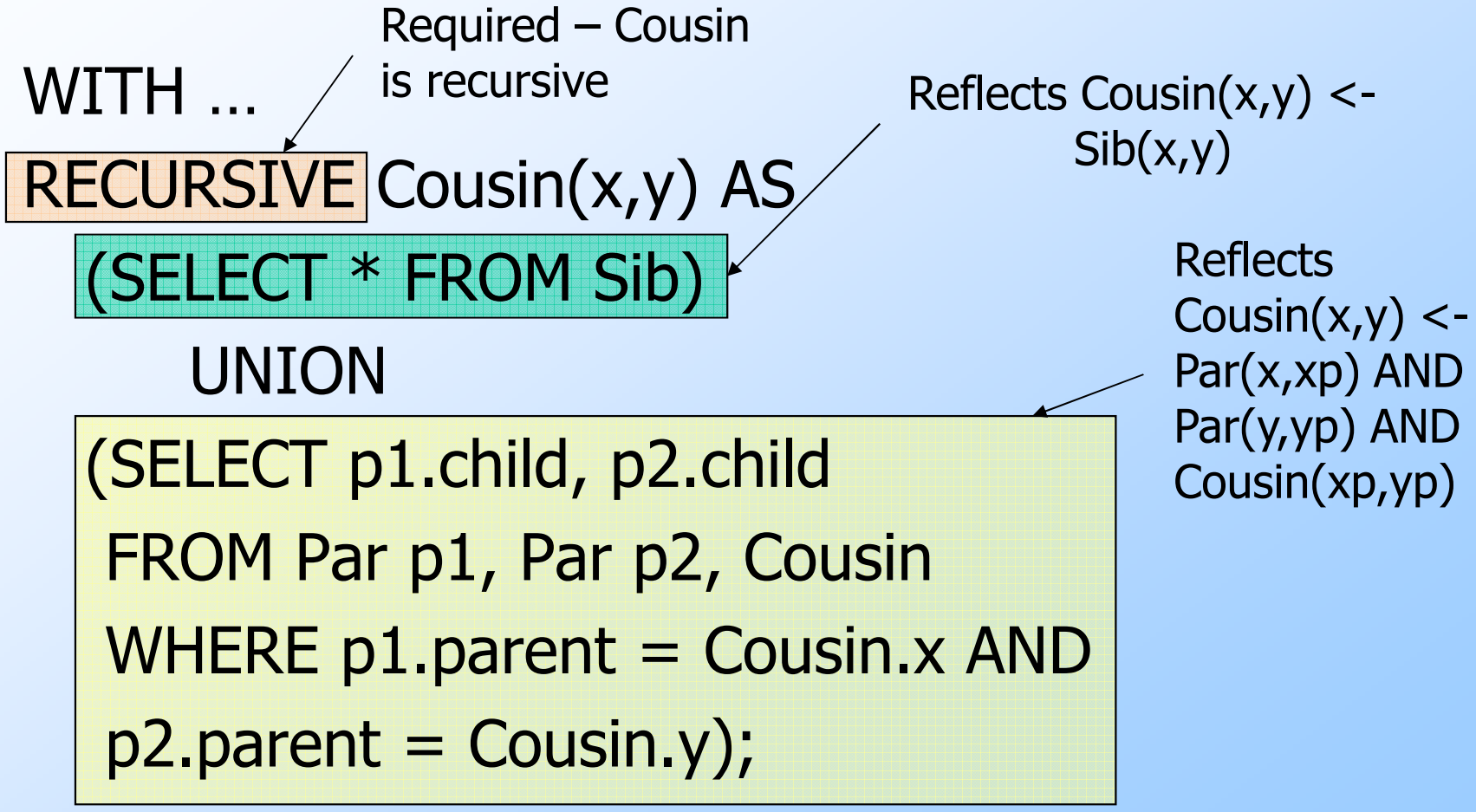
- Find Sally's cousins, using SQL like the recursive Datalog example.
- Par(child,parent) is the EDB.

WITH Sib(x,y) AS

```
SELECT p1.child, p2.child
FROM Par p1, Par p2
WHERE p1.parent = p2.parent AND
      p1.child <> p2.child;
```

Like Sib(x,y) <-
Par(x,p) AND
Par(y,p) AND
x <> y

Example: SQL Recursion – (2)



Example: SQL Recursion – (3)

- With those definitions, we can add the query, which is about the virtual view Cousin(x,y):

```
SELECT y
FROM Cousin
WHERE x = 'Sally';
```

Legal SQL Recursion

- It is possible to define SQL recursions that do not have a meaning.
- The SQL standard restricts recursion so there is a meaning.
- And that meaning can be obtained by seminaïve evaluation.

Example: Meaningless Recursion

- EDB: P(x) = {(1)}.
- IDB: Q(x) <- P(x) AND NOT Q(x).
- Is (1) in Q(x)?
  - If so, the recursive rule says it is not.
  - If not, the recursive rule says it is.

Plan to Explain Legal SQL Recursion

1. Define “monotone” recursions.
2. Define a “stratum graph” to represent the connections among subqueries.
3. Define proper SQL recursions in terms of the stratum graph.

Monotonicity

- If relation P is a function of relation Q (and perhaps other relations), we say P is *monotone* in Q if inserting tuples into Q cannot cause any tuple to be deleted from P.
- Examples:
  - $P = Q \cup R$.
  - $P = \sigma_{a=10}(Q)$.

Rekurzió a gyakorlatban

- SQL-99 szabvány csak az ún. „monoton” rekurziót támogatja (viszont Datalogban megengedett a negáció és rekurzió együtt, de igen bonyolulttá válik a lekérdezés értelmezése, mint például a rétegzés),
- SQL-99 szabvány lineáris rekurziót enged meg, vagyis az értékadás kifejezésben egyetlen rekurzív relációt használhatunk.

Tankönyv példája

- Jaratok(legitarsasag, honnan, hova, koltseg, indulas, erkezes) táblában repülőjáratok adatait tároljuk.
- Mely (x,y) párokra lehet eljutni x városból y városba?
- Eljut(x, y) <- Jaratok(_, x, y, _, _, _)
  Eljut(x, y) <- Eljut(x, z) AND Jaratok(_, z, y, _, _, _)
- WITH RECURSIVE Eljut AS
  (SELECT honnan, hova FROM Jaratok
  UNION
  (SELECT Eljut.honnan, Jaratok.hova
  FROM Eljut, Jaratok
  WHERE Eljut.hova = Jaratok.honnan)
  SELECT * FROM eljut;
- Lásd még a gyakorlaton is PL/SQL programmal kifejezve!