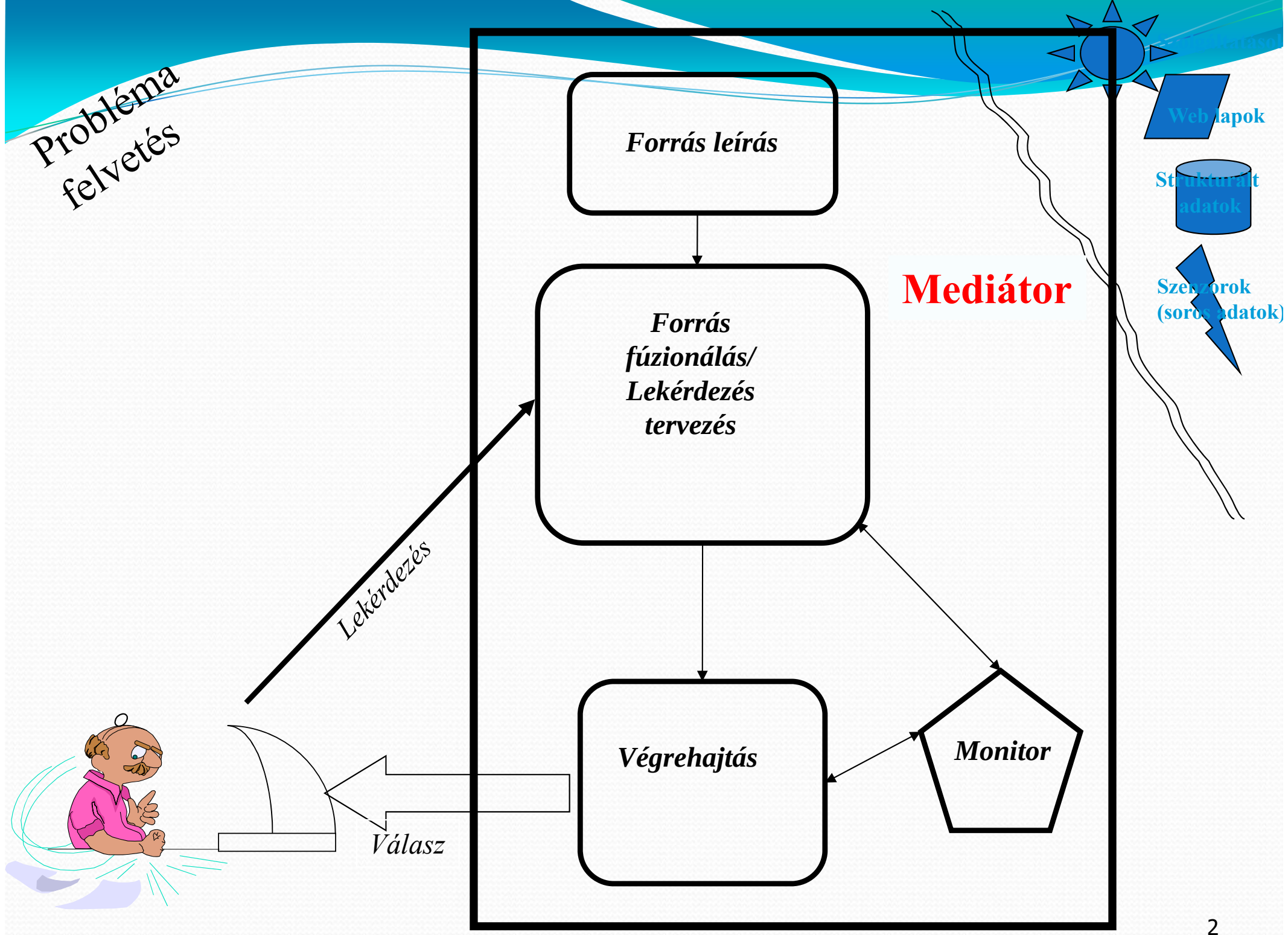


Információ integráció (Szemantikus Web megközelítés a másik irányból)

5. Előadás

Méréstechnika és Információs Rendszerek Tanszék
<https://www.mit.bme.hu/oktatas/targyak/vimiaco4>



- Felhasználói lekérdezések megfogalmazása a *mediált sémán*.
- Adatok tárolva *lokális sémában*.
- A tárolt információ (tartalom) ismerete alapján megfogalmazható a leképezés a sémák között.
- A mediátor alkalmazza a leképezést a felhasználói kérdés lefordítására a forrás lekérdezésekre.



Információ
menedzser

Válasz

Preferenciák
Elérési modell

Lekérdezés

Forrás leírás

Ontológiák,
Forrás és
szolgáltatás leírások

Teszt
lekérdezések

Forrás fúzionálás Lekérdezés tervezés

Számos cél együttese,
Szolgáltatások
kompozíciója,
Forrás minőség, átfedés

Végrehajtás

Kezel: forrás és hálózati
kapcsolatokat,
futtatási
bizonytalanságokat,
újratervezést

Forrás elérések

Újratervezési
kérdések

Statistikák frissítése

Monitor

Weblapok

Strukturált
adatok

Szenzorok
(soros
adatok)

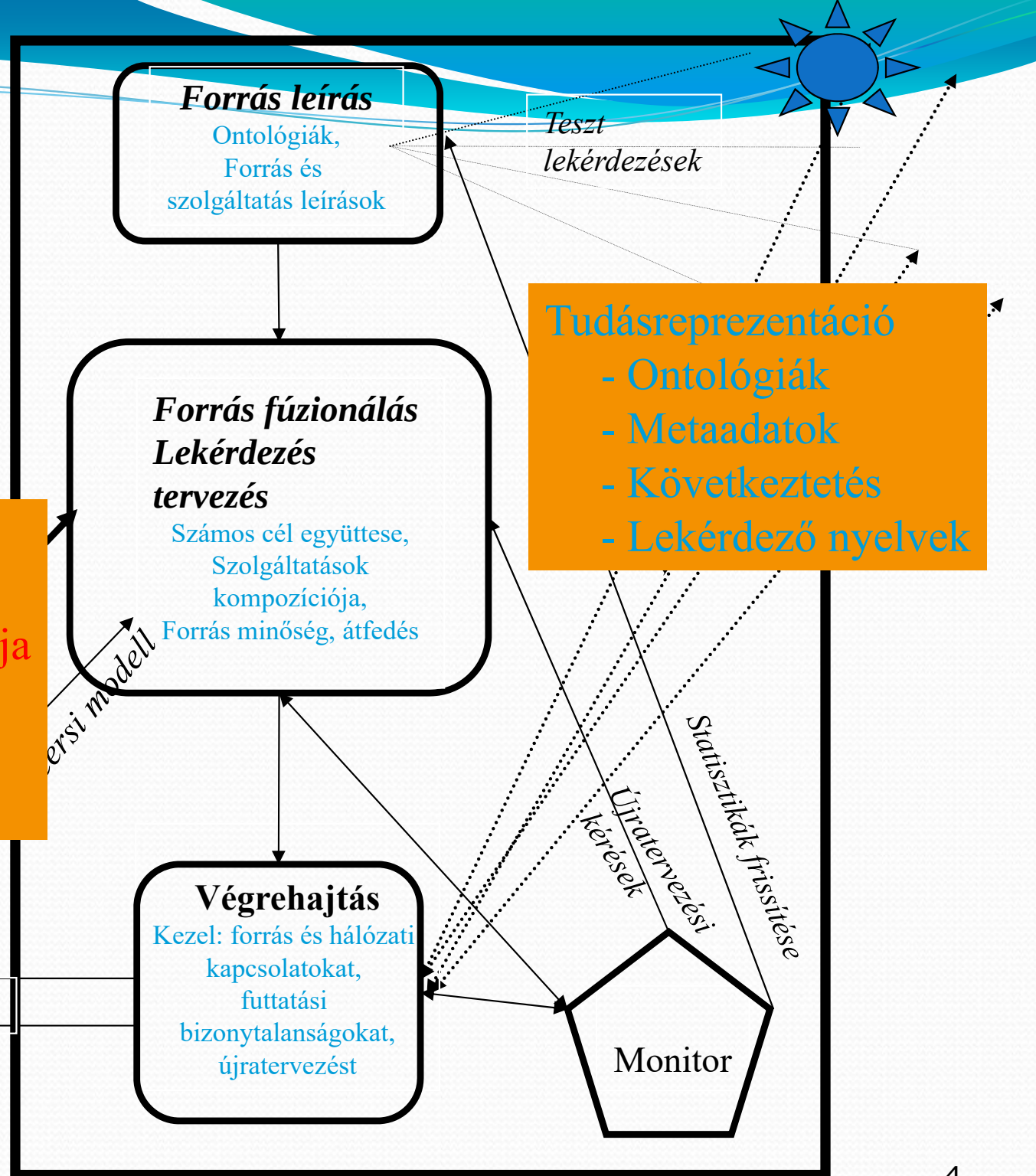
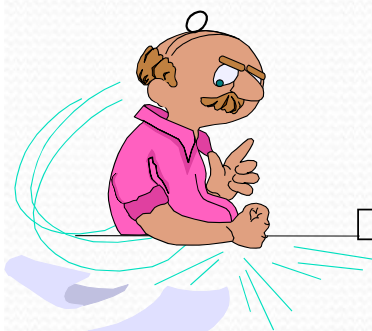
Hol az MI szerepe?

Tanulás/bányászás

- Forrás felkutatás
- Forrás statisztikák
- Wrapper tanulás

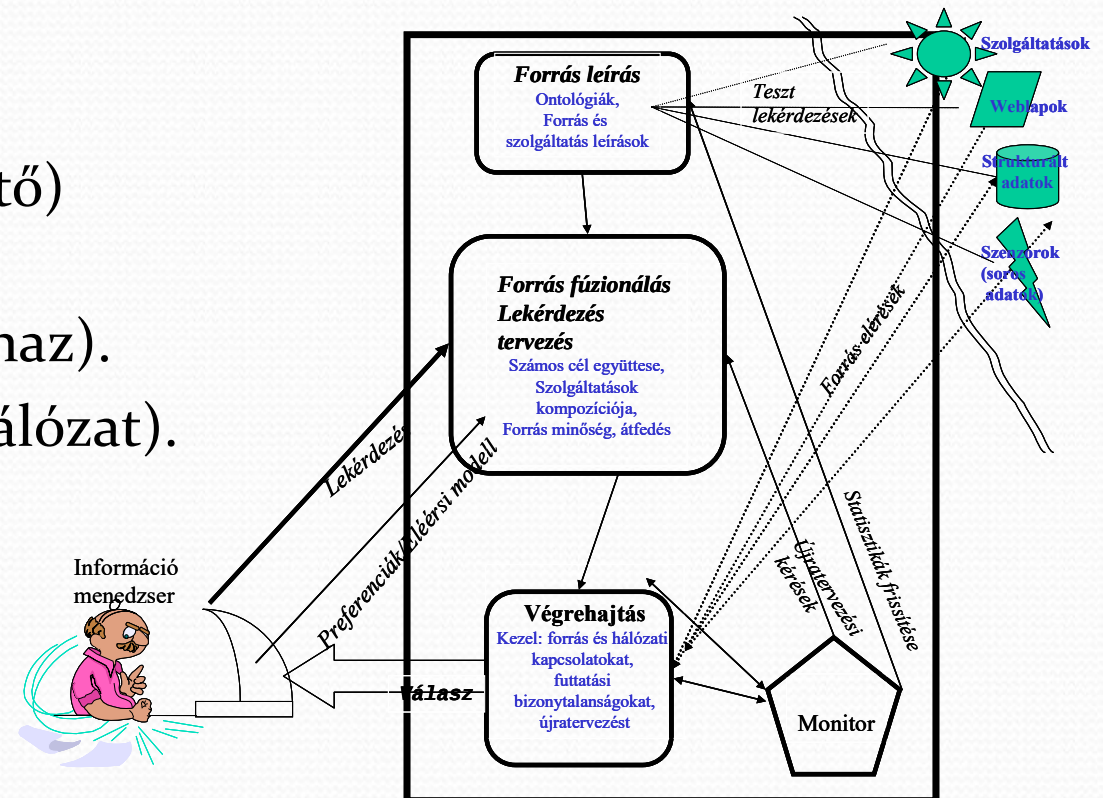
Automata tervezés

- Nyelvek tervezése
- Szolgáltatások kompozíciója
- Reaktív tervezés/
terv monitorozás



Forrás leírások

- Minden meta-adat információt tartalmaz a forrásokkal kapcsolatban:
 - Forrás tartalom logikai leírása (könyvek, új autók).
 - Forrás képességek (pl. SQL lekérdezés feltehető).
 - Forrás teljesség (pl. *minden* könyvet tartalmaz).
 - Fizikai jellemzők (forrás, hálózat).
 - Statisztikák az adatokról
 - Tükör források
 - Frissítési frekvencia.

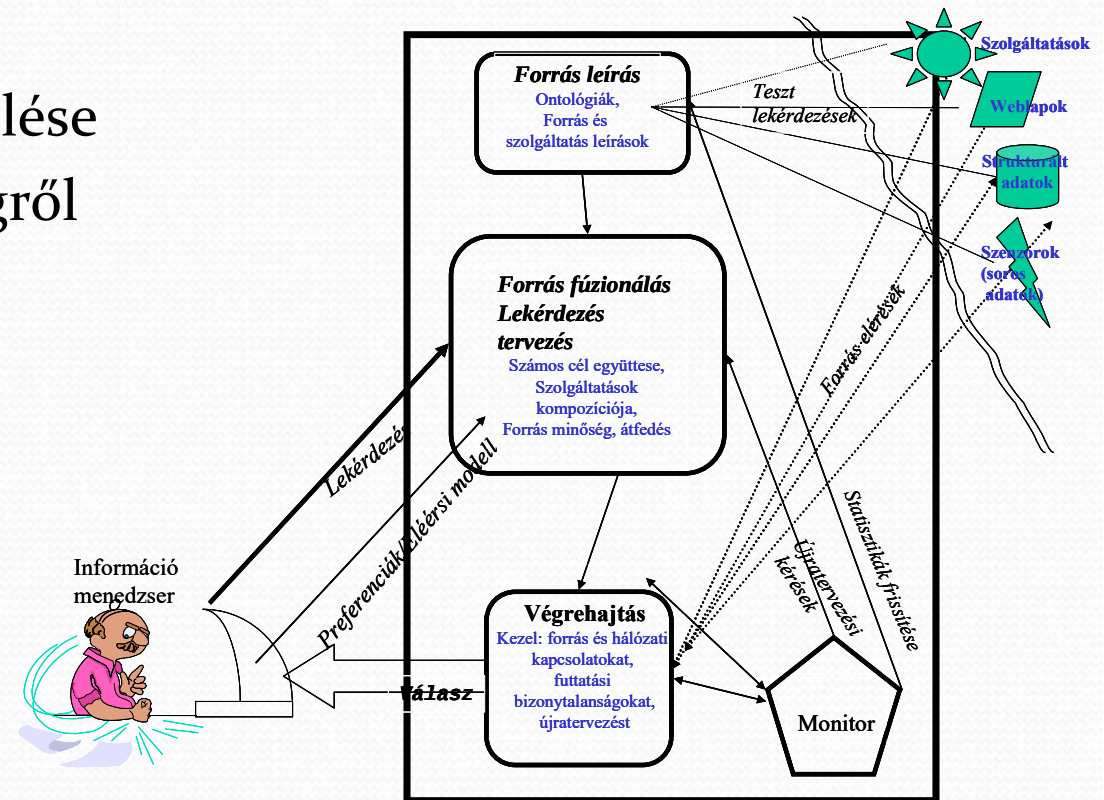


Forrás elérések

- Hogyan kapunk n-eseket
 - Számos forrás strukturálatlan adatokat tartalmaz
 - Bizonyos források inherensen strukturálatlanok, mások természetes nyelvi köntösben vannak
 - Vissza kell csomagolni az adatokat
 - Wrapper építés/információ kinyerés

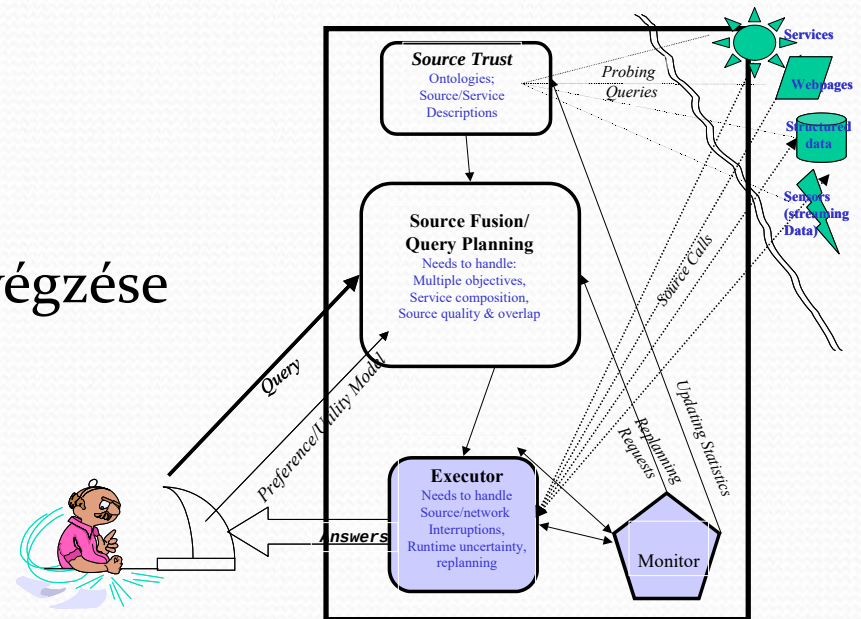
Forrás fúzió/ lekérdezés tervezés

- Feldolgozza a felhasználói lekérdezést és előállítja a végrehajtási tervet:
 - Költség és hatékonyság közti optimalizáció
 - Forrás elérési korlátok kezelése
 - Információ a forrásminőségről



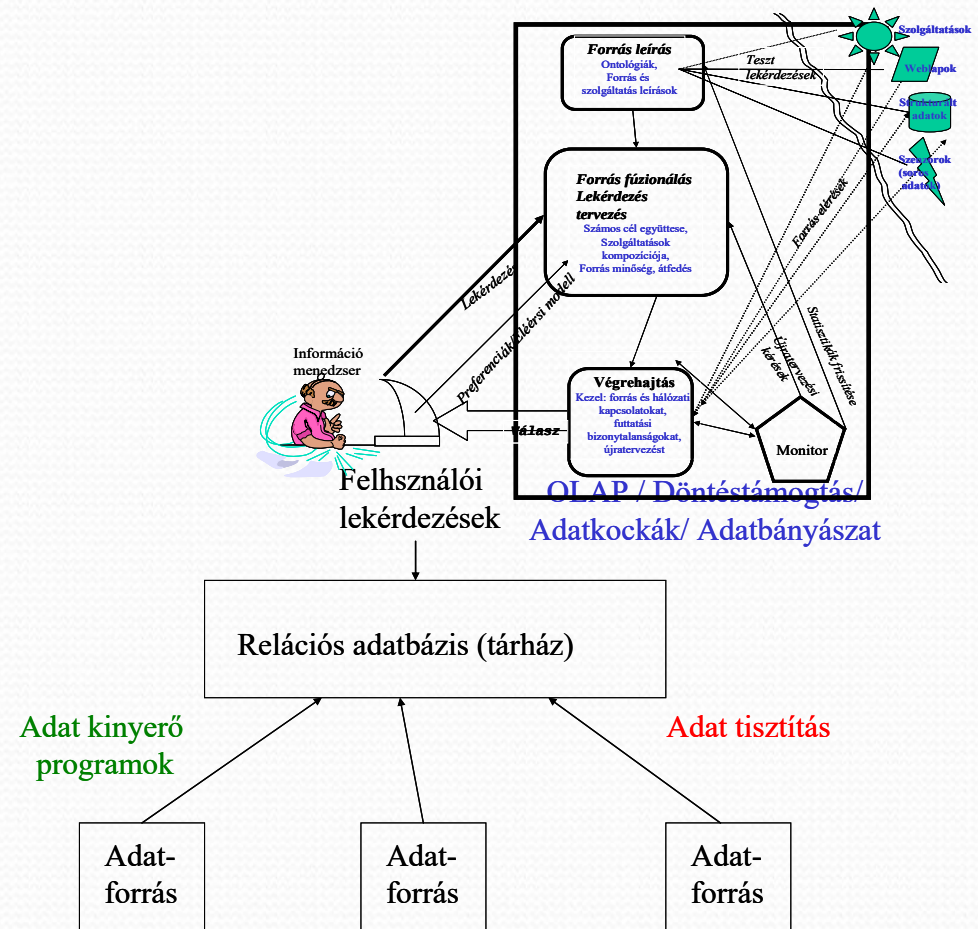
Monitoring/ Végrehajtás

- Lekérdezési terv alapján elvégzi az adatok begyűjtését a forrásokból:
 - Forrás késleltetések kezelése
 - Hálózati, tranziens kimaradások
 - Forrás elérési korlátok
 - Szükséges lehet újratervezések elvégzése
- Méretezés, tervezés:
 - Hány forrást kell elérni?
 - Mennyire autonómok ezek?
 - Van ismeretünk a forrásokról?
 - Strukturáltak az adatok?
 - Csak lekérdezés lehetséges vagy módosítás is?
 - Követelmények: pontosság, teljesség, teljesítmény, inkonzisztenciák kezelése
 - Zárt vagy nyílt világ feltételezés?



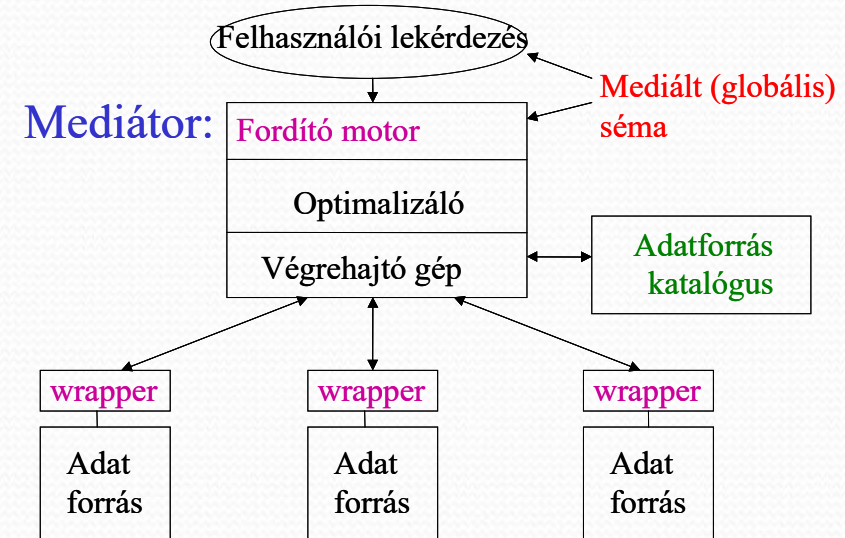
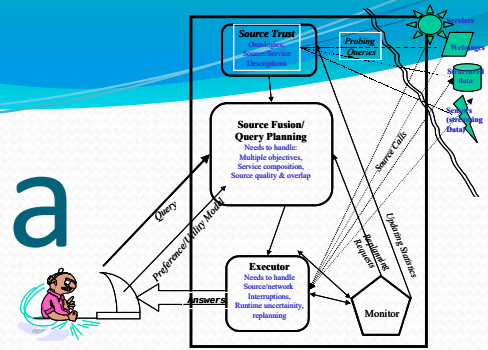
Kis forrás szám melletti integráció

- **Általában ad-hoc programozás:** speciális eset megvalósítása minden esetre, sok konzultáció.
- **Adattárházak:** minden adat periódikus feltöltése az adattárházba.
 - 6-18 hónap bevezetési idő
 - Operációs és döntéstámogatási RDBMS elválasztás. (nem csak adatintegrációra megoldás).
 - Teljesítmény jó,
 - adat lehet, hogy nem friss;
 - Rendszeres adattisztítás szükséges.



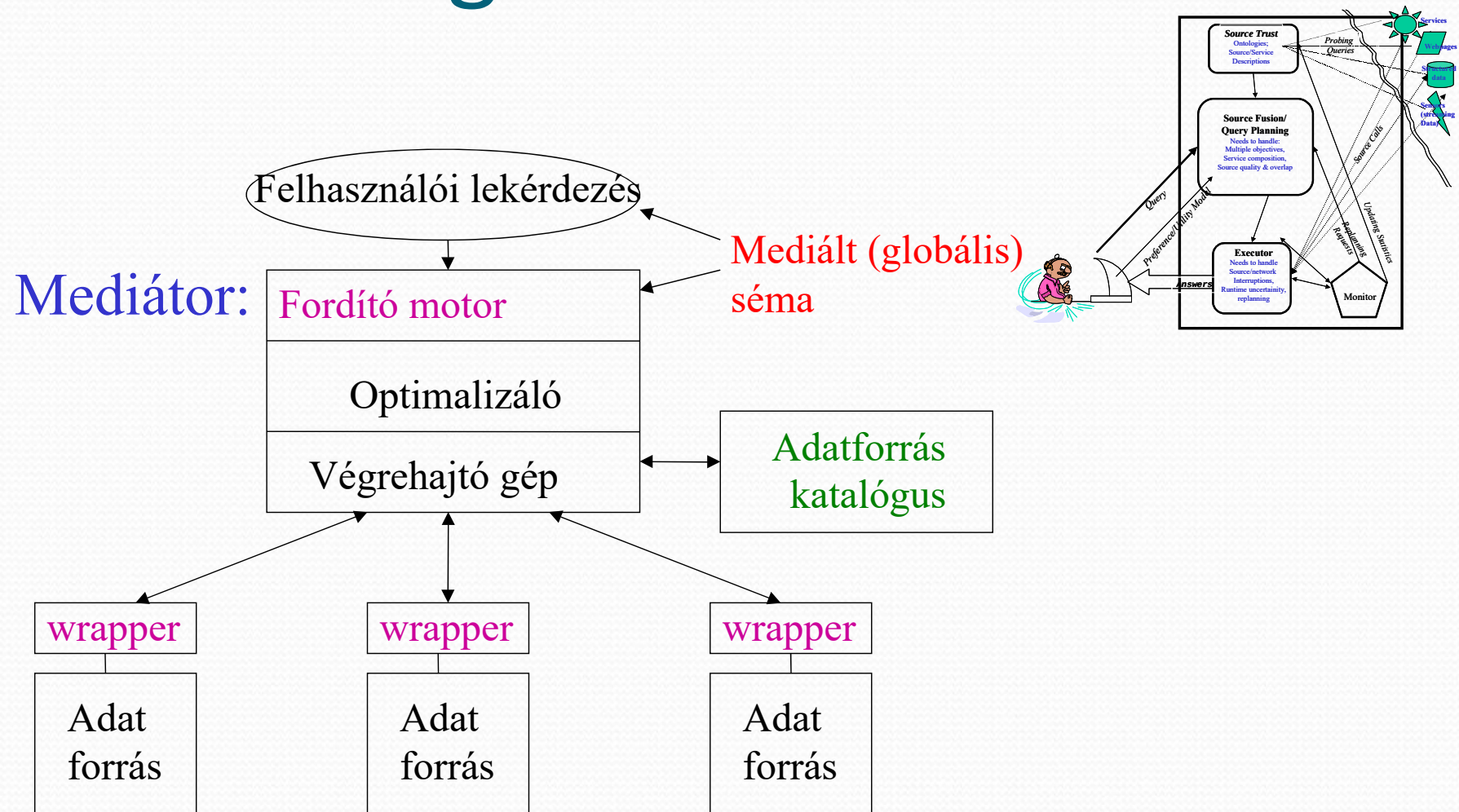
Virtuális integrációs séma

- Adatok a forrásokban maradnak
- Lekérdezés végrehajtásakor:
 - Releváns források meghatározása
 - Lekérdezés szétválasztása forrásokra vonatkozó lekérdezésekre.
 - Válaszok begyűjtése a forrásokból, és megfelelő kombinálása a válasz előállításához.
- Friss adatok
- A megoldás skálázható



Garlic [IBM], Hermes[UMD];Tsimmis, InfoMaster[Stanford]; DISCO[INRIA]; Information Manifold [AT&T]; SIMS/Ariadne[USC];Emerac/Havasu[ASU]

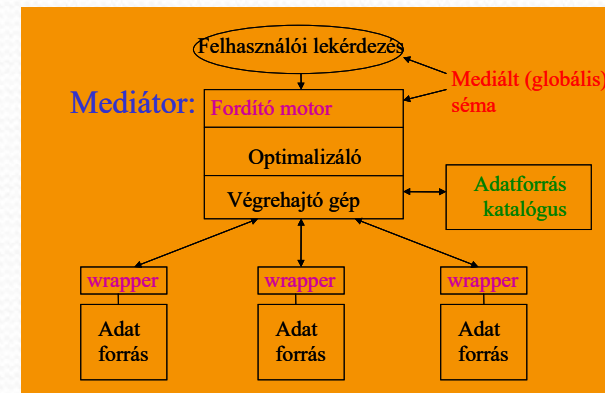
Virtuális integrátor architektúra



Források: relációs adatbázisok, weblapok, szövegek.

Forrás-mediátor relációs sémával szembeni elvárások

- **Kifejező erő:** hasonló adattartalommal rendelkező források megkülönböztetése, irreleváns források felismerése.
- **Egyszerű bővíthetőség:** tegyük könnyűvé források hozzáadását.
- **Fordítás/átalakítás:** felhasználói lekérdezés lefordítása forrásokon értelmezett lekérdezésekre hatékonyan és eredményesen.
- **Vesztességmentesség:** minden lehetséges adatelérés biztosítása



Lekérdezés átalakítás

- **Adott:**
 - Egy Q lekérdezés a mediátor sémára vonatkozóan
 - Adat források leírása
- **Létrehozandó:**
 - Egy Q' lekérdezés az adat forrásokra vonatkozóan, amely:
 - Q' csak *helyes válaszokat* ad a Q lekérdezéshez és
 - Q' minden lehetséges választ megtalál Q-hoz az elérhető forrásokból.

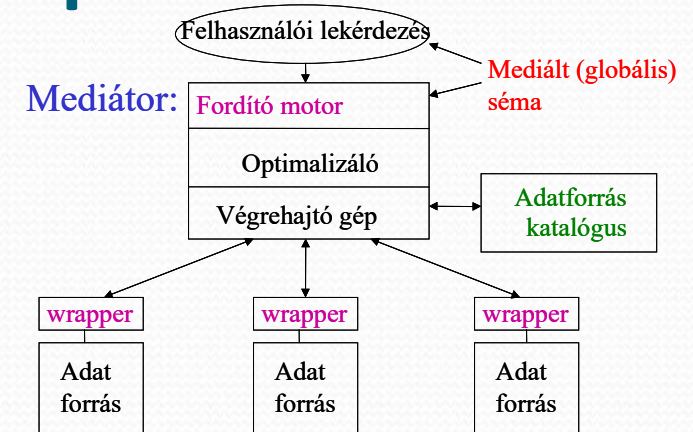
Fordítási/átfogalmazási probléma

- **Adott:**

- Egy Q lekérdezés a mediátor sémára vonatkozóan
- Adat források leírása

- **Létrehozandó:**

- Egy Q' lekérdezés az adat forrásokra vonatkozóan, amely:
 - Q' csak *helyes válaszokat* ad a Q lekérdezéshez és
 - Q' minden lehetséges választ megtalál Q-hoz az elérhető forrásokból.



Forrás és felhasználói sémák reláció leírásának megközelítései

- **Globális mediált sémák (Global-as-view, GAV):**
a mediált séma kifejezése a forrásokra vonatkozó nézetek relációjaként
- **Lokális mediált sémák (Local-as-view, LAV):**
forrás relációk kifejezése a mediált sémákon értelmezett relációkkal.
- Módszerek kombinációja...?

“Nézet” frissítés

```
CREATE VIEW Seattle-view AS  
  
SELECT buyer, seller, product, store  
FROM Person, Purchase  
WHERE Person.city = "Seattle" AND  
       Person.name = Purchase.buyer
```

A nézet felhasználása:

```
SELECT name, store  
FROM Seattle-view, Product  
WHERE Seattle-view.product = Product.name AND  
       Product.category = "shoes"
```

Mintapélda

- Hasonlítsuk össze a virtuális mediátor megközelítéseket film/mozi tárgyterületen
- Minta: Mediátor struktúra egy film adatbázishoz
 - Információk szolgáltatása filmekről, illetve mozi programról néhány forrás adatbázis és két mediált nézet felhasználásával

Globális mediált nézet

GAV (Global-As-View)

Mediált/felhasználói séma:

Filmek(cím, rendező, év, típus),
Műsor(mozi, cím, idő).

Mediált séma kifejezése
a forrásokra vonatkozó
nézetek relációjaként.

Create View Filmek AS

select * from S1

[S1(cím, rendező, év, típus)]

union

select * from S2

[S2(cím, rendező, év, típus)]

union

[S3(cím,rendező), S4(cím,év,típus)]

select S3.cím, S3.rendező, S4.év, S4.típus

from S3, S4

where S3.cím=S4.cím

GAV

Mediált/felhasználói séma:

Filmek(cím, rendező, év, típus),

Műsor(mozi, cím, idő).

Create View Filmek AS

select * from S1

union

select * from S2

union

select S3.cím, S3.rendező, S4.év, S4.típus

from S3, S4

where S3.cím=S4.cím

Mediált séma kifejezése
a forrásokra vonatkozó
nézetek relációjaként.

[S1(cím, rendező, év, típus)]

[S2(cím, rendező, év, típus)]

[S3(cím,rendező), S4(cím,év,típus)]

A mediátor séma relációk virtuális
nézetek a forrásrelációkon.

GAV: példa 2.

Mediált/felhasználói séma:

Filmek(cím, rendező, év, típus),
Műsor(mozi, cím, idő).

Mediált séma kifejezése
a forrásokra vonatkozó
nézetek relációjaként.

Create View Filmek AS

```
select * from S1
```

```
select cím, rendező, év, NULL
```

```
from S1
```

union

```
select cím, rendező, NULL, típus
```

```
from S2
```

[S1(cím,rendező,év)]

Null értékek

[S2(cím, rendező,típus)]

GAV: példa 2.

Mediált/felhasználói séma:

Filmek(cím, rendező, év, típus),

Műsor(mozi, cím, idő).

Forrás S4: S4(mozi, típus)

Mediált séma kifejezése
a forrásokra vonatkozó
nézetek relációjaként.

Create View Filmek AS

```
select NULL, NULL, NULL, típus  
from S4
```

Create View Műsor AS

```
select mozi, NULL, NULL  
from S4.
```

*De mit lehetne tenni, ha minket a vígjátékokat játszó
mozik érdekelnének?*

“Veszteséges medáció”

LAV: példa 1

Mediált/felhasználói séma:

Filmek(cím, rendező, év, típus),
Műsor(mozi, cím, idő).

Forrás séma kifejezése
a mediált nézeteken
értelmezett relációkként.

Create Source S1 AS

select * from Filmek

S1(cím, rendező, év, típus)

Create Source S3 AS

select cím, rendező from Filmek

S3(cím, rendező)

Create Source S5 AS

select cím, rendező, év

from Filmek

where év > 1960 AND típus="vígjáték"

S5(cím, rendező, év), év > 1960

A források “materializált nézetek”
a mediált sémák felett.

LAV: példa 1

Mediált/felhasználói séma:

Filmek(cím, rendező, év,
típus),

Műsor(mozi, cím, idő).

Create Source S4 AS

```
select mozi, típus
```

```
from Filmek m, Műsor s
```

```
where m.cím=s.cím
```

*Van remény a vígjátékokat játszó mozik
felderítésére!*

Forrás séma kifejezése
a mediált nézeteken
értelmezett relációkként.

S4(Mozi,Típus)

GAV vs. LAV

Mediált séma:

Filmek(**cím, rendező, év,**
típus),
Műsor(**mozi, cím, idő**).

Forrás S4: S4(mozi, típus)

Create View Filmek AS

```
select NULL, NULL, NULL, típus  
from S4
```

Create View Műsor AS

```
select mozi, NULL, NULL  
from S4.
```

*De mit lehetne tenni, ha minket a vígjátékokat játszó
mozik érdekelnének?*

Create Source S4 AS

```
select mozi, típus  
from Filmek m, Műsor s  
where m.cím=s.cím
```

Veszteséges mediáció

GAV

vs.

LAV

- Nem moduláris
 - Források hozzáadása módosítja a meglévő mediált séma definícióját
- Nehézkes lehet veszteségmentes mediátort készíteni.
- Lekérdezés átalakítás egyszerű
 - *Nézetek kibontását jelenti (polinomiális)*
 - Hierarchikus mediátor sémák létrehozása lehetséges
- Hatékony, ha
 - Kis számú, ritkán változó adatforrás van
 - Feladat teljesen ismert a mediátor tervezésekor (pl. vállalati adatintegráció)
 - Garlic, TSIMMIS, HERMES

- Moduláris—új forrás hozzáadása egyszerű
- Igen rugalmas – a lekérdező nyelv közvetlenül alkalmazható a források leírására
- Lekérdezés átalakítás bonyolult
 - Válaszokat a nézeteken keresztül kell előállítani (nem mindig megoldható)
- Hatékony, ha
 - Sok, kevésbé korrelált forrás
 - Források dinamikus hozzáadása és törlése
 - Information Manifold, InfoMaster, Emerac, Havasu

Lokális mediált nézetek átalakítása

- Adott nézetek egy halmaza $V_1, \dots, V_n$, és egy Q lekérdezés.

Megválaszolható-e a Q lekérdezés a $V_1, \dots, V_n$ nézetek felhasználásával?

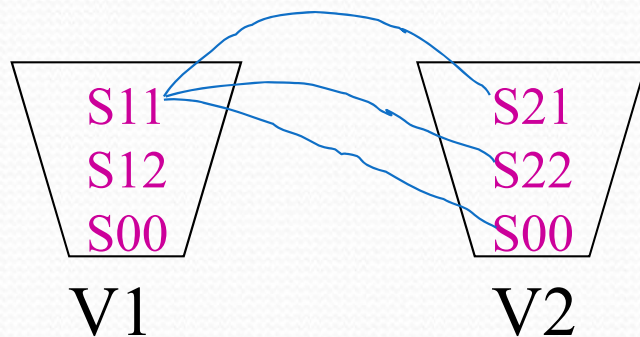
- A lekérdezéseket materializált nézeteken (forrásokon) futtatjuk végül!
- Megközelítések
 - Veder algoritmus (Bucket algorithm [Levy; 96])
 - Inverz szabályok algoritmus [Duschka, 99]
 - Hibrid algoritmusok
 - SV-Bucket [2001], MiniCon [2001]

Átalakítási algoritmusok

$Q(.) :- V1() \& V2()$

$S11() :- V1()$ $S12 :- V1()$
 $S21() :- V2()$ $S22 :- V2()$
 $S00() :- V1(), V2()$

Veder algoritmus



- Veder algoritmus
 - Vedrek kombinációjából előállított terv
 - Utána tartalmazási ellenőrzés

Inverz szabályok

$Q(.) :- V1() \& V2()$

$V1() :- S11()$

$V1() :- S12()$

$V1() :- S00()$

$V2() :- S21()$

$V2() :- S22()$

$V2() :- S00()$

- Inverz szabályok
 - Rész lekérdezések tervezés

Kitérő: Datalog

Deduktív adatbázisok

- **Alap gondolat:** relációk egy része az adatbázisban tárolt (extensional), más része a logikában (intensional).
- Relációkat predikátumokkal írjuk le.
- N-esek közti relációkat datalog szabályokkal írjuk le
 - (Horn klózok, függvényszimbólumok nélkül)
 - Lekérdezések megfelelnek egy datalog programnak

Datalog

- Csak egy másik lekérdező nyelv
- Tiszta elméleti alapok
- Közelebb a logikai megközelítéshez
- Könnyen elemezhető
- Relációs algebrában nem támogatott elemek (rekurzió).
- Nincs group by, order, bags, aggregate.

Datalog

- A Datalog egy nem procedurális, logika alapú lekérdező nyelv



- Nem pontos illeszkedések kezelése adatbázisokban.
- Kényelmesebb, olvashatóbb elemzés
- Kifejező erő: rekurzív lekérdezések.
- Negáció bevezetése
- (Rekurzió és negáció nélkül a kifejezőerő megegyezik a relációs adatbázisokéval.)

Datalog fogalmak

- Atomok
- Datalog szabályok, datalog programok
- Konjunktív lekérdezések
- Rekurzió
- Beépített predikátumok
- Negáció
- Szemantika: legkisebb azonosság (fix pont).

Datalog fogalmak

- EDB (extenzionális adatbázis) predikátumok
 - olyan relációk, amelyek minden eleme adatbázisban tárolt, csak a szabályok törzsében fordulhatnak elő
 - Példa: $SzülőDB(SzNév, GyNév)$
- IDB (intenzionális adatbázis) predikátumok –
 - Reláció tartalmaz nem adatbázis elem változókat, pl. nézet definíciók, lekérdezések következtetett elemei; szabályok törzsében és fejében is előfordulhat
 - Példa:
$$ÖsszesSzülő(SzNév, GyNév) :- SzülőDB(SzNév, Sz_1), \\ ÖsszesSzülő(Sz_1, Gynév)$$

Predikátumok és atomok

- Relációk reprezentálása predikátumokkal
- N-esek reprezentálása atomokkal:
Vásárlás(“Péter”, “Alle- Nike bolt”, “Nike Air”, 2/2/16)
- Aritmetika:
 $X < 100, \quad X+Y+5 > Z/2$
- Negáció:
NOT Termék(“Nike Air”, 100, “Microsoft”)

Datalog szabályok értelmezése

Az EDB tényeiből kiindulva az IDB tényeinek iteratív levezetése.

Algoritmus

Ismételjük az alábbi lépéseket addig, mindaddig újabb tények levezethetők:

- Vegyük a szabályok törzsében szereplő változók összes lehetséges illesztését (hozzárendelését) az adatbázisban felelhető elemekkel;
- Ha minden atom a szabály törzsében igazgá tehető egy hozzárendeléssel

akkor

Adjuk hozzá a megfelelő n -est a szabály fejében szereplő relációhoz

Példa a Termék(név, gyártó, ár) tábla alapján:

DrágaTermék(X,Y) :- Termék(X,Y,P) , $P > 100$

Biztonságos szabályok, lekérdezések

Véges lépésszámmal kielégíthető szabályok, lekérdezések feltétele a datalog-ban (véges méretű adatbázisokra):

Minden a fej részben található változónak meg kell jelenni egy nem negált relációban a szabály törzsében.

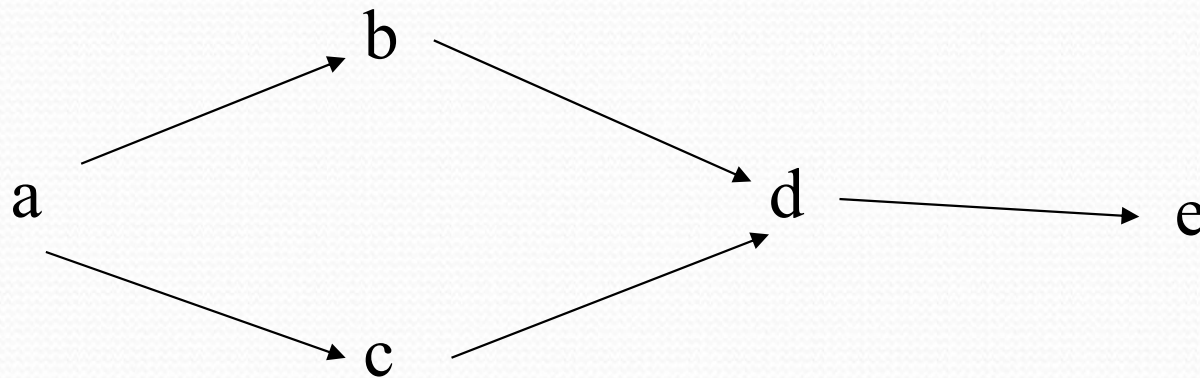
$Q(X,Y,Z) :- R1(X,Y) \ \& \ X < Z$ *nem biztonságos*

$Q(X,Y,Z) :- R1(X,Y) \ \& \ NOT \ R2(X,Y,Z)$ *nem biztonságos*

Tranzitív lezárás

Reprezentáljunk egy gráfot éleivel: $Edge(X,Y)$:

$Edge(a,b)$, $Edge(a,c)$, $Edge(b,d)$, $Edge(c,d)$, $Edge(d,e)$



Írjunk lekérdezést:

Keressük az a-ból elérhető csúcsokat.

Datalog megoldás

Rekurzív lekérdezés:

$Path(X, Y) \text{ :- } Edge(X, Y)$

$Path(X, Y) \text{ :- } Path(X, Z), Path(Z, Y).$

.

Rekurzív lekérdezés kiértékelése

$Path(X, Y) \text{ :- } Edge(X, Y)$

$Path(X, Y) \text{ :- } Path(X, Z), Path(Z, Y).$

Szemantika: kiértékelni a szabályokat egy fix pontig:

Iteration #0: Edge: $\{(a,b), (a,c), (b,d), (c,d), (d,e)\}$

Path: $\{\}$

Iteration #1: Path: $\{(a,b), (a,c), (b,d), (c,d), (d,e)\}$

Iteration #2: Path gets the new tuples:

$(a,d), (b,e), (c,e)$

Iteration #3: Path gets the new tuple:

(a,e)

Iteration #4: Nothing changes -> We stop.

Iterációk száma függ az adatoktól!

Maximális tartalmazás

- Lekérdezési tervnek helyesnek és teljesnek kell lennie
 - Helyes akkor, ha az új tervet tartalmazza az eredeti lekérdezés (Például minden n -es válasz az eredeti lekérdezésben is megtalálható)
 - Teljesség?
 - Eredeti adatbázis megközelítés teljességre törekszik
 - Itt a megközelítés: maximális tartalmazás!

P tartalmazza Q -t if $P \models Q$
(exponenciális futási idejű algoritmusok,
még konjunktív lekérdezésekre is.)

Tartalmazás (lekérdezések)

- Legyen $Q_1(.) :- B_1(.)$ $Q_2(.) :- B_2(.)$
- $Q_1 \subseteq Q_2$ (“tartalmaz”) ha a Q_1 -re kapott válasz részhalmaza a Q_2 -re kapott válasznak tetszőleges adatbázisra,
 - Igaz, ha $B_1(x) \models B_2(x)$
- Ha adott egy Q lekérdezés, és egy Q_1 , átírt lekérdezési terv, akkor:
 - Q_1 **helyes** lekérdezési terv ha Q_1 -t tartalmazza Q
 - Q_1 **teljes** lekérdezési terv, ha Q -t tartalmazza Q_1
 - Q_1 egy **maximálisan tartalmazó** lekérdezési terv, ha nem létezik olyan Q_2 , amelyik helyes és Q_1 -t tartalmazza Q_2

Tartalmazás ellenőrzése

- Tekintsünk két lekérdezést: $Q_1(.) :- B_1(.)$ $Q_2(.) :- B_2(.)$
 - $Q_1 \subseteq Q_2$ („tartalmazó”), ha Q_1 lekérdezésre kapott minden válasz (n-es) részhalmaza Q_2 -nek
 - A tartalmazás áll, ha $B_1(x) \models B_2(x)$
 - (a vonzat reláció helyessége általánosságban nem eldönthető...)
 - Konjunktív lekérdezések (select/project/join lekérdezések, kényszerek nélkül) ellenőrzését az egyes kifejezések közötti leképezések megadhatóságával ellenőrizzük (exponenciális időigényű algoritmus)
 - m legyen egy (tartalmazási) leképezés $\text{Vars}(Q_2)$ változókról $\text{Vars}(Q_1)$ változókra, ha
 - ★ m leképezi Q_2 törzsében (feltétel részében) található minden rész célját egy Q_1 törzsében található rész célra
 - 🕒 m leképezi Q_2 fej részét (következmény részét) Q_1 fej részére
- Eg: $Q_1(x,y) :- R(x), S(y), T(x,y)$ $Q_2(u,v) :- R(u), S(v)$
Kapcsolódó leképezés: $[u/x ; v/y]$

Inverz szabályok módszere – példa

Mediált (saját) nézetek:

Önlab(hallgató, tanszék), Tárgyfelvétel(hallgató, kurzus).

S_1 távoli adatforrás, amelyet nézetként definiálunk a mi mediált nézeteink felett:

$S_1(\text{tanszék}, \text{kurzus}) :- \text{Önlab}(\text{hallgató}, \text{tanszék}), \text{Tárgyfelvétel}(\text{hallgató}, \text{kurzus})$

Létrehozzunk egy inverz szabályt a nézet minden egyes konjunktjára:

$\text{Önlab}(\text{fi}(\text{T}, \text{kurzus}), \text{tanszék}) :- S_1(\text{tanszék}, \text{K})$

$\text{Tárgyfelvétel}(\text{fi}(\text{tanszék}, \text{K}), \text{kurzus}) :- S_1(\text{T}, \text{kurzus})$

(Minden egyes) **Z** ismeretlenhez, amely több kifejezésben is szerepel, hozzunk létre egy funkcionális kifejezést: pl. $\text{fi}(\text{tanszék}, \text{hallgató})$

Inverz szabályok módszere – példa

Mediált (saját) nézetek:

Önlab(hallgató, tanszék), Tárgyfelvétel(hallgató, kurzus).

S_1 távoli adatforrás, amelyet nézetként definiálunk a mi mediált nézeteink felett:

$S_1(\text{tanszék}, \text{kurzus}) :- \text{Önlab}(\text{hallgató}, \text{tanszék}), \text{Tárgyfelvétel}(\text{hallgató}, \text{kurzus})$

Létrehozzunk egy inverz szabályt a nézet minden egyes konjunktjára:

$\text{Önlab}(\text{fi}(\text{T}, \text{kurzus}), \text{tanszék}) :- S_1(\text{tanszék}, \text{K})$

$\text{Tárgyfelvétel}(\text{fi}(\text{tanszék}, \text{K}), \text{kurzus}) :- S_1(\text{T}, \text{kurzus})$

Lekérdezés: $q(\text{tanszék}) :- \text{Önlab}(\text{H}, \text{tanszék}), \text{Tárgyfelvétel}(\text{H}, \text{„Adatbázisok”})$

S_1 tartalmazza a következő ketteseket:

$< (\text{TMIT}, \text{„Adatbázisok”}), (\text{MIT}, \text{„Adatbázisok”}), (\text{MIT}, \text{„Mesterséges intelligencia”})$

Inverz szabályok módszere – példa

Önlab($fi(T, kurzus)$, tanszék) :- $S_1(tanszék, K)$

Tárgyfelvétel($fi(tanszék, K)$, kurzus) :- $S_1(T, kurzus)$

Lekérdezés: $q(tanszék)$:- $Önlab(H, tanszék), Tárgyfelvétel(H, „Adatbázisok”)$

S_1 tartalmazza a következő ketteseket:

$< (TMIT, „Adatbázisok”), (MIT, „Adatbázisok”), (MIT, „Mesterséges intelligencia”)$

Eredmény:

$Önlab : < (fi(TMIT, „Adatbázisok”), „Adatbázisok”),$
 $(fi(MIT, „Adatbázisok”), „Adatbázisok”),$
 $(fi(MIT, „Mesterséges intelligencia”), „Mesterséges intelligencia”) >$

$Tárgyfelvétel: < (fi(TMIT, „Adatbázisok”), TMIT),$
 $(fi(MIT, „Adatbázisok”), MIT),$
 $(fi(MIT, „Mesterséges intelligencia”), MIT) >$

Válasz: TMIT, MIT

Forrás elérési korlátok

- A források nem feltétlenül relációs adatbázisok
 - Jogosultsági korlátok
 - Korlátos elérési minták
 - (Pl. telefonkönyv lekérdezése)
 - Korlátos kiszolgáló erőforrás
 - (Csak attribútumokon értelmezett szűréseken keresztül érhetőek el adatok.)
- Elérési korlátozások modellezhetők:
 - b: kötelezően megadandó keresési attribútum
- - f: szabadon elérhető attribútum

Elérési korlátok – rekurzív algoritmusok

Create Source S₁ as
SELECT *
from Hivatkozások
adott cikkhez

Create Source S₂ as
SELECT cikk
from ASU-Cikkek

Create Source S₃ as
SELECT cikk
from DíjazottCikkek
adott cikkhez

Query: SELECT * from
DijazottCikkek

S₁^{bf}(p₁,p₂) :- Hivatkozások(p₁,p₂)

S₂(p) :- Asu-C(p)

S₃^b(p) :- DíjC(p)

Q(p) :- DíjC(p)

Díjc(p) :- ÖsszC(p), S₃^b(p)

Asu-C(p) :- S₂(p)

Hivatkozások(p₁,p₂) :- ÖsszC(p₁), S₁^{bf}(p₁,p₂)

ÖsszC(p) :- S₂(p)

ÖsszC(p) :- ÖsszC(p₁), S₁(p₁,p)

Veder algoritmus

- A Q lekérdezés minden rész célját lefedjük releváns nézetekkel
- Készítünk egy listát (veder) azokból a nézetekből, amelyek adhatnak eredményt az adott rész célhoz
- Megvizsgáljuk a vedrekben listázott nézetek kombinációját
- Nem feltétlenül minden kombináció megfelelő
- Megtartjuk a megfelelő nézetek kombinációit és minimalizáljuk a megoldást
- Elhagyjuk a redundáns rész célokat
- A megoldás a megmaradó lekérdezések uniója

Veder algoritmus példa

Mediált (saját) nézetek:

$\text{reg}(\text{Std}, \text{Crs}, \text{Qtr})$, $\text{course}(\text{Crs}, \text{Title})$, $\text{teaches}(\text{Prof}, \text{Crs}, \text{Qtr})$

S_1 , S_2 , S_3 , S_4 távoli adatforrások, amelyeket nézetként definiálunk a mi mediált nézeteink felett:

$S_1(\text{Std}, \text{Crs}, \text{Qtr}, \text{Title})$:- $\text{reg}(\text{Std}, \text{Crs}, \text{Qtr})$, $\text{course}(\text{Crs}, \text{Title})$,
 $\text{Crs} \geq 500$, $\text{Qtr} \geq \text{Aut98}$

$S_2(\text{Std}, \text{Prof}, \text{Crs}, \text{Qtr})$:- $\text{reg}(\text{Std}, \text{Crs}, \text{Qtr})$, $\text{teaches}(\text{Prof}, \text{Crs}, \text{Qtr})$

$S_3(\text{Std}, \text{Crs})$:- $\text{reg}(\text{Std}, \text{Crs}, \text{Qtr})$, $\text{Qtr} \leq \text{Aut94}$

$S_4(\text{Prof}, \text{Crs}, \text{Title}, \text{Qtr})$:- $\text{reg}(\text{Std}, \text{Crs}, \text{Qtr})$, $\text{course}(\text{Crs}, \text{Title})$,
 $\text{teaches}(\text{Prof}, \text{Crs}, \text{Qtr})$, $\text{Qtr} \leq \text{Aut97}$

$q(S, C, P)$:- $\text{teaches}(P, C, Q)$, $\text{reg}(S, C, Q)$, $\text{course}(C, T)$,
 $C \geq 300$, $Q \geq \text{Aut95}$

1. lépés: Minden rész célhoz összegyűjtjük a releváns nézeteket

Veder algoritmus példa

$S_1(\text{Std}, \text{Crs}, \text{Qtr}, \text{Title}) :- \text{reg}(\text{Std}, \text{Crs}, \text{Qtr}), \text{course}(\text{Crs}, \text{Title}),$

$\text{Crs} \geq 500, \text{Qtr} \geq \text{Aut98}$

$S_2(\text{Std}, \text{Prof}, \text{Crs}, \text{Qtr}) :- \text{reg}(\text{Std}, \text{Crs}, \text{Qtr}), \text{teaches}(\text{Prof}, \text{Crs}, \text{Qtr})$

$S_3(\text{Std}, \text{Crs}) :- \text{reg}(\text{Std}, \text{Crs}, \text{Qtr}), \text{Qtr} \leq \text{Aut94}$

$S_4(\text{Prof}, \text{Crs}, \text{Title}, \text{Qtr}) :- \text{reg}(\text{Std}, \text{Crs}, \text{Qtr}), \text{course}(\text{Crs}, \text{Title}),$

$\text{teaches}(\text{Prof}, \text{Crs}, \text{Qtr}), \text{Qtr} \leq \text{Aut97}$

$q(S, C, P) :- \text{teaches}(P, C, Q), \text{reg}(S, C, Q), \text{course}(C, T),$

$C \geq 300, Q \geq \text{Aut95}$

$P \rightarrow \text{Prof}, C \rightarrow \text{Crs}, Q \rightarrow \text{Qtr}$

Buckets

teaches	reg	course
S2		
S4		

Megjegyzés: Itt az aritmetikai predikátumok nem okoznak problémát

Veder algoritmus példa

$S_1(\text{Std}, \text{Crs}, \text{Qtr}, \text{Title}) :- \text{reg}(\text{Std}, \text{Crs}, \text{Qtr}), \text{course}(\text{Crs}, \text{Title}),$

$\text{Crs} \geq 500, \text{Qtr} \geq \text{Aut98}$

$S_2(\text{Std}, \text{Prof}, \text{Crs}, \text{Qtr}) :- \text{reg}(\text{Std}, \text{Crs}, \text{Qtr}), \text{teaches}(\text{Prof}, \text{Crs}, \text{Qtr})$

$S_3(\text{Std}, \text{Crs}) :- \text{reg}(\text{Std}, \text{Crs}, \text{Qtr}), \text{Qtr} \leq \text{Aut94}$

$S_4(\text{Prof}, \text{Crs}, \text{Title}, \text{Qtr}) :- \text{reg}(\text{Std}, \text{Crs}, \text{Qtr}), \text{course}(\text{Crs}, \text{Title}),$

$\text{teaches}(\text{Prof}, \text{Crs}, \text{Qtr}), \text{Qtr} \leq \text{Aut97}$

$q(S, C, P) :- \text{teaches}(P, C, Q), \text{reg}(S, C, Q), \text{course}(C, T),$

$C \geq 300, Q \geq \text{Aut95}$

Buckets

teaches	reg	course
S2	S1	
S4	S2	

$S \rightarrow \text{Std}, C \rightarrow \text{Crs}, Q \rightarrow \text{Qtr}$

Megjegyzés: S_3 nem ad megoldást, az aritmetikai predikátumok nem kielégíthetők

S_4 nem ad megoldást: S nem szerepel a S_4 kimenetében

Veder algoritmus példa

$S_1(\text{Std}, \text{Crs}, \text{Qtr}, \text{Title}) :- \text{reg}(\text{Std}, \text{Crs}, \text{Qtr}), \text{course}(\text{Crs}, \text{Title}),$

$\text{Crs} \geq 500, \text{Qtr} \geq \text{Aut98}$

$S_2(\text{Std}, \text{Prof}, \text{Crs}, \text{Qtr}) :- \text{reg}(\text{Std}, \text{Crs}, \text{Qtr}), \text{teaches}(\text{Prof}, \text{Crs}, \text{Qtr})$

$S_3(\text{Std}, \text{Crs}) :- \text{reg}(\text{Std}, \text{Crs}, \text{Qtr}), \text{Qtr} \leq \text{Aut94}$

$S_4(\text{Prof}, \text{Crs}, \text{Title}, \text{Qtr}) :- \text{reg}(\text{Std}, \text{Crs}, \text{Qtr}), \text{course}(\text{Crs}, \text{Title}),$

$\text{teaches}(\text{Prof}, \text{Crs}, \text{Qtr}), \text{Qtr} \leq \text{Aut97}$

$q(S, C, P) :- \text{teaches}(P, C, Q), \text{reg}(S, C, Q), \text{course}(C, T),$

$C \geq 300, Q \geq \text{Aut95}$

$C \rightarrow \text{Crs}, T \rightarrow \text{Title}$

Buckets

teaches	reg	course
S2	S1	S1
S4	S2	S4

Veder algoritmus példa

2. lépés:

- Nézetek összes kombinációját figyelembe kell venni, minden vederből egy elemet véve
- Aritmetikai predikátumok ellenőrzése a kombinációra
 - Például két nézet lehet átfedő vagy inkonzisztens
- Eredmény átírás= a megmaradók uniója

Lekérdezés átírás1:

teaches	reg	course
S2	S1	S1
S4	S2	S4

$q1(S,C,P) :- S2(S',P,C,Q), S1(S,C,Q,T'), S1(S'',C,Q',T)$

- Nincs probléma az aritmetikai predikátumokkal(nincs a S2-ben)
- Minimális-e a megoldás

Veder algoritmus példa

Átírás kibontása 1:

$q1'(S,C,P) :- r(S',C,Q), t(P,C,Q), r(S,C,Q), c(C,T'), r(S'',C,Q'),$
 $c(C,T), C \geq 500, Q \geq \text{Aut98}, C \geq 500, Q' \geq \text{Aut98}$

- Fekete r-ek leképezhetőek a zöld r-ekre:

$S' \rightarrow S, S'' \rightarrow S, Q' \rightarrow Q$

- Fekete c leképezhető a zöld c-re:

leképezés kibővíthető: $T \rightarrow T'$

Minimális kibontása az átírásnak1:

$q1m'(S,C,P) :- t(P,C,Q), r(S,C,Q), c(C,T'), C \geq 500, Q \geq \text{Aut98}$

Minimális átírás 1:

$q1m(S,C,P) :- S2(S',P,C,Q), S1(S,C,Q,T')$

Veder algoritmus példa

Lekérdezés átírás 2:

teaches	reg	course
S2	S1	S1
S4	S2	S4

$q2(S,C,P) :- \textcolor{blue}{S2}(S',P,C,Q), \textcolor{blue}{S1}(S,C,Q,T'), \textcolor{blue}{S4}(P',C,T,Q')$

$q2'(S,C,P) :- r(S',C,Q), t(P,C,Q), r(S,C,Q),$
 $r(S,C,Q), c(C,T'), C \geq 500, Q \geq \text{Aut98},$
 $r(S'',C,Q'), c(C,T), t(P',C,Q'), Q' \leq \text{Aut97}$

- A kombináció nem kielégíthető: Nézzük meg **S1** és **S4** kombinációját

Lekérdezés átírás 3:

teaches	reg	course
S2	S1	S1
S4	S2	S4

$q3(S,C,P) :- \textcolor{blue}{S2}(S',P,C,Q), \textcolor{blue}{S2}(S,P',C,Q), \textcolor{blue}{S4}(P'',C,T,Q')$

Veder algoritmus példa

Lekérdezés átírás kibontása 3:

$q3'(S,C,P) :- r(S',C,Q), t(P,C,Q), r(S,C,Q), t(P',C,Q), r(S'',C,Q'),$
 $c(C,T), t(P'',C,Q'), Q' \leq \text{Aut97}$

- A zöld részcélok lefedhetők a feketékkel a következő megfeleltetésekkel: $S' \rightarrow S, S'' \rightarrow S, P' \rightarrow P, P'' \rightarrow P, Q' \rightarrow Q$

Minimális átírás3:

$q3m(S,C,P) :- S2(S,P,C,Q), S4(P,C,T,Q)$

Tehát két átírás maradt.

Maximális tartalmazó átírás:

$$q' = q1m \cup q3m$$

Veder algoritmus példa 2.

Lekérdezés:

$q(X) \text{ :- cites}(X,Y), \text{ cites}(Y,X), \text{ sameTopic}(X,Y)$

Nézetek (adatforrások):

$S_4(A) \text{ :- cites}(A,B), \text{ cites}(B,A)$

$S_5(C,D) \text{ :- sameTopic}(C,D)$

$S_6(F,H) \text{ :- cites}(F,G), \text{ cites}(G,H), \text{ sameTopic}(F,G)$

Buckets

cites	cites	sameTopic
S4	S4	S5
S6	S6	S6

Megjegyzés: S_4 -et kétszer szerepeltetni a vedrekben?