Capitolul 4

DEPENDENȚE MULTIVALOARE ŞI DEPENDENȚE JONCȚIUNE

Modelul relațional utilizează dependențele pentru exprimarea constrângerilor pe care datele din baza de date trebuie să le satisfacă. Schema bazei de date relaționale este definită de o varietate de constrângeri ce sunt impuse componentelor sale. Dependențele funcționale sunt un exemplu de astfel de constrângeri de integritate. Ele au fost studiate detaliat în capitolul 3.

O generalizare a dependențelor funcționale, numite dependențe multivaloare, a fost descoperită de mai mulți cercetători în domeniu. Cea mai importantă proprietate a dependenței multivaloare constă în faptul că existența ei într-o relație este o condiție necesară şi suficientă pentru ca relația să poată fi înlocuită fără pierderi de informații, independent de extensia curentă, cu două proiecții ale sale. Această proprietate face ca dependența multivaloare să joace un rol important în teoria şi practica proiectării bazelor de date relaționale.

O dată ce dependențele multivaloare au devenit parte a teoriei relațiilor, o cerință de bază ce trebuie să fie satisfăcută este cunoaşterea proprietăților lor şi, în particular, metodelor de manipulare. Întrucât dependențele multivaloare sunt o generalizare a celor funcționale, metodele aplicate asupra ultimelor pot servi drept ghid în susținerea acestei cerințe.

Este bine cunoscut că existența într-o relație a dependențelor funcționale implică că în ea există dependențe funcționale adiționale. Aceasta e valabil şi pentru dependențele multivaloare. Noțiunea de implicare este formalizată în conceptul de reguli de inferență. Sunt cunoscute mulțimi închise şi complete de reguli de inferență pentru dependențele multivaloare.

Dependențele joncțiune sunt o generalizare a dependențelor multivaloare. E cunoscut faptul că o mulțime de dependențe funcționale plus o dependență joncțiune se consideră suficiente pentru exprimarea dependențelor dintre atributele unei scheme a bazei de date.

Acest capitol cuprinde noțiuni generale despre dependențele multivaloare, regulile de inferență, dependențele multivaloare incluse, regulile de inferență ale dependențelor joncțiune etc.

4.1. Dependențe multivaloare

Definiția 4.1. Fie relația r cu schema R şi $X,Y\subseteq R$. Notăm $Z=R \setminus XY$. Vom spune că relația r(R) satisface *dependența multivaloare* $X\rightarrow\rightarrow Y$ (sau $X\rightarrow\rightarrow Y$ e validă în r(R)), dacă pentru orice pereche de tupluri t_1 şi t_2 din r(R) ce satisfac $t_1[X]=t_2[X]$ există în r(R) un tuplu t_3 pentru care au loc egalitățile $t_3[X]=t_1[X]$, $t_3[Y]=t_1[Y]$ şi $t_3[Z]=t_2[Z]$.

Remarcă. Din proprietatea de simetrie a acestei definiții urmează că în r(R) mai există un tuplu t_4 ce satisface egalitățile $t_4[X]=t_1[X]$, $t_4[Y]=t_2[Y]$ şi $t_4[Z]=t_1[Z]$.

Teorema 4.1. O dependență multivaloare $X \to\to Y$ e validă în relația r(R) dacă şi numai dacă $X \to\to Z$ e validă în r(R), unde $Z = R \setminus XY$.

Demonstrație. Din remarca definiției 4.1 urmează că, dacă relația r(R) satisface dependența multivaloare $X \to\to Y$, atunci de fiecare dată când $t_1[X]=t_2[X]$ în r(R) există nu numai un tuplu t_3 ce satisface $t_3[X]=t_1[X]$, $t_3[Y]=t_1[Y]$ şi $t_3[Z]=t_2[Z]$, dar şi un tuplu t_4 pentru care au loc egalitățile $t_4[X]=t_1[X]$, $t_4[Y]=t_2[Y]$ şi $t_4[Z]=t_1[Z]$. În consecință, tupluri distincte cu aceleaşi X-valori şi cu Y-valori (Z-valori) identice trebuie să aibă diferite Z-valori (Y-valori) pentru a menține toate tuplurile distincte. Din această proprietate simetrică rezultă că relația r(R) satisface dependența multivaloare $X \to\to Y$ dacă şi numai dacă satisface dependența multivaloare $X \to\to Z$.

Exemplul 4.1. Relația r(ABCD) din fig.4.1 satisface dependența multivaloare $BC \to\to A$. În relația r(ABCD) e validă de asemenea dependența multivaloare $BC \to\to D$. Dacă, însă, din relația r(ABCD) este eliminat un tuplu, atunci dependențele multivaloare $BC \to\to A$ şi $BC \to\to D$ devin invalide în r(ABCD).

r	A	B	C	D
	a_1	b_1	c_1	d_1
	a_1	b_1	c_1	d_2
	a_1	b_1	c_2	d_1
	a_1	b_1	c_2	d_2
	a_2	b_1	c_1	d_1
	a_2	b_1	c_1	d_2
	a_2	b_1	c_2	d_1
	a_2	b_1	c_2	d_2

Fig.4.1

În definiția 4.1 nu s-au pus condiții asupra mulțimilor X şi Y. Deci $X \cap Y \neq \varnothing$ în caz general. Determinatul Y poate fi redus. Să demonstrăm că varianta redusă, $X \to\to Y \setminus X$, e echivalentă dependenței $X \to\to Y$.

Teorema 4.2. Dependența funcțională $X \to\to Y$ e validă în relația r(R), dacă şi numai dacă $X \to\to Y \setminus X$ e validă în r(R).

Demonstrație. *Necesitatea.* Fie relația r(R) satisface dependența multivaloare $X \to\to Y$. Notăm $Y^1 = Y \setminus X$. Atunci $Z= R \setminus XY = R \setminus XY^1$. Fie t_1 şi t_2 două tupluri cu X-valori egale, adică $t_1[X] = t_2[X]$. Fiindcă $X \to\to Y$ e validă în r(R), atunci în r trebuie să existe un tuplu t_3 ce satisface $t_3[X]=t_1[X]$, $t_3[Y]=t_1[Y]$ şi $t_3[Z] = t_2[Z]$. Egalitatea $t_3[Y] = t_1[Y]$ implică egalitatea $t_3[Y^1] = t_1[Y^1]$. Prin urmare, relația r satisface şi dependența multivaloare $X \to\to Y^1$.

Suficiența. Fie r(R) satisface dependența multivaloare $X \to\to Y^1$, unde $Y^1 = Y \setminus X$ şi fie $X^1 \subseteq X$. Să arătăm că dependența $X \to\to Y^1X^1$ e validă în r(R). Întrucât r satisface $X \to\to Y^1$ şi dacă $t_1, t_2 \in r$ şi $t_1[X] = t_2[X]$, atunci există un tuplu t_3, pentru care $t_3[X] = t_1[X]$, $t_3[Y^1] = t_1[Y^1]$ şi $t_3[Z] = t_2[Z]$. Din $X^1 \subseteq X$ şi $t_3[Y^1] = t_1[Y^1]$ urmează $t_3[Y^1X^1] = t_1[Y^1X^1]$. Deci $X \to\to Y^1X^1$.

Exemplul 4.2. Relația r(ABCD) din fig.4.1 satisface dependența multivaloare $BC \to\to A$. Conform teoremei 4.2 în r e validă şi dependența multivaloare $BC \to\to AB$.

Teorema ce urmează poate fi considerată o metodă de verificare dacă o dependență multivaloare e validă într-o relație.

Teorema 4.3. Fie relația r(R), $X,Y \subseteq R$ şi $Z = R \setminus XY$. Dependența multivaloare $X \to\to Y$ e validă în r(R) dacă şi numai dacă r este joncțiunea proiecțiilor sale $\pi_{XY}(r)$ şi $\pi_{XZ}(r)$.

Demonstrație. *Necesitatea.* Fie dependența multivaloare $X \to\to Y$ e validă în r(R) şi fie $r_1=\pi_{XY}(r)$, $r_2=\pi_{XZ}(r)$. Fiindcă întotdeauna are loc corelația $r(R) \subseteq r_1 |\times| r_2$, pentru a demonstra necesitatea, e de ajuns să arătăm că orice tuplu t din joncțiunea $r_1 |\times| r_2$ este şi în r(R), adică $r_1 |\times| r_2 \subseteq r(R)$. Fie t un tuplu în $r_1 |\times| r_2$. Atunci r_1 şi r_2 trebuie să conțină corespunzător tuplurile t_1 şi t_2 încât $t[X] = t_1[X] = t_2[X]$, $t[Y] = t_1[Y]$ şi $t[Z] = t_2[Z]$. Întrucât r_1 şi r_2 sunt proiecții ale relației r, atunci r conține tuplurile t_1^1 şi t_2^1 pentru care $t_1[XY]=t_1^1[XY]$ şi $t_2[XZ]=t_2^1[XZ]$. În r este un tuplu t_3 (dat fiind faptul că $X \to\to Y$ e validă în r) ce satisface $t_3[X] = t_1^1[X]$, $t_3[Y] = t_1^1[Y]$ şi $t_3[Z] = t_2^1[Z]$. Se vede că acest tuplu t_3 este t.

Suficiența. Presupunem acum că relația r se descompune în două relații r_1 şi r_2 fără pierderi. Să arătăm că pentru orice două tupluri t_1^1 şi t_2^1 ce satisfac $t_1^1[X] = t_2^1[X]$ există un tuplu t încât $t[X] = t_1^1[X]$, $t[Y] = t_1^1[Y]$ şi $t[Z] = t_2^1[Z]$, adică în r e validă dependența multivaloare $X \to\to Y$.

Fie t_1^1 şi t_2^1 două tupluri în r(R). Întrucât r(R) se descompune fără pierderi asupra XY şi XZ (adică $r=\pi_{XY}(r)|\times|\pi_{XZ}(r)$), atunci în r_1 şi r_2 se găsesc, respectiv, tuplurile t_1 şi t_2 şi $t_1 = t_1^1[XY]$, $t_2 = t_2^1[XZ]$. Fiindcă $r = r_1 |\times| r_2$, r conține un tuplu t ce satisface $t[XY]=t_1[XY]$ şi $t[XZ]=t_2[XZ]$. Întrucât tuplurile t_1^1, t_2^1 şi t satisfac definiția 4.1 relația r(R) satisface dependența multivaloare $X \to\to Y$.

Din teorema 4.3 se poate face următoarea concluzie.

Concluzie. Relația r(R) se descompune fără pierderi în relațiile $r_1(R_1)$ şi $r_2(R_2)$ dacă şi numai dacă $R_1 \cap R_2 \to\to R_1$ (sau $R_1 \cap R_2 \to\to R_2$).

Este ineficientă utilizarea metodei din această teoremă pentru a verifica dacă o relație satisface sau nu o dependență multivaloare. Testarea necesită două proiecții şi o joncțiune. Să examinăm un alt procedeu de verificare, dacă o dependență multivaloare e validă într-o relație.

Fie relația r(R) satisface dependența multivaloare $X \to\to Y$, atunci conform teoremei 4.3 $r(R) = \pi_{XY}(r) |\times| \pi_{XZ}(r)$, unde $Z = R \setminus XY$.

Expresiile $|\pi_{XY}(\sigma_{X=x}(r))|$ şi $|\pi_{XZ}(\sigma_{X=x}(r))|$ reprezintă numerele de tupluri în proiecțiile relației r asupra mulțimilor XY şi XZ, corespunzător, pentru X-valoarea dată egală cu x. Este evident că, dacă relația r se descompune fără pierderi în relațiile $\pi_{XY}(r)$ şi $\pi_{XZ}(r)$, atunci

$$|\sigma_{X=x}(r)| = |\pi_{XY}(\sigma_{X=x}(r))| \cdot |\pi_{XZ}(\sigma_{X=x}(r))|. \qquad (4.1)$$

Întrucât $|\pi_{XW}(\sigma_{X=x}(r))| = |\pi_{W}(\sigma_{X=x}(r))|$, atunci egalitatea (4.1) poate fi simplificată:

$$|\sigma_{X=x}(r)| = |\pi_{Y}(\sigma_{X=x}(r))| \cdot |\pi_{Z}(\sigma_{X=x}(r))|. \qquad (4.2)$$

Această procedură de verificare a validităţii unei dependenţe multivaloare este mai puţin laborioasă decât precedenta. Ea presupune sortarea tuplurilor după X-valori, apoi pentru orice X-valoare x se testează egalitatea de mai sus.

Exemplul 4.3. Relaţia r(ABCD) din fig.4.1 satisface teorema 4.3. Proiecţiile $\pi_{BCA}(r)$ şi $\pi_{BCD}(r)$ sunt prezentate în fig.4.2. Joncţiunea lor este egală cu r(ABCD). Atunci egalitatea (4.2) e satisfăcută fiindcă:

(1) $|\pi_A(\sigma_{BC=b1c1}(r))|=|\pi_D(\sigma_{BC=b1c1}(r))|=2$, şi $|(\sigma_{BC=b1c1}(r)| = 4$

şi

(2) $|\pi_A(\sigma_{BC=b1c2}(r))|=|\pi_D(\sigma_{BC=b1c2}(r))|=2$, şi $|(\sigma_{BC=b1c2}(r)| = 4$.

$\pi_{BCA}(r)$	B	C	A
	b_1	c_1	a_1
	b_1	c_2	a_1
	b_1	c_1	a_2
	b_1	c_2	a_2

$\pi_{BCD}(r)$	B	C	D
	b_1	c_1	d_1
	b_1	c_1	d_2
	b_1	c_2	d_1
	b_1	c_2	d_2

Fig.4.2.

Proprietatea de mai sus poate fi utilizată într-o nouă definiţie a dependenţei multivaloare.

Definiţia 4.2. Fie r o relaţie asupra schemei R, $X,Y\subseteq R$ şi $Z = R \setminus XY$. Relaţia r(R) satisface dependenţa multivaloare $X\to\to Y$, dacă pentru orice X-valoare x şi XZ-valoare xz e satisfăcută egalitatea $\pi_Y(\sigma_{X=x}(r)) = \pi_Y(\sigma_{XZ=xz}(r))$.

Cu alte cuvinte, în cadrul relaţiei r(R) există o dependenţă multivaloare $X\to\to Y$, dacă şi numai dacă mulţimea valorilor lui Y corespunzătoare unei perechi xz depinde numai de X-valoarea x nu şi de valoarea lui Z.

4.2. Reguli de inferenţă ale dependenţelor multivaloare

Primele şase reguli sunt similare regulilor de inferenţă omonime ale dependenţelor funcţionale, însă numai primele trei conţin aceleaşi aserţiuni.

Fie o relaţie r cu schema R şi $W,X,Y,Z\subseteq R$.

DM1. Regula reflexivităţii. Dacă $Y\subseteq X$, atunci $X\to\to Y$.
Validitatea acestei reguli urmează din definiţia dependenţei multivaloare.

DM2. Regula incrementării. Dacă $Z\subseteq W$ şi $X\to\to Y$, atunci $XW\to\to YZ$.
Validitatea acestei afirmaţii reiese din definiţia 4.1 şi teorema 4.2.

DM3. Regula aditivităţii. Dacă $X\to\to Y$ şi $X\to\to Z$, atunci $X\to\to YZ$.
Demonstraţie. Fie în r sunt două tupluri t_1 şi t_2 ce au X-valori egale, $t_1[X]=t_2[X]$. Trebuie arătat că în r există un tuplu t, încât $t[X] = t_1[X]$, $t[YZ] = t_1[YZ]$ şi $t[U] = t_2[U]$, unde $U=R\setminus XYZ$.

Întrucât r(R) satisface dependența $X \rightarrow\rightarrow Y$, atunci r conține un tuplu t_3 şi $t_3[X] = t_1[X]$, $t_3[Y] = t_1[Y]$, $t_3[V] = t_2[V]$ pentru orice t_1 şi t_2 ce satisfac egalitatea $t_1[X]=t_2[X]$, unde $V=R \setminus XY$. Acelaşi raționament este şi pentru $X \rightarrow\rightarrow Z$: există în r(R) un tuplu t_4 care satisface $t_4[X] = t_1[X]$, $t_4[Z] = t_1[Z]$ şi $t_4[W] = t_3[W]$ (fiindcă $t_1[X] = t_3[X]$), unde $W = R \setminus XZ$.

Să arătăm că $t = t_4$. E evident că $t[X] = t_4[X]$. De asemenea $t_4[Z] = t_1[Z] = t[Z]$. Dar $t_4[Y \cap W] = t_3[Y \cap W] = t_1[Y \cap W] = t[Y \cap W]$ şi atunci $t_4[YZ] = t[YZ]$. Din $U \subseteq W \cap V$, urmează $t_4[U] = t_3[U] = t_2[U] = t[U]$. Întrucât $R=XYZU$, atunci $t_4=t$.

DM4. Regula proiectivității. Dacă $X \rightarrow\rightarrow Y$ şi $X \rightarrow\rightarrow Z$, atunci $X \rightarrow\rightarrow Y \cap Z$, $X \rightarrow\rightarrow Y \setminus Z$.

Demonstrație. Conform regulii DM3, $X \rightarrow\rightarrow Y$ şi $X \rightarrow\rightarrow Z$ implică $X \rightarrow\rightarrow YZ$. Aplicând teorema 4.1 asupra dependenței multivaloare $X \rightarrow\rightarrow YZ$, obținem $X \rightarrow\rightarrow R \setminus XYZ$. Aplicând regula DM3 asupra dependențelor $X \rightarrow\rightarrow R \setminus XYZ$ şi $X \rightarrow\rightarrow Z$, obținem $X \rightarrow\rightarrow (R \setminus XYZ)Z$. Dar conform teoremei 4.1, $X \rightarrow\rightarrow (R \setminus XYZ)Z$ implică $X \rightarrow\rightarrow R \setminus X(R \setminus XYZ)Z$. Determinatul ultimei dependențe se simplifică în felul următor (vezi fig. 4.3): $R \setminus X(R \setminus XYZ)Z = R \setminus X(R \setminus Y)Z = Y \setminus XZ = (Y \setminus Z) \setminus X$. Deci $X \rightarrow\rightarrow (Y \setminus Z) \setminus X$ şi conform teoremei 4.2 dependența $X \rightarrow\rightarrow Y \setminus Z$ este validă în r(R).

Am demonstrat validitatea regulii: dacă $X \rightarrow\rightarrow Y$ şi $X \rightarrow\rightarrow Z$, atunci $X \rightarrow\rightarrow Y \setminus Z$.

Să arătăm acum că, dacă $X \rightarrow\rightarrow Y$ şi $X \rightarrow\rightarrow Z$, atunci $X \rightarrow\rightarrow Y \cap Z$.

Dependența $X \rightarrow\rightarrow Y$ implică dependența $X \rightarrow\rightarrow R \setminus XY$. Combinând, conform regulii DM3, dependențele $X \rightarrow\rightarrow Y \setminus Z$ şi $X \rightarrow\rightarrow R \setminus XY$, obținem $X \rightarrow\rightarrow (R \setminus XY)(Y \setminus Z)$. Aplicând teorema 4.1 asupra ultimei dependențe multivaloare, obținem $X \rightarrow\rightarrow R \setminus X(R \setminus XY)(Y \setminus Z)$. Să examinăm partea dreaptă a dependenței multivaloare obținute, utilizând diagrama din fig.4.3.

$$R \setminus X(R \setminus XY)(Y \setminus Z) = R \setminus X(R \setminus Y)(Y \setminus Z) = Y \setminus X(Y \setminus Z) = (Y \cap Z) \setminus X.$$

Prin urmare, relația r(R) satisface dependența multivaloare $X \rightarrow\rightarrow (Y \cap Z) \setminus X$ şi, conform teoremei 4.2, satisface dependența $X \rightarrow\rightarrow Y \cap Z$.

DM5. Regula tranzitivității. Dacă $X \rightarrow\rightarrow Y$ şi $Y \rightarrow\rightarrow Z$, atunci $X \rightarrow\rightarrow Z \setminus Y$.

Demonstrație. Dacă vom arăta că $X \rightarrow\rightarrow YZ$ e validă în relația r, atunci asupra acestei dependențe şi dependenței $X \rightarrow\rightarrow Y$ poate fi aplicată regula DM4, pentru a obține dependența $X \rightarrow\rightarrow YZ \setminus Y$ sau $X \rightarrow\rightarrow Z \setminus Y$.

Notăm $W=R \setminus XYZ$ şi să arătăm că $X \rightarrow\rightarrow Y$ şi $Y \rightarrow\rightarrow Z$ implică $X \rightarrow\rightarrow YZ$. Adică, dacă în r sunt două tupluri t_1, t_2 şi $t_1[X] = t_2[X]$, atunci r conține un tuplu t ce satisface egalitățile $t[X] = t_1[X]$, $t[YZ] = t_1[YZ]$ şi $t[W] = t_2[W]$.

Întrucât dependența $X \rightarrow\rightarrow Y$ e validă în r, relația r conține un tuplu t_3 ce satisface $t_3[X] = t_1[X]$, $t_3[Y] = t_1[Y]$ şi $t_3[V] = t_2[V]$, unde $V = R \setminus XY$. Dar dependența $Y \rightarrow\rightarrow Z$ presupune că în r este un tuplu t_4 ce satisface condițiile $t_4[Y]=t_1[Y]$, $t_4[Z] = t_1[Z]$ şi $t_4[U] = t_3[U]$., unde $U = R \setminus YZ$.

Să arătăm că tuplul t_4 este tuplul căutat t. Întrucât $t_1[X]=t_2[X] = t_3[X] = t_4[X]$ e evident că $t_4[YZ] = t_1[YZ]$. Dat fiind faptul că $t_4[U]=t_3[U]$ şi $W \subseteq U \setminus X$, atunci $t_4[W] = t_3[W]$. Din $t_3[V] = t_2[V]$ şi $(U \setminus X) \subseteq V$ reiese că $t_3[W] = t_2[W]$. Deci, tuplul t_4 este cel căutat.

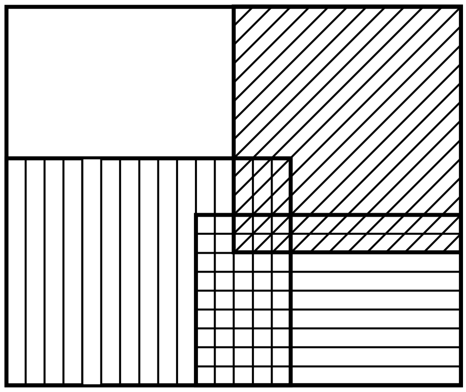

Fig. 4.3. - R, - X, - Y, - Z.

DM6. Regula pseudotranzitivităţii. Dacă $X\to\to Y$ şi $YW\to\to Z$, atunci $XW\to\to Z \setminus YW$.

Demonstrarea acestei reguli e similară regulii tranzitivităţii şi se propune în calitate de exerciţiu.

DM7. Regula complementării. Dacă $X\to\to Y$, atunci $X\to\to Z$, unde $Z = R \setminus XY$.

Validitatea acestei reguli este demonstrată de teorema 4.1.

Simbol	Denumire	Regulă
DM1	Reflexivitatea	$Y\subseteq X \Rightarrow X\to\to Y$
DM2	Incrementarea	$Z\subseteq W,\ X\to\to Y \Rightarrow XW\to\to YZ$
DM3	Aditivitatea	$X\to\to Y,\ X\to\to Z \Rightarrow X\to\to YZ$
DM4	Proiectivitatea	$X\to\to Y,\ X\to\to Z \Rightarrow X\to\to Y\cap Z$, $X\to\to Y \setminus Z$
DM5	Tranzitivitatea	$X\to\to Y,\ Y\to\to Z \Rightarrow X\to\to Z \setminus Y$
DM6	Pseudotranzitivitatea	$X\to\to Y,\ YW\to\to Z \Rightarrow$ $XW\to\to Z \setminus YW$
DM7	Complementarea	$X\to\to Y\Rightarrow X\to\to R \setminus XY$

Fig.4.4. Reguli de inferenţă ale dependenţelor multivaloare

În fig.4.4 sunt prezentate regulile de inferenţă ale dependenţelor multivaloare.

După cum s-a observat din demonstrările validităţii regulilor de inferenţă, unele reguli se pot deduce din celelalte. Similar mulţimii de reguli {DF1, DF2, DF5}, pentru dependenţele funcţionale, există submulţimi de reguli pentru dependenţele multivaloare, din care se deduc celelalte.

Teorema 4.4. Regulile DM3, DM4 şi DM6 se deduc din regulile DM1, DM2, DM5 şi DM7.

Demonstraţie. Să arătăm validitatea regulii DM3 utilizând DM1, DM2, DM5, DM7, adică $\{X\to\to Y, Y\to\to Z\} \mid\!-\ X\to\to YZ$. Într-adevăr:

$m_1{:=}X\to\to Z$ (dată),

$m_2{:=}X\to\to XZ$ (DM2:m_1),

$m_3{:=}X\to\to Y$ (dată),

m_4:=XZ→→YZ (DM2:m_3),
m_5:=XZ→→R \ XYZ (DM7:m_4),
m_6:=X→→R \ XYZ (DM5:m_2,m_5 şi fiindcă (R\XYZ)\XZ = R\XYZ)),
m_7:=X→→ R \ X(R \ XYZ) (DM7:m_6).
Din fig.4.3 se vede că R \ X(R \ XYZ) = YZ, deci m_7=X→→YZ.

Să demonstrăm că regula DM4 urmează din DM1, DM2, DM5, DM7. Validitatea expresiei {X→→Y, X→→Z}|−{X→→Y \ Z, X→→Y∩Z} este confirmată de următoarea consecutivitate de inferenţă.
m_1:=X→→Y (dată),
m_2:=X→→Z (dată),
m_3:=X→→R \ XY (DM7:m_1),
m_4:=X→→Z(R \ XY) (DM3:m_2, m_3),
m_5:=X→→Y \ Z (DM7:m_4 (vezi fig.4.3.)),
m_6:=X→→(Y\Z)(R\XY)=
X→→R \ (X∩Y) (DM3:m_3, m_5 (vezi fig.4.3.)),
m_7:=X→→ X∩Y (DM7:m_6).
Întrucât regula DM3 urmează din DM1, DM2, DM5 şi DM7 substituirea dependenţelor m_4 şi m_6 cu consecutivităţi de inferenţă corespunzătoare se propune în calitate de exerciţiu.

Şi, în sfârşit, să arătăm că pseudotranzitivitatea, DM6, urmează din DM1, DM2, DM5, DM7. Pentru aceasta vom defini o consecutivitate de inferenţă pentru a arăta validitatea expresiei {X→→Y, YW→→Z}|−XW→→Z \ YW, aplicând doar regulile DM2 şi DM5.
m_1:=X→→Y (dată),
m_2:=XW→→YW (DM2:m_1),
m_3:=YW→→Z (dată),
m_4:=XW→→Z \ YW (DM5:m_2, m_3).

4.3. Reguli de inferenţă mixte

Fie M o mulţime de dependenţe funcţionale şi multivaloare asupra schemei R şi m o dependenţă funcţională sau multivaloare. Sunt reguli de inferenţă ce pot fi utilizate pentru a verifica dacă M|=m.
Fie relaţia r(R) şi W,X,Y,Z⊆R.

DFM1. Regula replicării. Dacă X→Y, atunci X→→Y.
Validitatea acestei reguli este evidentă. Apelând la definiţia 4.2 a dependenţei multivaloare are loc egalitatea $\pi_Y(\sigma_{X=x}(r)) = \pi_Y(\sigma_{XZ=xz}(r))$, unde Z = R \ XY, fiindcă dependenţa funcţională presupune că pentru orice X-valori egale corespund Y-valori egale ale tuplurilor. Deci valorile pe care le primeşte atributul Z sunt imateriale.

Dependenţa funcţională reprezintă un tip particular de dependenţă multivaloare, pentru care mulţimea valorilor dependente este constituită dintr-o singură valoare, adică $\pi_Y(\sigma_{XZ=xz}(r))$ este o relaţie ce conţine nu mai mult de un tuplu.

DFM2. Regula coalescenţei. Dacă $X \to\to Y$ şi $Z \to W$, unde $W \subseteq Y$ şi $Y \cap Z = \emptyset$, atunci $X \to W$.

Demonstraţie. Deoarece $X \to\to Y$ şi dacă în r sunt două tupluri t_1, t_2, pentru care $t_1[X] = t_2[X]$, atunci în r există un tuplu t_3 ce satisface egalităţile $t_3[X] = t_1[X]$, $t_3[Y] = t_1[Y]$ şi $t_3[R \backslash XY] = t_2[R \setminus XY]$. Să observăm că, deoarece $Y \cap Z = \emptyset$, avem $Z \subseteq X(R \backslash XY)$ şi întrucât $t_3[X] = t_1[X] = t_2[X]$, rezultă că $t_3[Z] = t_2[Z]$.

Conform dependenţei funcţionale $Z \to W$ avem $t_3[W]=t_2[W]$. Însă $W \subseteq Y$, de unde urmează că $t_3[W]=t_1[W]=t_2[W]$ şi prin urmare dependenţa funcţională $X \to W$ e validă în r.

În fig.4.5 sunt prezentate regulile mixte de inferenţă.

Simbol	Denumire	Regulă
DFM1	Replicarea	$X \to Y \Rightarrow X \to\to Y$
DFM2	Coalescenţa	$X \to\to Y,\ Z \to W,\ W \subseteq Y,\ Y \cap Z = \emptyset,$ $\Rightarrow X \to W$

Fig.4.5. Reguli mixte de inferenţă

Definiţia 4.3. Fie relaţia r(R) şi $X,Y \subseteq R$. Dependenţa multivaloare $X \to\to Y$ se numeşte *trivială*, dacă $X \subseteq Y$ sau $XY=R$.

Astfel, regula de inferenţă DM1, generează numai dependenţe multivaloare triviale.

Aici ne vom limita numai la formularea unor rezultate despre completitudinea regulilor de inferenţă fără a aduce demonstrările corespunzătoare.

Teorema 4.4. Regulile DM1-DM7 formează o mulţime completă de reguli de inferenţă a dependenţelor multivaloare.

Teorema 4.5. Sistemul de reguli DF1, DF2, DF5, DM1, DM2, DM5, DM7, DFM1, DFM2 formează o mulţime închisă şi completă de reguli de inferenţă a dependenţelor funcţionale şi multivaloare.

4.4. Problema calităţii de membru

Fie M o mulţime de dependenţe funcţionale şi multivaloare şi m o dependenţă funcţională sau multivaloare. Deseori e necesar de determinat dacă urmează logic dependenţa m din M. Această problemă se numeşte problema calităţii de membru (membership problem). Bineînţeles că asemănător cu cazul numai dependenţelor funcţionale calcularea M^+, adică a mulţimii tuturor dependenţelor ce se deduc din M, este o procedură destul de laborioasă. Procesul necesită un timp care depinde exponenţial de dimensiunile mulţimii M. Similar noţiunii de închidere a unei mulţimi de atribute în raport cu o mulţime de dependenţe funcţionale, pentru mulţimea M se introduce noţiunea de bază a dependenţelor (dependency basis).

Definiţia 4.4. Fie relaţia r cu schema R, o mulţime M de dependenţe multivaloare şi funcţionale şi $X \subseteq R$. *Baza dependenţelor* a mulţimii de atribute X, notată cu DEP(X), în raport cu M este o partiţie a lui R: $DEP(X) = \{W_1,...,W_k | 1 \le k \le n, n=|R|\}$ ce satisface

(1) $X \to\to W_i \in M^+$, $1 \le i \le k$

şi

(2) $X \to\to Y \in M^+$, dacă şi numai dacă Y este uniunea a unor mulţimi W_i din DEP(X).

Să remarcăm că regula proiectivităţii pentru dependenţele multivaloare este mai slabă decât omologul său pentru dependenţele funcţionale. Proiectivitatea pentru dependenţele funcţionale ne spune că, dacă dependenţa $X \to Y$ e validă într-o relaţie, atunci e validă în această relaţie şi dependenţa $X \to A$, pentru orice $A \in Y$. Regula proiectivităţii pentru dependenţele multivaloare ne permite să spunem că, dacă $X \to\to Y$ e validă într-o relaţie, atunci dependenţa $X \to\to A$ e validă în aceeaşi relaţie numai dacă există în schema relaţiei o mulţime de atribute Z ce satisface condiţiile: dependenţa $X \to\to Z$ e validă şi sau $Z \cap Y = A$, sau $Y \setminus Z = A$.

Cu toate acestea regulile aditivităţii (DM3) şi proiectivităţii (DM4) ne permit să formulăm următoarea teoremă despre mulţimea de atribute Y, ca Y să depindă de o mulţime de atribute X, adică $X \to\to Y$. Această teoremă stă la temelia noţiunii de bază a dependenţelor şi a algoritmului de calculare a bazei dependenţelor descris ceva mai jos.

Teorema 4.6. Fie relaţia r(R) şi $X \subseteq R$. Mulţimea de atribute $R \setminus X$ poate fi partiţionată în submulţimi disjuncte $W_1,...,W_k$, încât pentru $Y \subseteq R \setminus X$ are loc $X \to\to Y$ atunci şi numai atunci când Y este uniunea a unor mulţimi W_i, $1 \le i \le k$.

Demonstraţie. La început fie că avem o singură submulţime constituită din atributele $R \setminus X$. Presupunem că la un oarecare pas de partiţionare am obţinut submulţimile $W_1,...,W_m$ şi e validă dependenţa $X \to\to W_i$, $1 \le i \le m$. Dacă $X \to\to Y$, dar Y nu este uniunea unor W_i, atunci substituim toate mulţimile W_i care satisfac expresiile $W_i \cap Y \ne \emptyset$ şi $W_i \setminus Y \ne \emptyset$, cu mulţimile $W_i \cap Y$ şi $W_i \setminus Y$, respectiv. Conform regulii proiectivităţii, dependenţele $X \to\to W_i \cap Y$ şi $X \to\to W_i \setminus Y$ sunt valide în r(R). Fiindcă o mulţime finită de atribute nu poate fi partiţionată la infinit, în cele din urmă, vom avea o partiţie încât Y din dependenţa $X \to\to Y$, va fi uniunea unor submulţimi W_i. Conform regulii aditivităţii, X va determina uniunea oricărei submulţimi din partiţie.

Remarcă. Dependenţele multivaloare triviale $X \to\to Y$, unde $Y \subseteq X$, ce se obţin din regula reflexivităţii nu sunt considerate în teorema de mai sus. Din definiţia bazei dependenţelor mulţimii de atribute X urmează că DEP(X) conţine şi toate atributele singulare A, unde $A \in X$.

Algoritmul DEPBASIS (R, X, M, DEP(X))

Intrare: R – o schemă relaţională;

X – o mulţime de atribute;

M – o mulţime de dependenţe funcţionale şi multivaloare definite pe schema R.

Ieşire: DEP(X) – baza dependenţelor mulţimii X în raport cu M.

begin

```
0        DEP(X):={A1,...,An} ( unde A1…An=X);
1        W1=R \ X;
2        k:=1;
3        for i=1 until k do
4            while ∃ S→→T∈M & S∩Wi=∅ & ∅⊂T∩Wi⊂Wi do
             begin
5                k:=k+1;
6                Wk:= T∩Wi;
7                Wi:= Wi \ Wk;
             end
8        DEP(X):=DEP(X) ∪{W1,...,Wk};
         return (DEP(X));
     end
```

Algoritmul DEPBASIS construieşte baza dependențelor pentru o mulțime dată de atribute X în raport cu o mulțime de dependențe M şi este de complexitate polinomială.

Exemplul 4.4. Fie R=ABCDEFGHI, X=HI şi

$M=\{m_1: HI \rightarrow A,$

$m_2: AEFH \rightarrow\rightarrow ABF,$

$m_3: CFI \rightarrow\rightarrow EH,$

$m_4: AI \rightarrow\rightarrow BCDI,$

$m_5: AHI \rightarrow\rightarrow B\}$. Să se calculeze DEP(X).

La început, conform liniilor 0,1,2,3 ale algoritmului, sunt setate următoarele valori pentru DEP(HI), W_1, k şi i:

DEP(HI)={H,I},

W_1=ABCDEFG,

k=1, i=1.

Variabila k indică numărul de blocuri, iar i blocul curent. Conform liniei 4 a algoritmului, vom considera pe rând dependențele din M în privința satisfacerii condițiilor corespunzătoare. Fiindcă i=1, este selectat blocul de atribute W_1. Pentru W_1, dependența m_1 satisface condițiile liniei 4: $HI \cap W_1 = \emptyset$ şi $\emptyset \subset A \cap W_1 \subset W_1$, unde HI şi A sunt, corespunzător, determinantul şi determinatul dependenței m_1. Deci, conform liniilor 5-7 pentru k, W_2 şi W_1 sunt setate următoarele valori:

k=2,

W_2=A,

W_1=BCDEFG.

Pentru blocul W_1 dependența m_4 e prima care satisface condițiile din linia 4 (dependența m_5, de asemenea, satisface). Prin urmare, după executarea liniilor 5-7, k, W_3 şi W_1 obțin valorile:

k=3,

W_3=BCD,

W_1=EFG.

Blocul W_2 nu e satisfăcut de nici o dependență. El va intra în DEP(HI). Atunci, în linia 3 variabila i creşte cu o unitate, adică i=2. Dar pentru W_2 nu sunt dependențe în M ce satisfac condițiile liniei 4 şi atunci i creşte cu o unitate, devenind 3. Pentru W_3 există dependența m_2. Prin urmare,

k=4,

W_4=B,
W_3=CD.

Blocul W_3 nu e satisfăcut de nici o dependență şi atunci i devine egal cu 4. Pentru blocul W_4, de asemenea, nu există nici o dependență. Aici generarea blocurilor sfârşeşte, şi în final vom obține:

DEP(HI)={H,I,W_1,W_2,W_3,W_4}={H, I, EFG, A, CD, B}.

Exemplul 4.5. Fie R şi M din exemplul 4.4. Să se confirme sau să se infirme că:

(a) $M|=HI\rightarrow\rightarrow AB$,

(b) $M|=HI\rightarrow\rightarrow H$,

(c) $M|=HI\rightarrow\rightarrow ABC$.

Pentru a verifica dacă $X\rightarrow\rightarrow Y$, se deduce din M, conform definiției bazei dependențelor, Y trebuie să fie uniunea unor mulțimi din DEP(X). În cazul nostru, DEP(HI)={H, I, EFG, A, CD, B}. Deci:

(a) $M|=HI\rightarrow\rightarrow AB$,

(b) $M|=HI\rightarrow\rightarrow H$,

(c) $M\not\models HI\rightarrow\rightarrow ABC$.

4.5. Acoperiri nonredundante cu dependențe multivaloare

Ca şi în cazul dependențelor funcționale, o mulțime de dependențe multivaloare este redundantă, dacă cel puțin una din dependențe este derivabilă din celelalte. Vom spune, de asemenea, că această dependență este redundantă în mulțimea dată. O acoperire nonredundantă a unei mulțimi de dependențe este o mulțime nonredundantă echivalentă. Problema construirii acoperirilor nonredundante pentru dependențele multivaloare se soluționează similar acoperirilor nonredundante ale dependențelor funcționale.

Fie M o mulțime de dependențe multivaloare. O acoperire nonredundantă a mulțimii M se construieşte de următoarea procedură simplă. Se selectează o dependență din M. Dacă ea se deduce din celelalte dependențe multivaloare, atunci se elimină din M. Acest proces continuă până nu poate fi găsită nici o dependență redundantă. Fiindcă în acest proces fiecare dependență multivaloare se examinează o singură dată, complexitatea acestui proces este proporțională complexității construirii mulțimii DEP(X) înmulțită la numărul de dependențe în M.

Un proces similar poate fi aplicat asupra unei mulțimi de dependențe funcționale şi multivaloare. Dar trebuie menționat că în acest caz ordinea, în care dependențele sunt examinate, determină care dependențe vor rămâne în acoperirea nonredundantă. Intuitiv, se observă că pentru utilizatori (de asemenea şi pentru SGBD) dependențele funcționale poartă mai multă informație decât dependențele multivaloare. De aceea e rezonabil pentru eliminare mai întâi de examinat dependențele multivaloare.

Fie F şi M mulțimi de dependențe funcționale şi multivaloare, respectiv. Fie F^1 o acoperire nonredundantă a tuturor dependențelor funcționale din $(F\cup M)^+$. Orice dependență funcțională din $(F\cup M)^+$ poate fi dedusă din F^1 cu algoritmul MEMBERSHIP. Apoi se elimină dependențele redundante multivaloare din $F^1\cup M$

pentru a obţine $F^1 \cup M^1$. Orice dependenţă multivaloare din $(F \cup M)^+ = (F^1 \cup M^1)^+$ poate fi dedusă din $F^1 \cup M^1$, utilizând numai algoritmul DEPBASIS.

Să menţionăm că $F^1 \cup M^1$ nu este neapărat nonredundantă, fiindcă unele dependenţe funcţionale pot deveni redundante în F^1, fiindcă dependenţele multivaloare M^1 n-au fost considerate când se construia F^1.

De asemenea F nu este neapărat o acoperire pentru dependenţele funcţionale din $(F \cup M)^+$. Regulile DFM1 şi DFM2 pot deduce dependenţe funcţionale noi ce pot fi adăugate la F. Dar această metodă e destul de ineficientă.

Există metode mai eficiente de generare a astfel de acoperiri ale dependenţelor funcţionale fiind prezente şi dependenţe multivaloare. Dar discuţiile asupra algoritmilor eficienţi depăşeşte cadrul acestei lucrări.

4.6. Dependenţe multivaloare incluse

Vom considera o generalizare a clasei dependenţelor multivaloare care se numeşte dependenţe multivaloare incluse (embedded multivalued dependencies).

Definiţia 4.5. Fie relaţia r asupra mulţimii de atribute R şi $R^1 \subseteq R$, $X,Y \subseteq R^1$. Dependenţa $X \to\to Y$ se numeşte multivaloare inclusă, dacă $X \to\to Y$ este dependenţă multivaloare în $\pi_{R^1}(r)$.

Este evident din definiţie că orice dependenţă multivaloare este dependenţă multivaloare inclusă. Exemplul de mai jos arată că afirmaţia inversă e greşită.

Exemplul 4.6. Considerăm relaţia r(ABCD) şi proiecţiile ei $\pi_{ABC}(r)$ şi $\pi_{ABD}(r)$ din fig.4.6. Proiecţiile $\pi_{ABC}(r)$ şi $\pi_{ABD}(r)$ satisfac respectiv dependenţele $A \to\to C$ şi $A \to\to D$. Conform regulii complementării (DM7), ambele proiecţii satisfac dependenţa $A \to\to B$. În acelaşi timp relaţia r(ABCD) nu satisface dependenţa $A \to\to B$, deci şi $A \to\to CD$ nu e validă în r.

r	A	B	C	D
	a	b	c	d
	a	b_1	c_1	d_1
	a	b	c_1	d
	a	b_1	c	d_1
	a	b	c	d_1
	a	b_1	c_1	d

$\pi_{ABC}(r)$	A	B	C
	a	b	c
	a	b_1	c_1
	a	b	c_1
	a	b_1	c

$\pi_{ABD}(r)$	A	B	D
	a	b	d
	a	b_1	d_1
	a	b	d_1
	a	b_1	d

Fig.4.6.

Trebuie menționat că nu există o mulțime completă de reguli de inferență pentru dependențele multivaloare incluse.

4.7. Dependențe multivaloare noncontradictorii

Definiția 4.6. Fie o mulțime M de dependențe multivaloare definite pe schema R. Vom spune că dependența multivaloare $X\rightarrow\rightarrow Y$ din M *separă* două atribute A şi B, dacă unul din atribute, fie A, este în Y, iar B este în R \ XY. Mulțimea M de dependențe multivaloare separă o mulțime de atribute V, dacă M separă două atribute din V.

Exemplul 4.7. Fie schema R=ABCD. Dependența $AB\rightarrow\rightarrow C$ separă atributele C şi D. Similar, dependența multivaloare $CD\rightarrow\rightarrow A$ separă A şi B. Dacă schema R=ABCD e proiectată în baza dependenței $AB\rightarrow\rightarrow C$ în două subscheme ABC şi ABD, atunci în ele sunt valide doar dependențele multivaloare $AB\rightarrow\rightarrow C$ şi $AB\rightarrow\rightarrow D$, respectiv. Nici într-o schemă nu e validă dependența $CD\rightarrow\rightarrow A$.

Problema descrisă în exemplul de mai sus e cunoscută sub denumirea de problemă a separării determinanților.

Definiția 4.7. Fie determinanții X,Y a două dependențe multivaloare şi fie DEP(X), DEP(Y). Determinanții X şi Y sunt *noncontradictorii* (conflict free), dacă

$$DEP(X) \setminus \{A|\ A\in X\} = \{V_1,\ldots,V_k, X_1,\ldots,X_i, Z_x, Y_1,\ldots,Y_j\},$$

$$DEP(Y) \setminus \{B|\ B\in Y\} = \{V_1,\ldots,V_k, Y_1,\ldots,Y_j, Z_y, X_1,\ldots,X_i\},$$

unde $\{V_1,\ldots,V_k\}\subseteq DEP(X\cap Y)$ şi $Z_x\cup X = Z_y\cup Y$. Vom spune că o mulțime M de dependențe multivaloare este noncontradictorie, dacă sunt noncontradictorii determinanții ai oricăror două dependențe din M.

Din definițiile 4.6 şi 4.7, urmează că mulțimea M de dependențe multivaloare este noncontradictorie, dacă nici o dependență din M nu separă determinantul altei dependențe multivaloare din M.

Exemplul 4.8. Fie R = ABCDEF şi M = $\{B\rightarrow\rightarrow A,\ B\rightarrow\rightarrow C,\ B\rightarrow\rightarrow DEF,\ D\rightarrow\rightarrow ABC,\ D\rightarrow\rightarrow EF,\ E\rightarrow\rightarrow ABCD,\ E\rightarrow\rightarrow F\}$. Să se arate că mulțimea M de dependențe multivaloare e noncontradictorie.

Trebuie să arătăm că orice pereche din mulțimea de determinanți {B, D, E} este noncontradictorie. Într-adevăr, DEP(B) \ B = {A, C, DEF}, unde $X = B$, $X_1 = A$, $X_2 = C$, $Z_x=D$, $Y_1 = E$, $Y_2 = F$; DEP(D) \ D = {EF, BAC}, unde $Y = D$, $Y_1 = E$, $Y_2 = F$, $Z_y = B$, $X_1 = A$, $X_2 = C$. Întrucât $X\cap Y = B\cap D = \varnothing$ şi $DEP(\varnothing)=\varnothing$, atunci $DEP(B)\cap DEP(D) \subseteq DEP(B\cap D)$. Este satisfăcută şi condiția $Z_x\cup X = Z_y\cup Y$, fiindcă $Z_x\cup X=\{D,B\}$ şi $Z_y\cup Y = \{B, D\}$. Prin urmare, B şi D sunt determinanți noncontradictorii. Similar poate fi arătat că sunt noncontradictorii şi perechile B, E şi D, E. Verificarea acestor din urmă se lasă în calitate de exercițiu.

Exemplul 4.9. Fie R = ABCD şi M = $\{AB\rightarrow\rightarrow C,\ AB\rightarrow\rightarrow D,\ CD\rightarrow\rightarrow A,\ CD\rightarrow\rightarrow B\}$. Atunci DEP(AB)\{A,B}={C, D}, DEP(CD) \ {C, D} = {A, B}. Definiția 4.7 nu e satisfăcută şi, prin urmare, mulțimea M de dependențe multivaloare e contradictorie.

Unii cercetători afirmă că, dacă o mulțime de dependențe multivaloare reflectă o parte a lumii reale, atunci mulțimea neapărat este noncontradictorie. Dar dacă mulțimea specificată nu e noncontradictorie, atunci înseamnă că o parte de semantică a lumii reale nu a fost capturată.

4.8. Dependențe joncțiune

Definiția 4.8. Fie U mulțimea universală de atribute şi fie relațiile r_1, ..., r_m definite pe schemele $R_1,...,R_m$, respectiv, unde $R_i \subseteq U$, $1 \leq i \leq m$. Joncțiunea relațiilor r_1, ..., r_m, notată cu $|\times|(r_1,...,r_m)$, este o relație definită pe schema $U^1=R_1...R_m \subseteq U$:

$$|\times|(r_1, ..., r_m)=\{t \mid t[R_i] = t_i \ \& \ t_i \in r_i(R_i),\ 1 \leq i \leq m\}.$$

De noțiunea joncțiune $|\times|(r_1, ..., r_m)$ a relațiilor $r_1,...,r_m$ e strâns legată noțiunea de dependență joncțiune asupra U^1, care este o constrângere asupra U^1 de forma $|\times|(R_1,...,R_m)$. Vom spune că dependența joncțiune este inclusă dacă $U^1 \subseteq U$. Dacă $U^1 = U$, vom spune simplu dependența joncțiune.

Definiția 4.9. Vom spune că relația r(U) satisface dependența joncțiune $|\times|(R_1,...,R_m)$ sau dependența joncțiune $|\times|(R_1...R_m)$ e validă în r(U), dacă r(U) se descompune fără pierderi pe schemele $R_1,...,R_m$, adică

$$r(U) = |\times|(\pi_{R1}(r), ..., \pi_{Rm}(r)) \qquad (4.3)$$

Exemplul 4.10. Relația r(ABC) satisface (vezi relația şi proiecțiile corespunzătoare în fig.4.7) dependența joncțiune $|\times|(AB, AC, BC)$, fiindcă $r(ABC) = |\times|(\pi_{AB}(r), \pi_{AC}(r), \pi_{BC}(r))$.

O condiție necesară, ca egalitatea (4.3) să fie satisfăcută, este $U=R_1...R_m$.

Este evident că dependența de joncțiune inclusă este o generalizare a dependenței joncțiune. La rândul său, apelând la teorema 4.3, despre condiția necesară şi suficientă ca o relație să se descompună fără pierderi în două proiecții, conchidem că dependența joncțiune este o generalizare a dependenței multivaloare. Într-adevăr, teorema 4.3 ne spune că r(R) satisface dependența multivaloare $X \rightarrow\rightarrow Y$, atunci şi numai atunci când r se descompune fără pierderi pe schemele XY şi XZ, unde $Z = R \setminus XY$. Condiția coincide cu definiția dependenței joncțiune $|\times|(XY, XZ)$.

r	A	B	C
	a_1	b_1	c_1
	a_1	b_2	c_2
	a_3	b_3	c_3
	a_4	b_3	c_4
	a_5	b_5	c_5
	a_6	b_6	c_5

$\pi_{AB}(r)$	A	B
	a_1	b_1
	a_1	b_2
	a_3	b_3
	a_4	b_3
	a_5	b_5
	a_6	b_6

$\pi_{AC}(r)$	A	C

$\pi_{BC}(r)$	B	C

a_1	c_1
a_1	c_2
a_3	c_3
a_4	c_4
a_5	c_5
a_6	c_5

b_1	c_1
b_2	c_2
b_3	c_3
b_3	c_4
b_5	c_5
b_6	c_5

Fig.4.7.

4.9. Tablouri

Vom descrie o metodă tabelară de testare a dependenței joncțiune, bazată pe noțiunea de tablou. Tabloul, de fapt este o relație, cu numai o singură deosebire – în loc de valori tuplurile conțin variabile. Variabilele sunt luate din două mulțimi: mulțimea variabilelor distincte şi mulțimea variabilelor nondistincte. Variabilele distincte sunt formate din litera a cu indice, iar cele nondistincte din litera b cu doi indici. Mulțimea de atribute ce denumesc coloanele este schema tabloului. O variabilă corespunde unei singure coloane. O coloană nu poate avea mai mult de o variabilă distinctă.

Definiția 4.10. Fie o dependență joncțiune $|\times|(R_1,...,R_m)$. *Tablou* al dependenței $|\times|(R_1,...,R_m)$ este o relație ce are un nume (fie tab) şi $|R_1...R_m| = |U|$ coloane, notate cu atributele din U, şi m tupluri câte unul pentru fiecare schemă R_i, $1 \leq i \leq m$. Tuplul t_i conține în coloana A_j variabila distinctă a_j, dacă A_j aparține lui R_i. Celelalte componente ale tuplului t_i sunt variabile nondistincte b_{ij} ce nu se repetă în alt tuplu. Deci $t_i[A_j] = a_j$, dacă $A_j \in R_i$ şi $t_i[A_j] = b_{ij}$, dacă $A_j \notin R_i$.

tab	A	B	C	D
	a_1	a_2	b_{13}	b_{14}
	b_{21}	a_2	a_3	b_{24}
	a_1	b_{32}	a_3	a_4

Fig.4.8. Tabloul dependenței $|\times|$(AB, BC, ACD)

Exemplul 4.11. Fie dependența joncțiune $|\times|$(AB,BC,ACD). Tabloul acestei dependențe arată ca în fig.4.8.

Tuplurile într-un tablou pot fi transformate conform unor reguli ce corespund anumitor clase de dependențe: funcționale şi joncțiune. Scopul unor astfel de transformări constă în obținerea unui tuplu alcătuit numai din variabile distincte. Tuplul format exclusiv din variabile distincte se numeşte *tuplu-scop*. Dacă în urma transformărilor tabloul conține tuplul scop, atunci dependența joncțiune e validă, adică joncțiunea este fără pierderi. Sunt cunoscute două tipuri de reguli de transformare a tabloului: F-reguli şi J-reguli.

F-reguli. Fie tab un tablou a unei dependențe joncțiune şi J o mulțime de dependențe joncțiune, multivaloare şi funcționale. Pentru orice dependență funcțională

$X \to Y$ din J modificarea tabloului tab se produce în felul următor. În tab se caută tuplurile ce coincid pe atributele din X. Fiind descoperite astfel de tupluri, se egalează variabilele atributelor din Y. Fie $t_i[X] = t_j[X]$ şi pentru $A \in Y$ $t_i[A] = \nu_1$, $t_j[A] = \nu_2$. Atunci,

(a) dacă una din variabilele ν_1 şi ν_2 este distinctă, atunci variabila nondistinctă e substituită de cea distinctă;
(b) dacă ambele variabile sunt nondistincte, atunci variabila cu indice mai mare e substituită de variabila cu indice mai mic.

J-reguli. Fie tab un tablou determinat de dependența joncțiune $|\times|(R_1,...,R_m)$ şi fie J o mulțime de dependențe joncțiune, multivaloare şi funcționale. Pentru orice dependență joncțiune $|\times|(S_1,...,S_k)$ din J la tabloul tab se adaugă tuplul t (dacă el nu este deja un tab) dacă $t \in |\times|(\pi_{S1}(tab),...,\pi_{Sk}(tab))$.

Trebuie de menționat că $S_1,...,S_k$ trebuie să formeze mulțimea universală U de atribute, adică $R_1,...,R_m = S_1,...,S_k$. Considerăm un raționament de executare eficientă a joncțiunii. Nu toate tuplurile ce vor participa la joncțiune sunt esențiale în formarea tuplului t ce se adaugă la tab. Dacă tuplurile t_i şi t_j sunt în tab şi dacă tuplul t_j are componente distincte pentru toate atributele pentru care are variabile distincte tuplul t_i, atunci t_j nu trebuie să participe la joncțiune.

Regulile de transformare a tabloului legate de dependențele multivaloare sunt de prisos, întrucât dependența multivaloare este un caz particular al dependenței joncțiune: dependența multivaloare $X \to\to Y$ este dependența joncțiune $|\times|(XY, XZ)$, unde $Z = R \setminus XY$.

Să aducem următorul algoritm de testare a dependențelor joncțiune.

Algoritmul LOSSLESSTEST (J, j)
Intrare: J - o mulțime de dependențe joncțiune, multivaloare şi funcționale;
$|\times|(R_1,...,R_m)$ - o dependență joncțiune.
Ieşire: Adevăr, dacă dependența joncțiune e validă (sau joncțiunea e fără pierderi) şi fals, în caz contrar.

(1) Se defineşte tabloul dependenței joncțiune $|\times|(R_1,...,R_m)$.
(2) Se aplică F-regulile şi J-regulile asupra tabloului până nu mai poate fi schimbat.
(3) După substituirile produse de toate dependențele din J, se verifică dacă tabloul conține un tuplu ce are toate componentele distincte. Dacă există în tablou tuplul-scop, atunci return (adevăr) în caz contrar - return(fals).

Exemplul 4.12. Fie relația r(ABCDEF) şi $J=\{A \to B, F \to E\}$. Să se verifice dacă dependența joncțiune $|\times|(ABDE,ACDF,BCEF)$ este validă în relația r(ABCDEF).

Aplicând algoritmul de mai sus, pasul (1) produce tabloul din fig.4.9(a). Acest tabel are trei tupluri şi şase coloane.

	A	B	C	D	E	F
t_1	a_1	a_2	b_{13}	a_4	a_5	b_{16}

t_2	a_1	b_{22}	a_3	a_4	b_{25}	a_6
t_3	b_{31}	a_2	a_3	b_{34}	a_5	a_6

Fig.4.9(a). Tabloul iniţial al dependenţei joncţiune |x|(ABDE, ACDF, BCEF)

Urmând F-regulile în aplicarea dependenţei $A \rightarrow B$ din J (pasul (2)), obţinem tabloul modificat din figura 4.9(b). În tablou tuplul $t_2[B] = a_2$, fiindcă $t_1[A] = t_2[A]$ în tabloul iniţial şi b_{22} a fost substituit de a_2.

	A	B	C	D	E	F
t_1	a_1	a_2	b_{13}	a_4	a_5	b_{16}
t_2	a_1	a_2	a_3	a_4	b_{25}	a_6
t_3	b_{31}	a_2	a_3	b_{34}	a_5	a_6

Fig.4.9(b). Tabloul modificat de $A \rightarrow B$

Aplicând apoi dependenţa funcţională $F \rightarrow E$ în pasul (2), obţinem tabloul modificat din fig.4.9(c). Aici $t_2[E] = a_5$, întrucât variabila b_{25} din tabloul precedent este substituită de a_5. Examinând tabloul obţinut, observăm că $t_2 = \langle a_1,a_2,a_3,a_4,a_5,a_6 \rangle$ este tuplul-scop. Deci relaţia r(ABCDEF) satisface dependenţa joncţiune |x|(ABDE, ACDF, BCEF) sau joncţiunea |x|(π_{ABDE}(r), π_{ACDF}(r), π_{BCEF}(r)) este fără pierderi.

	A	B	C	D	E	F
t_1	a_1	a_2	b_{13}	a_4	a_5	b_{16}
t_2	a_1	a_2	a_3	a_4	a_5	a_6
t_3	b_{31}	a_2	a_3	b_{34}	a_5	a_6

Fig.4.9(c). Tabloul modificat de $F \rightarrow E$

Exemplul 4.13. Fie relaţia r(ABCDE) şi mulţimea de dependenţe joncţiune J={|x|(AC,ABD,BDE), |x|(ABD,CDE)} validă în r. Să se arate că relaţia r(ABCDE) satisface şi dependenţa joncţiune |x|(AC,ABD,DE).

Construim tabloul iniţial tab pentru dependenţa joncţiune |x|(AC,ABD,DE) din fig. 4.10(a) ce constă din trei tupluri t_1, t_2 şi t_3, determinate respectiv de mulţimile AC, ABD şi DE.

tab	A	B	C	D	E
t_1	a_1	b_{12}	a_3	b_{14}	b_{15}
t_2	a_1	a_2	b_{23}	a_4	b_{25}
t_3	b_{31}	b_{32}	b_{33}	a_4	a_5

Fig.4.10(a) Tabloul iniţial tab

Aplicăm dependența joncțiune |x|(ABD,CDE) din J asupra tabloului tab. Proiectăm tabloul tab asupra schemelor ABD şi CDE. Obținem proiecțiile π_{ABD}(tab) şi π_{CDE}(tab) din fig.4.10(b) şi fig.4.10(c), respectiv.

π_{ABD}(tab)	A	B	D
	a_1	b_{12}	b_{14}
	a_1	a_2	a_4
	b_{31}	b_{32}	a_4

Fig.4.10(b). π_{ABD}(tab)

π_{CDE}(tab)	C	D	E
	a_3	b_{14}	b_{15}
	b_{23}	a_4	b_{25}
	b_{33}	a_4	a_5

Fig.4.10(c). π_{CDE}(tab)

În relația π_{ABD}(tab) tuplurile <a_1 b_{12} b_{14}> şi <b_{31} b_{32} a_4> nu sunt esențiale, fiindcă tuplul <a_1 a_2 a_4> le recapitulează. În relația π_{CDE}(tab) tuplurile reprezentative sunt <a_1 b_{14} b_{15}> şi <b_{33} a_4 a_5>. Tuplul <b_{23} a_4 b_{25}> nu va participa la joncțiune, fiindcă se substituie de tuplul <b_{33} a_4 a_5>. Joncționând tuplurile reprezentative din relațiile π_{ABD}(tab) şi π_{CDE}(tab), obținem tuplul t_4=<a_1 a_2 b_{33} a_4 a_5>. Formăm tabloul tab_1=tab $\cup\{t_4\}$ din fig.4.10(d).

tab_1	A	B	C	D	E
t_1	a_1	b_{12}	a_3	b_{14}	b_{15}
t_2	a_1	a_2	b_{23}	a_4	b_{25}
t_3	b_{31}	b_{32}	b_{33}	a_4	a_5
t_4	a_1	a_2	b_{33}	a_4	a_5

Fig.4.10(d) Tabloul tab_1= tab $\cup\{t_4\}$

Să examinăm acum acțiunea asupra tabloului tab_1 a dependenței |x|(AC,ABD,BDE) din J, proiectându-l pe schemele AC, ABD şi BDE (vezi fig.4.10(e), fig.4.10(f), fig.4.10(g), respectiv)

$\pi_{AC}(tab_1)$	A	C
	a_1	a_3
	a_1	b_{23}
	b_{31}	b_{33}
	a_1	b_{33}

Fig.4.10(e). $\pi_{AC}(tab_1)$

$\pi_{ABD}(tab_1)$	A	B	D
	a_1	b_{12}	b_{14}

a_1	a_2	a_4
b_{31}	b_{32}	a_4

Fig.4.10(f). π_{ABD}(tab1)

$\pi_{BDE}(tab_1)$	B	D	E
	b_{12}	b_{14}	b_{15}
	a_2	a_4	b_{25}
	b_{32}	a_4	a_5
	a_2	a_4	a_5

Fig.4.10(g). π_{BDE}(tab1)

Să observăm tuplurile ce vor participa la joncțiune. În $\pi_{AC}(tab_1)$ este tuplul $<a_1\ a_3>$, în $\pi_{ABD}(tab_1)$ e tuplul $<a_1\ a_2\ a_4>$ şi în $\pi_{BDE}(tab_1)$ e tuplul $<a_2\ a_4\ a_5>$. În urma joncțiunii acestor tupluri (câte unul din fiecare proiecție) obținem tuplul $t_5=<a_1a_2a_3a_4a_5>$. Se construieşte tabloul $tab_2=tab_1\cup\{t_5\}$ prezentat în fig.4.10(h).

tab_2	A	B	C	D	E
t_1	a_1	b_{12}	a_3	b_{14}	b_{15}
t_2	a_1	a_2	b_{23}	a_4	b_{25}
t_3	b_{31}	b_{32}	b_{33}	a_4	a_5
t_4	a_1	a_2	b_{33}	a_4	a_5
t_5	a_1	a_2	a_3	a_4	a_5

Fig.4.10(h). Tabloul $tab_2=tab_1\cup\{t_5\}$

Tabloul tab_2 conține tuplul-scop, t_5. Deci dependența joncțiune |x|(AC,ABD,DE) este validă în relația r(ABCDE) sau, ceea ce e echivalent, joncțiunea |x|($\pi_{AC}(r)$, $\pi_{ABD}(r)$, $\pi_{DE}(r)$) este fără pierderi.

Deci tablourile pot fi utilizate pentru soluționarea problemei calității de membru pentru dependențele joncțiune. Adică, dacă J e o mulțime de dependențe joncțiune, multivaloare şi funcționale şi j e o dependență joncțiune, atunci cu ajutorul tablourilor putem determina dacă j urmează logic din J. Această metodă este aplicabilă, numai dacă dependențele joncțiune din J sunt definite pe toată mulțimea universală de atribute U

Pentru mulțimea de dependențe joncțiune nu există o mulțime completă de reguli de inferență.

Definiția 4.11. Dependența joncțiune |x|(R_1,..., R_m) asupra U = $R_1...R_m$ este *trivială,* dacă e validă în orice relație r cu schema U.

4.10. Reguli de inferență ale dependențelor joncțiune

Considerăm următoarea mulțime de reguli de inferență ale dependențelor joncțiune. Este clar că mulțimea este închisă, dar e puțin probabil să fie şi completă.

DJ1. Dacă $R \subseteq U$, atunci $|\times|(R)$.

Această regulă ne spune că orice relație r(R) satisface dependența joncțiune $|\times|(R)$, fiindcă $r(R) = |\times|(\pi_R(r))$.

DJ2. Dacă $|\times|(R_1,\ldots,R_m)$ şi $Y \subseteq R_1\ldots R_m$, atunci $|\times|(R_1,\ldots,R_m,Y)$.

Validitatea acestei reguli reiese din egalitatea $|\times|(|\times|(R_1,\ldots,R_m)Y) = |\times|(R_1,\ldots,R_m,Y)$.

r	A	B	C
	a_1	b_1	c_1
	a_1	b_2	c_2
	a_2	b_1	c_3

Fig. 4.11.

Exemplul 4.14. Fie dependența joncțiune $|\times|(AC,BC)$ şi fie Y = AB. Atunci $|\times|(AC,BC)|=|\times|(AC,BC,AB)$, adică relația r(ABC) ce satisface dependența (vezi fig.4.11) $|\times|(AC,BC)$ satisface şi dependența $|\times|(AC,BC,AB)$. Aceste două dependențe sunt definite pe mulțimea universală de atribute, deci ele pot fi testate prin intermediul tabloului.

DJ3. Fie $|\times|(R_1,\ldots,R_m)$ şi $Y,Z \subseteq U$. Atunci $|\times|(R_1,\ldots,R_m,Y,Z)$ implică $|\times|(R_1,\ldots,R_m,YZ)$.

Exemplul 4.15. Fie în relația r(ABCDE) este validă dependența joncțiune $|\times|(AC,DE)$. Atunci $|\times|(AC,DE,ABD,BD) \;|=|\times|(AC,DE,ABD)$. Întrucât ambele dependențe antrenează toate atributele, deducția poate fi verificată cu ajutorul tabloului.

DJ4. Fie $|\times|(R_1,\ldots,R_m)$, $|\times|(S_1,\ldots,S_k)$ şi $Y = S_1\ldots S_k$. Atunci $|\times|(R_1,\ldots,R_m,Y)$ şi $|\times|(S_1,\ldots,S_k)$ implică $|\times|(R_1,\ldots,R_m,S_1,\ldots,S_k)$.

Exemplul 4.16. Fie $|\times|(AC,ABD)$, $|\times|(BD,DE)$ şi Y=BDE. Atunci $\{|\times|(AC,ABD,BDE), |\times|(BD,DE)\} \;|=\; |\times|(AC,ABD,BD,DE)$. Aici tabloul nu poate fi utilizat pentru verificarea implicației, fiindcă dependența joncțiune $|\times|(BD,DE)$ este inclusă, adică nu e definită pe mulțimea universală.

DJ5. Fie $|\times|(R_1,\ldots,R_m)$, $A \notin R_1\ldots R_m$ şi $Y \subseteq U$. Dacă $|\times|(R_1,\ldots,R_m,YA)$ atunci $|\times|(R_1,\ldots,R_m,Y)$.

Exemplul 4.17. Fie $|\times|(BCD)$, Y = DE şi $A \notin BCD$. Atunci $|\times|(BCD,DEA) \;|=\; |\times|(BCD,DE)$. Întrucât dependența $|\times|(BCD,DE)$ e inclusă, tabloul nu poate fi utilizat pentru verificarea acestei reguli.

Regulile DJ4 şi DJ5 conţin dependenţe joncţiune incluse. Vom combina aceste reguli pentru obţinerea unei reguli DJ6 echivalente, dar care antrenează numai dependenţe joncţiune definite pe mulţimea universală.

DJ6. Dacă $|\times|(R_1,\ldots,R_m,Y)$, $|\times|(S_1,\ldots,S_k)$ şi $\{A|$ A aparţine cel puţin la două scheme $S_i,S_j\}\subseteq Y$, atunci $|\times|(R_1,\ldots,R_m,S_1\cap Y,\ldots,S_k\cap Y)$.

Exemplul 4.18. $\{|\times|(AC,AB,BDE)\ |\times|(ABD,CDE)\}\ |=\{|\times|(AC,ABD,BD,DE)$. Această inferenţă poate fi verificată, utilizând tabloul.

Pentru dependenţele joncţiune nu este găsită o mulţime completă de reguli de inferenţă. Probabil că nici nu există.

4.11. Exerciţii

4.1. Fie relaţia r(ABC) din fig.4.12.

(a) Să se arate că relaţia r(ABC) satisface dependenţa multivaloare $A\to\to B$.

(b) Să se construiască din r(ABC) patru relaţii diferite, fiecare conţinând câte trei tupluri şi să se arate că dependenţa $A\to\to B$ nu e validă în nici o relaţie.

(c) Să se construiască din r(ABC) şase relaţii diferite a câte două tupluri. Să se arate că patru din ele satisfac dependenţa multivaloare $A\to\to B$, iar două nu o satisfac.

r	A	B	C
	a	b_1	c_1
	a	b_2	c_2
	a	b_1	c_2
	a	b_2	c_1

Fig.4.12.

4.2. Să se infirme că, dacă $Z\subseteq W$ şi $X\to\to Y$, atunci $XW\to YZ$.

4.3. Să se infirme că, dacă $X\to\to Y$, atunci $X\to Y$.

4.4. Fie relaţia r(ABCDEFGHIJ) şi M={$AB\to\to DEFG$, $CGJ\to\to ADHI$}. Să se calculeze DEP(ACGJ).

4.5. Fie relaţia r(ABC) şi J={$AB\to C$, $C\to A$}. Să se arate că relaţia r(ABC) nu satisface dependenţa joncţiune $|\times|(AB, AC)$.

4.6. Fie U=ABCDEFGH şi J={B→E, C→E, EF→G, G→ABH}. Să se arate că schema bazei de date Db={ABFG, BC, CDFH, AEH} se bucură de proprietatea joncţiunii fără pierderi.

4.7. Să se arate că dependenţa joncţiune |x|(R_1,...,R_m) asupra U= R_1...R_m este trivială, dacă şi numai dacă R_i=U pentru careva i, $1 \le i \le m$.

4.8. Să se completeze cu tupluri relaţia r(ABCDE) din fig.4.12, ca să satisfacă dependenţele multivaloare A→→BC şi CD→→BE

r	A	B	C	D	E
	a_1	b_1	C_1	d_1	e_1
	a_1	b_2	C_1	d_2	e_1
	a_2	b_1	C_1	d_1	e_2

Fig.4.12.

4.9. Să se descrie clasa de dependenţe multivaloare ce pot fi deduse din dependenţa funcţională X→Y.

4.10. Să se găsească DEP(AC) pentru mulţimea de dependenţe multivaloare definite pe schema R=ABCDEI.

4.11. Să se arate că o relaţie r(R) nu poate fi descompusă fără pierderi în două relaţii cu schemele R_1 şi R_2, unde $R_1 \neq R$ şi $R_2 \neq R$, atunci şi numai atunci când în r sunt valide numai dependenţe multivaloare triviale.

4.12. Fie relaţia r(ABCDE) şi fie F={A→C, B→C, C→D, DE→C, CE→A} o mulţime de dependenţe funcţionale valide în r. Să se arate că în r e validă şi dependenţa joncţiune |x|(AD, AB, BE, CDE, AE).

4.13. Fie R = ABCDE şi M={E→→B, AE→→C}. Să se arate că mulţimea M de dependenţe multivaloare este noncontradictorie.