3.2a. Reliacinė algebra

Jau minėjome, kad lentelės reliacinė schema pagrindinai nusakoma lentelės vardu ir jos stulpelių vardais. Norėdami pabrėžti pirminio rakto svarbą, jį taip pat priskyrėme reliacinei schemai. Kai pirminio rakto vaidmuo nėra esminis, nurodydami lentelės schemą, jį praleisime. Prisilaikydami reliacinėje teorijoje naudojamais terminais, lentelės stulpelius taip pat vadinsime **atributais**. Tarkime, $A_1, A_2,..., A_n$ yra lentelės L atributų (stulpelių) aibė, čia A_i – atributas, $i = 1,...,n$. Tuomet $L(A_1, A_2,..., A_n)$ žymi lentelės schemą. Lentelės atributų aibę pažymėję $R = \{A_1, A_2,..., A_n\}$, jos schemą galime užrašyti taip $L(R)$.

Lentele l, kurios schema $L(A_1, A_2,..., A_n)$, yra vadinama sutvarkytų reikšmių rinkinių (eilučių) aibė $l = \{e_1, e_2,..., e_m\}$. Konkreti lentelė l su schema $L(A_1, A_2,..., A_n)$ dar žymima taip $l(A_1, A_2,..., A_n)$. Kiekviena eilutė e_j yra sutvarkytas reikšmių rinkinys $e_j = <a_1, a_2,..., a_n> \in l$, čia $a_i \in dom(A_i)$, $dom(A_i)$ žymi atributo A_i galimų reikšmių aibę, t.y. jo domeną, $i = 1,...,n$; $j = 1,...,m$. $e(A_i)$ žymi atributo A_i reikšmę eilutėje e. Atributų aibės R poaibio A, $A \subseteq R$, reikšmių rinkinys eilutėje e žymimas panašiai: $e(A)$.

Pasinaudodami pateiktaisiais apibrėžimais, galime sakyti, kad lentelės $l(R) = \{e_1, e_2,..., e_m\}$ atributų aibės R poaibis $V \subseteq R$ yra viršraktis, jei $\forall\ i, j = 1,...,m$ ir $i \neq j$ yra teisinga nelygybė $e_i(V) \neq e_j(V)$.

Lentelę $l(A_1, A_2,..., A_n)$ galima apibrėžti ir kaip Dekarto sandaugos poaibį

$$l(A_1, A_2,..., A_n) \subseteq (dom(A_1) \times dom(A_2) \times ... \times dom(A_n)).$$

Suprantama, lentelės schema (sandara) $L(R)$ ir konkretus jos turinys $l(R)$ nėra tas pat. Lentelė $l(R)$ – tai lentelės, kurios schema yra $L(R)$, konkretus turinys (užpildymas konkrečiais duomenimis) konkrečiu momentu. Dėl šių sąvokų dualumo mes dažnai naudosime tą patį žymenį tiek lentelei, tiek jos schemai žymėti, laikydami kad kontekstas apsprendžia, kas konkrečiai turima omenyje.

Sudarinėdami užklausas, mes jau manipuliavome lentelėmis. Dabar formaliai apibrėšime pagrindines lentelių operacijas, kurios sudaro reliacinę algebrą. Neformaliai, **reliacine algebra** vadinama formali operacijų kalba (sistema), kuria iš vienos ar kelių lentelių (santykių), nekeičiant jų turinio, galima gauti kitas lenteles. Kadangi operacijos rezultatas yra lentelė, tai su juo taip pat galima atlikti reliacinės algebros operacijas. Taip konstruojami **reliacinės algebros reiškiniai**.

Reliacinės algebros operacijos dažniausiai yra skirstomos į dvi grupes. Pirmąją grupę sudaro matematinės aibių teorijos operacijos: sąjunga ($\cup$), sankirta ($\cap$), skirtumas ($-$) ir Dekarto sandauga ($\times$). Šias operacijas galima taikyti lentelėms, nes jos yra eilučių aibės. Antrąją reliacinės algebros operacijų grupę sudaro specifinės duomenų bazėms operacijos: išrinkimas (σ), projekcija (π), jungimas ($\bowtie$) ir kitos. Aibių operacijos mums yra žinomos iš aibių teorijos pagrindų, be to, neformaliai mes jas jau aptarėme ankstesniame šios knygos skyriuje. Detaliau aptarsime išrinkimą, projekciją ir jungimą, kurios yra vienos iš pagrindinių reliacinėje algebroje.

3.2.1. Pagrindinės operacijos su lentelėmis

Išrinkimas yra unarinė operacija su lentele. Pritaikius šią operaciją lentelei, gaunama kita lentelė, kurią sudaro pradinės lentelės eilučių poaibis su konkrečia konkretaus atributo reikšme. Tarkime, $l = l(R)$ yra lentelė, A – atributas iš aibės R ir $a \in dom(A)$. Tuomet $\sigma_{A=a}(l)$ žymi išrinkimo operaciją „iš l išrinkti visas eilutes, kuriose atributo A reikšmė yra lygi a“. Operacijos rezultatas yra lentelė $l'(R)$

$$\sigma_{A=a}(l) = l'(R) = \{e \in l : e(A) = a\}.$$

Išrinkimo operacija yra komutatyvi jų kompozicijos atžvilgiu

$$\sigma_{A=a}(\sigma_{B=b}(l)) = \sigma_{B=b}(\sigma_{A=a}(l)).$$

Ši savybė išplaukia iš išrinkimo operacijos apibrėžimo

$$\sigma_{A=a}(\sigma_{B=b}(l)) = \sigma_{A=a}(\{e\in l : e(B) = b\}) = \{e' \in \{e\in l : e(B) = b\} : e'(A) = a\} =$$
$$= \{e\in l : e(A) = a \text{ ir } e(B) = b\} = \{e' \in \{e\in l : e(A) = a\} : e'(B) = b\} = \sigma_{B=b}(\sigma_{A=a}(l)).$$

Kadangi išrinkimo tvarka nesvarbi, tai kompoziciją $\sigma_{A=a} \circ \sigma_{B=b}$ galima išreikšti taip $\sigma_{A=a,B=b}$. Išrinkimo operacija yra distributyvi aibių binarinių operacijų ($\cup$, $\cap$, $-$) atžvilgiu. Šiai savybei įrodyti taip pat pakanka išrinkimo operacijos apibrėžimo.

Projekcijos operacija taip pat yra unarinė. Priešingai išrinkimo operacijai, projekcijos operacija yra išrenkamas ne eilučių, bet stulpelių poaibis. Lentelės $l = l(R)$ projekcija aibėje A, $\pi_A(l)$, $A \subseteq R$, yra lentelė $l'(A)$, kuri gaunama iš $l(R)$ išbraukiant stulpelius $R–A$,

$$\pi_A(l) = l'(A) = \{e(A) : e\in l\}.$$

Jei dvi projekcijos atliekamos paeiliui ir antrojoje išrenkamųjų stulpelių aibė yra pirmosios stulpelių aibės poaibis, tai pirmoji operacija yra nereikšminga, ją galima praleisti. Tiksliau, jei $A \subseteq B \subseteq R$, tai lentelei $l(R)$ galioja lygybė

$$\pi_A(\pi_B(l)) = \pi_A(l).$$

Projekcija komutuoja su išrinkimu, kai išrinkimo atributas yra vienas iš projekcijos atributų. Jei $A \in B$, $B \subseteq R$ ir $l(R)$ yra lentelė, tai

$$\pi_B(\sigma_{A=a}(l)) = \pi_B(\{e\in l : e(A) = a\}) = \{e'(B) : e' \in \{e\in l : e(A) = a\}\} =$$
$$= \{e(B) : e\in l \text{ ir } e(A) = a\} = \sigma_{A=a}(\{e(B) : e\in l\})= \sigma_{A=a}(\pi_B(l)).$$

Kita labai svarbi operacija su lentelėmis yra jų **jungimas**. Tai binarinė operacija, kuria kombinuojamos dvi lentelės. Dviejų lentelių $l_1(R_1)$ ir $l_2(R_2)$ **junginiu** $l_1 \bowtie l_2$ yra vadinama lentelė $l_3(R_3)$, $R_3 = R_1 \cup R_2$, tokia, kad

$$l_1 \bowtie l_2 = l_3(R_3) =$$
$$= \{e : \exists\, e_1,e_2\ (e_1\in l_1,\ e_2\in l_2,\ e(R_1) = e_1,\ e(R_2) = e_2,\ e_1(R_1\cap R_2) = e_2(R_1\cap R_2))\}.$$

Jungimo operacijos apibrėžime nereikalaujama, kad $R_1\cap R_2$ būtų netuščia aibė. Jeigu $R_1\cap R_2 = \varnothing$, tai jungimas yra ekvivalentiškas Dekarto sandaugai, $l_1 \bowtie l_2 = l_1 \times l_2$. Taigi, Dekarto sandauga yra ypatingasis jungimo operacijos atvejis.

Yra ir daugiau reliacinės algebros operacijų. Tačiau čia pateiktosios yra laikomos vienomis pagrindinių. Pavyzdžiui, yra įrodyta, kad operacijų aibė $\{\sigma, \pi, \cup, -, \times\}$ yra **pilnoji**. Tai reiškia, kad bet kurią kitą reliacinę operaciją galima išreikšti šios aibės operacijų seka. Pavyzdžiui, sankirtos operaciją ($\cap$) galima išreikšti per sąjungą ($\cup$) ir skirtumą ($-$)

$$l_1 \cap l_2 = (l_1\cup l_2) - ((l_1 - l_2) \cup(l_2 - l_1)).$$

Lentelių $l_1(R_1)$ ir $l_2(R_2)$ junginį $l_1 \bowtie l_2$ galima išreikšti per σ, π ir $\times$

$$l_1 \bowtie l_2 = \pi_{R_1\cup R_2}(\sigma_{l_1.A_1=l_2.A_1,\ldots,l_1.A_N=l_2.A_N}(l_1 \times l_2)),$$

čia $\{A_1, A_2,..., A_N\} = R_1 \cap R_2$. Jungimo operacijos savybes aptarsime detaliau.

3.2.2. Jungimo savybės

Jungimo operacija galima išreikšti išrinkimo operaciją. Tarkime, lentelėje $l(R)$ reikia atlikti išrinkimo operaciją $\sigma_{A=a}(l)$. Apibrėžiame naują lentelę $l'(A)$ su vienintele eilute e, kurioje $e(A) = a$. Tuomet yra teisinga lygybė $\sigma_{A=a}(l) = l \bowtie l'$, nes

$$l \bowtie l' = \{e : \exists\, e_1,e_2\, (e_1\in l,\ e_2\in l',\ e(R) = e_1,\ e(A) = e_2)\} =$$
$$= \{e : \exists\, e_1\ (e_1\in l,\ e(R) = e_1,\ e(A) = a)\} = \sigma_{A=a}(l).$$

Kad lentelėje $l(R)$ sumodeliuoti bendresnę išrinkimo operaciją $\sigma_{A=a_1,A=a_2}(l)$, papildomoje lentelėje l' reikia įrašyti dvi eilutes e_1,e_2, kurioms $e_1(A) = a_1$, $e_2(A) = a_2$.

Jei papildomoje dviejų stulpelių lentelėje $l'(AB)$ yra viena eilutė e, kuriai $e(A) = a$, $e(B) = b$, tai $l \bowtie l' = \sigma_{A=a,B=b}(l)$. Taip galima modeliuoti išrinkimą pagal konkrečias dviejų stulpelių reikšmes.

Dėl simetrijos jungimo operacijos apibrėžime, gauname, kad ši operacija yra asociatyvi

$(l_1 \bowtie l_2) \bowtie l_3 = l_1 \bowtie (l_2 \bowtie l_3)$.

Todėl tokiuose reiškiniuose skliaustelius galima praleisti.

Sujunkime, pavyzdžiui, tris lenteles

l_1

A	B
a_1	b_1
a_1	b_2
a_2	b_1

l_2

B	C
b_1	c_2
b_2	c_1

l_3

A	C
a_1	c_2
a_2	c_2

Jų junginys yra lentelė

$l_1 \bowtie l_2 \bowtie l_3$

A	B	C
a_1	b_1	c_2
a_2	b_1	c_2

Jungiant šias lenteles, lentelės l_1 eilutė $<a_1, b_2>$ ir lentelės l_2 eilutė $<b_2, c_1>$ liko nesujungtos. Lentelės l_1, l_2,..., l_m vadinamos **visiškai sujungiamos**, jei kiekviena kiekvienos lentelės eilutė yra lentelių junginio konkrečios eilutės dalis. Jei lentelę l_3 papildytume eilute $<a_1, c_1>$, tai pateiktosios lentelės l_1, l_2, l_3 taptų visiškai sujungiamomis. Tuomet jų junginys atrodytų taip

$l_1 \bowtie l_2 \bowtie l_3$

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

Jungimo ir projekcijos operacijos nėra viena kitai visiškai atvirkščios. Tarkime, $l_1(R_1)$ ir $l_2(R_2)$ – lentelės. Nesunku įsitikinti, kad

$\pi_{R_1}(l_1 \bowtie l_2) \subseteq l_1$.

Čia poaibio ženklas tampa lygybe, kai kiekvienai lentelės l_1 eilutei e_1 lentelėje l_2 yra eilutė e_2 tokia, kad $e_1(R_1 \cap R_2) = e_2(R_1 \cap R_2)$. Poaibis gali tapti lygybe ir tuo atveju, kai l_1 ir l_2 nėra visiškai sujungiamos. Imkime dvi pateiktąsias lenteles l_2 ir l_3, kiekvienoje iš kurių yra po dvi eilutes. l_2 ir l_3 nėra visiškai sujungiamos, nes lentelės l_2 eilutės $<b_2, c_1>$ negalima sujungti nei su viena lentelės l_3 eilute. Šių dviejų lentelių junginį $l_2 \bowtie l_3$, sudarys tos pačios dvi eilutės, kaip ir jau pateiktąjį junginį $l_1 \bowtie l_2 \bowtie l_3$, gautą prieš papildant l_3 trečiąja eilute. Nesunku įsitikinti, kad $l_3 = \pi_{AB}(l_2 \bowtie l_3)$.

Tarkime $l_3(R_1 \cup R_2)$ yra lentelė, o $l_1(R_1)$ ir $l_2(R_2)$ – jos projekcijos, $l_1 = \pi_{R_1}(l_3)$, $l_2 = \pi_{R_2}(l_3)$. Jeigu e yra l_3 eilutė, tai $e(R_1) \in l_1$ ir $e(R_2) \in l_2$. Vadinasi, $l_3 \subseteq l_1 \bowtie l_2$. Jeigu galioja lygybė $l_3 = l_1 \bowtie l_2$, tai toks lentelės l_3 skaidymas į dvi lenteles l_1 ir l_2 yra vadinamas lentelės (l_3) **dekompozicija be praradimo**. Taip skaidant lentelę l_3 į l_1 ir l_2 jokie duomenys nėra prarandami, nes pradinę lentelę l_3 galima gauti jungiant l_1 ir l_2.

Tarkime, turime lentelę

l_4

A	B	C
a_1	b	c_1
a_2	b	c_2

Sudarome dvi šios lentelės projekcijas

$\pi_{AB}(l_4)$

A	B
a_1	b
a_2	b

$\pi_{BC}(l_4)$

B	C
b	c_1
b	c_2

Akivaizdu, kad šių projekcijų junginį $\pi_{AB}(l_4) \bowtie \pi_{BC}(l_4)$ sudaro net keturios eilutės. Tai reiškia, kad $l_4 \subset \pi_{AB}(l_4) \bowtie \pi_{BC}(l_4)$. Todėl lentelės l_4 suskaidymas į dvi jos projekcijas: $\pi_{AB}(l_4)$ ir $\pi_{BC}(l_4)$ nėra dekompozicija be praradimo. Nesunku įsitikinti, jeigu lentelės l_4 stulpelyje B esančios reikšmės būtų skirtingos, tai atitinkamų projekcijų junginys sutaptų su pradine lentele ir gautume dekompoziciją be praradimo.

3.2.3. Palyginimo apibendrinimas operacijose

Reliacinės algebros operacijose dviems domeno (atributo reikšmių aibės) reikšmėms palyginti iki šiol naudojome tik vieną lygybės (=) operaciją. Reliacinėje teorijoje laikomasi prielaidos, kad bet kurias konkretaus domeno reikšmes galima įvertinti ar jos tarpusavyje lygios (=), ar nelygios ($\neq$). Pakankamai dažnai domenų reikšmės būna sutvarkytos taip, kad bet kurioms dviems nelygioms domeno reikšmėms galima nustatyti, kuri iš jų yra didesnė, o kuri – mažesnė. Daugelyje praktikoje naudojamų domenų (skaičiai, simboliniai duomenys, datos, laikai ir pan.) naudojamos palyginimo operacijos: $=, \neq, <, \leq, \geq, >$. Šią palyginimo operacijų aibę žymėsime Θ. Bendruoju atveju gali būti lyginamos skirtingų domenų reikšmės, pvz. sveikąjį skaičių galime lyginti su realiuoju. Jei $\theta \in \Theta$ - palyginimo operacija, o A ir B – atributai, tai A ir B – vadinsime θ–palyginamais, jei θ yra apibrėžta aibėje $dom(A) \times dom(B)$.

Apibrėžtosiose išrinkimo ir jungimo operacijose naudojome lygybės operaciją. Jų apibrėžimą galima išplėsti ir kitoms palyginimo operacijoms. Tarkime, $l = l(R)$ yra lentelė, A, $B \in R$, $a \in dom(B)$, $\theta \in \Theta$, A ir B yra θ–palyginami. Tuomet $\sigma_{A\theta a}(l)$ žymi **θ – išrinkimo operaciją**

$$\sigma_{A\theta a}(l) = \{e \in l : e(A)\ \theta\ a\}.$$

Vietoje atributo lyginimo su konkrečia reikšme, galima lyginti du atributus

$$\sigma_{A\theta B}(l) = \{e \in l : e(A)\ \theta\ e(B)\}.$$

Patogumo dėlei, išrinkimo operaciją galima dar labiau **apibendrinti**, leidžiant išrinkimo sąlygoje naudoti logines operacijas bei skliaustelius, pvz. $\sigma_{A \leq B \wedge (A=a \vee B\neq b)}(l)$.

Jungiant lenteles taip pat galima naudoti bet kurią palyginimo operaciją $\theta \in \Theta$. Dviejų lentelių $l_1(R_1)$ ir $l_2(R_2)$ **θ – junginiu** $l_1 \bowtie_\theta l_2$ yra vadinama lentelė, sudaryta iš stulpelių $R_3 = R_1 \cup R_2$, ir kurios eilutės apibrėžiamos taip

$$l_1 \bowtie_\theta l_2 = \{e : \exists\, e_1, e_2\ (e_1 \in l_1,\ e_2 \in l_2,\ e(R_1) = e_1,\ e(R_2) = e_2,\ e_1(R_1 \cap R_2)\ \theta\ e_2(R_1 \cap R_2))\}.$$

Praktikoje, tiek θ – išrinkimas, tiek ir apibendrintasis išrinkimas yra labai plačiai naudojami, θ – jungimas sutinkamas žymiai rečiau. Jungimo operaciją galima dar labiau apibendrinti, tačiau praktikoje to beveik neprireikia.

Svarbu tai, kad visos anksčiau apibrėžtos išrinkimo ir jungimo operacijų savybės būdingos ir jų pateiktiesiems apibendrinimas.

3.2.4. Operacijos su lentelėmis SQL kalboje

Apibrėžtosios operacijos mums nėra visiškai naujos. Sudarinėjant užklausas SQL kalba mes jau susidūrėme su panašiomis operacijomis. Jau minėjome, kad iš pradžių buvo sukurta reliacinė teorija, vėliau buvo sudaryta SQL kalba ir dar vėliau pasirodė pirmosioms reliacinių duomenų bazių valdymo sistemos. Visas pagrindines reliacines operacijas DB sistemose realizuoja SQL sakinys `SELECT`.

Lentelės $l(R)$ išrinkimo operacija $\sigma_{A=a}(l)$ išreiškiama tokia užklausa

`SELECT` * `FROM` l `WHERE` $A = a$

Nesunku išreikšti ir apibendrintuosius išrinkimus, pvz. išrinkimą $\sigma_{A \le B \wedge (A=a \vee B \ne b)}(l)$ galima užrašyti tokiu SQL sakiniu

`SELECT` * `FROM` l `WHERE` A <= B `AND` (A = a `OR` B <> b)

Lentelės $l(R)$ projekciją $\pi_{\mathrm{A}}(l)$ galima išreikšti sakiniu

`SELECT DISTINCT` A `FROM` L

Kadangi reliacinėje teorijoje lentelėje vienodų eilučių nėra, tai projekcijai išreikšti prireikė raktažodžio `DISTINCT`.

Dviejų lentelių $l_1(R_1)$ ir $l_2(R_2)$ junginį $l_1 \bowtie l_2$ galima gauti SQL užklausa, kurią struktūriškai galima užrašyti taip

`SELECT` $R_1 \cup R_2$ `FROM` l_1, l_2 `WHERE` $l_1.A_1 = l_2.A_1$ `AND ... AND` $l_1.A_N = l_2.A_N$

čia $(A_1, A_2,..., A_N) = R_1 \cap R_2$. θ–junginio $l_1 \bowtie_\theta l_2$ rezultatas išreiškiamas labai panašiai – tereikia visus lygybės ženklus SQL sakinyje pakeisti palyginimo operacijos θ SQL ženklu.

Trys aibių operacijos: sąjunga ($\cup$), sankirta ($\cap$), skirtumas ($-$) turi tiesioginius atitikmenis SQL kalboje: `UNION`, `INTERSECT` ir `EXCEPT`. Dekarto sandaugą ($\times$) galima laikyti ypatinguoju jungimo operacijos atveju, kai abiejų lentelių (operacijos operandų) visų stulpelių pavadinimai yra laikomi skirtingais. Dviejų lentelių $l_1(R_1)$ ir $l_2(R_2)$ Dekarto sandaugą SQL kalba galima išreikšti taip

`SELECT` R_1, R_2 `FROM` l_1, l_2

Dviejų lentelių Dekarto sandaugą SQL kalboje užrašėme kaip ypatingąjį jungimo atvejį. Galima daryti ir atvirkščiai. Iš pradžių galima apibrėžti Dekarto sandaugą SQL kalba. Tada dviejų lentelių junginys SQL kalboje išreiškiamas per išrinkimo, projekcijos ir Dekarto sandaugą.

3.7′. Funkcinių priklausomybių ir atributų uždariniai

Algoritmas atributų aibės S uždariniui S^+ FP F atžvilgiu rasti.

```
S+ := S ;
repeat
  T := S+ ;
  for each X → Y ∈ F
    if X ⊆ S+ then S+ := S+ ∪ Y ;
  endfor
until T <> S+.
```

Kad patikrinti, ar dvi FP aibės F ir G yra ekvivalenčios, pakanka patikrinti, ar kiekviena FP $X \to Y \in F$ yra išvedama aibėje G, t.y. $X \to Y \in G^+$ ir atvirkščiai, ar kiekviena FP $U \to V \in G$ yra išvedama aibėje F, t.y. $U \to V \in F^+$.

FP uždarinyje yra galimos perteklinės FP, kurios išvedamos iš kitų uždarinio FP. Tai gali labai apsunkinti FP savybių tyrimą. Todėl įvedama minimalaus (standartinio) denginio sąvoka.

FP aibė F vadinama **minimaliąja**, jeigu ji tenkina reikalavimus:

a) kiekvienai FP $X \to Y \in F$, dešinioji pusė Y yra sudaryta tik iš vieno atributo;
b) aibėje F nėra FP, kurią pašalinę, gautume aibę, ekvivalentišką duotajai aibei F;
c) aibėje F nėra FP $X \to A$, kurią pakeitus FP $Y \to A$, $Y \subset X$, gautume FP aibę, ekvivalentišką aibei F.

FP aibės F **minimaliu denginiu** vadinama minimalioji FP aibė $F_{\min}$, kuri yra ekvivalenti aibei F. FP aibei F gali egzistuoti kelios $F_{\min}$. Tačiau visada galima surasti bent vieną jų.

Algoritmas FP aibės F minimaliajam denginiui $F_{\min}$ rasti.

```
Fmin := F;
for each X → A1 A2 ... An ∈ Fmin
   Fmin := Fmin − (X → A1 A2 ... An);
   for i := 1 to n
      Fmin := Fmin ∪ X → Ai;
   endfor
endfor
for each X → A ∈ Fmin
   T := X+ aibės (Fmin − (X → A)) atžvilgiu;
   if A ∈ T then Fmin := Fmin − (X → A) endif;
endfor
for each X → A ∈ Fmin
   for each B ∈ X
      T := (X − B)+ aibės ((Fmin − (X → A)) ∪ ((X − B) → A)) atžvilgiu;
      if A ∈ T then
         Fmin := Fmin − (X → A);
         Fmin := Fmin ∪ ((X − B) → A);
      endif
   endfor
endfor.
```

Atliekant šio algoritmo pirmąjį ciklą `for` visos FP yra suskaidomos taip, kad dešiniosiose FP pusėse būtų tik po vieną atributą. Antrajame cikle `for` iš pirmajame etape sudarytos aibės $F_{\min}$ yra pašalinamos visos perteklinės FP, kurių dešiniosios pusės gali būti išvestos iš likusių FP nepanaudojant šalinamosios. Tuomet analizuojamos visos likusios FP pašalinant perteklinius determinantų atributus. Determinanto atributas yra perteklinis, jei FP dešinėje esantį atributą galima išvesti net ir tuomet, kai perteklinis atributas pašalinamas iš determinanto.

Turėdami algoritmą atributų uždariniui rasti, galima nesudėtingai rasti lentelės raktą.

Algoritmas lentelės $L(R)$ raktui K FP aibės F atžvilgiu rasti.

```
K := R ;
for each A ∈ K
  T := (K − A)+ aibės F atžvilgiu;
  if T = R then K := K − A endif;
endfor.
```

Šio algoritmo pradžioje yra tariama, kad raktą sudaro visi lentelės atributai. Tuomet iš rakto pašalinami visi atributai, be kurių vis tiek galima išvesti visus lentelės atributus. Šiuo algoritmu surandamas tik vienas raktas. Jei lentelė turi kelis raktus, visiems jiems efektyviai surasti reikalingas šiek tiek sudėtingesnis algoritmas.

3.8′. Antroji norminė forma

Gali atrodyti, kad naujosiose lentelėse galiojančios FP skiriasi nuo galiojusiųjų pradinėje lentelėje. Tačiau taip nėra. Suskaidę lentelę nepraradome jokių savybių, nes funkcinė priklausomybė $AB \rightarrow CD$ yra išvedama iš $AB \rightarrow C$ ir $A \rightarrow D$, panaudojant kompozicijos taisyklę. Lentelių skaidymas, kai išsaugomos visos atributų tarpusavio priklausomybės yra vadinamas **dekompozicija išsaugant priklausomybes**.

Suskaidžius pradinę lentelę L į dvi lenteles L_1 ir L_2, lentelės L_2 raktas A gali būti pirminiu, o lentelėje L_1 galima apibrėžti išorinį raktą, susiejant jos atributą A su L_2 pirminiu raktu.

Abi naujosios lentelės L_1 ir L_2 yra pradinės lentelės L projekcijos. Pradinės lentelės L duomenis galima atkurti jungiant jos projekcijas L_1 ir L_2. Aptariant reliacinės algebros projekcijos ir jungimo operacijas, minėjome, kad lentelių skaidymas, kai pradinės lentelės duomenys sutampa su jos projekcijų junginiu, vadinamas dekompozicija be praradimo. Tai, kad pasiūlytas lentelės skaidymas, siekiant užtikrinti 2NF, yra dekompozicija be praradimo, garantuoja Hezo (*I.J. Heath*) teorema.

Hezo (*I.J. Heath*) teorema. Tarkime, lentelėje $L(A, B, C)$ galioja funkcinė priklausomybė $A \rightarrow B$ arba funkcinė priklausomybė $A \rightarrow C$, kur A, B ir C – lentelės atributų aibės poaibiai. Tuomet lentelę L galima gauti jungiant jos projekcijas $L_1(A, B)$ ir $L_2(A, C)$.

Įrodymas. Teoremą įrodysime prieštaros būdu. Tarkime, $A \rightarrow B$ arba $A \rightarrow C$ ir lentelė $L(A, B, C)$ nėra jos projekcijų $L_1(A, B)$ ir $L_2(A, C)$ junginys, t.y. $L \neq L_1 \bowtie L_2$. Kadangi kiekvienai lentelei L ir jos projekcijoms L_1 ir L_2 galioja sąryšis $L \subseteq L_1 \bowtie L_2$, tai $L \neq L_1 \bowtie L_2$ reiškia, kad lentelių L_1 ir L_2 junginyje $L_1 \bowtie L_2$ yra eilutė $<a, b, c>$, kurios nėra lentelėje L. Jei eilutė $<a, b, c>$ priklauso L_1 ir L_2 junginiui, tai lentelėje L_1 yra eilutė $<a, b>$, o lentelėje L_2 – eilutė $<a, c>$. Vadinasi, lentelėje L yra eilutės $<a, b', c>$ ir $<a, b, c'>$, kuriose $b' \neq b$ ir $c' \neq c$. Todėl joje negalioja nei $A \rightarrow B$, nei $A \rightarrow C$. Tai prieštarauja prielaidai.

Hezo teorema garantuoja, jei lentelėje $L(A, B, C)$ galioja $A \rightarrow B$ arba $A \rightarrow C$, tai $L = \pi_{AB}(L) \bowtie \pi_{AC}(L)$. Jungdami lentelės L projekcijas, gauname tą pačią lentelę L. Kitaip tariant, jei $A \rightarrow B$ arba $A \rightarrow C$, tai lentelės L skaidymas į $\pi_{AB}(L)$ ir $\pi_{AC}(L)$ yra dekompozicija be praradimo.

Hezo teoremos teiginys nėra apverčiamas. Tai, kad $\pi_{AB}(L)$ ir $\pi_{AC}(L)$ yra lentelės L dekompozicija be praradimo, nereiškia, kad $A \rightarrow B$ arba $A \rightarrow C$. Pavyzdžiui,

L

A	B	C
a	b	c
a	b'	c
a	b	c'
a	b'	c'

L_1

A	B
a	b
a	b'

L_2

A	C
a	c
a	c'

Nors $L_1 = \pi_{AB}(L)$, $L_2 = \pi_{AC}(L)$ ir $L = L_1 \bowtie L_2$, tačiau šiems duomenims negalioja nei $A \rightarrow B$, nei $A \rightarrow C$.

Ne kiekvienas lentelės skaidymas yra skaidymas be praradimo. Pavyzdžiui, išskaidykime lentelę *Projektai_Vykdymas* į tokias dvi lenteles

Projektai_Vykdymas1(*Projektas*, *Pavadinimas, Svarba, Trukmė, Pradžia*, *Vykdytojas*),
Projektai_Vykdymas2(*Vykdytojas, Statusas, Valandos*).

Tuomet, net tik negalėtume antrajai lentelei apibrėžti teisingo rakto, bet taip skaidydami prarandame dalį informacijos. Nesunku įsitikinti, kad

$$Projektai\_Vykdymas \subset Projektai\_Vykdymas1 \bowtie Projektai\_Vykdymas2.$$

Jungdami lenteles *Projektai_Vykdymas1* ir *Projektai_Vykdymas2* negauname tikslių duomenų apie vykdytojų dalyvavimą konkrečiuose projektuose. Taip atsitinka dėlto, kad kiekviena lentelės *Projektai_Vykdymas1* eilutė jungiama su tiek *Projektai_Vykdymas2* eilučių, kiek kartų toje lentelėje yra paminėtas konkretus vykdytojas. Junginyje gausime, pavyzdžiui, kad vykdytojas Nr. 2 projektą Nr. 1 vykdo ir dokumentuotojo, ir analitiko, ir vadovo statusuose, nors iš tiesų jis tame projekte yra tik dokumentuotojas. Kadangi teoremos teiginys nėra apverčiamas, tai Hezo teorema nepaaiškina, kodėl taip atsitinka. Neformaliai galime tvirtinti, kad taip atsitinka todėl, kad taip skaidydami lentelę *Projektai_Vykdymas* prarandame funkcinę priklausomybę $\{Projektas, Vykdytojas\} \rightarrow \{Statusas, Valandos\}$. Vėliau suformuluosime bendresnę teoremą su tokia sąlyga, kuri bus apverčiama.

3.9′. Trečioji norminė forma

(Įterpiama prieš pastraipą, prasidedančią žodžiu „Literatūroje“)

Hezo (*I.J. Heath*) teorema mus užtikrina, kad toks skaidymas yra dekompozicija be praradimo. Akivaizdu, kad taip skaidant yra išsaugomos visos funkcinės priklausomybės.

Duotąją lentelę $L(\underline{A}, B, C)$, kurioje galioja dvi FP $A \to B$ ir $B \to C$, galima skaidyti ir kitaip, pvz. į tokias dvi jos projekcijas: $L_3(\underline{A}, B)$ ir $L_4(\underline{A}, C)$. Lentelėje L_3 galioja $A \to B$, o L_3 – galioja $A \to C$. Hezo (*I.J. Heath*) teorema garantuoja, kad ir šis skaidymas yra dekompozicija be praradimo. Tačiau tai nėra dekompozicija išsaugant priklausomybes, nes FP $B \to C$ nėra išvedama iš lentelėse L_3 ir L_4 galiojančių FP. Nepageidaujamas skaidymo, kai neišsaugomos FP, savybes detaliau aptarsime kitame šios knygos skyriuje.

Naudojant pateiktąjį skaidymą, kuriuo sudaroma 3NF reikalavimus tenkinančios lentelės, galime gauti ir pakankamai neefektyvią DB schemą. Tarkime, turime lentelę $L(\underline{A}, B, C, D)$, kurioje galioja FP $A \to B$, $B \to C$, $B \to D$. Tokia lentelė nėra 3NF, nes joje du atributai C ir D tranzityviai priklauso nuo rakto A. Jei lentelę skaidysime stengdamiesi panaikinti atributų CD tranzityviąją priklausomybę nuo rakto, tai gausime dvi lenteles: $L_1(\underline{A}, B)$ ir $L_2(\underline{B}, C, D)$. Tačiau, jei iš pradžių likviduojame vieno atributo tranzityviąją priklausomybę nuo rakto, o po to kito atributo, tai gauname tris lenteles: $L_1(\underline{A}, B)$, $L_3(\underline{B}, C)$ ir $L_4(\underline{B}, D)$. Abiem skaidymais visos gautosios lentelės yra 3NF. Nesunku įsitikinti, kad abudu skaidymai yra dekompozicijos be praradimo ir išsaugančios funkcines priklausomybes. Tačiau pirmasis skaidymas yra efektyvesnis, nes lentelės L_2 saugojimui kompiuterio atmintyje reikės mažiau vietos negu saugant dvi lenteles: L_3 ir L_4.

Egzistuoja nesudėtingas algoritmas, užtikrinantis efektyvią lentelių dekompoziciją be praradimo kai išsaugomos funkcinės priklausomybės.

<u>Algoritmas</u> lentelei, kurioje galioja FP aibė F, skaidyti, užtikrinant 3NF.

$G := F_{\min}$;
$i := 0$;
`for each` $X : \exists(X \to Y) \in G$
  $R := X$;
  `for each` $Y : \exists(X \to Y) \in G$
    $R := R \cup Y$;
  `endfor`
  $i := i + 1$;
  Į DB reliacinę schemą įtraukti $L_i(R)$;
`endfor`

Šio algoritmo pagrinde yra FP minimaliojo denginio $F_{\min}$ radimas. Toliau siekiama, kad visi atributai, priklausantys nuo to pačio atributų rinkinio, patektų į tą pačią lentelę. Pagal šį algoritmą, gausime tiek lentelių, kiek minimaliajame FP aibės denginyje yra skirtingų determinantų (kairiųjų FP pusių). Pasibaigus algoritmui kintamasis i yra lygus sudarytų lentelių skaičiui.

Pritaikykime šį algoritmą jau minėtai lentelei $L(\underline{A}, B, C, D)$, kurioje galioja FP aibė F: $A \to B$, $B \to C$, $B \to D$, skaidyti. Nesunku įsitikinti, kad ši FP aibė tenkina minimaliosios FP aibės reikalavimus, todėl ji kartu yra ir minimalusis denginys, $F_{\min} = F$. Kadangi aibėje $F_{\min}$ yra tik du skirtingi determinantai (A ir B), tai išorinio ciklo kūną reikia atlikti 2 kartus. Pirmąjį kartą imame $X := A$. Įvykdę vidinį ciklą gauname pirmosios lentelės schemą $L_1(\underline{A}, B)$. Vykdant išorinį ciklą antrą kartą, imame $X := B$. Įvykdę vidinio ciklo kūną du kartus (pirmąjį kartą $Y := C$, o antrąjį $Y := D$) gauname antrosios lentelės schemą $L_2(\underline{B}, C, D)$, kurią sudaro visi atributai, esantys funkcinėse priklausomybėse su determinantu B.

3.10'. Boiso-Kodo norminė forma

Abu pasiūlytieji lentelės *Projektų_Etapai* pertvarkymai yra skaidymai be praradimo. Nesunku įsitikinti, kad abiem lentelės skaidymo variantais yra išsaugojamos visos FP.

Prisiminus visiškai netrivialiosios priklausomybės ir lentelės viršrakčio apibrėžimus, BKNF apibrėžimą galima šiek tiek performuluoti. Lentelė yra BKNF FP aibės F atžvilgiu, jei kiekvienoje tenkinamoje FP $X \to A$ atributai X yra lentelės viršraktis.

BKNF apibrėžimas pastebimai skiriasi nuo 3NF apibrėžimo. Iš apibrėžimų nėra lengva nustatyti šių norminių formų tarpusavio priklausomybę. Parodysime, kad BKNF reikalavimai yra griežtesni už 3NF reikalavimus.

Teiginys. Jei lentelė yra BKNF, tai ji yra ir 3NF.
Įrodymas. Įrodysime prieštaros būdu. Tarkime lentelė $L(R)$ yra BKNF, bet ji nėra 3NF. Jei lentelė $L(R)$ nėra 3NF, tai joje būtinai egzistuoja atributų aibės poaibiai K, A, B tokie, kad K – raktas, A – nepirminiai, $A \not\subset KB$, kuriem galioja FP $K \to B$, $B \to A$ bei negalioja FP $B \to K$. Kadangi negalioja FP $B \to K$, tai B nėra raktas. Kadangi $A \not\subset B$, tai $B \to A$ yra visiškai netriviali, t.y. determinante nėra atributų, esančių dešiniojoje FP pusėje. Todėl FP $B \to A$ netenkina BKNF reikalavimų. Tai reiškia, kad lentelė $L(R)$ nėra BKNF.

Praktikoje dauguma lentelių, esančių 3NF, yra ir BKNF. Tik tuomet, kai lentelėje yra tenkinama FP $X \to A$, kurioje X nėra viršraktis ir A yra pirminis atributas, lentelė yra 3NF, bet nėra BKNF. Formalizuokime šį teiginį.

Teiginys. Jei lentelė yra 3NF ir nėra BKNF FP aibės F atžvilgiu, tai egzistuoja $X \to A \in F^+$, kurioje X nėra viršraktis ir A yra pirminis atributas.
Įrodymas. Tarkime, lentelė $L(R)$ yra 3NF ir nėra BKNF. Jei nėra BKNF, tai egzistuoja netrivialioji FP $X \to A$, kurioje X nėra viršraktis. Prieštaros būdu parodysime, kad A yra pirminis atributas. Tarkime, A – nepirminis. Jei X - ne viršraktis, tai bet kuriam lentelės L raktui K negalioja FP $X \to K$. Kadangi K - raktas, tai galioja $K \to X$. Kadangi FP $X \to A$ yra netrivialioji, tai $A \notin X$. Gavome, kad nepirminis atributas A tranzityviai priklauso nuo rakto K ($K \to X \in F^+$, $X \to K \notin F^+$, $X \to A$, ir $A \notin KX$). Vadinasi L nėra 3NF, o tai prieštarauja prielaidai.

Teiginys. Jei lentelė yra 3NF FP aibės F atžvilgiu ir egzistuoja FP $X \to A \in F^+$, kurioje X nėra viršraktis ir A yra pirminis atributas, tai ta lentelė nėra BKNF.
Įrodymas. Tarkime, lentelė $L(R)$ yra 3NF FP F atžvilgiu ir iš aibės F galima išvesti FP $X \to A$, kurioje X nėra viršraktis ir A yra pirminis atributas. Kadangi FP $X \to A$ determinantas X nėra viršraktis, tai ši lentelė netenkina BNKF reikalavimų, t.y. ji nėra BKNF.

3.10a. Nedalomosios lentelės

Tarkime, kiekvienas firmos darbuotojas gali turėti kelias kvalifikacijas, kurios įgyjamos sėkmingai baigus mokymo įstaigoje konkrečią mokymo programą, o informacijai apie tai yra skirta lentelė *Studijos*(*Nr*, *Kvalifikacija*, *Mokymo_programa*), kurios stulpeliai reiškia darbuotojo numerį, įgytos kvalifikacijos pavadinimą ir mokymo programos, kurią baigus buvo įgyta kvalifikacija, pavadinimą. Tarkime, ta pati kvalifikacija gali būti suteikta baigus kelias skirtingas mokymo programas, tačiau kiekvienai konkrečiai mokymo programai atitinka tik viena kvalifikacija. Papildomai tarkime, kad kiekvienas darbuotojas konkrečią kvalifikaciją gali įgyti tik vieną kartą baigęs vieną konkrečią mokymo programą. Tuomet ši lentelė turi du raktus: {*Nr*, *Kvalifikacija*} ir {*Nr*, *Mokymo_programa*}. Kadangi kvalifikacijos pavadinimas dažniausiai būna trumpesnis už programos pavadinimą, pirmąjį raktą parenkame pirminiu. Pateiksime šios lentelės duomenų fragmentą.

Studijos

Nr	*Kvalifikacija*	*Mokymo_programa*
1	Informatikas	Kompiuterių mokslas
1	Statistikas	Ekonometrija
5	Informatikas	Programų sistemos
5	Statistikas	Matematinė statistika

Šioje lentelėje galioja trys FP:

$\{Nr, Mokymo\_programa\} \to Kvalifikacija$,
$\{Nr, Kvalifikacija\} \to Mokymo\_programa$,
$Mokymo\_programa \to Kvalifikacija$.

Lentelėje *Studijos* visi atributai yra pirminiai, todėl ji yra 3NF. Kadangi trečiojoje visiškai netrivialioje (nesuprastinamoje) FP determinantas nėra raktas, tai lentelė nėra BKNF. Todėl lentelę *Studijos* reikia skaidyti į dvi:

Vykdytojų_studijos(*Nr*, *Mokymo_programa*),
Mokymo_programos(*Mokymo_programa*, *Kvalifikacija*).

Anksčiau pateiktieji lentelės *Studijos* duomenys pasiskirsto taip

Vykdytojų_studijos

Nr	*Mokymo_programa*
1	Kompiuterių mokslas
1	Ekonometrija
5	Programų sistemos
5	Matematinė statistika

Mokymo_programos

Mokymo_programa	*Kvalifikacija*
Kompiuterių mokslas	Informatikas
Ekonometrija	Statistikas
Programų sistemos	Informatikas
Matematinė statistika	Statistikas

Tačiau dabar pirmoje lentelėje galime apibrėžti tik trivialiąją FP

$\{Nr, Mokymo\_programa\} \to \{Nr, Mokymo\_programa\}$,

o antrojoje – tik vieną FP, išreiškiančią lentelės raktą,

$Mokymo\_programa \to Kvalifikacija$.

Iš šių dviejų FP mes negalime išvesti FP {*Nr*, *Kvalifikacija*} → *Mokymo_programa*, kuri galioja pradinėje lentelėje *Studijos*. Todėl gautąsias dvi lenteles, kurios yra pradinės lentelės *Studijos* projekcijos, negalima atnaujinti nepriklausomai. Pavyzdžiui, esant apibrėžtiesiems šių lentelių raktams, nėra kliūčių įterpti į lentelę *Vykdytojų_studijos* eilutę, kurioje nurodytas darbuotojo Nr. 1 ir mokymo programa „Programų sistemos“. Tačiau toks įterpimas turėtų būti atmestas, kadangi baigusiesiems šią programą yra suteikiama informatiko kvalifikacija, kurią darbuotojas Nr. 1 jau turi. Pastarajam faktui patikrinti, papildomai reikia kreiptis į kitą lentelę *Mokymo_programos*. Pradinėje lentelėje *Studijos*, nepažeidžiant raktų vientisumo, tokių duomenų įterpti nebuvo galima. Suskaidydami lentelę *Studijos* į dvi jos projekcijas: *Vykdytojų_studijos* ir *Mokymo_programos* sudarėme DB schemą tenkinančią BKNF reikalavimus, tačiau praradome vieną svarbią duomenų tarpusavio priklausomybę. Pastebėsime, kad šis suskaidymas yra be praradimo, t.y. lentelės *Studijos* duomenys yra visiškai atstatomi jungiant jos projekcijas. Tai mes žinome iš Hezo (*I.J. Heath*) teoremos. Netgi skaidymas be praradimo ne visada užtikrina atributų tarpusavio FP nepraradimą.

Lentelė *Studijos* yra nedalomosios lentelės pavyzdys. Lentelė vadinama **nedalomąja lentele**, jeigu jos negalima suskaidyti į tarpusavyje nepriklausomas lenteles, formaliau, jos negalima suskaidyti taip, kad išsaugotume visas priklausomybes. Jei lentelę *Studijos* neskaidome, tai susiduriame su duomenų pertekliumi, kuris sukelia duomenų anomalijas. Tačiau, jei ją skaidome, prarandame FP. Lentelei *Studijos* negalime rasti visais atžvilgiais teisingo sprendimo. Tai reliacinio modelio ribotumas. Konkrečiose DBVS lentelę *Studijos* reikėtų skaidyti, kad ji tenkintų BKNF reikalavimus, o dėl skaidymo prarandamas FP galima užtikrinti reliaciniam modeliui nebūdingomis priemonėmis, pvz. trigeriais, kuriuos aptarsime kituose knygos skyriuose.

Priminsime, kad ankstesniajame skyrelyje atliktas lentelės *Projektų_Etapai* pertvarkymas abiem būdais yra skaidymai be praradimo ir išsaugojantys FP. Todėl lentelei, esančiai 3NF, bet neesančiai BKNF, gali egzistuoti suskaidymas be praradimo kai išsaugojamos visos FP.

3.11′. Ketvirtoji norminė forma

Daugiareikšmės priklausomybės (DRP) apibrėžimas yra simetriškas. Tarkime, lentelėje $L(A, B, C)$ galioja DRP $A \twoheadrightarrow B$. Tuomet lentelėje L yra trys eilutės <a, b, c>, <a, b′, c′>, <a, b′, c>. Pritaikykime DRP apibrėžimą toms pačioms pradinėms eilutėms, tik sukeiskime jas vietomis. Gauname: jei lentelėje yra <a, b′, c′> ir <a, b, c>, tai lentelėje turi būti ir <a, b, c′>. Todėl, jei lentelėje $L(A, B, C)$ galioja DRP $A \twoheadrightarrow B$, tai bet kurioms dviem lentelės L eilutėms <a, b, c>, <a, b′, c′> lentelėje yra dar dvi eilutės <a, b′, c> ir <a, b, c′>. Šią simetriškumo savybę galima suformuluoti teiginiu.

Teiginys. Jei lentelėje $L(A, B, C)$ galioja DRP $A \twoheadrightarrow B$, tai joje galioja ir DRP $A \twoheadrightarrow C$.

Įrodymas. Tarkime, lentelėje $L(A, B, C)$ galioja $A \twoheadrightarrow B$, bet negalioja $A \twoheadrightarrow C$. Jei negalioja $A \twoheadrightarrow C$, tai atributo A reikšmei a lentelėje galima rasti 2 eilutes: <a, b, c>, <a, b′, c′>, bet nėra bent vienos iš šių dviejų eilučių: <a, b′, c>, <a, b, c′>. Tai reiškia, kad lentelėje L negalioja $A \twoheadrightarrow B$. Prieštaravimas.

Pateiksime, dar vieną DRP apibrėžimo galimą formuluotę. Lentelėje $L(A, B, C)$ galioja DRP $A \twoheadrightarrow B$ tada ir tik data, kai bet kurioms dviems lentelės L eilutėms: e_1 ir e_2, kurioms $e_1(A) = e_2(A)$, lentelėje L egzistuoja eilutės e_3 ir e_4 tokios, kad $e_3(A) = e_1(A)$, $e_4(B) = e_1(B)$, $e_4(C) = e_2(C)$, $e_4(A) = e_1(A)$, $e_3(B) = e_2(B)$, $e_3(C) = e_1(C)$.

Lentelė $L(R)$ yra **ketvirtosios norminės formos** (**4NF**) FP ir DRP aibės F atžvilgiu, tada ir tik tada, kai kiekviena DRP $A \twoheadrightarrow B \in F^+$, $A \subset R$, $B \subset R$, arba yra triviali, arba jos determinantas A yra lentelės L viršraktis.

Jeigu DRP $A \twoheadrightarrow B$ determinantas A yra lentelės raktas, tai ši DRP faktiškai yra FP. Todėl, 4NF apibrėžime vietoje reikalavimo, kad kiekvienos netrivialios DRP determinantas yra lentelės viršraktis, galima pakeisti reikalavimu, kad kiekviena netriviali DRP būtų FP. Tam, kad lentelė L būtų 4NF, lentelėje turi galioti tik FP, kuriose determinantas yra viršraktis, arba, kitaip tariant, tik visiškai netrivialios FP, kuriose determinantas yra raktas. Todėl lentelė, esanti 4NF yra ir BKNF.

Jau minėjome, kad kiekviena FP yra ir DRP. Tačiau ne kiekviena DRP yra FP. Paimkime, pavyzdžiui tokią paprastą lentelę *Pavyzdys*.

Pavyzdys

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

Šios lentelės duomenys tenkina FP $A \rightarrow B$ reikalavimus. Vadinasi, yra tenkinama ir DRP $A \twoheadrightarrow B$. Iš pirmo žvilgsnio gali atrodyti, kad taip nėra, bet ši DRP visiškai tenkina DRP apibrėžimo reikalavimus. Iš aukščiau įrodyto teiginio seka, kad šioje lentelėje galioja ir $A \twoheadrightarrow C$. Nesunku įsitikinti, kad ir ši DRP tenkina jai keliamus reikalavimus. Tačiau atitinkama FP $A \rightarrow C$ negalioja.

Tarkime, papildomai kalbų, kurias moka darbuotojai, pavadinimams duomenų bazėje reikia įsiminti ir jų mokėjimo lygmenį. Tam papildykime lentelę *Kalbos* stulpeliu *Lygmuo*

Kalbų_mokėjimas

Nr	*Kalba*	*Lygmuo*
1	Anglų	Puikiai
1	Vokiečių	Gerai
2	Anglų	Gerai

Naujojoje lentelėje galioja tik viena FP $\{Nr, Kalba\} \rightarrow Lygmuo$, kurios determinantas yra lentelės raktas. Lentelėje *Kalbos*, kurioje nėra stulpelio *Lygmuo*, galioja DRP $Nr \twoheadrightarrow Kalba$, tačiau naujojoje lentelėje ji negalioja. Ši lentelė yra 4NF, nes $\{Nr,\ Kalba\} \to\to Lygmuo$ yra trivialioji DRP, be to ji yra ir FP, kurios determinantas yra raktas.