Modelul relațional / Dependențe

Nicolae-Cosmin Vârlan

October 30, 2017

Elemente ale modelului relațional

- U mulțime de atribute: $U = \{A_1, A_2, \ldots, A_n\}$;
- $dom(A_i)$ - domeniul valorilor atributului A_i;

Definim *uplu* peste U ca fiind funcția:

$$\varphi : U \to \bigcup_{1 \leq i \leq n} dom(A_i) \quad \text{a.i. } \varphi(A_i) \in dom(A_i), 1 \leq i \leq n$$

Fie valorile v_i astfel încât $v_i = \varphi(A_i)$.

Notăm cu $\{A_1 : v_1,\ A_2 : v_2,\ \ldots,\ A_n : v_n\}$ asocierea dintre atributele existente în U și valorile acestora. In cazul în care sunt considerate mulțimi ordonate (de forma $(A_1, A_2, \ldots, A_n)$), notația va fi de forma: $(v_1, v_2, \ldots, v_n)$.

Elemente ale modelului relațional

Consideram mulțimea ordonată $(A_1, A_2, \dots A_n)$. Pentru orice uplu φ, există vectorul $(v_1, v_2, \dots v_n)$ a.i. $\varphi(A_i) = v_i$, $1 \leq i \leq n$.

Pentru un vector $(v_1, v_2, \dots v_n)$ cu $v_i \in dom(A_i)$, $1 \leq i \leq n$ există un uplu φ a.i. $\varphi(A_i) = v_i$.

In practică este considerată o anumită ordonare a atributelor.

Elemente ale modelului relațional

O mulțime de uple peste U se numește *relație* și se notează cu r.

r poate varia în timp dar nu și în structură.

Exemplu:
$r = \{(v_{11}, v_{12}, \ldots v_{1n}), (v_{21}, v_{22}, \ldots v_{2n}), \ldots, (v_{m1}, v_{m2}, \ldots v_{mn})\}$.

Structura relației se va nota cu $R[U]$ unde R se numește *numele relației* iar U este mulțimea de *atribute* corespunzătoare.

Notații echivalente $R(U)$, $R(A_1, A_2, \ldots, A_n)$, $R[A_1, A_2, \ldots, A_n]$.
$R[U]$ se mai numește și *schemă de relație*.
Prin r construit peste $R[U]$ ne referim la tabela r ce corespunde schemei $R[U]$.

Ex: $R[U]$ poate fi: studenti(id int, nume varchar2(10), bursa int)

Elemente ale modelului relațional

In practică, o relație r poate fi reprezentată printr-o matrice:

$$r: \begin{array}{cccc} A_1 & A_2 & \dots & A_n \\ \hline v_{11} & v_{12} & \dots & v_{1n} \\ \dots & \dots & \dots & \dots \\ v_{m1} & v_{m2} & \dots & v_{mn} \\ \hline \end{array}$$

unde $(v_{i1}, v_{i2}, \dots, v_{in})$ este un uplu din r, $1 \leq i \leq m$ și
$v_{ij} \in dom(A_j),\ 1 \leq j \leq n, 1 \leq i \leq m$

Vom nota cu t_i linia cu numarul i din matrice:
$t_i = (v_{i1}, v_{i2}, \dots, v_{in})$

Elemente ale modelului relațional

O mulțime finită D de scheme de relație se numește *schemă de baze de date*. Formal, $D = \{R_1[U_1], \ldots, R_h[U_h]\}$ unde $R_i[U_i]$ este o schemă de relație, $1 \leq i \leq h$.

O *bază de date peste D* este o corespondență ce asociază fiecărei scheme de relație din D o relație.

Exemplu:
$r_1, r_2, \ldots r_h$ este o bază de date peste $D = \{R_1[U_1], \ldots, R_h[U_h]\}$.

Considerând D ca fiind ordonată $D = (R_1[U_1], \ldots, R_h[U_h])$, putem nota baza de date sub forma $(r_1, r_2, \ldots r_h)$

Operații

Asupra unei mulțimi de relații putem efectua o serie de operații. Există două categorii de operatori:

- Operatori din teoria mulțimilor: Reuniunea($\cup$), Intersecția ($\cap$), Diferența($-$), Produsul Cartezian($\times$)
- Operatori specifici algebrei relaționale: Proiecția (π), Selecția(σ), Redenumirea(ρ), Joinul Natural($\bowtie$), θ-Joinul, equijoinul, Semijoinul($\ltimes$ și $\rtimes$), Antijoinul($\triangleright$), Divizarea($\div$), Joinul la Stânga (⟕), Joinul la Dreapta(⟖), Joinul Exterior(⟗)

Operații pe mulțimi de tuple - *Reuniunea*: $\cup$

În cazul operațiilor pe mulțimi (cu excepția Produsului Cartezian), acestea se realizează între două relații r_1 și r_2 care sunt NEAPĂRAT construite peste aceeași mulțime de atribute.

Reuniunea a două relații r_1 și r_2, ambele peste $R[U]$, este o relație notată cu $r_1 \cup r_2$ definită astfel:

$$r_1 \cup r_2 = \{t \mid t = \text{uplu},\ \ t \in r_1 \text{ sau } t \in r_2\}$$

În practică, acest lucru se realizează utilizând cuvântul cheie UNION. Studenții din anii 1 și 3 sunt selectați de interogarea:

```
SELECT * FROM studenti WHERE an=1
UNION
SELECT * FROM studenti WHERE an=3;
```

Operații pe mulțimi de tuple - *Diferența*: $-$

Diferența a două relații r_1 și r_2, ambele peste $R[U]$, este o relație notată cu $r_1 - r_2$ definită astfel:

$$r_1 - r_2 = \{t \mid t = \text{uplu},\;\; t \in r_1 \text{ și } t \notin r_2\}$$

În practică, acest lucru se realizează utilizând cuvântul cheie MINUS. Pentru a-i selecta pe studenții din anul 2 fără bursă, putem să îi selectăm pe toți studenții din anul 2 și apoi să îi eliminăm pe cei cu bursa:

```
SELECT * FROM studenti WHERE an=2
MINUS
SELECT * FROM studenti WHERE bursa IS NOT NULL;
```

Se observa ca, la fel ca in cazul reuniunii, cele doua multimi de tuple peste care s-a facut diferenta sunt construite peste aceeasi multime de atribute.

Operații pe mulțimi de tuple - *Intersecția*: $\cap$

Intersecția a două relații r_1 și r_2, ambele peste $R[U]$, este o relație notată cu $r_1 \cap r_2$ definită astfel:

$$r_1 \cap r_2 = \{t \mid t = \text{uplu},\ \ t \in r_1 \text{ și } t \in r_2\}$$

În practică, acest lucru se realizează utilizând cuvântul cheie INTERSECT. Putem afla care studenți din anul 2 au bursa rulând:

```
SELECT * FROM studenti WHERE an=2
INTERSECT
SELECT * FROM studenti WHERE bursa IS NOT NULL;
```

Operatorul de intersecție poate fi obținut din ceilalți doi:
$r_1 \cap r_2 = r_1 - (r_1 - r_2)$

Operații pe mulțimi de tuple - *Produsul Cartezian*: $\times$

Produsul cartezian a două relații r_1 definită peste $R_1[U_1]$ și r_2 definită peste $R_2[U_2]$ cu $U_1 \cap U_2 = \emptyset$ este o relație notată cu $r_1 \times r_2$ definită astfel:

$$r_1 \times r_2 = \{t \mid t = \text{uplu peste } U_1 \cup U_2,\ t[U_1] \in r_1 \text{ și } t[U_2] \in r_2\}$$

De aceasta dată, cele două relații nu trebuie să fie peste aceeași mulțime de atribute. Rezultatul va fi o nouă relație peste o mulțime de atribute formată din atributele relațiilor inițiale.

Operația de proiecție $t[X]$ va avea ca rezultat o nou u uplu care este construit din t dar luând doar valorile asociate atributelor din X (care sunt și în U). Se mai notează cu $\pi_X[t]$.

Operații pe mulțimi de tuple - *Produsul Cartezian*: $\times$

Dacă un atribut s-ar repeta, el va fi identificat diferit. Spre exemplu, chiar dacă tabelele note și cursuri au un același atribut (id_curs), nu se face nici o sincronizare după acesta ci se vor crea două atribute diferite: note.id_curs respectiv cursuri.id_curs.

Produsul cartezian între aceste tabele, în practică, se obține executând interogarea:

```
SELECT * FROM cursuri, note;
```

Operații specifice algebrei relaționale

Operațiile pe mulțimi aveau ca elemente tuplele. Uneori aceste tuple nu sunt compatibile (de exemplu nu putem reuni o relație peste $R_1[U_1]$ cu una peste $R_2[U_2]$ dacă $U_1 \neq U_2$).

Pentru a opera asupra atributelor ce definesc tuplele din rezultat, avem nevoie de o serie de operatori specifici algebrei relaționale.

Operații în algebra relațională - *Proiecția:* π

Considerăm:

- $R[U]$ = schemă de relație;
- $X \subseteq U$;
- t = uplu peste $R[U]$ $(t \in r)$.

Se numește *proiecția lui t relativă la X* și notată cu $\pi_X[t]$, restricția lui t la mulțimea de atribute X. (Uneori vom scrie t[X])

Exemplu:
Dacă $U = (A_1, A_2, \ldots, A_n)$ atunci $t = (v_1, v_2, \ldots, v_n)$.
Considerăm $X = (A_{i_1}, A_{i_2}, \ldots, A_{i_k})$, $1 \leq i_1 < i_2 < \ldots < i_k \leq n$.

atunci $\pi_X[t] = (v_{i_1}, v_{i_2}, \ldots, v_{i_k})$;

Operații în algebra relațională - *Proiecția:* π

Dacă r este o relație peste $R[U]$ și $X \subseteq U$, atunci *proiecția lui r relativă la X* este $\pi_X[r] = \{\pi_X[t] \mid t \in r\}$

Exemplu:
Dacă $U = (A_1, A_2, \ldots, A_n)$ atunci
$r = \{(v_{11}, v_{12}, \ldots v_{1n}), (v_{21}, v_{22}, \ldots v_{2n}), \ldots, (v_{m1}, v_{m2}, \ldots v_{mn})\}$.
Considerăm $X = (A_{i_1}, A_{i_2}, \ldots, A_{i_k})$, $1 \leq i_1 < i_2 < \ldots < i_k \leq n$.

atunci
$\pi_X[r] = \{(v_{1_{i_1}}, v_{1_{i_2}}, \ldots v_{1_{i_k}}), (v_{2_{i_1}}, \ldots v_{2_{i_k}}), \ldots, (v_{m_{i_1}}, \ldots v_{m_{i_k}})\}$

In practica, proiectia se realizeaza selectand doar anumite campuri alte tabelei (anumite atribute):

SELECT nume, prenume FROM studenti;

Operații în algebra relațională - *Proiecția:* π

Ca si exemplu, vom scrie o interogare care sa returneze toate persoanele care trec pragul Facultatii (studenti si profesori):

```
SELECT nume, prenume FROM studenti
  UNION
  SELECT nume, prenume FROM profesori;
```

In cazul in care campurile cele doua campuri (nume, prenume) din cele doua tabele au acelasi tip (de exemplu nume este de tip VARCHAR2(10) in ambele tabele), interogarea va afisa toate persoanele ce "trec pragul Facultatii".

Observatie: Pentru a modifica tipul nume din tabela profesori la VARCHAR2(10) executati comanda:

```
ALTER TABLE profesori MODIFY nume VARCHAR2(10);
```

Operații în algebra relațională - *Selecția:* σ

Fie r o relație peste $R[U]$.
Considerăm pentru început expresiile elementare de selecție: $A\varphi B$, $A\varphi c$, $c\varphi B$, unde $A, B \in U$ și c este o constantă.

Dacă φ_1 și φ_2 sunt expresii de selecție (elementare sau nu), atunci următoarele sunt expresii de selecție: (φ_1), $\varphi_1 \wedge \varphi_2$, $\varphi_1 \vee \varphi_2$, $(\varphi_1 \wedge \varphi_2)$, $(\varphi_1 \vee \varphi_2)$.

Operații în algebra relațională - *Selecția:* σ

Fie θ o expresie de selecție. Atunci:

- când $\theta = A\varphi B$, t satisface θ dacă $\pi_A[t] \ \varphi \ \pi_B[t]$,
- când $\theta = A\varphi c$, t satisface θ dacă $\pi_A[t] \ \varphi \ c$,
- când $\theta = c\varphi B$, t satisface θ dacă $c \ \varphi \ \pi_B[t]$,
- când $\theta = \varphi_1 \wedge \varphi_2$, t satisface θ dacă t satisface atât pe φ_1 cât și pe φ_2,
- când $\theta = \varphi_1 \vee \varphi_2$, t satisface θ dacă t satisface măcar pe unul dintre φ_1 și φ_2.

Dacă θ este o expresie de selecție atunci *selecția* se notează cu $\sigma_\theta(r)$ și este definită ca:
$\sigma_\theta(r) = \{t | t = \text{tuplu peste } R[U], t \text{ satisface } \theta\}$

Operații în algebra relațională - *Selecția:* σ

In SQL, selectia se obtine utilizand o formula logica ce are rolul de a selecta doar anumite randuri.

De exemplu:
SELECT * FROM studenti WHERE an=2 and bursa IS NULL;

In acest exemplu, φ_1 este $AN = 2$, φ_2 este $bursa\ IS\ NULL$, $\varphi = \varphi_1 \wedge \varphi_2$ si r este multimea de randuri din tabela studenti.

Rezultatul este multimea studentilor din anul 2 care nu au bursa.

Operații în algebra relațională - *Redenumirea:* ρ

Operatorul de *redenumire* are rolul de a schimba numele unui atribut cu alt nume. Formal, daca dorim sa schimbam atributul A_1 in A_1' vom utiliza scrierea $\rho_{A_1/A_1'}(r)$. Restul atributelor peste care a fost construit r vor ramane neschimbate.

In SQL, redenumirea se realizeaza prin utilizarea cuvantului AS:

Exemplu:
SELECT bursa * 1.25 AS "BursaNoua" FROM studenti;
SELECT bursa + bursa/4 AS "BursaNoua" FROM studenti;
Daca nu am redenumi atributul nou obtinut, cele doua relatii ar fi considerate diferite (in prima numele atributului ar fi "bursa * 1.25", iar in a doua ar fi fost "bursa + bursa/4") - ATENTIE cand introduceti exercitii.

Operații în algebra relațională - *Join natural*: $\bowtie$

Considerăm:

- r_1 relație peste $R_1[U_1]$;
- r_2 relație peste $R_2[U_2]$;

Se numește *Join natural* a relațiilor r_1 si r_2, relația $r_1 \bowtie r_2$ peste $U_1 \cup U_2$ definită prin:

$$r_1 \bowtie r_2 = \{t \mid t \text{ uplu peste } U_1 \cup U_2,\ t[U_i] \in r_i,\ i = 1, 2\}$$

Dacă R este un nume pentru relația peste $U_1 \cup U_2$ atunci $r_1 \bowtie r_2$ este definită peste $R[U_1 \cup U_2]$
Pentru simplitate vom nota $U_1 \cup U_2$ cu U_1U_2.

Operații în algebra relațională - *Join natural*: $\bowtie$

Exemplu:

Fie $R_1[A, B, C, D]$, si $R_2[C, D, E]$ si r_1, r_2 a.i.:

r_1 :

A	B	C	D
0	1	0	0
1	1	0	0
0	0	1	0
1	1	0	1
0	1	0	1

r_2 :

C	D	E
1	1	0
1	1	1
0	0	0
1	0	0
1	0	1

Atunci: $r_1 \bowtie r_2$:

A	B	C	D	E
0	1	0	0	0
1	1	0	0	0
0	0	1	0	0
0	0	1	0	1

Operații în algebra relațională - *Join natural*: $\bowtie$

Urmatoarea interogare identifica cui apartine fiecare nota din tabelul note. Joinul se face dupa campul nr_matricol intre tabelele studenti si note:

```
SELECT nume, valoare FROM studenti
  NATURAL JOIN note;
```

```
SELECT nume, valoare FROM studenti
  JOIN note ON studenti.nr_matricol = note.nr_matricol;
```

Se poate observa ca daca din produsul cartezian am elimina acele cazuri in care campul "nr_matricol" nu este identic in ambele tabele, am obtine, de fapt, acelasi rezultat. Din acest motiv, joinul de mai sus poate fi scris si sub forma:

```
SELECT nume, valoare FROM studenti,note
  WHERE studenti.nr_matricol = note.nr_matricol;
```

Proprietăți ale Joinului natural

- $(r_1 \bowtie r_2)[U_1] \subseteq r_1$
- $(r_2 \bowtie r_1)[U_2] \subseteq r_2$

Dacă $X = U_1 \cap U_2$ și:
$r'_1 = \{t_1 | t_1 \in r_1, \exists t_2 \in r_2$ a.i. $t_1[X] = t_2[X]\}$ și $r_1" = r_1 - r'_1$,
$r'_2 = \{t_2 | t_2 \in r_2, \exists t_1 \in r_1$ a.i. $t_1[X] = t_2[X]\}$ și $r_2" = r_2 - r'_2$,
atunci: $r_1 \bowtie r_2 = r'_1 \bowtie r'_2$, $(r_1 \bowtie r_2)[U_1] = r'_1$, $(r_2 \bowtie r_1)[U_2] = r'_2$.

Dacă $\overline{r_1} \subseteq r_1, \overline{r_2} \subseteq r_2$ și $\overline{r_1} \bowtie \overline{r_2} = r_1 \bowtie r_2$ atunci $r'_1 \subseteq \overline{r_1}$ si $r'_2 \subseteq \overline{r_2}$

Dacă $U_1 \cap U_2 = \emptyset$ atunci $r_1 \bowtie r_2 = r_1 \times r_2$.

Extindere Join natural

Fie r_i relație peste $R_i[U_i], i = \overline{1,h}$ atunci:

$r_1 \bowtie r_2 \bowtie \ldots \bowtie r_h = \{t | t$ uplu peste $U_1, \ldots U_h,$ a.i. $t[U_i] \in r_i, i = \overline{1,h}\}$

Notații echivalente:

- $r_1 \bowtie r_2 \bowtie \ldots \bowtie r_h$
- $\bowtie \langle r_i, i = 1, h \rangle$
- $* \langle r_i, i = 1, h \rangle$

Operația join este asociativă.

Operații în algebra relațională - *θ-join, equijoin*

Fie r_i peste $R_i[U_i]$, $i = \overline{1,2}$ cu $A_{\alpha_1}, A_{\alpha_2}, \dots A_{\alpha_k} \in U_1$ și $B_{\beta_1}, B_{\beta_2}, \dots B_{\beta_k} \in U_2$ și
$\theta_i : dom(A_{\alpha_i}) \times dom(B_{\beta_i}) \to \{true, false\}, \forall i = \overline{1,k}$

θ-joinul a două relații r_1 și r_2, notat cu $r_1 \underset{\theta}{\bowtie} r_2$, este definit prin:

$$r_1 \underset{\theta}{\bowtie} r_2 = \{(t_1, t_2) | t_1 \in r_1, t_2 \in r_2, t_1[A_{\alpha_i}]\theta_i t_2[B_{\beta_i}], i = \overline{1,k}\}$$

unde $\theta = (A_{\alpha_1}\theta_1 B_{\beta_1}) \wedge (A_{\alpha_2}\theta_2 B_{\beta_2}) \wedge \dots \wedge (A_{\alpha_k}\theta_k B_{\beta_k})$

Daca θ_i este operatorul de egalitate, atunci θ-joinul se mai numeste si equijoin.

Operații în algebra relațională - *θ-join, equijoin*

Observație: un join oarecare cu condiția TRUE pentru orice combinatie de tuple este un produs cartezian: $r_1 \underset{true}{\bowtie} r_2 = r_1 \times r_2$

Observatie2: Joinul oarecare poate fi considerat ca fiind o filtrare dupa anumite criterii ale rezultatelor unui produs cartezian: $r_1 \underset{\theta}{\bowtie} r_2 = \sigma_\theta(r_1 \times r_2)$

Exemplu SQL:

```
SELECT s.nume, p.nume FROM studenti s, profesori p
  WHERE s.nume > p.nume;
```

Operații în algebra relațională - *Semijoin:* $\ltimes$ *și* $\rtimes$

Operatia de semijoin stang selecteaza acele randuri din relatia aflata in partea stanga ($\ltimes$) care au corespondent (in sensul joinului natural) in relatia din partea dreapta.

Formal, definim semijoinul stang a doua relatii r_1 peste $R_1[U_1]$ si r_2 peste $R_2[U_2]$ ca fiind:

$$r_1 \ltimes r_2 = \pi_{U_1}(r_1 \bowtie r_2)$$

Deja intalnit la proprietatile Joinului natural sub denumrea r_1'.

Semijoinul drept este definit similar dar preluand liniile din relatia aflata in dreapta (doar cele ce au corespondent in relatia din stanga).

Operații în algebra relațională - *Antijoin:* $\triangleright$

Tuplele ramase din relatia din stanga (care nu au fost preluate de semijoinul stang), formeaza rezultatul operatorului Antijoin.

Formal, definim antijoinul a doua relatii r_1 peste $R_1[U_1]$ si r_2 peste $R_2[U_2]$ ca fiind:

$$r_1 \triangleright r_2 = r_1 - \pi_{U_1}(r_1 \bowtie r_2)$$

$\ldots r_1$"

Operații în algebra relațională - *Joinul la Stânga:* ⟕

Fie r_1 si r_2 doua relatii in care nu toate tuplele din r_1 au un corespondent in r_2.
Operatia Join la Stanga a celor doua relatii r_1 si r_2 este reuniunea dintre tuplele existente in $r_1 \bowtie r_2$ si tuplele din r_1 ce nu sunt utilizate in join dar care au fost completate cu valoarea NULL pentru atributele din U_2.

$$r_1 ⟕ r_2 = r_1 \bowtie r_2 \text{ UNION } \pi_{U_1 U_2}(r_1 - (r_1 \bowtie r_2))$$

Joinul la Dreapta este definit similar, de aceasta data preluand liniile ce nu au folosit in Joinul natural din tabela din dreapta (r_2).

Operații în algebra relațională - *Joinul Extern:* ⟗

Operatia de Join exterior cuprinde toate liniile din Joinul la Stanga si din Joinul la Dreapta.

$$r_1 ⟗ r_2 = (r_1 ⟕ r_2) \cup (r_1 ⟖ r_2)$$

Operații în algebra relațională - *Joinul Extern:* ⟗

Cateva exemple (atentie la egalitate)

SELECT * FROM studenti LEFT JOIN profesori ON studenti.prenume = profesori.prenume;

(Toti studentii si asociati cu profesorii cu acelasi prenume cand e cazul)

SELECT * FROM studenti RIGHT JOIN profesori ON studenti.prenume = profesori.prenume;

(Unii studenti care sunt asociati cu profesorii avand acelasi prenume impreuna cu restul profesorilor)

SELECT * FROM studenti FULL JOIN profesori ON studenti.prenume = profesori.prenume;

(Studentii si profesorii si asocierile intre ei daca exista)

Exerciții:

1. Pentru r_1, r_2 exemplificate la Joinul natural, construiti restul tipurilor de Join studiate.
2. Utilizand schema de baze de date de la laborator, scrieti in algebra relationala urmatoarele:

- Cursurile din facultate impreuna cu numele profilor ce le tin.
- Numele si prenumele studentilor din anul 1 si care au bursa mai mare de 300 ron.
- Prenumele studentilor care au acelasi nume de familie ca macar unul din profesori.
- Numele si prenumele studentilor, cursurile pe care le-au urmat si notele pe care le-au obtinut.

Scrieti interogarile SQL asociate formulelor din algebra relationala scrise mai sus.

Notatii (alternative) operatori alg. relationala

Proiectia ($r_1[U]$): $\pi_U(r_1)$
Join natural ($r_1 * r_2$): $r_1 \bowtie r_2$
Join oarecare: $r_1 \underset{\theta}{\bowtie} r_2$
Selectia : $\sigma_\theta(r_1)$ [obs: $r_1 \underset{\theta}{\bowtie} r_2 = \sigma_\theta(r_1 \times r_2)$]
Join la stanga: $r_1 \triangleright \circ \triangleleft_L r_2$
Join la dreapta: $r_1 \triangleright \circ \triangleleft_R r_2$
Full outer join : $r_1 \triangleright \circ \triangleleft r_2$
Redenumirea: Daca r este definit peste $B_1, B_2, \ldots, B_n$ si vrem sa redenumim numele atributelor, vom folosi operatorul de redenumire $\rho : r' = \rho(r_1)_{A_1, A_2, \ldots, A_n}$ - redenumirea atributelor lui r in $A_1, A_2, \ldots, A_n$

Dependențe funcționale

Dependențe funcționale

Fie $X, Y \subseteq U$. Vom nota o dependență funcțională cu $X \rightarrow Y$.

O relație r peste U satisface *dependența funcțională* $X \rightarrow Y$ dacă:

$$(\forall t_1, t_2)(t_1, t_2 \in r)[t_1[X] = t_2[X] \Rightarrow t_1[Y] = t_2[Y]]$$

$X = \emptyset$ avem $\emptyset \rightarrow Y$ dacă $(\forall t_1, t_2)(t_1, t_2 \in r)[t_1[Y] = t_2[Y]]$

$Y = \emptyset$ atunci orice $\forall r$ peste U avem că $X \rightarrow \emptyset$

Dacă r satisface $X \rightarrow Y$, atunci există o funcție $\varphi : r[X] \rightarrow r[Y]$ definită prin $\varphi(t) = t'[Y]$, unde $t' \in r$ și $t'[X] = t \in r[X]$.
Dacă r satisface $X \rightarrow Y$ spunem că X determină funcțional pe Y în r.

Proprietăți ale dependențelor funcționale

FD1. (Reflexivitate) Dacă $Y \subseteq X$, atunci r satisface $X \to Y$, $\forall r \in U$.

FD2. (Extensie) Dacă r satisface $X \to Y$ și $Z \subseteq W$, atunci r satisface $XW \to YZ$.

FD3. (Tranzitivitate) Dacă r satisface $X \to Y$ și $Y \to Z$, atunci r satisface $X \to Z$.

FD4. (Pseudotranzitivitate) Dacă r satisface $X \to Y$ și $YW \to Z$, atunci r satisface $XW \to Z$.

Proprietăți ale dependențelor funcționale

FD5. (Uniune) Dacă r satisface $X \rightarrow Y$ și $X \rightarrow Z$, atunci r satisface $X \rightarrow YZ$.

FD6. (Descompunere) Dacă r satisface $X \rightarrow YZ$, atunci r satisface $X \rightarrow Y$ și $X \rightarrow Z$.

FD7. (Proiectabilitate) Dacă r peste U satisface $X \rightarrow Y$ și $X \subset Z \subseteq U$, atunci $r[Z]$ satisface $X \rightarrow Y \cap Z$

FD8. (Proiectabilitate inversă) Dacă $X \rightarrow Y$ este satisfacută de o proiecție a lui r, atunci $X \rightarrow Y$ este satisfacută de r.

Dependențe funcționale - consecință și acoperire

Dacă Σ este o mulțime de dependențe funcționale peste U atunci spunem că *$X \to Y$ este consecință din Σ* dacă orice relație ce satisface toate consecințele din Σ satisface și $X \to Y$.
Notație: $\Sigma \models X \to Y$

Fie $\Sigma^* = \{X \to Y | \Sigma \models X \to Y\}$. Fie Σ_1 = mulțime de dependențe funcționale. Σ_1 constituie o *acoperire* pentru Σ^* dacă $\Sigma_1^* = \Sigma^*$.

Proprietăți ale dependențelor funcționale

Propoziție

Pentru orice mulțime Σ de dependențe funcționale există o acoperire Σ_1 pentru Σ^*, astfel încat toate dependențele din Σ_1 sunt de forma $X \to A$, A fiind un atribut din U.

Propoziție

$\Sigma \models X \to Y$ *dacă și numai dacă* $\Sigma \models X \to B_j$ *pentru* $j = \overline{1, h}$, *unde* $Y = B_1 \dots B_h$.

Reguli de deducere

Fie $\mathcal{R}$ o mulțime de formule de deducere pentru dependențe funcționale și Σ o mulțime de dependențe funcționale. Spunem că $X \to Y$ este o *demonstrație* în Σ utilizând regulile $\mathcal{R}$ și vom nota $\Sigma \vdash_{\mathcal{R}} X \to Y$, dacă există șirul $\sigma_1, \sigma_2, \ldots, \sigma_n$, astfel încât:

- $\sigma_n = X \to Y$ și
- pentru $\forall i = \overline{1, n}$, $\sigma_i \in \Sigma$ sau există în $\mathcal{R}$ o regulă de forma $\frac{\sigma_{j_1}, \sigma_{j_2}, \ldots \sigma_{j_k}}{\sigma_i}$, unde $j_1, j_2, \ldots, j_k < i$.

Reguli de deducere

Conform proprietăților FD1-FD5 putem defini regulile:

FD1f: $\frac{Y \subseteq X}{X \rightarrow Y}$

FD2f: $\frac{X \rightarrow Y,\ Z \subseteq W}{XW \rightarrow YZ}$

FD3f: $\frac{X \rightarrow Y,\ Y \rightarrow Z}{X \rightarrow Z}$

FD4f: $\frac{X \rightarrow Y,\ YW \rightarrow Z}{XW \rightarrow Z}$

FD5f: $\frac{X \rightarrow Y,\ X \rightarrow Z}{X \rightarrow YZ}$

FD6f: $\frac{X \rightarrow YZ}{X \rightarrow Y}, \frac{X \rightarrow YZ}{X \rightarrow Z}$

Propoziție

Regulile FD4f, FD5f, FD6f se exprimă cu ajutorul regulilor FD1f, FD2f, FD3f.

Notăm cu $\mathcal{R}_1 = \{$FD1f, FD2f, FD3f$\}$,
și cu $\mathcal{R}_2 = R_1 \cup \{$FD4f, FD5f, FD6f$\}$

Propoziție

Regulile FD4f, FD5f, FD6f se exprimă cu ajutorul regulilor FD1f, FD2f, FD3f.

Idei de demonstratie:

- FD4f:Se aplica FD2f pentru $X \to Y$ si $W \subseteq W$ iar din rezultat si din $YW \to Z$ prin FD3f se obtine rezultatul;
- FD5f: Se aplica FD2f pentru $X \to Y$ si $X \subseteq X$ si la fel pentru $X \to Z$ si $Y \subseteq Y$ apoi FD3f (tranzitivitatea) intre rezultate;
- FD6f: din FD1f avem ca $YZ \to Y$ si $YZ \to Z$ si din FD3f rezulta $X \to Y$ si $X \to Z$

Axiomele lui Armstrong

Armstrong a definit (în *Dependency structures of database relationships* Proc. IFIP 74, Amsterdam, 580-583) următoarele reguli de inferența (numite *Axiomele lui Armstrong*):

A1: $\frac{}{A_1 ... A_n \rightarrow A_i}, i = \overline{1, n}$

A2: $\frac{A_1, ... A_m \rightarrow B_1, ... B_r}{A_1 ... A_m \rightarrow B_j}, j = \overline{1, r}$

$\frac{A_1, ... A_m \rightarrow B_j, \ j = \overline{1, r}}{A_1 ... A_m \rightarrow B_1, ... B_r}$

A3: $\frac{A_1, ... A_m \rightarrow B_1, ... B_r, \quad B_1, ... B_r \rightarrow C_1, ... C_p}{A_1 ... A_m \rightarrow C_1, ... C_p}$

unde A_i, B_j, C_k sunt atribute. Notăm $\mathcal{R}_A = \{$A1, A2, A3$\}$.

Obs: regula A3 este de fapt FD3f (tranzitivitatea).

Propoziție

Regulile din $\mathcal{R}_1$ se exprimă prin cele din R_A și invers.

Notatie:

$$\Sigma^+_{\mathcal{R}} = \{X \to Y | \Sigma \vdash_{\mathcal{R}} X \to Y\}$$

Propoziție

Fie $\mathcal{R}'_1$ si $\mathcal{R}'_2$ doua multimi de reguli astfel incat $\mathcal{R}'_1$ se exprima prin $\mathcal{R}'_2$ si invers. Atunci $\Sigma^+_{\mathcal{R}'_1} = \Sigma^+_{\mathcal{R}'_2}$ pentru orice multime Σ de dependente functionale.

Consecinta: $\Sigma^+_{\mathcal{R}_1} = \Sigma^+_{\mathcal{R}_A}$

Fie $X \subseteq U$ si $\mathcal{R}$ o multime de reguli de inferenta. Notam cu

$$X^+_{\mathcal{R}} = \{A | \Sigma \vdash_{\mathcal{R}} X \to A\}$$

Lema

$\Sigma \vdash_{\mathcal{R}} X \to Y$ *daca si numai daca* $Y \subseteq X^+_{\mathcal{R}_1}$.

Lema

Fie Σ o multime de dependente functionale si $\sigma : X \to Y$ o dependenta functionala astfel incat $\Sigma \nvdash_{\mathcal{R}_1} X \to Y$. Atunci exista o relatie r_σ ce satisface toate dependentele functionale din Σ si r_σ nu satisface $X \to Y$.

Theorem

Fie Σ o multime de dependente functionale. Atunci exista o relatie r_0 ce satisface exact elementele lui $\Sigma^+_{\mathcal{R}_1}$, adica:

- *r_0 satisface τ, $\forall \tau \in \Sigma^+_{\mathcal{R}_1}$ si*
- *r_0 nu satisface γ, $\forall \gamma \notin \Sigma^+_{\mathcal{R}_1}$*

Dependențe multivaluate

Exemplu

Presupunem că persoana cu CNP $= 1$ a fost admisă la două facultăți și are permis de conducere pentru categoriile A și B:

r :

CNP	Admis la facult.	Are permis categ.
1	Informatică	A
1	Matematică	B

Deși anumite rânduri nu sunt scrise în tabelă, putem să intuim că persoana cu CNP $= 1$ a dat la Facultatea de Informatică și are permis de conducerea categoria B. Deci, deși în r nu există t-uplul $\langle$ 1,Informatica,B $\rangle$, ar trebui să existe și el (pentru că poate fi dedus din cele existente).

Care alt t-uplu mai poate fi dedus ?

Exemplu

r :

CNP	Admis la facult.	Are permis categ.
1	Informatică	A
1	Matematică	B
1	Informatică	B
1	Matematică	A

t-uplele marcate cu roșu ar putea lipsi, ele fiind redundante deoarece pot fi obținute din primele două t-uple.

Prin intermediul dependențelor funcționale pot afla la care coloane pot renunța astfel încât să le pot reface ulterior.

Prin intermediul dependențelor multivaluate pot afla la care linii pot renunța astfel încât să le pot reface ulterior.

Dependențe multivaluate - definiție

Fie $X, Y \subseteq U$. O dependență multivaluată este notată cu $X \twoheadrightarrow Y$.

Definition

Relația *r peste U satisface dependența multivaluată $X \twoheadrightarrow Y$* dacă pentru oricare două tuple $t_1, t_2 \in r$ și $t_1[X] = t_2[X]$, există tuplele t_3 și t_4 din r, astfel încât:

- $t_3[X] = t_1[X], t_3[Y] = t_1[Y], t_3[Z] = t_2[Z]$;
- $t_4[X] = t_2[X], t_4[Y] = t_2[Y], t_4[Z] = t_1[Z]$

unde $Z = U - XY$ (Z mai este denumită și $rest$).

Exemplul 2 (mai formal)

	A	B	C	D			
	a_1	b_1	c_1	d_1	t_1		t_1"
	a_1	b_2	c_2	d_2	t_2		
r :	a_1	b_1	c_1	d_2	t_3		t_2"
	a_1	b_2	c_2	d_1	t_4		
	a_2	b_3	c_1	d_1		t'_1, t'_4	
	a_2	b_3	c_1	d_2		t'_2, t'_3	

r satisface $A \twoheadrightarrow BC$

Intrebare: cum alegem t_3", t_4" ?

Deoarece atunci când $t_1[A] = t_2[A]$ avem că:
$t_3[A] = t_1[A], t_3[BC] = t_1[BC], t_3[D] = t_2[D]$ și
$t_4[A] = t_2[A], t_4[BC] = t_2[BC], t_4[D] = t_1[D]$

Definiție echivalentă

Relația r *peste* U *satisface dependența multivaluată* $X \twoheadrightarrow Y$, dacă pentru orice $t_1, t_2 \in r$ cu $t_1[X] = t_2[X]$ avem că $M_Y(t_1[XZ]) = M_Y(t_2[XZ])$

unde $M_Y(t[XZ]) = \{t'[Y] | t' \in r, t'[XZ] = t[XZ]\}$ = valorile lui Y din diferite tuple in care XZ sunt egale (cu XZ-ul din parametru).

r :

A	B	C	D		
a_1	b_1	c_1	d_1	$=$	t_1
a_1	b_2	c_2	d_2	$=$	t_2
a_1	b_1	c_1	d_2		
a_1	b_2	c_2	d_1		
a_2	b_3	c_1	d_1		
a_2	b_3	c_1	d_2		

$M_Y(t_1[AD]) = M_Y(t_2[AD]) = \{(b_1, c_1), (b_2, c_2)\}$

Observații

- Dacă r satisface dependența funcțională $X \to Y$, atunci pentru orice $t \in r$, avem $M_Y(t[XZ]) = \{t[Y]\}$.
- Dacă r satisface dependența funcțională $X \to Y$, atunci r satisface și dependența multivaluată $X \twoheadrightarrow Y$.
- Dacă r satisface dependența multivaluată $X \twoheadrightarrow Y$, atunci putem defini o funcție $\psi : r[X] \to \mathcal{P}(r[Y])$, prin $\psi(t[X]) = M_Y(t[XZ]), \forall t \in r$ (returneaza valorile diferite din proiectia pe Y). Când r satisface $X \to Y$, atunci $\psi : r[X] \to r[Y]$ (deoarece valorile pe Y nu sunt diferite in cadrul dependentei functionale).

Proprietăți ale dependențelor multivaluate

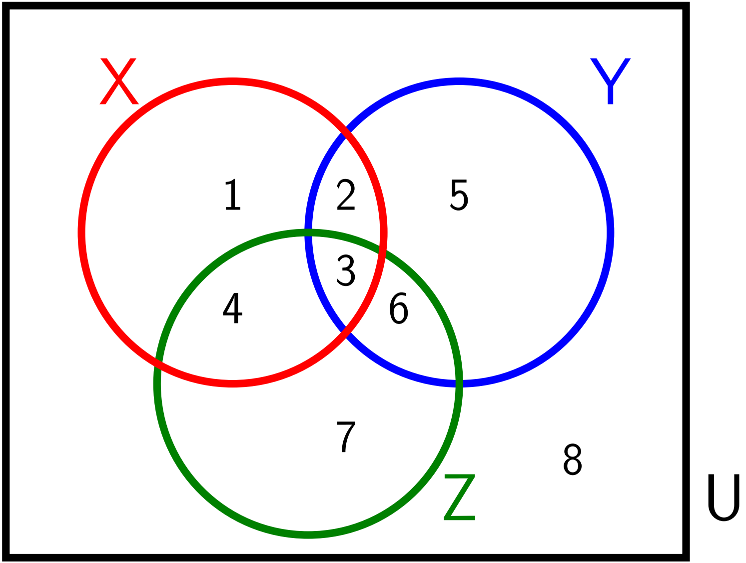
MVD0 (Complementariere) Fie $X, Y, Z \subseteq U$, asfel încât $XYZ = U$ și $Y \cap Z \subseteq X$. Dacă r satisface $X \twoheadrightarrow Y$, atunci r satisface $X \twoheadrightarrow Z$.

MVD1 (Reflexivitate) Dacă $Y \subseteq X$, atunci orice relație r satisface $X \twoheadrightarrow Y$.

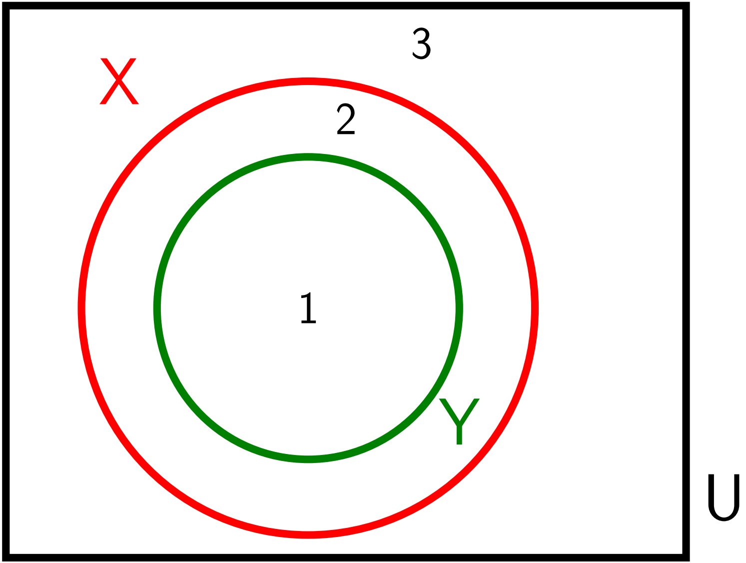
MVD2 (Extensie) Fie $Z \subseteq W$ și r satisface $X \twoheadrightarrow Y$. Atunci r satisface $XW \twoheadrightarrow YZ$

MVD3 (Tranzitivitate) Dacă r satisface $X \twoheadrightarrow Y$ și $Y \twoheadrightarrow Z$, atunci r satisface $X \twoheadrightarrow Z - Y$

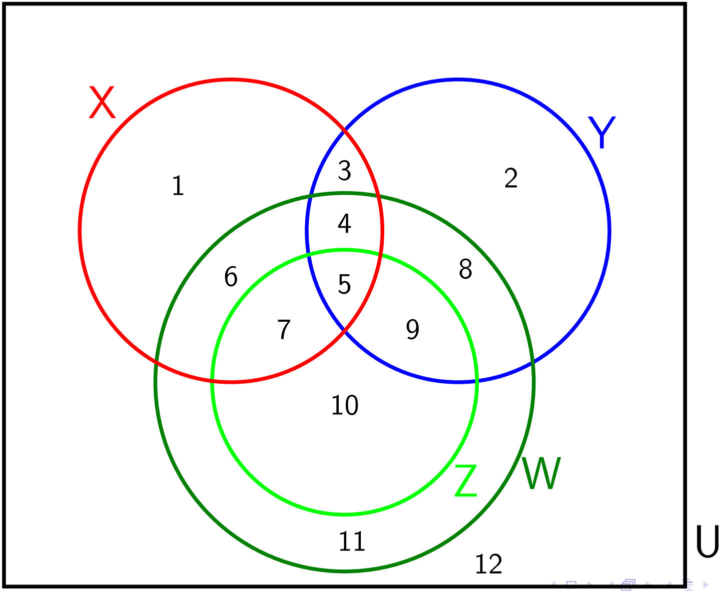
MVD0



MVD1



MVD2



Proprietăți ale dependențelor multivaluate

MVD4 (Pseudotranzitivitate) Dacă r satisface $X \twoheadrightarrow Y$ și $YW \twoheadrightarrow Z$, atunci r satisface și $XW \twoheadrightarrow Z - YW$.

MVD5 (Uniune) Dacă r satisface $X \twoheadrightarrow Y$ și $X \twoheadrightarrow Z$ atunci r satisface $X \twoheadrightarrow YZ$.

MVD6 (Descompunere) Dacă r satisface $X \twoheadrightarrow Y$ și $X \twoheadrightarrow Z$, atunci r satisface $X \twoheadrightarrow Y \cap Z$, $X \twoheadrightarrow Y - Z$, $X \twoheadrightarrow Z - Y$

Proprietăți mixte ale dependențelor multivaluate

FD-MVD1. Dacă r satisface $X \to Y$, atunci r satisface și $X \twoheadrightarrow Y$.

FD-MVD2. Dacă r satisface $X \twoheadrightarrow Z$ și $Y \to Z'$, cu $Z' \subseteq Z$ și $Y \cap Z = \emptyset$, atunci r satisface $X \to Z'$.

FD-MVD3. Dacă r satisface $X \twoheadrightarrow Y$ și $XY \twoheadrightarrow Z$, atunci r satisface $X \to Z - Y$.

Reguli de inferență

MVD0f: $\frac{XYZ=U,\ \ Y\cap Z\subseteq X,\ \ X\twoheadrightarrow Y}{X\twoheadrightarrow Z}$

MVD1f: $\frac{Y\subseteq X}{X\twoheadrightarrow Y}$

MVD2f: $\frac{Z\subseteq W,\ \ X\twoheadrightarrow Y}{XW\twoheadrightarrow YZ}$

MVD3f: $\frac{X\twoheadrightarrow Y,\ \ Y\twoheadrightarrow Z}{X\twoheadrightarrow Z-Y}$

MVD4f: $\frac{X\twoheadrightarrow Y, YW\twoheadrightarrow Z}{XW\twoheadrightarrow Z-YW}$

Reguli de inferență

MVD5f: $$\frac{X \twoheadrightarrow Y,\ \ X \twoheadrightarrow Z}{X \twoheadrightarrow YZ}$$

MVD6f: $$\frac{X \twoheadrightarrow Y,\ \ X \twoheadrightarrow Z}{X \twoheadrightarrow Y \cap Z,\ \ X \twoheadrightarrow Y - Z,\ \ X \twoheadrightarrow Z - Y}$$

FD-MVD1f: $$\frac{X \rightarrow Y}{X \twoheadrightarrow Y}$$

FD-MVD2f: $$\frac{X \twoheadrightarrow Z,\ \ Y \rightarrow Z',\ \ Z' \subseteq Z,\ \ Y \cap Z = \emptyset}{X \rightarrow Z'}$$

FD-MVD3f: $$\frac{X \twoheadrightarrow Y,\ \ XY \twoheadrightarrow Z}{X \twoheadrightarrow Z - Y}$$

Propoziție

Fie $\mathcal{R}$ o multime de reguli valide si γ o regula $\frac{\alpha_1, \alpha_2, \ldots \alpha_k}{\beta}$, astfel incat $\{\alpha_1, \ldots \alpha_k\} \vdash_{\mathcal{R}} \beta$, atunci si regula γ este valida.

Propoziție

Fie $\mathcal{R}_{FM} = \{FD1f \ - \ FD3f$[1], $\ MVD0f \ - \ MVD3f$, $FD - MVD1f \ - \ FD - MVD3f\}$. Avem:

- *$FD - MVD3f$ se exprima cu celelalte regulid din $\mathcal{R}_{FM}$ si FD*
- *$MVD2f$ se exprima prin celelalte reguli din $\mathcal{R}_{FM}$.*

Propoziție

Regulile $MVD4f - MVD6f$ se exprima cu ajutorul regulilor $MVD0f - MVD3f$

[1] cele de la dependente functionale

Theorem

Fie Σ o multime de dependente functionale sau multivaluate si X o submultime de atribute. Atunci exista o partitie a lui $U - X$ notata prin $Y_1 \dots Y_k$, astfel incat pentru $Z \subseteq U - X$ avem $\Sigma \vdash_{\mathcal{R}_{FM}} X \twoheadrightarrow Z$ daca si numai daca Z este reuniunea unui numar de multimi din partitia $\{Y_1, \dots Y_k\}$

Definition

Pentru Σ o multime de dependente functionale sau multivaluate si X o submultime de atribute, numim *baza de dependenta pentru X cu privire la Σ* partitia $B(\Sigma, X) = \{\{A_1\} \dots \{A_h\}, Y_1 \dots Y_k\}$, unde $X = A_1, \dots A_h$, iar $Y_1, \dots Y_k$ este partitia construita in teorema precedenta.

Observatii

- Avem $\Sigma \vdash_{\mathcal{R}_{FM}} X \twoheadrightarrow Z$ daca si numai daca Z este o reuniune de elemente din partitia $B(\Sigma, X)$.
- Fie $X^*_\Sigma = \{A | \Sigma \vdash_{\mathcal{R}_{FM}} X \rightarrow A\}$. Atunci pentru orice $A \in X^*_\Sigma$ avem $\{A\} \in B(\Sigma, X)$.