VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
INFORMATIKOS KATEDRA

Biografinių užklausų apdorojimas paskirstytose biometrinio identifikavimo sistemose

Biographic Query Processing in Distributed Biometric Identification Systems

Magistro baigiamasis darbas

Atliko: Rytis Karpuška (parašas)

Darbo vadovas: prof. habil. dr. Mindaugas Bloznelis (parašas)

Recenzentas: (parašas)

Vilnius – 2019

Turinys

Įvadas

Kiekvieno žmogaus kūnas turi aibę požymių, pagal kuriuos jį galima unikaliai identifikuoti (pvz.: pirštų atspaudai). Šių požymių egzistavimas davė pagrindą *biometrinėms identifikavimo sistemoms*. Šių sistemų paskirtis yra tarp visų užregistruotų žmonių surasti tą, kuriam priklauso duotieji biometriniai požymiai. Šiame darbe nagrinėjamoje sistemoje [Neu18] sąrašas žmonių, tarp kurių yra vykdoma paieška, yra papildomai atrenkamas pagal vartotojo pateiktą užklausą. Ši užklausa sąrašą paieškai sudaro pagal papildomus, ne biometrinius, duomenis (pvz.: amžių, lytį, gyvenamąją vietą) priskirtus kiekvienam sistemoje užregistruotam žmogui.

Biografiniai atributai bei biografinė schema. Aptartieji papildomi duomenys vadinami *biografiniais atributais*. Verta atkreipti dėmesį į tai, kad biografiniai atributai yra papildomi, būtinai ne biometriniai, duomenys.

Kiekvienas biografinis atributas turi savo vardą (pvz.: „Miestas“, „Amžius“), bei reikšmę (pvz.: „Vilnius“, „25-eri metai“). Visų atributų vardų aibė yra vadinama *biografine schema*. Pavyzdžiui 1 lentelėje pateikiamame duomenų bazės pavyzdyje biografinė schema būtų {„Miestas“, „Amžius“}.

Žmogus	**Biometriniai požymiai**	**Biografiniai atributai**	
		Miestas	**Amžius**
Mindaugas	biometrinių požymių įrašas	Vilnius	35
Petras	biometrinių požymių įrašas	Utena	15
Eglė	biometrinių požymių įrašas	Zarasai	10
Dovilė	biometrinių požymių įrašas	Kelmė	20
Rytis	biometrinių požymių įrašas	Marijampolė	45
Tomas	biometrinių požymių įrašas	Anykščiai	30

1 lentelė. Pavyzdiniai biometrinės identifikavimo sistemos duomenys

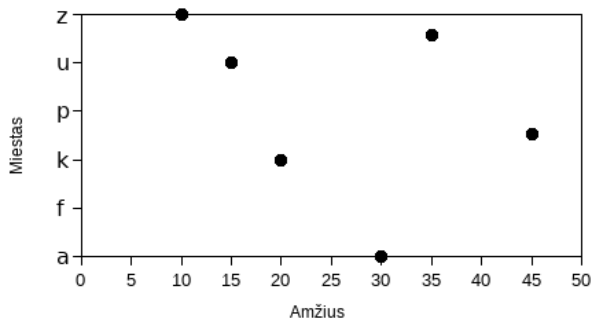
Kiekvienam žmogui gali būti priskiriamas daugiau negu vienas biografinis atributas, tačiau visiems žmonėms priskiriamų biografinių atributų schema (biografinė schema) turi būti tokia pati. Tai reiškia, kad, pavyzdžiui, jeigu Petrui (žr.: 1 lentelę) buvo priskirtas gyvenamasis miestas ir amžius, tai visiems likusiems užregistruotiems žmonės irgi bus priskiriamas gyvenamasis miestas ir amžius.

Nagrinėjama sistema palaiko dviejų tipų biografinius atributus:

- skaitinio tipo
- simbolių eilutės tipo

Tačiau esant poreikiui šis atributų tipų sąrašas gali būti plečiamas. Sistemoje visi tą patį vardą turintys atributai yra ir to pačio tipo.

Duomenų bazės modelis. Šiame darbe remiamasi duomenų bazės modeliu „Data cube“ [Mar00]. Nagrinėjamos sistemos atveju, kiekvienas atributas atitinka vieną, konkrečią, daugiamatės erdvės dimensiją.



1 pav. Pavyzdiniai biografinių atributų rinkiniai dvimatėje erdvėje.

1 paveikslėlyje pateikiamas tokios erdvės pavyzdys atitinkantis lentelėje 1 pateiktą duomenų bazės pavyzdį.

Duomenų bazės įrašams apdoroti bus naudojama palyginimo funkciją (1).

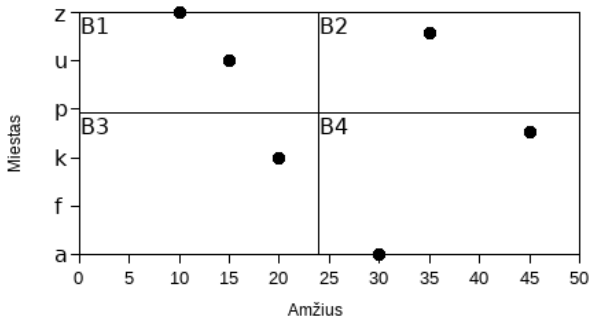
$$f_A(x_1, x_2) = \begin{cases} -1, & \text{jeigu } x_1 < x_2 \\ 0, & \text{jeigu } x_1 = x_2 \\ 1, & \text{jeigu } x_1 > x_2 \end{cases} \tag{1}$$

Čia x_1 ir x_2 yra tą patį vardą A turinčių atributų reikšmės. Simbolių eilutės tipo atributų reikšmės gali būti lyginamos leksikografine tvarka.

Šiame darbe daroma prielaida, kad palyginimo funkcijos (1) atžvilgiu atributo reikšmių aibė yra pilnai sutvarkyta [HJ99].

Biometrinių įrašų blokai daugiamatėje erdvėje. Nagrinėjamos sistemos bazinis elementas yra įrašas. Vieną įrašą sudaro biometrinių požymių rinkinys susietas su biografinių atributų reiškmių rinkiniu. Sistemoje, operatyviojoje atmintyje, įrašai yra saugomi ir apdorojami ne po vieną, bet grupėmis. Šios grupės yra vadinamos *biometrinių įrašų blokais*. Kiekvienam įrašui registracijos metu yra laisvai parenkamas vienas blokas, kuriame jis bus saugomas.

Šiame darbe nagrinėjamame duomenų bazės modelyje biometrinių įrašų blokai yra ankščiau aptartos daugiamatės erdvės sritys, kurios apima visus konkrečiam blokui priskirtus įrašus (žr.: 2 pav.)



2 pav. B1-B4 - erdvės sritys atitinkančios biometrinių įrašų blokus

Biografinės užklausos. Nurodžius pasirinktų biografinių atributų kitimo sritis – apibrėžiama įrašų aibė, kurioje ieškomi pasirinktų biometrinių požymių atitikmenys. Tokios paieškos užduotis vadinama *biografine užklausa*. Šiai užklausai aprašyti yra skirta gramatika, kurios Backus ir Nauro forma [MR03] yra pateikiama 2 lentelėje.

užklausa	::=	<vienaris operatorius> <operandas> |
		<operandas> <dvinaris operatorius> <operandas> |
		”(” <užklausa> ”)” |
		<atributo vardas> <sąrašo operatorius> <sąrašas>
operandas	::=	<užklausa> | <atributo vardas> | ”” <skaičius> ”” | ”” <žodis> ””
vienaris operatorius	::=	<neprivalomas tarpas> ”NOT” <neprivalomas tarpas>
dvinaris operatorius	::=	<neprivalomas tarpas> <dvinario operatoriaus ženklas> <neprivalomas tarpas>
dvinario operatoriaus ženklas	::=	”>” | ”>=” | ”<” | ”<=” | ”=” | ”<>” | ”AND” | ”OR”
sąrašo operatorius	::=	<neprivalomas tarpas> ”IN” <neprivalomas tarpas>
sąrašas	::=	”(” <sąrašo elementai> ”)”
sąrašo elementai	::=	<sąrašo elementas> | <sąrašo elementas> ”,” <sąrašo elementai>
sąrašo elementas	::=	”” <žodis> ”” | ”” <skaičius> ””
atributo vardas	::=	<neprivalomas tarpas> <žodis> <neprivalomas tarpas>
neprivalomas tarpas	::=	”” | ” ” <neprivalomas tarpas>
žodis	::=	<raidė> | <žodis> <raidė> | <žodis> <skaičius>
skaičius	::=	”0” | ”1” | ”2” | ”3” | ”4” | ”5” | ”6” | ”7” | ”8” | ”9”
raidė	::=	”A” | ”B” | ”C” ... ”Z” | ”a” | ”b” | ”c” ... ”z”

2 lentelė. Biografinės užklausos aprašymo gramatikos Backus ir Nauro forma

Keletas užklausos pavyzdžių pateikiama 3 lentelėje laikant, kad biografinė schema yra {„Miestas“, „Amžius“}.

	Užklausos aprašymas	**Užklausos interpretacija**
Pavyzdys 1	Amžius >= ’18’	Visi suaugę žmonės
Pavyzdys 2	Amžius >= ’18’ AND Miestas IN (’Vilnius’, ’Kaunas’)	Visi suagę vilniečiai ir kauniečiai
Pavyzdys 3	NOT (Miestas = ’Vilnius’ AND Amžius >= ’18’)	Visi žmonės išskyrus suaugusius vilniečius

3 lentelė. Užklausų aprašymo pavyzdžiai.

Biografinių užklausų apdorojimas. Smulkiausias įrašų sąrašas kuriame sistema [Neu18] šiuo metu gali vykdyti paiešką yra vienas biometrinių įrašų blokas. Todėl vartotojo biografinės užklausos apdorojimas yra skaidomas į tokius tris etapus:

1. Atmetimo etapas: Atmetami visi blokai, neturintys įrašų, kurių atributų rinkiniai atitinka vartotojo užklausą.
2. Paieškos etapas: Atliekama paieška blokuose, kurie liko po atmetimo etapo.
3. Filtravimo etapas: Atmetami visi paieškos etape rasti įrašai, kurių atributų rinkiniai neatitinka vartotojo užklausos.

Verta pastebėti, kad po atmetimo etapo likęs blokų kiekis priklauso nuo metodo, pagal kurį yra parenkama daugiamatės erdvės sritis atributų rinkinių saugojimui.

Darbo tikslas. Šio darbo metu siekiama padidinti biometrinės identifikavimo sistemos [Neu18] pralaidumą (užklausų skaičių per laiko vienetą). Tuo tikslu ketinama minimizuoti biometrinių įrašų blokų skaičių, kuris lieka po atmetimo etapo.

Tam bus pritaikyti ir palyginti du įrašų priskirimo blokams metodai:

- Metodas paremtas priešdėliniu rūšavimu.
- Metodas paremtas K-d medžiu.

Darbą planuojama skirstyti į tokius uždavinius:

1. Išanalizuoti tipinių užklausų statistines savybes, pasiruošti testavimo duomenis.
2. Empiriškai įvertinti sistemos pralaidumo priklausomybę nuo įrašų skaičiaus bloke.
3. Įdiegti metodą paremtą priešdėliniu rūšiavimu.
4. Įdiegti metodą paremtą K-d medžiu.
5. Palyginti įdiegtus metodus tarpusavyje ir su sistema [Neu18]

Numatomas darbų eiliškumas:

1. Apžvelgiami kiti panašūs darbai bei pateikiami šaltiniai, kuriais remiasi šis darbas.
2. Apžvelgiama kokios užklausų savybės išplaukia iš apibrėžtos gramatikos, kokiomis statistinėmis savybėmis pasižymi vartotojų dažniausiai pateikiamos užklausos.
3. Aprašoma kaip bus palyginami įrašų priskyrimo sritims metodai, bei aprašomas testavimo duomenų rinkinys.
4. Empiriškai nustatoma sistemos pralaidumo priklausomybė nuo maksimalaus bloko dydžio. Parenkamas bloko dydis, kuris bus naudojamas palyginant metodus.

5. Įgyvendinamas ir palyginamas metodas paremtas priešdėliniu rūšiavimu.

6. Įgyvendinamas ir palyginamas metodas paremtas K-d medžiu.

7. Aprašomi pasiekti rezultatai bei pateikiamos išvados.

1. Užklausų klasifikacija

Panaši problema į aptartąją įvade yra nemažai tyrinėta duomenų bazių kontekste. Jei šiame darbe siekiama minimizuoti biometrinių įrašų blokų skaičių likusį po atmetimo etapo, tai duomenų bazėse siekiama minimizuoti disko operacijų skaičių siekiant sumažinti duomenų bazės atsako laiką bei padidinti pralaidumą (apdorotų užklausų skaičių per laiko vienetą) [GUW00]. Tuo tikslu yra naudojamos duomenų struktūros, vadinamos *duomenų indeksu*. Jeigu duomenys yra daugiamačiai (vienas įrašas gali susidėti iš daugiau negu vienos reikšmės), tuomet indeksą vadiname *daugiamačių duomenų indeksu*, o metodą pagal kurį šis indeksas yra sudaromas vadiname *daugiamačių duomenų indeksavimo metodu*.

1.1. Užklausų klasės

Šiame darbe palyginame biometrinių įrašų priskyrimo blokams metodus sistemoje [Neu18] ir siekiame apimti kiek galima daugiau užklausų klasių. Gaede ir Günther [GG98] patiekia tokią daugiamačių užklausų klasifikaciją:

1. Griežto atitikmens užklausa
2. Taško užklausa
3. Lango užklausa
4. Regiono užklausa
5. Apgaubiančioji užklausa
6. Pilno regiono užklausa
7. Kaimynų užklausa
8. Artimiausio kaimyno užklausa

Gaede ir Günther [GG98] duomenis ir užklausas nagrinėja d dimensijų Euklido erdvėje E^d. Atskirus įrašus apibrėžia kaip objektus o. Objektas o turi vieną atributą (žymime $o.G$), kuris apibūdina objekto o padėtį erdvėje E^d. Šis $o.G$ atributas yra aibė taškų, kuriuos objektas o užima erdvėje E^d. Taip pat objektas o gali turėti ir papildomų atributų, nesusijusių su erdve E^d (pvz.: vardas, pavadinimas, amžius...). Objektams yra apibrėžiami operatoriai $=$, $\cap$, bei $dist(o_1, o_2)$. Du objektai o_1 ir o_2 yra lygūs, $o_1 = o_2$, tada ir tik tada jeigu abiejų objektų užimamų taškų aibės, t.y. atributai, $o_1.G$ ir $o_2.G$ sutampa. Dviejų objektų o_1 ir o_2 sankirta $o_1 \cap o_2$ yra aibė taškų, kurie patenka ir į $o_1.G$ ir į $o_2.G$. Ir galiausiai atstumas tarp dviejų objektų o_1 ir o_2 $dist(o_1, o_2)$ skaitomas atstumas tarp artimiausių taškų p_1, p_2, kur $p_1 \in o_1.G$ ir $p_2 \in o_2.G$.

1.1.1. Griežto atitikmens užklausa

Griežto atitikmens užklausa yra tokia užklausa, kuri duotąjam objektui o' erdvėje E^d randa visus objektus, kurių erdvės sritys sutampa su duotąja.

$$EMQ(g) = \{o | o'.G = o.G\} \tag{2}$$

1.1.2. Taško užklausa

Taško užklausa yra tokia užklausa, kuri duotam taškui p erdvėje E^d randa visus objektus, kurie turi tašką p.

$$PQ(p) = \{o | \{p\} \cap o.G = \{p\}\} \tag{3}$$

Žiūrėti priedą Nr. 2.

1.1.3. Lango užklausa

Lango užklausa, tai tokia užklausa, kuri duotiems d intervalams $I^d = [l_1, u_1], [l_2, u_2], ..., [l_d, u_d]$ (po vieną intervalą kiekvienai erdvės E^d koordinačių ašiai) sudaro objektą o', kurio atributas:

$$o'.G = o'_1.G_1 \cap o'_2.G_2 \cap ... \cap o'_d.G_d \tag{4}$$

Čia o'_i žymi objektą turintį vienintelį atributą $o'_i.G_i$ sudarytą iš tų E^d taškų, kurių i-toji koordinatė patenka į intervalą $[l_i, u_i]$. Taigi $o'.G \subset E^d$ yra stačiakampis gretasienis $[l_1, u_1] \times [l_2, u_2] \times ... \times [l_d, u_d]$. Lango užklausa randa visus objektus, kurie turi nors vieną bendrą tašką su objektu o'.

$$WQ(o') = \{o | o'.G \cap o.G \neq \emptyset\} \tag{5}$$

Verta pastebėti, kad lango užklausos atveju stačiakampio gretasienio $o'.G$ kraštinės visada yra lygegriačios erdvės E^d koordinačių ašims. Žiūrėti priedą Nr. 2.

1.1.4. Regiono užklausa

Regiono užklausa yra labai panaši į lango užklausą, tačiau o' kraštinės gali būti laisvai pasirinktos ir neprivalo būti lygegriačios koordinačių ašims. Regiono užklausa randa visus objektus, kurie turi bendrų E^d taškų su duotuoju objektu o'.

$$IQ(o') = \{o | o'.G \cap o.G \neq \emptyset\} \tag{6}$$

Žiūrėti priedą Nr. 2.

1.1.5. Apgaubiančioji užklausa

Apgaubiančioji užklausa yra tokia užklausa, kuri randa visus objektus o, kurių aibė $o.G$ pilnai apima pateikto objekto o' aibę $o'.G$.

$$EQ(g) = \{o|(o'.G \cap o.G) = o'.G\} \tag{7}$$

Žiūrėti priedą Nr. 2.

1.1.6. Pilno regiono užklausa

Pilno regiono užklausa iš esmės yra priešinga apgaubiančiąjai. Pilno regiono užklausa yra tokia užklausa, kuri randa visus objektus o, kurių erdvės sritis $o.G$ pilnai patenka į pateikto objekto o' erdvės sritį $o'.G$.

$$CQ(g) = \{o|(o'.G \cap o.G) = o.G\} \tag{8}$$

Žiūrėti priedą Nr. 2.

1.1.7. Kaimynų užklausa

Kaimynų užklausa yra tokia užklausa, kuri pagal duotąjį objektą o' suranda visus šio objekto kaimynus, t.y. objektus o kurių atributai $o.G$ turi bendrą kraštą su $o'.G$ erdvėje E^d.

$$AQ(o') = \{o|\delta(o'.G) \cap \delta(o.G) \neq \emptyset \text{ ir } (o'.G)^\circ \cap (o.G)^\circ = \emptyset\} \tag{9}$$

Čia $(o.G)^\circ$ reiškia visus vidinius taškus, kurie priklauso aibei $o.G$. Ir $\delta(o.G) = (o.G) \setminus (o.G)^\circ$. Žiūrėti priedą Nr. 2.

1.1.8. Artimiausio kaimyno užklausa

Artimiausio kaimyno užklausa yra tokia užklausa, kuri duotam objektui o' randa artimiausią kaimyną (verta pastebėti, kad šiuo atveju kaimynai gali ir neturėti bendrų taškų).

$$NNQ(o') = \{o|\forall o'' : dist(o'.G, o.G) \leq dist(o'.G, o''.G)\} \tag{10}$$

2. Daugiamačių duomenų indeksavimo metodų klasifikacija

Lu ir Ooi [LO93], Gaede ir Günther [GG98], bei Böhm, Christian, Stefan, Daniel A [BBK01] apžvelgia įvairius daugiamačių duomenų indeksus, bei pateikia schemą padedančią suprasti jų istoriją (žiūrėti priedą Nr. 1).

Autoriai šiuos metodus suskirsto į tris grupes:

- Metodai paremti maišos funkcijomis.
- Hierarchiniai metodai.
- Metodai paremti erdvę užpildančiomis kreivėmis [Bad12].

2.1. Maišos funkcijomis paremti metodai daugiamačių duomenų indeksavimui

Bendruoju atveju maišos funkcijomis paremti metodai yra taikomi griežto atitikmens, bei taško užklausoms. Lango, regiono ir kitoms, panašioms, užklausoms apdoroti šie metodai yra pernelyg sudėtingi [NHS81] [TS82] ir mūsų atveju mažiau efektyvūs. Dėl šios priežasties jie nebus nagrinėjami, įgyvendinami ir lyginami šiame darbe.

2.2. Hierarchiniai metodai daugiamačių duomenų indeksavimui

Hierarchiniai daugiamačių duomenų indeksai yra paremti dvejetainiais ar aukštesnio šakų skaičiaus (viršūnės laipsnio) medžiais (trejetainiai, ketvirtainiai...) [GG98]. Šie indeksai saugo įrašus ne po vieną, bet grupėmis. Dažniausiai kiekviena grupė yra randama medžio lapuose (vadinama *duomenų viršūne*). Vidinės viršūnės (vadinamos *indekso viršūnėmis*) yra naudojamos kaip kelrodžiai ieškant duomenų viršūnių. Kiekviena indekso viršūnė nurodo visas duomenų viršūnes, kurios gali būti rastos šios indekso viršūnės pomedyje.

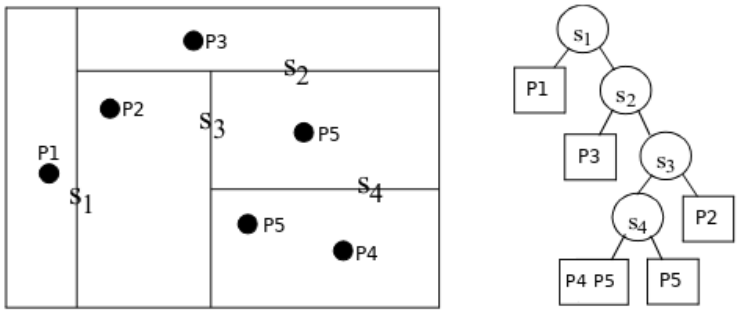
Keletas tokių medžių pavyzdžių yra R-medis [Gut84], R*-medis [BKS+90], R+-medis [SRF87], Kd-medis [Ben79], Kdb-medis [Rob81], X-medis [BKK96], Quad-medis [Hab83] ir daug kitų.

Užklausų apdorojimas yra skaidomas į du etapus [BBK01]:

1. Paieška medyje.
2. Įrašų filtravimas.

Paieškos etape, medis yra peržvelgiamas nuo šaknies iki lapų ir atrenkamos visos duomenų viršūnės, kurios gali turėti įrašų, atitinkančių mūsų užklausą. Filtravimo metu atrenkami įrašai esantys paieškos etape atrinktose duomenų viršūnėse ir tenkinatys užklausą [BKS+94] [BBK01]. Verta pastebėti, kad sistemos [Neu18] užklausų apdorojimo etapai yra labai panašūs.

Kd-medis. Šiame darbe aptariame biometrinių įrašų priskyrimo blokams metodą paremtą Kd-medžiu [Ben79]. Kd-medis yra dvejetainis medis, kurio lapuose (duomenų viršūnėse) yra saugomi vienas ar daugiau d-mačių taškų. Kiekviena vidinė medžio viršūnė (indekso viršūnė) dalina erdvę E^d į dvi nepersidengiančias dalis. Taškai patenkantys į vieną dalį patenka į kairyjį pomedį, o patenkantis į kitą dalį – į dešinį pomedį (žr.: 3 pav.). Erdvės dalinimas į dvi dalis yra parenkamas taip: kiekvienai indekso viršūnei yra parenkama koordinačių ašis (pvz.: x) ir reikšmė (pvz .: 5). Tuomet visi taškai, kurių x koordinatės reikšmė yra mažesnė už parinktą reikšmę (pavyzdyje $[-\infty; 5)$), patenka į kairyjį pomedį, o likę taškai (pavyzdyje $[5; \infty)$) – į dešinį pomedį. Nėra griežtai apibrėžta kokia koordinačių ašis ar reikšmė šioje ašyje turi būti parinkta kiekvienai viršūnei, todėl šiuo tikslu taikomos įvairios euristikos.



3 pav. Kd-medžio pavyzdys.

Paveiksliuke 3 pateikiamas Kd-medžio pavyzdys. Čia kairėje – šį medį atitinkanti erdvė E^d. Taškai $P1$-$P5$ atitinka daugiamačius duomenis, o tiesės $S1$-$S4$ - erdvės E^d padalijimus atitinkmai medžio viršūnėse $S1$ - $S4$.

Hierarchiniai metodai daugiamačių duomenų indeksavimui yra plačiai taikomi duomenų bazių [BBK01], sensorių tinklų [LKG+03], paveiksliukų požymių atpažinimo [SH08], privatumo užtikrinimo [HMC+12] [XXY10] ir kitose srityse.

2.3. Erdvę užpildančiomis kreivėmis paremti metodai daugiamačių duomenų indeksavimui

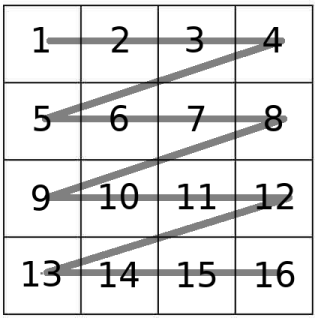
Kaip ir hierarchiniuose metoduose taip ir erdvę užpildančiomis kreivėmis paremtuose metoduose duomenys nagrinėjami kaip taškai d-matėje erdvėje E^d.

Daugiamačių duomenų negalime griežtai surikiuoti į eilę, taip kad, eilėje greta esantys taškai būtų arti ir erdvėje E^d. Dėl šios priežasties daugiamačių duomenų indeksavimo metodai yra sudėtingesni lyginant su vienmačiu atveju [GG98] [BBK01] Tačiau egzistuoja įvairiomis euristikomis paremtų griežtos tvarkos sudarymo metodų, kurie į eilę sudeda taškus taip, kad eilėje greta esantys taškai su tam tikra tikimybe būtų arti ir erdvėje E^d. Surikiavus taškus pagal šią tvarką juos galima indeksuoti vienmačiais indeksavimo metodais (B-medžiu [Com79] RB-medžiu [HOS97] ir pan.).

Siekiant sudaryti tokią tvarką, E^d erdvė suskirstoma į gardelę (į kiekvieną laukelį šioje gardelėje gali patekti nulis, vienas ar daugiau taškų). Kiekvienam laukeliui šioje gardelėje yra priskiriamas unikalus skaičius pagal kurį laukeliai yra surikiuojami ir indeksuojami vienmačiais indeksavimo metodais. Šie unikalūs skaičiai yra priskiriami pagal tvarką, kuria erdvę užpildanti kreivė „prabėga“ pro laukelius [Bad12]. Aptarsime dvi tokias kreives:

1. Eilutinę kreivę.
2. Z-kreivę.

Eilutinė kreivė. Eilutinė kreivė yra viena iš paprasčiausių erdvę užpildančių kreivių. Ji negali būti naudojama begalinėje erdvėje, todėl tarkime, kad $[L_1, U_1] \times [L_2, U_2] \times ... \times [L_d, U_d]$ yra stačiakampis gretasienis erdvėje E^d kurį eilutinė kreivė užpildo (paveiksle 4 šis greitasienis būtų $[1, 4] \times [1, 4]$). Eilutinė kreivė iš pradžių „prabėga“ visus taškus $[x, L_2, ..., L_d]$, kur x patenka į intervalą $[L_1, U_1]$. Šis prabėgimas atliekamas x didėjimo tvarka. Paskui „prabėga“ $[x, L_2 + 1, ..., L_d]$, paskui $[x, L_2 + 2, ..., L_d]$ ir t.t. (žr.: 4 pav.).



4 pav. Eilutinės erdvę užpildančios kreivės pavyzdys.

Formaliai pilna tvarka yra apibrėžiama sąryšio:

$$p_1 < p_2 \text{ jeigu } RWSFC(p_1) < RWSFC(p_2) \tag{11}$$

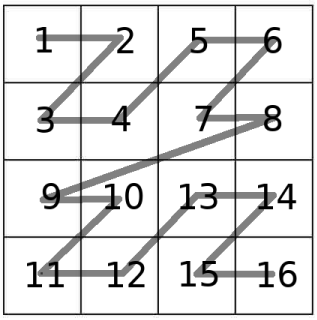
Čia p_1 ir p_2 yra taškai erdvėje E^d, o $RWSFC(p)$:

$$RWSFC(p) = 1 + \sum_{i=1}^{d} [(p_i - L_i) \prod_{j=0}^{i-1} RL(j)] \tag{12}$$

$$RL(j) = \begin{cases} U_j - L_j, \text{jeigu } j > 0 \\ 1, \text{ kitu atveju} \end{cases} \tag{13}$$

Z-kreivė. Z-kreivė daugiamačių duomenų indeksavimo metodų, paremtų erdvę užpildančiomis kreivėmis, kontekste yra viena iš populiariausių [RMF+00]. Nagrinėsime atvejį, kuriame Z-kreivė užpildo stačiakampį gretasienį $[1, 2^b] \times [1, 2^b] \times ... \times [1, 2^b]$ erdvėje E^d (5 paveiksliuke pateiktame pavyzdyje $b = 2$). Z-kreivė visų pirma gretasienį padalina į dvi lygias sritis

$P_0 = [1, 2^{b-1}] \times [1, 2^b] \times ... \times [1, 2^b]$ ir $P_1 = [2^{b-1}, 2^b] \times [1, 2^b] \times ... \times [1, 2^b]$. Visi gardelės laukeliai esantys srityje P_0 yra pirmiau tų, kurie yra srityje P_1. Sekančiame žingsnyje P_0 yra padaliniama į dvi lygias sritis $P_{00} = [1, 2^{b-1}] \times [1, 2^{b-1}] \times ... \times [1, 2^b]$ ir $P_{01} = [1, 2^{b-1}] \times [2^{b-1}, 2^b] \times ... \times [1, 2^b]$. Atitinkamai P_{00} laukeliai yra pirmesni už P_{01}. Vėliau taip pat padalinama P_1 ir t.t. Šis rekursiškas dalinimas yra vykdomas tol, kol kiekviena erdvės sritis P_i yra nedidesnė negu gardelės laukelio dydis.



5 pav. Z erdvę užpildančios kreivės pavyzdys.

Formaliai tvarka yra apibrėžiama sąryšio:

$$p_1 < p_2 \text{ jeigu } ZCSFC(p_1) < ZCSFC(p_2) \tag{14}$$

Čia p_1 ir p_2 yra taškai erdvėje E^d, o $ZCSFC(p)$:

$$ZCSFC(p) = 1 + \sum_{i=0}^{b-1} \sum_{j=0}^{d-1} [2^{i*b+j} BITSET(p_j, i)] \tag{15}$$

$$BITSET(p, i) = \begin{cases} 1, & \text{jeigu i-asis bitas yra 1 skaičiaus p dvejetainėje formoje} \\ 0, & \text{kitu atveju} \end{cases} \tag{16}$$

3. SP-GIST karkasas

Walid G. Aref, Ihab F. Ilyas [AI01] pastebėjo, kad siekiant įdiegti ar palyginti įvairius hierarchinius duomenų indeksavimo metodus tenka kiekvieną jų atskirai integruoti į duomenų bazės branduolį. Tai yra sudėtinga ir daug laiko reikalaujanti užduotis. Autoriai šią problemą sprendė ieškodami bendrų savybių visiems hierarchiniams duomenų indeksavimo metodams ir pasiūlė sąsają, kuri galėtų būti integruota į duomenų bazės branduolį. Suprogramuoti indeksavimo metodą pagal šią sąsaja yra kur kas lengvesnė užduotis, negu integruoti tiesiai į duomenų bazės branduolį. Verta pastebėti, kad tokia pati problema panašiu būdu buvo spręsta ir vienmačių duomenų indeksavimo metodams [HNP95].

Autorių pasiūlyta SP-GIST sąsaja susideda iš 7 parametrų ir 3 metodų. Naudodamasis šiais parametrais ir metodais duomenų bazės branduolys gali išsaugoti, ištrinti įrašus bei apdoroti užklausas.

3.1. SP-GIST sąsajos parametrai

Tokie parametrai yra apibrėžiami autorių (pateikiami originalo kalba):

1. NodePredicate.
2. KeyType.
3. NumberOfSpacePartitions.
4. Resolution.
5. PathShrink.
6. NodeShrink.
7. BucketSize.

Trumpai aptarsime šiuos parametrus.

NodePredicate. Parametras *NodePredicate* apibrėžia duomenų struktūrą, kuri yra saugoma indekso viršūnėse. Pagal šiuos duomenis yra parenkamas indekso viršūnės vaikas, pro kurį bus iteruojama toliau. Pavyzdžiui Kd-medžio atveju šis parametras apibrėžtų duomenų tipą saugantį koordinačių ašį ir reikšmę, pagal kurią indekso viršūnėje dalijama erdvė E^d (žr.: 2.2 skyrelį).

KeyType. Parametras *KeyType* nurodo kokio tipo duomenys yra saugomi duomenų viršūnėse. Šiame darbe nagrinėsime tik tuos atvejus, kur šio parametro reikšmė yra aibė „aibė“ (erdvės E^d taškų aibė).

NumberOfSpacePartitions. Parametras *NumberOfSpacePartitions* apibrėžia kiek daugiausiai vaikų gali turėti indekso viršūnės. Pavyzdžiui Kd-medžio atveju šio parametro reikšmė yra 2, quad-medžio atveju – 4.

Resolution. Parametras *Resolution* apibrėžia maksimalų medžio gylį.

PathShrink. Parametras *PathShrink* gali turėti tris reikšmes: *NeverShrink*, *LeafShrink* arba *Tree Shrink*. Kai kuriais atvejais hierarchiniuose duomenų indeksavimo metoduose dvi ar daugiau indekso viršunių gali būti apjungtos tarpusavyje, šitaip sumažinant medžio gylį. Tačiau šiame darbe šie apjungimo metodai nebus nagrinėjami ir bus daroma prielaida, kad parametro *PathSrhink* reikšmė visada bus *NeverShrink*, t.y. niekada nebus vykdomi indekso viršūnių apjungimai.

NodeShrink. Verta prisiminti, kad hierarchiniai duomenų indeksavimo metodai duomenis laiko grupėmis (žr.: 3 paveiksliuką). Parametras *NodeShrink* gali turėti dvi reikšmes: „Įjungta“ arba „Išjungta“. Jeigu reikšmė yra „Įjungta“, tuomet tuščios duomenų viršūnės (pavyzdžiui po visų įrašų priklausančių kuriai nors duomenų viršūnei ištrynimo) bus pašalinamos iš medžio. Atitinkamai, jeigu indekso viršūnė nebeturi vaikų – ji irgi bus pašalinta. Jeigu reikšmė yra „Išjungta“, tuščios viršūnės yra paliekamos medyje.

BucketSize. *BucketSize* parametras nurodo kiek daugiausiai įrašų gali būti vienoje duomenų viršūnėje (grupėje).

3.2. SP-GIST sąsajos metodai

Autoriai apibrėžė šiuos metodus SP-GIST sąsajoje (pateikiami originalo kalba):

1. *Consistent(Entry E, Query q)*
2. *PickSplit(P)*
3. *Cluster()*

Metodas Consistent. Metodas *Consistent(Entry E, Query q)* yra būlio funkcija, kuri gražina rezultatą *Melas* tada ir tik tada, jeigu viršūnėje E (ir jos pomedyje jeigu viršūnė E yra indekso viršūnė) negali būti įrašų, kurie tenkina užklausą q. Šis metodas yra naudojamas duomenų bazės branduolio kaip kelrodis iteruojant per medį.

Metodas PickSplit. Jeigu duomenų viršūnėje, į kurią bandoma įdėti naują įrašą, nebėra laisvos vietos (įrašų skaičius joje pasiekė parametro *BucketSize* reikšmę), tuomet ši viršūnė yra skaidoma į dvi ar daugiau naujų viršūnių. Metodas *PickSplit(P)* yra naudojamas siekiant parinkti kaip duomenų viršūnė P pasidalins į mažesnes. Šis metodas gražina medį, saugantį visus įrašus buvusius viršūnėje P. Duomenų bazės branduolys, duomenų viršūnę P, pakeičia gražinto medžio šaknine viršūne.

Metodas Cluster. Metodas *Cluster()* parenka kaip medžio viršūnės bus saugomos disko blokuose. Kadangi šiame darbe nagrinėjame indeksus, kurie saugo įrašus operatyviojoje atmintyje, tai šis metodas nebus detalizuojamas.

Šis karkasas yra naudojamas populiariose duomenų bazėse [EEA06]. Taip pat modifikuota (konkrečios modifikacijos aprašomos sekančiuose skyreliuose) ši sąsaja bus taikoma ir vertinant biometrinių įrašų priskyrimo blokams metodus paremtus ir Kd-medžiu ir erdvę užpildančiomis kreivėmis sistemoje [Neu18].

Literatūra

[AI01] Walid G Aref ir Ihab F Ilyas. Sp-gist: an extensible database index for supporting space partitioning trees. *Journal of Intelligent Information Systems*, 17(2-3):215–240, 2001.

[Bad12] Michael Bader. *Space-filling curves: an introduction with applications in scientific computing*, tom. 9. Springer Science & Business Media, 2012.

[BBK01] Christian Böhm, Stefan Berchtold ir Daniel A Keim. Searching in high-dimensional spaces: index structures for improving the performance of multimedia databases. *ACM Computing Surveys (CSUR)*, 33(3):322–373, 2001.

[Ben79] Jon Louis Bentley. Multidimensional binary search trees in database applications. *IEEE Transactions on Software Engineering*, (4):333–340, 1979.

[BKK96] S Berchtold, DA Keim ir HP Kriegel. The x-tree: an index structure for high-dimensional data. proceedings of the 22nd international conference on very large data bases (vldb), bombay, india: 28-39. 1996.

[BKS+90] Norbert Beckmann, Hans-Peter Kriegel, Ralf Schneider ir Bernhard Seeger. The r*-tree: an efficient and robust access method for points and rectangles. *Acm Sigmod Record*, tom. 19 numeris 2, p.p. 322–331. Acm, 1990.

[BKS+94] Thomas Brinkhoff, Hans-Peter Kriegel, Ralf Schneider ir Bernhard Seeger. *Multi-step processing of spatial joins*, tom. 23 numeris 2. ACM, 1994.

[Com79] Douglas Comer. Ubiquitous b-tree. *ACM Computing Surveys (CSUR)*, 11(2):121–137, 1979.

[EEA06] Mohamed Y Eltabakh, Ramy Eltarras ir Walid G Aref. Space-partitioning trees in postgresql: realization and performance. *Data Engineering, 2006. ICDE'06. Proceedings of the 22nd International Conference on*, p.p. 100–100. IEEE, 2006.

[GG98] Volker Gaede ir Oliver Günther. Multidimensional access methods. *ACM Computing Surveys (CSUR)*, 30(2):170–231, 1998.

[Gut84] Antonin Guttman. *R-trees: A dynamic index structure for spatial searching*, tom. 14 numeris 2. ACM, 1984.

[GUW00] Hector Garcia-Molina, Jeffrey D Ullman ir Jennifer Widom. *Database system implementation*, tom. 654. Prentice Hall Upper Saddle River, NJ: 2000, p.p. 34–38.

[Hab83] Walter Habenicht. Quad trees, a datastructure for discrete vector optimization problems. *Essays and Surveys on Multiple Criteria Decision Making*, p.p. 136–145. Springer, 1983.

[HJ99] Karel Hrbacek ir Thomas Jech. *Introduction to Set Theory, Revised and Expanded*. Crc Press, 1999.

[HMC+12] Bijit Hore, Sharad Mehrotra, Mustafa Canim ir Murat Kantarcioglu. Secure multi-dimensional range queries over outsourced data. *The VLDB Journal*, 21(3):333–358, 2012.

[HNP95] Joseph M Hellerstein, Jeffrey F Naughton ir Avi Pfeffer. *Generalized search trees for database systems*. September, 1995.

[HOS97] Sabine Hanke, Th Ottmann ir Eljas Soisalon-Soininen. Relaxed balanced red-black trees. *Italian Conference on Algorithms and Complexity*, p.p. 193–204. Springer, 1997.

[LKG+03] Xin Li, Young Jin Kim, Ramesh Govindan ir Wei Hong. Multi-dimensional range queries in sensor networks. *Proceedings of the 1st international conference on Embedded networked sensor systems*, p.p. 63–75. ACM, 2003.

[LO93] Hongjun Lu ir Beng Chin Ooi. Spatial indexing: past and future. *IEEE Data Eng. Bull.*, 16(3):16–21, 1993.

[Mar00] Patrick Marcel. Modeling and querying multidimensional databases: an overview. *Networking and Information Systems Journal*, 2(5/6):515–548, 2000.

[MR03] Daniel D McCracken ir Edwin D Reilly. Backus-naur form (bnf), 2003.

[Neu18] UAB Neurotechnology. Megamatcher accelerator solution for large-scale multi-biometric systems. `http://download.neurotechnology.com/MegaMatcher_Accelerator_Brochure_2018-07-26.pdf`, 2018. tikrinta 2018-09-11.

[NHS81] Jürg Nievergelt, Hans Hinterberger ir Kenneth C Sevcik. The grid file: an adaptable, symmetric multi-key file structure. *Conference of the European Cooperation in Informatics*, p.p. 236–251. Springer, 1981.

[RMF+00] Frank Ramsak, Volker Markl, Robert Fenk, Martin Zirkel, Klaus Elhardt ir Rudolf Bayer. Integrating the ub-tree into a database system kernel. *VLDB*, tom. 2000, p.p. 263–272, 2000.

[Rob81] John T Robinson. The kdb-tree: a search structure for large multidimensional dynamic indexes. *Proceedings of the 1981 ACM SIGMOD international conference on Management of data*, p.p. 10–18. ACM, 1981.

[SH08] Chanop Silpa-Anan ir Richard Hartley. Optimised kd-trees for fast image descriptor matching. *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, p.p. 1–8. IEEE, 2008.

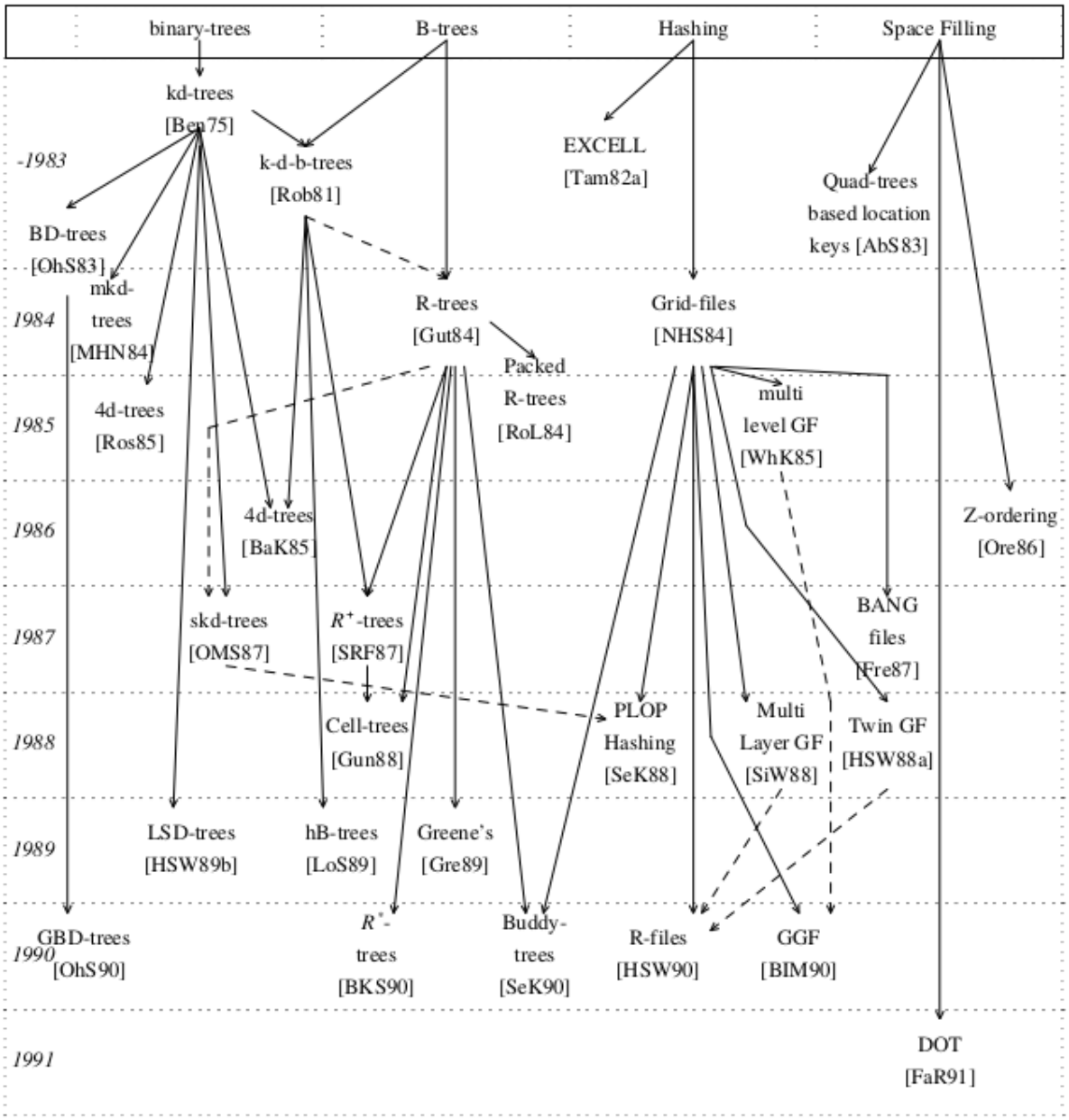
[SRF87] Timos Sellis, Nick Roussopoulos ir Christos Faloutsos. The R+-Tree: A Dynamic Index for Multi-Dimensional Objects. Tech. atask., 1987.

[TS82] Markku Tamminen ir Reijo Sulonen. The excell method for efficient geometric access to data. *Design Automation, 1982. 19th Conference on*, p.p. 345–351. IEEE, 1982.

[XXY10] Yonghui Xiao, Li Xiong ir Chun Yuan. Differentially private data release through multidimensional partitioning. *Workshop on Secure Data Management*, p.p. 150–168. Springer, 2010.

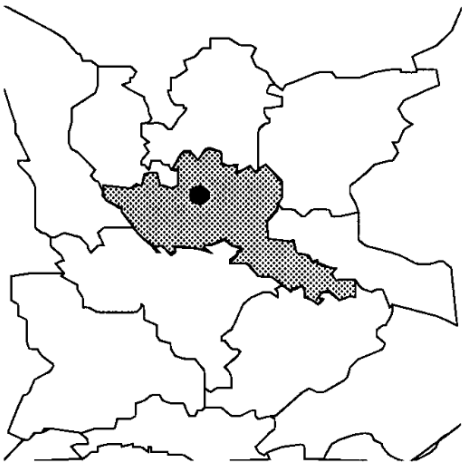
Priedas Nr. 1

Metodų skirtų daugiamačių duomenų indeksavimui apžvalga



6 pav. [Bad12] pateikiama metodų, skirtų daugiamačių duomenų indeksavimui, schema

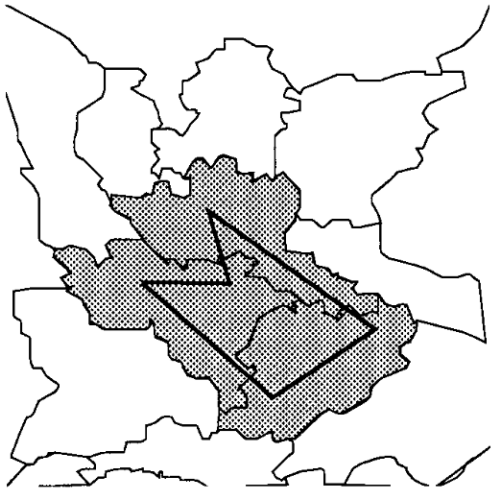
Priedas Nr. 2
Įvairių užklausų pavyzdžiai



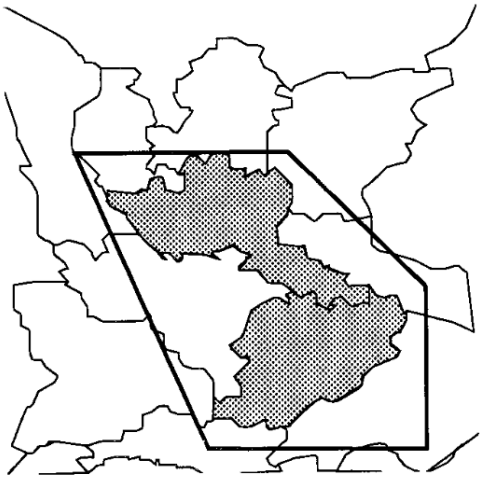
7 pav. Taško užklausos pavyzdys

8 pav. Lango užklausos pavyzdys

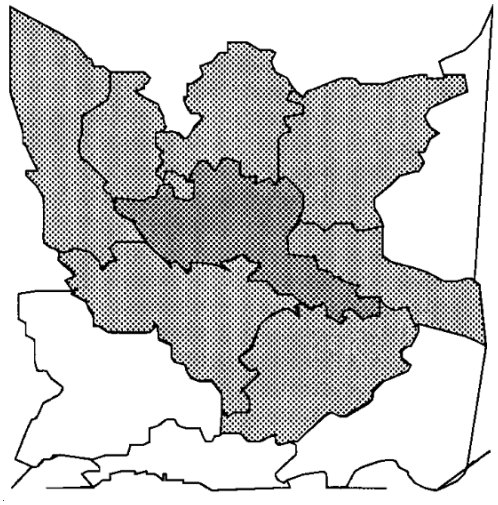
9 pav. Apgaubiančiosios užklausos pavyzdys



10 pav. Regiono užklausos pavyzdys



11 pav. Pilno regiono užklausos pavyzdys



12 pav. Kaimynų užklausos pavyzdys