2. Glava

Normalizacija

Ovo poglavlje predstavlja logični i prirodni nastavak prve glave, u kojoj su opisani motivi i osnovni principi projektovanja šeme baze podataka. Posvećeno je normalnim formama i normalizaciji.

Definisanjem normalnih formi, u ovom poglavlju se daje odgovor na pitanje do kog nivoa treba vršiti dekompoziciju šeme univerzalne relacije. Normalizacija je metoda projektovanja skupa šema relacija šeme relacione baze podataka. Postupak je strogo formalan, a njegov krajnji cilj je zamena šeme univerzalne relacije skupom šema relacija sa poželjnim osobinama. U te poželjne osobine spadaju: određena normalna forma, spoj bez gubitaka, konzervacija skupa obeležja i skupa funkcionalnih zavisnosti.

Eliminisanje anomalija ažuriranja predstavlja osnovni razlog za primenu normalizacije. Saglasno tome, normalizacijom se ostvaruje jedan od preduslova za efikasnu kontrolu integriteta baze podataka. Postoje dva osnovna postupka normalizacije. To su algoritam dekompozicije i algoritam sinteze, koji su detaljno i precizno obrađeni u ovom poglavlju.

2.1. Normalne forme

Postizanje određene ***normalne forme*** predstavlja jedan od ciljeva dekomponovanja šeme univerzalne relacije. U teoriji projektovanja šeme relacione baze podataka je definisan veći broj normalnih formi. Na ovom mestu će biti razmatrano sledećih šest: prva normalna forma, druga normalna forma, treća normalna forma, Boyce-Coddova normalna forma, četvrta normalna forma i peta normalna forma.

Prva i druga normalna forma imaju samo didaktički značaj. Najveći značaj za praksu projektovanja šeme baze podataka imaju: treća, Boyce-Coddova i, eventualno, četvrta normalna forma. Značaj pete normalne forme se može označiti kao čisto teoretski. Prve tri i Boyce-Coddova normalna forma se definišu isključivo na osnovu funkcionalnih zavisnosti. Četvrta normalna forma je zasnovana na višeznačnim, a peta na zavisnostima spoja.

Pri definisanju prve tri i Boyce-Coddove normalne forme šeme relacije (R, C), smatraće se da skup funkcionalnih zavisnosti $\mathcal{F} \subseteq C$ sadrži sve funkcionalne zavisnosti, definisane na skupu obeležja R.

2.1.1. Prva normalna forma

Za definisanje prve normalne forme se koristi pojam ***elementarnog obeležja***, u smislu da je to ono obeležje čiji domen sadrži ***atomične*** elemente. Elementi domena se nazivaju atomičnim ili ako ne predstavljaju torke reda $n \geq 2$, ili ako se struktura elemenata ne posmatra. Saglasno rečenom, obeležje je elementarno ili ako se dalje ne može dekomponovati na komponente, koje takođe predstavljaju obeležja, ili ako posmatrana primena ne zahteva dekompoziciju. U svakom slučaju, ne može se tvrditi da je definicija ovog pojma precizna. Međutim, insistiranje na elementarnim obeležjima, ima svoje opravdanje. Dekomponovanje složenog na elementarna obeležja može dovesti do definisanja onih funkcionalnih zavisnosti koje se inače ne bi mogle izraziti.

Primer 2.1. Posmatra se skup obeležja $\mathcal{U} = \{MBG, IME, PRZ, ADR, TEL\}$, gde je MBG matični broj građana. Neka u skupu $\mathcal{U}$ važi sledeći skup funkcionalnih zavisnosti $\mathcal{F} = \{MBG \rightarrow IME+PRZ+ADR+TEL\}$. Obeležja ADR i TEL predstavljaju skupove obeležja: $ADR = \{PTT, MESTO, ULICA, KBROJ\}$ i $TEL = \{POZBR, BRTEL\}$. Ako se u skupu $\mathcal{U}$ složena obeležja ADR i TEL zamene elementima odgovarajućih skupova obeležja, u tako dobijenom skupu obeležja $\mathcal{U}'$ važi i funkcionalna zavisnost $PTT \rightarrow POZBR$. □

Definicija 2.1. Šema relacije je u ***prvoj normalnoj formi***, ako su sva njena obeležja elementarna. Šema baze podataka je u prvoj normalnoj formi, ako su sve njene šeme relacija u prvoj normalnoj formi. □

Primer 2.2. Šema relacije iz primera 1.1 prvog poglavlja ove knjige, je u prvoj normalnoj formi. Za pojave nad šemom relacije, koja je u prvoj, ali ne i u nekoj od viših normalnih formi, karakteristična je pojava anomalija ažuriranja. □

U daljem tekstu će se smatrati da je svaka šema relacije u, barem, prvoj normalnoj formi.

2.1.2. Druga normalna forma

Druga normalna forma predstavlja mogući metodološki međukorak između prve i treće normalne forme. Retko predstavlja cilj projektovanja šeme relacione baze podataka.

Definicija 2.2. Šema relacije (R, C) sa skupom ključeva $\mathcal{K}$ je u ***drugoj normalnoj formi***, ako, za $\mathcal{F} \subseteq C$, važi:

$$(\forall X \in \mathcal{K})((A \notin X \wedge A \in R) \Rightarrow (\forall Y \subset X)(Y \to A \notin \mathcal{F}^+)).$$

Šema baze podataka je u drugoj normalnoj formi, ako je svaka njena šema relacije u drugoj normalnoj formi. □

Netrivijalna funkcionalna zavisnost $X \to A$ se naziva ***neredukovanom*** ili ***nepotpunom***, ako postoji $Y \subset X$ takvo, da važi $Y \to A \in \mathcal{F}^+$. Ako za svako $Y \subset X$ važi $Y \to A \notin \mathcal{F}^+$, funkcionalna zavisnost $X \to A$ se naziva ***redukovanom*** ili ***potpunom***. Šema relacije je u drugoj normalnoj formi, ako je svako neprimarno obeležje skupa obeležja R u potpunoj funkcionalnoj zavisnosti od svakog ključa šeme relacije (R, C).

Primer 2.3. Šema relacije *Fakultet* iz primera 1.1 nije u drugoj normalnoj formi, jer je, na primer, nepotpuna funkcionalna zavisnost $BRI+OZP \to IMS$ logička posledica definisanog skupa funkcionalnih zavisnosti. Pri tome, $BRI+OZP$ je jedan ključ šeme relacije *Fakultet*, a *IMS* ne pripada nijednom ključu. Šema relacije

$$Nast\_Pred(\{OZN, OZP, NAP, PRN\}, \{OZN \to OZP+NAP+PRN, OZP \to NAP, NAP \to OZP\}),$$

sa ključem *OZN* je u drugoj normalnoj formi, jer je svako neprimarno obeležje u potpunoj funkcionalnoj zavisnosti od svakog ključa. Šema relacije *Nast_Pred* predstavlja jednu od šema relacija, dobijenih dekomponovanjem šeme relacije *Fakultet* na osnovu funkcionalne zavisnosti $OZN \to OZP+NAP+PRN$. □

Pojave nad šemom relacije u drugoj, ali ne i u nekoj višoj normalnoj formi i dalje poseduju anomalije ažuriranja, kako to ilustruje i sledeći primer.

r(Nast_Pred) =

OZP	*NAP*	*OZN*	*PRN*
p_1	*Mat*	n_1	*Han*
p_2	*Fiz*	n_2	*Kun*
p_1	*Mat*	n_3	*Pap*
p_3	*Meh*	n_4	*Kiš*
p_4	*Mat*	n_5	*Car*

Slika 2.1.

Primer 2.4. Na slici 2.1 je prikazana pojava nad šemom relacije *Nast_Pred*, dobijena projektovanjem relacije *Fakultet* sa slike 1.1 na skup obeležja šeme relacije *Nast_Pred*. Analiza efekata ažuriranja ove relacije pokazuje:

- da se u relaciju ne mogu upisati podaci o novom predmetu, ako se ne zna nastavnik koji će ga predavati,

- da se brisanjem podataka o, na primer nastavniku (n_2, *Kun*), gube i podaci o predmetu (p_2, *Fiz*) i
- da modifikacija naziva predmeta, na primer (p_1, *Mat*) u (p_1, *Mal*), zahteva pristupanje svim torkama sa $OZP = p_1$. □

2.1.3. *Treća normalna forma*

Treća normalna forma predstavlja najčešći praktični cilj normalizacije. Dovođenjem skupa šema relacija u treću normalnu formu, eliminiše se većina, ali ne i svi uzroci anomalija ažuriranja. Osim toga, treća normalna forma je najviša normalna forma kod koje se za svaki skup funkcionalnih zavisnosti može isprojektovati takav skup šema relacija da zadovoljava uslov (1.2).

Definicija 2.3. Šema relacije $(R, \mathcal{C})$ sa skupom ključeva $\mathcal{K}$ je u ***trećoj normalnoj formi***, ako, za $\mathcal{F} \subseteq \mathcal{C}$, važi:

$$(2.1)\quad (\forall X \to A, X \to Y, Y \to A \in \mathcal{F}^+)((X \in \mathcal{K} \wedge (\forall X \in \mathcal{K})(A \notin X) \wedge A \notin Y) \Rightarrow Y \to X \in \mathcal{F}^+).$$

Šema baze podataka je u trećoj normalnoj formi, ako je svaka njena šema relacije u trećoj normalnoj formi. □

Netrivijalna funkcionalna zavisnost $X \to A$ se naziva ***tranzitivnom***, ako važi $X \to Y$, $Y \to A \in \mathcal{F}^+$ i $Y \to X \notin \mathcal{F}^+$. Saglasno rečenom, šema relacije je u trećoj normalnoj formi, ako se svako neprimarno obeležje skupa R nalazi u netranzitivnoj funkcionalnoj zavisnosti od svakog ključa X šeme relacije $(R, \mathcal{C})$.

Postoji nekoliko ekvivalentnih definicija treće normalne forme. Naredna se odlikuje jednostavnošću, a pogodna je i za uspostavljanje veze između treće i Boyce- Coddove normalne forme.

Definicija 2.4. Šema relacije $(R, \mathcal{C})$ sa skupom ključeva $\mathcal{K}$ je u ***trećoj normalnoj formi***, ako, za $\mathcal{F} \subseteq \mathcal{C}$, važi

$$(2.2)\qquad (\forall Y \to A \in \mathcal{F})(((\forall X \in \mathcal{K})(A \notin X) \wedge A \notin Y) \Rightarrow (\exists X \in \mathcal{K})(X \subseteq Y)). \ \square$$

Prema definiciji 2.4, šema relacije $(R, \mathcal{C})$ je u trećoj normalnoj formi, ako svaka netrivijalna funkcionalna zavisnost u $\mathcal{F}^+$, čija desna strana sadrži neprimarno obeležje, na levoj strani sadrži ključ šeme relacije.

Stav 2.1. Uslov (2.1) je ekvivalentan uslovu (2.2).

Dokaz. ($\Rightarrow$) Da bi se pokazalo da (2.1) implicira (2.2), pretpostavlja se da, suprotno uslovu (2.2), važi $(\exists Y \to A \in \mathcal{F})((\forall X \in \mathcal{K})(A \notin X) \wedge A \notin Y \wedge (\forall X \in \mathcal{K})(X \not\subseteq Y))$. Neka je X jedan ključ šeme relacije $(R, \mathcal{C})$, tada važi: $X \to A$, $X \to Y$, $Y \to A \in \mathcal{F}^+$ i, pošto Y nije superključ, važi $Y \to X \notin \mathcal{F}^+$, te se zaključuje da ne važi uslov (2.1).

($\Leftarrow$) Neka važi (2.2). Tada je Y superključ šeme relacije (R, C). Za $X \in \mathcal{K}$ važi $X \rightarrow A$, $X \rightarrow Y$, $Y \rightarrow A$, $Y \rightarrow X \in \mathcal{F}^{+}$, te se zaključuje da važi (2.1). □

Nepotpuna funkcionalna zavisnost predstavlja poseban slučaj tranzitivne zavisnosti. Naime, ako je netrivijalna funkcionalna zavisnost $X \rightarrow A \in \mathcal{F}$ nepotpuna, tada postoji Y takvo da je $Y \subset X$ i važi $Y \rightarrow A \in \mathcal{F}^{+}$. Na osnovu pravila izvodjenja $\mathfrak{F}_1$ (refleksivnost), sledi $X \rightarrow Y \in \mathcal{F}^{+}$ i $Y \rightarrow X \notin \mathcal{F}^{+}$. Saglasno tome, funkcionalna zavisnost $X \rightarrow A$ je tranzitivna. Naravno, tranzitivna zavisnost ne mora biti i nepotpuna. Isto tako, ako skup funkcionalnih zavisnosti ne sadrži nepotpunu, ne znači da ne sadrži tranzitivnu funkcionalnu zavisnost.

***Stav** 2.2.* Šema relacije u trećoj normalnoj formi je i u drugoj normalnoj formi. □

***Primer** 2.5.* Šema relacije *Nast_Pred* iz primera 2.3 nije u trećoj normalnoj formi, jer skup funkcionalnih zavisnosti sadrži tranzitivnu zavisnost $OZN \rightarrow NAP$. Šema relacije *Nastavnik*$(\{OZN, PRN\}, \{OZN \rightarrow PRN\})$ i *Predmet*$(\{OZP, NAP\}, \{OZP \rightarrow NAP, NAP \rightarrow OZP\})$ su u trećoj normalnoj formi. Lako se proverava da pojave nad ovim šemama relacija ne poseduju anomalije ažuriranja. □

Pogrešno bi bilo zaključiti da pojave nad svakom šemom relacije, koja se nalazi u trećoj, ali ne i u nekoj višoj normalnoj formi, ne poseduju anomalije ažuriranja. Ovo tvrđenje ilustruje sledeći primer.

r(Nast_Pred_Stud) =

BRI	*OZP*	*OZN*
159	p_1	n_1
159	p_2	n_2
13	p_1	n_3
119	p_3	n_4
159	p_3	n_4
119	p_1	n_1
159	p_4	n_5
37	p_1	n_1
213	p_3	n_4

Slika 2.2.

***Primer** 2.6.* Posmatra se šema relacije *Nast_Pred_Stud*$(\{OZN, OZP, BRI\}, \{OZN \rightarrow OZP, BRI+OZP \rightarrow OZN\})$. Skup ključeva ove šeme relacije je $\{BRI+OZN, BRI+OZP\}$. Semantika šeme relacije je: student sluša predmet kod jednog nastavnika, a nastavnik predaje jedan predmet. Pošto su sva obeležja šeme relacije primarna, zaključuje se da je šema relacije u trećoj normalnoj formi.

Na slici 2.2 je prikazana pojava nad šemom relacije *Nast_Pred_Stud*, dobijena projektovanjem relacije *r*(*Fakultet*) na skup obeležja {*OZN*, *OZP*, *BRI*}. Ažuriranje ove relacije dovodi do sledećih problema:

- ne može se upisati podatak da je nekom novom nastavniku poveren neki predmet, ako taj predmet kod njega ne sluša bar jedan student,
- ako, na primer, nastavnik n_2 prestane da izvodi predmet p_2, brisanjem torke $(159, p_2, n_2)$ bi se izgubila informacija da je student sa $BRI = 159$ slušao predmet p_2 kod nastavnika n_2. □

S tačke gledišta anomalija ažuriranja, šema relacije u trećoj normalnoj formi predstavlja bolje rešenje nego šeme relacija u prvoj ili drugoj normalnoj formi. Objašnjenje leži u činjenici da skup ograničenja šeme relacije u trećoj normalnoj formi ne sadrži netrivijalne tranzitivne, pa ni nepotpune funkcionalne zavisnosti, kod kojih je neprimarno obeležje na desnoj strani. Ako šema relacije (R, C) nije u trećoj normalnoj formi, za $AXY \subseteq R$, njen skup funkcionalnih zavisnosti $\mathcal{F} \subseteq C$ implicira zavisnosti $X \rightarrow AY$ i $Y \rightarrow A$. U pojavi nad šemom relacije (R, C), različitim X vrednostima mogu biti pridružene iste AY vrednosti. To dovodi do anomalija modifikacije. Zbog integriteta entiteta, u pojavu se ne mogu upisati AY vrednosti za nepoznatu X vrednost, a brisanje određene X vrednosti može dovesti do gubljenja AY vrednosti, koja se u pojavi javlja samo jedanput.

Ovo razmatranje ukazuje i na kriterijum za izbor funkcionalnih zavisnosti putem kojih se šema relacije prevodi u treću normalnu formu. Dekompoziciju šeme relacije $(AXY, \{X \rightarrow Y, Y \rightarrow A\})$ treba vršiti na osnovu funkcionalne zavisnosti $Y \rightarrow A$. Dekomponovanjem na osnovu tranzitivne funkcionalne zavisnosti $X \rightarrow A$ se gubi funkcionalna zavisnost $Y \rightarrow A$. Šeme relacija $(XY, \{X \rightarrow Y\})$ i $(AX, \{X \rightarrow A\})$ jesu u trećoj normalnoj formi, s obzirom na u njih ugrađene funkcionalne zavisnosti, ali funkcionalna zavisnost $Y \rightarrow A$ nije ugrađena ni u jednu od ovih šema relacija. Osim toga, dekompozicija na osnovu funkcionalne zavisnosti, na čijoj levoj strani se nalazi ključ šeme relacije, ne dovodi do smanjenja redundanse podataka, pa ni do izbegavanja anomalija ažuriranja.

Anomalije ažuriranja se manifestuju i pri ažuriranju šeme relacije u trećoj normalnoj formi, ako njen skup ograničenja sadrži tranzitivnu funkcionalnu zavisnost na čijoj desnoj strani je primarno obeležje. Problem se rešava prevođenjem šeme relacije u Boyce-Coddovu normalnu formu.

2.1.4. Boyce-Coddova normalna forma

Boyce-Coddova normalna forma je stroža od treće normalne forme, mada je i ona zasnovana samo na funkcionalnim zavisnostima. Veća strogost Boyce-Coddove normalne forme potiče od činjenice da se pri njenom definisanju zabranjuje postojanje svake tranzitivne zavisnosti u skupu ograničenja šeme relacije, pa i one na čijoj desnoj strani je primarno obeležje.

Definicija 2.5. Šema relacije (R, C) je u ***Boyce-Coddovoj normalnoj formi***, ako, za $\mathcal{F} \subseteq C$, važi

2.2. *Metode normalizacije*

Postoje dve osnovne metode normalizacije. To su: metoda dekompozicije i metoda sinteze. Svaka od ovih osnovnih metoda poseduje i niz varijanti. Metoda dekompozicije je zasnovana na postupnom rastavljanju šeme univerzalne relacije ($\mathcal{U}$, $\mathcal{C}$) saglasno tipskim zavisnostima u $\mathcal{C}$, dok se ne dobije skup šema relacija u željenoj normalnoj formi.

Metoda sinteze je zasnovana na sasvim drugačijem postupku. Realizuje se korišćenjem samo funkcionalnih zavisnosti. Polazi se od šeme univerzalne relacije ($\mathcal{U}$, $\mathcal{F}$), a rezultat je skup šema relacija, najmanje u trećoj normalnoj formi. Svaka šema relacije tog skupa, "sintetizuje" se od funkcionalnih zavisnosti iz kanoničnog pokrivanja skupa $\mathcal{F}$. Osnovna ideja metode sinteze je zasnovana na redukciji i eliminaciji svih suvišnih funkcionalnih zavisnosti iz polaznog skupa $\mathcal{F}$.

Definicija 2.8. Neka je ($\mathcal{U}$, $\mathcal{C}$) šema univerzalne relacije, $\mathcal{F} \subseteq \mathcal{C}$ skup funkcionalnih zavisnosti, a $S = \{(R_i, C_i) \mid i = 1, \ldots, n\}$ skup šema relacija i neka su dati sledeći uslovi:

1° $$\mathcal{U} = \bigcup_{i=1}^{n} R_i,$$

2° $$(\forall r \in SAT(\mathcal{U}, \mathcal{C}))(r = \bowtie_{i=1}^{n} (\pi_{R_i}(r)),$$

3° $$\mathcal{F}^+ = (\bigcup_{i=1}^{n} \mathcal{F}|_{R_i})^+ .$$

Ako skup šema relacija S zadovoljava uslove 1° i 2°, naziva se ***dekompozicijom*** šeme ($\mathcal{U}$, $\mathcal{C}$) sa ***spojem bez gubitaka***. Ako zadovoljava uslove 1° i 3°, naziva se takvom ***reprezentacijom*** šeme ($\mathcal{U}$, $\mathcal{C}$), koja ***konzervira*** funkcionalne zavisnosti. Konačno, ako skup šema relacija S zadovoljava uslove 1° i 2° i 3°, naziva se takvom ***dekompozicijom*** šeme ($\mathcal{U}$, $\mathcal{C}$) sa ***spojem bez gubitaka***, koja ***konzervira*** funkcionalne zavisnosti. □

Primer 2.13. Neka je $\mathcal{U} = \{A, B, C\}$, a $\mathcal{C} = \mathcal{F} = \{A \to B, C \to B\}$. Lako se proverava da važi: $\mathcal{F} \models \bowtie(AB, AC)$, $\mathcal{F} \models \bowtie(AB, AC, BC)$ i $\neg(\mathcal{F} \models \bowtie(AB, BC))$. Skup šema relacija: $S_1 = \{(\{A, B\}, (\{A \to B\}), (\{A, C\}, \{\})\}$ je dekompozicija šeme ($\mathcal{U}$, $\mathcal{C}$) sa spojem bez gubitaka, $S_2 = \{(\{A, B\}, \{A \to B\}), (\{B, C\}, \{C \to B\})\}$ je takva reprezentacija šeme ($\mathcal{U}$, $\mathcal{O}$), koja konzervira skup funkcionalnih zavisnosti, a $S_3 = \{(\{A, B\}, \{A \to B\}), (\{B, C\}, \{C \to B\}), (\{A, C\}, \{\})\}$ je takva dekompozicija šeme ($\mathcal{U}$, $\mathcal{C}$) sa spojem bez gubitaka, koja konzervira skup funkcionalnih zavisnosti. □

2.3. *Metoda dekompozicije*

Osnovnu ideju metode dekompozicije dao je Codd u radu [C1]. Metodom dekompozicije su se bavili i mnogi drugi autori, na primer, [CD], [F1], [D4], [De], [U1], [M] i

2.4. Metoda sinteze

Algoritam sinteze je bio predmet intenzivnog istraživanja tokom druge polovine sedamdesetih i prve polovine osamdesetih godina. Kao rezultat tog istraživanja, publikovano je više varijanti tog algoritma u radovima [B], [BB], [KTY], [LTK], [DM] i knjigama [M], [U], [PBG], na primer. Algoritam sinteze, izložen na ovom mestu, pretežno se oslanja na radove [B], [BB] i [KTY], mada sadrži i rezultate originalnih istraživanja autora ove knjige.

Postupak se primenjuje na šemu relacije $(\mathcal{U}, \mathcal{F})$. Rezultat je skup šema relacija $S = \{(R_i, \mathcal{K}_i) | i = 1,..., n\}$, gde je $R_i \subseteq \mathcal{U}$, a $\mathcal{K}_i$ skup određenih (sintetizovanih) ključeva. Pri tome, takođe, važi:

1° Skup šema relacija S sadrži kompletnu informaciju o skupu obeležja $\mathcal{U}$ (***konzervira*** skup obeležja), jer je $\mathcal{U} = \bigcup_{i=1}^{n} R_i$.

2° Skup šema relacija S sadrži kompletnu informaciju o skupu funkcionalnih zavisnosti $\mathcal{F}$, (***konzervira*** skup funkcionalnih zavisnosti), jer je $\mathcal{F} \equiv \{X \to A | (\exists (R_i, \mathcal{K}_i) \in S)(X \in \mathcal{K}_i \land A \in (R_i \setminus X))\}$.

3° Svaka šema relacije $(R_i, \mathcal{K}_i)$ iz S je, najmanje, u ***trećoj*** normalnoj formi.

4° Skup šema relacija S implicira ***zavisnost spoja*** $\rhd\lhd(R_1,..., R_n)$.

5° Ne postoji skup šema relacija T sa manjim brojem šema relacija nego S, a da zadovoljava uslove 1°, 2° i 3°.

Na osnovnu razliku između algoritma dekompozicije i algoritma sinteze ukazuje osobina 2° skupa S. Algoritam sinteze obezbeđuje konzerviranje inicijalnog skupa funkcionalnih zavisnosti. Pri tome, leve strane svih funkcionalnih zavisnosti iz kanoničnog pokrivača $\mathcal{H}$ polaznog skupa funkcionalnih zavisnosti $\mathcal{F}$ ugrađene su u šeme relacija skupa S putem određenih (sintetizovanih) ključeva.

Definicija 2.9. Određeni (ili ***sintetizovani***) ***ključ*** šeme relacije $(R, \mathcal{K})$ je skup obeležja $X \in \mathcal{K}$, takav da važi

$$(X \subseteq R) \land (X \in lhs(\mathcal{H})). \square$$

Algoritam sinteze ne pronalazi sve ključeve šeme relacije, već samo one koji su posledica levih strana funkcionalnih zavisnosti u inicijalnom skupu funkcionalnih zavisnosti $\mathcal{F}$. Kao što je u tački 5.3 knjige "Principi baza podataka" pokazano, pronalaženje svih ključeva šeme relacije je postupak sa eksponencijalnom ocenom kompleksnosti.

Primer 2.20. Neka je $\mathcal{U} = \{A, B, C, D, E\}$, a $\mathcal{F} = \{AB \to C, AB \to D, AB \to E, C \to A, D \to B\}$. Lako se proverava da je $\mathcal{F}$ kanonično pokrivanje. Algoritam sinteze daje skup šema relacija $S = \{(\{A, B, C, D, E\}, \{AB\}), (\{A, C\}, \{C\}), (\{B, D\}, \{D\})\}$. Pri tome je AB jedini određeni (sintetizovani) ključ šeme relacije $(\{A, B, C, D, E\}, \{AB\})$, mada ta šema relacije poseduje sledeći skup ključeva $\mathcal{K} = \{AB, AD, CB, CD\}$. $\square$

Ključevi, koji nisu određeni, nazivaju se ***implicitnim*** ili ***nesintetizovanim*** . Svi ključevi šeme relacije, određeni i implicitni, međusobno su ekvivalentni. Jedan od određenih ključeva se proglašava za ***primarni*** . Primarnim ključem se proglašava onaj ekvivalentni ključ šeme relacije, putem čijih vrednosti se, najčešće, vrši traženje određene torke u relaciji. U dobro projektovanoj šemi baze podataka bi trebalo da se jedino primarni ključevi šema relacija javljaju kao strani ključevi u drugim šemama relacija.

Na slikama 2.12 i 2.13 je prikazana varijanta algoritma sinteze prema [BB], koja generiše skup šema relacija u barem trećoj normalnoj formi sa minimalno mogućim brojem šema relacija i koji konzervira skup funkcionalnih zavisnosti $\mathcal{F}$. Proširenje algoritma sinteze putem dva dodatna koraka, opisana u tačkama 2.4.2 i 2.4.3 ovog poglavlja, obezbeđuju i konzerviranje skupa obeležja $\mathcal{U}$, kao i spojivost bez gubitaka. Prikazani algoritam se primenjuje na šemu univerzalne relacije $(\mathcal{U}, \mathcal{F})$, gde je $\mathcal{F}$ skup takvih funkcionalnih zavisnosti, koje na desnoj strani sadrže jednočlane skupove obeležja.

Algoritam sinteze, prikazan na slikama 2.12 i 2.13, sadrži sedam koraka. Prva dva prevode inicijalno zadati skup funkcionalnih zavisnosti $\mathcal{F}$ u njegov kanonični pokrivač $\mathcal{H}$, tako što se u prvom koraku vrši redukcija levih strana, a u drugom eliminacija trivijalnih i tranzitivnih funkcionalnih zavisnosti.

Treći korak deli $\mathcal{H}$ na disjunktne podskupove funkcionalnih zavisnosti sa istim levim stranama. Tako dobijeni podskupovi bi se mogli upotrebiti za sintetizovanje šema relacija u trećoj normalnoj formi, gde bi svaki podskup generisao po jednu šemu relacije. Međutim, takav skup šema relacija ne bi posedovao osobinu minimalnosti broja šema relacija, jer bi sadržao onoliko šema relacija koliko iznosi $|lhs(\mathcal{H})|$. Drugim rečima, ako u $\mathcal{H}$ postoje funkcionalne zavisnosti $X \rightarrow A$ i $Y \rightarrow B$ takve, da je $X \neq Y$ i $(X)^+_{\mathcal{H}} = (Y)^+_{\mathcal{H}}$, dobile bi se dve šeme relacije, mada je prirodno očekivati da X i Y predstavljaju ekvivalentne ključeve jedne šeme relacije.

Da bi se dobio minimalan skup šema relacija, u četvrtom koraku se pronalaze i izdvajaju u poseban skup funkcionalnih zavisnosti $\mathcal{J}$, ekvivalentne leve strane funkcionalnih zavisnosti, a njihovi podskupovi funkcionalnih zavisnosti se uniraju. Generisanjem funkcionalnih zavisnosti tipa $X \rightarrow Y$ i $Y \rightarrow X$ i njihovim smeštanjem u $\mathcal{J}$, uvode se nove funkcionalne zavisnosti, koje jesu logička posledica funkcionalnih zavisnosti iz $\mathcal{H}$, ali nisu morale postojati u $\mathcal{H}$. To može dovesti do situacije da neke od postojećih funkcionalnih zavisnosti iz $\mathcal{H}$ postanu tranzitivne. Pronalaženje ovih tranzitivnih zavisnosti se vrši u petom koraku algoritma. Neophodnost primene ovog koraka ilustruje primer 2.21.

U šestom koraku algoritma sinteze, vrši se rekonstrukcija podskupova tako, da se iz njih eliminišu, eventualne tranzitivne zavisnosti, a uključuju se odgovarajuće funkcionalne zavisnosti tipa $X \rightarrow Y$ i $Y \rightarrow X$. Konačno, u sedmom koraku, formira se skup šema relacija S tako, što se skup obeležja funkcionalnih zavisnosti jednog podskupa proglašava za skup obeležja šeme relacije, a skup levih strana funkcionalnih zavisnosti tog podskupa, za skup sintetizovanih ključeva iste šeme relacije.

ALGORITAM SINTEZE

Ulaz: $(\mathcal{U}, \mathcal{F})$, $\mathcal{F} = \{X \to A \mid AX \subseteq \mathcal{U}\}$
Izlaz: $S = \{(R_i, \mathcal{K}_i) \mid i = 1, \ldots, n\}$

POSTUPAK :
(* *1. Redukuj* $\mathcal{F}$ *u* $\mathcal{E}$ *)
POSTAVI $\mathcal{E} \leftarrow \emptyset$
RADI *redukcija* $(\forall X \to A \in \mathcal{F})$
  RADI *eliminacija_obeležja* $(\forall B \in X)$
    AKO JE $A \in (X \setminus \{B\})^+_{\mathcal{F}}$ ***TADA***
      POSTAVI $X \leftarrow X \setminus \{B\}$
    KRAJ AKO
  KRAJ RADI *eliminacija_obeležja*
  POSTAVI $\mathcal{E} \leftarrow \mathcal{E} \cup \{X \to A\}$
KRAJ RADI *redukcija*

(* *2. Nađi neredundantno pokrivanje* $\mathcal{H}$ *za* $\mathcal{E}$ *)
POSTAVI $\mathcal{H} \leftarrow \mathcal{E}$
RADI *eliminacija_fz* $(\forall X \to A \in \mathcal{H})$
  AKO JE $A \in (X)^+_{\mathcal{H} \setminus \{X \to A\}}$ ***TADA***
    POSTAVI $\mathcal{H} \leftarrow \mathcal{H} \setminus \{X \to A\}$
  KRAJ AKO
KRAJ RADI *eliminacija_fz*

(* *3. Particioniraj* $\mathcal{H}$ *u podskupove* $\mathcal{G}(X_i)$, $i = 1, \ldots, m$, $m = |lhs(\mathcal{H})|$ *)
POSTAVI $\mathcal{G}(X) \leftarrow \emptyset$
RADI *formiranje_podskupova* $(\forall Y \to A \in \mathcal{H})$
  POSTAVI $ind \leftarrow 0$
  RADI traženje_podskupa $\forall \mathcal{G}(X_i) \in \mathcal{G}(X)$ ***DOK JE*** $ind = 0$
    AKO JE $Y = X_i$ ***TADA***
      POSTAVI $\mathcal{G}(X_i) \leftarrow \mathcal{G}(X_i) \cup \{Y \to A\}$
      POSTAVI $ind \leftarrow 1$
    KRAJ AKO
  KRAJ RADI traženje_podskupa
  AKO JE $ind = 0$ ***TADA***
    POSTAVI $\mathcal{G}(Y) \leftarrow \{Y \to A\}$
    POSTAVI $\mathcal{G}(X) \leftarrow \mathcal{G}(X) \cup \{\mathcal{G}(Y)\}$
  KRAJ AKO
KRAJ RADI *formiranje_podskupova*

Slika 2.12.

(* *4. Izdvoj ekvivalentne leve strane* *)
POSTAVI $\mathcal{J} \leftarrow \varnothing$
RADI *ekvi_lhs* $(\forall \mathcal{G}(X_i), \mathcal{G}(X_j) \in \mathcal{G}(X) = \{\mathcal{G}(X_1), \ldots, \mathcal{G}(X_m)\})(\mathcal{G}(X_i) \neq \mathcal{G}(X_j))$
  AKO JE $(X_i)_{\mathcal{H}}^+ = (X_j)_{\mathcal{H}}^+$ **TADA**
    POSTAVI $\mathcal{J} \leftarrow \mathcal{J} \cup \{X_i \rightarrow X_j, X_j \rightarrow X_i\}$
    POSTAVI $\mathcal{G}(X_i) \leftarrow \mathcal{G}(X_i) \cup \mathcal{G}(X_j)$
    POSTAVI $\mathcal{G}(X) \leftarrow \mathcal{G}(X) \setminus \{\mathcal{G}(X_j)\}$
    POSTAVI $\mathcal{G}(X_i) \leftarrow \mathcal{G}(X_i) \setminus \{Y \rightarrow A \mid (A \in X_i \cup X_j) \wedge (Y = X_i \vee Y = X_j)\}$
  KRAJ AKO
KRAJ RADI *ekvi_lhs*

(* *5. Pronađi tranzitivne zavisnosti* *)
POSTAVI $\mathcal{G} \leftarrow \bigcup_{i=1}^{m} \mathcal{G}(X_i)$
POSTAVI $\mathcal{L} \leftarrow \varnothing$
RADI *tranzitivna_fz* $(\forall X \rightarrow A \in \mathcal{G})$
  AKO JE $A \in (X)^+_{(\mathcal{G} \cup \mathcal{J}) \setminus \{X \rightarrow A\}}$ **TADA**
    POSTAVI $\mathcal{L} \leftarrow \mathcal{L} \cup \{X \rightarrow A\}$
  KRAJ AKO
KRAJ RADI *tranzitivna_fz*

(* *6. Rekonstruiši podsupove* *)
RADI *rekonstrukcija_podskupova* $(\forall \mathcal{G}(X_i) \in \mathcal{G}(X))$
  POSTAVI $\mathcal{G}(X_i) \leftarrow (\mathcal{G}(X_i) \setminus \mathcal{L}) \cup \{X_j \rightarrow X_k \mid (X_j \rightarrow X_k \in \mathcal{J}) \wedge ((X_i)^+_{\mathcal{H}} = (X_j)^+_{\mathcal{H}})\}$
KRAJ RADI *rekonstrukcija_podskupova*

(* *7. Formiraj šeme relacija* *)
POSTAVI $S \leftarrow \varnothing$
RADI *skup_šema_relacija* $(\forall \mathcal{G}(X_i) \in \mathcal{G}(X))$
  POSTAVI $R_i \leftarrow attr(\mathcal{G}(X_i))$
  POSTAVI $\mathcal{K}_i \leftarrow lhs\,(\mathcal{G}(X_i))$
  POSTAVI $S \leftarrow S \cup \{N_i(R_i, \mathcal{K}_i)\}$
KRAJ RADI *skup_šema_relacija*
KRAJ POSTUPKA

Slika 2.13.

Primer 2.21. Neka je $\mathcal{U} = \{A, B, C, D, E, F\}$, a $\mathcal{F} = \{AB \rightarrow C, AB \rightarrow D, DE \rightarrow A, DE \rightarrow B, AC \rightarrow F, BF \rightarrow E, E \rightarrow C\}$. Skup funkcionalnih zavisnosti $\mathcal{F} = \mathcal{H}$ je kanoničan, što se lako proverava. Na primer, funkcionalna zavisnost $AB \rightarrow C$ nije suvišna, jer važi $(AB)_{\mathcal{F} \setminus \{AB \rightarrow C\}}^+ = \{A, B, D\}$, te se zaključuje $C \notin (AB)_{\mathcal{F} \setminus \{AB \rightarrow C\}}^+$.

U trećem koraku algoritma se dobijaju sledeći podskupovi funkcionalnih zavisnosti: $\mathcal{G}(AB) = \{AB \rightarrow C, AB \rightarrow D\}$, $\mathcal{G}(DE) = \{DE \rightarrow A, DE \rightarrow B\}$, $\mathcal{G}(AC) = \{AC \rightarrow F\}$, $\mathcal{G}(BF) = \{BF \rightarrow E\}$, $\mathcal{G}(E) = \{E \rightarrow C\}$.

Pošto važi $AB \rightarrow DE$, $DE \rightarrow AB \in \mathcal{H}^+$, u četvrtom koraku se dobija $\mathcal{J} = \{AB \rightarrow DE, DE \rightarrow AB\}$, $\mathcal{G}(AB) = \{AB \rightarrow C\}$ i $\mathcal{G}(X) = \{\mathcal{G}(AB), \mathcal{G}(AC), \mathcal{G}(BF), \mathcal{G}(E)\}$.

U petom koraku se vrši traženje neredundantnog pokrivanja skupa funkcionalnih zavisnosti $\mathcal{G} = \{AB \rightarrow C, AC \rightarrow F, BF \rightarrow E, E \rightarrow C\}$, s obzirom na skup funkcionalnih zavisnosti $\mathcal{J} \cup \mathcal{G}$. Pri tome se utvrđuje da je funkcionalna zavisnost $AB \rightarrow C$ tranzitivna, jer važi $C \in (AB)^+_{(\mathcal{J} \cup \mathcal{G}) \setminus \{AB \rightarrow C\}} = \{A, B, C, D, E, F\}$. Zato se postavlja $\mathcal{L} = \{AB \rightarrow C\}$. U skupu $\mathcal{H}$, funkcionalna zavisnost $AB \rightarrow C$ nije suvišna. Neophodna je za izvođenje zaključka da važi $AB \rightarrow E$, što se izvodi putem funkcionalnih zavisnosti $AB \rightarrow C$, $AC \rightarrow F$ i $BF \rightarrow E$. U skupu $\mathcal{J} \cup \mathcal{G}$ funkcionalna zavisnost $AB \rightarrow C$ je postala suvišna, jer $\mathcal{J}$ već sadrži $AB \rightarrow E$.

Na izlasku iz algoritma sinteze se dobija $S = \{(\{A, B, D, E\}, \{AB, DE\}), (\{A, C, F\}, \{AC\}), (\{B, E, F\}, \{BF\}), (\{C, E\}, \{E\})\}$. Skup šema relacija S je u trećoj normalnoj formi. Da nije vršena eliminacija tranzitivnih funkcionalnih zavisnosti, na izlasku iz algoritma bi se dobio skup šema relacija $S' = \{(\{A, B, C, D, E\}, \{AB, DE\}), (\{A, C, F\}, \{AC\}), (\{B, E, F\}, \{BF\}), (\{C, E\}, \{E\})\}$, koji je u prvoj, ali ne i nekoj višoj normalnoj formi.

Do opisanog fenomena pojave tranzitivnih zavisnosti, nakon uvođenja funkcionalnih zavisnosti tipa $X \rightarrow Y$ i $Y \rightarrow X$ u skup $\mathcal{J} \cup \mathcal{H}$, dolazi samo u retkim slučajevima, kada je u šemi relacije neko neprimarno obeležje neophodno za izvođenje zaključka da važi $X \rightarrow Y$, $Y \rightarrow X \in \mathcal{H}^+$. U posmatranom slučaju, reč je o šemi relacije $(\{A, B, C, D, E\}, \{AB, DE\})$ i obeležju C. □

Primer 2.22. Neka je inicijalni skup funkcionalnih zavisnosti iz primera 1.1, prvog poglavlja, zadat kao $\mathcal{F} = \{BRI \rightarrow IME, BRI \rightarrow PRZ, BRI \rightarrow BPI, BRI{+}IME \rightarrow BPI, OZP \rightarrow NAP, NAP \rightarrow OZP, OZN \rightarrow PRN, OZN \rightarrow OZP, BRI{+}OZP \rightarrow OCE, BRI{+}OZP \rightarrow OZN, BRI{+}OZP{+}OZN \rightarrow PRN\}$.

Nakon prolaska kroz korak redukcije algoritma sinteze, dobija se ili $\mathcal{E}_1 = \{BRI \rightarrow IME, BRI \rightarrow PRZ, BRI \rightarrow BPI, OZP \rightarrow NAP, NAP \rightarrow OZP, OZN \rightarrow PRN, OZN \rightarrow OZP, BRI{+}OZP \rightarrow OCE, BRI{+}OZP \rightarrow OZN\}$ ili $\mathcal{E}_2 = \{BRI \rightarrow IME, BRI \rightarrow PRZ, BRI \rightarrow BPI, OZP \rightarrow NAP, NAP \rightarrow OZP, OZN \rightarrow PRN, OZN \rightarrow OZP, BRI{+}OZP \rightarrow OCE, BRI{+}OZP \rightarrow OZN, BRI{+}OZP \rightarrow PRN\}$, u zavisnosti od toga da li je, pri redukciji funkcionalne zavisnosti $BRI{+}OZP{+}OZN \rightarrow PRN$, redom, prvo eliminisano obeležje OZP (zbog funkcionalne zavisnosti $OZN \rightarrow OZP$), ili obeležje OZN (zbog funkcionalne zavisnosti $BRI{+}OZP \rightarrow OZN$).

Neka je $\mathcal{E} = \mathcal{E}_2$. Neredundantno pokrivanje $\mathcal{H}$ skupa $\mathcal{E}$ je $\{BRI \rightarrow IME, BRI \rightarrow PRZ, BRI \rightarrow BPI, OZP \rightarrow NAP, NAP \rightarrow OZP, OZN \rightarrow PRN, OZN \rightarrow OZP, BRI{+}OZP \rightarrow OCE, BRI{+}OZP \rightarrow OZN\}$.

Nakon particioniranja, dobijaju se sledeći podskupovi $\mathcal{G}(X_i)$: $\mathcal{G}(BRI) = \{BRI \rightarrow IME, BRI \rightarrow PRZ, BRI \rightarrow BPI\}$, $\mathcal{G}(OZP) = \{OZP \rightarrow NAP\}$, $\mathcal{G}(NAP) = \{NAP \rightarrow OZP\}$, $\mathcal{G}(OZN) = \{OZN \rightarrow PRN, OZN \rightarrow OZP\}$, $\mathcal{G}(BRI{+}OZP) = \{BRI{+}OZP \rightarrow OCE, BRI{+}OZP \rightarrow OZN\}$.

Pošto OZP i NAP predstavljaju jedine ekvivalentne leve strane funkcionalnih zavisnosti iz $\mathcal{H}$, na izlasku iz koraka izdvajanja ekvivalentnih strana algoritma sinteze se do-

$S = \{(\{BRI, IME, PRZ, BPI\}, \{BRI\}), (\{OZP, NAP\}, \{OZP, NAP\}), (\{OZN, PRN, OZP\}, \{OZN\}), (\{BRI, OZP, OZN, OCE\}, \{BRI+OZP\})\}$.

Skup šema relacija S je samo u trećoj normalnoj formi, jer šema relacije ($\{BRI, OZP, OZN, OCE\}$, $\{BRI+OZP\}$) nije u Boyce-Coddovoj normalnoj formi s obzirom na skup funkcionalnih zavisnosti $\mathcal{H}$, koji sadrži funkcionalnu zavisnost $OZN \rightarrow OZP$. Ta funkcionalna zavisnost je, putem ključa, ugrađena u šemu relacije ($\{OZN, PRN, OZP\}$, $\{OZN\}$). Takođe treba zapaziti, da šema relacije ($\{BRI, OZP, OZN, OCE\}$, $\{BRI+OZP\}$), pored određenog ključa $BRI+OZP$ poseduje još jedan nesintetizovan ključ. To je $BRI+OZN$.

BRI	*OZP*	*OZN*	*OCE*
159	p_1	n_1	9
159	p_2	n_2	8
13	p_1	n_3	6
119	p_3	n_4	7
159	p_3	n_4	10
119	p_1	n_1	9
159	p_4	n_5	10
37	p_1	n_1	10
213	p_3	n_4	10

Slika 2.14.

Na slici 2.14 je prikazana relacija nad šemom relacije ($\{BRI, OZP, OZN, OCE\}$, $\{BRI+OZP, BRI+OZN\}$). Relacija je dobijena projekcijom relacije *Fakultet* sa slike 1.1 na skup obeležja $\{BRI, OZP, OZN, OCE\}$.

Šema relacije ($\{BRI, OZP, OZN, OCE\}$, $\{BRI+OZP, BRI+OZN\}$) sadrži dve semantičke komponente. To su:

- student s (sa $BRI = b$) je položio predmet p sa ocenom o, kod nastavnika n i
- student s sluša predmet p samo kod nastavnika n.

Relacija na slici 2.14 poseduje određene anomalije ažuriranja. Na primer, ne može se upisati da je student položio ispit iz nekog predmeta, ako se ne zna kod kojeg je nastavnika taj predmet slušao. Do još većih problema dolazi ako jedan nastavnik prestane predavati jedan predmet, a počne predavati neki drugi predmet. Ako se počnu unositi u relaciju podaci o studentima, koji sad slušaju taj novi predmet kod posmatranog nastavnika, dovodi se u kontradikciju sadržaj relacije sa tvrđenjem da važi funkcionalna zavisnost $OZN \rightarrow OZP$.

Ako se prethodno izbrišu sve torke, vezane za nastavnika i predmet, koji je predavao ranije, gube se bitni podaci. □

Stav 2.6. Neka je $\mathcal{F}$ skup funkcionalnih zavisnosti, a $X \to Y$ jedna funkcionalna zavisnost u $\mathcal{F}$. Ako važi $W \subseteq V_{\mathcal{F}}^{+}$ i $W \not\subseteq (V)_{\mathcal{F} \setminus \{X \to Y\}}^{+}$, tada je $X \subseteq V_{\mathcal{F}}^{+}$.

Dokaz. Pretpostavlja se, suprotno tvrđenju stava 2.6, da važi $X \not\subseteq V_{\mathcal{F}}^{+}$. Tada može biti ili $W \subseteq X_{\mathcal{F}}^{+}$ ili $W \not\subseteq X_{\mathcal{F}}^{+}$. Ako je $(X \not\subseteq V_{\mathcal{F}}^{+}) \wedge (W \subseteq X_{\mathcal{F}}^{+})$, tada, suprotno pretpostavci stava 2.6, važi $(W \not\subseteq V_{\mathcal{F}}^{+}) \vee (W \subseteq V_{\mathcal{F} \setminus \{X \to Y\}}^{+})$. Ako je $(X \not\subseteq V_{\mathcal{F}}^{+}) \wedge (W \not\subseteq X_{\mathcal{F}}^{+})$, ponovo se zaključuje da, suprotno pretpostavci stava 2.6, važi $(W \not\subseteq V_{\mathcal{F}}^{+}) \vee (W \subseteq V_{\mathcal{F} \setminus \{X \to Y\}}^{+})$, čime je stav dokazan. □

Stav 2.6 definiše uslov kada je funkcionalna zavisnost $X \to Y$ neophodna za izvođenje, putem Armstrongovih aksioma, funkcionalne zavisnosti $V \to W$ iz nekog skupa funkcionalnih zavisnosti $\mathcal{F}$. Sam pojam izvođenja funkcionalne zavisnosti $V \to W$ iz skupa $\mathcal{F}$ je ekvivalentan tvrđenju da je $V \to W$ logička posledica skupa funkcionalnih zavisnosti $\mathcal{F}$. Drugim rečima, ako važi $V \to W \in \mathcal{F}^{+}$ i $V \to W \notin (\mathcal{F} \setminus \{X \to Y\})^{+}$, tada je i $V \to X$ u $\mathcal{F}^{+}$.

Stav 2.7. Skup funkcionalnih zavisnosti $(\mathcal{G} \cup \mathcal{J}) \setminus \mathcal{L}$, koji se dobija nakon šestog koraka algoritma sinteze, je levo redukovan i neredundantan.

Dokaz. Pošto važi $\mathcal{G} \subseteq \mathcal{H}$, $\mathcal{G}$ je levo redukovan skup funkcionalnih zavisnosti, koji ne sadrži trivijalne funkcionalne zavisnosti. Dalje, pošto $\mathcal{L}$ sadrži sve one funkcionalne zavisnosti iz $\mathcal{G}$, koje su tranzitivne s obzirom na $\mathcal{G} \cup \mathcal{J}$, $\mathcal{G} \setminus \mathcal{L}$ je redukovan i neredundantan skup funkcionalnih zavisnosti.

Neka je $X_i \to X_j \in \mathcal{J}$. S obzirom na način formiranja skupa $\mathcal{J}$, važi X_i, $X_j \in lhs(\mathcal{H})$. Neka je X_j takav skup obeležja, da važi $(\exists A \in X_j)(X_i \to A \in \mathcal{H})$. Skup $\mathcal{J}$ mora sadržati takvu funkcionalnu zavisnost $X_i \to X_j$, a X_j mora sadržati takvo A, inače ne važi $X_j \subseteq (X_i)_{\mathcal{H}}^{+}$. Pošto je $X_i \to A \in \mathcal{H}$, važi $(\forall B \in X_i)((X_i \setminus \{B\}) \to A \notin \mathcal{F}^{+})$. Znači skup $\mathcal{J}$ je levo redukovan.

S obzirom na način formiranja skupa $\mathcal{J}$ i činjenicu da, za $X \to Y$, $Y \to Z \in \mathcal{F}^{+}$, funkcionalna zavisnost $X \to Z$ nije tranzitivna, ako važi $Y \to X \in \mathcal{F}^{+}$, zaključuje se da $\mathcal{J}$ ne sadrži tranzitivne funkcionalne zavisnosti, što okončava dokaz. □

Stav 2.8. Neka su $\mathcal{H}_1$ i $\mathcal{H}_2$ dva kanonična skupa funkcionalnih zavisnosti takva, da je $\mathcal{H}_1^{+} = \mathcal{H}_2^{+}$. Ako je funkcionalna zavisnost $Y \to B$ u $\mathcal{H}_2$, tada postoji funkcionalna zavisnost $X \to A$ u $\mathcal{H}_1$ i važi $X \to Y$, $Y \to X \in \mathcal{H}_1^{+}$.

Dokaz. Ako je $\mathcal{H}_1 = \mathcal{H}_2$, tada tvrđenje stava trivijalno sledi. Neka su $\mathcal{H}_1$ i $\mathcal{H}_2$ dva ekvivalentna i različita kanonična skupa funkcionalnih zavisnosti. Tada za svaku funkcionalnu zavisnost iz $\mathcal{H}_1$ postoji izvođenje na osnovu $\mathcal{H}_2$ i obrnuto. Neka se za izvođenje $X \to A$ iz $\mathcal{H}_1$ koriste funkcionalne zavisnosti $Y_1 \to B_1, \ldots, Y_n \to B_n \in \mathcal{H}_2$. Bar za jednu od tih funkcionalnih zavisnosti $Y_i \to B_i$, postoji takvo izvođenje u $\mathcal{H}_1$, za koje se koristi $X \to A$. Pretpostavlja se da, suprotno ovom tvrđenju, važi $Y_1 \to B_1, \ldots, Y_n \to B_n \in (\mathcal{H}_1 \setminus \{X \to A\})^{+}$. Tada je $X \to A \in (\mathcal{H}_1 \setminus \{X \to A\})^{+}$, što je u kontradikciji sa tvrđenjem da je $\mathcal{H}_1$ kanoničan skup.

Znači, postoji funkcionalna zavisnost $Y \to B \in \{Y_1 \to B_1, \ldots, Y_n \to B_n\}$, za čije izvođenje je potrebna funkcionalna zavisnost $X \to A$. Na osnovu stava 2.6 sledi da je $Y \to X$ u $\mathcal{H}_1^+$. Pošto je za izvođenje $X \to A$, na osnovu $\mathcal{H}_2$, potrebna funkcionalna zavisnost $Y \to B$, na osnovu stava 2.6 sledi da je $X \to Y$ u $\mathcal{H}_2^+$, čime je dokaz okončan. □

Primer 2.23. Neka je $\mathcal{H}_1 = \{A \to C, B \to C, C \to B\}$, a $\mathcal{H}_2 = \{A \to B, B \to C, C \to B\}$. Očigledno važi $\mathcal{H}_1^+ = \mathcal{H}_2^+$. Za izvođenje funkcionalne zavisnosti $A \to B \in \mathcal{H}_2$, na osnovu $\mathcal{H}_1$, koristi se funkcionalna zavisnost $A \to C$ i važi $A \to A \in \mathcal{H}_1^+$. □

Primer 2.24. Neka je $\mathcal{H}_1 = \{C \to D, D \to C, CE \to F\}$, a $\mathcal{H}_2 = \{C \to D, D \to C, DE \to F\}$. Očigledno važi $\mathcal{H}_1^+ = \mathcal{H}_2^+$. Za izvođenje funkcionalne zavisnosti $DE \to F$, na osnovu $\mathcal{H}_1$, koristi se funkcionalna zavisnost $CE \to F$ i važi $DE \to CE \in \mathcal{H}_1^+$ i $CE \to DE \in \mathcal{H}_1^+$. □

Lema 2.1. Algoritam sinteze konzervira inicijalni skup funkcionalnih zavisnosti $\mathcal{F}$.

Dokaz. Do eliminacije funkcionalnih zavisnosti iz skupa $\mathcal{F}$ ili skupa $\mathcal{H}$, dolazi u sledećim koracima algoritma sinteze: 1. (redukcija), 2. (neredundantno pokrivanje), 4. (ekvivalentne leve strane) i 6. (rekonstrukcija grupa), dok se u koraku 4 (ekvivalentne leve strane), pored redukovanog i neredundantnog pokrivanja skupa funkcionalnih zavisnosti $\mathcal{H}$, formira skup funkcionalnih zavisnosti $\mathcal{J}$. Skup funkcionalnih zavisnosti $\mathcal{J}$ je logička posledica kanoničkog skupa funkcionalnih zavisnosti $\mathcal{H}$, što znači da se formiranjem skupa $\mathcal{J}$ svakako ne menja zatvaranje inicijalnog skupa funkcionalnih zavisnosti.

Da se u koraku 1 algoritma sinteze ne menja zatvaranje inicijalnog skupa funkcionalnih zavisnosti $\mathcal{F}$ sledi iz činjenice da, nakon izvršene redukcije, važi $(\forall X \to A \in \mathcal{F})(\exists (X \setminus Z) \to A \in \mathcal{E})$ te, pošto je $\varnothing \subseteq Z$, na osnovu pravila izvođenja $\mathfrak{I}_2$, sledi $\mathcal{F}^+ = \mathcal{E}^+$.

U drugom koraku algoritma se iz $\mathcal{E}$ eliminišu trivijalne i tranzitivne funkcionalne zavisnosti. Neka je $X \to A$ funkcionalna zavisnost u $\mathcal{E}$, koja nije u $\mathcal{H}$. Ako je $X = \{A\}$, na osnovu pravila izvođenja $\mathfrak{I}_1$ sledi da je $X \to A$ u $\mathcal{H}^+$. Ako je $X \to A$ netrivijalna funkcionalna zavisnost, na osnovu $A \in X_{\mathcal{E} \setminus \{X \to A\}}^+$, se zaključuje da u $\mathcal{H}$ postoji funkcionalna zavisnost $Y \to A$, takva da je $X \to Y$ u $\mathcal{H}^+$, te je i $X \to A$ u $\mathcal{H}^+$.

Eliminacijom funkcionalnih zavisnosti tipa $X_i \to A$ iz podskupa $\mathcal{G}(X_i)$ za $A \in X_j$ i $X_i \to X_j \in \mathcal{J}$, u koraku 4, ne narušava se uslov $\mathcal{F}^+ = (\mathcal{G} \cup \mathcal{J})^+$, jer, pod opisanim uslovima, $(\mathcal{G} \cup \mathcal{J})^+ = ((\mathcal{G} \cup \mathcal{J}) \setminus \{X_i \to A\})^+$.

U koraku 5 se otkrivaju sve tranzitivne funkcionalne zavisnosti iz $\mathcal{G}$ s obzirom na $(\mathcal{G} \cup \mathcal{J})$, i postavljaju u $\mathcal{L}$. Pošto su funkcionalne zavisnosti iz $\mathcal{L}$ tranzitivne, zaključuje se da, nakon njihove eliminacije u koraku 6, važi $\mathcal{F}^+ = ((\mathcal{G} \cup \mathcal{J}) \setminus \mathcal{L})^+$. □

Funkcionalne zavisnosti iz $\mathcal{F}$ su ugrađene u skup šema relacija S, dobijen algoritmom sinteze, putem ključeva. Saglasno tome, na osnovu leme 2.1 sledi, da važi $\mathcal{F} \equiv \{X \to A \mid (\exists (R_i, \mathcal{K}_i) \in S)(X \in \mathcal{K}_i \wedge A \in (R_i \setminus X))\}$.

Lema 2.2. Algoritam sinteze generiše skup šema relacija S u trećoj normalnoj formi, s obzirom na $\mathcal{F}$.

Dokaz. Pretpostavlja se da je neprimarno obeležje A iz R tranzitivno zavisno od ključa Y šeme relacije $(R, \mathcal{K})$ iz nekog skupa S. Znači, postoji $Z \subseteq R$, takvo, da važi $Z \rightarrow A$, $Y \rightarrow Z \in \mathcal{F}^+$, $Z \rightarrow Y \notin \mathcal{F}^+$ i $A \rightarrow Z \notin \mathcal{F}^+$.

Treba naglasiti da Y ne mora predstavljati sintetizovani ključ šeme relacije $(R, \mathcal{K})$, te će prvo biti pokazano da A tranzitivno zavisi i od svakog sintetizovanog ključa $X \in \mathcal{K}$. Pošto je X ključ, važi $X \rightarrow Z \in \mathcal{F}^+$ i $Z \rightarrow X \notin \mathcal{F}^+$, inače bi Z sadržalo ključ šeme relacije $(R, \mathcal{K})$, što bi bilo u kontradikciji sa tvrđenjem $Z \rightarrow Y \notin \mathcal{F}^+$. Saglasno tome, $X \rightarrow A$ je tranzitivna funkcionalna zavisnost.

Neka je $(\mathcal{G} \cup \mathcal{J}) \backslash \mathcal{L}$ pokrivanje skupa funkcionalnih zavisnosti $\mathcal{F}$, generisano putem algoritma sinteze. Skup funkcionalnih zavisnosti $(\mathcal{G} \cup \mathcal{J}) \backslash \mathcal{L}$ je, saglasno stavu 2.7, levo redukovan i neredundantan. Treba pokazati da funkcionalna zavisnost $X \rightarrow A$ nije u $(\mathcal{G} \cup \mathcal{J}) \backslash \mathcal{L}$. Ovo tvrđenje je posledica činjenice da su funkcionalne zavisnosti $X \rightarrow Z$ i $Z \rightarrow A$ u $(((\mathcal{G} \cup \mathcal{J}) \backslash \mathcal{L}) \backslash \{X \rightarrow A\})^+$, što će značiti da je $X \rightarrow A$ tranzitivna u $(\mathcal{G} \cup \mathcal{J}) \backslash \mathcal{L}$. Stvarno, pošto A nije u Z, za svako $B \in Z$, $X \rightarrow B$ je u $(((\mathcal{G} \cup \mathcal{J}) \backslash \mathcal{L}) \backslash \{X \rightarrow A\})^+$. Ako je za izvođenje funkcionalne zavisnosti $Z \rightarrow A$ potrebna funkcionalna zavisnost $X \rightarrow A$, tada je $Z \rightarrow X \in ((\mathcal{G} \cup \mathcal{J}) \backslash \mathcal{L})^+$. Međutim, to je u kontradikciji sa zaključkom da važi $Z \rightarrow X \notin \mathcal{F}^+$. Znači, i $Z \rightarrow A$ je u $((\mathcal{G} \cup \mathcal{J}) \backslash \mathcal{L}) \backslash \{X \rightarrow A\}$.

Pošto je funkcionalna zavisnost $X \rightarrow A$ tranzitivna, ona nije u $(\mathcal{G} \cup \mathcal{J}) \backslash \mathcal{L}$, te se zaključuje da ili skup obeležja šeme relacije $(R, \mathcal{K})$ ne sadrži A ili šema relacije $(R, \mathcal{K})$ nije dobijena primenom algoritma sinteze. □

Lema 2.3. Neka je $\mathcal{F}$ skup funkcionalnih zavisnosti. Algoritam sinteze generiše isti broj šema relacija za svako levo redukovano i neredundantno pokrivanje skupa $\mathcal{F}$.

Dokaz. Neka su $\mathcal{H}_1$ i $\mathcal{H}_2$ dva redukovana i neredundantna pokrivanja skupa funkcionalnih zavisnosti $\mathcal{F}$. Na osnovu stava 2.8, ako je $X \rightarrow A$ u $\mathcal{H}_1$, tada je $Y \rightarrow B$ u $\mathcal{H}_2$ i važi $X \rightarrow Y$, $Y \rightarrow X \in \mathcal{H}_1^+$. Saglasno tome, za svaki neprazan podskup funkcionalnih zavisnosti $\mathcal{G}(X)$, dobijen primenom koraka 6 (rekonstrukcija grupa) algoritma sinteze na $\mathcal{H}_1$, postoji tačno jedan podskup $\mathcal{G}(Y)$, sa istim skupom ekvivalentnih levih strana, dobijen na osnovu $\mathcal{H}_2$. Pošto svaki takav podskup generiše jednu šemu relacije, algoritam sinteze proizvodi isti broj šema relacija za $\mathcal{H}_1$ i $\mathcal{H}_2$. □

Lema 2.3 ukazuje da, sa tačke gledišta broja šema relacija, sva neredundantna pokrivanja jednog skupa funkcionalnih zavisnosti predstavljaju jednako dobru osnovu za generisanje skupa šema relacija S. Ovo tvrđenje je, na prvi pogled, u kontradikciji sa intuitivnom pretpostavkom da neredundantno pokrivanje sa manjim brojem funkcionalnih zavisnosti treba da bude bolja osnova za generisanje skupa šema relacija. Objašnjenje leži u činjenici da sva pokrivanja generišu isti broj klasa ekvivalentnih levih strana.

Primer 2.25. Neka je $\mathcal{H}_1 = \{A \rightarrow B, B \rightarrow A, A \rightarrow C, C \rightarrow A\}$, a $\mathcal{H}_2 = \{A \rightarrow B, B \rightarrow C, C \rightarrow A\}$. Važi $\mathcal{H}_1^+ = \mathcal{H}_2^+$, $|\mathcal{H}_1| = 4$ i $|\mathcal{H}_2| = 3$. Za oba skupa funkcionalnih zavisnosti, algoritam sinteze generiše isti skup šema relacija $S = \{(\{A, B, C\}, \{A, B, C\})\}$. □

Posledica 2.1. Neka je S skup šema relacija, dobijen primenom algoritma sinteze na skup funkcionalnih zavisnosti $\mathcal{F}$. Ne postoji skup šema relacija, koji je u trećoj normalnoj formi, u koji je ugrađen skup $\mathcal{F}$, a da ima manji broj šema relacija.

Dokaz. Ako je skup šema relacija S u trećoj normalnoj formi i konzervira polazni skup funkcionalnih zavisnosti $\mathcal{F}$, tada je, putem ključeva, u S ugrađeno jedno pokrivanje skupa $\mathcal{F}$.

Neka je skup šema relacija S'' formiran tako da je od svake funkcionalne zavisnosti iz kanoničnog pokrivača $\mathcal{H}$ skupa funkcionalnih zavisnosti $\mathcal{F}$, napravljena jedna šema relacije. Neka je skup šema relacija S' formiran tako, da je od svakog podskupa skupa $\mathcal{G}(X)$, dobijenog u koraku 3 algoritma sinteze, napravljena jedna šema relacije. Konačno, neka je skup šema relacija S dobijen na izlasku iz algoritma sinteze. Tada važi $|S| \leq |S'| \leq |S''|$. Saglasno lemi 2.3, ovaj odnos važi za svako kanoničko pokrivanje $\mathcal{H}$, skupa funkcionalnih zavisnosti $\mathcal{F}$. □

Postoje dve motivacije za formiranje jedne šeme relacije od onih, koje imaju ekvivalentne ključeve. Jedna je da se za svaku klasu entiteta sačini jedinstven model u obliku šeme relacije. Druga leži u činjenici da realizacija baze podataka sa više relacija, koje sadrže podatke o entitetima iste klase, dovodi do intenzivne primene operatora prirodnog spoja u korišćenju baze podataka. Operacije prirodnog spajanja su veliki potrošači vremena rada računarskih resursa.

Primer 2.26. Neka je $\mathcal{F} = \{A\to B, C\to B, B\to D, A\to D, D\to B, AC\to B, A\to E\}$. Tada je $\mathcal{H} = \{A\to B, C\to B, B\to D, D\to B, A\to E\}$, a $\mathcal{G}(X) = \{\{A\to B, A\to E\}, \{C\to B\}, \{B\to D\}, \{D\to B\}\}$. Pošto je $\mathcal{J} = \{B\to D, D\to B\}$, dobija se $|S''| = 5$, $|S'| = 4$, i $|S| = 3$. □

Teorema 2.2. Na osnovu skupa funkcionalnih zavisnosti $\mathcal{F}$ nad skupom obeležja $\mathcal{U}$, algoritam sinteze sa slika 2.12 i 2.13 generiše skup šema relacija S sa sledećim osobinama:

1° S je, barem, u trećoj normalnoj formi,

2° u S je ugrađeno jedno redukovano, neredundantno pokrivanje skupa $\mathcal{F}$,

3° broj šema relacija u S je minimalan.

Dokaz. Tvrđenje 1° sledi na osnovu leme 2.2. Tvrđenje 2° sledi na osnovu leme 2.1. Tvrđenje 3° sledi na osnovu posledice 2.1. □

Izračunavanje zatvaranja skupa obeležja predstavlja vremenski najkompleksniju aktivnost algoritma sinteze sa slika 2.12 i 2.13. U [ML] je prikazan algoritam za izračunavanje zatvaranja skupa obeležja, čija ocena kompleksnosti je $O(|\mathcal{U}||\mathcal{F}|min\{|\mathcal{U}|, |\mathcal{F}|\})$. U literaturi, na primer [BB], [PBG], [M], opisani su algoritmi za izračunavanje zatvaranja skupa obeležja, čija ocena kompleksnosti je linearna, s obzirom na dužinu ulaza.

Teorema 2.3. Ocena kompleksnosti algoritma sinteze sa slika 2.12 i 2.13 je polinomijalna.

Dokaz. U koraku redukcije, izračunava se zatvaranje skupa obeležja, za svako obeležje leve strane svake funkcionalne zavisnosti iz $\mathcal{F}$. Nakon izračunavanja zatvaranja, proverava se da li određeno obeležje pripada zatvaranju. Ocena kompleksnosti ovog koraka algoritma sa slika 2.12 i 2.13 je $O(|\mathcal{U}||\mathcal{F}|(|\mathcal{U}||\mathcal{F}|min\{|\mathcal{U}|, |\mathcal{F}|\} + |\mathcal{U}|))$, odnosno $O((|\mathcal{U}||\mathcal{F}|)^2 min\{|\mathcal{U}|, |\mathcal{F}|\})$.

Pri traženju neredundantnog pokrivanja, izračunava se zatvaranje skupa obeležja leve strane svake funkcionalne zavisnosti iz $\mathcal{E}$. Nakon izračunavanja svakog zatvaranja, proverava se da li mu pripada desna strana funkcionalne zavisnosti. Pošto je $|\mathcal{E}| \leq |\mathcal{F}|$, ocena kompleksnosti ovog koraka je $O(|\mathcal{F}|(|\mathcal{U}||\mathcal{F}|min\{|\mathcal{U}|, |\mathcal{F}|\} + |\mathcal{U}|))$, odnosno $O(|\mathcal{U}||\mathcal{F}|^2 min\{|\mathcal{U}|, |\mathcal{F}|\})$.

Pri particioniranju skupa funkcionalnih zavisnosti $\mathcal{H}$, iterativni blok "traženje_podskupa" se izvršava za svaku funkcionalnu zavisnost iz $\mathcal{H}$. Ocena kompleksnosti ovog iterativnog bloka je $O(|\mathcal{F}||\mathcal{U}|)$, jer je $|\mathcal{H}| \leq |\mathcal{E}| \leq |\mathcal{F}|$, a ocena aktivnosti upoređivanja levih strana je $O(|\mathcal{U}|)$. Saglasno tome, ocena kompleksnosti koraka particioniranja skupa funkcionalnih zavisnosti $\mathcal{H}$ je $O(|\mathcal{F}|^2|\mathcal{U}|)$.

U okviru svakog prolaza kroz iterativni blok "ekvi_lhs", vrši se upoređivanje zatvaranja skupa obeležja jednog $\mathcal{G}(X_i)$ sa zatvaranjem skupa obeležja svakog od ostalih $\mathcal{G}(X_j)$. Ocena kompleksnosti tog prolaza je $O(|\mathcal{U}||\mathcal{F}|^2)$, jer je $|\mathcal{G}(X)| \leq |\mathcal{F}|$. Pri tome je realno pretpostaviti da se, za svako $\mathcal{G}(X_i)$, zatvaranje skupa obeležja izračunava samo jedanput. Ocena kompleksnosti ovog koraka je $O(|\mathcal{U}||\mathcal{F}|^2 min\{|\mathcal{U}|, |\mathcal{F}|\})$.

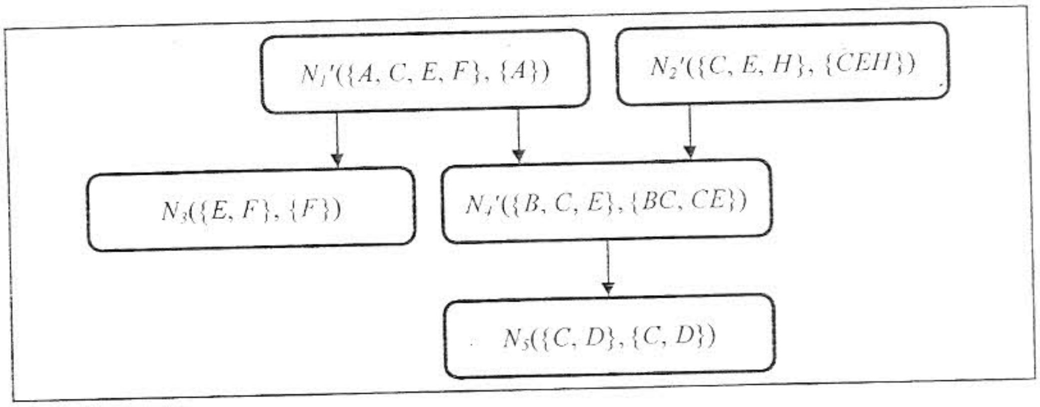
U petom koraku se izvršava izračunavanje zatvaranja skupa obeležja leve strane svake funkcionalne zavisnosti iz $\mathcal{G}$. Pri tome važi $|\mathcal{G}| \leq |\mathcal{H}| \leq |\mathcal{F}|$. Ta zatvaranja se izračunavaju s obzirom na skup funkcionalnih zavisnosti $\mathcal{G} \cup \mathcal{J}$. Veličina skupa funkcionalnih zavisnosti $\mathcal{J}$ zavisi od načina formiranja tog skupa. Da bi se odredila veličina skupa, prvo treba zapaziti da važi $|lhs(\mathcal{G})| \leq |lhs(\mathcal{H})|$. Neka je $\{Y_1, Y_2,..., Y_k\}$ jedan, tekući, skup ekvivalentnih levih strana, dobijen pri formiranju $\mathcal{J}$, i neka indeks $i = 1, 2,..., k$ ukazuje na redosled uključivanja Y_i u skup $\{Y_1, Y_2,..., Y_k\}$. Neka je postupak formiranja $\mathcal{J}$ takav da kadgod se nova leva strana Z iz $lhs(\mathcal{H})$, takva da je $Z^+ = Y_1^+$, uključuje u $lhs(\mathcal{J})$, u skup $\mathcal{J}$ se upisuju sledeće dve nove funkcionalne zavisnosti $Y_k \rightarrow Z$ i $Z \rightarrow Y_k$. Tada je $O(\mathcal{J}) = O(\mathcal{H})$. Ocena kompleksnosti petog koraka je $O(|\mathcal{U}||\mathcal{F}|^2 min\{|\mathcal{U}|, |\mathcal{F}|\})$.

Ocene kompleksnosti šestog i sedmog koraka algoritma su iste i iznose $O(|\mathcal{F}|)$.

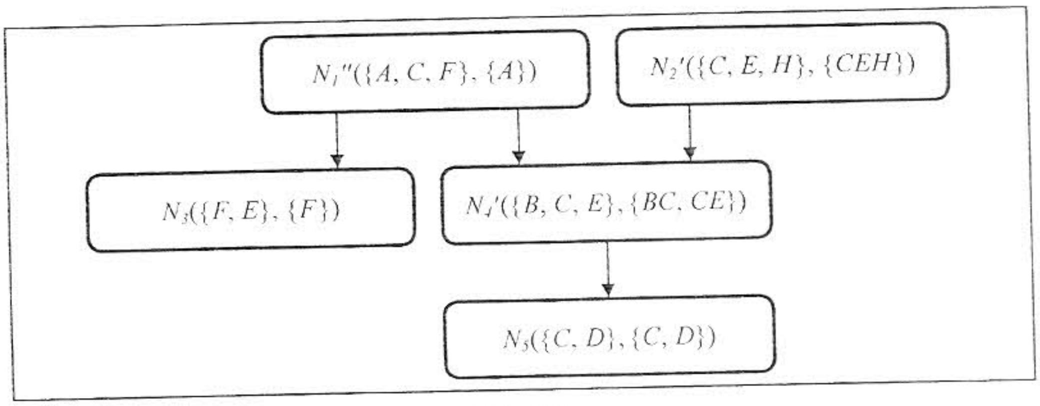
Ocena kompleksnosti celog algoritma je polinomijalna i iznosi $O((|\mathcal{U}||\mathcal{F}|)^2 min\{|\mathcal{U}|, |\mathcal{F}|\})$. □

2.4.1. Nefunkcionalni odnos

Funkcionalne, višeznačne i zavisnosti spoja ne predstavljaju jedina tipski invarijantna ograničenja između vrednosti obeležja realnog sistema. Često se u skupu $\mathcal{U}$ mogu identifikovati takvi podskupovi obeležja $X = \{A_1, A_2,..., A_m\}$, da pojave nad tim skupovima nose informaciju o komponenti stanja realnog sistema. Po pravilu, elementi skupa X predstavljaju identifikaciona obeležja različitih klasa entiteta. Povezivanje ovih obeležja nije proizvoljno. Njihovim povezivanjem se iskazuje činjenica da su i u realnom sistemu entiteti posmatranih klasa u nekakvoj vezi, ali i da ta veza nije niti funkcionalna, niti višeznačna niti da odgovara zavisnosti spoja. Pošto je reč o modeliranju delova realnog sistema, svaki



Slika 3.8.

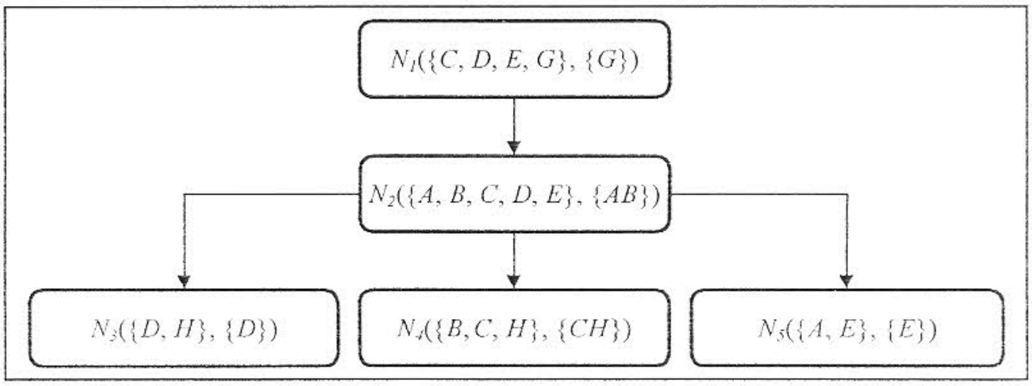


Slika 3.9.

Međutim, postoje skupovi šema relacija, za koje primena posmatranog algoritma transformacije ne dovodi do očekivanih efekata. Jedan od problema izaziva pojava nesintetizovanih ključeva, kao propagiranih ključeva. Drugi problem se javlja zbog činjenice da opisani postupak prostiranja primarnog ključa može dovesti do gubitka informacija o ograničenjima realnog sistema. Treći je problem da ne moraju svi ekvivalentni ključevi jedne šeme relacije, predstavljati kandidate za primarni ključ. Ovi problemi se razmatraju i rešavaju u narednim tačkama poglavlja.

3.3. Nesintetizovani ključevi

Algoritam sinteze generiše samo one ključeve šema relacija, koji predstavljaju leve strane u neredundantnom pokrivanju skupa funkcionalnih zavisnosti. Ti ključevi se nazivaju sintetizovanim. Međutim, šema relacije $N(R, {}^{\prime}K)$, gde je ${}^{\prime}K$ skup sintetizovanih ključeva, može posedovati i nesintetizovane ključeve. Skup obeležja $X \subset R$ predstavlja nesintetizovani ključ šeme relacije $N(R, {}'K)$, ako predstavlja ključ i važi $X \notin {}'K$.



Slika 3.10.

Primer 3.7. Graf zatvaranja na slici 3.10 ilustruje tri problema, do kojih može dovesti postojanje nesintetizovanih ključeva. Te probleme predstavljaju sledeće činjenice:

- da CDE predstavlja nesintetizovani ključ šeme relacije N_2, jer važi $\{CD \rightarrow B, E \rightarrow A, AB \rightarrow CDE\} \subset \mathcal{G}^+$,
- da se, tek nakon zaključka da CDE predstavlja nesintetizovani ključ šeme relacije N_2, dolazi do uverenja da putem SQL upita

SELECT A, B, G
FROM N_1, N_2
WHERE $N_1.C = N_2.C$ **AND** $N_1.D = N_2.D$ **AND** $N_1.E = N_2.E$

nije definisan spoj sa gubicima,

- da skup šema relacija S ne zadovoljava uslov (3.2), jer važi ${}'K_2 = \{AB\}$, $R_2 \subseteq (R_1)^+_{\mathcal{G}}$ i $K_p(R_2, R_1) = \emptyset$,
- da primena algoritma transformacije ne može dati očekivani rezultat.

Poslednje tvrđenje traži opširnije obrazloženje. Prvo, pošto je ${}'\widehat{K}_2 = \{A, B\}$, važi $R_1 \cap {}'\widehat{K}_2 = \emptyset$. Ako se ta činjenica zanemari i postavi $R_1' = R_1 \setminus {}'\widehat{K}_2 \cup V$, za $K_p(R_2) = AB$ i $V = AB$, dobija se $R_1' = \{A, B, C, D, E, G\}$. Pošto je $\{G \rightarrow CDE, G \rightarrow AB\} \subset (\mathcal{G}')^+$, ako se, pri primeni algoritma sinteze, u neredundantnom pokrivanju skupa funkcionalnih zavisnosti $\mathcal{G}'$ dobije $G \rightarrow CDE$, a ne $G \rightarrow AB$, primena algoritma transformacije ne bi dovela do željenog efekta. Da je CDE bilo identifikovano kao jedan od ključeva šeme relacije N_2, algoritam transformacije bi dao očekivani rezultat. □

Očigledno, problemi vezani za graf zatvaranja u primeru 3.7 su posledica činjenice da algoritam sinteze ne otkriva sve ključeve šeme relacije. S druge strane, primer 3.7 sugeriše da je za uspešnu primenu algoritma transformacije, neophodno generisati sve ključeve svake šeme relacije iz S (osim za one, koje predstavljaju minimalne čvorove grafa zatvaranja). Algoritam za generisanje svih dodatnih ključeva šeme relacije, prikazan je u tački 5.3 knjige "Principi baza podataka" [ML].

3.4. Kandidati za primarni ključ

Uslov (3.2) inicira potrebu da se skup šema relacija S sa nekorektnim prostiranjem ključa transformiše u skup šema relacija S', u cilju postizanja boljih eksploatacionih karakteristika. Prirodno se nameće i zahtev da transformisani skup šema relacija nosi istu količinu informacije o realnom sistemu, kao polazni S. Drugim rečima, skup S' treba, pored uslova (3.2) i treće normalne forme, da zadovolji i sledeće uslove:

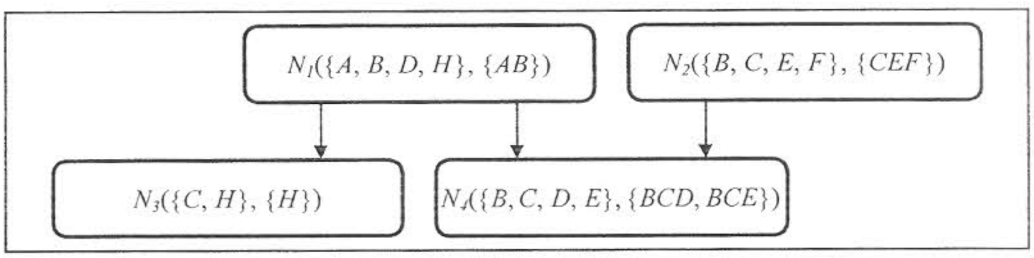
(3.6) $\mathcal{G}^+ = (\mathcal{G}')^+$ i

(3.7) $attr(\mathcal{G}) = attr(\mathcal{G}')$.

U krajnje pojednostavljenom posmatranju, algoritam transformacije predstavlja postupak zamene onih ekvivalentnih ključeva jedne šeme relacije, koji su propagirani u druge šeme relacija, jednim - primarnim ključem. Nažalost, taj cilj nije uvek moguće postići za svaki ekvivalentni ključ šeme relacije tako, da budu zadovoljeni uslovi (3.2) i (3.6), što sledeći primer i ilustruje.

Primer 3.8. Na slici 3.11 je prikazan graf zatvaranja skupa šema relacija S, u koji je ugrađen skup funkcionalnih zavisnosti $\mathcal{G} = \{AB \to D, AB \to H, H \to C, CEF \to B, BCD \to E, BCE \to D\}$.

Skup S ne zadovoljava uslov (3.2), jer je iz šeme relacije N_4 ka šemi relacije N_1 propagiran ključ BCD, a ka šemi relacije N_2 ključ BCE. Zato se na S primenjuje osnovni algoritam transformacije. Ako se odredi $K_p(N_4) = BCE$, tada će biti transformisana samo šema relacije N_1. Ako se, pri prostiranju ključa BCE u šemu relacije N_1, postavi $V = BE$ i $W = B$, dobija se skup šema relacija S', čiji graf zatvaranja je prikazan na slici 3.12. U skup šema relacija S' je ugrađen skup funkcionalnih zavisnosti $\mathcal{G}' = \{AB \to E, AB \to H, H \to C, CEF \to B, BCD \to E, BCE \to D\}$. Funkcionalna zavisnost $AB \to D$ iz $\mathcal{G}$ ne pripada skupu $\mathcal{G}'$, dok funkcionalna zavisnost $AB \to E$ iz $\mathcal{G}'$ ne pripada skupu $\mathcal{G}$. Međutim, lako se proverava da važi $AB \to D \in (\mathcal{G}')^+$ i $AB \to E \in (\mathcal{G})^+$, tako da skupovi šema relacija, evidentno, zadovoljavaju uslov (3.6).



Slika 3.11.