



VILNIAUS UNIVERSITETAS  
MATEMATIKOS IR INFORMATIKOS FAKULTETAS  
INFORMATIKOS INSTITUTAS

Magistro baigiamasis darbas

# **Ontologijomis grįstas vaistų rekomendavimo metodikos kūrimas**

Atliko:

Inga Jegelevičienė

parašas

Vadovas:

Dr. Linas Bukauskas

Vilnius  
2018

# Turinys

<b>Santrauka</b>	<b>3</b>
<b>Summary</b>	<b>4</b>
<b>Iyadas</b>	<b>5</b>
<b>1. Susijusių darbų analizė</b>	<b>8</b>
1.1. Semantinis tinklas ir jo savybės . . . . .	8
1.2. Ontologija ir jos savybės . . . . .	10
1.3. Ontologijų ir semantinio tinklo tarpusavio sąveika . . . . .	11
1.4. Pagrindinės semantinio tinklo technologijos . . . . .	11
1.4.1. RDF . . . . .	12
1.4.2. RDFS (RDF Schema) . . . . .	13
1.4.3. OWL . . . . .	13
1.5. Užklausų kalba SPARQL ir jos savybės . . . . .	14
1.6. Įgyvendinti modeliai . . . . .	15
1.6.1. Drugbank žinių bazė . . . . .	15
1.6.2. GalenOWL ontologija . . . . .	16
1.6.3. Vaistų ontologija lietuvių kalboje . . . . .	18
1.6.4. Vaistų suderinamumo sistema. Gintarinė vaistinė. . . . .	18
<b>2. Modelis</b>	<b>19</b>
2.1. Ontologijos modelis . . . . .	19
2.2. Formaliojo modelio funkcijos . . . . .	22
2.2.1. Tripletų paieška ontologijoje . . . . .	22
2.2.2. Žymės paieška ontologijoje . . . . .	22
2.2.3. Sinonimų paieška ontologijoje . . . . .	22
2.2.4. Hierarchinės struktūros paieška ontologijoje . . . . .	23
2.3. Vaistų rekomendavimo prototipo funkcijos . . . . .	24
<b>3. Sistemos įgyvendinimas</b>	<b>34</b>
3.1. Ontologijos projektavimas . . . . .	34
3.2. Vaistų ontologijos įgyvendinimas . . . . .	35
3.3. Vaistų ontologijos užpildymas duomenimis . . . . .	39
3.4. Modelio funkcijų ir sistemos įgyvendinimas . . . . .	41
<b>4. Metodų sudėtingumo vertinimas</b>	<b>44</b>
<b>Išvados ir rekomendacijos</b>	<b>46</b>
<b>Ateities darbų gairės</b>	<b>47</b>
<b>Literatūros šaltiniai</b>	<b>49</b>

## Santrauka

Šiame darbe pristatomos semantinio tinklo ir ontologijų technologinės galimybės, kurios, pritaikius atitinkamus programinius modulius, leidžia interpretuoti duomenis ir daryti išvadas ne tik žmogui, bet ir kompiuteriui. Apžvelgiami tokio tinklo veikimo principai bei panaudojimo galimybės globaliame semantiniame tinkle. Šio darbo tikslas panaudoti semantinio tinklo ir ontologijų technologijas farmacijos srityje. Analizuojamos jau įgyvendintos specifinių sričių ontologijos, pritaikyti metodai jas projektuojant ir įgyvendinat. Aprašomi ir sukuriami metodai, leidžiantys vykdyti paiešką farmacinės srities ontologijoje konkrečių vaistų kontekste. Sukuriamas ir aprašomas prototipas, vykdamas informacijos paiešką konkrečių vaistų kontekste arba pagal atitinkamus parametrus nustatant reikalingus vaistus. Įvertinamas modelio funkcijų vykdymo sudėtingumas. Apžvelgiamos tobulintinos vietos ir ateities darbų gairės.

**Raktiniai žodžiai:** semantinis tinklas, ontologijos, tripletai, OWL, RDF.

# Summary

## Research of a Method of Ontology-based Drugs Recommendation

This thesis presents the possibilities of the semantic network and ontological technologies, that if applied with certain software modules, could allow not only human beings but also computers to interpret data and come up with relevant conclusions. It examines the principles of how such network operates and what possibilities it could offer if it is used in the global network. The thesis aims to showcase how the semantic network and ontological technologies can be used in the field of pharmaceuticals. First, the paper reviews some cases of applied ontologies in other fields and analyses the methods used in their development and application. Following that, this work discusses the methods that could be implemented in the field of pharmaceuticals, particularly focusing on how it could allow to conduct the search of specific medicines. Furthermore, it describes the prototype that is specifically created in order to run the information search in the context of particular medicine or following certain parameters to identify required medicine. Besides, the paper evaluates how complex the performance of the model function is and identifies the areas for improvement and its possible applications in the future.

**Keywords:** Semantic web, ontologies, Triples, RDF, OWL.

## Ivydas

Kiekvienais metais farmacinėje pramonėje patvirtinama apie 50 [34] naujų vaistinių preparatų. Šie vaistai gali turėti naujas chemines savybes bei naujas nepageidaujamas sąveikas su kitais vaistais. Todėl patvirtinus vaistus, reikia atnaujinti ir vaistų, su kuriais atsiranda šalutiniai poveikiai, informaciją. Vien Lietuvoje egzistuoja apie 4000 skirtingų vaistų rūšių[2]. Todėl patvirtinus naują vaistinį preparatą (Lietuvoje), reikia atnaujinti informaciją su kitų 4000 vaistų informacija, jei egzistuoja kokia nors tarpusavio reakcija.

Remiantis 2015 metų statistikos duomenimis, Lietuvoje buvo parduota 72 332 200 farmacinių preparatų[2]. Statistiškai, vienam žmogui per metus teko apie 24 vaistinius preparatus (2015m. Lietuvoje buvo apie 2 921 000 gyventojų). Tai yra didelis vaistų kiekis, tenkančių vienam žmogui. Dažnai vienu metu vartojamas ne vienas medikamentas, o tai didina nepageidajamų šalutinių poveikių reakcijų tikimybę. Taip pat kelių vaistų vartojimas vienu metu gali būti neveiksmingas ar apskritai negalimas. Pavyzdžiui, vartojant geležies papildus mažakraujystės gydymui, kartu negalima vartoti kalcio ar cinko, nes kalcis ar cinkas ženkliai mažina geležies absorbciją. Tarp šių papildų vartojimo, rekomenduojama daryti mažiausiai dviejų valandų pertrauką[2].

Informacijos apie vaistus teikimui egzistuoja sistemų, gebančių atlikti paiešką pagal dominančią informaciją. VVKT egzistuojanti paieška vykdoma pagal medikamento pavadinimą, veikliąją medžiagą, ATC kodą ir panašią informaciją, leidžia gauti informaciją apie konkretų vaistą, gauti šių vaistų informacinių lapelių informaciją. Tačiau vykdyti vaistų palyginimą kelių vaistų kontekste ar vykdyti paiešką apribojant pagal medžiagas, kurios turėtų arba neturėtų egzistuoti vaistų sudėtyje, tokios paieškos sistemos šiai dienai nepavyko rasti.

Šio darbo tikslas yra sukurti vaistų rekomendavimo informacinės sistemos prototipinį modelį, kuris gebėtų panaudoti vaistų ontologijos semantinį tinklą paieškai konkrečių vaistų kontekste, taip pat išanalizuoti kelių vaistų tarpusavio sąveiką, atlikti paiešką pagal vaisto sudedamąsias medžiagas bei eliminuoti tuos vaistus, kuriuose egzistuoja nepageidajamų sudėtinių dalių. Tikslui pasiekti, keliami šie uždaviniai:

- Sukurti ontologiją, kuri aprašo vaistų informaciją. Talpina vaistų sudėtį, ATC vaistų klasifikacijos duomenis, farmacinę formą, ar vaistas yra receptinis, kokios vaisto indikacijos bei kontraindikacijos, šalutiniai poveikiai, kas yra vaisto gamintojas;
- Užpildyti vaistų ontologiją testiniais duomenimis pagal VVKT teikiamą informaciją;
- Sukurti sistemos modelį, gebantį atlikti semantinės agregacijos užklausas;
- Sukurti prototipinę vartotojo sąsają;
- Įvertinti prototipo galimybes ir tobulintinas vietas.

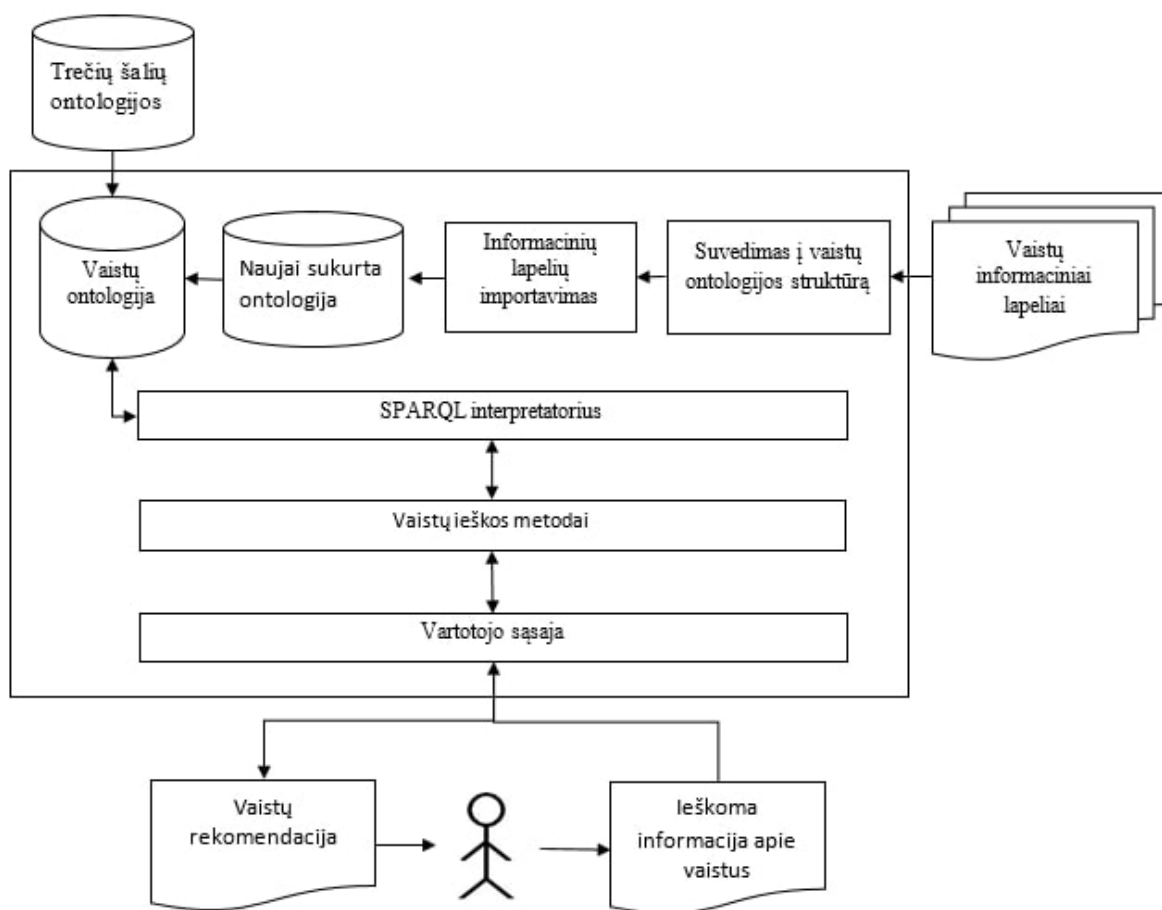
Sukurtu sistemos modeliu galės naudotis paprasti vaistų vartotojai, kurie rūpinasi savo sveikata. Vartotojai galės patikrinti ar vaistai tarpusavyje neturi nepageidajamų sąveikų, jeigu vartojama daugiau nei vienas vaistinis preparatas. Taip pat neretai pasitaiko netoleravimas specifinėms medžiagoms, kurios gali būti naudojamos vaisto sudėtyje. Vartotojas, žinodamas, kokia veiklioji medžiaga veikia slopina/gydo jo tuometinį negalavimą, ir žinodamas, kokio komponento negali būti sudėtyje, naudodamasis prototipine sistema gali atrasti jam tinkamą vaistinį preparatą.

Sistemos modelis taip pat gali būti naudojamas elektroninėje farmacijos prekyboje. Pavyzdžiui, vartotojui susidėjus vaistus į internetinį krepšelį, patikrinti, ar krepšelio turinys neturi tarpusavyje

nepageidaujamų sąveikų. Jeigu sąveikos aptinkamos—atitinkamai informuoti vartotoją apie krepšelio turinio nustatytas nepageidaujamas reakcijas ar rekomendacijas, kaip vaistus vartoti, kad būtų kuo saugiau vartoti išsirinktus vaistus.

Taip pat sistema galėtų naudotis gydytojais ar vaistininkais, skirdami pacientams vaistus. Ypač tinkama informacinė sistema gydytojams, kai pacientui skiriama ar pacientas jau vartoja ne vieną vaistinį preparatą, o skiriamas vaistas yra pakankamai retai pasitaikantis gydymo praktikoje, kad jo savybės būtų įsimintos dėl dažno naudojimo. Gydytojas ar vaistininkas sistemoje galės patikrinti ar skiriamas vaistas tinkamas pacientui. Sistema palengvintų darbą gydytojui ar vaistininkui, jam nereikėtų tiek atsiminti veikliųjų medžiagų ir kokias ligas/negalavimus veikliosios medžiagos veikia. Tokiu būdu galimai sumažinama žmogiškosios klaidos rizika.

Į modelį suvedus vaistų pavadinimus, kurie jau vartojami ir norima pradėti vartoti naujus vaistus, atliekama paieška SPARQL užklausų kalbos pagalba vaistų ontologijoje ir pateiktuose rezultatuose vartotojas gali matyti ar vaistai suderinami, ar nėra per didelis veikliųjų medžiagų sustiprėjimas, ar pradėjus vartoti naujus vaistus nepadidės šalutinio poveikio rizika. Sistemos modelio veikimo principas atvaizduojamas 1 pav., kuriame vartotojas įvedęs vaisto/kelių vaistų pavadinimą, gauna apie vaistus dominančią informaciją. Paveikslėlyje matoma struktūra susideda iš dviejų etapų. Tai būtų *duomenų surinkimas* ir *rezultatų gavimas*. *Duomenų surinkimo* etape surenkami



1 pav. Informacinės sistemos veikimo principas

vaistų informaciniai lapeliai VVKT[2] internetiniame puslapyje. Duomenys sunorminami ir suvedami į ontologijoje numatytą struktūrą. Pagal numatytas taisykles [24], duomenys suimportuojami

į vaistų ontologiją. *Rezultatų gavimas.* Vartotojas per vartotojo sąsajos prototipą pasirenka paieškos metodą ir įveda prašomą informaciją. Pasirinktas metodas vykdo SPARQL užklausą semantiniame tinkle (vaistų ontologijoje) ir metodas grąžina ontologijoje aptiktą vaisto informaciją. Išskaičiuotas rezultatas grąžinamas vartotojui.

Šio darbo struktūra yra: 1 skyrius. Susijusių darbų analizė—apžvelgtos technologijos, kurios pasirinktos įgyvendinti informacinės sistemos modelį. Taip pat apžvelgta kas yra ontologija ir semantinis tinklas bei jų savybės, kaip jie koreliuoja tarpusavyje, SPARQL užklausų kalbos ypatybės. Apžvelgiamos įgyvendintos ontologijos, kokie metodai pasirinkti įgyvendinant su vaistais susijusias ontologijas. 2 skyrius. Aprašoma ontologijos struktūra formaliąja kalba. Taip pat aprašomos ontologijų modeliavimo galimybės, aprašomi metodai, skirti vaistų informacijos paieškai farmacinės srities ontologijoje. 3 skyrius. Apžvelgiamas ontologijų projektavimo metodas, aprašoma įgyvendinta farmacinės srities ontologija, užpildymas duomenimis. Aprašomas įgyvendintas sistemos prototipas. 4 skyrius. Įvertinamas užklausų vykdymo sudėtingumas semantiniame tinkle.

# 1. Susijusių darbų analizė

Metodų įgyvendinimui buvo pasirinkta ontologijų ir semantinio tinklo koncepcija. Semantinis tinklas ir ontologijos yra technologinė sritis, skirta susieti duomenis prasmingais (semantiniais) ryšiais. Taip atsiranda galimybė vykdyti paiešką globaliame interneto tinkle ne tik pagal raktinius žodžius, kaip yra dabar didžiojoje dalyje duomenų internete pagal Web 2.0 standartus[6], bet ir pagal prasmę(Web 3.0 [6])[8, 33]. Semantinio tinklo duomenų modelis leidžia perskaityti, suprasti ir interpretuoti duomenis ne tik informacijos ieškančiam žmogui, bet ir sudaryti taip, kad tą gali padaryti ir programiniai agentai ar kiti programiniai moduliai.

Pradėjus analizuoti semantinio tinklo koncepciją, gretimai teikiama informacija ir apie ontologijas. Dažnai nėra aišku, kaip šie terminai tarpusavyje siejasi. Kodėl ieškant informacijos apie vieną dominantį šaltinį, dažnai paraleliai pateikiama ir apie kitą. Todėl šiame skyriuje aptariami ontologijų ir semantinio tinklo terminai bei jų tarpusavio sąveika.

## 1.1. Semantinis tinklas ir jo savybės

Nors vieningo aprašymo, kas yra semantinis tinklas, nėra, tačiau apibrėžimų koncepcija panaši. Semantinio tinklo apibrėžimą galima išskaidyti į dvi sąvokas[14], t.y. a) semantinis karkasas (angl. framework), kuris reprezentuoja duomenų prasmę, ir b) kurio duomenys yra internete ir dažniausiai naudojami tiesioginei paieškai internete. Literatūroje dažniausiai analizuojama pirmoji sąvokos dalis, ir dažnai antroji sąvokos dalis neakcentuojama. Iš kitos perspektyvos žiūrint, semantinį tinklą galima įvardinti kaip įvairių sričių ontologijų sąjungą į vieną didelį duomenų tinklą, kuris vadinamas globali duomenų bazė[13] internete. Semantinis tinklas taip pat yra RDF grafas [33], kur duomenų viršūnės atvaizduoja narius, o kraštai atvaizduoja prasmingus ryšius tarp šių narių. Žiūrint iš standartizavimo pusės—semantinis tinklas yra tam tikrų standartų rinkinys[16], kuris yra geriausia praktika dalintis duomenimis ne tik žmonėms, bet ir programiniams agentams ar kitiems programiniams moduliams. Semantinio tinklo sukūrimas inicijuotas tam, kad tiek žmonėms, tiek programinei įrangai nesunkiai galėtų naudotis duomenimis semantiniame tinkle, gebėtų juos analizuoti, interpretuoti ir daryti išvadas.

Kad būtų galima naudotis semantiniu tinklu pagal prasmę, duomenys turi būti siejami pagal tam tikrus kriterijus. Dėl šios priežasties paieška internete gali būti veiksminga ne tik pagal raktinius žodžius. Todėl semantiniame tinkle duomenys suvedami ir susaistomi, jeigu įmanoma, pagal tokius ryšius[23]:

- **Sinonimai**—sąvoka A išreiškiama taip pat, kaip sąvoka B. Pavyzdžiui: slopinti žodžio reikšmė yra tokia pat kaip malšinti; sieti reiškia tą patį, kaip ir jungti, rišti; pasibaigęs galiojimas yra sugedęs;
- **Antonimai**—sąvoka A išreiškiama priešingai, nei sąvoka B. Pavyzdžiui: pasibaigęs galiojimas antonimas yra šviežias; sveikti antonimas yra sirgti;
- **Meronimai**—Sąvokos A dalis yra sąvoka B. Pavyzdžiui: ibuprofenas (veiklioji vaisto medžiaga) yra dalis vaisto "Ibuprofen";
- **Holonimai** — Sąvoka A - turi dalį - sąvoką B. Pavyzdžiui(imant pavyzdžius iš meronimų): Vaistas "Ibuprofenas" turi dalį ibuprofeną.

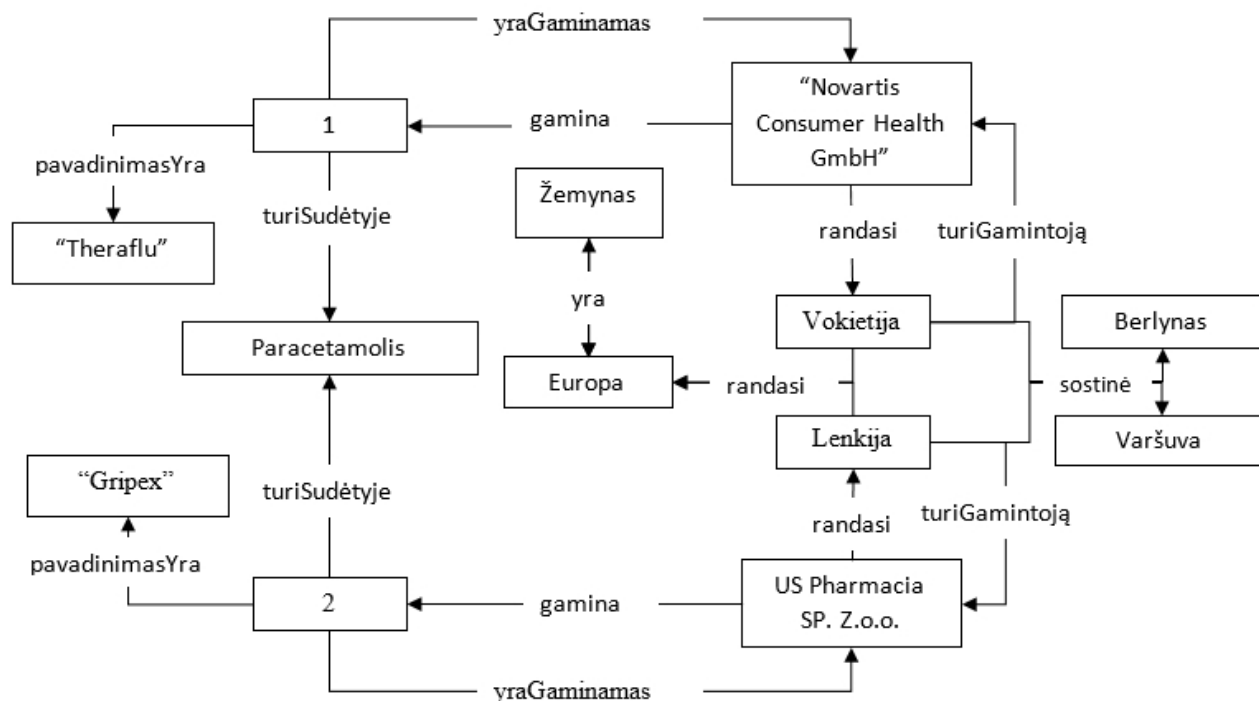


Tokie semantinio tinklo duomenų susiejimai išplečia paieškos gaunamus rezultatus, naudojant duomenis pagal prasmę. Vartotojas gali gauti tikslesnį paieškos rezultatą, nei tikimasi iš paieškos pagal raktinius žodžius.

Žiūrint iš techninės semantinio tinklo įgyvendinimo pusės, tai semantinio tinklo duomenys yra failai [14, 13], egzistuojantys internete, bei įgyvendinti semantinio tinklo programavimo kalbomis. Kitaip jie dar vadinami semantinio tinklo dokumentais (angl. Semantic Web Documents (SWD)). Tačiau tai neprivalo būti statiniai failai internete. Tai gali būti dinamiškai generuojami Web servais pagal duomenų bazėse egzistuojančius duomenis.

Semantinio tinklo duomenis taip pat galima išvaizduoti kaip internetinį puslapį [14], kuris adresuojamas pagal URL. Tuo pačiu tai yra ir RDF grafas, kur prasmingi ryšiai adresuojami atitinkamu URL, bei jais siejami semantinio tinklo duomenys. Platesnė RDF grafo apžvalga bus aptariama 1.4 poskyryje.

Semantinio tinklo fragmentas, kurio dalis įgyvendinta šio darbo metu, vaizduojamas 2 pav. Semantinis tinklas (didelis RDF grafas) sudarytas iš trumpų sakinių, vadinamų tripletais. Šie saki-



2 pav. Semantinio tinklo pavyzdys

niai turi veiksnį, tarinį ir papildinį. Semantinio tinklo smulčiausios ląstelės vadinamos atitinkamai subjektu (veiksny), predikatu (tarinys) ir objektu (papildinys). Analizuojant 2 pav. vaizduojamą semantinio tinklo fragmentą, tripletus galima išreikšti taip: subjektas – "1", predikatas – "pavadinimasYra", objektas – "Theraflu". Kitas tripletas: subjektas – "2", predikatas – "turiSudėtyje", objektas – "paracetamolis". Schemoje galima aptikti daugiau tripletų. Kaip minima [16] šaltinyje, į tripletus taip pat galima žiūrėti kaip į resurso aiškinimą. Imant ankstesnį pavyzdį: "1" (resursas) – "pavadinimasYra" (ypatybės (angl. properties) pavadinimas) – "Theraflu" (ypatybės reikšmė). Naudojantis tokiu semantiniu tinklu, galima gauti duomenų įvairiais pjūviais. Pavyzdžiui: "Kokie yra vaistai?", "Kokiuose vaistuose yra paracetamolio?", "Koks gamintojas gamina Theraflu?", "Kur vaistas gaminamas?" ir pan.

Semantiniame tinkle duomenys gali būti registruojami labai įvairūs. Tai gali būti įvairių sričių pavadinimai, žodynai, įvairių sričių hierarchijos, cheminės savybės, architektūriniai objektai, geografiniai objektai ir visokie kitokie duomenys, kokius tik galima įsivaizduoti. Semantinio tinklo nariai, apibrėžiantys tam tikrą dalykinę sritį, vadinami ontologijomis. Šis terminas detaliau bus aptariamas 1.2 poskyryje. Su semantinio tinklo ir ontologijų technologijų atsiradimu, įvairioms aplikacijoms atsirado galimybė vykdyti užklausas semantiniame tinkle, programinei įrangai daryti išvadas naudojantis tokiais duomenimis, atlikti semantinę analizę ir gautus duomenis interpretuoti.

Nors semantinės technologijos pradėtos plėtoti pakankamai seniai, tačiau šaltinyje [13] išvelgiami iššūkiai, dėl kurių sunku visiškai įgyvendinti semantinio tinklo architektūros aspektus:

- internete duomenys nuolat pildomi. Norint surasti semantinius duomenis pasirinktose svetainėse, juos sunku rasti, nes dažnai semantinio tinklo dokumentai paslėpti/sutapatinti su įprastais svetainės dokumentais;
- norint ištraukti semantinius duomenis iš pasirinktos svetainės reikia atlikti ne trivialius skaičiavimus. Pavyzdžiui, natūralios kalbos apdorojimas.

Todėl norint įdiegti semantinio tinklo technologijas bei padaryti semantinį tinklą globalia tinkline duomenų baze—būtina viešinti sukurtus ontologijų duomenis, padaryti juos prieinamus programiniams agentams ir suteikti leidimą jais naudotis kitoms ontologijoms.

## 1.2. Ontologija ir jos savybės

Kaip ir semantinio tinklo apibrėžimas, taip ir terminas "Ontologija" neturi vieno konkretaus apibrėžimo. Šis terminas taikomas keliose srityse [23], todėl ontologija gali būti apibėžiama:

- Ontologija—filosofijos moksle yra "būties teorija".
- Ontologija—dalykinės srities specifikacijos koncepcija (angl. explicit specification of a conceptualization). Tai dažniausiai pasitaikantis apibrėžimas informacinių technologijų srityje.
- Ontologija—žinių visuma, aprašanti tam tikrą duomenų sritį.

Suliejant antrąjį ir trečiąjį ontologijos apibrėžimą ir pritaikant jį informatikos moksle, galima teigti, kad ontologija apima tam tikrą duomenų sritį su visais galimais duomenų srities apribojimais. Ontologijų technologijos pagrindinis tikslas yra išskirti tam tikrą duomenų sritį bei ją aprašyti ontologijos modelyje, pritaikant pasirinktos srities apribojimus. Ontologijoms kurti naudojama specifinė ontologijoms skirta modeliavimo kalba, kuri yra lygiai tokia pati kaip ir semantiniam tinklui. Kadangi modeliavimo kalba, leidžianti aprašyti ontologų duomenis yra standartizuota [23], todėl ontologijų duomenys gali būti automatiškai interpretuoti ir apdoroti programinių agentų, programinių modulių bei žmonių.

Ontologijos sąvoka gali būti išskirstyta į keletą smulkesnių ontologijos sąvokų[23], kuri gali būti naudojama aprašant pasirinktos srities duomenis. Tai gali būti:

- Pagrindo ontologija (angl. top-level ontology, generic ontology) – šioje ontologijoje apibrėžiama pagrindinės ontologijos žinios, tokios kaip ontologijoje egzistuojantys ryšiai, savybės, klasės ir panaši informacija. Pagrindo ontologijos duomenis turi visos ontologijos.
- Srities ontologija (angl. domain ontology) – šioje ontologijoje aprašomi duomenys yra tam tikros srities. Pavyzdžiui, vaistų ontologija, terminų ontologija, vietovių ontologija, cheminių savybių ontologija ir pan.

- Aplikacijoms pritaikytos ontologijos (angl. application ontology) – tai yra tokios ontologijos, kurios pritaikytos specifinėms aplikacijoms.

Dažnai visi aukščiau aprašyti ontologijų apibrėžimai aptinkami konkrečiose įgyvendintose ontologijose. Visos be išimties ontologijos turi pagrindo ontologiją, įgyvendinama ontologija dažnai pritaikyta konkrečiai sričiai, bei dažniausiai ontologijos kuriamos konkrečioms aplikacijoms naudoti.

Po to, kai ontologijos yra įgyvendintos ir užpildytos duomenimis, jos gali būti naudojamos, siejamos su kitomis ontologijomis ir iš jų kuriamos naujos. Kelių ontologijų apjungimą lengviau įgyvendinti, kai tai numatoma dar konkrečios srities ontologijos kūrimo eigoje. Kuriant reikia laikytis taisyklių, kad vėliau būtų galima pateikti ontologijų analizatoriams ir šie galėtų interpretuoti sukurtą ontologiją. Apžvelgus nemažai literatūros, nėra vieningo metodo kaip reikia kurti ontologiją, tačiau straipsnyje [32] išskiriami šie pagrindiniai žingsniai, kurių reikėtų laikytis:

- Pasirinktos srities analizė ir žinių apie pasirinktą sritį surinkimas. T.y. pasirinktos srities (arba kelių sričių) žinių surinkimas ir jų klasifikavimas yra pagrindiniai naujos ontologijos kūrimo žingsniai;
- Ontologija struktūrizavimas. Šiame žingsnyje kuriama struktūrizuota ontologija pagal surinktos srities žinių grafą. Tai reiškia, kad nustatomos pasirinktos srities sąvokos, jų koncepcija ir susijusios reikšmės. Taip pat nustatomos, įvertinamos ir įgyvendinamos taksonomijos;
- Ontologijos suvedimas. Turi būti naudojama specifinė ontologijų įgyvendinimo kalba, kad suvesti duomenys ontologijoje būtų korektiški. Jeigu reikalinga—naudojama objektiškai orientuoti metodai, kad sukurta ontologija išpildytų pasirinktą duomenų sritį;
- Ontologijos vertinimas. Tai yra pati pagrindinė vieta, kurios metu įvertinama ir koreguojama ontologija. Įvertinus jau sukurtą ontologiją ir aptikus neatitikimų ją vis dar galima koreguoti.

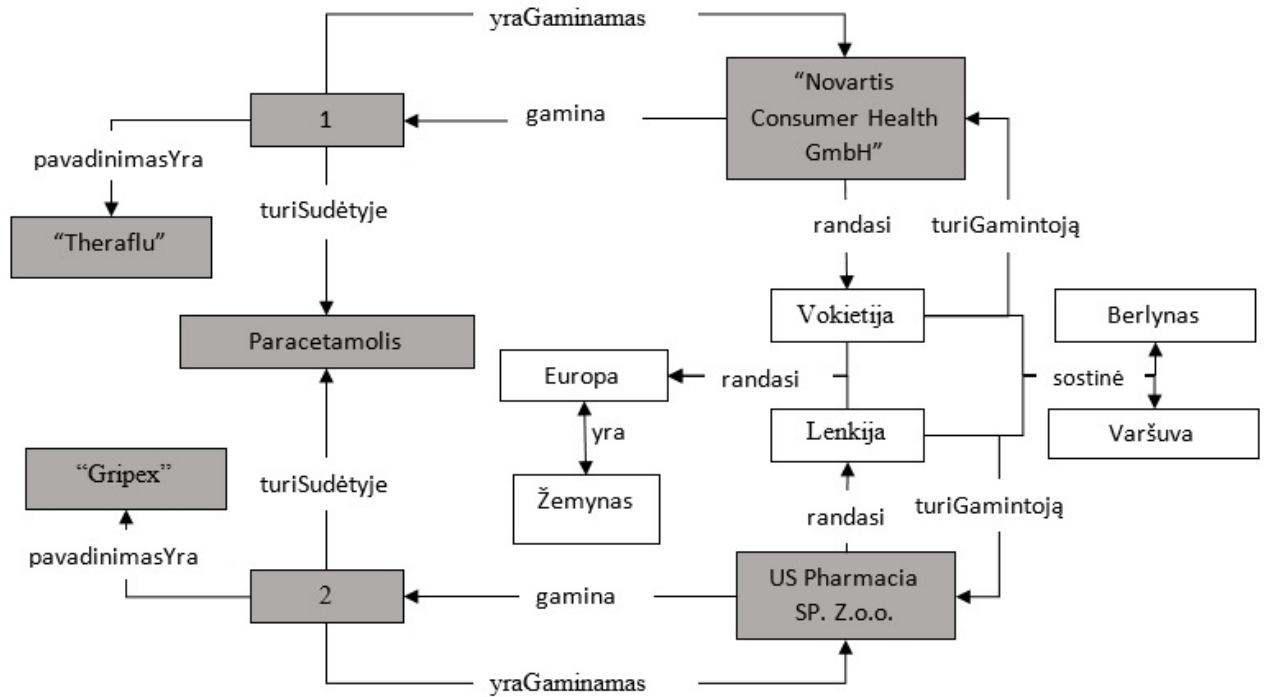
Kaip matoma aukščiau aprašytuose punktuose, ontologijų kūrimas yra iteratyvus procesas, leidžiantis pasiekti norimą rezultatą duomenų paieškai semantiniame tinkle/ontologijoje.

### **1.3. Ontologijų ir semantinio tinklo tarpusavio sąveika**

Semantinio tinklo ir ontologijų sąvokos susijusios taip, kad ontologija žymi tam tikrą duomenų sritį, o semantinis tinklas gali apimti vieną arba daugiau ontologijų, kurios tarpusavyje siejasi. Semantinis tinklas nėra ribojamas pagal konkrečią sritį, kai tuo tarpu ontologijos tiklas yra būtent duomenų surinkimas konkrečiai sričiai. Semantinis tinklas gali būti įgyvendintas tiek vienoje ontologijoje tiek kelių ontologijų rinkinyje. Pavyzdyje 3 pav. matomos dvi ontologijos, tai yra vaistų (pilki langeliai) ir vietos (balti langeliai). Šias ontologijas jungia vaistų ontologijos dalyvis "Gamintojas".

### **1.4. Pagrindinės semantinio tinklo technologijos**

Semantinio tinklo įgyvendinimo technologijų yra nemažai, tačiau šiame darbe bus apžvelgiamos dažniausiai naudojamos technologijos. Tai būtų RDF, RDFS ir OWL.



3 pav. Semantinio tinklo ir ontologijos koreliacijos pavyzdys

#### 1.4.1. RDF

RDF – resursus aprašantis karkasas (angl. Resource Description Framework), skirtas semantinio tinklo agentams keistis duomenimis internete. RDF turi labai ribotas galimybes modeliuoti semantinius tinklus, tačiau turi funkcijas, palengvinančias duomenų sujungimą, net kai schemas skiriasi. Naudojant semantinio tinklo technologijas, nereikia perrašyti visų duomenų, kad juos susaistyti tarp skirtingų ontologijų [18] ar modifikuoti įgyvendintas ontologijas. Visos kitos ontologijų modeliavimo kalbos dažniausiai remiasi RDF karkasu [11]. RDF naudoja nedidelį modeliavimo konstrukcijų rinkinį, pavyzdžiui: `rdf:Type`, `rdf:value`, `rdf:Resource`, `rdf:Description`, `rdf:datatype`, `rdf:Property`. Toks ontologijų modeliavimo rinkinys neleidžia pilnai aprašyti ir išpildyti taikymo srities žodyno, juo labiau išnaudoti hierarchinio modelio galimybių.

RDF ir RDFS dokumentus (RDF(S) ontologijas) galima išskirstyti į tris skirtingus abstrakcijos lygius [9]:

- sintaksiniu atžvilgiu—RDF ir RDFS dokumentai yra XML dokumentai, kurie gali būti skaitomi tiek programinių agentų tiek žmogaus;
- struktūriniu atžvilgiu—tai yra tripletų rinkinys dokumente, kur kiekvienas tripleto duomuo nurodo subjektą, predikatą ir objektą.
- semantiniu atžvilgiu—dokumentas sudaromas iš vieno ar daugiau grafų su iš anksto apibrėžta semantika.

Kaip buvo minėta anksčiau—semantinio tinklo ir ontologijų pagrindas yra tripletai. Informacinėse technologijose trumpi sakiniai, turintys tik veiksnį, tarinį ir papildinį, gali būti vadinami tripletais. Dar kitaip šie sintaksiniai objektai vadinami subjektu, predikatu ir objektu [16, 23, 14].

Veiksny/predikatas yra URI(angl. Uniform Resource Identifier) nuorodos. URI nuorodos yra prasmingi ryšiai, jungiantys semantinio tinklo narius subjektą ir objektą. URI nuorodos gali būti dalinamos į dvi semantikas [14]: a) unikalus identifikatorius tam tikriems šaltiniams ir b) internetinis adresas, kuris apibrėžia šaltinį SWD. Predikatas taip pat gali būti ir IRI (angl. Internationalized Resource Identifier). Šie du terminai atlieka tą pačią funkciją, tačiau URI gali būti aprašomas tik lotyniškais simboliais, o IRI gali būti aprašomas naudojant pasirinkto regiono raidyną. URI nuorodos naudojamos ne tik sieti duomenis vienos ontologijos kontekste, tačiau per URI nuorodas gali būti sujungiami dvi ar daugiau ontologijų į semantinį tinklą. Subjektas dažniausiai būna URI/IRI nuoroda [16], tačiau gali būti ir tuščia reikšmė (angl. blank node). URI nuoroda dažnai apibrėžiama klase arba individualas [16], kuris vėliau siejamas per predikatą su subjektu. Objektas gali būti taip pat URI nuoroda, tučias mazgas arba skaitinė/tekstinė (angl. literal) reikšmė [16]. Kai objektas įgauna skaitinę/tekstinę reikšmę, tai yra grafo galutinė viršūnė ir toliau ši reikšmė nebegali tapti subjektu kitame grafo triplete.

Pažymėtina, kad pats RDF nėra duomenų formatas. Tai yra duomenų modelis, kuriame galima pasirinkti sintaksę, kaip saugoti duomenis. Dažniausiai naudojamos serializacijos yra RDF/XML, OWL/XML (angl. Web Ontology Language), Turtle ir Notation3 (N3). RDF/XML ir OWL/XML remiasi tiesiogiai XML struktūra, todėl paprastam žmogui gali būti sunku perskaityti ontologijos duomenis. Turtle dažniausiai naudojama dėl to, kad atsidarius ontologijos dokumentą duomenų struktūra lengviausiai skaitoma žmogui, tačiau taip pat lengvai gali perskaityti ir programiniai moduliai ir agentai.

#### 1.4.2. RDFS (RDF Schema)

RDFS (RDF Schema) – labiau išplėstas ontologijų modeliavimo konstrukcinis lygis nei RDF. Atsiranda tokie ontologijos kūrimą įgalinantys terminai kaip *rdfs:label*, *rdfs:domain*, *rdfs:range*, *rdfs:subClassOf*, *rdfs:subPropertyOf*, *rdfs:Container* ir pan.

Metodai *rdfs:subClassOf* ir *rdfs:subPropertyOf* įgalina ontologijose naudoti duomenų taksonomiją. Ko pasekoje, galima naudoti objektinio programavimo principus[12], tokius kaip polimorfizmas ar paveldėjimas. Visi duomenų srities resursai gali būti išskaidyti į tam tikrą sritį apibrėžiančias klases. Klasės gali turėti daugiau išskaidančiųjų subklasių. Sub-klasės paveldi super-klasės savybes, o tai yra objektiškai orientuoto programavimo metodai. RDFS metodai *rdfs:subClassOf* ir *rdfs:subPropertyOf* yra tranzityvūs. Transityvumo sąvoka bus aptariama vėlesniuose skyriuose.

RDFS skiriasi nuo daugelio kitų panašių sistemų tuo, kad dažniausiai tose sistemose aprašomos terminų klasės pagal savybes, o RDFS aprašo terminų savybes pagal išteklių klases, kurioms taikomi[12]. Tai yra pagrindinė *rdfs:domain* ir *rdfs:range* semantinio tinklo žodyno savybė. RDFS leidžia apibrėžti nesudėtingą taikymo srities žodyną, tačiau vis dar negeba išreikšti duomenų be tiksliai aprašytos reikšmės. T.y. duomenys, naudojantys RDFS, vis dar yra be loginio duomenų išvedimo modelio.

#### 1.4.3. OWL

OWL (Web Ontology Language) – yra ontologijų kalba, kurta tiksliai ir kompleksiškai aprašyti žinias apie pasirinktas duomenų sritis, aprašyti sričių grupes ir ryšius tarp pasirinktų sričių. OWL yra apskaičiuojamąja logika grįsta ontologijų kalba. Todėl tokios ontologijos, išreikštos OWL, gali būti naudojamos kompiuterių programose, kurios naudoja išskaičiuojamą duomenų logiką bei prasmę. Tokias ontologijas kompiuteriai, naudodamiesi programiniais agentais ir moduliais, geba numanomas žinias paversti į aiškiai apibrėžtas žinias.

OWL skirstomas į tris dialektus [28, 23]:

- OWL Lite leidžia nesudėtingai aprašyti duomenis, pasiteliant paprastą hierarchinį modelį bei apibrėžti nesudėtingus apribojimus. Tai yra pagrindinis skirtumas tarp RDF ir RDFS. RDF ir RDFS neleidžia įgyvendinti net nesudėtingų apribojimų.
- OWL DL skirta aprašyti duomenų sritis aprašančiąja logika, kur galutinis rezultatas leidžia išlaikyti skaičiavimo baigtinumą ir išsprendžiamumą. Dėl šios priežasties programinė įranga gali numanomas reikšmes paversti į aiškiai apibrėžiamas reikšmes. Nors OWL DL apima visas OWL modeliavimo kalbos konstrukcijas, tačiau jas galima naudoti tik su tam tikrais apribojimais.
- OWL Full apima tiek OWL Lite tiek OWL DL konstrukcijas. Nors iš pažiūros atrodo, jog OWL Full yra daugiausiai galimybių turinti modeliavimo kalba, tačiau naudojant OWL Full duomenys gali tapti nenuosekliais ir iškyla samprotavimo klaidų tikimybė.

Dialektai yra išdėstyti tokia tvarka, jog kiekvienas yra prieš tai buvusio plėtinys[28]. Ne tik tai, kad OWL yra dažniausiai naudojama ontologijų modeliavimo kalba semantiniame tinkle, bet ir tarp OWL kalbų dažniausiai naudojamas OWL DL dėl logikos aprašymo galimybės.

Sugeneruoti OWL dokumentai (ontologijos) [14, 17], gali būti talpinami internete kaip paprasti failai. OWL failuose esantys duomenys gali būti susieti su kitais OWL dokumentais (ontologijomis) URI/IRI nuorodomis. Tokiu būdu sukuriamas semantinis tinklas, grįstas duomenų srities aprašančiąja logika. OWL yra dalis W3C semantinio tinklo technologijų, kuri įtraukia tokius ontologijų konstrukcinius modelius kaip RDF, RDFS.

## 1.5. Užklausų kalba SPARQL ir jos savybės

SPARQL (**SPARQL Protocol And RDF Query Language**) yra užklausų kalba, skirta vykdyti užklausas semantiniame tinkle. SPARQL užklausoje naudojami tripletų dalys—subjektas, predikatas, objektas. Kaip minima šaltinyje [25], iš pasirinkto duomenų šaltinio vykdoma užklausa, pačioje užklausoje atitinkamose vietose paskirstant paieškos atributus. Pagal šiuos duomenis gaunamas atitinkamas rezultatas.

SPARQL užklausoje galima naudoti šias užklausų formas[16, 28]:

- *SELECT* grąžina visus duomenis esančius ontologijoje, arba tuos duomenis, kurie buvo išfiltruoti pagal įrašytas sąlygas užklausoje šablone;
- *CONSTRUCT* grąžina grafą iš ontologijos, kuris buvo sukonstruotas užklausoje šablone;
- *ASC* grąžina loginio tipo reikšmę, kuri nusako ar egzistuoja faktinė informacija pagal sukonstruotą šabloną;
- *DESCRIBE* grąžina grafą su aptiktais resursais.

SPARQL užklausoje įgalina naudoti ne tik paprastus grafo dalyvius—subjektą, predikatą ir objektą, bet ir sudėtingus agreguotus užklausų šablonus, kurie atrenka atitinkamą informaciją.

Didžiąja dalimi SPARQL užklausų kalba panaši į reliacinių duomenų bazių užklausų kalbą. SPARQL užklausų kalba gali atlikti tam tikras duomenų projekcijas, išrinkti skirtingus įrašus, juos rūšiuoti, apriboti tam tikrą išrinktinių duomenų skaičių [25]. Pats semantinis tinklas iš dalies

primena reliacinių duomenų modelį [7], kur įrašas yra RDF grafo viršūnė–subjektas, stulpelio pavadinimas yra RDF grafo savybė (predikatas) ir lentelės įrašo celė yra reikšmė (objektas). Todėl tiek SPARQL tiek reliacinių duomenų bazių užklausų kalbos sąlyginai yra panašios. Reliaciniame modelyje formuojant užklausą nurodomi šaltiniai iš kur turi būti paimami duomenys. Semanti- niame tinkle dažnai duomenys ne tik nėra žinomi, tačiau nėra žinoma ir jų struktūra. SPARQL užklausų kalboje ieškomi duomenys yra formuojami tripletų pagalba *WHERE* dalyje. *SELECT* dalyje duomenys atvaizduojami iš *WHERE* dalies. O *FROM* dalis gali būti naudojama tada, kai paieška vykdoma konkrečioje ontologijoje ir yra žinoma ontologijos adresas. Jeigu naudojamas ontologijos apvalkalas (angl. endpoint), nurodoma *FROM* dalis užklausos rezultatams įtakos netu- ri. Naudojant apvalkalą, duomenys išrenkami iš tos ontologijos/semantinio tinklo dalies, kuri yra po apvalkalu.

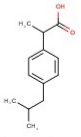

## 1.6. Igyvendinti modeliai

Šiame darbe nagrinėjamos ontologijos, kurios gali būti panaudotos ontologijų projektavime ir užklausų kūrime. 1.6.1 poskyryje aptariama žinių bazė, kuri buvo konvertuota į ontologiją. Ontolo- gijos ištekliai išoriniam tiesioginiam vartojimui nepasiekiami arba nepavyko rasti. 1.6.2 poskyryje įgyvendinta ontologija yra graikų kūrėjų. Greičiausiai todėl, tuos išteklius, kuriuos pavyko ras- ti, įgyvendinti graikų kalba. Lietuvių kalboje pritaikomas vaistų ontologijos karkasas aptariamasis 1.6.3 poskyryje.

### 1.6.1. Drugbank žinių bazė

DugBank (lietuviškai būtų "vaistų bankas") yra išsami vaistų žinių bazė, kurioje talpinama in- formacija ne tik apie vaistų struktūra/sudėtį, bet ir ką gydyti vaistai yra skirti. Žinių bazė pradėta plėtoti 2006 metais [34]. Žinių bazės pirmoji versija buvo skirta labiau kompiuteriniam simulia- vimui. Ji talpino tik pagrindinę vaistų informaciją. Kadangi žinių bazė pradėta plačiai naudoti, ji buvo plėtojama pagal maisto ir vaistų administracijoje (JAV, Kanada) patvirtintus/atmestus vaistus, tiek pati žinių bazės struktūra buvo išplėsta ir pradėta naudoti molekulinio lygiu surinkta vaistų informacija. Šaltinyje [5] minima, kad šiuo metu vaistų banke talpinama virš 8000 vaistų, kur didžioji dalis vaistų yra patvirtinti. Taip pat šioje žinių bazėje egzistuoja ir tokių vaistų, kurie yra neteisėti, panaikinti ar eksperimentiniai. Vaistų banko žinių bazė labiau skirta JAV ir Kanadoje parduodamiems vaistams. Paieškos rezultatuose kartais galima aptikti ir Europoje parduodamus vaistus. Tačiau Lietuvoje parduodamų daugumos "žymių" nereceptinių vaistų pagal vaisto pava- dinimą nepavyko rasti, nors straipsnyje [34] minima, kad vaistai surinkti pagal šiaurės Ameriko- je, Europoje ir Azijoje egzistuojančius vaistus. DrugBank paieškos sistemoje įvedus pavyzdžiui "Ibuprofen", paieškos rezultatuose pateikiama molekulinė išraiška, vaisto aprašymas, pateikiamos alternatyvos šiam vaistui, kokios kategorijai priklauso vaistas (pavyzdžiui: širdies ligų, skausmo malšinimo, nervų sistemos vaistai ir pan.), vaisto faktūra, indikacijos, poveikio trukmė ir panaši informacija. DrugBank dalis paieškos rezultatų pateikiama 4 pav.

Straipsnyje [34] atkreipiamas dėmesys į pašalintus vaistus ir į nelegalius vaistus. Minima, kad svarbu žinoti ne tik tokių vaistų/narkotinių medžiagų sudėtį ir kaip jie veikia, tačiau svarbu ir kaip jie suderinami su kitais vaistais/narkotinėmis medžiagomis. Todėl vaistų bankas tokius vaistus/narkotines medžiagas įtraukęs duomenyse, kad visada vartotojas galėtų surasti reikalingą informaciją.

Name	Ibuprofen
Accession Number	DB01050 (APRD00372)
Type	Small Molecule
Groups	Approved
Description	Ibuprofen, a propionic acid derivative, is a prototypical nonsteroidal anti-inflammatory agent (NSAIA) with analgesic and antipyretic properties.
Structure	 <div style="display: flex; justify-content: space-between; align-items: center;"> <span>🔍</span> <span>MOL</span> <span>SDF</span> <span>3D-SDF</span> <span>PDB</span> <span>SMILES</span> <span>InChI</span> <span>View 3D Structure</span> </div>
Synonyms	<ul style="list-style-type: none"> <li>Haltran</li> <li>Ibu-atritin</li> <li>Ibumetin</li> <li>Ibuprocin</li> </ul>
Categories	<ul style="list-style-type: none"> <li>• Acids, Carbocyclic</li> <li>• Analgesics</li> <li>• Analgesics, Non-Narcotic</li> </ul>
Weight	Average: 206.2808 Monoisotopic: 206.13067982
Chemical Formula	C <sub>13</sub> H <sub>18</sub> O <sub>2</sub>
Indication	For symptomatic treatment of rheumatoid arthritis, juvenile rheumatoid arthritis and osteoarthritis. May be used to treat mild to moderate pain and for the management of dysmenorrhea. May be used to reduce fever. Has been used with some success for treating ankylosing spondylitis, gout and psoriatic arthritis. May reduce pain, fever and inflammation of pericarditis. May be used IV with opiates to relieve moderate to severe pain. Ibuprofen lysine may be used IV to treat patent ductus arteriosus (PDA) in premature neonates.
Structured Indications 	<ul style="list-style-type: none"> <li>• Cystic Fibrosis</li> <li>• Fever</li> <li>• Gout</li> <li>• Headache</li> <li>• Juvenile Idiopathic Arthritis</li> <li>• Menstrual cramp</li> </ul>

4 pav. DrugBank paieškos rezultatas, pagal vaistą "Ibuprofen".

Drugbank yra viena didžiausių duomenų bazių vaistų srityje. Kaip minima straipsnyje [34], taip yra dėl to, kad:

- turi didžiausią kolekciją apie mažų molekulių vaistus;
- apima biotechnologijų vaistus ir gydomąsias medžiagas, gaunamas iš maisto;
- daugumoje kitų vaistų šaltinių, prie vaisto nurodomas tik pagrindinė vaisto paskirtis, o ne visos paskirtys;
- DrugBank yra labai detalus vaistų baltymų kompleksuose;
- vaistų banko anotatoriai išskira molekules, jei jos atlieka tam tikrą rolę vaistų sąveikoje.

Pati vaistų banko žinių bazė nėra ontologija. Ji neturi subjekto, predikato ir objekto. Taip pat duomenys tarpusavyje nėra susieti URI nuorodomis. Tačiau egzistuoja sistemų (pavyzdžiui: D2R Server [20]), kurios leidžia paversti ne ontologijų duomenis į susietus duomenis (angl. Linked Data). DrugBank, suvestų duomenų į RDF struktūrą, sistema [4] nebuvo naujinama nuo 2011 metų, todėl greičiausiai reali vaistų informacija kažkiek yra pasikeitusi. Pagal pateiktus [34] duomenis, kiekvienais metais patvirtinama apie 50 naujų vaistų.

### 1.6.2. GalenOWL ontologija

Ši ontologija duomenų prasme yra panaši į vaistų banko (aptartas 1.6.1 poskyryje). Ji sukurta graikų, todėl dauguma duomenų yra graikų kalba. GalenOWL papildomai pridėtos ligos, bei su



ligomis susijusi informacija. Kaip teigiama šaltinyje [15], tokia sistema labai naudinga, kai pacientui norima priskirti daugiau nei vieną vaistą. Tokiu atveju galima matyti su kokiais vaistais šis vaistas nėra suderinamas, ar įvedus kokioms sudedamosioms dalims vartojantysis gali būti alergiškas, parenkamas vaistas, neturintis alergizuojančių sudedamųjų dalių. Taip pat tokia ontologija naudinga, kai norima nustatyti kokias kūno vietas vaistai gali paveikti, jei, tarkim, keli vaistai turi tą pačią sudedamąją dalį ir ši sudedamoji dalis veikia tam tikrą organą. Tokiu atveju galima įvertinti rizikas arba pasirinkti kitą vaistą, stipraus kelių vaistų poveikio mažinimui konkrečiam organui. Taip pat GalenOWL naudoja vaistų su vaistais ir vaistų su ligomis sąveikas, jas aprašo tam tikromis galimomis OWL(angl. Web Ontology Language) taisyklėmis ir aksiomomis. GalenOWL yra įgyvendintas kaip interaktyvus servisas, kur galima gauti išsamius rezultatus apie vaistų su vaistais arba vaistų su ligomis sąveikų rezultatus.

Pagal statistinius duomenis [15], vyresni nei 55 metų vyrai turi vartoti daugiau nei keturis skirtingus medikamentus, kurių sąveikos reakcijas sunku prognozuoti. Kaip pateikta šio šaltinio pavyzdyje, vaistinis preparatas "Donepezilis" yra skirtas Alzheimerio ligai gydyti, sąveikauja su 9 kitomis medžiagomis ir 3 kitomis ligomis. Lietuviškame šio vaistinio preparato informaciniame lapelyje parašyta, jog šis vaistas gydo lengvos ar vidutinio stiprumo Alzheimerio ligą ir demencijos (racionalaus elgesio sutrikimui) simptomus. Trečiosios ligos, kaip [15] šaltinyje minima, nepavyko aptikti. Taip pat šis vaistinis preparatas negali būti vartojamas (arba turi būti pasitarta su gydytoju ar vaistininku) su NVNU (nesteroidiniai vaistai nuo uždegimo) vaistais, tam tikrais antibiotikais, tam tikrais vaistais nuo grybelio, tam tikrais raumenis atpalaiduojančiais vaistais, atidepresantais, vaistais nuo traukulių, vaistais širdies ligoms gydyti, kitais vaistais, kurie veikia panašiai kaip pavyzdyje minimas, vaistams atsmaili gydyti. Ir kaip minima šaltinyje [15], jei yra 18 tūkst. veikliųjų vaistų dalių, ir dar papildomai pagalbinių medžiagų, tai be semantinio tinko labai sunku sugaudyti tokią informaciją ir ja manipuliuoti.

GalenOWL sistemos unikalumas yra toks, kad ši ontologija leidžia gauti duomenis ne tik vaistų su vaistais (kaip yra daugumoje vaistų ontologijose), bet turi galimybę pateikti informaciją apie vaistų su ligomis sąveiką. Toks ontologijos išplėtimas yra labai naudingas, nes galima atrasti neigiamą poveikį, jei sergama ne viena liga iš karto. Kaip teigiama šaltinyje [15], į šią kategoriją (neigiama vaistų sąveika su kitomis ligomis) patenka tokie vaistai, kurie tenkina tokius punktus:

- jei vaistas, kuris gydo vienos ligos simptomus, tačiau gali pasunkinti kitos ligos simptomus;
- jei liga slopina/stiprina tam tikras vaistines savybes, ir dėl to gali kaupti organizme toksinus;
- jei vaistas turi tiksliai kontraindikacijas.

GalenOWL ontologija naudoja tam tikras ontologijas, kad būtų galima susisteminti tarptautiniu lygiu ligų ir vaistų terminologiją. Ligoms klasifikuoti naudojama TLK-10 (Tarptautiniai ligų kodai) sistema. Tai yra sistema, kuri talpina ligų kodus ir šie kodai naudojami visame pasaulyje. GalenOWL ontologijoje šie kodai naudojami aprašyti indikacijas ir kontraindikacijas. Taip pat GalenOWL ontologija naudoja UNII (Unique Ingredient Identifier) sistemą. Šioje sistemoje talpinamos unikalios vaistų sudedamosios dalys. Tai vėl gi naudojama taptautiniu lygiu identifikuoti vaistų sudedamąsias dalis. Šie duomenys naudojami indikacijoms ir kontraindikacijoms susijusioms su tam tikromis sudedamosiomis dalimis aprašyti. GalenOWL taip pat naudoja ACT (Anatomical Therapeutic Chemical) klasifikatorių. Šis klasifikatorius naudojamas panašioje srityje kaip ir UNII.

Taip pat GalenOWL ontologija naudoja aibę taisyklių kaip vaistus surišti, kad būtų lengva identifikuoti reikalingus duomenis. Įgyvendinant taisykles, reikalingas medicininis išsilavinimas

ar tiksliai vaistinių preparatų analizė, kad duomenys būtų susieti tinkamai. Šiame magistro baigiamajame darbe aprašomos taisyklės yra pakankamai abstrakčios, nes nėra surinkta pakankamai medicininių žinių apie tam tikras vaistų indikacijas ir kontraindikacijas ar vaistų sudedamąsias dalis. Be medicininio išsilavinimo nėra žinoma ar konkrečios sudedamosios dalys skirtos tai pačiai ligai gydyti ar simptomams malšinti, ar jos veikia tą patį simptomą vienodai, tačiau skiriasi jų pavadinimas.

### **1.6.3. Vaistų ontologija lietuvių kalboje**

Jau buvo pradėta kurti ontologija lietuvių kalbai, kuri minima šaltinyje [22]. Tačiau ji nebuvo vėliau plėtojama arba šios temos resursų nepavyko rasti. Be to, apžvelgiamas šaltinyje ontologijos karkasas yra labiau tiriamasis ontologijų karkasas, nei įgyvendinamas realus projektas. Šiame magistro baigiamajame darbe buvo panaudota dalis šaltinyje [22] ontologijos karkaso ir išplėstas pagal susijusių ontologijų kitose šalyse įgyvendintas ontologijas.

### **1.6.4. Vaistų suderinamumo sistema. Gintarinė vaistinė.**

Šiuo metu Lietuvoje egzistuoja sistema[31], įgyvendinta "Gintarinė vaistinė" iniciatyva Švedijoje, skirta vaistų tarpusavio koreliacijai nustatyti. Ši sistema įdiegta vaistinėse, todėl norint sužinoti ar vartojami vaistai tarpusavyje kaip nors sąveikauja, būtina nuvykti į vaistinę. Vaistinėje vaistininkas suveda vartojamų vaistinių preparatų pavadinimus ir pasidalina sistemoje aptikta informacija su pacientu. Sistema nėra prieinama paprastam vartotojui. Taip pat nebuvo rasta informacija, kokios technologijos ir kokie metodai naudojami kuriant sistemą, todėl ši sistema plačiau neapžvelgiama.

## 2. Modelis

Šiame skyriuje aprašomas modelis formaliąja kalba, kuriuo remiantis sukurta vaistų rekomendavimo sistemos prototipas. Modelio ir vaistų rekomendavimo sistemos funkcijų įgyvendinimą teoriniame lygmenyje, būtų galima išskirti į du etapus. Tai būtų ontologijos modelio galimybės, bei sistemos prototipo funkcijų aprašymas.

### 2.1. Ontologijos modelis

Kad būtų galima naudoti semantiniu tinklu, ontologijos duomenų struktūra bei joje esančiais duomenimis, turi būti apibrėžti metodai, įgalinantys tai padaryti. Šiame skyriuje aprašomos formaliosios modelio funkcijos ir apibrėžimai, kurie įgyvendinami ir aprašomi tolimesniuose skyriuose.

Semantinis tinklas( $\mathcal{S}$ )—tai globalus duomenų tinklas, talpinantis sričių ontologijas ir/arba aukštesnio lygio standartizuotas ontologijas. Ontologija ( $O_i$ ) galima vadinti duomenų aibę, kuri apibrėžia tam tikrą duomenų sritį, kai visas pasiekiamas ontologijas žymėsime  $O^*$ . Ontologijoje gali būti suprojektuota arba talpinama bet kokios dominančios srities informacija. Ji taip pat gali būti kaip atskiras objektas ir nepriklausyti semantiniam tinklui. Pagal šiuos teiginius, galima apibrėžti ontologijas ir semantinį tinklą.

**1 apibrėžimas.** (*Semantinis tinklas, Ontologija*) Tarkime,  $\mathcal{S}$ —semantinis tinklas.  $O_i, O_j$  yra tam tikrų sričių ontologijos. Galima teigti, jog egzistuoja tam tikrų sričių ontologijos, kurios gali priklausyti arba nepriklauso semantiniam tinklui  $\mathcal{S}$

$$\exists O_i, O_j | O_i \in \mathcal{S} \wedge O_j \notin \mathcal{S} \wedge O_i, O_j \in O^* \wedge O^* \in \mathcal{S}.$$

■

Kaip buvo minima ankstesniuose skyriuose, ontologija sudaryta iš tripletų  $t$ . Visi tripletai  $t$  priklauso tripletų aibei  $T$ . Tripletams priklausančios atskiros dalys gali būti [25]: URI/IRI nuorodos— $\mathbf{U}$ , tekstinės reikšmės— $\mathbf{L}$  (angl. literal) arba tuščios reikšmės— $\mathbf{B}$  (angl. blank nodes).

**2 apibrėžimas.** (*Tripletas*) Tarkime turime  $\mathcal{T}$ —tripletų aibę, kuri visada priklauso nors vienai ontologijai  $\mathcal{T} \in O$ . Tripletų aibę sudaro tripletai  $t$ , kurie yra apibrėžiami kaip

$$t ::= \{(s, p, o) | s \in (\mathbf{U} \cup \mathbf{B}) \wedge p \in (\mathbf{U}) \wedge o \in (\mathbf{U} \cup \mathbf{B} \cup \mathbf{L})\}$$

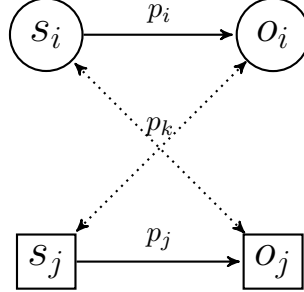
Čia  $s$ —subjektas,  $p$ —predikatas,  $o$ —objektas.

■

Tripletų rinkiniai  $\mathcal{T}$  tarpusavyje siejasi prasmingais ryšiais. Tripletų sudėtinės dalys subjektas ir objektas yra viršūnės, o predikatas yra prasmingas ryšys, siejantis viršūnes. Dėl šios priežasties ontologijoje esančius duomenis galima įvardinti grafu. Ontologijose įgyvendinti tripletai ir tripletų susiejimai prasmingais ryšiais, literatūroje vadinamas RDF grafu.

**3 apibrėžimas.** (*RDF grafas*) Tarkime,  $\mathcal{G}$  yra grafas, kur tripletų aibę  $\mathcal{T}$ , susidedanti iš tripletų  $t$ , sudaro grafą  $\mathcal{G}$ . RDF grafas turi viršūnes  $s, o$ , bei prasmingus ryšius  $p$ . Todėl grafą  $\mathcal{G}$  galime apibrėžti taip

$$\mathcal{G} ::= \forall t_i(s_i, p_i, o_i), t_j(s_j, p_j, o_j) \exists t_k(s_k, p_k, o_k) | t_i \neq t_j \wedge t_k \neq t_i \wedge t_k \neq t_j, \\ t_k(s_k, p_k, o_k) = ((o_i, *, s_j) \vee (o_j, *, s_i) \vee (s_j, *, o_i) \vee (s_i, *, o_j)) \wedge \\ t_i, t_j, t_k \in \mathcal{T}.$$



5 pav. RDF grafo fragmentas

Šiame apibrėžime teigiama, kad kiekvienam tripletui  $t_i, t_j$ , kurie tarpusavyje nėra lygūs, egzistuoja tripletas  $t_k$ , kur  $t_k$  viršūnės atitinka atitinkamas  $t_i, t_j$  viršūnes. Taip pat viršūnės jungiasi per bet kokį ryšį  $*$ , sudarydamos naują tripletą. Tripletų pavyzdinis grafas atvaizduojamas 5 pav. ■

Ontologijose tripletai susieti taip, kad išlaikytas duomenų vientisumas pagal prasmę. Vienas iš būdų suteikti ontologijai kuo daugiau galimybių duomenis nagrinėti pagal prasmę, o ne pagal raktinius žodžius, yra ontologijoje registruojami sinonimai. Vykdamt paiešką ontologijoje ir naudojantis sinonimų teikiama galimybė rasti semantiškai panašius žodžius–paieškos rezultatai išplečiami. Dėl to atsiranda galimybė surasti dominančią informaciją ne tik pagal įvestus raktinius žodžius, bet ir pagal prasmę.

**4 apibrėžimas.** (*Sinonimai*) Tarkime,  $S$ —sinonimų aibė. Tripleto prasmę žymėsime  $t'$ . Sinonimais apibrėžiamą ontologijos tripletų aibę galima išreikšti taip, kad egzistuoja tripletai, kurie lyginant morfologiškai tarpusavyje nėra lygūs

$$\exists t_i t_j | t_i(s_i, p_i, o_i) \neq t_j(s_j, p_j, o_j) \wedge i \neq j.$$

Taip pat jų atskiri nariai nėra lygūs  $s_i \neq s_j \wedge p_i \neq p_j \wedge o_i \neq o_j$ . Tiriant tripletus  $t_i, t_j$  pagal jų prasmę  $t'_i, t'_j$ , jie yra lygūs arba labai artimos reikšmės. Todėl galima apibrėžti, jog tripletų prasmės yra labai panašios arba vienodos

$$t'_i \simeq t'_j, t'_i, t'_j \in \mathcal{T}$$

arba tripletų sudedamosios dalys, RDF grafo nariai, yra prasmingai vienodi

$$s'_i \simeq s'_j \vee p'_i \simeq p'_j \vee o'_i \simeq o'_j.$$

Pagal 4-ą apibrėžimą išplaukia, kad tripletams užregistruoti sinonimai arba tripletų sudedamųjų dalių  $(s, p, o)$  sinonimai sudaro naujus tripletus. ■

Antonimų reikšmė ontologijoje taip pat labai svarbi. Analizuojant dominantę ontologijos užklausimą, gali prireikti duomenų, priešingų įvestiems kriterijams. Tokiuose atvejuose naudinga ontologijoje apsibėžti antonimus.

**5 apibrėžimas.** (*Antonimai*) Tarkime,  $A$ —antonimų aibė.  $W$ —traktuojama kaip paprastų žodžių aibė, tačiau ontologijoje apibrėžiama klasėmis ir būtinai yra URI nuorodos. Antonimais apibrėžiamą ontologijos tripletų aibę galima išreikšti taip, kad egzistuoja tripletai, kur patys tripletai nėra lygūs

$$\exists t_i, t_j | t_i(s_i, p_i, o_i) \neq t_j(s_j, p_j, o_j) \wedge i \neq j, t_i, t_j \in \mathcal{T}$$

taip pat tripletų sudėtinės dalys tarpusavyje nėra lygios

$$s_i \neq s_j \wedge p_i \neq p_j \wedge o_i \neq o_j; s_i, s_j, p_i, p_j, o_i, o_j \in W,$$

o jų prasmė yra priešinga, kuri šiame darbe žymima  $\doteq$  simboliu

$$t'_i \doteq t'_j; t_i, t'_j \in \mathcal{T}, \hat{t}'_j \in A$$

arba tripletų narių reikšmės priešingos

$$s'_i \doteq \hat{s}'_j \vee p'_i \doteq \hat{p}'_j \vee o'_i \doteq \hat{o}'_j; s'_i, p'_i, o'_i \in W; \hat{s}'_j, \hat{p}'_j, \hat{o}'_j \in A.$$

Nors antonimų suvedimas į ontologiją yra galimas, tačiau jis nėra taip plačiai paplitęs ontologijose kaip sinonimai. ■

Apibrėžiant hierarchinį modelį srities ontologijoje, naudinga apsirašyti klases ir poklasius. Holonimai atitinka klasės apibrėžimą taksonominėje struktūroje. Kaip ir OOP, klasės įgalina ontologijoje įgyvendinti paveldėjimą ir polimorfizmą.

**6 apibrėžimas.** (*Holonimai*) Tarkime,  $H$ —holonimų aibė. Kaip minėta anksčiau, holonimai atitinka klasės apibrėžimą ontologijos struktūroje.

$$H ::= \exists t(s, p, o), s \supseteq o.$$

Pagal holonimų apibrėžimą, galima daryti prielaidą, kad egzistuoja tripletas, kai jo nariai - subjektas ir objektas sudaro klasių hierarchiją. t.y. subjektas yra klasė, o objektas–poklasis. Ryšys  $\supseteq$  atitinka tripleto predikatą.

Taip pat paminėtina, kad holonimai gali būti tik URI nuorodos. Skaitinės, tekstinės reikšmės (**L**) arba tušti mazgai (**B**) gali būti klasės atributai, o ne klasės definicija. Todėl teigiama, kad holonimas visada bus URI nuoroda. ■

Meronimai hierarchiniame ontologijos struktūroje atitinka poklasio funkciją.

**7 apibrėžimas.** (*Meronimai*) Tarkime,  $M$ —meronimų aibė. Juos galima apibrėžti

$$M ::= \exists t(s, p, o), s \subseteq o.$$

Tai reiškia, kad ontologijoje egzistuoja tokie tripletai, kurie nusako ontologijos taksonominę hierarchiją, kur tripleto objektas yra subjekto poklasis (meronimas). Meronimų aibei ( $M$ ), taip pat kaip ir holonimų aibe ( $H$ ), galimos reikšmės yra tik URI nuorodos. Todėl apibrėžiant meronimus taip pat galime sakyti, kad meronimai taip pat visada yra URI nuorodos.  $M \exists! U$ .

## 2.2. Formaliojo modelio funkcijos

Šiame skyriuje aptariami metodai, skirti naudotis ontologijų ir semantinio tinklo galimybėmis sukurtoje farmacinės srities ontologijoje. Aptiriamos semantinio tinklo galimybių panaudojimo funkcijos formaliaja kalba.

### 2.2.1. Tripletų paieška ontologijoje

Kaip jau buvo minėta ankstesniuose skyriuose, ontologijose duomenys saugomi tripletų pavidalu. Taip pat primintina, kad visa ontologijos struktūra sudaro RDF grafą, kur viršūnės yra nariai ( $s$  ir  $o$ ), o kraštinės, jungiančios narius–prasmingi semantinio tinklo ryšiai ( $p$ ). Išraiškos rezultatas

$$\Pi_{\langle s,p,o \rangle}(O)$$

gražina visus galimus ontologijoje užregistruotus duomenis tiek aukštesnio lygio (struktūros) ontologijos duomenis, tiek duomenis, kurie suvesti į ontologiją konkrečiai sričiai. Taip sužinoma ontologijos struktūrą, kurios pagrindu galima įgyvendinti užklausas **SPARQL** užklausų kalbos pagalba.

### 2.2.2. Žymės paieška ontologijoje

Suvedant į ontologiją subjektus, predikatus ir objektus, turi būti išlaikomos **W3C** standartui priklausančios taisyklės. Pavyzdžiui: subjekto, predikato, objekto pavadinimai turi būti unikalūs, ir tik vėliau tarpusavyje jie susaistomi ryšiais pagal 1.1 skyriuje aprašytą sąrašą. Taip pat registruojamoms subjekto, objekto ir predikato reikšmėms turi būti pritaikytos atitinkamos simbolikos. Pvz.: subjekto pavadinimas turi būti be tarpų, prasidėti didžiąja raide, o kiekvienas naujas pridėtinis žodis taip pat prasideda didžiąja raide. Predikato reikšmė turi prasidėti mažąja raide ir kiekvienas sudurtinis žodis turi prasidėti didžiąja raide. Todėl tiksliai, vartotojui suprantama informacija apie šaltinį dažnai talpinama *rdfs:label*(1 lentelė) ar *rdfs:comment*(2 lentelė) ryšiu į ontologiją.

1 lentelė. *rdfs:label* (A)

Pavadinimas	<i>rdfs:label</i>
vo:1	"ACC Long"
vo:3	"Ibuprofen"

2 lentelė. *rdfs:comment* (B)

Pavadinimas	<i>rdfs:comment</i>
vo:2	"Faringospray"
vo:4	"Gripex Thermal"

Šiuos prasmingus ontologijos ryšius galima pažymėti *specify*, kur  $specify ::= A \cup B$ . Todėl norint gauti, pavyzdžiui, ontologijoje aprašytos klasės pavadinimą, pagal žinomą kintamąjį ( $var$ ), galima vykdyti tokią užklausą:

$$FindSubject(var) ::= \Pi_{?x} \sigma_{\langle ?x, specify, var \rangle}(O). \quad (2.1)$$

Čia  $var \in (U \cup B \cup L)$ . Tokia projekcijos struktūra gali gražinti klasės pavadinimo rezultatą.

### 2.2.3. Sinonimų paieška ontologijoje

Yra žinoma, kad sinonimai aprašomi ryšiu *owl : sameAs*. Pavadinkime šį ryšį tiesiog *sameAs*. *SameAs* turi simetriškumo savybę, kai  $A \rightarrow B$  yra tas pats kaip  $B \rightarrow A$ . Simetriškumo savoka bus

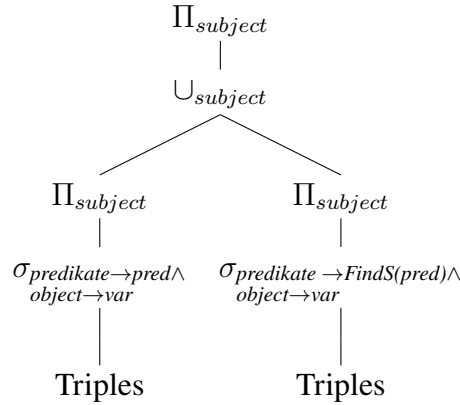
apžvelgiama vėlesniuose skyriuose. Taip pat pažymima, jog  $\sqsubseteq$  aprašo racionalų teksto lyginimą (angl. regular expression). Sinonimų paieška ontologijoje vykdoma tokiu principu

$$FindS(x) ::= \Pi_{?syn} \sigma_{?obj \sqsubseteq x \bowtie (?syn, sameAs, ?obj)}(O).$$

Norint ontologijoje nustatyti tam tikrą ieškomą informaciją, pakanka žinoti ryšį (predikatą), kuriuo nariai (subjektas ir objektas) jungiasi. Jeigu ontologijoje aprašyti sinonimai, galima žinoti arba numanyti artimą predikatui reikšmę. Taip išplečiami paieškos rezultatai pagal prasmę. T.y.

$$FindBI(pred, var) ::= \Pi_{?subj} \sigma_{?obj \sqsubseteq var}^{\langle ?subj, pred, ?obj \rangle \cup \langle ?subj, FindS(pred), ?obj \rangle} (O)$$

Tokią išraišką galima išplokštinti ir aprašyti diagrama, kuri matoma 6 pav. Į šią užklausą paduodami du kintamieji:  $var$  ir  $pred$ . Kintamasis  $var$  nurodo dominantį šaltinį(objektą), apie kurį ieškoma informacija, o kintamasis  $pred$  nurodo prasmingą ryšį, kuriuo kintamasis  $var$  jungiasi su subjektu. Tokią išraišką galima vadinti **bazine išraiška**.



6 pav. Bazinė išraiška, surandanti  $subject$ , kur ryšiui  $pred$  surandami sinonimai, o objektas prilyginamas kintamajam  $var$ .

#### 2.2.4. Hierarchinės struktūros paieška ontologijoje

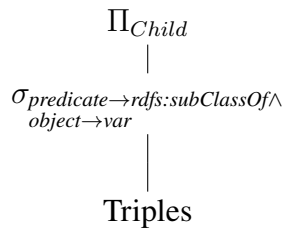
Informacijos paieška, kuri yra suvesta hierarchinėje struktūroje, šiame darbe vadinama **hierarchinės struktūros išraiška**. Tokioje išraiškoje dalyvauja predikatas  $rdfs : subClassOf$ . Prie kiekvieno tripleto, kuris užregistruotas hierarchinėje struktūroje, yra nuoroda (predikatas), kuris rodo į vaikinio resurso viršūnę. Todėl norint vykdyti paiešką, žinant aukščiausią grandį, galima vykdyti tokią užklausą:

$$FindChild(var) ::= \Pi_{?Child} \sigma_{(?Child, rdfs:subClassOf, var)}(O)$$

Čia  $var$  yra ieškomasis, kuris yra tėvinis resursas. Aiškumo dėlei pateikiamas šios išraiškos išplokštintas variantas, matomas 7 pav.

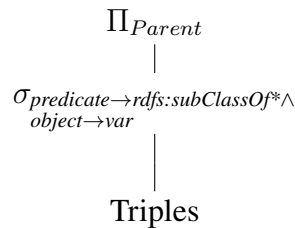
Norint vykdyti paiešką leidžiantis aukštyn nuo viršūnės( $Child$ ) į viršūnę ( $Parent$ ), naudojamas ryšys  $rdfs : subClassOf*$ . Simbolis  $*$  nusako, kad turi būti kylama aukštyn hierarchine struktūra, įtraukiant ir ieškomą duomenį. Taip pat, norint surasti tik tėvinius resursus naudojamas  $rdfs : subClassOf+$ . Simbolis  $+$  nusako, jog turi būti surastas tėvinis resursas, neįskaitant ieškomo duomens. Norint surasti tėvinį resursą, gali būti naudojama tokia užklausa:

$$FindParent(var) ::= \Pi_{?Parent} \sigma_{(?Parent, rdfs:subClassOf*, var)}(O)$$



7 pav. Surandamas tėvinis resursas

Šios *FindParent* hierarchinės struktūros funkcijos išplokštintas variantas atvaizduojamas 8 pav. Čia kintamasis *var* nurodo vaikinį resursą, kur paieška vykdoma tėviniam resursui surasti.



8 pav. Surandamas vaikinis resursas

Pritaikant bazinės išraiškos bei hierarchinės struktūros modelį, galimos įvairios informacijos paieškos vaistų ontologijoje. Todėl toliau aprašomi modeliai, skirti konkrečios informacijos apie vaistus paieškai.

### 2.3. Vaistų rekomendavimo prototipo funkcijos

Šiame poskyryje bus apžvelgtos funkcijos, kurios aktualios vaistų rekomendavimo sistemos prototipui. Rekomendavimo sistemoje bus naudojamos tokios funkcijos:

1. **Nustatyti vaistų sudėtį.** Vaistų sudėtis visada pateikiama vaistų informaciniuose lapeliuose. Vartotojui visada suteikta galimybė peržiūrėti tiek vaisto veikliąsias medžiagas, tiek pagalbines. Tokia informacija svarbi, norint įsitikinti, jog preparatą vartoti saugu, ypačingai esant netoleravimui konkrečioms vaistų sudėtinėms dalims. Pavyzdžiui, vienas dažniausiai pasitaikančių netoleravimų yra pieno laktozei (hipolaktazija). Todėl vaistų, turinčių sudėtyje pieno laktozės, net ir nedidelį kiekį, vartoti nepatartina. Įdomu yra tai, kad laktozė pakankamai dažna vaisto sudėties pagalbinė medžiaga. Todėl vaisto vartotojas turi įdėmiai skaityti vaistų informacinį lapelį, arba pasitikėti rekomendavimo sistemos teikiama informacija.

Pagal tai, ko reikia vartotojui, vaisto sudėties paiešką galima vykdyti šiais būdais:

- (a) Ieškoti vaisto sudėties veikliųjų medžiagų. Šis metodas leidžia vartotojui žinoti vaisto sudėties veikliąsias medžiagas. Tokia informacija pakankama, jei vaisto vartotojas neturi netoleravimo tam tikroms medžiagoms;
- (b) Ieškant visos vaistų sudėties. Į šį užklauso rezultatą įeina tiek veikliosios medžiagos tiek pagalbinės medžiagos. Tokia informacija naudinga vartotojui, turinčiam netoleravimą tam tikroms medžiagoms.

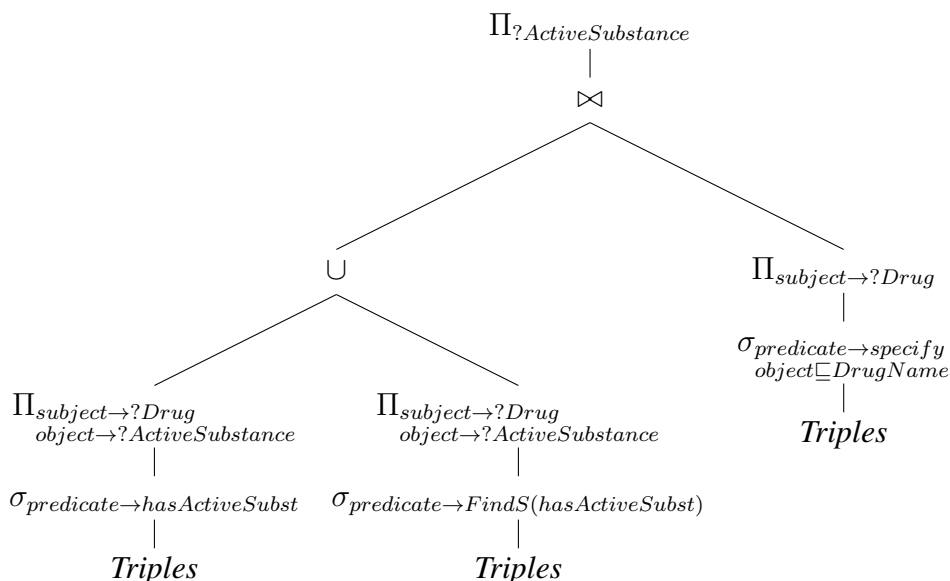
Ieškant ontologijoje vaisto veikliosios medžiagos, predikatas turėtų būti turiVeikliąjąMedžiagą (angl. *hasActiveSubstances*). Todėl analogiškai turėtų būti vykdoma užklausa tokiu



principu, kaip pavaizduota  $FindAS1$  ( 2.2) funkcijoje.

$$FindAS1(DrugsName, hasActiveSubst) ::= \Pi_{?ActiveSubst} \sigma_{?Drug \sqsubseteq DrugsName \bowtie \langle ?Drug, FindS(hasActiveSubst), ?ActiveSubst \rangle \cup \langle ?Drug, hasActiveSubst, ?ActiveSubst \rangle} \quad (O) \quad (2.2)$$

Naudojant 2.2 išraišką, tokiu pačiu principu galima nustatyti vaisto pagalbinės medžiagos, kai  $hasActiveSubst$  nusako vaisto pagalbinių medžiagų ryšį. Užklauso vykdyimo medis pavaizduotas 9 pav. Į funkciją perduodamas parametras  $DrugName$ , nurodantis ieškomo vaisto pavadinimą. Kitas perduodamas parametras–turiVeikliąjąMedžiagą ( $hasActiveSubst$ ). Parametras nurodo predikatą, kuriuo subjektas turi jungtis su objektu, kai objektas yra  $DrugName$ .



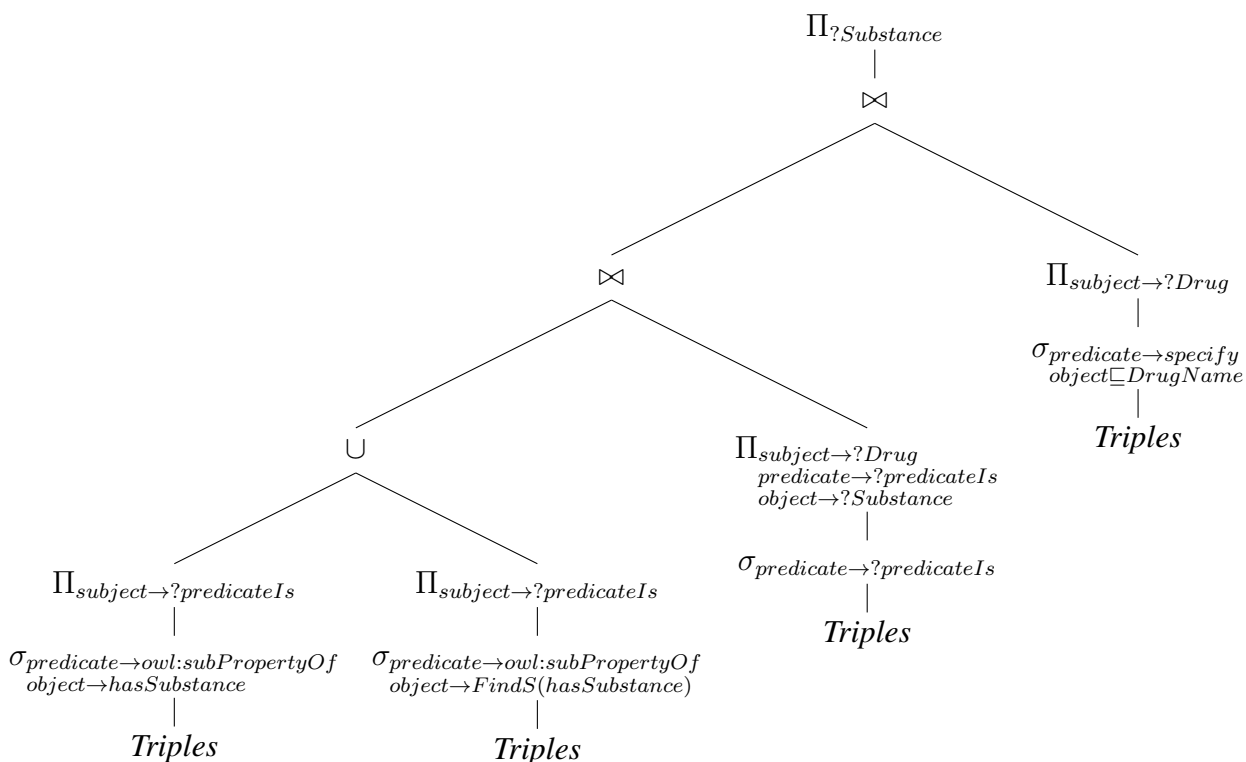
9 pav. Užklauso, kai ieškomos vaistų veikliosios medžiagos, vykdyimo medis

Kai norima žinoti visą vaisto sudėtį, naudojamas kitoks metodas. Savybės (angl. Properties), taip pat kaip ir klasės gali būti įgyvendinamos hierarchinės struktūros principu. Tikėtina, jog nustatant vaistų sudėtį, pagalbinės medžiagos ir veikliosios medžiagos savybės turės tėvinę savybę, kurią pavadinkime turiMedžiagą( $hasSubstance$ ). Todėl aprašant kitą metodą, skirtą gauti visos vaistų sudėties informaciją, naudojamas būtent tokiu metodu . Funkcija  $FindAS2$ ( 2.3) suranda visas pasirinkto vaisto sudėtines medžiagas.

$$FindAS2(DrugsName) ::= \Pi_{?Subst} \sigma_{?Drug \sqsubseteq DrugsName \bowtie \langle ?Drug, ?pred, ?Subst \rangle \bowtie \langle ?pred, rdfs:subPropertyOf, hasSubst \rangle \cup \langle ?pred, rdfs:subPropertyOf, FindS(hasSubst) \rangle} \quad (O) \quad (2.3)$$

Funkcijai perduodamas tik vienas parametras– $DrugName$ . Vaistų sudėtis randama per savybės "turiSudėtyje"( $hasSubstance$ ) ryšį, leidžiantis hierarchine struktūra žemyn. Tokiu būdu aptinkamas ryšys "turiVeikliąjąMedžiagą" ir "turiPagalbinęMedžiagą". Šių galimų predikatų sąjungos pagalba surandama vaistų pilna sudėtis. Užklauso vykdyimo medis vaizduojamas 10 pav.

Taip pat galima medžio transformacija–vietoj medžio šakų(nuo  $\cup$  žemyn), kur naudojama predikato reikšmė  $subPropertyOf$ , naudoti aprašytą funkciją  $FindChild(hasSubstance)$ , kas iš esmės yra ekvivalentu.



10 pav. Užklauso, kai ieškoma pilna vaisų sudėtis, vykdymo medis

2. **Nustatyti vaistų indikacijas ir kontraindikacijas.** Dažnai pasitaiko, kad skaitant vaistinių preparatų informacinius lapelius, praleidžiama informacija apie indikacijas ir kontraindikacijas. Besąlygiškai pasitikima gydytojo ar vaistininko žiniomis apie rekomenduotą vaistą. Tačiau niekas nėra apsaugotas nuo žmogiškojo klaidos faktoriaus. Todėl tokia informacija yra ypatingai svarbi, jei apie tam tikrus negalvimus gydytojui ar vaistininkui nebuvo užsiminta arba esant keliems simptomams iš karto. Įvedus vaisto pavadinimą, pateikiamas sąrašas indikacijų ir kontraindikacijų, kuriuos vartotojas perskaitęs gali įsitikinti, jog vaistą vartoti, kiek tai įmanoma, yra saugu. Kilus įtarimams, jog vaistai priskirti netinkamai, galima pasitikslinti su gydytoju ar vaistininku.

$$FindI(\begin{matrix} DrugsName, \\ indicationIs \end{matrix}) ::= \Pi_{?indication} \sigma_{?Drug \sqsubseteq DrugsName \times \left( \begin{matrix} (?Drug, indicationIs, ?indication) \cup \\ (?Drug, FindS(indicationIs), ?indication) \end{matrix} \right)} \quad (O)$$

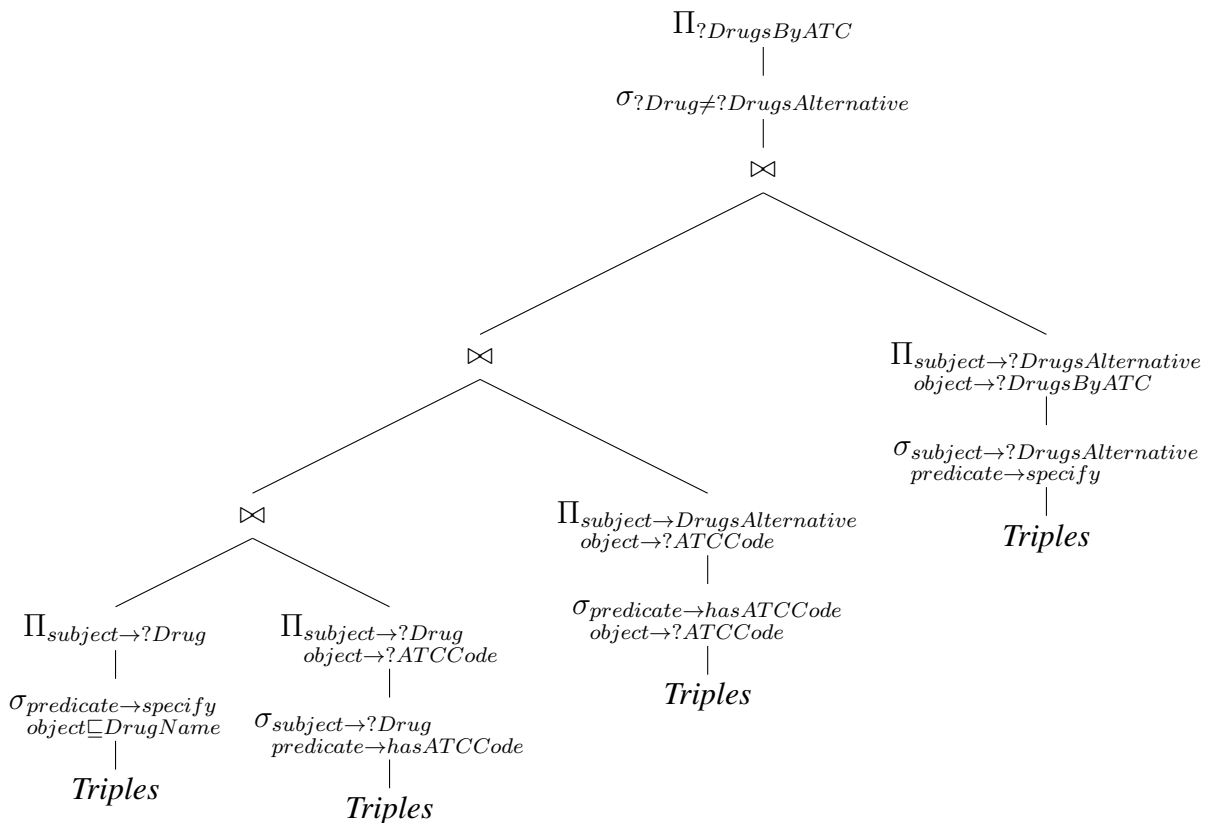
Užklauso vykdymo medis yra toks pat, kaip 9 pav. pavaizduotas užklauso medis. Į užklausą perduodamas kitas predikatas, kuris nurodo indikacijos arba kontraindikacijos prasmingą ryšį.

3. **Pagal vaisto ATC klasifikacijos kodą, surasti tuo pačiu kodu egzistuojančius vaistus.** ATC klasifikacija yra tarptautiniu mastu naudojama vaistų klasifikacija. Vaistams suteikiamas kodas, kuris nurodo, kokią organizmo sistemą ar organą vaistas gydo/malšina simptomus. Todėl ir užsienyje galima nusipirkti vaistinių preparatų, žinant vartojamo vaisto ATC klasifikacijos kodą. Analizuojant VVKT pateiktus duomenis apie vaistus, pastebėta, kad homeopatiniai vaistai ATC kodo neturi. Dėl šios priežasties toks metodas nėra patikimas 100 procentų, nes duomenų apie homeopacinius vaistus ir jų suklasifikuoti pagal ATC klasifikaciją nėra įmanoma. Metodas tinkamas tik cheminių vaistų informacijai gauti. Funkcija,

grąžinanti einamajam vaistui alternatyvas pagal ATC klasifikaciją, vaizduojama 2.4 pav.

$$FindByATC( DrugsName ) ::= \Pi_{? DrugsByATC} \sigma_{? Drug \neq ? DrugsAlternative} \times \begin{matrix} \langle ? Drug, specify, DrugsName \rangle \times \\ \langle ? Drug, hasATCCode, ? ATCCode \rangle \times \\ \langle ? DrugsAlternative, hasATCCode, ? ATCCode \rangle \times \\ \langle ? DrugsAlternative, specify, ? DrugsByATC \rangle \end{matrix} \quad (O) \quad (2.4)$$

Užklauso vykdymo medis vaizduojamas 11 pav. Šis užklauso vykdymo medis leidžia įvertinti užklauso, kur surandamos pasirinkto vaisto alternatyvos pagal ATC klasifikaciją, sudėtingumą.



11 pav. Užklauso vykdymo medis, kai parenkamos alternatyvos pagal ATC klasifikaciją.

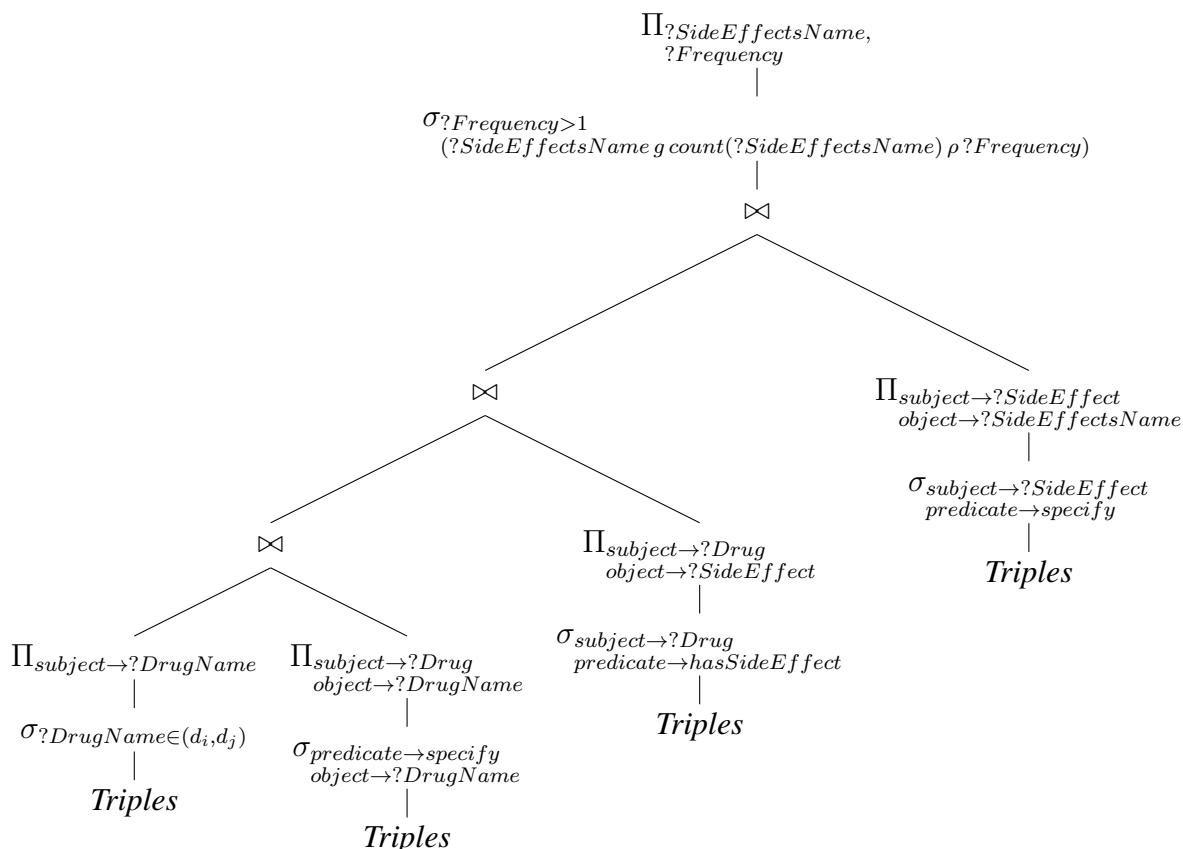
4. **Surinkti vaisto galimus šalutinius poveikius.** Šalutiniai poveikiai yra neatsiejama vaisto informacijos dalis. Vartotojas, vartodamas vaistą, prisiima riziką dėl šalutinių poveikių organizmui atsiradimo. Kartais vartojamo vaisto šalutinio poveikio rizika būna didesnė, nei ligos simptomų gydymo nauda. Ypatingai, jei vartotojo organai paveikti skirtingų ligų. Žinant galimas šalutinio poveikio rizikas, galima apsispręsti ar vartoti vaistinį preparatą, ar ieškoti alternatyvų.

Šalutinio poveikio nustatymo modelį būtų galima išskaidyti į keletą smulkesnių modelių:

- (a) **Nustatyti įvesto vaisto galimus šalutinius poveikius.** Modelio funkcija nustatanti tiesiogiai vartojamo vaisto galimus šalutinius poveikius, kurie užregistruoti su vaistais susijusioje informacijoje. Užklauso vykdymo medis yra analogiškas 9 pav. pavaizduo-

tam medžiui. Čia perduodami parametrai vaisto pavadinimas ir ryšys turiŠalutinįPoveikį (*hasSideEffect*). Jeigu toks ryšys aprašytas įprastai arba per sinonimus–vartotojui atvaizduojami visi užregistruoti šalutiniai poveikiai.

- (b) **Lyginant vaistų porą, nustatyti galimus šalutinio poveikio sustiprėjimus.** Vartojant ne vieną vaistinių preparatą iš karto, gali pasitaikyti tokių vaistinių preparatų, kurių galimi šalutiniai poveikiai persidengia. Taip išauga tikimybė, jog vartojamų keletu vaistų kombinacija sukels konkretų šalutinį poveikį. Todėl naudinga peržiūrėti galimus šalutinių poveikių persidengimus ir įvertinti riziką. Šalutinio poveikio užklausa vykdomo medis, aptinkantis galimus vaistų šalutinių poveikių sustiprėjimus, pavaizduotas 12 pav. Į šią užklausą perduodami du parametrai  $d_i, d_j$ . Jie atitinka vaisto pavadinimus,



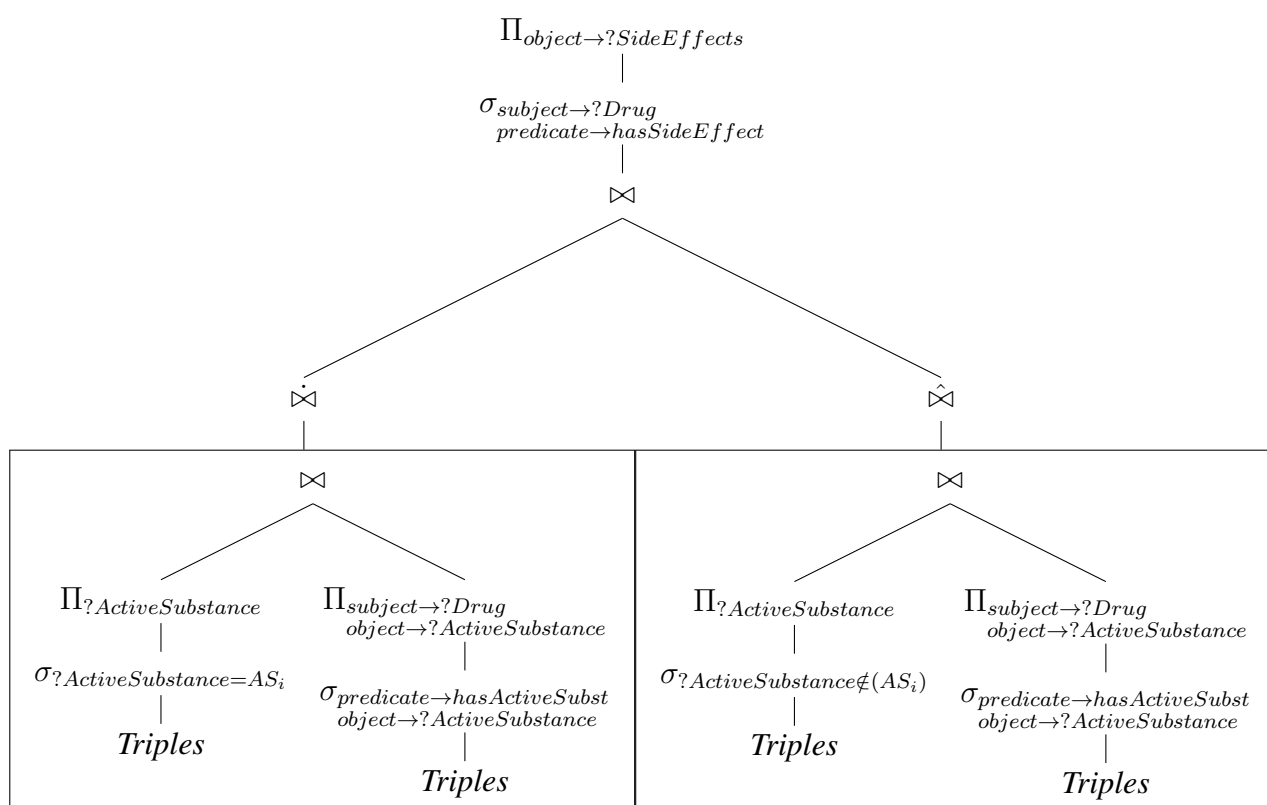
12 pav. Užklausa, kai ieškomas šalutinių poveikių persidengimas, vykdomo medis

kuriems ieškomi galimi šalutinių poveikių persidengimai. Taip pat šioje užklausoje pagrindinis ryšys, nusakantis šalutinio poveikio vaistui priskyrimą, aprašomas predikatu turiŠalutinįPoveikį (*hasSideEffect*). Rezultate gaunami tie šalutiniai poveikiai, kurie pasikartoję abiem vaistams priskirtoje informacijoje.

- (c) **Pagal veikliąją medžiagą surinkti galimus šalutinius poveikius.** Gali būti, kad dalis vaistų nėra pilnai ištirti vaistų kūrėjų laboratorijose, ir tik patekus į rinką, pamatomi realūs galimi šalutiniai poveikiai. Kartais vaistų informaciniuose lapeliuose būna įrašyta, jog vaistui patekus į vartojamąją rinką, buvo užregistruoti tam tikri šalutiniai poveikiai. Analizuojant veikliųjų medžiagų sukeltus šalutinius poveikius, galima numanyti nepageidaujamų poveikių riziką. Pavyzdžiui, vaistui A nurodyti X ir Y šalutiniai poveikiai, o vaistui B tik Y, nors vaisto sudedamosios veikliosios dalys tos pačios. Todėl galima teigti, jog vartojant vaistą B, galimi tokie patys šalutiniai poveikiai kaip

ir vaisto A.

Vienos iteracijos šalutinių poveikių persidengimui surasti nepakanka. Reikia panaudoti anksčiau aprašytą funkciją 2.2 pav., kuri surenka pasirinkto vaisto veikliąsias medžiagas. Vėliau pagal šias veikliąsias medžiagas surandami kiti vaistiniai preparatai ir apjungiami visi šalutiniai poveikiai. Būtina sąlyga, jog veikliosios medžiagos būtų identiškos įvestam vaistiniam preparatui, nes egzistuojant vaisto sudėtyje nors vienai kitokiai veikliajai medžiagai, ji gali sukelti nepageidaujamą šalutinį poveikį, kuris užregistruotas susijusioje vaisto informacijoje. Todėl 13 pav. vaizduojamas būtent toks užklausos medis, kuris suranda analogiškomis veikliosioms medžiagoms galimus šalutinius poveikius.

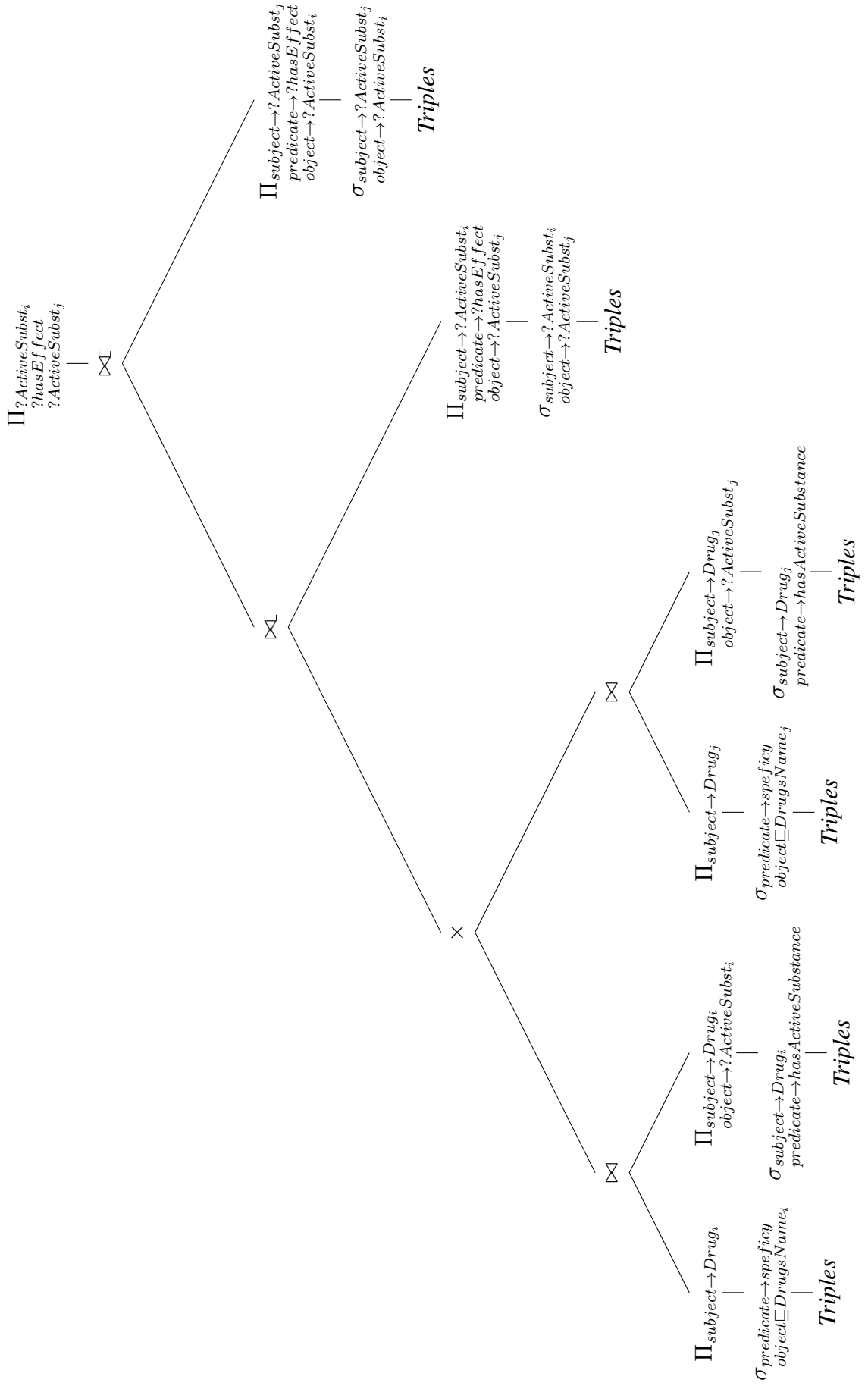


13 pav. Užklausos, kurioje surandami vaistai analogiškėmis veikliosiomis medžiagomis ir nustatomi jų šalutiniai poveikiai, vykdymo medis

Tai yra dinamiškai formuojamas užklausos modelis, kurio pagalba surenkamos įvesto vaisto veikliosios medžiagos ( $AS_i | i \in 1 \dots n$ ). Metodais  $\exists$  (apibrėžiamas  $\bowtie$ ) bei  $\nexists$  (apibrėžiamas  $\hat{\bowtie}$ ) surandami vaistai, turintys identišką veikliąsias medžiagas įvestam vaistui. Užklausos metodo  $\exists$  dalis formuojama  $n$  kartų, kiek susijęs vaistas turi veikliųjų medžiagų. Kai surandami vaistai, turintys identišką veikliąsias medžiagas, surenkami šių vaistų visi skirtingi šalutiniai poveikiai.

- 5. Nustatyti vaistų poros tarpusavio koreliaciją.** Vienas iš prototipo funkcijų yra aptikti terapinę vaistų sąveiką. Pavyzdžiui: vaistas "Gripex" yra skirtas peršalinimo ar gripo simptomams lengvinti. Šio vaisto veikliosios medžiagos yra paracetamolis, pseudoefedrinas ir dekstrometorfano hidrobromidas. "Gripex" vaisto nerekomenduojama vartoti su kitais vaistais, kurių sudėtyje yra paracetamolio. Su "Gripex" vaistiniu preparatu nerekomenduojama

gerti pavyzdžiui vaisto "Theraflu", nes jo veikliosios medžiagos yra paracetamolis, fenilef-rino hidrochloridas, gvajfenezinas. Vartojant šiuos vaistus kartu, iškyla didelė rizika sunkiai ir negrįžtamai pažeisti kepenis dėl per didelio paracetamolio kiekio. Todėl įgyvendinamas funkcijos modelis turėtų informuoti pacientą apie abiejų vaistų veikliųjų medžiagų sustiprėjimą ir vartojimo riziką. Taip pat atsiranda tokių variacijų, kai vaistiniai preparatai tarpusavyje negali būti vartojami, nes slopina ar stiprina tam tikras savybes, dėl kurių žmogaus organizmas labiau alinamas nei gydomas. Tokiai informacijai apskaičiuoti pateikiamas užklauso vykdomo medis, kuris atvaizduojamas 14 pav.



14 pav. Užklauso, kuri nustato dviejų vaistų tarpusavio sąveikas, vykdymo medis

6. **Pagal įvestą vaisto veikliąją medžiagą, bei medžiagą, kurios negali būti vaistų sudėtyje, nustatyti vaisto pavadinimą.** Pastaruoju metu dažnai girdima apie žmonių netoleravimą ar alergijas tam tikroms medžiagoms. Egzistuojančios sistemos dažnai neleidžia vykdyti paieškos, kai vaistų sudėtyje turi būti nustatytos veikliosios medžiagos, bei negali būti kitų tam tikrų veikliųjų ar sudėtinių medžiagų. Todėl naudinga turėti sistemoje tokį metodą, gebantį atrinkti reikalingus duomenis pagal įvestus parametrus. Užklausos vykdymo medis, skirtas surasti tokią informaciją, vaizduojamas 15 pav.

Taip pat, kaip ir ankstesniame punkte, tai yra dinamiškai formuojamas užklausos modelis, skirtas rasti vaistinius preparatus pagal pasirinktus parametrus. Į sistemos prototipą įvedus parametą  $SubstanceName_i$ , kai  $i \in 1..n$ , kuris atitinka vaisto veikliąsias ar pageidaujamas papildomas vaisto medžiagas, randami tokie vaistai pagal pasirinktus kriterijus.  $\exists$  sąlyga ( $\bowtie$ ) užklausoje kartojama  $n$  kartų. Parametrai, žymintys vaistų sudėtinę dalis  $SubstanceName_j$ , kai  $j \in 1..m$ , kurių neturi būti sudėtyje, yra neprivalomi. Tai reiškia, kad šis metodas gali būti naudojamas rasti vaistus pagal parametrais įvestas veikliąsias medžiagas. Tačiau jei įvedama daugiau nei viena nepageidaujama sudėtinė vaisto medžiaga,  $\nexists$  sąlyga ( $\bowtie$ ) kartojama  $m$  kartų. Jei atsiranda poreikis surasti vaistus, kurie sudėtyje turi tam tikras sudedamąsias medžiagas ir tam tikrų sudedamųjų medžiagų neturi, kurių vartotojas netoleruoja, tokiu atveju vartotojas turi užpildyti tiek  $SubstanceName_i$  tiek  $SubstanceName_j$ .





### 3. Sistemos įgyvendinimas

Sistemos prototipo įgyvendinimas susideda iš kelių etapų:

- Ontologijos projektavimas;
- Ontologijos užpildymas testiniais duomenimis;
- Prototipo įgyvendinimas, naudojant SPARQL užklausų interpretatorių.

Kiekvienas išvardintas punktas turi plačiau aprašantį poskyrį.

#### 3.1. Ontologijos projektavimas

Kuriant ontologiją, kuri apibrėžia konkrečią sritį, reikalingas veiksmų planas ontologijos įgyvendinimui. Argumentų, kokiais principais vadovaujantis reikia sukurti ontologiją, galima rasti daug. Iš esmės, ontologijos kūrimas yra iteratyvus procesas[21]. Kuriam ir taisoma ontologija tol, kol pasiekiamas reikiamas rezultatas. Taip pat [22, 21] teigiama, kad vienintelio teisingo metodo kuriant ontologijas nėra. Svarbu, kad ontologija išliktų tiksli, nuosekli, pilna ir glausta. Šio magistro baigiamojo darbo ontologijos modelis taip pat buvo modifikuojamas daug kartų, kol liko darbo specifika tenkinantis rezultatas. [21] straipsnyje aprašomas veiksmų planas, koks turi būti kuriant ontologiją. Taigi, išskiriami šios pagrindinės idėjos:

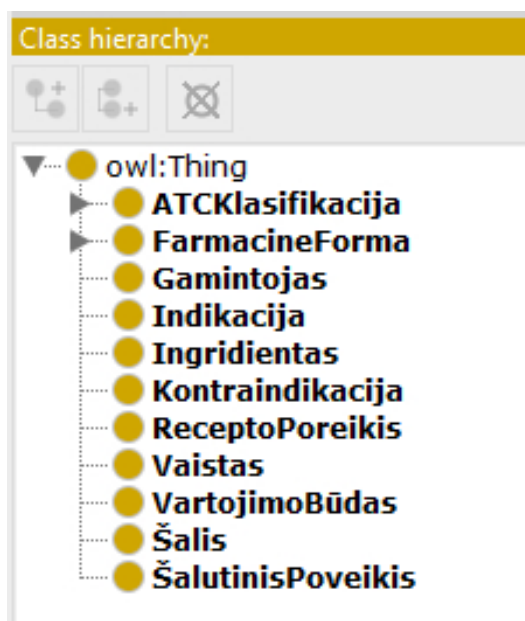
- Apibrėžti ontologijos konceptą, sukurti klases. Šio magistro baigiamojo darbo konceptas yra vaistų ontologija. Todėl išskiriamos tokios pagrindinės klasės kuriamoje vaistų ontologijoje: vaistas, ingredientas, vaistų ATC klasifikacija, farmacinė forma, gamintojas, šalis, kurioje vaistas gaminamas, indikacija, kontraindikacija, recepto poreikis, vartojimo būdas, nepageidaujamas poveikis.
- Įvilkti konceptą į taksonominę hierarchiją. Hierarchiškai sudėliojamos šios klasės: ATC klasifikacija ir farmacinė forma. Tarptautinėje vaistų ATC klasifikacijoje suklasifikuojami vaistai pagal veikiamą organą, sritį ar organų sistemą. Plačiau ATC klasifikacija aptarpinama 3.2 skyriuje. Vaisto farmacinė forma numato ar vaistas yra tabletė, tirpalas, suspensija, tepalas ar kt.
- Apibrėžti ryšius tarp klasių, nustatyti ontologijos pagrindines sritis ir rangus. Vaistų ontologijoje aprašomi šie ryšiai tarp klasių: gamina, randasi, turiIndikaciją, turiKodą, turiKontraindikaciją, turiPagalbinęMedžiagą, turiPoveikį (šis ryšys dar taksonomiškai suskaidomas į didinaKoncentraciją, nerekomenduojamaVartotiSu, slopinaPoveikį, stiprinaPoveikį, vartotiDraudžiama), turiŠalutinįPoveikį, turiVeikliąjąMedžiagą, vartojamasBūdu, yra, yraFormos, yraGaminamas. Šie ryšiai jungia tarpusavyje klases ir klasių objektus.
- Apibrėžti atributus (duomenų sritį) ir kokias reikšmes jie gali laikyti. Vaistų ontologijos modelyje nebuvo numatyta įgyvendinti duomenų sričių.
- Užpildyti ontologiją individualiais, aprašyti objektų savybių šaltinius, duomenų savybių šaltinius.
- Apibrėžti aksiomas ir funkcijas, jei surinkta pakankamai žinių apie aprašomą sritį.

Pritaikius ontologijų kūrimui naudojamą metodiką, šiame darbe ontologijos modelis neturi daug klasių ir objektų. Taip yra todėl, kad norint ją užpildyti pilnai, reikia turėti daug farmacinių žinių, kad ontologiją būtų galima išplėsti iki reikiamo dydžio. Tačiau ontologijomis grįstam vaistų rekomendavimo metodikos kūrimui toks ontologijos modelis yra pakankamas.

### 3.2. Vaistų ontologijos įgyvendinimas

Vaistų ontologijos įgyvendinimui buvo pasirinktas jau egzistuojantis ontologijų kūrimo įrankis *Protégé 5.1.0*. Protégé įrankis sukurtas Stanfordo medicinos mokykloje, Stanfordo biomedicinos informatikos tyrimų centre. Šis įrankis yra nemokama atviro kodo programa, skirta egzistuojančių ontologijų modifikavimui ir naujų sistemų kūrimo karkasas (angl. framework). Ji įgyvendinta Java programavimo kalba. Kadangi Protégé yra atviro kodo programa, jos galimybes galima išplėsti pagal poreikį. Ji taip pat turi galimybę suteikti plug-and-play aplinką, kuri leidžia lanksčiai kurti prototipus ir aplikacijas. Taip pat Protégé pilnai palaiko vėliausią OWL 2 Web ontologijų kalbą ir RDF specifikacijas [29].

Kuriant ontologijas, turi būti nustatoma duomenų taksonomija. Kuriamai vaistų ontologijai buvo įgyvendintos tokios klasės ir jų hierarchija, vaizduojama 16 pav. Kad išivaizduoti dėl kokių



16 pav. Informacinės sistemos veikimo principas

priežasčių buvo pasirinkta įgyvendinti būtent tokias klases ir savybes, žemiau pateikiams detalesnis klasių aprašymas:

- *ATC klasifikacija*. ATC (Anatominę – Terapinę – Cheminę) vaistų klasifikaciją Pasaulinė sveikatos organizacija (PSO) rekomenduoja kaip tarptautinę vaistų klasifikaciją [10]. ATC klasifikacija susideda iš 14 pagrindinių klasių, kur kiekviena klasė reiškia tam tikrą poveikį organui ar organizmo sistemai. Išskiriamos tokios pagrindinės ATC klasės:
  - Virškinimo trakto ir metabolizmo vaistiniai preparatai;
  - Kraujo ir kraujo formavimo organų vaistai;
  - Širdies ir kraujagyslių sistemos vaistai;

- Dermatologiniai vaistai;
- Urogenitalinės sistemos vaistai ir lytiniai hormonai;
- Sistemiskai veikiantys hormonų preparatai, išskyrus lytinius hormonus ir insulinus;
- Antiinfekciniai vaistai, skirti sisteminiam naudojimui;
- Antinavikiniai vaistai ir imunomodulatoriai;
- Raumenų ir skeleto sistemos vaistai;
- Nervų sistemos vaistai;
- Antiparazitiniai vaistai, insekticidai ir repelentai;
- Kvėpavimo sistemos vaistai;
- Jutimo organų vaistai;
- Įvairūs kiti vaistiniai preparatai.

Aukščiau išvardintos pagrindinės ATC klasės yra pirmo lygio. Vėliau jos skaidomos į mažesnes klases, kol pasiekiamas 5-tasis kodo formavimo lygis ir kodas priskiriamas vaistiniam preparatui. Pavyzdžiui: kodas A10BA02 suteikiamas tokiam vaistui, kurio cheminė medžiaga yra metforminas.

Analizuojant Lietuvoje parduodamų vaistų sąrašą, kuris prieinamas viešai [www.vvkt.lt](http://www.vvkt.lt), pastebėta, jog homeopatiniams vaistams ATC klasifikacijos kodas nėra suteikiamas. Todėl nagrinėjant homeopatinius vaistus, jų saveika organui ar organizmo sistemai nėra nustatoma.

- *Farmacinė forma.* Vaisto farmacinė forma, kurioje saugoma vaisto tekstūra. Klasė taip pat turi taksonomiją. Pavyzdžiui, vaisto farmacinė forma vizualiai yra tabletė, tačiau ji gali būti burnoje tirpstanti tabletė, burnoje disperguojanti tabletė, skrandyje neiri tabletė ar pan.
- *Gamintojas.* Klasei priskiriami vaistų gamintojų subjektai (angl. Individuals).
- *Indikacija.* Klasėje registruojamos visos galimos vaistų indikacijos. Tam tikros indikacijos kartais gali sutapti su kontraindikacijomis ar šalutiniais poveikiais. Pavyzdžiui, vaisto "Tavegil" indikacija gali būti tokia, kad vartoti anafilaksinio šoko metu. Tačiau šis vaistas taip pat gali turėti tokį šalutinį poveikį ir sukelti anafilaksinį šoką.
- *Ingridientas.* Registruojamos visos galimos vaistinių preparatų medžiagos.
- *Kontraindikacija.* Registruojamos visos galimos kontraindikacijos. Kaip minėta aukščiau, kartais konkreti kontraindikacija gali sutapti ir su indikacija ar šalutiniais poveikiais.
- *Vartojimo būdas.* Registruojami visi galimi vaistų vartojimo būdai. Pavyzdžiui, leisti į veną, leisti į raumenį, vartoti per burną, vartoti ant odos ir panašiai.
- *Šalis.* Registruojamos šalys, kuriose gamintojai yra. Tai yra tarsi kitos srities ontologija, per kurią būtų galima sujungti su vietos ontologija.
- *Šalutinis poveikis.* Registruojami visi galimi šalutiniai poveikiai. Šalutiniai poveikiai taip pat gali sutapti su indikacijomis ir kontraindikacijomis.

Įgyvendinant klases galima nurodyti klasių ekvivalentumą su kitomis klasėmis. Taip apibrėžiami sinonimai tarp klasių. Taip pat įgyvendinant ontologijas galima naudoti klasių "atsiskyrimą" (angl. disjoint). Šis metodas padeda išlaikyti ontologiją vientisą. Pavyzdžiui, kad klasė tabletė nėra tirpalas, nors jie abu yra poklasiai klasės "Farmacinė forma". Tai padeda sunorminti duomenis pagal numatytą struktūrą. Taip pat galima nurodyti sudėtingas klases, tačiau tokiais duomenimis tampa sudėtingiau naudotis ir gali tapti nepatogu vykdyti užklausas ontologijose.

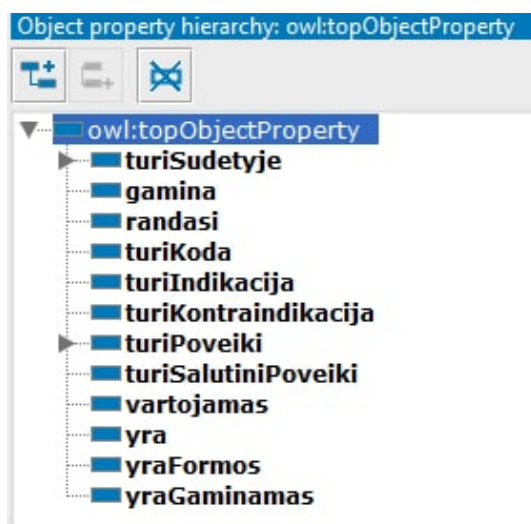
Ontologijoje taip pat registruojama ir objektų savybės (angl. object property). Objektų savybė dar kitaip vadinama predikatas. Jis paprastai būna veiksmažodis (bet nebūtinai), pavyzdžiui: yra, susideda iš, yra dalis, slopina, gamina ir pan. Šaltinyje [19] aprašomos tokios galimos ryšių savybės:

- **Funkcinė savybė.** Jei ryšys (predikatas) tarp subjekto ir objekto yra funkcinis, tai reiškia, kad gali egzistuoti tik vienas toks objektas. Kadangi vaistų ontologijoje nebuvo įgyvendinta funkcinė savybė, tai pavyzdys pateikiamas nesusijusia su vaistais informacija. "Vilnius yra Lietuvos sostinė" ir kitame šaltinyje "Vilnius yra LT sostinė", tai reiškia, kad Lietuva ir LT yra tas pats objektas.
- **Atvirkščiai funkcinė savybė.** Jei predikatas atvirkščiai funkcinis, tai reiškia, kad vėl gi gali egzistuoti tik vienas toks ryšys. Imant pavyzdį pateiktą pirmame taškelyje, tai "Lietuvos sostinė yra Vilnius" ir "LT sostinė yra Vilnius", tai daroma išvada, kad Lietuva ir LT yra tas pats objektas.
- **Pereinanti, tranzityvi savybė.** Jei predikatas yra tranzityvus ir jis jungia subjektą a su objektu b ir taip pat b subjektas jungiasi šiuo predikatu su c, tokiu atveju daroma išvada, jog a yra taip pat susijusi su individualu c. Pavyzdžiui: Jonas turi brolių Tomą, o Tomas turi brolių Artūrą, tokiu atveju Jonas turi brolių Artūrą.
- **Simetrinė savybė.** Jei predikatas yra simetrinis, tai subjektas yra susijęs su objektu taip, kaip objektas susijęs su subjektu. Pavyzdžiui: Šalis turi miestą taip pat kaip miestas turi šalį.
- **Asimetrinė savybė.** Jei ryšys yra asimetrinis, tai subjektas susijęs su objektu tokiu ryšiu reiškia, kad objektas negali būti susijęs pačiu ryšiu su subjektu. Pavyzdžiui: Akvariume gyvena žuvis, tačiau žuvis gyvena ne tik akvariume.
- **Į save orientuota (angl. Reflexive) savybė.** Tai reiškia, kad predikatas turintis tokią savybę, nukreipia ryšį į patį subjektą. Pavyzdžiui: Aurimas pažįsta save.
- **Į save neorientuota (angl. irreflexive) savybė.** Tai reiškia, kad gali būti sujungtas subjektas su objektu per šią savybę, bet subjektas ir objektas nėra tas pats. Pavyzdžiui: Laurynas yra Sigitos vadovas, o Laurynas negali būti sau vadovas ir Sigita negali būti sau pavaldinė. O ryšys tarp Lauryno ir Sigitos yra galimas.

taip pat kaip ir kuriant ontologijų klases, taip ir projektuojant predikatus, turi būti numatoma duomenų taksonomija. Todėl šio darbo įgyvendinti predikatai matomi 17 pav.

Detalesnė informacija pateikiama:

- **Gamina.** Ryšys, jungiantis klasių "Gamintojas" ir "Vaistas" narius. Šis predikatas yra atvirkščiai vartojamas predikatui "yra gaminamas". Tai reiškia, kad ryšiai tarpusavyje koreliuoja ir yra atvirkšti vienas kitam, kaip antonimai. Šiuo atveju, "Gamintojas" yra pagrindinė sritis, o "Vaistas" yra rangas (angl. Ranges).



17 pav. Informacinės sistemos veikimo principas

- *Turi indikaciją.* Ryšys, jungiantis sritis "Indikacija" ir "Vaistas". "Vaistas" yra sritis, o "Indikacija" yra rangas.
- *Turi kontraindikaciją.* Ryšys, jungiantis sritis "Kontraindikacija" ir "Vaistas".
- *Randasi.* Predikatas, jungiantis "Gamintojas" ir "Šalis". "Gamintojas" yra pagrindinė sritis, o "Šalis" yra rangas.
- *Turi kodą.* Ryšys, apibūdinantis "ATC klasifikacija" ir "Vaistas" klasių narius. "Vaistas" yra sritis, o "ATC klasifikacija" – rangas.
- *Turi pagalbinę medžiagą.* Ryšys, apibūdinantis vaisto sudedamąsias dalis. Jungia klasių "Vaistas" ir "Ingridientas" narius.
- *Turi poveikį.* Predikatas, apibūdinantis "Ingridiento" su "Ingridientu" ryšį. Tai yra į save orientuota savybė. Predikatai taip pat kaip klasių taksonomijos gali būti skaidomi į labiau aiškinančius predikatus. Predikatas "Turi poveikį" išskaidomas:
  - Didina koncentraciją;
  - Nerekomenduojama;
  - Slopina poveikį;
  - Stiprina poveikį;
  - Vartoti draudžiama. Predikatas yra simetrinis. Nes jei ingredientą A vartoti draudžiama su ingredientu B, tai ir B ingredientą draudžiama vartoti su ingredientu A.
- *Turi šalutinį poveikį.* Predikatas, jungiantis sritis "Vaistas" ir "Šalutinis poveikis". Klasė "Vaistas" yra pagrindinė sritis, o "Šalutinis poveikis" yra rangas.
- *Turi veikliąją medžiagą.* Predikatas, apibūdinantis "Vaistas" ir "Ingridientas" ryšį. "Vaistas" yra pagrindinė sritis, o "Ingridientas" yra rangas.
- *Vartojamas.* Ryšys, nusakantis kaip koreliuoja klasių "Vaistas" ir "Vartojimo būdas" nariai. "Vaistas" yra pagrindinė sritis, o "Vartojimo būdas" – rangas.

- *Turi recepto poreikį.* Predikatas tarp "Vaistas" ir "Recepto poreikis". "Vaistas" yra sritis, o rangas – "Recepto poreikis".
- *Farmacinė forma.* Predikatas tarp "Vaistas" ir "Farmacinė forma". "Vaistas" yra sritis, o "Farmacinė forma" – rangas.
- *Gaminamas.* Predikatas nusako ryšį tarp "Vaistas" ir "Gamintojas". Priešingai nei aukščiau aprašytame predikate "Gamina", čia sritis yra "Vaistas", o rangas yra "Gamintojas".

Igyvendinant vaistų ontologiją nebuvo pasirinkta įgyvendinti duomenų savybių pagal tipą (data property). Duomenų tipais dažnai galima aprašyti nežinomas, nuolat kintančias, savybes. Tokias kaip vaisto svoris ar veikliosios medžiagos kiekis vaisto sudėtyje.

Aprašius ontologijų duomenis ir supildžius ją duomenimis, Protégé įrankis leidžia sumodeliuotą ontologiją saugoti RDF/XML, Turtle ir OWL/XML duomenų formatais.

### 3.3. Vaistų ontologijos užpildymas duomenimis

Duomenų įkėlimas buvo vykdomas pagal VVKT pateiktus duomenis "Pilnas vaistinių preparatų sąrašas pakuočių lygyje", kuris VVKT puslapyje patalpintas CSV formatu. Šiame faile buvo maža dalis vaistinių preparatų informacijos, kuri egzistuoja vaistų informaciniuose lapeliuose. Iš vaistinių preparatų sąrašo buvo paimami šie duomenys: vaisto pavadinimas, veiklioji medžiaga, farmacinė forma, recepto poreikis, ATC klasifikacijos kodas, kuris priskirtas vaistui, gamintojas bei gamintojo šalis. Prieš įkeliant duomenis į vaistų ontologiją, failas buvo modifikuotas. Išskaidytos vaistų veikliosios medžiagos, nes kartais vaistai veikliųjų medžiagų turi ne vieną. Likę reikalingi duomenys, tokie kaip vaisto pagalbinės sudedamosios dalys, indikacijos, kontraindikacijos ir šalutiniai poveikiai buvo surinkti rankiniu būdu iš vaistų informacinių lapelių, kurie taip pat įkelti VVKT puslapyje kiekvienam vaistui atskirai. Iš informacinių lapelių įkelti į Excel ir iš Excel su Cellfie įskiepiu (Protégé įrankyje) sukelti į ontologiją. Cellfie gali apdoroti tik excel tipo dokumentus.

Dėl aukščiau išvardintų priežasčių, į vaistų ontologiją buvo įkelta tik apie 300 vaistinių preparatų. Rankomis surinkta informacija apie šalutinius poveikius, indikacijas ir kontraindikacijas buvo 80-čiai vaistinių preparatų. Duomenims sukelti buvo sukurta 33 aksiomos. Šių aksiomų pagalba duomenys buvo sukelti į vaistų ontologiją.

Buvo planuojama įgyvendinti automatinį duomenų nuskaitymą iš vaistų informacinių lapelių. Tačiau susidurta su sunkumais, dėl kurių nepavyko to įgyvendinti. Nors egzistuoja metodų [27], kaip automatinio būdu iš sakinio surasti tripletą, tačiau tai nėra pritaikoma lietuvių kalbai. Straipsnyje [27] minimi metodai/įrankiai skirti apdoroti anglų kalbos sakinius ir aptikti juose tripletus. Didžioji dauguma natūralios kalbos apdorojimo metodai ir algoritmai yra pritaikyti apdoroti anglų kalbos sakinius, nes tai yra viena populiariausių kalbų pasaulyje. Remiantis [30] šaltiniu, lietuvių kalbos palaikymas kalbų technologijose tiek šnekamosios kalbos apdorojime, tiek automatiniam vertime, tiek teksto analizėje, tiek kalbos ir tekstų ištekliuose, yra vertinama menkai arba išvis nėra jokio palaikymo. Tai reiškia, kad lietuvių kalba šiose srityse, priešingai nei anglų kalba, turi mažai įgyvendintų sistemų, kurios gebėtų atlikti lietuviško teksto analize. Lietuvių kalbos apdorojimui vis dar kuriamos programos, kurios geba automatiškai analizuoti tekstus.

Kaip minima [30] straipsnyje, apdoroti lietuvių kalbos sakinius ir surasti jų kontekstinę informaciją bei ištraukti iš jų tripletus yra sudėtinga, nes:

- lietuviu kalbos sakinio sandara nėra fiksuota. Todėl iškyta sunkumų, nes iš karto nėra žinoma sakinio sandaros eilės tvarka, padedanti identifikuoti tripletui reikalingas reikšmes. Sakinio vientisuma dažniausiai užtikrina žodžių galūnės.
- egzistuoja eliptiniai sakiniai, kai praleisti žodžiai numanomi tik iš konteksto.
- lietuvių kalbos didelę dalį sudaro homonimai. Tai yra tokie žodžiai, kurie rašomi taip pat, bet žodžių prasmė yra skirtinga. Pavyzdžiui: žodis "galvos" daiktavardžio daugiskaitos vardininkas ir veiksmažodžio busimojo laiko trečiasis asmuo. Kaip minima šaltinyje [26], homonimų lietuvių kalbos tekstynuose yra apie 39 proc.

Nagrinėjant vaistų informacinius lapelius, buvo susidurta su aukščiau išvardintomis lietuviškų tekstų analizės automatinio nustatymo problemomis. Taip pat buvo susidurta su farmacijos srities neišmanymu, todėl ontologija buvo kuriama remiantis [22] pateiktu metodu. Farmacinės sritys, kurios apsunkino ontologijos įgyvendinimą ir duomenų suvedimą:

- Informaciniuose lapeliuose duomenys pateikiami apibendrintai. Pavyzdžiui: "vartoti vaisto negalima su antikoguliantais". Nežinant kas yra antikoguliantai, kokios veikliosios medžiagos priklauso antikoguliantams, sunku supildyti duomenis ontologijoje. Jei informaciniuose lapeliuose suvesta, jog, pavyzdžiui, negalima vaisto vartoti su konkrečia veikliąja medžiaga, toks atvejis magistro baigiamojo darbo vaistų ontologijoje užregistruotas. Ši problema gali būti sprendžiama dvejopai: a) išskaidant Ingridiento klasę į taksoniją, kur atsiranda ingredientų sritys arba b) gali būti surandamos visos antikoguliantų sričiai priklausančios veikliosios medžiagos ir užregistruojama einamojo vaisto veikliosios medžiagos sąveika su antikoguliantams priklausančiomis veikliosiomis medžiagomis.
- Duomenys nėra surašyti sistemiskai: pavyzdžiui, nagrinėjant vaisto indikacijas, vieno vaistų informaciniame lapelyje rašo "galvos skausmo malšinimas, dantų skausmo malšinimas", kitame rašo "galvos, dantų skausmo malšinimas". Todėl norint apibendrinti, kad vaistai slopina tokius pačius simptomus, reikia žmogaus įsikišimo. Dėl šios priežasties buvo labai apsunkintas duomenų įkėlimas į ontologiją, nes reikalavo daug žmogiškojo faktoriaus.
- Dažnai prie kontraindikacijų parašyta, jog vaisto X negalima vartoti su ingredientu Y. Taip daroma prielaida, jog tam tikra veiklioji vaisto medžiaga negali būti vartojama su vaistiniu ingredientu Y. Kartais vaistinis preparatas turi daugiau nei vieną veikliąją medžiagą. Todėl vykdoma dekartio sandauga, nes nėra aišku, kuri konkreči medžiaga nėra suderinama su vaistiniu ingredientu Y.
- Indikacijos dažnai aprašomos apibendrintai. T.y. ligos ne išvardinamos, o aprašomas bendras jų simptomas. Pavyzdžiui, vietoj ligų sąrašo (sisteminės raudonosios vilkligės, reumatinio artrito), rašomas "sisteminės uždegiminės ligos". Todėl tai vėl apsunkina indikacijų / kontraindikacijų / šalutinių poveikių suvedimą į ontologiją.
- Neaišku ar vieningai aprašomi vaistinių preparatų veikliųjų medžiagų pavadinimai. Pavyzdžiui, vaistai "Lioton" gelis ir "Heparoid" kremas malšina tuos pačius simptomus. "Lioton" vaisto veiklioji medžiaga heparinoidas, Vaisto "Lioton" veiklioji medžiaga yra heparino natrio druska. Nėra aišku ar heparinoidas yra tas pats kaip heparino natrio druska. Jeigu tai yra ta pati veiklioji medžiaga, šias veikliąsias medžiagas aprašyti kaip sinonimus. Todėl būtų tikslinga, vaistų ontologiją pildyti kartu su gydytoju ar vaistininku.



Duomenims sukelti buvo naudojamas Cellfie įrankis, kuris leidžia Excel formatu surinktus duomenis pagal sukurtas aksiomas sukelti į ontologiją. Aksiomas aprašomos, naudojant Mapping Master srities specifinę kalbą (DSL (domain specific language)). Cellfie įrankio veikimo principas: pagal aprašytas aksiomas (pvz: 18 pav. ) imami duomenis iš nurodytų stulpelių, sugeneruojamas

```
Individual: @E* (mm:SkipIfEmptyLocation)
Types: Ingridientas
Facts: vartotiDraudžiama @G*
```

18 pav. Aksioma, sugeneruojanti vaistų sudedamųjų dalių sąveiką (draudžiama vartoti).

jų aprašas, kurį galima peržiūrėti ir jei jis netinkamas, galima pakeisti. Tik tada, kai įsitikinama, jog duomenys bus suimportuoti korektiškai, jie importuojami į ontologiją.

Duomenys Excel faile turi būti sutvarkyti. Pavyzdžiui, jei keli duomenys apie vaistą buvo surašyti vienoje celėje, atskirti kableliu, tai šie duomenys turi būti išskaidomi į eilutes. Tas pats ir jei norima suvesti duomenis į hierarchinę struktūrą. Tokiu atveju, kiekviena žemesnė klasė turi būti naujame stulpelyje. Nors duomenys sukeliama pakankamai greitai, tačiau duomenų paruošimas sukėlimui į ontologiją labai ilgai užtrunka ir reikalauja daug rankinio darbo. Todėl ateities darbų gairė galėtų būti natūralios kalbos apdorojimas, ištraukiant duomenis iš teksto.

### 3.4. Modelio funkcijų ir sistemos įgyvendinimas

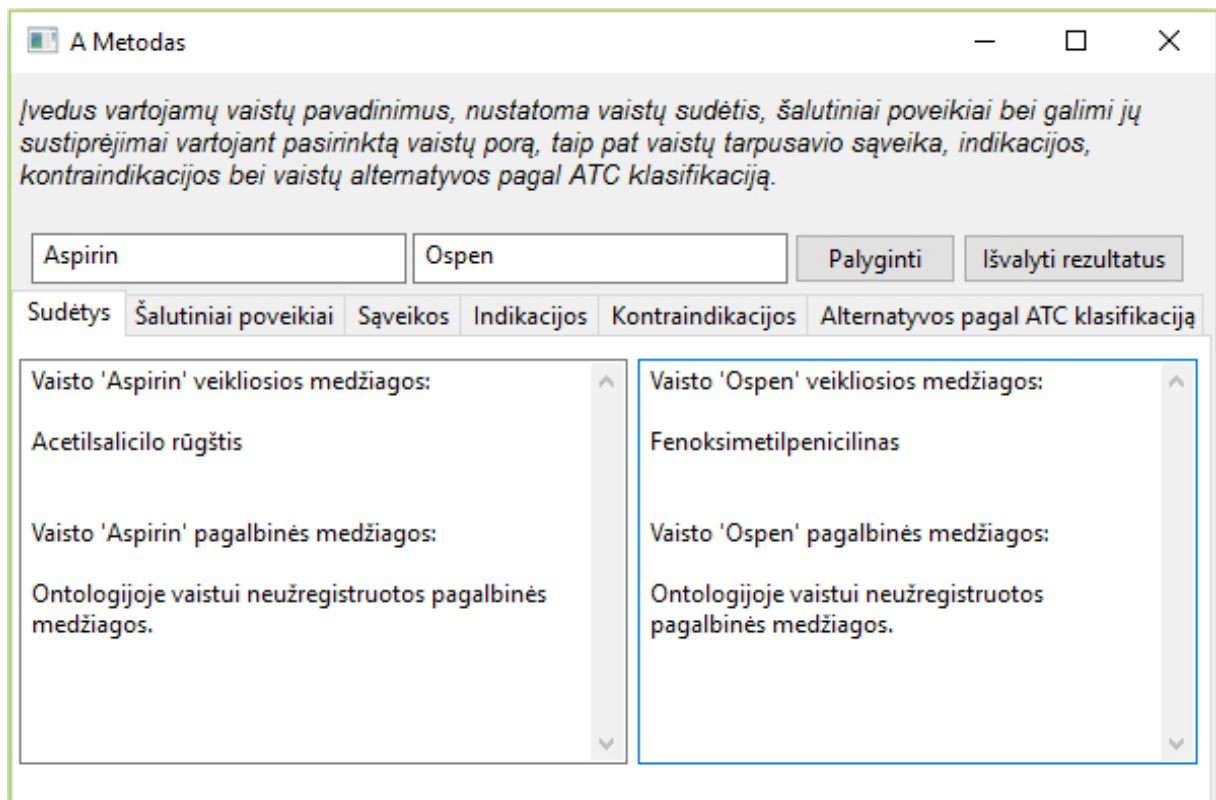
Vaistų rekomendavimo sistemos modeliui įgyvendinti buvo pasirinkta Python 2.7 programavimo kalba. Grafinei vartotojo sąsajai įgyvendinti naudojama WXPYTHON biblioteka. Užklausų vykdymui semantiniame tinkle buvo naudojamas nutolęs šaltinis SPARQLer[1], vėliau pakeistas į virtuoso SPARQL procesorių[3], nes SPARQLer, prototipo įgyvendinimo eigoje, panaikino galimybę vykdyti užklausas iš konkrečios nutolusios ontologijos. Taip pat prototipo įgyvendinimui naudojama Python 2.7 SPARQL Wrapper biblioteka, skirta SPARQL užklausų išsiuntimui į SPARQL procesorių bei gautų rezultatų apdorojimui.

Nors pati SPARQL užklausų kalba nėra priklausoma nuo naudojamos platformos, tačiau SPARQL redaktorius realizuojantys įrankiai gali būti skirtingi. Šiame darbe užklausos buvo vykdomos SPARQLer ir virtuoso SPARQL redaktoriais. SPARQLer buvo galingsnis įrankis, nes galėjo be apribojimų vykdyti užklausas taksonominėje hierarchijoje tiek ieškant tėvinių įrašų, tiek vaikinių. Virtuoso SPARQL redaktorius tokios galimybės neduoda. Jis leidžia vykdyti užklausas hierarchinėje struktūroje ieškant vaikinių įrašų. Taip pat virtuoso SPARQL redaktoriaus grupavimo algoritmas labai apsunkinamas, ir tos užklausos, kurias SPARQLer sėkmingai įvykdydavo, virtuoso SPARQL redaktoriuje užstrigdavo.

Sistemos prototipe įgyvendinti du pagrindiniai metodai, kurių rezultatai skaidomi į smulkesnius metodus ir teikiami jų rezultatai. T.y. vienas iš pagrindinių metodų yra nustatyti dviejų vaistų informaciją. Geresniam išvaizdavimui apie ką kalbama, pateikiamas prototipo langas, matomas 19 pav.

Taigi, viršuje vartotojas įveda dominančių vaistų pavadinimus ir spaudžia mygtuką [Palyginti]. Rezultatas gaunamas toks, kad kiekviename lange pateikiama ontologijoje rasta informacija. Visa informacija surenkama 2.3 poskyryje aprašytomis metodų funkcijomis.

Kortelėje "Sudėtys" pateikiami rezultatai apie konkrečių vaistų veikliašias medžiagas ir pagalbines medžiagas.



19 pav. A metodas. Lygina vaistų poros informaciją ir pateikia apibendrintus rezultatus.

Kortelėje "Šalutiniai poveikiai" teikiama išskaidyta informacija į tris dalis. T.y.:

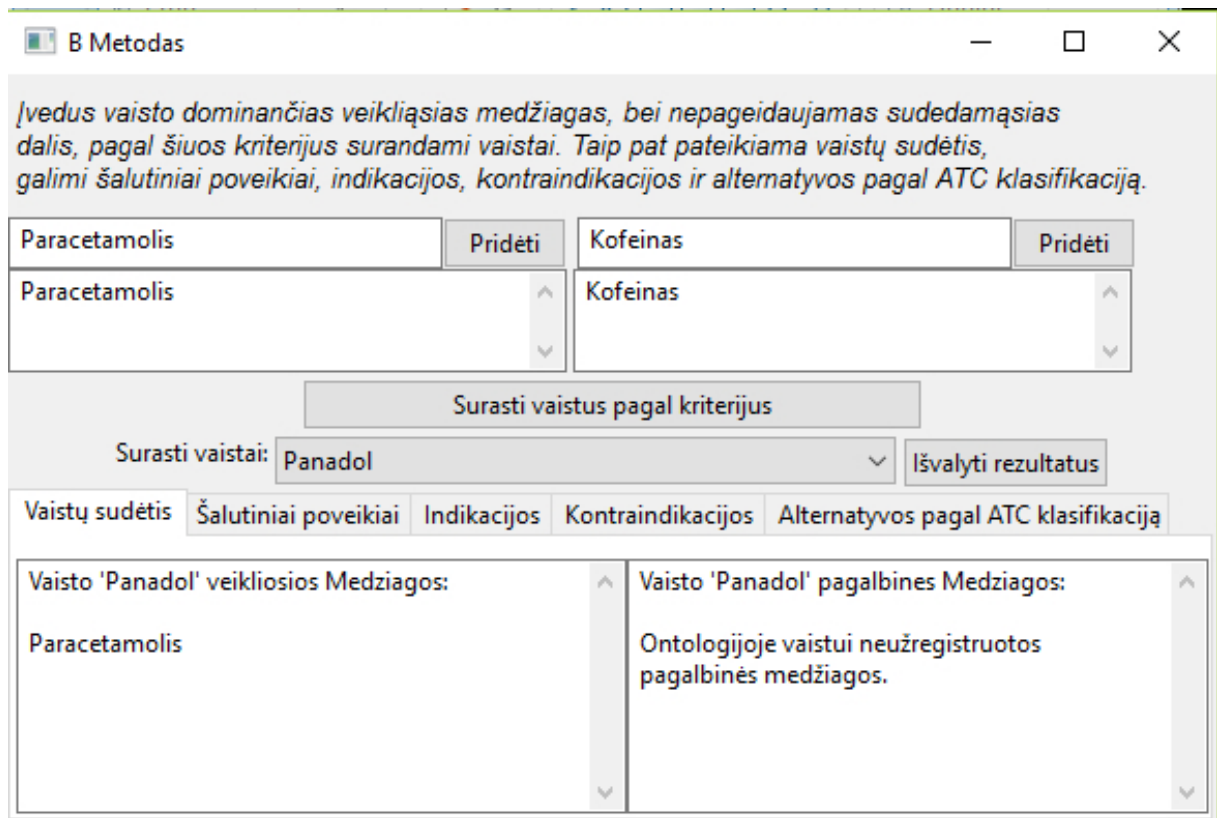
- konkrečiam vaistui nustatyti šalutiniai poveikiai;
- nustatyti visi galimi šalutiniai poveikiai visų vaistų, turinčių analogiškas veikliąsias medžiagas. Toks metodas taikomas, nes kartais vaistai patenka į vartojamąją rinką pilnai neištyrus galimų šalutinių poveikių. Todėl šalutinių poveikių apjungimas tų vaistų, kurie pateko į rinką pilnai neištirti, gali išspręsti šalutinių poveikių surinkimą;
- jeigu vaistų poroje buvo užregistruoti šalutinių poveikių duomenys, kurie tarpusavyje persidengia, tai galima daryti prielaidą, jog didesnė tikimybė sulaukti būtent tokio šalutinio poveikio, kuris užregistruotas prie abiejų vaistų.

Kortelėje "Sąveikos" pateikiama informacija apie vaistų veikliųjų medžiagų tarpusavio sąveikas. t.y. ar vaistų poros vartojimas vienu metu nėra labiau kenksmingas sveikatai, nei simptomų malšinimas.

Kortelėse "Indikacijos" ir "Kontraindikacijos" duomenys pateikiami kiekvienam vaistiniam preparatui atskirai. T.y. kokios indikacijos ir kontraindikacijos priskiriamos konkrečiam vaistui.

Kortelėje "Alternatyvos pagal ATC klasifikaciją" pateikiama informacija apie alternatyvas vaistų, turinčių priskirtą tą patį ATC klasifikacijos kodą. Susiejus duomenis su kitomis vaistų ontologijomis, gal net kitomis užsienio kalbų ontologijomis, ATC kodas galėtų būti ontologijas rišančioji sritis.

Kitas metodas, matomas 20 pav. yra nustatyti vaistus, kai žinomos veikliosios/pagalbinės medžiagos, kurios turi būti vaistų sudėtyje, ir tokios sudėtinės dalys, kurių neturėtų būti vaistų



20 pav. B metodas. Metodas nustato vaistų pavadinimus, kai įvedamos žinomos veikliosios/pagalbinės medžiagos, kurios turėtų būti sudėtyje, ir kitos medžiagos, kurios vaistų sudėtyje neturi egzistuoti.

sudėtyje. Sudėtinės dalys, apibrėžiamos kaip medžiagos, kurių neturi būti vaisto sudėtyje yra neprivalomas atributas. Tai reiškia, kad vartotojas gali vykdyti užklausą surasti vaistus tik pagal veikliąsias/pagalbines medžiagas. Šis metodas naudingas netoleravimą ar alergiją turintiems žmonėms, kurie negali vartoti tam tikrų medžiagų. Nes tokiu atveju žmogaus savijauta gali prastėti vietoj to, kad žmogus jaustųsi komfortiškai. Metodas, taip pat kaip ir ankstesnis, skaidomas į smulkesnius metodus. Ne visi metodai, naudoti lyginant vaistų informaciją, tinkami šiam metodui. Smulkesnieji metodai gali būti įvardijami kaip pagalbiniai, skirti surinkti informaciją apie nustatytus vaistus.

Kortelėje "Vaistų sudėtis" vaizduojama informacija apie pasirinkto vaisto sudedamąsias vaisto dalis. T.y. viename lange teikiama informacija apie vaisto veikliąsias medžiagas, o kitame lange apie vaisto sudėties pagalbines medžiagas.

Kortelėje "Šalutiniai poveikiai" vaizduojama dviejų tipų informacija:

- konkrečiam vaistui užregistruoti šalutiniai poveikiai;
- surinkti visi galimi šalutiniai poveikiai einamojo vaisto ir kitų vaistų, turinčių analogiškas veikliąsias medžiagas. Kaip ir A metode, šis funkcionalumas leidžia įvertinti visus šalutinius poveikius, nes kartais vaistai patenka į rinką pilnai neištirti.

Kortelėse "Indikacijos" ir "Kontraindikacijos" teikiama informacija apie vaistui priskirtas indikacijas ir kontraindikacijas. Kortelėje "Alternatyvos pagal ATC klasifikaciją" teikiama informacija apie vaistus, turinčius priskirtą tą patį ATC kodą. Šioje kortelėje duomenys imami tiesiogiai pagal ATC kodą, nenufiltruojant jei buvo pasirinktos medžiagos, kurių neturėtų būti vaistų sudėtyje.

## 4. Metodų sudėtingumo vertinimas

Kadangi eksperimentinio vertinimo sistemos prototipui nepavyko atlikti dėl naudojamo virtualio SPARQL redaktorius subtilių, bus pabandyta įvertinti naudojamų metodų sudėtingumas.

Vertinant užklausų sudėtingumą, įprasta naudoti sąnaudomis grįstą vertinimą (angl. cost based estimation). Įprastai šis vertinimas naudojamas reliacinių duomenų bazių užklausų vykdymui. Tokio vertinimo pagrindas yra įrašų skaičiaus įtaka vykdomai užklausiai. Tačiau šiame darbe vertinamas užklausų sudėtingumas ne įrašų skaičius duomenų bazių lentelėse, o tripletų skaičius ontologijose. Kadangi tripletų skaičius iš dalies atitinka įrašų skaičių duomenų bazėse, todėl ir šiame darbe pateikiamas užklausų sudėtingumo įvertinimas.

Vertinimas bus atliekamas remiantis 2.3 poskyryje aprašytais užklausų vykdymo medžiais. Kiekvienai ontologijoje vykdomai užklausiai bus atliekamas blogiausios ( $\theta$ ) ir geriausios ( $\omega$ ) vykdymo kainos įvertinimas. Kaina nurodo teorinį reikalingą atminties kiekį, kad užklausa galėtų būtų įvykdyta. Pasirinktas teorinis įvertinimas todėl, kad paprastai ontologijos ir jų dokumentai būna talpinami į RDF struktūrą atitinkančius serverius, kurie turi užklausų optimizatorius. Dėl šios priežasties vykdymo kaina gali būti mažesnė, nei įvertinta teoriniame lygmenyje.

Tarkime, tripletų aibę  $\mathcal{T}$  sudaro  $n$  tripletų. Taip pat įsivaizduokime, kad kiekviena tripleto sudedamoji dalis t.y. subjektas ( $s$ ), predikatas ( $p$ ) ir objektas ( $o$ ), sudaro po  $n$  atskirų tripletų dalių. Kadangi tripletų aibė  $\mathcal{T} = \Pi_t(O)$ , tai  $\Pi_t = \Pi_s \times \Pi_p \times \Pi_o$ . Iš to išplaukia, kad  $\theta_t = n^3$ . Tai yra pati brangiausia operacija vykdoma vienos projekcijos lygyje. Žinant nors vieną dėmenį— $n$  laipsnis kinta į mažesnę pusę.

Skaičiuojant užklausos vykdymo kainą, kai ieškoma pasirinkto vaisto veikliųjų medžiagų ( 9 pav.), įvertinami tam tikri aspektai. Kaip jau buvo minėta anksčiau, žymuo  $\sqsubseteq$  reiškia reguliariąją išraišką, kur simbolių kiekis ar simbolių rinkinys nebūtinai tiesiogiai atitinka įvestą reikšmę. Tai reiškia, kad įvedus vaistinio preparato pavadinimą, gali būti aptikta  $\{0..n\}$  vaistų, kuriems reikia nustatyti veikliąsias medžiagas. Taip yra todėl, kad vaistų pavadinimai nėra unikalūs. Pavyzdžiui, vaistai "Theraflu" ir "Theraflu Night" yra skirtingi, o įvedus į užklausą "Theraflu" gaunami du vaistiniai preparatai. Jeigu ontologijoje užregistruoti visi vaistai, turintys pavadinime simbolių rinkinį "Theraflu", tokiu atveju projekcija  $\Pi_{subject \rightarrow ?Drug}$  grąžins  $n$  įrašų. Dėl šios priežasties užklausos įvykdymo kaina išauga. Tokia pati situacija yra ir vertinant užklausos vykdymą iš veikliųjų medžiagų perspektyvos. Jeigu vaistui užregistruotos visos ontologijoje įregistruotos veikliosios medžiagos, tokiu atveju gaunamas įvertis iš veikliųjų medžiagų perspektyvos  $\Pi_{\substack{subject \rightarrow ?Drug \\ object \rightarrow ?ActiveSubstance}} = n$ . Todėl šios užklausos blogiausias įvertis būtų  $\theta_{?ActiveSubstance} = n^2$ . Geriausiu atveju užklausos vykdymo įvertis yra  $\omega_{?ActiveSubstance} = n$ , kai randamas vienas vaistinis preparatas su  $n$  veikliųjų medžiagų. Analogiškas įvertis būtų vykdant užklausą pilnai vaistų sudėčiai nustatyti, indikacijas ir kontraindikacijas.

Analizuojant užklausos vykdymo medį, kai ieškoma pasirinktam vaistui alternatyva pagal ATC klasifikaciją ( 11 pav.), taip pat galima įvertinti blogiausią ir geriausią vykdymo kainą. Kaip ir aukščiau aprašytame užklausos medžio, kai ieškoma vaisto veikliųjų medžiagų, įvertyje, taip ir ieškant alternatyvų pagal ATC klasifikaciją, vaisto pavadinimas gali būti aptinkamas ontologijoje  $n$  kartų. Tokiu atveju, gali būti aptikta vaistui priskirtų ATC klasifikacijos kodų taip pat  $n$ . Vaistų, kuriems priskirti tokie patys ATC klasifikacijos kodai taip pat gali būti  $n$ . Tačiau dar egzistuoja ribojimas, kuris sako, jog turi būti eliminuoti vaistai, kurie atitinka pradinę įvestį. Todėl šio užklausos vykdymo medžio blogiausias įvertis būtų  $\theta_{?DrugsByATC} = n^3 - n$ , o tai yra  $\theta_{?DrugsByATC} \approx n^3$ . Skaičiuojant geriausią užklausos vykdymo kainą būtų aptinkamas vienas įrašas, atitinkantis vaisto

pavadinimą. Jis turi priskirtą vieną ATC klasifikacijos kodą. Ieškant vaistų, turinčių priskirtą tą patį ATC kodą, aptinkama  $n$  vaistų. Todėl geriausio atvejo įvertis būtų  $\omega_{?DrugsByATC} = n - 1$ , o tai yra  $\omega_{?DrugsByATC} \approx n$ .

Vertinant užklausos vykdymo medžio, atsakingo už šalutinių poveikių galimą sustiprėjimą (12 pav.), blogiausią ir geriausią vykdymo kainą, galima pasiekti labai blogus greitaveikos rezultatus. Jau žinoma, kad vartotojas įveda du vaistinius preparatus  $d_i, d_j$ . Jeigu vaistiniai preparatai ontologijoje aptinkami  $n$  kartų, tai tokiu atveju vaistų vykdymo kaina išauga  $2n$  karto. Taip pat jeigu kiekvienam vaistiniam preparatui aptinkama  $n$  šalutinių poveikių, šiuo metu vykdymo kaina jau  $2n^2$ . Tačiau šioje užklausoje naudojamos viena brangiausiai kainuojančių operacijų—*group by*. Naudojant *group by* metodą, turi būti praeinama dar kartą per aptiktus duomenis, o tai reiškia, kad vykdymo kaina tampa  $2n^3$ . Todėl  $\theta_{?SideEffectsName} = 2n^3$ . Geriausias šios užklausos rezultatas būtų aptikus vaistų porą, surinkti jų šalutinius poveikius ir sugrupavus gaunamas rezultatas  $\omega_{?SideEffectsName} = 2n^2$ .

Nagrinėjant užklausos medį (13 pav.), kai surandami vaistai analogiškais veikliosiomis medžiagomis ir nustatomi visų šių vaistų skirtingi šalutiniai poveikiai, atkreipiamas dėmesys, jog tai yra dinamiškai formuojamas medis. Tai reiškia, kad  $\exists$  ir  $\nexists$  sąlygos vykdomos tiek kartu, kiek veikliųjų medžiagų buvo aptikta pirminio vaisto kontekste. Tarkime, kad tai yra skaičius  $k$ . Užklausos vykdymo kaina *exists* dalyje yra  $k * n$ . Tokia pati situacija yra su  $\nexists$  sąlyga. Tai reiškia, kad blogiausiu atveju vykdymo kaina bus  $\theta_{?SideEffects} = k^2 * n^3$ . Geriausia užklausos vykdymo kaina, kai pasirinktas vaistas turi tik vieną veikliąją medžiagą, yra  $\omega_{?SideEffects} = n^3$ .

Užklausos medžiui, nustatančiam vaistų tarpusavio sąveiką, kuris pavaizduotas 14 pav. kainos vykdymo kaina apskaičiuojama tokiu būdu. Kadangi metodas vėl gi naudoja reguliariąją išraišką, tai aptinkamas vaistų kiekis ontologijoje blogiausiu atveju yra  $n$ . Kiekvienas vaistas taip pat gali turėti priskirtų veikliųjų medžiagų  $n$ . Todėl surasti vaistų veikliąsias medžiagas kaina yra  $n^2$ . Kadangi šiame metode lyginamos dviejų vaistų veikliųjų medžiagų sąveikos, tai kad išsirinkti visas galimas vaistų sąveikų kombinacijas reikia vykdyti  $n^4$  kartų. O aptikti sąveikas, kurios užregistruotos ontologijoje, taip pat reikia praeiti  $n$  kartų. Jeigu aprašytos predikatų savybės nėra simetrinės, tokiu atveju turi būti praeinamas tripletų tikrinimas iš kitos pusės. Todėl blogiausios kainos įvertis yra  $\theta_{?hasEffect} = n^6$ . Geriausias įvertis tokios užklausos yra  $\omega_{?hasEffect} = n^4$ , jeigu ieškoma konkreitiems dviems vaistams veikliųjų medžiagų sąveikos, vis tiek turi būti praeinama likusiu medžiu.

Nagrinėjant užklausos medį (15 pav.), kur ieškomi vaistai pagal įvestas veikliąsias medžiagas ir tokias medžiagas, kurių sudėtyje neturi būti, galima numanyti, jog paieškos metodas neefektyvus vykdymo kainos prasme. Taip yra todėl, kad net jei į paiešką įvedama konkreti viena veiklioji medžiaga tai turi būti vykdoma dekartu sandauga su visais galimais tripletais, turinčiais ryšį *hasSubstance*. Tik vėliau iš šios dekartu sandaugos projekcijos duomenys nufiltruojami. Kadangi tai yra dinamiškai formuojama užklausa, tai tokiu atveju  $\exists$  dalis gali būti vykdoma  $k$  kartų. Analogiškai su  $\nexists$ , šioje dalyje vykdoma  $m$  kartų. Todėl nors metodas ir naudingas vartotojui, tačiau vykdymo kainos prasme neefektyvus.

## Išvados ir rekomendacijos

Atlikus susijusios literatūros analizę, galima daryti prielaidą, kad semantinio tinklo ir ontologijų technologijos labai plėtojamos, siekiant maksimalių rezultatų kompiuterių samprotavime. Aprašomos naujos arba modifikuojamos jau egzistuojančios įvairių sričių ontologijos, kurios siejamos tarpusavyje į semantinį tinklą. Taip pat kuriami programiniai moduliai, apdorojantys natūralios kalbos sakinius ir susiejantys juos su semantinio tinklo duomenimis. Dėl šios priežasties sistemoje, grįstose ontologijų technologijomis, gali būti apdorojami natūralia kalba užduoti klausimai ir į juos atsakyta ontologijose aptikta kontekstine informacija.

Šiame darbe buvo apžvelgiami semantinio tinklo ir ontologijų kūrimo principai. Išsamiai išnagrinėtos technologijos, naudojamos ontologijų kūrimui ir modifikavimui. Taip pat apžvelgiamos pagrindinės ontologijų modeliavimo kalbos, leidžiančios kompiuteriams nagrinėti semantiškai susietus duomenis.

Remiantis surinkta informacija apie ontologijų kūrimo ir modifikavimo galimybes, buvo nuspręsta įgyvendinti naują vaistų ontologiją. Pasirinkta farmacinė sritis dėl to, nes tai yra aktuali sritis visais laikais visoms amžiaus grupėms. Taip pat temos aktualumas pasižymėjo tame, jog susirgus, ne visada konsultuojamasi su gydytoju ar vaistininku dėl tam tikrų vaistų vartojimo. Dažnai nekreipiamas dėmesys į tai, ar vaistai tarpusavyje suderinami, jei vartojama daugiau nei vienas vaistinis preparatas. Taip pat dažnai perskaitoma tik maža dalis informacijos, kuri teikiama vaistų informaciniuose lapeliuose. O informacija apie galimus šalutinius poveikius ar sąveikas su kitais vaistiniais preparatais, perskaitomos neįdomiai.

Įgyvendinus vaistų ontologiją, duomenų užpildymas tapo komplikuoatas. Duomenų šaltinis, teikiantis su vaistais susijusią informaciją, ją teikė pakankamai siauros apimties. Tai buvo vaisto pavadinimas, veikliosios medžiagos, gamintojas, šalis, vartojimo forma, ar vaistas yra receptinis, ATC klasifikacijos kodas. Tokia informacija kaip indikacijos, kontraindikacijos, vaistų pagalbinės medžiagos, šalutiniai poveikiai buvo renkami tiesiogiai iš vaistų informacinių lapelių, kurie buvo priskirti kiekvienam vaistiniam preparatui atskiru dokumentu. Taip pat automatinio duomenų nuskaitymo nepavyko įgyvendinti dėl natūralios kalbos apdorojimo niuansų.

Taip pat buvo sukurti metodai, atliekantys užklausas sukurtoje vaistų ontologijoje. Metodai nustato vaistų sudėtį, indikacijas, kontraindikacijas, nustato šalutinius poveikius. Nustatomos pasirinktam vaistui alternatyvos pagal ATC klasifikaciją. Jeigu vartojama daugiau nei vienas vaistinis preparatas, tokiu atveju vertinama rizika dėl galimų šalutinių poveikių sustiprėjimų. Taip pat įvertinama vaistų poros tarpusavio sąveika, jei tokia egzistuoja. Dar vienas metodas leidžia nustatyti vaisto pavadinimą, kai žinoma kokia veiklioji medžiaga turi būti vaistų sudėtyje. Jeigu vartotojas turi netoleravimą tam tikroms medžiagoms, nebūtinai veikliosioms, pažymėjęs tokią informaciją gaus atitinkančius rezultatus.

Labiau išplėtotas prototipas galėtų būti naudojamas elektroninėse vaistų parduotuvėse, kai į krepšelį sudedami vaistai, patikrinti ar prekių krepšelio turinys tarpusavyje suderinamas. Jeigu aptinkamas rizika, prieš perkant, vartotoją įspėti apie galimą poveikį ir pateikti rekomendacijas kaip vartoti vaistus. Taip pat toks prototipas galėtų būti išplėtojamas iki viešai prieinamos sistemos, kai vartotojas į sistemą įveda dominančią informaciją, ir gauna atitinkamus rezultatus.

## Ateities darbų gairės

Įvertinus jau atliktus darbus, matomos tobulintinos vietos, kurios turėtų būti įgyvendinamos, norint prototipą paversti į pilnai funkcionuojančią sistemą. Žemiau pateikiami punktai, apibrėžiantys tobulintinas vietas:

- Ontologijos duomenys turėtų būti pilnai aprašyti ir sukelti į ontologiją. Tai reiškia, kad ontologijos duomenų rinkinys būtų kaip įmanoma didesnis, leidžiantis semantiškai sieti duomenis, ir artimesnis natūraliai kalbai. Tokiu būdu būtų galima įgyvendinti sistemą, kuri nagrinėtų natūralios kalbos klausimus ir atpažinus klausime tripletus, gebėtų į juos atsakyti semantiškai susijusiuose ontologijos duomenyse.
- Įgyvendinti metodą, tokį kaip GalenOWL[15], kur būtų registruojama ne tik sąveika tarp vaistų, bet ir vaistų sąveika su ligomis. Išnagrinėjus literatūroje įgyvendintus metodus, dažniausiai tokia informacija nėra registruojama. Dažniausiai registruojama vaistų su vaistais sąveika ir tik konkrečiai ligai ar simptomui, kurį reikia gydyti ar malšinti. Pavyzdžiui, vaistas "tertensif" skirtas "Suaugusiųjų pirminės arterinės hipertenzijos gydymui". Vaistų informacinio lapelio aprašyme yra teikiama informacija—"gydomiems diabetikams, ypač jeigu yra hipokaliemi, svarbu sekti gliukozės koncentraciją kraujyje". Būtų tikslinga užregistruoti ryšį tarp vaisto "tertensif" veikliosios medžiagos ir ligos apie cukrinį diabetą, kad reikalinga sekti gliukozės kiekį kraujyje.
- Kaip įmanoma detaliau aprašyti vaistų sudėtinių dalių sąveikas su kitų vaistų sudėtinėmis dalimis. Tai padėtų tiksliau nustatyti galimas rizikas ir sąveikas, ieškant informacijos apie daugiau nei vieną vaistinį preparatą.
- Išskirstyti vaistinius ingredientus į poklasius. Įgyvendinant ontologiją vienas iš sunkumų buvo, kai kontraindikacijose ar sąveikose su kitais vaistais būdavo nurodoma veikliosios medžiagos klasė. Pavyzdžiui, vaisto "tertensif" informaciniame lapelyje rašoma apie veikliosios medžiagos "Indapamido" galimos sąveikos su "Sisteminio vartojimo nesteroidiniai vaistai nuo uždegimo (NVNU), įskaitant selektyvaus poveikio COX-2 inhibitorius ir dideles salicilatų dozes". "Nesteroidiniai vaistai nuo uždegimo(NVNU)" ir "selektyvaus poveikio COX-2 inhibitoriai" yra vaistinių preparatų veikliųjų medžiagų viršklasiai. Todėl būtų tikslinga veikliąsias medžiagas įkelti į taksonominę hierarchiją.
- Būtų naudinga ontologiją papildyti "paros normos" klase, kur būtų registruojama kokia kiekvienos veikliosios medžiagos paros norma suaugusiam žmogui, vaikui iki 1, iki 2m ir t.t. Tada būtų galima apskaičiuoti ar vartojant kelis vaistus su ta pačia veikliąja medžiaga nebūtų viršijama veikliosios medžiagos paros norma.
- Būtų tikslinga įgyvendinti metodą, kai užregistruojamas veikliųjų medžiagų kiekis konkrečiame vaistiniame preparate. Tokiu atveju būtų galima sekti ar vartojant ne vieną vaistą iš karto, neviršijama leistina paros norma.
- Reikalinga surinkti kokius simptomai gali nustatyti tam tikras ligas. Tokiu atveju, galima būtų nustatyti kokius vaistinius preparatus galima vartoti, esant tam tikriems simptomams. Arba tokioje ontologijoje galima būtų ieškoti ligos pagal išvardintus simptomus. Tačiau šis metodas turėtų būti labiau informacinio pobūdžio, nes pačiam žmogui objektyviai nusistatyti

simptomus ne visada pavyksta. Nebent tokiu metodu naudotųsi gydytojas, turėdamas aišų paciento simptomų.

Igyvendinus visus aukščiau aprašytus punktus, šiame darbe sukurtas sistemos prototipas galėtų tapti sistema, leidžianti vartotojui domėtis jo vartojamais vaistais, tikrinti jų suderinamumą, nevykstant į konsultaciją su gydytoju ar vaistininku. Taip pat toks modelis galėtų būti naudojamas elektroninėse vaistų parduotuvėse, prieš vartotojui išsigyjant vaistus, būtų peržiūrima prekių krepšelio vienetų tarpusavio sąveika ir radus galimas sąveikas ar rizikas, apie tai informuoti vartotoją.



## Literatūros šaltiniai

- [1] Sparqler. Online. <https://www.w3.org/2001/sw/wiki/SPARQLer>.
- [2] Valstybinė vaistų kontrolės tarnyba(vvkt). Online. [www.vvkt.lt](http://www.vvkt.lt).
- [3] Virtuoso sparql query editor. Online. <http://lod.openlinksw.com/sparql>.
- [4] D2r server publishing the drugbank database. Online, 12 2016. <http://wifo5-03.informatik.uni-mannheim.de/drugbank/>.
- [5] Drugbank. Online, 2016. <https://www.drugbank.ca>.
- [6] Sareh Aghaei, Mohammad Ali Nematbakhsh, and Hadi Khosravi Farsani. Evolution of the world wide web: From web 1.0 to web 4.0.
- [7] Tim Berners-Lee. Relational databases on the semantic web, 1998. <https://www.w3.org/DesignIssues/RDB-RDF.html>.
- [8] Tim Berners-Lee, James Hendler, Ora Lassila, et al. The semantic web. *Scientific american*, 284(5):28–37, 2001.
- [9] Jeen Broekstra, Arjohn Kampman, and Frank Van Harmelen. Sesame: A generic architecture for storing and querying rdf and rdf schema. In *International semantic web conference*, pages 54–68. Springer, 2002.
- [10] Lei Chen, Wei-Ming Zeng, Yu-Dong Cai, Kai-Yan Feng, and Kuo-Chen Chou. Predicting anatomical therapeutic chemical (atc) classification of drugs by integrating chemical-chemical interactions and similarities. *PloS one*, 7(4):e35254, 2012.
- [11] World Wide Web Consortium et al. Rdf 1.1 concepts and abstract syntax. 2014.
- [12] World Wide Web Consortium et al. Rdf schema 1.1. 2014.
- [13] Li Ding. Enhancing semantic web data access. 2006.
- [14] Li Ding and Tim Finin. Characterizing the semantic web on the web. In *International Semantic Web Conference*, pages 242–257. Springer, 2006.
- [15] Charalampos Doulaverakis, George Nikolaidis, Athanasios Kleontas, and Ioannis Kompatsiaris. Galenowl: Ontology-based drug recommendations discovery. *Journal of Biomedical Semantics*, 3(1):14, 2012.
- [16] Bob DuCharme. *Learning Sparql*. " O'Reilly Media, Inc.", 2013.
- [17] OWL Working Group. Web ontology language (owl). Online, 12 2012. <https://www.w3.org/2001/sw/wiki/OWL>.
- [18] RDF Working Group. Resource description framework (rdf). Online, 2014. <https://www.w3.org/RDF/>.
- [19] Matthew Horridge, Holger Knublauch, Alan Rector, Robert Stevens, and Chris Wroe. A practical guide to building owl ontologies using the protégé-owl plugin and co-ode tools edition 1.0. *University of Manchester*, 2004.

- [20] Anja Jentzsch, Jun Zhao, Oktie Hassanzadeh, Kei-Hoi Cheung, Matthias Samwald, and Bosse Andersson. Linking open drug data. In *I-SEMANTICS*, 2009.
- [21] Bermejo Julita. A simplified guide to create an ontology. *The Autonomous Systems Laboratory*, 2007.
- [22] Jelena Liachovskaja and Linas Būtėnas. *Ontologijos kūrimas bei jos teisingumo užtikrinimas*. PhD thesis, Vilniaus universitetas, 2009.
- [23] Marek Obitko. Ontologies and semantic web. Online, 2007. <http://www.obitko.com/tutorials/ontologies-semantic-web/>.
- [24] Martin J O'connor, Christian Halaschek-Wiener, and Mark A Musen. Mapping master: a flexible approach for mapping spreadsheets to owl. In *International Semantic Web Conference*, pages 194–208. Springer, 2010.
- [25] Jorge Pérez, Marcelo Arenas, and Claudio Gutierrez. Semantics and complexity of sparql. In *International semantic web conference*, pages 30–43. Springer, 2006.
- [26] Erika Rimkutė. Homoformos dabartinės lietuvių kalbos tekstyne. *Lituanistica*, 2(50):86–101, 2002.
- [27] Delia Rusu, Lorand Dali, Blaz Fortuna, Marko Grobelnik, and Dunja Mladenic. Triplet extraction from sentences. In *Proceedings of the 10th International Multiconference" Information Society-IS*, pages 8–12, 2007.
- [28] Lina Tankelevičienė and Rimantas Sakalauskas. Dalykinės srities ontologija grįsta intelektualioji gedimų diagnostikos sistema. *Jaunųjų mokslininkų darbai*, 5 (21), pages 65–72, 2008.
- [29] Stanford University. Protégé. Online, 2016. <http://protege.stanford.edu/>.
- [30] Daiva Vaišnienė. The lithuanian language in the digital age. 2012.
- [31] Gintarinė vaistinė. Vaistų suderinamumo sistema. Online, 2017. <https://www.suderinamumas.lt/>.
- [32] Nianbin Wang and Xiaofei Xu. A method to build ontology. In *High Performance Computing in the Asia-Pacific Region, 2000. Proceedings. The Fourth International Conference/Exhibition on*, pages 672–673 vol.2, May 2000.
- [33] Semantic Web. Creating semantic web contents with protege-2000. 2001.
- [34] David S Wishart, Craig Knox, An Chi Guo, Dean Cheng, Savita Shrivastava, Dan Tzur, Bijaya Gautam, and Murtaza Hassanali. Drugbank: a knowledgebase for drugs, drug actions and drug targets. *Nucleic acids research*, 36(suppl 1):D901–D906, 2008.