

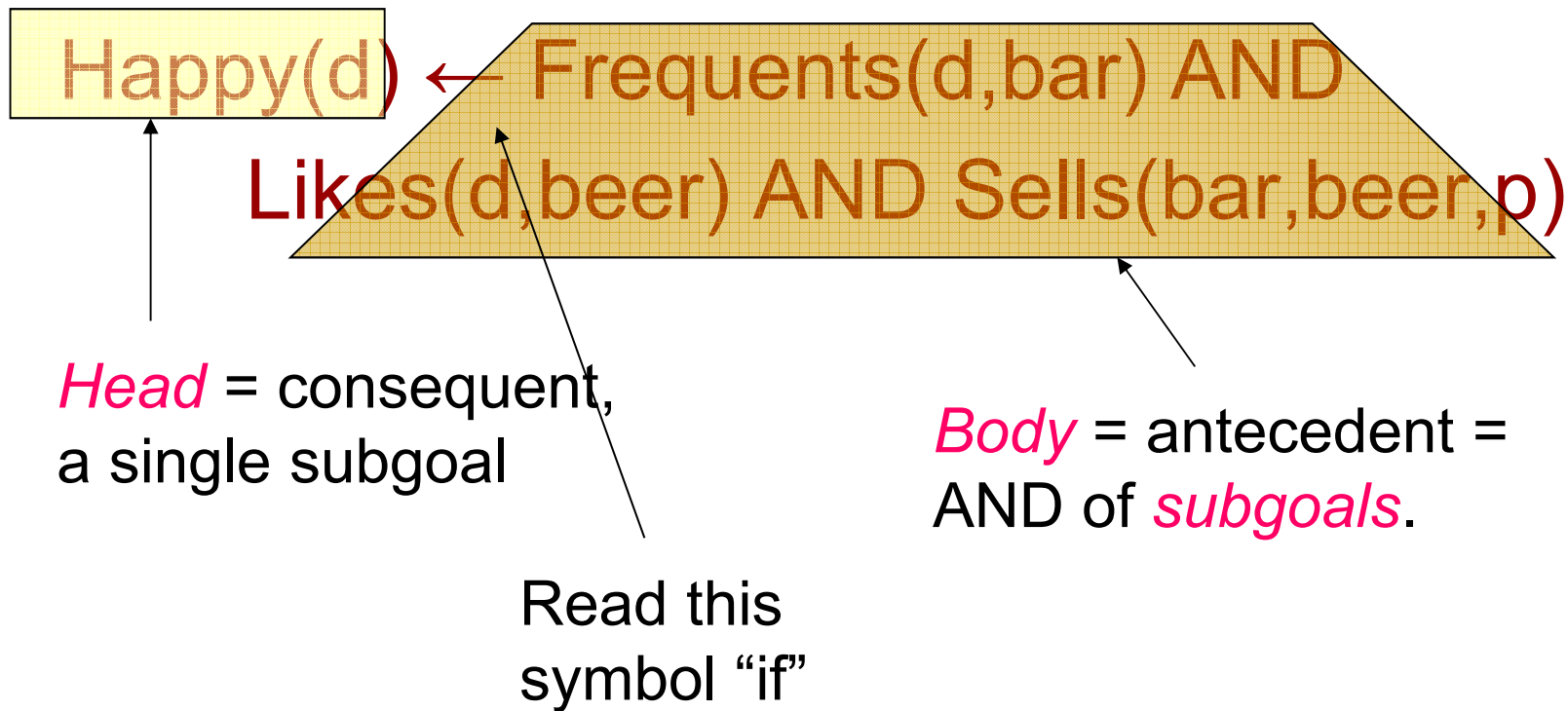
Datalog: logikai alapú lekérdező nyelv

Tankönyv: Ullman-Widom:
Adatbázisrendszerek Alapvetés
Második, átdolgozott kiadás,
Panem, 2009

5.4. Relációs algebra és
nem-rekurzív biztonságos Datalog
kifejező erejének összehasonlítása



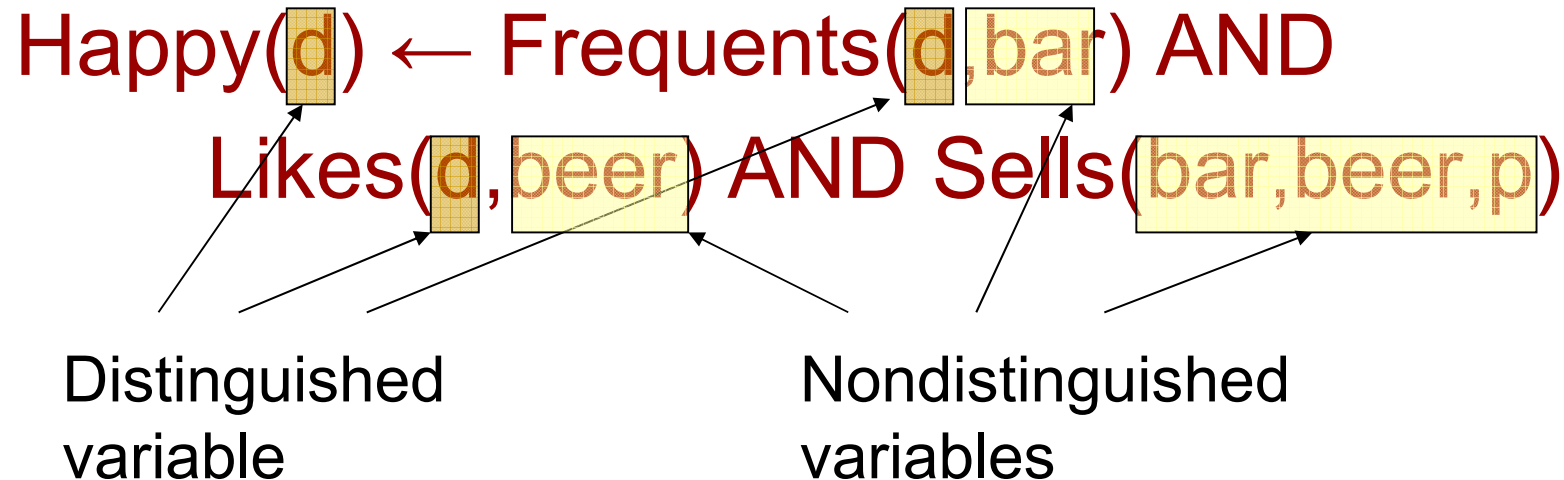
Emlékeztető: Datalog szabályok



fej ← törzs (jelentése, kiértékelése)

Relációs atomok, aritmetikai atomok

Példa: szabály kiértékelése



Interpretation: drinker d is happy if there exist a bar, a beer, and a price p such that d frequents the bar, likes the beer, and the bar sells the beer at price p .

Milyen problémák merülnek fel?

- Relations are finite sets.
- We want rule evaluations to be finite and lead to finite results.
- “Unsafe” rules like $P(x) \leftarrow Q(y)$ have infinite results, even if Q is finite.
- Even $P(x) \leftarrow Q(x)$ requires examining an infinity of x -values.

Példa: Aritmetikai atomok

- A beer is “cheap” if there are at least two bars that sell it for under \$2.

Cheap(beer) \leftarrow Sells(bar1,beer,p1) AND
Sells(bar2,beer,p2) AND $p1 < 2.00$
AND $p2 < 2.00$ AND $bar1 \neq bar2$

Negált részcélok

- NOT in front of a subgoal negates its meaning.
- **Example:** Think of $\text{Arc}(a,b)$ as arcs in a graph.
 - $S(x,y)$ says the graph is not transitive from x to y ; i.e., there is a path of length 2 from x to y , but no arc from x to y .

$$S(x,y) \leftarrow \text{Arc}(x,z) \text{ AND } \text{Arc}(z,y) \\ \text{AND NOT } \text{Arc}(x,y)$$

Biztonságossági elvárás

- A szabályok kiértékelhetőek legyenek, ehhez:
- A szabályban szereplő minden változónak elő kell fordulnia a törzsben nem-negált relációs atomban

Biztonságos szabályok

- A rule is *safe* if:
 1. Each distinguished variable,
 2. Each variable in an arithmetic subgoal, and
 3. Each variable in a negated subgoal,
also appears in a nonnegated,
relational subgoal, amivel az x korlátozott:
 - $\text{pred}(x, y, \dots)$ argumentuma (értéke a táblából)
 - vagy $x=c$ (konstans)
 - vagy $x=y$ (ahol y korlátozott)
- Safe rules prevent infinite results.

Példa: Nem biztonságos szabályokra

- Each of the following is unsafe and not allowed:
 1. $S(x) \leftarrow R(y)$
 2. $S(x) \leftarrow R(y) \text{ AND } x < y$
 3. $S(x) \leftarrow R(y) \text{ AND NOT } R(x)$
- In each case, an infinity of x 's can satisfy the rule, even if R is a finite relation.

Datalog programok

- **Datalog program** = collection of rules.
- In a program, predicates can be either
 1. EDB relációk = **Extensional Database** = stored table (csak a törzsben szereplő relációk)
 2. IDB relációk = **Intensional Database** = relation defined by rules (szerepel fej-ben)
- Never both! No EDB in heads.

Datalog programok kiértékelése

- As long as there is no recursion, we can pick an order to evaluate the IDB predicates, so that all the predicates in the body of its rules have already been evaluated.
- If an IDB predicate has more than one rule, each rule contributes tuples to its relation.

Példa: Datalog program

- Using EDB $Sells(bar, beer, price)$ and $Beers(name, manf)$, find the manufacturers of beers Joe doesn't sell.

$JoeSells(b) \leftarrow Sells('Joe's Bar', b, p)$

$Answer(m) \leftarrow Beers(b, m)$

AND NOT $JoeSells(b)$

Példa: Kiértékelése

- **Step 1:** Examine all **Sells** tuples with first component 'Joe's Bar'.
 - Add the second component to **JoeSells**.
- **Step 2:** Examine all **Beers** tuples (b,m) .
 - If b is not in **JoeSells**, add m to Answer.

Datalog kifejező ereje

- 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.

Relációs algebrai kifejezésfák átírása

- Mi a leggyakrabban előforduló típus, amiből építkeznek? $\Pi_{\text{Lista}}(\sigma_{\text{Felt}}(\mathbf{R} \bowtie \mathbf{S} \bowtie \dots))$
Ezt a komponenst támogatja legerősebben az SQL is: **SELECT lista**
FROM táblák összekapcsolása
WHERE felt
Ez felel meg egy Datalog szabálynak...
Halmazműveletek: unió Datalog programmal

Relációs algebra és Datalog ---1

Rel.algebrai műveletek hogyan néznek ki Datalogban?

Halmazműveletek: T.f.h $R(x_1, \dots, x_n)$, $S(x_1, \dots, x_n)$

predikátumokhoz tartozó reláció $R(A_1, \dots, A_n)$, $S(A_1, \dots, A_n)$

- $R \cap S$ metszetnek megfelelő szabály:

Válasz(x_1, \dots, x_n) $\leftarrow R(x_1, \dots, x_n) \text{ AND } S(x_1, \dots, x_n)$

- $R - S$ különbségnek megfelelő szabály:

Válasz(x_1, \dots, x_n) $\leftarrow R(x_1, \dots, x_n) \text{ AND NOT } S(x_1, \dots, x_n)$

- $R \cup S$ unió műveletet egyetlen szabállyal nem tudom felírni, mert a törzsben csak AND lehet, OR nem. Ehhez több szabályból álló Datalog program kell:

Válasz(x_1, \dots, x_n) $\leftarrow R(x_1, \dots, x_n)$

Válasz(x_1, \dots, x_n) $\leftarrow S(x_1, \dots, x_n)$

Relációs algebra és Datalog ---2

Kiválasztás:

- $\sigma_{x_i \theta x_j}(R)$ kifejezésnek megfelelő szabály :
Válasz(x_1, \dots, x_n) $\leftarrow R(x_1, \dots, x_n)$ AND $x_i \theta x_j$
- $\sigma_{x_i \theta c}(E1)$ kifejezésnek megfelelő szabály:
Válasz(x_1, \dots, x_n) $\leftarrow R(x_1, \dots, x_n)$ AND $x_i \theta c$

Vetítés:

- $\Pi_{A_{i_1}, \dots, A_{i_k}}(R)$ kifejezésnek megfelelő szabály:
Válasz(x_{i_1}, \dots, x_{i_k}) $\leftarrow R(x_1, \dots, x_n)$

Megjegyzés: név nélküli anonimus változók, amelyek csak egyszer szerepelnek és mindegy a nevük azt aláhúzás helyettesítheti. Például:

HosszúFilm(c,é) \leftarrow Film(c,é,h,_,_,_) AND $h \geq 100$

Relációs algebra és Datalog ---3

Természetes összekapcsolás: Tegyük fel, hogy $R(A_1, \dots, A_n, C_1, \dots, C_k)$ és $S(B_1, \dots, B_m, C_1, \dots, C_k)$

- $R \bowtie S$ kifejezésnek megfelelő szabály:

Válasz($x_1, \dots, x_n, y_1, \dots, y_m, z_1, \dots, z_k$) \leftarrow

$\leftarrow R(x_1, \dots, x_n, z_1, \dots, z_k) \text{ AND } S(y_1, \dots, y_m, z_1, \dots, z_k)$

- A felírt szabályok biztonságosak.
- Minden Q relációs algebrai kifejezéshez van nem rekurzív, biztonságos, negációt is tartalmazó Datalog program, amelyben egy kitüntetett IDB predikátumhoz tartozó kifejezés ekvivalens a Q lekérdezéssel.
- A nem rekurzív, biztonságos, negációt is tartalmazó Datalog kifejezőerő tekintetében EKVIVALENS a relációs algebrával.

Példák átírásokra --- 1

- Lekérdezések megadása: Tk.2.4.1.Termékes feladata:
Korábbi feladatokat lásd a relációs algebrai kifejezésekre
ezekhez rajzoljuk fel a kiértékelő fájt és azokat alakítsuk át
SQL lekérdezéssé: SELECT utasításra

- **Példa:** Adottak az alábbi **relációs sémák** feletti relációk

Termék (gyártó, modell, típus)

PC (modell, sebesség, memória, merevlemez, cd, ár)

Laptop (modell, sebesség, memória, merevlemez, képernyő, ár)

Nyomtató (modell, színes, típus, ár)

- Jelölje: T(gy, m, t)

PC(m, s, me, ml, ár)

L(m, s, me, ml, k, ár)

Ny(m, sz, t, ár)

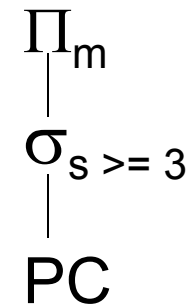
Megj.: a két típus attr.név
nem ugyanazt fejezi ki és
így $T \bowtie Ny$ természetes
összekapcsolásnál „zűr”

Példák átírásokra --- 2

a.) Melyek azok a PC modellek, amelyek sebessége legalább 3.00?

$\Pi_m(\sigma_{s \geq 3.00}(\text{PC}))$

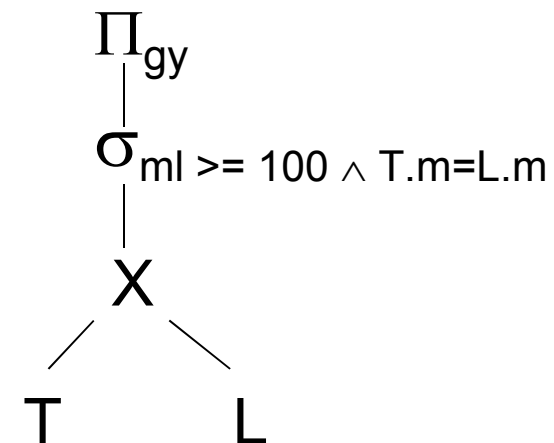
SELECT modell
FROM PC
WHERE sebesség ≥ 3 ;



b.) Mely gyártók készítenek legalább egy gigabájt méretű merevlemezzel rendelkező laptopot?

$\Pi_{gy}(\sigma_{ml \geq 100}(T \bowtie L))$

SELECT gyarto
FROM Termek T, Laptop L
WHERE merevlemez ≥ 100
AND T.modell=L.modell;

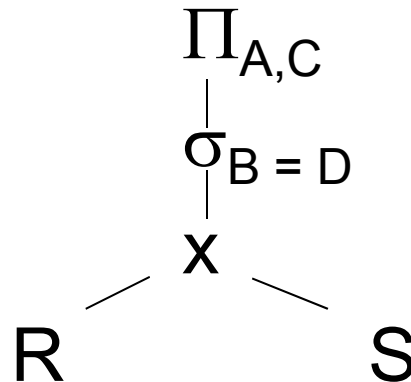


Példák átírásokra ---3

- **1.Példa:** Legyen $R(A,B)$ és $S(C,D)$

$$\Pi_{A,C} (\sigma_{B=D}(R \times S))$$

- Ehhez felrajzolva a **kiértékelő fát:**



- Átalakítjuk SQL lekérdezésre:

```
SELECT A, C  
FROM R, S  
WHERE B=D;
```

Példák átírásokra ---4

- **2.Példa:** Legyen $R(A,B)$, $S(C,D)$ és $T(E,F)$

$$\Pi_{A,C,E} [(\Pi_{A,C} (\sigma_{B=D} (R \times S))) \\ \bowtie (\Pi_{A,E,B} (\sigma_{B=F} (R \times T)))]$$

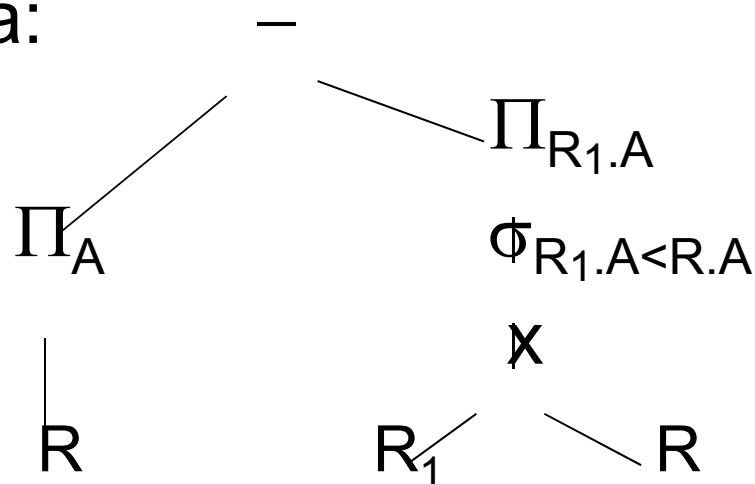
- Ehhez felrajzolva a **kiértékelő fát**
- **ÁTÍRVA** Datalog programmá (lépésenként, aztán egyszerűsítve felírható egyetlen szabállyal)

Példák átírásokra ---5

- **4.Példa:** Nézzük meg a maximum előállításának a kérdését! Legyen $R(A,B)$. **Feladat:** Adjuk meg $MAX(A)$ értékét!

$$\pi_A(R) - \pi_{R1.A}(\sigma_{R1.A < R.A}(\rho_{R1}(R) \times R))$$

- Kiértékelő fa:



Következik

- Rekurzió Datalogban és
- az SQL-1999 szabvány WITH RECURSIVE
- Oracle megoldások a gyakorlatban