KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
KOMPIUTERIŲ KATEDRA

Mindaugas Surdokas

Nestruktūrizuotų duomenų modelio sudarymas ir tyrimas

Magistro darbas

Darbo vadovas
Doc. dr. V. Kiauleikis

Kaunas
2006

CONSTRUCTION AND ANALYSIS OF UNSTRUCTURED DATA MODEL. SUMMARY

The amount of information on the internet is increasing every day and it is very difficult to find exact information from big list of search results. Users typically generate query specifying *keywords* and the search engine displays documents, what contains keywords what were set by user.

Search within *unstructured data* starts from data processing: first of all documents are braked into sentences, then words and word groups are analyzed semantically and syntactically, to obtain *facts*. Facts describe objects of real world.

Facts obtained from unstructured data will be stored into database, and unstructured data will be transformed into structured data. Afterwards it will be easy to analyze, conjunct, filter or do other manipulations with structured data.

The goal of this paper is analysis of unstructured data for processing uncertain data, the creation of the uncertain data model.

TURINYS

LENTELIŲ SĄRAŠAS

PAVEIKSLĖLIŲ SĄRAŠAS

ĮVADAS

Apie du trečdalius žmogaus veikloje naudojamos informacijos yra pateikiama *nestruktūrizuota forma:* naujienų straipsniai internete, ataskaitos, elektroninio pašto pranešimai, komentarai. Nestruktūrizuotuose duomenyse informacija yra pateikiama bet kokia tvarka ar pavidalu, naudojant įvairias gramatines formas, kalbos dalis, dažnai duomenys nėra tikslūs ar aiškiai apibrėžti. Norint atlikti paiešką ar kitas operacijas tokiuose duomenyse – pirmiausia juos reikia apdoroti pervedant į struktūrizuotą formą.

Šiam darbui keliamas tikslas – ištirti nestruktūrizuotus duomenis, sudaryti duomenų modelį nestruktūrizuotiems duomenims saugoti, pasiūlyti algoritmą tokiems duomenims struktūrizuoti.

Nestruktūrizuotuose duomenyse gausu neapibrėžtumų, susijusių su informacijos trūkumu, netikslumu bei neaiškumu: duomenų nepilnumas, netikslumas, prieštaringumas, neatitikimas ir t.t. Kad būtų galima sukurti neapibrėžtų duomenų modelį, pirmiausia reikia išnagrinėti neapibrėžtumus, nustatyti jų savybes. Išanalizavus ir formalizavus įvairius neapibrėžtumų atvejus, galima pereiti prie nestruktūrizuotų duomenų turinio analizės. Neapibrėžta informacija dažniausiai yra gaunama kartu su tekstiniais dokumentais, elektroniniais laiškais, atsiunčiama iš interneto ir pan. Bandant iš tokių duomenų išgauti informaciją ir ją patalpinti į taip vadinamus *teiginius,* naudojamos įvairios lingvistinės priemonės. Atlikus nestruktūrizuotų duomenų turinio analizę ir aptarus kaip suformuoti teiginius, sudaromas nestruktūrizuotų duomenų modelis bei aprašomos operacijos su teiginiais. Eksperimentinėje dalyje aprašomas sukurto modelio panaudojimas nestruktūrizuotos informacijos apdorojimo uždavinyje, pateikiami eksperimento rezultatai.

Pirmame skyriuje yra aptariami duomenų modeliai, šiuolaikinės duomenų bazių valdymo sistemos, neapibrėžti sąryšiniai duomenys, analizuojamos duomenų nepilnumo, neišbaigtumo priežastys, išskiriant tokius aspektus, kaip netikslumas, neatitikimas bei netikrumas. Antrame skyriuje sintaksiniu požiūriu yra analizuojamas nestruktūrizuotų duomenų turinys, suformuluojama teiginio sąvoka bei išanalizuojami teiginį sudarantys komponentai.

Trečiame skyriuje sudaromas ir aptariamas nestruktūrizuotų duomenų modelis, pasiūlomas algoritmas nestruktūrizuotiems duomenims apdoroti. Ketvirtajame skyriuje aprašomos operacijos su teiginiais, o penktame skyriuje eksperimentiškai išbandomas nestruktūrizuotų duomenų modelis, pateikiami rezultatai.

1 ANALITINĖ APŽVALGA

Šiuo metu pasaulyje yra sukurta ir naudojama labai daug įvairiausio tipo, struktūros, apimties duomenų bazių, saugojančių skirtingo tipo duomenis. Sparčiai besivystant šiuolaikinėms technologijoms, tobulėjant duomenų surinkimo, apdorojimo ir saugojimo technikai, sparčiai plečiasi ir pačios duomenų bazės. Pavyzdžiui, telekomunikacijos paslaugų tiekėjams reikia terabaitais matuojamų duomenų bazių, kurios saugotų duomenis apie vartotojų skambučius, mokėjimo informaciją ir panašiai. Be to, daugumai duomenų bazių pradedami kelti ir kiti reikalavimai, pavyzdžiui, saugoti multimedijos failus: paveikslėlius, muziką ar vaizdo įrašus.

1.1 Duomenų apdorojimas

Duomenų apdorojimas *(data processing)* – tai procesas kurio metu informacija, duomenys (skaičiai, žodžiai) yra transformuojama į struktūrizuotus duomenis.

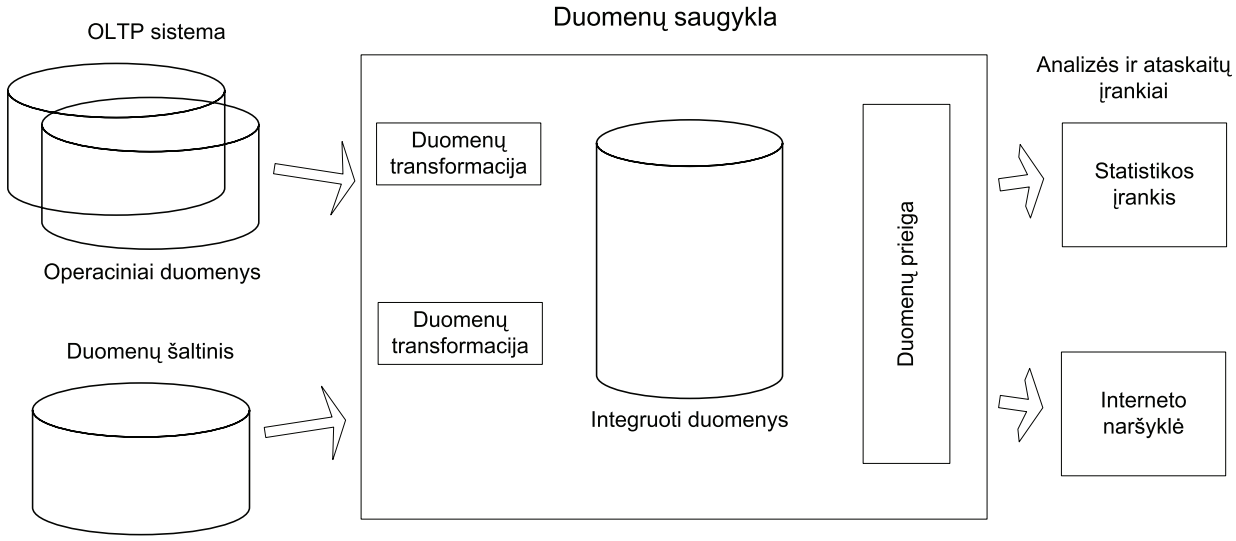
Duomenų bazės valdymo sistema

Duomenų bazės valdymo sistema DBVS *(Database management system – DBMS)* yra skirta atlikti duomenų saugojimo bei įvairių veiksmų su duomenimis funkcijas. Šiuolaikinėm DBVS įtaką daro ekonominiai bei technologiniai pasikeitimai, pavyzdžiui, kompiuterinių tinklų vystimasis. Tradicinės DBVS, pavyzdžiui, bankinės aplikacijos, reikalauja greito daugelio vartotojų užklausų apdorojimo, duomenis imant iš didelių, dinaminių duomenų bazių. Šiems uždaviniams įvykdyti, tradicinės DBVS naudoja tiesioginį transakcijų apdorojimo modelį *(On-Line Transaction Processing – OLTP),* transakcijas naudodamos kaip bazinį mechanizmą siekiant užtikrinti duomenų tikrumą ir teisingumą [7]. Duomenys dažniausiai būna griežtai struktūrizuoti. Tačiau naujose duomenų bazių valdymo sistemose vis dažniau atsiranda poreikis saugoti mažiau struktūrizuotus arba visai nestruktūrizuotus duomenis.

Duomenų saugykla

Duomenų saugykla *(data warehouse)* skirta išgauti duomenims, saugojamiems skirtinguose duomenų šaltiniuose [5], [7]. Pavyzdžiui, standartinėje įmonėje egzistuoja daug įvairių aplikacijų, kurios kaupia duomenis apie tam tikras veiklos rūšis ir su jomis chronologiškai susijusius duomenis. Šios aplikacijos galėtų būti transakcijų apdorojimo sistemomis, kurias naudoja tradicinės DBVS, arba jos galėtų būti naudojamos kaip specializuotos aplikacijos,

kaupiančios duomenis tam tikruose duomenų failuose. Duomenys, naudojami šiose skirtingose aplikacijose, yra vertingi, nes saugoja informaciją apie praeityje priimtus sprendimus, ir bus naudingi ateityje, nes padės priimti sprendimus ateityje. Norint atlikti šias užduotis, duomenų saugykla turi sujungti duomenis įvairiais būdais taip, kad ateityje būtų galima lengvai pasiekti ir analizuoti duomenis, priimti atitinkamus sprendimus.



1 pav. Duomenų saugyklos schema

Skaitmeninė biblioteka

Skaitmeninė biblioteka yra elektroninė klasikinės bibliotekos versija, kurioje informacijos resursai (pavyzdžiui, knygos, meno kūriniai, filmai) ir katalogo informacija, aprašanti resursus, yra saugomi skaitmeniniu būdu. Skaitmeninėje bibliotekoje turi būti numatyta galimybė saugoti bei atlikti tam tikrus veiksmus su įvairialypiais duomenimis: nestruktūrizuotais duomenimis (paveiksliukai, vaizdo įrašai), pusiau struktūrizuotais duomenimis (hiperteksto dokumentai) ir struktūrizuotais duomenimis (aprašantieji metaduomenys). Skaitmeninės bibliotekos papildo savo resursus dvejopai: gauta informacija iš struktūrizuotų duomenų bazių šaltinių ir nestruktūrizuota informacija.

Statistinės ir mokslinės duomenų bazių valdymo sistemos

Statistinės DBVS yra suprojektuotos operuoti sociologiniais – ekonominiais duomenų rinkiniais (pavyzdžiui, gyventojų visuotinio surašymo ar ekonominio prognozavimo duomenys). Mokslinės DBVS dažniausiai operuoja kompleksiniais duomenimis, gautais įvairių mokslinių eksperimentų metu. OLAP statistinės ir mokslinės DBVS atveju turi turėti duomenų peržiūros

įvairiais pjūviais, duomenų analizės bei ataskaitų generavimo funkcijas. Priešingu atveju, tai turi būti realizuota skirtingų kompleksinių duomenų sąjungomis, įskaitant ne tik skaitinius ar tekstinius duomenis, bet ir kompleksinius duomenų tipus. Šie tipai galėtų atvaizduoti tokius objektus kaip molekulinės struktūros, žemės paviršiaus žemėlapiai ar architektūriniai planai.

Pasaulinio žiniatinklio duomenų bazės

DBVS naudojimas pasaulinio žiniatinklio *(World-Wide-Web* – WWW) turiniui saugoti parodė dinaminių žiniatinklio serverių naudojimo efektyvumą. Pradėjus naudoti CGI *(Common gateway interface)* ir panašias sąsajas, žiniatinklio aplikacijų programos gali ne tik parinkti ir vartotojui pateikti reikiamą statinį puslapį, bet ir sugeneruoti tikslius užklausos rezultatus atitinkantį dinaminį puslapį. DBVS gali būti naudojama interneto svetainių turiniui saugoti, žiniatinklis taip pat leidžia elektroninį egzistuojančių duomenų bazių publikavimą. Publikuotų duomenų bazių vartotojai, priešingai nei tradicinių DBVS vartotojai, dažniausiai net nežino duomenų bazės, kurios duomenimis naudojasi, struktūros. Taigi, vartotojai net nemokėdami efektyviai suformuoti užklausų, gali ieškoti duomenų, saugomų kompleksinėse duomenų bazėse.

1.2 Duomenų modeliai

1.2.1 Struktūrizuoti failai

Struktūrizuotuose (plokštuminiuose) failuose *(flat file)* informacija saugoma įrašuose *(record)*. Struktūrizuotuose duomenų failuose [5] įrašai išdėliojami vienas po kito, sekoje. Kiekvienas įrašas gali susidaryti iš vieno arba keleto laukų.

Raktas	Vardas	Pavardė	Amžius

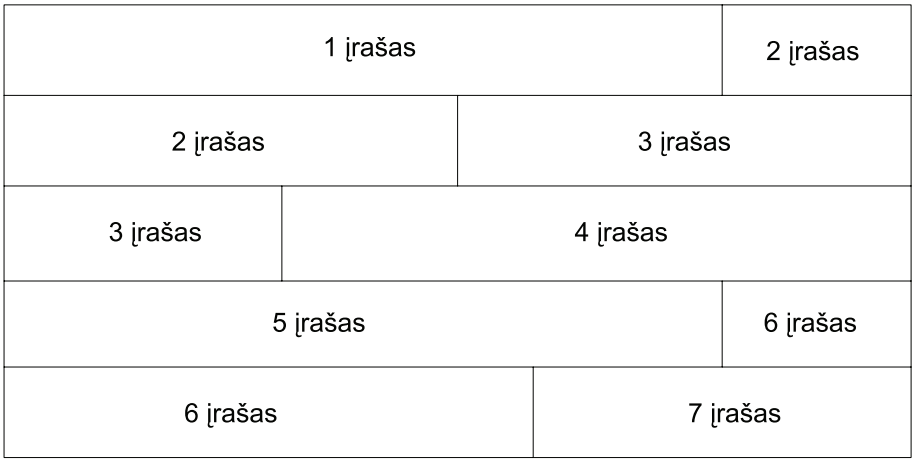
2 pav. Įrašo pavyzdys

Struktūrizuotų duomenų failų laukuose gali būti saugomi bet kokio tipo duomenys, laukai neturi pavadinimų. Įrašų ir laukų dydis yra ribojamas, tačiau atskiri laukai gali būti skirtingo dydžio (neviršinant maksimalaus nustatyto dydžio). Struktūrizuoti duomenų failai gali būti saugomi tekstiniu ar dvejetainiu formatu.

Būtinybės, kad įrašai būtų tokio paties duomenų tipo, struktūrinių failų atveju nėra (t.y. nėra būtinybės, kad visi laukai įraše būtų vienodų tipų, kad išlaikytų eiliškumą visuose įrašuose). Tačiau jei dirbdami su struktūriniais duomenų failais laikysimės tam tikros bendro tvarkos, tai

tuomet drąsiai galima teigti kad nesudėtiniems duomenims saugoti gali pakakti ir struktūrinių duomenų modelio.

Struktūrizuotuose duomenų failuose įrašų išdėstymo tvarka yra svarbi. Įrašo raktas *(key)* paprastai skirtas atskirti įrašus vieną nuo kito, nustatyti įrašo buvimo vietą faile. Įrašai – dublikatai (su ta pačia informacija) yra leidžiami.

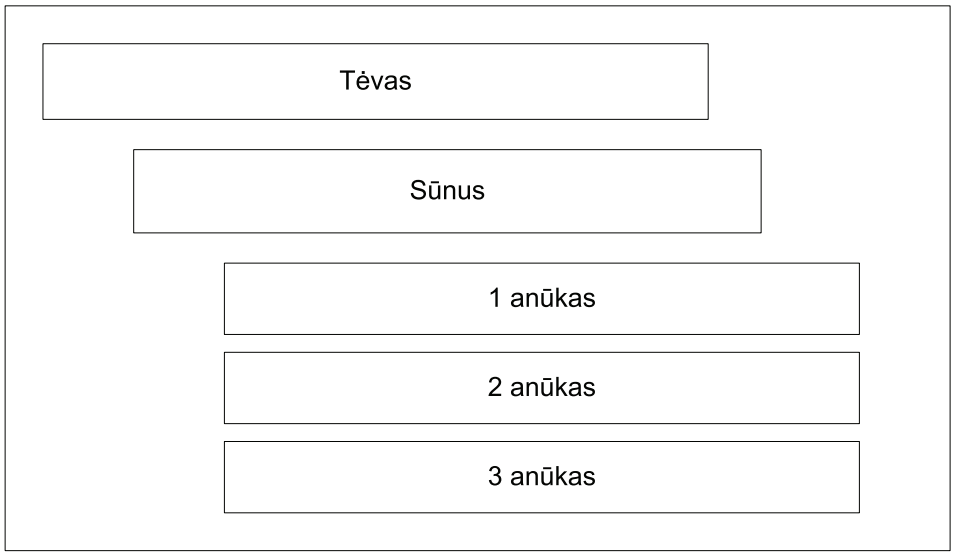


3 pav. Struktūrinio duomenų failo schema

Norint nustatyti, kur baigiasi vienas įrašas ir kur prasideda kitas, tarp įrašų yra naudojami tam tikri skyrikliai. Paprasto struktūrinio failo pavyzdžiu galėtų būti tekstinis failas, kurio kiekvienoje atskiroje eilutėje patalpinta informacija yra įrašas, o įrašus vieną nuo kito skiria žymeklis *eilutės pabaiga*. Kiti struktūrizuotų failų pavyzdžiai: prekių sąrašas, telefonų knyga.

1.2.2 Hierarchiniai failai

Struktūrizuotuose duomenų failuose įrašai yra viename ir tame pačiame lygmenyje. Hierarchiniai *(hierarchical)* duomenų failai leidžia įrašus grupuoti į grupes. Tai leidžia atsirasti *tėvas-sūnus* arba *vadovas-pavaldinys* tipo sąryšiams (sąryšio tipas – *vienas su daug*). Paprastose formose aukštesnio lygmens įrašas saugoja informaciją, bendrą visiems žemesniojo lygmens įrašams. Tai padeda sumažinti saugojamos informacijos apimtis. Žemesnio lygmens raktas paprastai paveldi visas aukštesnio lygio savybes.

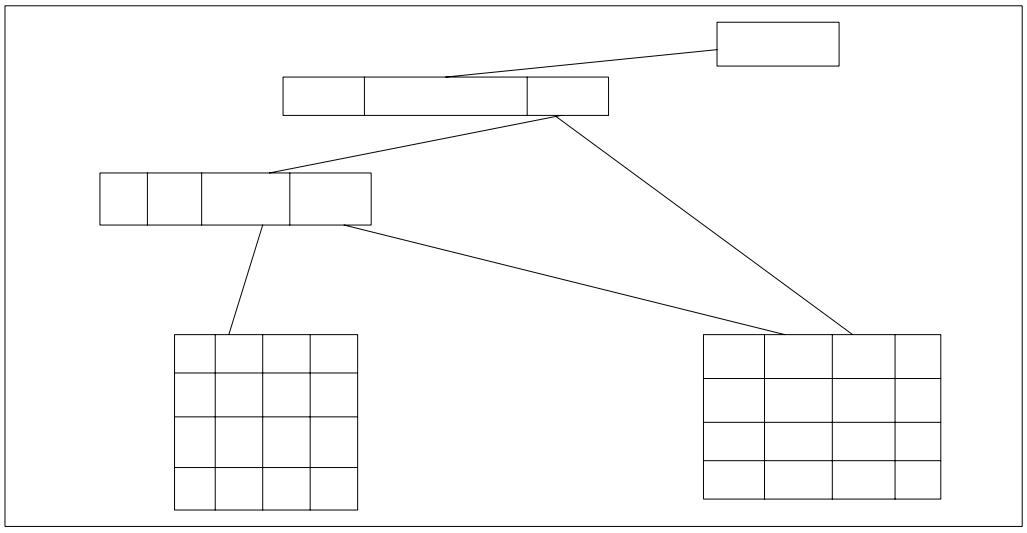


4 pav. Hierarchinių duomenų struktūra

Pavyzdžiai: įmonės padalinių telefonų duomenų bazė, Microsoft Windows registrai, šio darbo turinys.

1.2.3 Tinklinis duomenų modelis

Hierarchinis duomenų modelis aukštesnio lygmens įrašams leidžia naudoti vieną ryšį žemesnio lygmens įrašams pasiekti. Tinklinis *(network)* duomenų modelis aukštesnio lygmens įrašams leidžia naudoti keletą ryšių žemesnio lygmens įrašams ar įrašų aibėms pasiekti.



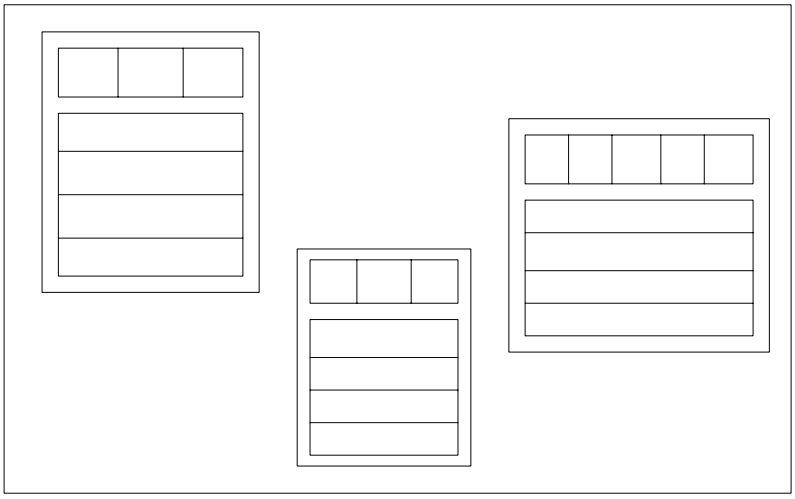
5 pav. Tinklinis duomenų modelis

Įrašai ir laukai yra fiksuoto dydžio, bet gali būti saugomi skirtingose duomenų failo (ar failų) vietose. Unikalus raktas nusako fizinę vietą (pavyzdžiui, failą, bloką ar šaką) duomenų bazės struktūroje.

Tinklinės duomenų bazės leidžia, kad įrašas turėtų kelis ryšius ar priklausytų keletui aibių. Šio tipo duomenų bazės yra labai efektyvios. Apdorojant užklausas gerai suprojektuotoje tinklinėje duomenų bazėje bus galima praleisti atšakas, kurių viršutinių lygmenų raktiniai žodžiai netenkina užklausos. Neanalizuojant visų įrašų ar jų aibių iš eilės, sutaupoma daug laiko. Dėl didelio efektyvumo, šis duomenų modelis dažnai taikomas duomenų bazių valdymo sistemose.

1.2.4 Sąryšinis duomenų modelis

Prieš atsirandant sąryšiniams duomenų modeliams, tuo metu buvę duomenų modeliai neturėjo galimybės atskirti koncepcinio modelio nuo realizacijos. Tuo metu egzistavusieji modeliai griežtai turėdavo apibrėžti kur ir kaip tam tikri įrašai turi būti saugojami. Pirmieji sąryšinių duomenų modelių šalininkai teigė, kad sąryšiniai duomenų modeliai informaciją peržiūri tiktai logiškai, o ne fiziškai, bet jie nebuvo visiškai teisūs. Pirmieji duomenų modeliai susiedavo loginius ir fizinius informacijos aspektus, o logiškai susieta informacija buvo saugojama arti duomenų failo (fiziniu požiūriu). Sąryšiniai duomenų modeliai visų pirma atskiria loginius aspektus nuo fizinių.



6 pav. Sąryšinis duomenų modelis

Sąryšiniai duomenų modeliai į informaciją žvelgia kaip į nesutvarkytą ryšių visumą. Kiekvienas ryšys yra patalpinamas kartu su kortežais ir atributais tokia pačia tvarka. Laukų reikšmė gali būti arba žinomos (aprašytos) struktūros arba tuščia. Norint geriau aprašyti

duomenis, ryšiai yra atvaizduojami kaip lentelės. *Kortežai* sudaro lentelės eilutes, *atributai* suprantami kaip stulpeliai. Ryšiai abstrakčiai atvaizduoja, kaip informacijas yra saugoma, o lentelės – tik vienas iš galimų atvaizdavimo būdų.

Skirtumai tarp kortežų yra vos pastebimi, nes jie nėra apibrėžiami. Priešingai nei hierarchinio ar tinklinio duomenų modelio atveju, kortežai vienas nuo kito yra atskiriami atsižvelgiant į jų turinį. Atitinkamai, esant vienam ir tam pačiam ryšiui, besikartojantys kortežai nėra galimi, be to, visi kortežai turi turėti savo unikalų raktą *(key)*.

Duomenims apdoroti naudojamos matematinės sąryšių algebros operacijos, pavyzdžiui, apjungimas *(union)*, sankirta *(intersection),* sujungimas *(join),* projektavimas *(projection)* ir kt.

1.3 Šiuolaikinės DBVS

Šiuolaikinės DBVS naudoja tokius struktūrizuotus duomenų modelius, kaip *sąryšiniai, hierarchiniai* ar *objektiniai* modeliai [6], [7]. Struktūrizuoti duomenų modeliai leidžia duomenis grupuoti į lentelių aibes arba klases, kiekviena kurių turi tam tikrą struktūrą ar schemą. Siekiant prisitaikyti prie naujai atsirandančių aplikacijų poreikių, duomenų modeliai buvo plečiami trimis kryptimis:

- tiesioginis abstrakčių duomenų tipų palaikymas;
- koncepcinių struktūrų pridėjimas, kad būtų pagerinta didelių kompleksinių duomenų peržiūra bei sumarizavimas;
- nestruktūrizuotų bei dalinai struktūrizuotų duomenų palaikymas.

Aptarsime kiekvieną duomenų modelio praplėtimo variantą.

1.3.1 Abstraktus duomenų tipas

Tradicinės DBVS palaiko fiksuotus paprastų duomenų tipų rinkinius (pavyzdžiui, skaičiai, datos). DBVS gali būti dinamiškai praplėsta vartotojo apibrėžtais duomenų tipais bei funkcijomis. Šie tipai gali būti naudojami kompleksiniams objektams modeliuoti, pavyzdžiui, molekulinei struktūrai aprašyti. Dauguma komercinių DBVS (pavyzdžiui, DB2, Informix ar Oracle) palaiko šią galimybę [9]. Siekiant pilnai palaikyti šiuos naujus duomenų tipus, DBVS turi būti numatytas duomenų valdymo mechanizmas, nauji indeksavimo bei užklausų apdorojimo metodai.

1.3.2 Multidimensiniai modeliai

Duomenų saugojimui OLAP bei kitos statistinės aplikacijos duomenis saugo keliomis dimensijomis. Pavyzdžiui, gyventojų surašymo duomenys gali būti matomi keliomis skirtingomis dimensijomis: amžiaus, lyties, profesijos ar gyvenamosios vietos ir kt. Prekes aprašantys duomenys gali būti atvaizduojami šiomis dimensijomis: gamintojas, prekės tipas, pagaminimo data, kaina ir kt. Šiuose pavyzdžiuose duomenys, nusakantys gyventojų surašymo ar prekių parametrus, yra vadinami *teiginių lentelėmis.* Toks multidimensinis duomenų atvaizdavimas palengvina tiesioginį duomenų modeliavimą. Pavyzdžiui, prekių pagaminimo data gali būti aprašyta sekančiomis subdimensijomis: diena, mėnuo, metai. Egzistuoja funkciniai ryšiai tarp produkcijos pagaminimo datos bei kiekvienos iš subdimensijų. Multidimensinis modelis taip pat padeda išsireikšti ar sumarizuoti agregatus iš skirtingų duomenų dimensijų (ar subdimensijų) pagal vartotojo poreikius. Pavyzdžiui, vartotojas gali norėti sužinoti, kiek kiekviename Kauno mikrorajone yra tam tikros specialybės žmonių, vyresnių nei 35 metai.

1.3.3 Nestruktūrizuotų duomenų modelis

Tradicinės informacijos išgavimo *(Information Retrieval – IR)* sistemos duomenims saugoti naudoja nestruktūrizuotus duomenų modelius. Duomenys, saugojami dokumentuose, yra skirtingų tipų bei struktūros. Dokumentai gali būti paveikslėliai, vaizdo informacija arba nestruktūrizuoti tekstai, saugojami bet kokiu formatu. Kiekvienas dokumentas yra modeliuojamas kaip atskirų žodžių rinkinys (tekstinio dokumento atveju tai būtų žodžių, sudarančių dokumentą, aibė; paveikslėlio ar vaizdo informacijos atveju tai gali būti tekstas, apibūdinantis pateiktą vaizdą). Su šiais žodžiais nėra susieta jokia struktūra, ir jei dokumente bus paminėtas žodis Kęstutis, tai duomenų modelis neįtraukia tokios informacijos, ar Kęstutis yra šio dokumento autorius, veikėjas, ar gatvės pavadinimas, nusakantis veiksmo vykimo vietą.

Nestruktūrizuotų duomenų modelis yra naudojamas tokiems duomenims, kuriems negalima pritaikyti kokios nors struktūrizuotos formos. Tačiau nestruktūrizuotų duomenų modelis negali užtikrinti pakankamo funkcionalumo, kai yra naudojamas informacija, turinti kažkokią struktūrą (tarkim, žiniatinklio puslapiai). Paimkite kaip pavyzdį XML dokumentą, kuris gali turėti žymes *(tag),* nusakančias autorių, sukūrimo datą ir dokumento pavadinimą bei didelį nestruktūrizuotų duomenų kiekį (dokumento tekstinė, turinio dalis).

Naudojant nestruktūrizuotų duomenų modelį, galima suformuoti tokią užklausą, kuri rastų dokumentus, kurių autorius būtų *Kęstutis*. Naudojant struktūrinį duomenų modelį atvaizduoti šią informaciją yra labai nepatogu, nes žiniatinklio dokumentai yra tos pačios struktūros. Geriausiu atveju galima būtų apibrėžti atributų lenteles, tinkančias daugumai dokumentų. Tokiems duomenims aprašyti reikia naudoti iš dalies struktūrizuotų duomenų modelį. Šie modeliai dažniausiai patys automatiškai aprašo duomenų modelius, kuriuose duomenų objektas aprašytas kiekybiškai bei struktūriškai. Be abejo, kiekvienas objektas gali turėti savitą unikalią struktūrą, be to ne retai objektai gali būti susieti su kitais objektais, o tam aprašyti naudojami grafai.

1.4 Užklausos

Struktūrinėse užklausų kalbose (pavyzdžiui, SQL, OQL) schemų komponentai (atributai, ryšiai ar klasių pavadinimai) yra naudojami duomenų tipui aprašyti. Taigi, vartotojas turi žinoti ir suprasti schemą, kad galėtų suformuluoti klausimus [14]. Nestruktūrizuotuose duomenų modeliuose, užklausos modelis remiasi raktiniais žodžiais. Loginių kombinacijų seka, kurią vartotojas sudarė naudodamas raktinius žodžius, yra priešpastatoma žodžių sekai, saugomai dokumente. Norint efektyviai apdoroti užklausas, raktiniai žodžiai yra indeksuojami. Sudėtingi algoritmai yra naudojami užtikrinti, jog visi panašūs dokumentai būtų įtraukti, o nepanašūs dokumentai – atmesti. Šie algoritmai taip pat apima lingvistinius metodus raktinių žodžių sinonimams atrasti.

Užklausų apdorojimo kalba dalinai struktūrizuotiems duomenims leidžia specifikuoti struktūrines užklausas duomenų objektams su aprašyta duomenų struktūra. Tačiau, jei laikysime, kad kiekvienas objektas turi savo duomenų struktūrą, suprasti visos duomenų bazės struktūrą kartais visai ne lengva, o kartais net neįmanoma. Duomenų saugykloje, kai duomenų struktūra yra ypatingai sudėtinga, vartotojas turi turėti teisę užduoti klausimus, pilnai nežinant pačios duomenų bazės struktūros. Taigi, dalinai struktūrizuotų duomenų užklausų apdorojimo kalbos leidžia suformuoti užklausas palyginant su šablonu (pavyzdžiui, „Rasti visus penkiaaukščius Vilniaus miesto daugiabučius namus, kurių bendrijos narių skaičius didesnis už 50“). Tokio pobūdžio užklausa leidžia ieškoti duomenų nežinomos arba dalinai žinomos struktūros duomenų bazėse.

Heterogeninėms struktūroms apdoroti taip pat tinka palyginančios su šablonu užklausos. Multi-duomenų bazės leidžia papildomų duomenų restruktūrizavimą ir operacijų sujungimą duomenų integracijos palengvinimui. Meta duomenys (aprašantys duomenų bazės schemas ir kitą susijusią informaciją) gali žymiai pagerinti heterogeninių duomenų šaltinių integraciją.

Siekdamos iš duomenų saugyklos išgauti reikiamą informaciją, duomenų saugyklos gali atlikti agregavimą ir sumarizavimą [1]. Agregavimo funkcija tipiškai apima bazines SQL funkcijas (ir objektais pagrįstus SQL variantus) skaičių, vidurkių, sumų, maksimalių ir minimalių reikšmių radimui, taip pat ir sudėtingesnes skaičių apdorojimo statistines funkcijas. Kai kurios DBVS leidžia vartotojui pačiam aprašyti naujas agregavimo funkcijas. Sumarizavimas išplečia paprastą horizontalųjį skaidymą SQL *„grupuoti pagal“* operatoriumi. *„Grupuoti pagal“ (group by)* operatorius skaido lentelę (ar klasės elementus) į grupes, suskirstytas pagal atributų aibių reikšmes. Kiekvienai grupei yra pritaikoma agregavimo funkcija, kad būtų apskaičiuota kiekvienos atributų aibės reikšmė. *Kubo* operatorius naudojamas atlikti paskaičiavimus išilgai visos lentelės. Priešingai nei *„grupuoti pagal“* operacija, paskaičiavimai, atliekami išilgai visos lentelės leidžia atlikti kiekvienos sub-aibės atributų sub-sumas.

Aptariami metodai leidžia patogiai įvesti duomenis naujiems vartotojams, nesunkiai peržiūrėti jam aktualius duomenis. Šiuos metodus galime grubiai sugrupuoti į dvi strategijas. Pirmoji naudoja koncepcinę klasifikaciją (nuo bibliotekos iki loginės duomenų bazės): dokumentai yra asocijuoti koncepcijomis (pavyzdžiui, agrokultūra, ekonomika ar suvirinimas). Koncepcijos yra susijusios viena su kita semantiniais ryšiais. Vartotojas gali peržiūrėti su kiekvienu dokumentu susijusias koncepcijas. Antra duomenų peržiūros strategija naudoja OLAP tipo duomenų bazės sumarizavimą vartotojo pageidaujamų duomenų peržiūrėjimui. Šis mechanizmas grupuoja tarpusavyje duomenų bazės sub-aibes ir atvaizduoja kiekvienos grupės duomenų elementų agregatus.

1.5 Duomenų neapibrėžtumas

Duomenų neapibrėžtumas *(uncertainty)* – tai būsena, kai objektui nusakyti trūksta informacijos. Duomenų neapibrėžtumas atsiranda dėl šių priežasčių: informacijos šaltinio patikimumo (patikimas, nepatikimas, patikimas iš dalies ir pan.), nusakančio kiek galima tikėti informacija, gauta iš šio šaltinio bei gautos informacijos patikimumo (galbūt, ko gero, manau ir pan.), parodančio kiek tikras ar galimas yra teiginys. Neapibrėžtumo įvertinimui yra skirta

nemažai darbų: [3], [4], [13], kuriuose neapibrėžtumas yra vertinamas statistiniais metodais. Tačiau prieš apdorojant duomenis statistiškai, pirmiausia reikia sukurti duomenų modelį, galintį kaupti ir apdoroti neapibrėžtą informaciją.

Dirbant su netiksliais duomenimis *(imperfect data),* nuolatos susiduriama su problema – kaip perprasti įvairius duomenų netikslumo ir neapibrėžtumo aspektus. Informacija yra patikima, kai ji yra aiškiai ir konkrečiai apibrėžta (išreikšta). Duomenų netikslumas atsiranda dėl duomenų *nepakankamumo*, kai informacija nėra aiški, tiksli ar konkreti arba kai jos nepakanka užduotam tikslui pasiekti; *netikslumo*, kai informacija apie faktą išreikšta netiksliai arba neatitinka tam tikrų normų, standartų; *prieštaringumo,* kai iš skirtingų šaltinių gauti faktai apie tą patį objektą prieštarauja vienas kitam; *neaiškumo*, kai informacija yra išreikšta neaiškiai ir gali būti suprasta skirtingai.

Kai informacija yra išreikšta tam tikromis lingvistinėmis formomis, jas konvertuojant į skaitines išraiškas atsiranda ir lingvistinis neapibrėžtumas.

Paanalizuokime pavyzdį:

- Jonas turi mažiausiai du vaikus ir aš dėl to esu tikras.
- Jonas turi tris vaikus, bet aš dėl to nesu tikras.

Pirmu atveju vaikų kiekis netikslus, bet patikimas. Antru atveju vaikų kiekis yra tikslus, bet nepatikimas.

1.5.1 Netikslumas

Teiginys „maistas yra degantis“ *nevienareikšmis (ambiguous)*, nes maistas gali būti aštrus, arba karštas.

„Jam – virš trisdešimt metų“ – *apytikslis (approximate)* išsireiškimas, jei turime omenyje kad amžius yra 36 metai. „Jis – apie trisdešimties“ – tai jau *fuzi (fuzzi)* informacija [10], [13]. Pirmu atveju visuomet galime nustatyti, ar informacija teisinga ar ne. Teiginys „Jam – virš trisdešimt metų“ yra teisingas, jei žmogus yra 36 metų amžiaus, ir neteisingas, jei žmogus yra 28 metų. Tačiau jei turime *fuzi* informaciją, tuomet galimybės nustatyti ar teiginys yra teisingas, ar ne – neturime. Teiginys „Jis – apie trisdešimties“ yra daugiau ar mažiau teisingas abiems žmonėms, tačiau daugiau teisingas 28-ečiui, nei 36 metų amžiaus žmogui.

Duomenys taip pat gali būti *nepilni* arba jų gali *trūkti (missing, incomplete)* [12]. Pavyzdžiui, turite teiginį „Mariaus žmonos vardas – Janė ar Jonė“, nors Jūs žinote kad Mariaus žmona vardu Julė. Įsivaizduokime, vartotojas atlieka paiešką pagal *sutuoktinio* atributą. Jei *sutuoktinio* atributas yra tuščias, tai dar nereiškia, kad žmogus yra viengungis – galbūt mes neturime duomenų apie jo sutuoktinį, ir todėl atributas yra tuščias. Jei vartotojas norėtų sužinoti Mariaus žmonos vardą – jis gautų netikslią informaciją (Janė ar Jonė), bet ji be to būtų ir netiksli, nes iš tiesų žinome, kad Mariaus žmonos vardas – Julė.

Iki šiol aptarėme informaciją, kuri buvo be klaidų. Jei analizuosime teiginius, kuriuose gali būti ir klaidinga informacija, tuomet atsiranda dar didesni neapibrėžtumai. Duomenys yra *klaidingi* arba *neteisingi (erroneous, incorrect)*, jei, pavyzdžiui, kalbama apie žmogų, kuris yra 25 metų amžiaus, o teigiama, kad „jis yra 37 metų amžiaus“. *Netikslūs (inaccurate)* duomenys vienaip ar kitaip yra neteisingi, bet paklaida gali būti taip pat nedidelė, pavyzdžiui, teigiama, kad „jam – 37“, nors iš tiesų jam 36. Šiuo atveju *klaidinga* informacija gali būti laikoma *netikslia*.

Duomenys gali būti *neteisingi (invalid)* ir tuo atveju, kai prieštarauja pačiam teiginiui. Pavyzdžiui, „Jono šeimyninė padėtis – našlys“. Jei Jonas būtų viengungis, jie negalėtų būti našliu.

Duomenų *iškraipymas (distortion)* yra analogiškas netikslumui su netikrumu. Duomenys gali būti sumaišyti, jei nurodo sisteminę klaidą. Pavyzdžiui, teigiama, kad jo amžius – 2 metai. Amžių galima nustatyti apskaičiuojant skirtumą tarp gimimo datos ir teiginio paskelbimo datos. Taigi, jei tas skirtumas nėra lygus 2 metams, tai turime iškraipytus duomenis.

Absurdiška ir *bereikšmė (nonsensical, meaningless)* informacija yra labai artima *klaidingai* informacijai. Teiginiai „amžius – 245 metai“ ar „šeimyninė padėtis – obuolys“ neduoda jokios informacijos, netgi klaidina.

1.5.2 Neatitikimas

Apjungiant kelis sakinius gali atsirasti duomenų neatitikimas. Informacija gali *konfliktuoti (conflicting)*, pavyzdžiui: „šeimyninė padėtis – išsiskyręs“ ir „sutuoktinio vardas – Julė“. Šiuo atveju išeina, kad Jonas yra ir išsiskyręs, ir tuo pačiu turi žmoną.

Nesuderinamumas (inconsistency) geriausiai pasimato iš konteksto, pavyzdžiui kalbant apie laiką: „kiaušinai buvo išvirti 15 valandą“ ir „15 val. 30 min. kiaušiniai dar buvo neišvirti“.

Nesuderinamumą galima nustatyti turint du vienas kitam prieštaraujančius teiginius ir naudojant logiką.

Priimant sprendimą galima ir *susimaišyti (confused)*, pavyzdžiui: jūs teigiate, kad į Kauną atvyksite 15 valandų 05 minutės, bet traukinių tvarkaraštyje tokio traukinio nėra – yra tik traukinys, atvykstantis 15:15. Iš čia galime daryti prielaidą, kad traukinys atvys po 15:00 valandos, arba, kad traukinys šiandien išvis neatvyks (gal kitą dieną?).

1.5.3 Netikrumas

Trečias duomenų netikslumo aspektas – *duomenų netikrumas (uncertaint data)*. Duomenų netikrumas atsiranda dėl skirtingo žinių apie tą patį realų pasaulio objektą interpretavimo. Jei apie objektą tiksliai žinai visą reikiamą informaciją – gali būti *tikras*, ir priešingai – jei apie objektą žinai tik dalinai ar netiksliai – niekuomet negali būti tikras, kitais žodžiais tariant *nesi tikras*. Taigi, pagrindinė duomenų netikrumo priežastis – duomenų netikslumas.

Objektyvus netikrumas

Kai kurie specialistai tvirtina, kad netikrumas yra susijęs su *atsitiktinumu (randomness)* – objektyvia savybe ir *tikimybe (likely)* – nusakančia įvykį, kuris tikriausiai įvyks [8]. Šios dvi savybės nurodo, kad „įvykis yra galimas“, Tai visai nepriklauso nuo vartotojo nuomonės ar tas įvykis įvyks ar ne. Tikimybė (kaip ir atsitiktinumas) yra objektyvi savybė ir vertinama pagal turimus duomenis. Duomenų modeliui keliamas uždavinys – vartotojui pateikti informaciją apie objektą, kurie remiantis būtų galima priimti sprendimą.

Subjektyvus netikrumas

Objektyvus netikrumas yra susijęs su realiu pasauliu ir informacija apie jį. Subjektyvus netikrumas yra susijęs su vartotojo nuomone apie iš turimų duomenų išgautos informacijos tikrumą [11]. Duomenys yra *tikėtini* ir *galimi (believable, probable),* jei vartotojas tuos duomenis pripažįsta, kad ir laikinai. Duomenys yra *abejotini (doubtful)*, jei vartotojas jų negali pripažinti, arba, blogiausiu atveju, jei vartotojas juos pripažįsta su dideliu nenoru.

Sąryšis tarp tikimybės ir galimybės gali būti aprašytas ir subjektyviame kontekste. *Galimybė* ir *būtinybė* yra savybės, kurios atvaizduoja vartotojo nuomonę apie teisingą teiginį. Iš tiesų reikėtų pasikliauti tik *galimybe.*

Pagrindinis netikrumo šaltinis – duomenų netikslumas, neapibrėžtumas. Peržvelkime paprasčiausią – dviprasmybės pavyzdį: „maistas yra degantis“. Kai vartotojas sužino, kad maistas yra „degantis“ – jis atsiduria netikrumo padėtyje, nes maistas gali būti arba karštas, arba aštrus. Tačiau, netikrumą galima eliminuoti. Paprastai mes apie maistą iš konteksto žinome daugiau, nei tik teiginį, kad „maistas yra degantis“. Jei mes esame indų ar kiniečių restorane – tai maistas ko gero bus aštrus, tačiau maistas gali būti karštas, jei mes būtume paprastame restorane.

1.6 Neapibrėžti sąryšiniai duomenys

Gerai žinoma skirtingų duomenų šaltinių schemų problema – kiekviena duomenų bazė turi savitą schemą. Be to, kartais atsitinka taip, kad keletas tos pačios duomenų bazės elementų prieštarauja vienas kitam, nors ir aprašo tą patį realaus gyvenimo objektą. Šias problemas reikia išmokti išspręsti be vartotojo įsikišimo.

Pabandysime išspręsti šią problemą darydami fundamentalią prielaidą, kad duomenų bazė turi turėti tikslius ir išbaigtus duomenis, bet kartu ir duomenų bazių valdymo sistemą (DBVS), kuri būtų atsakinga už veiksmus su duomenimis. Be to, DBVS turi turėti galimybę duomenų integraciją atlikti savarankiškai, be vartotojo įsikišimo [7], [9]. Kitais žodžiais tariant, DBVS turi sugebėti apdoroti neapibrėžtus duomenis *(uncertain data)*. Siūlau tam naudoti tikimybines sąryšinės duomenų bazių valdymo sistemas RDBVS *(probabilistic RDBMS)*.

Idėja sukurti duomenų bazę neapibrėžtiems duomenims nėra nauja. Pavyzdžiui, srityse dirbtinis intelektas *(artificial intelligence),* dedukcinės duomenų bazės *(deductive databases),* žinių paieška duomenų bazėse *(knowledge discovery in databases)* reguliariai užsimenama apie neapibrėžtų duomenų apdorojimą.

1.6.1 Neapibrėžti duomenys

Realaus pasaulio objektai gali būti atvaizduojami *kortežais (tuple),* t.y. aibe *atributų* $T \in D_1 \times ... \times D_n$, kur D_i yra atributo *i domenas*. Aibė realių objektų gali būti apibūdinama *ryšiais* $R \in P(D_1 \times ... \times D_n)$. Galiausiai realių objektų *duomenų bazė* gali būti aprašoma sekančiai: $DB \in P(D_1 \times ... \times D_n)$.

Tikimybinis atributas (probabilistic atribute) yra tradicinis atributas su tikimybės asociacija, pavyzdžiui, domenas $D_i = [0,1] \times D_i$. Dabar galima aprašyti *tikimybinius kortežus* $pT \in D_1 \times ... \times D_n$. *pT* aprašo realaus pasaulio objekto *tikimybę*. Tegul $P_i(pT)$ bus kortežo *pT* atributo *i* tikimybė, o $\pi_i(pT)$ – jos reikšminė dalis.

Tegul $pR \in P(D_1 \times ... \times D_n)$ bus *tikimybinis ryšys*. Lentelėje 1 parodytas tikimybinių ryšių pavyzdys. Mums reikia *pR,* tam, kad turėtume *priminį raktą k*. Pirminis raktas *k* unikaliai identifikuoja realaus pasaulio objektą, todėl to objekto tikimybė lygi 1. Tačiau reikia pastebėti, kad pirminio rakto reikšmės nėra unikalios *pR* tradiciniu požiūriu („vardas“, lentelėje 1).

1 lentelė. Tikimybinė ryšių adresų knygelė

Vardas		Kambarys	
„Jonas“	[1,0]	3122	[0,7]
„Jonas“	[1,0]	3120	[0,3]
„Petras“	[1,0]	2023	[0,4]
„Petras“	[1,0]	2012	[0,6]

Patogumo dėlei įvesime sekantį žymėjimą:

$$\Pi_v(pR) = \{pT \in pR \mid \pi_k(pT) = v\}$$
$$D_i \mid pR = \{v \in D_i \mid \exists pT \in pR) \bullet \pi_i(pT) = v\}$$

$\Pi_v(pR)$ yra aibė visų *pR* kortežų, kurių reikšmė lygi *v*. $D_i \mid pR$ yra domenas D*i,* apribotas reikšmėmis iš *pR*.

Tikimybiniams ryšiams galioja dvi ribinės sąlygos:

- Pirma ribinė sąlyga teigia, kad kiekvieno realaus pasaulio objekto suminė visų galimybių tikimybė bus lygi vienetui. Tai rodo, kad mes priimame uždarą pasaulį.
- Antra ribinė sąlyga teigia, kad kiekvienas atributas yra nepriklausomas.

Jei paprastą ryšį *R* sąlyginai laikysime realiu pasauliu, tai tuomet galima tikimybinius ryšius *pR* sąlyginai laikyti keletu galimų pasaulių. Kadangi individualių atributų tikėtinumai yra

nepriklausomi vienas nuo kito, tai realaus pasaulio objekto tikimybė randama sudauginant tikimybes iš asocijuotų atributų: $\prod_{1 \le i \le n} P_i(pT)$.

Kortežų aibė, atvaizduojanti kiekvieno realaus pasaulio objekto tikimybę, yra vadinama *galimu pasauliu* $pW \subseteq pR$.

Pirma ribinė sąlyga užtikrina, kad *pW* egzistuoja tik viena galimybė kiekvienam realiam objektui. Antra ribinė sąlyga užtikrina, kad kiekvienas realaus pasaulio objektas iš *pR* yra tik vieną kartą paminėtas *pW*.

1.6.2 Duomenų sujungimas

Pabandysime paanalizuoti dviejų adresų knygelių apsikeitimo duomenimis (sujungimo) atvejį. Čia iškyla duomenų sujungimo problema, kai duomenys apie tą patį realaus pasaulio objektą (žmogų adresų knygelėje) prieštarauja vienas kitam. Sujungiant kelias duomenų sekas reikia ne tik apsispręsti kokia bus sujungtų duomenų struktūra, bet ir įvertinti sujungiamų duomenų *patikimumą*.

2 ir 3 lentelėse yra pavaizduotos dvi skirtingos adresų knygelės, kurias sujungus rezultatai buvo patalpinti 1 lentelėje.

2 lentelė. Pirma adresų knygelės

vardas	kambarys
Jonas	3122
Petras	2023

3 lentelė. Antra adresų knygelė

vardas	kambarys
Jonas	3120
Petras	2012

Lentelėje 1 priskirtos tikimybės atvaizduoja atributo reikšmių patikimumo laipsnį. Duomenų tikimybės nustatymo plačiau nenagrinėsime, tačiau aišku, kad tikimybės nustatymui įtakos turi adresų knygelės savininko įvertinimas (kruopštumas, tikslumas), adresų knygelės sudarymo data – vėliau sudarytos adresų knygelės duomenys gali būti tikslesni už seniau sudarytos.

Lentelėse 4, 5, 6 atvaizduojami skirtingi tos pačios adresų knygelės atvejai: 4 lentelėje pavaizduoti ne vienalyčiai atributai, 5 lentelėje – pirma normalinė forma, 6 lentelėje – keli trečios normalinės formos atvejai.

4 lentelė. Adresų knyga ir atstumai. Ne normalinė forma

vardas	kambarys	Atstumas
Jonas	3122 [0,7] 3120 [0,3]	20 [0,7] 18 [0,3]
Petras	2023 [0,4] 2012 [0,6]	50 [0,4] 60 [0,6]

5 lentelė. Pirmos normalinės formos atvaizdavimas

vardas	kambarys	kambarys_tik	atstumas	atstumas_tik
Jonas	3122	0,7	20	0,7
Jonas	3122	0,7	18	0,3
Jonas	3120	0,3	20	0,7
Jonas	3120	0,3	18	0,3
Petras	2023	0,4	50	0,4
Petras	2023	0,4	60	0,6
Petras	2012	0,6	50	0,4
Petras	2012	0,6	60	0,6

6 lentelė. Trečios normalinės formos atvaizdavimas

PAGRINDINĖ	
Vardas	id
Jonas	1
Petras	2

KAMBARYS		
Id	kambarys	tikimybė
1	3122	0,7
1	3120	0,3
2	2023	0,4
2	2012	0,6

ATSTUMAS		
Id	atstumas	tikimybė
1	20	0,7
1	18	0,3
2	50	0,4
2	60	0,6

Pirmos normalinės formos atveju matome, kad čia galimi dideli duomenų atributų pasikartojimo, o kartu ir neatitikimo atvejai. Trečios normalinės formos atveju, priešingai, – gali prireikti daug ryšių, dėl ko gali sumažėti sistemos efektyvumas.

Atributas *kambarys* ir *atstumas* 4 lentelėje yra nepriklausomi vienas nuo kito, tai reiškia kad bet kokia nurodyta distancija gali tikti bet kokiam kambariui. Tai tampa aišku naudojant trečią normalinę formą, pagal kurią abu atributai yra atvaizduojami atskirose lentelėse. Šiame pavyzdyje atstumas ko gero priklausomas nuo kambario numerio. Trečioji normalinė forma gali modeliuoti šią priklausomybę įvedant papildomą priklausomybės atributą toje pačioje lentelėje. Atributas *kambarys* ir *atstumas* pagrindinėje lentelėje yra pakeisti vienu atributu, pavadintu *kambarys_tik*.

1.6.3 Užklausų apdorojimas

Tikimybinės užklausos [14] turi pateikti kitokius rezultatus nei paprastos užklausos. Panagrinėkime užklausą 1 lentelei.

```
SELECT vardas, kambarys
FROM adresuknyga
```

Žiūrint iš tradicinės pusės, atsakymas {(Jonas, 3122), (Jonas, 1320), (Petras, 2023), (Petras, 2012)} neduoda jokios prasmės, kadangi atributas *vardas* yra raktas, tai tikimasi tik vieno kortežo su reikšme *„Jonas“*. Žiūrint iš tikimybinių ryšių perspektyvos pusės, gautas atsakymas yra dalinai teisingas, nes netikrumas duomenyse dėl kambarių, neabejotinai turės įtakos atsakymo apie kambarius patikimumui.

Norint užklausomis apdoroti tikimybinius duomenis, reikia įvertinti sąryšinės algebros operatorių semantiką. Naudojantis galimų pasaulių metodu galime tikėtis sekančio atsakymo: {(0,7, Jonas, 3122), (0,3, Jonas, 3120), (0,4, Petras, 2023), (0,6, Petras, 2012)}, kuris nurodo, kad

atsakymas (Jonas, 3122) yra su 70% tikimybe, ir tik 30%, jog atsakymas (Jonas, 3120) yra teisingas.

2 NESTRUKTŪRIZUOTŲ DUOMENŲ TURINIO ANALIZĖ

Interneto svetainėse tekstinė informacija dažniausiai pateikiama nestruktūrizuotu pavidalu (įvairūs straipsniai, pranešimai ir kt.). Norint tokiuose duomenyse surasti pageidaujamą informaciją – pirmiausia reikia juos struktūrizuoti. Informacija struktūrizuojama analizuojant tekstinę informaciją po vieną sakinį.

2.1 Sintaksinė analizė

Nestruktūrizuotus tekstinius duomenis lingvistiniu požiūriu labai patogu analizuoti sakiniais [2]. Kiekviename sakinyje dažniausiai galime aptikti bent vieną *teiginį*. Teiginiu laikysime realaus pasaulio objektą, jo savybę ar kitą parametrą. Teiginys yra sudarytas iš predikato ir argumentų. Predikatas arba kažką teigia, konstatuoja, neigia ar pan., o argumentai papildo teiginį.

Teiginio struktūra gali būti labai sudėtinga, jis gali turėti savo sudėtyje sąlygas, sudėtinius argumentus, specifines frazes ir pan. Predikato, argumentų ir kitų teiginio sudėtinių dalių vietą teiginio struktūroje nusako sintaksės taisyklės, kurios būdingos konkrečiai kalbai

Veiksnys, objektas. Pirmiausia reikia išskirti veiksnį. Veiksnys – tai sakinio dalis, nusakanti veikėją. Priklausomai nuo sakinio struktūros, galimi keli atvejai:

- Jonas yra atleidžiamas iš pareigų. Direktorius paaukštino Joną. Aplinkybės privertė Joną atsistatydinti.

Tarinys. Antras argumentas, kurį reikia išskirti, yra tarinys. Tarinys nusako veiksmą, kuris vyksta su objektu. Tarinys taip pat gali nurodyti veiksmo vykimo laiką:

- Jonas nusprendė įsidarbinti. Laikraštyje bus parašyta apie Joną. Pajamos išaugo dvigubai.

Kiekinė išraiška. Teiginys gali nusakyti objekto kiekinę išraišką. Kiekinė išraiška gali būti nusakoma skaičiais ar žodžiais, pavyzdžiui:

- Jonas – antras pagal ūgį krepšinio komandoje. Jonas jau tris – keturis kartus bandė įsidarbinti. Apyvarta išaugo 23 procentais.

Laiko parametras. Laiko parametras nusako, kuriuo laiko momentu vyksta veiksmas. Laikas taip pat nusakomas skaitine arba žodine išraiška:

- Penktadienį Jonas išvažiuoja į kaimą. Algą pakels tik gruodį. Pirmą šių metų ketvirtį įmonė nepatyrė nuostolių.

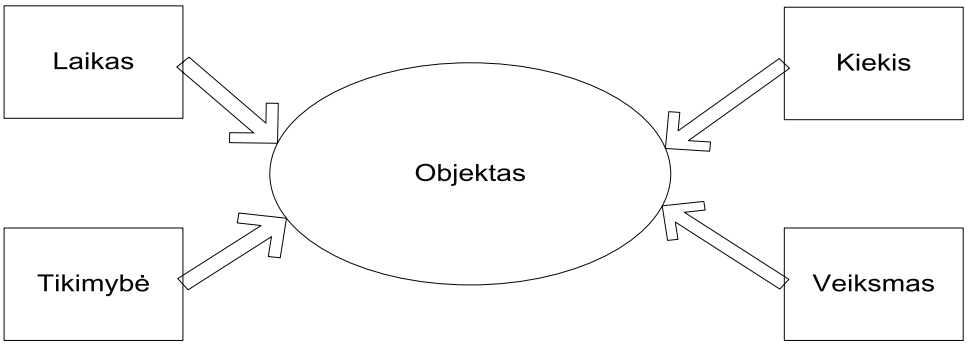
Tikimybė. Kartais sakinyje yra nurodoma teiginio tikimybė, leidžianti spręsti apie teiginio tikrumą, tikėtinumą:

- Planuojama, kad šiemet apyvarta bus didesnė 10%. Jonas tikisi nepavėluoti į darbą. Būsto paskolų palūkanos, tikimasi, nedidės.

Norint transformuoti lingvistikoje apibrėžtas teiginio dedamąsias – argumentus ir predikatus – į struktūrizuotų duomenų elementus – atributus ir santykius, - turime atskirti teiginio argumentų aibėje tuos argumentus, kurie parodo apie ką teiginyje pateikta informacija. Struktūrizuotuose duomenyse šis argumentas atitiks objektą, o kiti teiginio argumentai transformuojami į atributų reikšmes.

2.2 Teiginio sandara

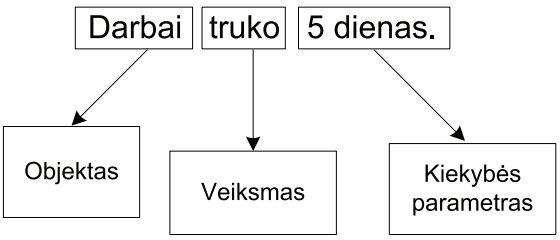
Iš nestruktūrizuotos tekstinės informacijos išrinkus ir suklasifikavus tam tikrus elementus, gauname teiginius. Teiginiuose saugoma informacija apie objektą.



7 pav. Teiginio struktūra

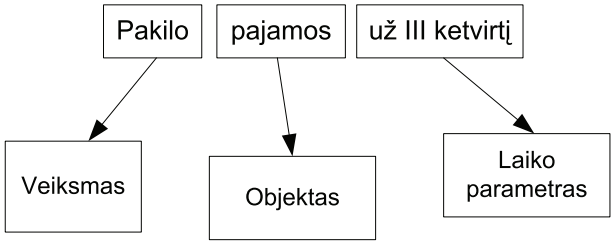
Teiginys – tai iš informacijos šaltinio išgauta informacija apie kažkokį realaus pasaulio objektą ar reiškinį. Viename sakinyje galime aptikti vieną ar daugiau teiginių. Teiginys būtinai turi turėti objektą ir bent vieną objekto parametrą (rodiklį, laiko, kiekio ar tikimybės parametrą).

Pavyzdys 1: „Darbai truko 5 dienas“.



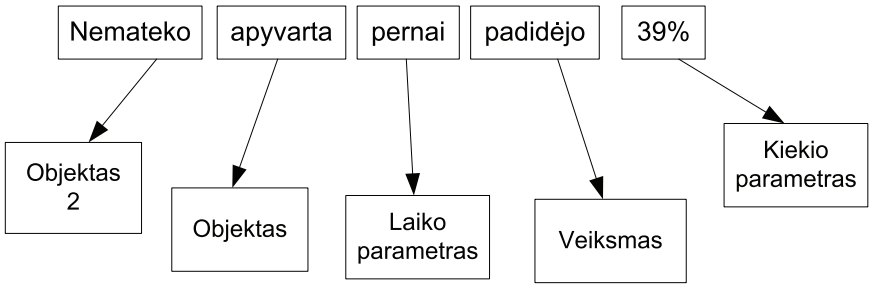
Šiuo atveju sakinyje aptinkame objektą, veiksmą nusakantį rodiklį bei vieną iš parametrų – kiekybės parametrą, išreikštą dienomis.

Pavyzdys 2: „Pakilo pajamos už III ketvirtį“.



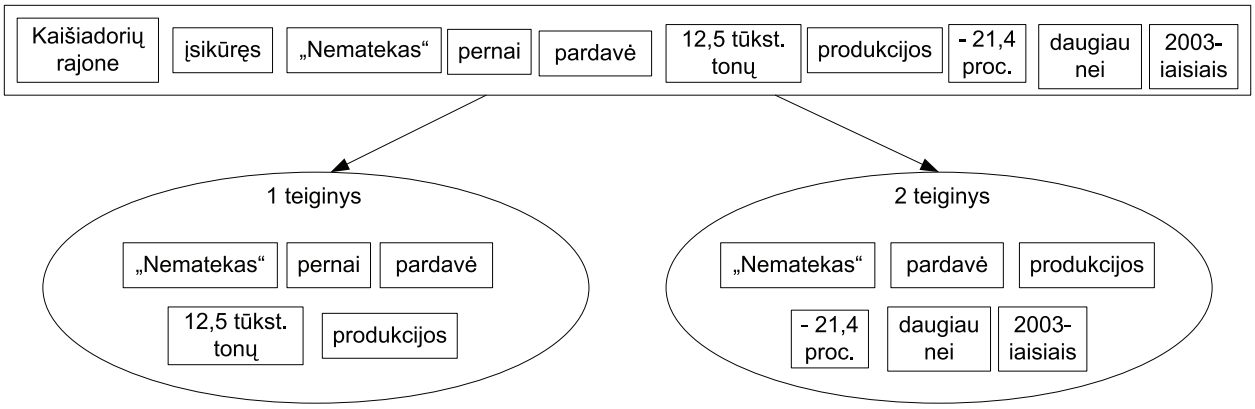
Šiame sakinyje tai pat matome objektą, veiksmą nusakantį rodiklį bei laiko parametrą, kuris nurodo kuriam momentui galioja šis teiginys.

Pavyzdys 3: „Nemateko apyvarta pagal pateiktus duomenis pernai padidėjo 39%“.



Šiame sakinyje objektas „apyvarta“ yra papildomai patikslinamas papildomu parametru, kurį mes vadiname objektas2, ir jis patikslina, kad yra kalbama apie Nemateko įmonės apyvartą. Šiuo atveju sakinys turi ir laiko, ir kiekio parametrus. Sakinyje be išskirtų teiginių yra ir informacijos, kurios negalima struktūrizuoti: „pagal pateiktus duomenis“.

Pavyzdys 4: „Kaišiadorių rajone įsikūręs „Nematekas“ pernai pardavė 12,5 tūkst. tonų produkcijos - 21,4 proc. daugiau nei 2003-iaisiais“.



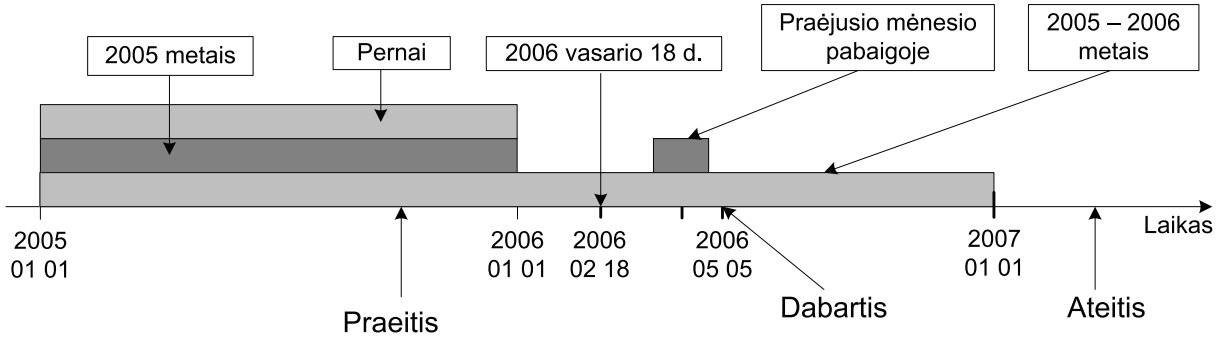
Šiuo atveju iš vieno sakinio gauname 2 teiginius, kuriuos vėliau analizuosime atskirai vieną nuo kito. Vieni parametrai naudojami abiejuose teiginiuose bendrai (produkcija, Nematekas, pardavė), kitus parametrus naudojame arba pirmam, arba antram teiginiui charakterizuoti (21,4 proc., 12,5 tūkst. tonų, pernai, 2003-iaisiais).

2.2.1 Laiko komponentas

Teiginys su laiko komponentu nurodo, kada vyksta veiksmas. Laikas gali būti nusakomas kaip konkreti skaitinė išraiška, intervalas ar laikotarpis. Įvykis taip pat gali įvykti kažkokiu tai laiko momentu praeityje, ateityje ar dabartyje, prieš ar po momento, kaip atskaitos taško. Detaliau paanalizuosime teiginius su laiko komponentais.

Laikas nusakomas konkrečia data, intervalu ar laikotarpiu:

„BBH įmonės *2005 metais* pardavė produkcijos už 193,3 mln. litų. Teismas turėtų atšaukti *2006 metų vasario 18 dienos* sprendimą. Drabužių mažmenininkė *praėjusio mėnesio pabaigoje* turėjo 1 tūkst. 116 parduotuvių. Alaus daryklos "Kalnapilio-Tauro grupė" pajamos *pernai* išaugo 1 proc. iki 97,3 mln. litų. Šį projektą numatoma įgyvendinti *2005-2006 metais“*.



8 pav. Įvykių išsidėstymas laike

Šiuo atveju laikas nusakomas laike apibrėžta konkrečia išraiška:

- *2005 metais* – omenyje turimas laikotarpis tarp 2005 m. sausio 01 d. iki 2005 m. gruodžio 31 d.;
- *2006 metų vasario 18 diena* – konkreti data;
- *Praėjusio mėnesio pabaigoje* – omenyje turimas laikotarpis nuo praėjusio mėnesio kažkurios iš paskutinių iki paskutinės dienos. Pastaba: šioje situacijoje reikia atsižvelgti į straipsnio publikavimo datą, kad būtų galima tiksliai nustatyti mėnesį.
- *Pernai* – omenyje turimas laikotarpis nuo praeitų metų pradžios iki pabaigos. Pastaba: reikia atsižvelgti į straipsnio publikavimo datą, kad būtų galima tiksliai nustatyti metus.
- *2005 – 2006 metais* – omenyje turimas laikotarpis nuo 2005 m. sausio 01 d. iki 2006 m. gruodžio 31 d.

Laikas gali būti nusakomas tiek datos formatu (2005 02 18, 2006 01 01), tiek žodine (pernai, pusmetis, liepa). Gali būti taip, kad ir skaičius 2005, ir žodis „pernai“ simbolizuoja tą patį laikotarpį. Pamėginsime išskirti kelis laiko nusakymo atvejus.

Laiko parametras gali būti išreikštas **konkrečiai** nusakant datą ar veiksmo vykimo laikotarpį**,** pavyzdžiui:

„PVM įstatymo pataisa priimta *2006 m. vasario mėn. 26 dieną*. Kazimieras Gudaitis gimė *1959 m. gruodžio mėn. 13 d*. Įmonės akcininkų susirinkimas vyks *šių metų vasario 16 d.,* 15:00 posėdžių salėje. Į kiną eisime *šeštadienį*. *2002 metų II ketvirtis* buvo labai pelningas“.

Veiksmo vykimo laikotarpis gali būti nusakomas priklausomai nuo kažkokio **atskaitos taško**, pavyzdžiui:

„Praeitą savaitę pasiektas dar vienas rekordas. Pokalbis dėl darbo turėtų įvykti *po keleto dienų*. Aplinkos užterštumas *per pastaruosius 10 metų* padidėjo nežymiai. Naudotis elektroniniu paštu negaliu jau *nuo 2005 m. gegužės 5 d*. Visiems darbuotojams *nuo šiol* įsakau nevėluoti į darbą“.

Laiko parametras gali būti nusakomas **intervalu.** Intervalas apibrėžiamas veiksmo pradžios ir pabaigos taškais, pavyzdžiui:

„Piktnaudžiavimas tarnybine padėtimi vyko n*uo 2003 metų pradžios iki 2005 metų vasario vidurio*. Suvirinimo darbus dirbu jau *penkis-šešis metus*. Planuojama, kad investicijos atsipirks *per 8-10 metus“*.

Veiksmo vykimo laikas gali būti nusakomas **apytiksliai**, pavyzdžiui:

„Statybos darbai pradėti *šių metų pavasarį*. Paskutinį kartą su Stasiu žvejojau *liepos vidury*. Piktžoles planuojame išnaikinti *per kelerius metus“.*

2.2.2 Kiekio komponentas

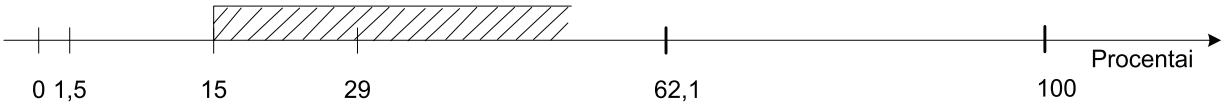
Kiekio komponentas teiginyje nurodo kiekybinę realaus pasaulio objekto išraišką. Kiekis gali būti išreiškiamas konkrečiu skaičiumi, intervalu arba žodine išraiška. Kiekinės išraiškos komponentus grupuojame pagal tipą: palūkanos, valiuta, pelnas, išlaidos, procentinė išraiška ar kt.

„Pradinė „Eismo“ akcijų kaina buvo *6 mln. litų*. Žemdirbiai per keturis šių metų mėnesius už perdirbimo įmonėms parduotą pieną gavo *204,8 mln. litų*. Plautų morkų kaina iš augintojų ūkių siekia *1050 litų* už toną“.

9 pav. Objektų kiekinės išraiškos. Litai

9 paveiksle pavaizduoti keli teiginių kiekiniai komponentai. Kiekvienas komponentas atvaizduotas ant valiutos (litų) ašies, atsižvelgiant į komponento skaitinę dalį.

- G. Kateiva Eltai teigė, kad šiemet pelningai dirbanti grupė tikisi per 15 proc. pardavimų augimo.
- Bendrovės pajamos minėtu laikotarpiu padidėjo 62,1 proc.
- Konsoliduoti pardavimai išaugo 1,5 proc.
- Lietuvoje yra tiek pat interneto vartotojų, kiek Lenkijoje ir Portugalijoje – 29 proc. gyventojų.



10 pav. Objektų kiekinės išraiškos. Procentai

10 paveiksle ant procentų ašies pavaizduoti keli teiginių kiekiniai komponentai.

Objekto kiekinė išraiška gali būti nusakoma **konkrečia skaitine išraiška**:

„Audimo“ grynasis pelnas išaugo *15,4%.* Pavasario akcijos metu riedučiams taikoma *15 procentų* nuolaida. Europos fondų paramos modernizuojant ūkį siekia *99* pareiškėjai.

Kiekinė išraiška gali priklausyti nuo kažkokio **atskaitos taško**:

Kompensacijos mokamos visiems, patyrusiems didesnę *nei 300 Lt* žalą. Būsto paskolos palūkanos gali siekti *net 8,1%.* Pasaulyje skraido daugiau *nei 600* tokio modelio lėktuvų.

Kiekis gali būti nusakomas **intervalu**. Intervalas nurodomas dviem skaičiais:

Projektuotas linijos pajėgumas – *nuo 10000 iki 12000 butelių* per parą. „Auvigos“ nuostoliai 2004 metais siekė *4-5 mln. litų*. Maksimalus skrydžio atstumas *5000-6000 km*.

Kiekis taip pat gali būti nusakomas **apytiksliai**, pavyzdžiui:

Deklaracijas pateikė tik *keletas* įmonių. Keliamų reikalavimų priešgaisrinei apsaugai neįvykdo maždaug *pusė* įmonių. Mokesčius sąžiningai moka tik *vienetai* įmonių.

2.2.3 Tikimybės komponentas

Neapibrėžtumui formalizuoti įvedamas tikimybės komponentas, kuris nusako, koks to teiginio patikimumas. Tikimybė gali būti išreiškiama skaitine išraiška (procentais) arba žodžiu, nusakančiu tikimybę. Pavyzdžiui:

Planuojamos II ketvirčio pajamos – 15,5 tūkst. Lt. Analitikai *prognozavo*, kad pardavimai padidės 11-14%. *Tikimasi* padidinti naujų automobilių pardavimus 2005 metais.

2.2.4 Rodiklio komponentas

Rodiklio komponentas apibūdina objekto veiksmą ar veiksmo pokytį, pavyzdžiui:

Populiariausio benzino kaina *pakilo* 3 centais. Vyriausybė *skyrė* 3 mln. Lt. papildomų lėšų labdarai. Infliacijos lygis per pastaruosius kelis metus *stabilizavosi*.

2.2.5 Nestruktūrizuojama informacija

Analizuojant straipsnius, kartais atrinkus struktūrizuoti galimą informaciją, lieka informacija, kurios negalima arba neverta struktūrizuoti. Kitaip tariant mes susiduriame su informacija, kurios negalima arba neverta sustruktūrizuoti. Pavyzdžiui:

"Ežio" paslaugos orientuotos į jaunus, mažesnes pajamas gaunančius gyventojus. Socialdemokratės Irenos Šiaulienės įsitikinimu, LRT finansavimas yra kompleksinis klausimas, ir "vien konvencinio mokesčio įvedimas neišspręstų problemų". Abonentinis mokestis garantuotų didesnę finansinę nepriklausomybę nuo politinės įtakos.

2.3 Apibendrinti teiginiai

Išanalizavus kelis publikuotus straipsnius interneto naujienų portaluose, gautus duomenis patalpiname 7 lentelėje. Lentelėje pateikiama iš straipsnių paimta informacija, atmetus nestruktūrizuojamą informaciją, suskirstyta pagal prasmę į atitinkamus stulpelius.

7 lentelė. Apibendrinti teiginiai

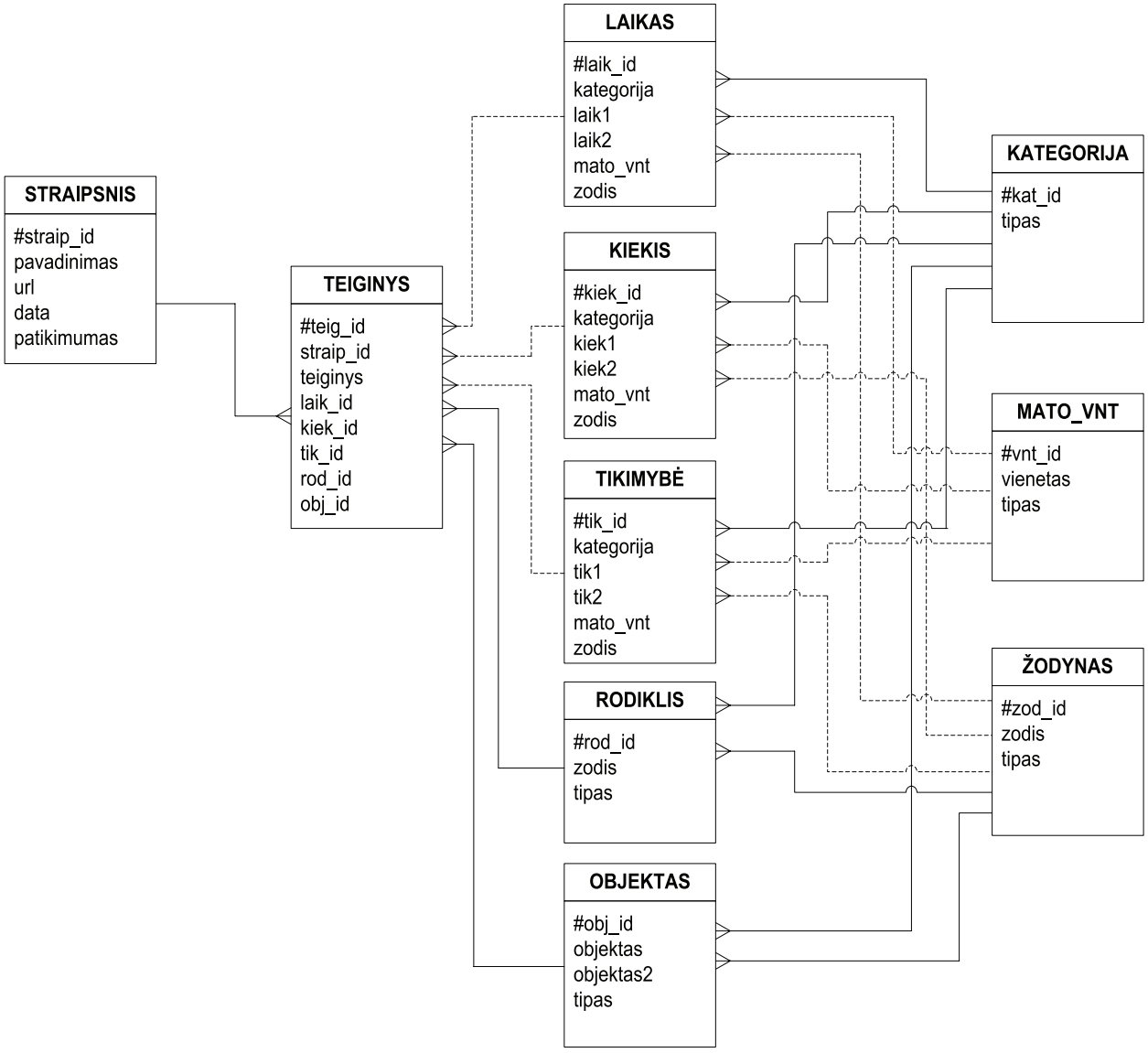
id	objektas	objektas2	rodiklis	kiekis	laikas	tikimybė
1	produkcija	Nemateko	buvo	1250 t	pernai	
2	Nematekas		yra	vienas didžiausių		
3	eksportas	Nematekas	padidės	3,4 karto	šiemet	tikisi
4	ūkio bankas		yra	penktas pagal dydį		
5	grynasis pelnas	Ūkio bankas	uždirbo	2,406 mln. Lt	2005 1 ketv.	
6	grynasis pelnas	ūkio bankas	uždirbo	0,544 mln. Lt	2004 1 ketv.	
7	pelnas	bankas	uždirbti	1,098 mln. Lt	1 ketv.	planavo
8	turtas	bankas	išaugo	10%	metų pradžia	
9	Žalgiris		laimės		šiandien	90-95%
10	turistai		laukiama	40000	Šiemet	
11	Nematekas	gamyba	didina			
12	Nematekas	produkcijos	pardavė	virš 1200 t	2004	
13	projektas		įgyvendinti		2005-2011 m	numatoma
14	Realtus		skirti	40 mln. Lt	5 metai	numato
15	Kaina	Plautų morkų	siekia	1050 Lt/t		
16	Rusija		Skolinga	43,1 mlrd. Lt		
17	Rusija		skirs	13 mlrd. dol.	birželį	
18	Lukoil	kompanija	yra	stambiausia		
19	Benzinas		atpigo	3 ct/ltr		
20	Benetton		uždirbo	23 mln. Eur.		

Kaip matome iš lentelės, kai kurie teiginiai turi tuščių atributų. Taip yra dėl to, kad informacija yra gauta iš nestruktūrizuotų duomenų, o nestruktūrizuoti duomenys gali būti neišbaigti ir pateikiami bet kokiame pavidale.

3 NESTRUKTŪRIZUOTŲ DUOMENŲ MODELIS

3.1 ER schema

Atlikus informacijos šaltinių analizę ir atsižvelgiant į gautų rezultatus (7 lentelė), sudarome duomenų bazės ER schemą.



11 pav. Duomenų bazės ER schema

11 paveiksle matome pavaizduotą duomenų bazės ER schemą, skirtą iš nestruktūrizuotų duomenų išgautai informacijai saugoti.

3.2 Esybės

Detaliau aptarsime paveikslėlyje 11 pavaizduotas esybes.

Esybė **STRAIPSNIS** saugoja informaciją apie informacijos šaltinius. Atributai:

- *#straip_id* – informacijos šaltinio (straipsnio) identifikacinis numeris;
- *pavadinimas* – straipsnio pavadinimas;
- *url* – straipsnio URL adresas;
- *data* – straipsnio publikavimo data;
- *patikimumas* priklauso nuo daugelio faktorių (svetainės, kurioje publikuotas, autoriaus ir kt.).

Esybė **STRAIPSNIS** yra susieta su kita esybe – **TEIGINYS**. Teiginio esybė saugoja informaciją apie straipsniuose esančius teiginius. Atributai:

- *#teig_id, straip_id, laik_id, kiek_id, tik_id, rod_id, obj_id* – teiginio, straipsnio, laiko, kiekio, tikimybės, rodiklio, objekto identifikaciniai numeriai;
- *teiginys* – originalus teksto fragmentas, iš kurio buvo gautas teiginys.

Teiginio esybė ryšiais yra susieta su penkiomis esybėmis, tiksliau nusakančiomis teiginį: **LAIKAS, KIEKIS, TIKIMYBĖ, RODIKLIS, OBJEKTAS.**

Esybė **OBJEKTAS** skirta apibūdinti realų ar įsivaizduojamą reikšmingą daiktą ar objektą, apie kurį rašoma straipsnyje. Objekto esybės atributai:

- *#obj_id* – objekto identifikacinis numeris;
- *objektas* – objekto pavadinimas. Kartais objektas būna patikslinamas – tuomet naudojamas parametras *objektas2*. Pavyzdžiui, sakinyje „Apyvarta išaugo dvigubai“ matome, kad objektas – apyvarta. Tačiau jei būtų sakinys „Firmos Nematekas apyvarta išaugo dvigubai“, tuomet turėtume užpildytus abu laukus.
- *tipas* – nusako kokiai kategorijai priklauso objektas.

Objekto pavyzdžiai: Nematekas, apyvarta, rinka, sandėris, direktorius, terminas, veikla, IBM, pienas, bedarbis, akcija ir kt.

Esybė **RODIKLIS** apibūdina veiksmą ar pokytį, vykstantį su objektu. Naudojami atributai:

- *#rod_id* – rodiklio identifikacinis numeris;
- *zodis* – rodiklio reikšmė;
- *tipas* – rodiklio tipą nusakantis parametras.

Rodiklio pavyzdžiai: pakilo, nusileido, yra, buvo, pagerėjo, uždirbo, kainuoja, skyrė, rezervuota, atskilo, krito ir kt.

Esybė **KIEKIS** apibūdina teiginio kiekinę išraišką. Kiekio esybei aprašyti naudosime atributus:

- *#kiek_id* – parametro identifikacinis numeris;
- *kategorija* nusako būdą, kaip teiginyje išreikštas kiekinis parametras. Kiekis gali būti nusakomas konkrečia skaitine išraiška, apibrėžiamas intervale ar atsižvelgiant į atskaitos tašką. Kiekis taip pat gali būti nusakomas ir žodine išraiška.
- *kiek1, kiek2, zodis*. *kiek1* skirtas aprašyti kiekio skaitinei išraiškai. Jei kiekis aprašomas intervalu – papildomai naudojamas *kiek2*. Jei kiekis nurodomas ne skaitine išraiška, naudojamas laukas *zodis*.
- *mato_vnt* – skirtas apibrėžti teiginio mato vienetui.

Kiekio pavyzdžiai: 2004, du, šešiolika, 14 mln. litų, 23 mlrd. Eurų, penki šimtai, virš 830 tūkst. Lt, maždaug 600 darbuotojų, nuo 2 iki 4 ir kt.

Esybė **LAIKAS** apibūdina teiginio laiko komponentę. Naudojami atributai:

- *#laik_id* – parametro identifikacinis numeris;
- *kategorija* – nusako būdą, kaip teiginyje išreikštas laiko parametras. Laikas gali būti nusakomas konkrečia skaitine išraiška, apibrėžiamas intervale ar atsižvelgiant į atskaitos tašką. Laikas taip pat gali būti nusakomas ir žodine išraiška.

- *laik1, laik2, zodis*. *laik1* skirtas aprašyti laiko skaitinei išraiškai. Jei laikas aprašomas intervalu – papildomai naudojamas *laik2*. Jei laikas nurodomas ne skaitine išraiška, naudojamas laukas *zodis*.
- *mato_vnt* – skirtas apibrėžti teiginio mato vienetui.

Laiko pavyzdžiai: 2004 vasario 25 d., šiandien, žiemą, iki šiol, praeitą pavasarį, per 2004 metus, kasmet, nuo metų pradžios iki birželio mėnesio.

Esybė **TIKIMYBĖ** apibūdina teiginio tikimybės komponentę. Naudojami atributai:

- *#tik_id* – parametro identifikacinis numeris;
- *kategorija* – nusako būdą, kaip teiginyje išreikštas tikimybės parametras. Tikimybė gali būti nusakoma konkrečia skaitine išraiška, apibrėžiama intervale. Tikimybė taip pat gali būti nusakomas ir žodine išraiška.
- *tik1, tik2, zodis*. *tik1* skirtas aprašyti tikimybės skaitinei išraiškai. Jei tikimybė aprašomas intervalu – papildomai naudojamas *tik2*. Jei tikimybė nurodoma ne skaitine išraiška, naudojamas laukas *zodis*.
- *mato_vnt* – skirtas apibrėžti teiginio mato vienetui.

Tikimybės pavyzdžiai: galbūt, ko gero, tikimasi, prognozuoja, 10%-15% tikimybė, ne daugiau 50%, planuojama ir kt.

Esybė **KATEGORIJA** nusako kategoriją, kuriai priklauso atitinkamas parametras. Naudojami atributai:

- *#kat_id* – kategorijos identifikacinis numeris;
- *tipas* nurodo kategorijos tipą.

Esybė **MATO_VNT** aprašo naudojamus mato vienetus. Naudojami atributai:

- *#vnt_id* – mato vienetų identifikacinis numeris;
- *vienetas* – mato vieneto reikšmė;
- *tipas* – nurodo, kokiam tipui priklauso mato vienetas.

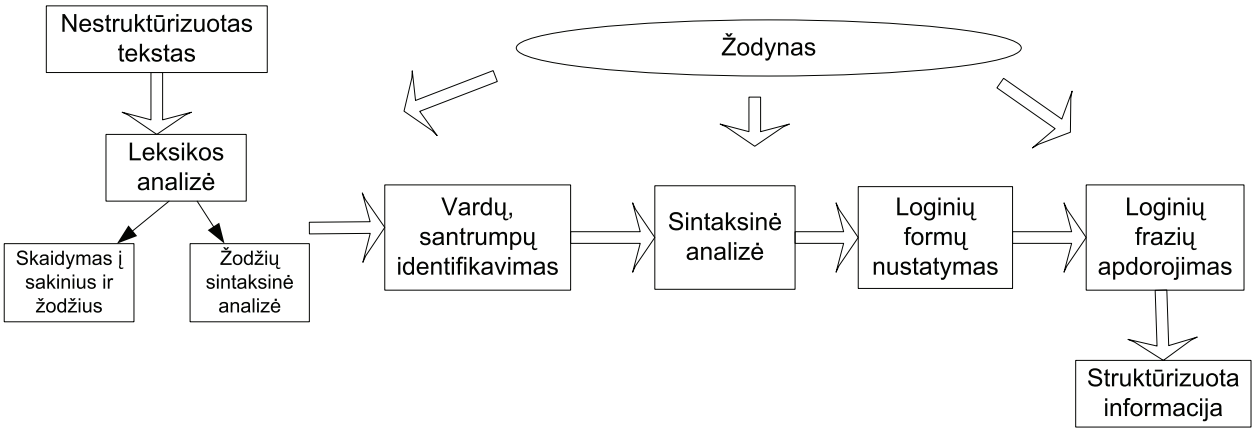
Esybė **ŽODYNAS** skirta aprašyti teiginio žodinius parametrus. Naudojami atributai:

- *#žod_id* – žodžio identifikacinis numeris;

- *zodis* – žodžio reikšmė;
- *tipas* – nurodo, kokiam tipui priklauso žodis.

3.3 Nestruktūrizuotų duomenų tyrimas

12 paveiksle pavaizduota schema nestruktūrizuotiems duomenims tirti. Iš schemos matyti, kad sistemą sudaro atskiri moduliai konkretiems uždaviniams spręsti. Sistemai yra pateikiami pradiniai duomenys – nestruktūrizuota informacija (tekstas), kuri, perėjusi visus tarpinius modulius, tampa struktūrizuota.



12 pav. Nestruktūrizuotų duomenų tyrimo schema

12 paveiksle pavaizduota schema parodo, kokių modulių pagalba galime tirti nestruktūrizuotus duomenis. Keliaujant iš kairės į dešinę, modeliai tampa sudėtingesni ir sudėtingesni. Kiekvienas iš šių modulių gali būti analizuojamas kaip atskiras objektas individualiai.

Aptarsime kiekvieną iš paminėtų modelių.

3.3.1 Leksikos analizė

Pirmiausiai reikia suskaidyti tekstą į atskirus sakinius, po to – į žodžius. Išanalizavus visus sakinį sudarančius žodžius, priskiriame kiekvieną žodį prie tam tikros sintaksinės kategorijos. Tuo pat metu sakinyje nustatomi teiginiai, o prie teiginių ryšiais susiejami kiti, teiginius patikslinantys, žodžiai (žr. skyrių „Sintaksinė analizė“).

Leksikos analizės modulis turi būti atsakingas už:

- dokumento suskaidymą į sakinius;

- sakinių skaidymą į žodžius;
- žodžių paiešką žodynuose ir priskyrimą sintaksinėms kategorijoms.

3.3.2 Vardų, santrumpų identifikavimas

Šis modulis turi būti atsakingas už tikrinių vardų identifikavimą tekste ir semantinių kategorijų priskyrimą jiems. Tikriniai vardai gali būti suskirstyti į daugelį kategorijų, pavyzdžiui:

- asmenys („Jonas“, „Jonas Paulius“, „Jonas Jonaitis“, „Kęstutis“); geografinės vietovės ar pavadinimai („Jonava“, „Jonaičiai“, „Kęstučio gatvė“); įmonės ir organizacijos („Sony“, „Jonas ir partneriai“, „Jonaičio įmonė“); daiktai, produkcija, prekiniai ženklai („VW-Golf“, „Barbė“, „Mona Liza“); sporto komandos, muzikos grupės, meno kolektyvai ir kt. („Žalgiris“, „Varpelis“, „Enigma“, „Metallica“); politika, ekonomika („Antras pasaulinis karas“, „Džordžas Bušas“, „Huseinas“); teisėtvarka, nuostatai, įstatymai („Lygių galimybių aktas“, „Tarptautinė prekybos sutartis“, „PVM įstatymas“).

Kai kurioms kategorijoms gali būti papildomai priskirti paprasti papildomi ženklai, pavyzdžiui:

- pirmosios vardo raidės trumpiniai („A.Petraitis“, „J.Tarutis“, „G.Navickienė“); išsilavinimo, laipsnio, pareigų žymekliai („dr.“, „doc.“, „prof.“, „p.“, „gerb.“, „dir.“, „vyr.“,); įmonių tipai („UAB“, „VšĮ“, „AB“, „Spec. pask. UAB“, „VĮ“); įvairūs specialūs sąrašai, pilnų pavadinimų santrumpos ir panašiai.

Vardų atpažinimo problema yra pakankamai sudėtinga ir ją išspręsti nėra lengva. Taip pat negalima pamiršti *dviprasmybės* problemos (kaip aukščiau minėtas pavyzdys: „maistas yra degantis“). Kita problema – išgalvoti vardai, slapyvardžiai ar trumpiniai, kurie paprastai dokumente pasitaiko labai dažnai ir įvairiomis formomis.

3.3.3 Sintaksinė analizė

Šiame etape visų pirma reikėtų atsižvelgti į sintaksinius aspektus, kad būtų galima nustatyti elementarius sintaksinius vienetus, tokius kaip daiktavardžiai (ar jų grupės) ir veiksmažodžiai (ar jų grupės). Sistemai išskyrus elementarius sintaksinius vienetus, reikia pažymėti atraminius tų vienetų žodžius. Kai sistema išskiria daiktavardį, dar papildomai yra

išskiriama daiktavardžių frazė. Daiktavardžių frazei sukuriama atitinkama semantinė struktūra ir *esybė,* kuri yra tiesiogiai susiejama su pirminiu daiktavardžiu.

Sekantis etapas – ką tik išgautos informacijos papildymas. Iš daiktavardžių frazės išskiriame daiktavardį, ir analizuojame, ar jis neturi patikslinančių žodžių. Dažnai iškyla problema nustatyti pagrindinį daiktavardį, ir atskirti jį patikslinančią aplinkybę, pavyzdžiui, situacijoje: „įmonės prezidentas“ ir žodis „įmonė“, ir „prezidentas“ yra daiktavardžiai.

Kitas etapas – įvertinti papildomai daiktavardžius apibūdinančią informaciją (priklausomybės, konjunkcijos), pavyzdžiui: „Nematekas, mėsos produktų gamintojas“ arba „Jonas Jonaitis, 40 metų amžiaus, UAB „ABC“ marketingo direktorius“.

3.3.4 Loginė forma

Veiksmai su išskirtais teksto segmentais atliekami priklausomai nuo *loginės formos*. Po to, kai visi segmentai yra apdoroti, teksto fragmentas atrodo kaip vidinė loginių formų, atitinkančių esybes, ir įvykių seka.

Kiekviena loginė forma yra teiginys su argumentais. Loginė formą galima pavaizduoti kaip objektą su jį nusakančiais parametrais, pavyzdžiui „COCA-COLA Inc“. Šiuo atveju turime esybę „kompanija“, o esybės parametras – „vardas“. Esybė taip pat gali turėti kitą parametrą, vardu „vietovė“, kuri savo ruoštu gali būti susieta su kita geografinio pobūdžio esybe.

3.3.5 Loginės frazės

Logiškai susiję gali būti ne tik atskiros kalbos dalys, bet ir atskiros frazės. Turint dvi, rodos, viena su kita nesusijusias frazes, logiškai galima susieti jų objektus, pavyzdžiui:

- „Jonas turi 2 vaikus“;
- „Įmonės direktorius – Jonas“.

Šiuo atveju abi frazės turi bendrą objektą – Joną, ir tai nustatome remdamiesi semantiniu požiūriu. Kadangi objektas abiejose frazėse yra vienas ir tas pats, tai galima daryti išvadą, kad „Įmonės direktorius turi 2 vaikus“.

3.3.6 Žodynas

Žodynas – labai svarbi šios schemos dedamoji. Žodynas nustato pageidaujamų žodžių priklausomybę vienai ar kitai kalbos daliai, nusako operacijas ir taisykles, susijusias su linksniais, laikais, kaityba, semantines ir sintaksines taisykles bei atlieka dar daug kitų nepaminėtų funkcijų.

Be informacijos apie įvairias kalbos dalis ir operacijas su jomis, žodyne turi būti numatyta galimybė saugoti informaciją apie realaus pasaulio objektų vardus, trumpinius ir panašiai.

Žodynas nebūtinai turi būti vienas ir universalus. Žodynas turi turėti galimybę būti praplėstas papildoma kitos kalbos lingvistine informacija.

4 OPERACIJOS SU TEIGINIAIS

Ankstesniuose skyriuose priėjome išvados, kad neapibrėžtus duomenis geriausia įsivaizduoti kaip visumą teiginių, kuriuos sudaro duomenų elementai. Duomenų elementai teiginiuose gali būti pateikiami įvairiai: techniniuose dokumentuose duomenys pateikiami tiksliai arba apibrėžiami aibėmis, intervalais; laiškuose, straipsniuose duomenys išreiškiami įvairiomis lingvistinėmis formomis, tokiomis kaip „ko gero“, „galbūt“ ir pan. Taigi, teiginių duomenų elementai priklauso nuo dokumentų žanro ir gali būti išreiškiami kaip fiksuoti dydžiai, intervalai arba aibės.

Atlikus lingvistinę neapibrėžtų duomenų transformaciją į teiginio modelį reikia aptarti galimas atlikti operacijas su duomenų elementais. Laikykime, kad mūsų kuriamo duomenų modelio tikslias – gauti iš neapibrėžtų duomenų tikslią, atrinktą bei savalaikę informaciją apie mus dominantį objektą. Kitaip tariant, mums reikalinga informacijos *atrinkimo* operacija, kurios rezultate: vartotojas gauna iš duomenų sankaupos atrinktą jam aktualią, naudingą informaciją; ta informacija yra savalaikė, t.y. nepasenusi; informacija yra tiksli, patikima ir neklaidinanti.

Operacijų su duomenų elementais rezultatas yra duomenų elementas, turintis tokią pačią struktūrą kaip ir operandai. Tai leidžia apdoroti neapibrėžtų duomenų elementus prieštaringos, nesuderintos, nevienareikšmiškos informacijos atvejais bei gauti vienareikšmišką rezultatą.

4.1.1 Loginės operacijos su laiko atributu

Atliekant atrinkimo operaciją vartotojui pateikiami rezultatai turi būti savalaikiai, tai yra būtinai reikia įvertinti laiko parametrą. Laiko parametras nusakomas turiniu bei tikėtinumu. Apdorojant teiginį laiko parametras gali saugoti informaciją apie laikotarpį, kuriuo aprašomas objektas turėjo saugomą reikšmę arba saugoti laiko momentą, kada buvo užfiksuotas šis teiginys.

„Rytoj perku automobilį (2006 m. vasario 18 d.)“.

Šiame teiginyje aptinkame du laiko parametrus: „rytoj“ ir „*2006 m. vasario 18 d.“*. Pirmas laiko parametras nurodo kada planuojamas teiginio faktas, o antras parametras nurodo skelbimo pasirodymo ar užfiksavimo datą. Pastarasis parametras yra reikalingas, kai yra ieškoma vėliausiai gauta informacija ir turėtų būti išreiškiama tik fiksuotu dydžiu, nes sistema automatiškai gali fiksuoti kada pasirodė skelbimas. Ši laiko reikšmė gali būti priskiriama prie teiginį charakterizuojančių parametrų, nes nurodo kada teiginys buvo suformuluotas.

Pirmasis laiko parametras „rytoj“ nusako, kuriuo laiko momentu teiginys bus teisingas. Šis parametras gali būti išreiškiamas fiksuotu dydžiu, aibe arba intervalu ir yra atributo charakteristika.

Kai laikas išreiškiamas fiksuotu dydžiu, jis nurodo tikslią datą, išreikštą vieninteliu laiko elementu. Yra galimas ir laiko išreiškimas elementų, einančių vienas po kito, aibe. Taip išreikštas laikas parodo, kad atributo turinio reikšmė galioja laikotarpiu, apibrėžtu laiko intervale („šiais metais“, „kitą savaitę“, „lapkritį“ ir pan.). Kartais laikas gali būti išreiškiamas tokiomis lingvistinėmis formomis, kurios negali tiksliai nurodyti laiko momento, kaip pvz. „neužilgo“, „ateityje“ ir pan. Laiko parametras gali būti išreiškiamas ir tokiais terminais, kaip „visada“ (tuomet laiko intervalas nustatomas nuo mažiausios galimos iki didžiausios galimos reikšmės), „niekada“ (laiko intervalas įgauna nulinę reikšmę), „dabar“ (teiginio užfiksavimo data) ir pan.

Kai prie atributo turinio laiko parametras yra nenurodomas – tuomet laiko parametras įgauna nulinę (neapibrėžtą) reikšmę.

Atliekant **logines operacijas** *(IR, ARBA, NE)* su laiko atributu, gaunamas tokios pačios struktūros atributas, kaip ir pirminių atributų. Panagrinėsime kiekvieną operaciją detaliau.

Loginės operacijos IR metu yra gaunamas laikas, kurio elementai priklauso abiems operacijoje dalyvavusiems laiko parametrams. Jei operacijoje laiko parametrai nesutampa, tai gausime nulinę laiko reikšmę, jei laiko parametrai turi tik vieną bendrą elementą, tai rezultatas bus fiksuotos reikšmės, jei turi daugiau nei vieną bendrą elementą – rezultatas bus išreiškiamas bendrų elementų aibe.

Jei laiko parametrai yra išreikšti *fiksuotais dydžiais*, tai atlikus *IR* operaciją, gaunamas fiksuotas laiko parametras, kai abu parametrai yra lygūs, ir neapibrėžtas laiko parametras, kai abu parametrai yra skirtingi:

- „gegužės 15 d.“ *IR* „kovo 27 d.“.

Atlikus *IR* operaciją šiuo atveju laiko parametras įgaus neapibrėžtą reikšmę, nes abiejų teiginių laiko parametrai nesutampa.

Jei laiko parametrai yra išreikšti *aibėmis*, operacijos *IR* rezultatas gali būti laiko elementų aibė arba fiksuotas dydis, gautas iš aibių sankirtos, arba neapibrėžtas laikas, jei operacijoje dalyvaujančios aibės nesusikerta:

- „šią savaitę“ *IR* „maždaug 15-16 gegužės mėnesio dienomis“.

Šiuo atveju operacijos *IR* laiko parametras įgaus reikšmę „15-16 d.“, jei 15 ir 16 dienos priklauso kažkuriai gegužės savaitei (kada buvo užfiksuotas teiginys) arba įgaus neapibrėžtą reikšmę priešingu atveju.

Jei atliekant *IR* operaciją nors vienas laiko parametras yra išreikštas *nuliniu dydžiu*, tai šios operacijos rezultatas bus neapibrėžtas dydis:

- „niekada“ *IR* „vakar“ – operacijos rezultatas įgaus neapibrėžtą reikšmę, nes parametras „niekada“ paneigia patį faktą.

Jei operacijos *IR* atlikimo metu vienas laiko parametras bus išreikštas elementų *aibe*, o kitas – *fiksuotu dydžiu*, tuomet rezultatas bus arba fiksuotas dydis arba neapibrėžta reikšmė, priklausomai nuo to, ar operacijoje naudojami laiko elementai sutampa ar ne:

- „šį mėnesį“ *IR* „*03.15 dieną*“.

Operacijos rezultatas įgaus „03.15“ reikšmę, jei teiginys užfiksuotas kovo mėnesį arba priešingu atveju bus išsaugota neapibrėžta reikšmė.

Loginės operacijos ARBA metu yra gaunamas laikas, kurio elementai priklauso nors vienam operacijoje dalyvavusiam laiko parametrui.

Jei abu laiko parametrai yra išreikšti *fiksuotu* dydžiu ir atliekant *ARBA* operaciją yra lygūs, tai rezultatas taip pat bus fiksuotas dydis. Priešingu atveju bus gaunamas rezultatas, išreikštas abiejų elementų aibe:

- „vasario 12 d.“ *ARBA* „vasario 15 d.“ operacijos rezultatas įgaus dviejų elementų aibę {„vasario 12 d.“, „vasario 15 d.“}.
- „šiandien“ *ARBA* „kovo 12 d.“ – rezultate gausime fiksuotą dydį „kovo 12 d.“, jei pirmasis teiginys bus užfiksuotas kovo 12 dieną.

Jei operacijos *ARBA* atlikimo metu vienas laiko parametras bus išreikštas elementų *aibe*, o kitas – *fiksuotu dydžiu*, tuomet rezultatas bus aibė visų skirtingų elementų iš abiejų laiko parametrų:

- „šiais metais“ *ARBA* „nuo 2006 metų“.

Operacijos rezultatas bus aibė {„šiais metais“, „nuo 2006 metų}, jei pirmasis teiginys užfiksuotas anksčiau nei 2006 metais, arba aibė {„nuo 2006 metų“} priešingu atveju.

Jei operacijoje abu laiko parametrai išreiškiami *aibėmis,* tai rezultatas bus aibė visų skirtingų elementų iš abiejų aibių:

- „vasario 15-23 d.“ *ARBA* „pavasarį“ – rezultate gausime aibę {„vasario 15-23 d., „pavasarį“}.

Jei *ARBA* operacijoje dalyvauja bent vienas *neapibrėžtas* laiko parametras, o kitas laiko parametras yra *fiksuotas* arba išreikštas *aibe*, tai operacijos rezultatas taip pat bus neapibrėžtas:

- „kažkada“ *ARBA* „šiais metais“ operacijos rezultatas bus „kažkada“.

Loginės operacijos NE rezultatas yra aibė visų likusių laiko elementų, atmetus elementą, kuriam buvo taikyta ši operacija, pavyzdžiui:

- *NE* „šiandien“ – šios operacijos rezultatas bus aibė laiko elementų be dienos, kada buvo užfiksuotas teiginio faktas.

4.1.2 Teiginių atrinkimo operacija

Bet kokioje duomenų bazėje informacijos yra saugoma daugiau, nei tuo metu jos reikia vartotojui. Vartotojas sistemai užduoda kriterijus, pagal kuriuos sistema turi atrinkti teiginius, saugančius vartotoją dominančią informaciją.

Kaip žinome, teiginio identifikatorius yra fiksuotas, konkretus dydis su tikėtinumu, lygiu 1, tai iš teiginių aibės atrenkame tik tuos teiginius, kurių objekto identifikatorius sutampa su vartotojo užduotu objekto identifikatoriumi. Teiginius atrinkti galima ir pagal ryšį tarp objekto identifikatoriaus bei atributo. Atlikus abi šias operacijas, gausime dvi teiginių aibes, kurių pirmoji bus sudaryta iš teiginių su vienodais objekto identifikatoriais, o antroji – iš teiginių su vienodais ryšio identifikatoriais.

Teiginius galima atrinkti ir taip, kad jų atributų charakteristikos tenkintų tam tikrus reikalavimus.

Teiginių atrinkimui naudojamos loginės operacijos *IR, ARBA, NE*. Jei šių operacijų metu gaunama tuščia aibė ar nulinė reikšmė, tai reiškia, kad nagrinėjamas teiginys netenkina vartotojo

užduotų sąlygų. Jei loginių operacijų rezultate gaunamas fiksuotas dydis arba aibė, vadinasi atributas tenkina užduotas sąlygas ir yra priskiriamas atrinktų atributų aibei.

4.1.3 Aritmetinės operacijos su teiginiais

Teiginiai gali būti naudojami skaičiavimuose. Sakykime, mes norime atlikti aritmetinę operaciją su tiksliais duomenimis. Prieš atliekant aritmetinę operaciją, visų pirmiausia reikia pervesti abu operandus į tą pačią išraišką, o šiuo atveju – tikslius duomenis į teiginio struktūrą. Panagrinėsime situaciją:

$Z = X + Y$, kur X – tiksli išraiška, o Y – gautas iš struktūrizuoto teiginio.

Pirmiausia X reikia pervesti į teiginį. Kadangi žinome, kad $X=63$, tai šią lygybę galime perfrazuoti į teiginį „X yra lygus 63“. Tuomet turėsime teiginį, kurio objekto identifikatorius bus X, ryšys tarp objekto identifikatoriaus ir atributo – „yra“, atributo reikšmė – „63“, atributo tikėtinumas – lygus 1, nes turime tikslius duomenis.

Kai turime abu operandus, išreikštus teiginiais, galime atlikti aritmetinę operaciją, kurios rezultatas Z – taip pat bus teiginys.

Kartais teiginyje tiesiogiai gali būti nenurodoma rodiklio reikšmė, o nusakomas jos pokytis arba kryptis. Tokiu atveju teiginyje ryšys gali būti išreikštas tokiomis reikšmėmis, kaip „pakils“, „sumažės“ ir panašiai, be to atributo turinio reikšmė taip pat gali būti išreiškiama santykiniu dydžiu procentais. Tokiu atveju, norint gauti rodiklio reikšmę, reikia žinoti pradinę jo reikšmę. Todėl atliekant operacijas su tokiais teiginiais reikia įvertinti laiko parametro reikšmę, parodančią laiko momentą, kada atributo turinys yra teisingas nurodytu atributo tikėtinumu.

5 EKSPERIMENTINĖ DALIS

Nestruktūrizuotų duomenų modelis gali būti naudojamas tose žmogaus veiklos srityse, kur yra betarpiškai susiduriama su nestruktūrizuota informacija. Tokio pobūdžio informacija gali būti gaunama elektroniniu paštu paprasto teksto formatu, parsisiųsta iš interneto, kitaip tariant ji gali būti bet kokio pavidalo ir patikimumo.

Norint nestruktūrizuotą informaciją apdoroti, ją reikia struktūrizuoti arba perkelti į įprastines duomenų bazes, pritaikytas saugoti nestruktūrizuoto pobūdžio informaciją. Kadangi žmogaus darbo veikloje didžioji dalis informacijos yra neapibrėžta, į duomenų bazes turi būti atrenkama ir perkeliama tik ta informacija, kuri yra aktuali sprendžiamai problemai spręsti – atrinkti teiginiai apie objektą.

Atlikdamas eksperimentinę tyrimo dalį nagrinėjau skelbimus apie parduodamus automobilius.

5.1 Informacijos surinkimas

Pirmiausia reikia surinkti iš įvairių informacijos šaltinių informaciją apie parduodamus automobilius. Skelbimuose informacija apie automobilius pateikiama nestruktūrizuota forma: fiksuotomis arba intervalinėmis reikšmėmis, su nurodytu laiko parametru iki kada galioja skelbimas, bei tikėtinumu, nurodančiu, kiek tikėtinas yra pats skelbimas.

8 lentelė. Analizuojami skelbimai

Skelbimo Nr.	Skelbimo tekstas	Šaltinis	Data
1	*Parduodu benzininį 2.0 ltr. 96 metų BMW automobilį, siūlyti nuo 1500 Lt. Kontaktai.*	A	gegužės 03 d.
2	*Parduodamas BMW benzininis automobilis, 95 metų, 1850 Lt. Kontaktai.*	B	04.23 d
3	*Parduodu 91-92 laidos BMW markės automobilį, siūlyti nuo 1000 Lt. Kontaktai.*	B	05.01 d.
4	*Parduosiu 3 klasės VW Golf TDI automobilį, 1994-1995 m. Kontaktai.*	B	04.15 d.
5	*Parduosiu nebrangiai benzininį BMW automobilį, 2000 Lt.*	B	05.02 d.
6	*Jonas galbūt parduos 1996 metų benzininį VW Golf už 2500-3000 Lt.*	C	2006.05.02
7	*Akcija-išpardavimas: paskutinis 92 metų dyzelinis BMW su 25% nuolaida! Kontaktai.*	A	balandžio 29 d.

8	*Skubiai parduodamas gerame stovyje BMW, benzinas, 95 metai, 3000 Lt. Kontaktai.*	B	04.25 d.

Skelbimų tekstas pateikiamas laisva šnekamąja kalba, todėl visų pirma kiekvieną skelbimą reikia išanalizuoti leksiniu požiūriu, išskiriant teiginius ir juos apibūdinančius elementus. Šalia kiekvieno skelbimo taip pat pridedama ir papildoma informacija, tokia kaip informacijos šaltinio patikimumo koeficientas, skelbimo atsiradimo data.

5.2 Nestruktūrizuotos informacijos apdorojimas

Iš nestruktūrizuota forma pateikto skelbimo teksto pirmiausia išskiriamas teiginio objektas, po to jį apibūdinantys elementai. Analizuojant tekstą dažnai tenka į pagalbą pasitelkti žodyną, padedantį teisingai įvertinti ir priskirti reikiamai kategorijai įvairiomis lingvistinėmis formomis, santrumpomis ar terminais išreikštus teiginio elementus:

- *„siūlyti nuo 1500 Lt“* – parodo, kad kaina nėra konkreti, o iš vienos pusės apibrėžtas intervalas $\{1500 \le X \le \infty\}$;
- *„91-92“* – čia apibrėžtu intervalu išreikšti automobilio pagaminimo metai $\{1991 \le X \le 1992\}$;
- *„3 klasės“, „TDI“* – čia turima omenyje VW Golf automobilio variklio klasė bei tipas;
- *„galbūt parduos“* – išreiškiama abejonė, netikrumas dėl automobilio pardavimo fakto;
- *„su 25% nuolaida“* – kaina išreiškiama procentine išraiška, tačiau tiksliai automobilio kainai nustatyti reikia žinoti pradinę automobilio vertę;
- *„skubiai parduodamas“* – čia išreiškiamas noras greitai parduoti automobilį, kas įtakoja skelbimo galiojimo terminus.

Skelbimai yra surinkti iš įvairių šaltinių, todėl reikia įvertinti kiekvieno šaltinio patikimumo laipsnį (skelbimas laikraštyje, interneto portale ir pan.). Skelbimo tekste taip pat gali būti išreikštas užtikrintumas ar abejonė, kas taip pat lemia skelbimo patikimumo laipsnį.

Įvertinus aukščiau aptartas situacijas ir apdorojus kiekvieną skelbimą, 9 lentelėje pavaizduoti skelbimai struktūrizuotoje formoje.

9 lentelė. Struktūrizuota informacija apie parduodamus automobilius

Nr.	Markė	Metai	Kaina	Kiti duomenys	Tiki-mybė	Data
1	BMW	1996	$\{1500 \le X \le \infty\}$	Benz., 2.0 ltr., kontaktai	0,8	2006-05-03
2	BMW	1995	1850	Benz., kontaktai	0,6	2006-04-23
3	BMW	1991-1992	$\{1000 \le X \le \infty\}$	Kontaktai	0,6	2006-05-01
4	VW Golf	1994-1995	-	Dyz., 3 klasė, kontaktai	0,6	2006-04-15
5	BMW	-	2000	Nebrangiai, benz.	0,8	2006-05-02
6	VW Golf	1996	$\{2500 \le X \le 3000\}$	Galbūt, benz., Jonas	0,4	2006-05-02
7	BMW	1992	1100	Dyz., kontaktai, 25% nuolaida	0,8	2006-04-29
8	BMW	1995	3000	Skubiai, gerame stovyje, benz., kontaktai	0,6	2006-04-25

5.3 Užklausos apdorojimas, duomenų atrinkimas

Tarkime, vartotojas nori pirkti 95-96 metų laidos automobilį ir yra pasiruošęs mokėti ne daugiau 3000 litų, dėl automobilio markės dar nėra apsisprendę, bet žino, kad tai turi būti benzininis automobilis. Vartotojas pageidauja matyti ne senesnius kaip dviejų savaičių skelbimus. Šiuo atveju mes turime kelis neapibrėžtumus – automobilio pagaminimo metai, kaina ir markė.

Vartotojo suformuluota užklausa yra išreiškiama teiginiu, kurio objektas yra „automobilis“, ryšys tarp objekto ir atributo „reikia“, „noriu“ ar panašiai. Atributo turinys parodo automobilio pagaminimo metus, pinigų sumą, kuri tenkina vartotoją, pageidaujamą variklio tipą bei laiko parametrą, nurodantį iš kokio intervalo galima atrinkti skelbimus.

Užklausa yra apdorojama naudojant neapibrėžtų duomenų modelį su galimybe operuoti intervalinėmis atributo turinio, laiko ir tikėtinumo reikšmėmis.

Turint parduodamų automobilių teiginius struktūrizuotoje formoje, jiems pritaikome atrinkimo pagal kelis kriterijus operacijas. Iš 9 lentelės atrenkame įrašus, tenkinančius vartotojo užduotus kriterijus ir rezultatus atvaizduojame 10 lentelėje.

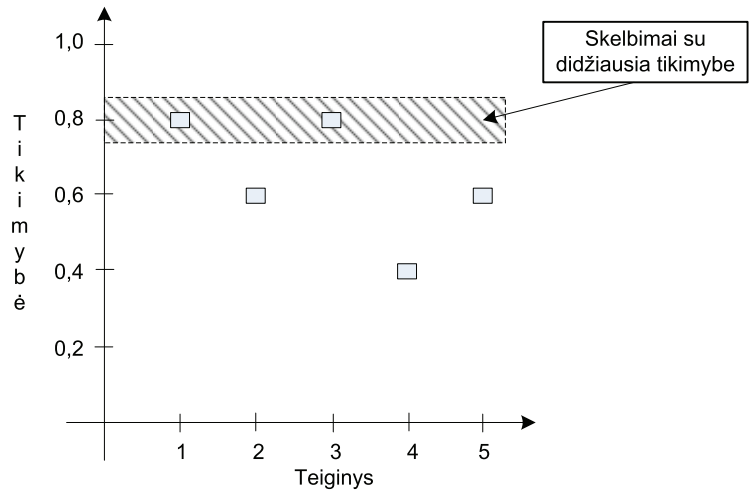
10 lentelė. Atrinkta informacija pagal vartotojo kriterijus

Nr.	Markė	Metai	Kaina	Kiti duomenys	Tiki-mybė	Data
1	BMW	1996	$\{1500 \le X \le \infty\}$	Benz., 2.0 ltr., kontaktai	0,8	2006-05-03
2	BMW	1995	1850	Benz., kontaktai	0,6	2006-04-23
3	BMW	-	2000	Nebrangiai, benz.	0,8	2006-05-02
4	VW Golf	1996	$\{2500 \le X \le 3000\}$	Galbūt, benz., Jonas	0,4	2006-05-02
5	BMW	1995	3000	Skubiai, gerame stovyje, benz., kontaktai	0,6	2006-04-25

5.4 Rezultatų įvertinimas

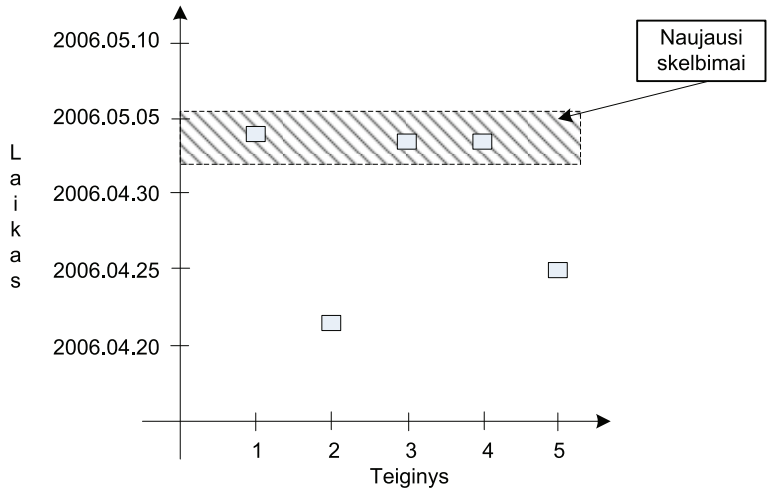
10 lentelėje pateikti pagal vartotojo užduotus kriterijus atrinkti teiginiai. Vartotojui pageidaujant, rezultatus galima atvaizduoti skirtingais pjūviais.

Vienas iš svarbiausių parametrų, padedančių vartotojui apsispręsti kurį rezultatą jam reikėtų peržiūrėti pirmiausia, yra *tikimybės* parametras. Kiekviename teiginyje yra saugojamas tikimybės koeficientas, nurodantis kiek yra tikėtina teiginyje įrašyta informacija. 13 paveikslėlyje atvaizduotas atrinktų teiginių tikimybių parametrų pasiskirstymas.



13 pav. Tikimybės reikšmių pasiskirstymas

Kitas ne mažiau svarbus parametras vartotojui, priimančiam sprendimą – *laiko* parametras. Kiekvienas skelbimas turi laiko parametrą, kuriame užfiksuota, kada skelbimas pateko į sistemą. Surūšiavus skelbimus pagal jų atsiradimo datą, vartotojui paranku visų pirma peržiūrėti naujausius arba tam tikru laikotarpiu paskelbtus skelbimus. 14 paveikslėlyje atvaizduotas atrinktų teiginių laiko parametrų pasiskirstymas.



14 pav. Laiko reikšmių pasiskirstymas

Teiginiuose gali būti ir kitos papildomos informacijos, kuri vartotojui gali būti aktuali priimant sprendimą, tokios kaip: „nebrangiai“, „skubiai“ ir pan. Sistemos vartotojui priimant sprendimą labai padės sistemos pateiktas kiekvieno teiginio tikėtinumo įvertinimas.

6 IŠVADOS

Atlikdamas šį darbą tyriau įvairius nestruktūrizuotų duomenų neapibrėžtumo, nepilnumo atvejus. Analizė parodė, kad bet kokio tipo neapibrėžtumas yra siejamas su informacijos trūkumu. Neapibrėžtumo atvejų specifikavimas sukuria prielaidas neapibrėžtų duomenų modeliui sukurti.

Nestruktūrizuotų duomenų turinio analizės metu išsiaiškinau teiginio sandarą, kaip struktūrizuojami teiginiai, aprašiau operacijas su teiginiais. Turint teiginius struktūrizuotoje formoje atsiranda galimybės teiginį saugoti ir apdoroti struktūrizuotų duomenų priemonėmis – įprastomis reliacinėmis duomenų bazėmis. Apibrėžtos operacijos su teiginiais įgalina apdoroti struktūrizuotą ir nestruktūrizuotą informaciją kartu.

Sukurtas neapibrėžtų duomenų modelis leidžia saugoti iš nestruktūrizuotos informacijos išgautą informaciją įprastinėse reliacinėse duomenų bazėse, atlikti įvairias operacijas su teiginiais. Modelis leidžia apjungti iš įvairių informacijos šaltinių surinktus duomenis apie teiginį, taip papildant teiginius trūkstamais duomenimis, kartu eliminuojant neapibrėžtumus.

Modelis buvo eksperimentiškai tirtas naudojant neapibrėžtų duomenų metodus, modeliuojant reikiamų skelbimų atrinkimą iš skirtingų šaltinių pagal tam tikrus kriterijus. Sukurtą modelį galima panaudoti tolimesniems nestruktūrizuotų duomenų tyrinėjimams, nes ši problematika yra labai aktuali.

7 NAUDOTA LITERATŪRA

[1] BERENDT Bettina, *Towards Semantic Web Mining* [interaktyvus]. [Berlin, Germany]: Institute of Information Systems, Humboldt University Berlin, 2002. Prieiga per internetą: <http://citeseer.ist.psu.edu/585344.html>.

[2] CRYSTAL David, *Introducing Basic Linguistics.* Penguin Books Ltd. 1992.

[3] DOUPNIK Timothy, RICHTER Martin, *Interpretation of uncertainty expressions: a crossnational study.* Accounting, Organization and Society. Vol. 28, No. 1, 2003.

[4] ELITH Jane, BURGMAN Mark, REGAN Helen, *Mapping epistemic uncertainties and vague concepts in predictions of species distribution.* Ecolohgical Modelling, 2002.

[5] GAROFALAKIS Minos N., *Very Large Databases* [interaktyvus]; Bell Laboratories, 1999. Prieiga per internetą: <http://citeseer.ist.psu.edu/25593.html>.

[6] KOSALA Raymond, *Web Mining Research: A Survey* [interaktyvus]. [Celestijnenlaan, Belgium]: Department of Computer Science Katholieke Universiteit Leuven, 2000. Prieiga per internetą: <http://citeseer.ist.psu.edu/kosala00web.html>.

[7] MCMAREN Lain, *Designing the Data Warehouse for Effective Data Mining* [interaktyvus]. [London, UK]. Prieiga per internetą: <http://citeseer.ist.psu.edu/35485.html>.

[8] MOENS David, VANDEPITTE Dirk. *A survey of non-probabilistic uncertainty treatment in finite element anglysis.* Computer Methods in Applied Mechanics and Engineering. Vol. 194, 2005.

[9] MYLOPOULOS John: *Information Modeling in the Time of the Revolution* [interaktyvus]. [Oxford, UK]: University of Toronto, 1998. Prieiga per internetą: <http://www.cs.toronto.edu/~jm/2507S/Readings/Survey.pdf>.

[10] RAUFASTE Eric: *Testing the descriptive validity of Possibility Theory in human judgments of uncertainty* [interaktyvus]. [France]: Université Toulouse, 2003. Prieiga per internetą: <http://portal.acm.org/citation.cfm?id=945947&coll=GUIDE&dl=GUIDE&CFID=45684382&CFTOKEN>.

[11] SHORTRIDGE Ashton, *Characterizing uncertainty in digital elevation models.* Spatial uncertainty in ecology: implications for remote sensing and GIS applications. Springer: New York, 2001.

[12] SMETS Philippe: *Imperfect information : Imprecision – Uncertainty* [Brussels, Belgium]: Université Libre de Bruxelles, 1997. Prieiga per internetą : <http://iridia.ulb.ac.be/~psmets/Imperfect_Data.pdf >.

[13] SMETS Philippe: *Probabilty, possibility, belief: which and where?* [Brussels, Belgium]: Universitre Libre de Bruxelles, 1998. Prieiga per internetą: <http://iridia.ulb.ac.be/~psmets/Prob_Poss_Bel.pdf>.

[14] TSAI PAuray: *Querying Uncertain Data in Heterogeneous Databases* [interaktyvus]. [Taiwan]: Department of Computer Science National Tsing Hua University, 1993. Prieiga per internetą: <http://make.cs.nthu.edu.tw/alp/alp_paper/QueryingUncertainDataInHeterogeneousDatabases.pdf>.