MAT-KOL (Banja Luka)
ISSN 0354-6969 (p), ISSN 1986-5228 (o)
Vol. XXII (1)(2016), 61-69
www.imvibl.org/dmbl/dmbl.htm

PRIMJENA USMJERENIH HIPERGRAFOVA ZA PREDSTAVLJANJE FUNKCIONALNIH ZAVISNOSTI U RELACIONIM MODELIMA

Marko Đukanović[1], Dejan Radić[2]

Sažak: U radu je dat prijedlog za predstavljanje funkcionalnih zavisnosti između podskupova, atributa relacionih šema korištenjem usmjerenih, hipergrafova. Usmjereni hipergrafovi se predstavljaju binarnim matricama, što za posljedicu omogućuje primjenu bitskih operacija na njihovim elementima. Realizovani su i diskutovani algoritmi koji na bazi takvog koncepta rješavaju određene probleme iz oblasti funkcionalnih zavisnosti.

Ključne riječi fraze: Usmjereni hipergraf; matrica susjedstva; relaciona baza podataka; normalizacija; funkcionalne zavisnosti.

Abstract

The paper presents proposal for presentation of functional dependencies between subsets of attributes of database schemes by use of directed hypergraphs. Directed hypergraphs are represented by binary matrices, which enables the use of bitwise operations on their elements. Algorithms for certain problems in functional dependency domain were implemented and discussed.

AMS Mathematics Subject Classification (2010): 05C65, 68R10, 94C15.

Key words and phrases: Directed hypergraphs; relational database; normal form; normalization; adjacency matrix; functional dependences.

1 Uvod

Ako je relaciona baza podataka loše projektovana, onda može doći do problema i anomalija u radu sa bazom. Postupkom normalizacije se ogranizacija relacione baze podataka dovodi u određenu normalnu formu sa ciljem eliminacije ili minimizacije pomenutih problema i anomalija. Postoji više normalnih formi koje su definisane tokom razvoja relacionog koncepta i relacionih baza podataka: 1NF, 2NF, 3NF, BCNF itd ([4]).

Relaciona šema R predstavlja skup atributa u oznaci R$(A_1, A_2, ..., A_n)$. Relacija r(R) je podskup Dekartovog proizvoda domena atributa relacije R, $r(R) \subset domA_1 \times domA_2 ... \times domA_n$ [1].

[1] Prirodno-matematički fakultet, Mladena Stojanovića 2, 78000 Banja Luka, e-mail:marko.djukanovic@ymail.com

[2] Codaxy d.o.o., Voždovačka 3, 78000 Banja Luka, e-mail: dejanradic@live.com

Definicija 1.1 *Između dva podskupa atributa α i β relacione šeme R postoji funkcionalna zavisnost $\alpha \to \beta$, ako za bilo koje dvije torke t_1 i t_2 relacije r(R), koje imaju iste vrijednosti na α, torke t_1 i t_2 trebaju imati iste vrijednosti i na β, $t_1[\alpha] = t_2[\alpha] \Rightarrow t_1[\beta] = t_2[\beta]$ [2]. Takođe se kaže da α identifikuje β ili je β funkcionalno zavisan od α. Trivijalna funkcionalna zavisnost nastaje ako vrijedi $\beta \subset \alpha$.*

Definicija 1.2 *Ako je α podskup skupa atributa relacione šeme R ($\alpha \subset attrs(R)$), na kojoj je definisan skup funkcionalnih zavisnosti F, onda je α^+ zatvaranje skupa atributa α pod F. α^+ predstavlja skup svih atributa logički zavisnih od α.*

Ako je nad relacionom šemom R definisan skup funkcionalnih zavisnosti F, onda je $r(R)$ legalna relacija na šemi R, ako r zadovoljava funkcionalne zavisnosti iz skupa F . Superključ SK na relacionoj šemi R identifikuje sve attribute iz relacije R, odnosno $SK \to attrs(R)$.
U ovom radu izlažemo ideju predstavljanja funkcionalnih zavisnosti pomoću kombinatorne strukture- usmjereni hipergrafovi. Takođe se razmatra mogućnost memorijske reprezentacije koncepta korištenjem binarnih matrica. Rad je organizovan na sljedeći način: U sekciji 2 je izložen način predstavljanja skupova preko neoznačenih binarnih podataka. U sekciji 3 su definisani pojam i predložene mogućnosti memorijske reprezentacije hipergrafa i usmjerenog hipergrafa sa fokusom na funkcionalne zavisnosti. U sekciji 4 se daju realizacije određenih algoritama koji koriste opisane koncepte. Na kraju je dat zaključak i smjernice za buduće istraživanje.

2 Memorijska reprezentacija skupova

Relaciona šema R se sastoji od konačnog skupa atributa attrs(R). Taj skup je u relaciji (tabeli) r(R) naveden određenim redosljedom, tako da se može posmatrati kao struktura liste. Skup se u memoriji može predstaviti vektorom bita (eng. *bit array*) ako je poznat konačan broj elemenata skupa i njihov raspored. Primjera radi, neka je dat uređeni podskup skupa od maksimalno 5 elemenata $\{A, B, C, D, E\}$. Taj podskup se može predstaviti vektorom od 5 bita ili neoznačenim 5-bitnim podatkom. Svakom bitu odgovara određeni element skupa. Ako element pripada skupu, na njegovu poziciju u nizu postavimo 1, u suprotnom 0. Primjera radi, posmatrajmo podskup $\{B, D, E\}$. Njegova binarna reprezentacija je data vektorom $[0\ 1\ 0\ 1\ 1]^T$. Prazan skup je ekvivalentan *nula* vektoru $[0\ 0...0]^T$. Skup koji ima sve elemente ili potpun skup je ekvivalentan vektoru jedinica $[1\ 1...1]^T$. Dekadna vrijednost neoznačenog podatka od n bita koji odgovara potpunom skupu je $2^n - 1$. Nad vektorima sa binarnim elementima se mogu vršiti logičke (bitske) operacije koje su ekvivalentne operacijama nad skupovima uz uslov da vektori moraju biti iste dužine. Unija koresponduje operaciji disjunkcije (bitsko ILI). Presjek koresponduje logičkoj operaciji konjunkcije (bitsko I). Komplement je unarna operacija i odgovara operaciji negacije (invertovanja). Za utvrđivanje da li je skup X podskup skupa Y koristi se funkcija ((not A) and B) gdje su A i B bit vektori koji odgovaraju skupovima X i Y respektivno. Naime, ako su dva vektora (skupa) X i Y data u bitskoj formi sa A=[1 1 0]^T i B=[0 1 0]^T, jasno je da je Y podskup skupa X ($Y \subset X$). Provjera da li je Y podskup od X može se vršiti korištenjem navedene logičke funkcije za svaki par korespodentnih bita iz A i B.

A	B	(notA) and B
0	0	0
0	1	1
1	0	0
1	1	0

Tabela 1: ¬ A ∧ B

Ako je rezultat prikazane logičke funkcije vektora A i B *nula* vektor onda je Y podskup od X (što je u ovom slučaju ispunjeno). Primjetimo da se za provjeru da li je X podskup od Y ne dobija *nula* vektor.

3 Usmjereni hipergrafovi

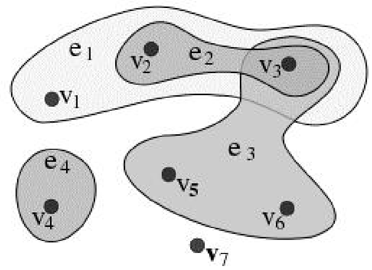
Primjena grafova za modelovanje funkcionalnih zavisnosti ne daje dobre rezultate. Ako bi funkcionalne zavisnosti uvijek bile oblika $A \to B$, gdje je lijevo samo jedan atribut, tada bi to bio dobar način. Graf se definiše kao G=(V,E), gdje je V skup čvorova, a E podskup skupa dvoelementnih podskupova od V . Hipergraf predstavlja uopštenje grafa. Sljedeće definicije prate [5].

Definicija 3.1 *Neka je* $X = \{x_1, x_2, ..., x_n\}$ *konačan skup. Hipergraf na X je familija* $H = \{E_1, E_2, ..., E_m\}$ *podskupova od X tako da vrijedi*
1.$E_i \neq \emptyset$
2.$\cup_{i=0}^{m} E_i = X.$
Prosti hipergraf je hipergraf $H=(E_1, E_2, ..., E_m)$ *kod koga vrijedi*

$$E_i \subset E_j \Longrightarrow i = j.$$

Elemente $x_1, x_2, ..., x_n$ *skupa X nazivamo čvorovima hipergrafa, dok skupove* $E_1, E_2, ..., E_m$ *nazivamo granama (hiperivicama).*

Hipergraf je k-uniforman ako sve hiperivice povezuju k čvorova. Ukoliko su grane hipergrafa E_j kardinalnosti 2, one se predstavljaju kao neprekidne krive koje spajaju čvorove (standardni graf), ukoliko je kardinalnosti 1, grana odgovara petlji a ukoliko ima više od 2 elementa, grana se vizuelizira prostom zatvorenom krivom.
Grafovi se memorijski zapisuju na različite načine, a najčešće matricama susjednosti (eng. adjacency matrix). Hipergrafu odgovara matrica incidencije (eng. incidence matrix) $A=(a_{ij})$, gdje kolona j matrice A odgovara grani E_j, a vrsta i odgovara elementu x_i. Pozicija a_{ij} je 0 akko $x_i \notin Ej$, inače je 1.



Slika 1: Primjer hipergrafa

Definicija 3.2 *Red hipergrafa H, u oznaci n(H), je broj čvorova hipergrafa H. Broj grana u H se označava sa m(H). Rang hipergrafa definišemo sa r(H)=* $\max_j |E_j|$.

Definicija 3.3 *Neka je* $x \in X$. *Neka je hipergraf H(x) konstruisan od grana u H koje sadrže čvor x. Stepen čvora x se definiše sa d(x)=dH(x)=m(H(x)).*

Na slici 1 je dat primjer hipergrafa, gdje skup čvorova X i skup hiperivica E odgovaraju sljedećim skupovima $X=\{x_1, x_2, x_3, x_4, x_5, x_6, x_7\}$ i $E=\{\{x_1, x_2, x_3\}$ $,\{x_2, x_3\}, \{x_3, x_5, x_6\}, \{x_4\}\}$. Jasno je da hipergraf sa slike 1 nije k-uniforman. Matrica incidencije A definiše pripadnost čvora datoj hiperivici, što je pokazano prikazujući a_{ij} sa

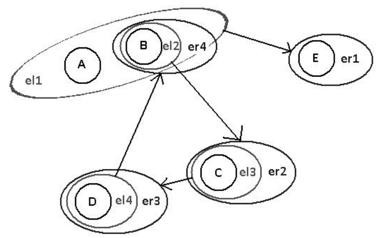
$$a_{ij} = \begin{cases} 0 & x_i \notin e_j \\ 1 & x_i \in e_j \end{cases}$$

Matrica incidencije koja odgovara hipergrafu H sa slike 1 je dimenzije 7×4 data sa

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Hipergraf ne definiše vezu između dva podskupa čvorova. Hipergraf definiše samo pripadnost čvora hiperivici. Sa njim se mogu predstaviti grupe atributa, tj. može se reći da skup $\{A, B\}$ pripada jednoj hiperivici, a skup $\{C, D\}$ drugoj. Pošto funkcionalna zavisnost oblika $AB \to CD$ predstavlja odnos između grupa čvorova (atributa), da bi se takva zavisnost predstavila pomoću hipergrafa, potrebno je uvesti veze između hiperivica. Takvu mogućnost ima usmjereni hipergraf kojeg definišemo u nastavku.

Definicija 3.4 *Direktna hipergrana (hiperivica) je uređen par I=(B,P) disjunktnih podskupova čvorova; B se naziva rep od I, dok se P naziva glava (koristimo*



Slika 2: Primjer usmjerenog hipergrafa

oznake T(I) i H(I) respektivno). Usmjereni hipergraf H=(V,E) je hipergraf kome su sve grane usmjerene ($E \subset P(V) \times P(V)$), gdje je P(V) partitivni skup od V.

E je skup hiperivica koje povezuju hiperivice neusmjerenog hipergrafa. Specijalan slučaj vezan za prazan skup u partitivnom skupu u ovom slučaju se neće razmatrati. Usmjereni hipergraf se može posmatrati kao uređeni par (L,R) gdje su L i R neusmjereni hipergrafovi sa istim čvorovima i korespodentnim hiperivicama. Definicije L=(V,EL) i R=(V,ER) su definicije lijevog i desnog hipergrafa, respektivno. EL je skup hiperivica lijevog, a ER je skup hiperivica desnog hipergrafa.
Bazirajući se na prethodno, lako je uočiti korespodenciju između usmjerenog hipergrafa i skupa funkcionalnih zavisnosti F. To je omogućeno uočavanjem korespodencije između funkcionalih zavisnosti $\alpha \to \beta$ i hiperivica usmjerenog hipergrafa. Iako je F strogo matematički gledano skup funkcionalnih zavisnosti, zbog jednostavnosti i razumljivosti koncepta, F će biti posmatran kao lista funkcionalnih zavisnosti. Svaka funkcionalna zavisnost $\alpha \to \beta$ ima lijevu stranu α i desnu stranu β, tako da se usmjereni hipergraf može posmatrati kao lista lijevih i desnih strana funkcionalnih zavisnosti. Kardinalnost lista lijevih i desnih strana je $|E| = |EL| = |ER| = k$. Skup EL se može posmatrati i kao lista hiperivica sa lijeve strane, tj. $EL = (el_i), i = 1..k$, a ER kao lista hiperivica sa desne strane $ER = (er_i), i = 1..k$. EL odgovara listi lijevih strana funkcionalnih zavisnosti iz liste F, dok ER odgovara listi desnih strana funkcionalnih zavisnosti iz F.

Neka V predstavlja skup atributa relacione šeme, i neka je kardinalnost skupa V ($|V| = n$). Svaka lijeva i desna strana funkcionalne zavisnosti se može posmatrati kao podskup od V. Za predstavljanje podskupova atributa na lijevim i desnim stranama funkcionalnih zavisnosti može se koristiti već opisani postupak predstavljanja skupova pomoću binarnih brojeva. Konsekventno, podskupovi atributa na lijevoj i desnoj strani liste funkcionalnih zavisnosti F mogu se predstaviti matricama incidencije LM i RM. Obe matrice su dimenzije $n \times k$, pri čemu je n odgovara broju čvorova $|V|$ (broju atributa), a k odgovara broju hiperivica $|E|$ (broju funkcionalnih zavisnosti u listi F). Za ilustraciju koncepta se koristi hipergraf sa slike 1. Data je lista funkcionalnih zavisnosti $F =$

($AB \to E, B \to C, C \to D, D \to B$) za skup atributa $ABCDE$ (radi jednostavnosti izostavljene su vitičaste zagrade za oznaku skupa). Posmatrajući listu F, imamo da su L=(V,EL) i R=(V,ER) date sa EL=($\{A, B\}, \{B\}, \{C\}, \{D\}$), a ER=($\{E\}, \{C\}, \{D\}, \{B\}$). Kardinalnosti su $|V| = n = 5$ i $|EL| = |ER| = k = 4$. Na slici 2 je prikazana vizuelna predstava (hipergraf) date liste funkcionalnih zavisnosti gdje primjećujemo povezanost elementa el_i sa korespodentnim er_i. Sa slike je lako formirati matrice incidencije LM i RM (dimenzija 5×4).

$$LM = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad RM = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

Ovakva matrično predstavljanje omogućava reprezentaciju funkcionalnih zavisnosti iz skupa (liste) F. Naime, ako se posmatra funkcionalna zavisnost $F_i, i = 1..k$ onda su lijeva i desna strana te funkcionalne zavisnosti predstavljene kolonama el_i i er_i matrica LM i RM. Primjera radi, četvrta kolona matrice LM ima jedinicu u četvrtoj vrsti, a RM ima jedinicu u četvrtoj koloni i drugoj vrsti, što predstavlja zavisnost $D \to B$. Matrica RM(u ovom slučaju) u svakoj koloni ima po jednu jedinicu jer svaka desna strana ima po jedan atribut. S obzirom da svaka kolona navedenih matrica predstavlja jedan skup atributa, evidentno je da će se kolonama pristupati u cjelini. Zbog toga preporučljivo je predstavljanje navedenih matrica po kolonama (eng. column major) u njihovoj memorijskoj reprezentaciji. Alternativni način predstavljanja koncepta je određen sa jednom matricom $n \times k$ gdje se na poziciji (i, j) nalazi -1 ako je dati atribut na lijevoj strani date funkcionalne zavisnosti, a 1 ako je na desnoj strani [3]. Takav pristup ima ograničenja jer ne omogućava predstavljanje trivijalnih funkcionalnih zavisnosti i rad sa bitima.

4 Algoritamska rješenja

U ovoj sekciji je izložena konstrukcija određenih algoritama nad opisanom strukturom podataka. Navedeni koncept predstavljanja liste funkcionalnih zavisnosti F preko matrica LM i LR, omogućava rad sa bitima. Formiraćemo algoritam koji generiše listu identifikatora trivijalnih funkcionalnih zavisnosti u listi F. Trivijalna funkcionalna zavisnost postoji ako $\alpha \to \beta \wedge \beta \subseteq \alpha$ (desna strana funkcionalne zavisnosti je podskup korespodentne lijeve strane). Provjera podskupa se vrši već prethodno ukazanim postupkom uz pomoć logičke funkcije (not A) and B. Lista identifikatora TFZ se inicijalizuje nad praznom listom. Za svaku funkcionalnu zavisnost iz F provjeravamo navedeni uslov. Ako je uslov ispunjen funkcijom *add* u listu indentifikatora TFZ se dodaje identifikator i za funkcionalnu zavisnost u toj iteraciji. Stil pisanja pseudo-koda kombinacija je jezika Pascal-a i C-a. Takav stil će se nadalje koristiti u radu. Slijedi pseudo-kod za opisani algoritam:

```
function trivijalne_fz()
{
TFZ=∅;
for i=1 to k do
if(not LM[*,i] and RM[*,i])=null then
add(TFZ,i);
return TFZ;
}
```

Za ispis funkcionalnih zavisnosti predstavljenih preko matrica LM i RM koristimo sljedeći pseudo-kod :

```
function ispis_fz()
{
for i=1 to k do
output(LM[*,i]+"→"+RM[*,i]);
}
```

U nastavku dajemo konstrukciju jednog od algoritama za nalaženje zatvaranja skupa atributa. Zatvaranje skupa atributa α (u oznaci α^+) se dobija iz skupa α sukcesivnim prolaskom kroz funkcionalne zavisnosti do dobijanja svih atributa zavisnih od α. Ako je data funkcionalna zavisnost $\beta \to \gamma$ pri čemu je β podskup od α^+, onda i γ treba biti podskup od α^+. Ako je $\alpha^+ =$ attrs(R), onda je α superključ na R. Provjera podskupa se vrši primjenom već opisanog postupka za svaku kolonu matrice LM sve dok se ne dobije ista vrijednost za α^+ u dvije sukcesivne iteracije. Ako je vrijednost logičke funkcije (not A) and B *nula* vektor zaključujemo da je $\beta \subseteq \alpha^+$. U tom slučaju se na α^+ dodaje γ (unija koja odgovara logičkoj operaciji ILI). Opisani postupak predstavljamo sljedećim pseudo- kodom

```
function zatvaranje(α)
{
α_plus= α;
do {
α_plus_preth= α_plus;
for i=1 to k do
if (not  α_plus  and LM[*,i])=null then
  α_plus= α_plus  or RM[*,i];
}while (α_plus_preth≠ α_plus);
return α_plus;
}
```

Slijedi konstrukcija algoritma za nalaženje svih superključeva. Algoritam podrazumijeva računanje zatvaranja za svaki podskup skupa atributa. Potrebno je odrediti samo one skupove atributa koji identifikuju attrs(R), odnosno čije je zatvaranje sadrži sve atribute. Reprezentacija skupa je realizovana vektorom bita ili binarnim brojem (na kojem se vrše aritmetičke operacije). Dekadna vrijednost binarnog neoznačenog podatka od n bita uzima vrijednosti iz opsega $[0, 2^n - 1]$. Te vrijednosti korespondiraju podskupovima skupa od n elemenata prema prethodno izloženom konceptu. Ako se, polazeći od 0, vrši inkrementovanje (uvećanje za 1) do $2^n - 1$, dobiće se vrijednosti koje odgovaraju svim podskupovima skupa od n elemenata. Za nalaženje skupa superključeva SK, polazna vrijednost SK je prazan skup. Za svaki podskup atributa potrebno je izračunati zatvaranje i provjeriti da li je zatvaranje jednako skupu atributa attrs(R) koji odgovara vektoru jedinica (VJ). U svakoj iteraciji se vrši data provjera zatvaranja za binarnu vrijednost brojača i. Brojač i se može posmatrati kao binarni podatak od n bita. Prolazak kroz sve vrijednosti binarnog podatka od n bita (ne uključujuću nulu) je ekvivalentan generisanju svih podskupova atributa (ne uključujući prazan skup). Algoritam za nalaženje skupa superključeva je dat sljedećim pseudo-kodom

```
function superkljucevi()
{
SK=∅;
for i=1 to 2^n − 1 do
if zatvaranje(i)= VJ then
SK = SK ∪  i;
return SK;
}
```

Posljednja iteracija prethodnog pseudo-koda je odbačena jer je attrs(R)→ attrs(R) trivijalna funkcionalna zavisnost. Tako se SK može inicijalizovati na skup sa elementom attrs(R) koji odgovara vektoru jedinica i ne izvršiti posljednju iteraciju.

5 Zaključak

Loša organizacija relacionih baza podataka može dovesti do problema i anomalija u toku rada sa relacionim bazama podataka zbog čega se vrši normalizacija organizacije relacione baze podataka. Funkcionalne zavisnosti se koriste za specifikaciju određenih normalnih formi organizacije relacionih baza podataka, kao i za konstrukciju različitih algoritama. Relacioni koncept podržava specifikaciju samo specijalnih formi ograničenja putem funkcionalnih zavisnosti (primarni, jedinstveni ključevi). U ovom radu je izložena mogućnost implementacije bilo koje forme funkcionalnih zavisnosti putem usmjerenih hipergrafova. Izloženi su algoritmi koji na bazi matričnih struktura za prikaz i čuvanje hipergrafova mogu da identifikuju karakteristična svojstva relacija i njihovih atributa. U budućem istraživanju biće analizirana mogućnost primjene ovog koncepta za specifikaciju ograničenja na realnim bazama podataka.

Literatura

[1] Hossein Saiedian, Thomas Spencer: An Efficient Algorithm to Compute the Candidate Keys of a Relational Database Schema, The Computer Journal Volume 39, issue 2, January 1996, pages 124-132

[2] S. Marić, D. Brđanin: Relacione baze podataka, Elektrotehnički fakultet, Banja Luka, BiH, 2012, ISBN 978-99955-46-07-6

[3] Giorgio Gallo, Giustino Longo, Stefano Pallottino, Sang Nguyen: Directed hypergraphs and applications, Journal Discrete Applied Mathematics - Special issue: combinatorial structures and algorithms archive Volume 42, Issue 2-3, April 27, 1993, Pages 177- 201, Elsevier Science Publishers B. V. Amsterdam, The Netherlands

[4] Gordana Pavlović Lažetić, Osnove relacionih baza podataka (drugo izdanje), Matematički fakultet Beograd, 1999, ISBN 86-7589-011-7

[5] C. Berge: Hypergraphs: Combinatorics of Finite Sets, North-Holland, 1989, ISBN 978-0444548887

Primljeno u redakciju 28.3.2015; Revidirana verzija 02.10.2015. i 05.01.2016.; Dostupno na internetu 11.01.2016.