ŠIAULIŲ UNIVERSITETAS

Informacinių technologijų katedra

Kiril Griazev

Pusiau struktūrizuoto internetinio puslapio duomenų įrašų stebėjimo modelis

Magistro darbas

Vadovė dr. S. Ramanauskaitė

Šiauliai, 2013

ŠIAULIŲ UNIVERSITETAS

Informacinių technologijų katedra

TVIRTINU

IT katedros vedėjas doc. dr. M. Bernotas
2013-05-27

Pusiau struktūrizuoto internetinio puslapio duomenų įrašų stebėjimo modelis

Informatikos inžinerijos magistro darbas

Autorius
ITM-11 gr. magistrantas K. Griazev
2013 m. gegužės 27 d.

Vadovė
IT katedros lektorė dr. S. Ramanauskaitė
2013 m. gegužės 27 d.

Recenzentai
IT katedros docentė dr. A.Slotkienė
2013 m. gegužės __ d.
IT katedros docentas dr. E. Paliulis
2013 m. gegužės __ d.

Šiauliai, 2013

TVIRTINU

Informacinių technologijų

katedros vedėjas doc. M.Bernotas

2013 m. _____mėn. _____d.

Magistro darbo užduotis

Studentui **KIRIL GRIAZEV**

Darbo tema: **PUSIAU STRUKTŪRIZUOTO INTERNETINIO PUSLAPIO DUOMENŲ ĮRAŠŲ STEBĖJIMO MODELIS (MODEL FOR MONITORING OF SEMI STRUCTURED DATA RECORDS IN WEB PAGES)**

Patvirtinta 2013 m. _____ mėn. _____ d. potvarkiu Nr. _____

1. Suprojektuoti duomenų stebėjimo internetiniuose puslapiuose algoritmą, kuris būtų kiek galima mažiau priklausomas nuo internetinio puslapio HTML kodo struktūros:
   1.1. Išanalizuoti duomenų išrinkimo iš internetinių svetainių problemą, principus ir egzistuojančius sprendimus.
   1.2. Ištirti duomenų atrankos iš internetinių puslapių technologijų našumą ir taikymo galimybes.
   1.3. Suprojektuoti naują duomenų išrinkimo algoritmą, kuris atrinkdamas duomenis iš internetinių svetainių atsižvelgia į jų panašumą su ankščiau sistemos surinktais duomenimis.
   1.4. Ištirti naujai pasiūlyto duomenų išrinkimo algoritmo tikslumą ir našumą, pritaikant jį stebėti valiutų kursų pokyčius.
2. Aiškinamojo rašto turinys turi atitikti Informacinių technologijų katedros studentų magistro darbų metodinius nurodymus.
3. Magistro darbas pateikiamas su įrišta darbo užduotimi, įdėtu vadovo atsiliepimu ir kompaktine plokštele.
4. Kompaktinėje plokštelėje įrašyti visus su sukurtu algoritmu ir atliktais tyrimais susijusius failus ir magistro darbo aiškinamąjį raštą.
5. Gynimo metu pateikiama pristatymo pateiktis ir liudijimas, kad magistro darbas yra įkeltas į Lietuvos ETD informacinę sistemą.
6. Paskutinioji magistro darbo pristatymo diena – 2013 m. gegužės 23 d.

Baigiamojo darbo vadovė **lekt. dr. S. Ramanauskaitė**

2013 m. vasario 4 d.

Studentas **K. Griazev**

2013 m. vasario 4 d.

SANTRAUKA

Pusiau struktūrizuoto internetinio puslapio duomenų įrašų stebėjimo modelis

Šiuolaikinės informacinės sistemos stengiasi duomenis atnaujinti automatizuotai, nenaudojant žmogiškųjų išteklių, tačiau susiduria su netiksliu duomenų identifikavimu pusiau struktūrizuotose internetiniuose puslapiuose problema.

Šiame darbe pateikiamas suprojektuotas duomenų stebėjimo internetiniuose puslapiuose algoritmas, kuris yra minimaliai priklausomas nuo internetinio puslapio HTML kodo struktūros, bei atlieka duomenų atpažinimą atsižvelgdamas į sistemai žinomus duomenys.

Pagal pasiūlytą modelį realizuotas valiutų kursų stebėjimo internetiniuose puslapiuose įrankis nenusileidžia panašioms egzistuojančioms sistemoms tikslumo atžvilgiu, o gebėjimu tinkamai išrinkti reikiamus duomenis kintant jų pateikimo formai žymiai jas lenkia.

SUMMARY

Model for Monitoring of Semi-Structured Data Records in Web Pages

Current information systems are made to update their data automatically, without any help from humans, but the problem of incorrect data recognition in semi-structured web pages arises.

In this thesis you can find an algorithm, which was made to monitor data that is available from a web page. This algorithm is almost non-dependant on web page structure and does data recognition based on the data that is already known.

Using this algorithm model a tool was created to monitor currency exchange rates. This tool performs on the same level as other available tools in terms of precision and accuracy of collected data and it also reacts to changes of the data source structure.

TERMINŲ IR SANTRUMPŲ ŽODYNĖLIS

DIV – HTML žymė, kuri nurodo sekciją dokumente, naudojama kaip konteineris kitiems elementams.

DOM – HTML ir XML dokumentų apdorojimo metodas, kuris dokumentą apdorojimo metu analizuoja kaip medį

Duomenų atrankos algoritmasmetodas – metodas, skirtas iš turimo duomenų šaltinio atrinkti tiksliai nurodytus duomenis (pagal jų buvimo kelią, šabloną ar pan.)

Duomenų išgavimas – procesas, kurio metu pagal tam tikrus požymius atrenkami ieškomi duomenys

Duomenų išrinkimo algoritmas – algoritmas, apjungiantis duomenų atrankos metodą ar metodus ir reikiamų duomenų atpažinimą

FALSE-NEGATIVE – klaidingai neigiamas įvykio įvertinimas (*angl.*)

FALSE-POSITIVE – klaidingai teigiamas įvykio įvertinimas (*angl.*)

F-SCORE – f-rodiklis, apjungia tikslumo rodiklio ir atrinkimo rodiklio reikšmes, kas leidžia išreikšti algoritmo našumą viena reikšme

HTML – (*angl. Hyper text Markup Language* „Hiperteksto žymėjimo kalba“) – tai kompiuterinė žymėjimo kalba, naudojama pateikti turinį internete

HTML PURIFIER – įrankis skirtas HTML kodo paruošimui apdorojimui, leidžia pašalinti nereikalingus HTML kodo elementus, klases, atributus ir t.t.

INDEX – gylis, tai tam tikro elemento pozicija HTML dokumente, kuri nustatoma atsižvelgiant į HTML dokumento hierarchiją (*angl.*)

Internetinė svetainė – vienas ar daugiau tarpusavyje susijusių internetinių puslapių. Visada turi pradinį puslapį, pasiekiamą internetu per tinklavietės internetinį adresą ir (arba) IP adresą.

Internetinis puslapis – (analogai tinklalapis) yra informacijos išteklis žiniatinklyje, kuris gali būti pasiektas naudojantis naršykle. Informacija dažniausiai pateikiama hipertekstinės žymėjimo kalbos (HTML) arba išplečiamos hiperteksto žymėjimos kalbos (XHTML) formatu.

NLP – natūralios kalbos apdorojimas (*angl. Natural Language Processing)*

PATH – kelias iki tam tikro dokumente esančio elemento, kuris yra sudaromas atsižvelgiant į dokumente esančiu elementu hierarchiją (*angl.*)

PHP – plačiai paplitusi dinaminė interpretuojama programavimo kalba (*angl.* Hypertext Preprocessor), sukurta 1995 m. ir specialiai pritaikyta interneto svetainių kūrimui.

PRECISION – atrinkimo rodiklis, vienas iš duomenų išgavimo algoritmo našumo įvertinimo kriterijų

RECALL – tikslumo rodiklis, vienas iš duomenų išgavimo algoritmo našumo įvertinimo kriterijų

REGEX – tai taisyklių rinkinys, kuris leidžia sukurti abstrakčius šablonus duomenų atpažinimui

SAX – įvykiais paremtas XML dokumentų apdorojimo metodas

SPAN – HTML žymė skirta kitų HTML elementų grupavimui

TABLE – tai HTML žymė, skirta lentelėms aprašyti

TAG – HTML žymės naudojamos HTML puslapių struktūros apibrėžimui. Tai puslapio žymėjimo elementas, kurio pagalba naršyklei nurodoma kaip „suprasti“ ir vaizduoti puslapį. (*angl.*)

TRUE-POSITIVE – teisingai teigiamas įvykio įvertinimas (*angl.*)

wrapper – aplankas arba duomenų konteineris, kuries leidžia atskirti pasirinkta duomenų bloka nuo kitų dokumente esančių duomenų, tai leidžia sumažinti apdorojamų duomenų kiekį

XML – bendros paskirties duomenų struktūrų bei jų turinio aprašomoji kalba.

XPATH – XML Path Language (*angl.*), tai užklausų kalba, kuri leidžia pasirinkti norima XML dokumento elementą

Žiniatinklis interneto dalis, resursai, kuriuos internete galima pasiekti naudojant internetinį adresą

PAVEIKSLELIŲ SARAŠAS

LENTELIŲ SARAŠAS

TURINYS

ĮVADAS

Informacijos kiekis internete didėja kasdien, atsiranda tūkstančiai naujų interneto svetainių, todėl atsiranda poreikis visą šią informaciją susisteminti ir pateikti vartotojams patogia forma. Dėl šio poreikio atsiranda įrankiai (svetainės/programos), kurie leidžia surinkti vartotoją dominančią informaciją ir ją pateikti vienoda forma iš visų šaltinių. Tuo tarpu jei vartotojui reikėtų aplankyti kiekvieną iš svetainių atskirai, jis sugaištų daug daugiau laiko, kadangi kiekviena iš svetainių turi savo duomenų pateikimo formatą. Būtent tokie informaciją renkantys įrankiai naudoja skirtingus duomenų surinkimo algoritmus, kurie analizuoja svetainių-šaltinių turinį ir išrenka reikalingą informaciją.

Šio darbo **tyrimo objektas** – duomenų nagrinėjimo technologijos ir principai, jų efektyvumas, našumas ir pritaikymo galimybės.

Šio darbo **tikslas** – suprojektuoti duomenų stebėjimo internetiniuose puslapiuose algoritmą, kuris būtų kiek galima mažiau priklausomas nuo internetinio puslapio HTML kodo struktūros. Darbe keliami šie uždaviniai:

1. Išanalizuoti duomenų išrinkimo iš internetinių puslapių problemą, principus ir egzistuojančius sprendimus;
2. Ištirti duomenų atrankos iš internetinių puslapių technologijų našumą ir taikymo galimybes;
3. Suprojektuoti naują duomenų išrinkimo algoritmą, kuris atrinkdamas duomenis iš internetinių svetainių atsižvelgtų į jų panašumą su sistemai žinomais duomenimis;
4. Ištirti naujai pasiūlyto duomenų išrinkimo algoritmo tikslumą ir našumą, pritaikant jį stebėti valiutų kursų pokyčius.

Šis darbas svarbus praktine prasme, nes pasiūlytas algoritmas leis supaprastinti duomenų išgavimo procesą iš stebimų internetinių puslapių, neprisirišant prie konkrečios puslapyje naudojamos kodo struktūros.

Siūlomas duomenų išrinkimo iš internetinių puslapių algoritmas yra naujas moksline prasme, nes duomenų išdėstymo puslapyje struktūrą nustato automatiškai pagal jame pateikiamų duomenų panašumą su sistemai jau žinomais duomenimis, tuo tarpu kiti egzistuojantys sprendimai remiasi iš anksto žinomais duomenų šablonais arba jų nustatymu pagal duomenų išdėstymo tvarką ar eiliškumą internetiniame puslapyje.

Darbe naudojama literatūros apžvalga, lyginamoji analizė, sisteminė ir eksperimentinė analizė, o jo metu gauti rezultatai pristatyti dvejose mokslinėse konferencijose:

- „Struktūrinių duomenų išgavimas nestruktūrizuotose informacinėse sistemose“. Šiaulių universiteto Technologijos fakulteto 7-oji tarptautinė mokslinė konferencija „Jaunųjų mokslininkų darbai“, 2012 gegužės 18 d.
- „Pusiau struktūrizuoto internetinio puslapio duomenų įrašų stebėjimo modelis“. Šiaulių universiteto Technologijos fakulteto 8-oji tarptautinė mokslinė konferencija „Jaunųjų mokslininkų darbai“, 2013 gegužės 15d.

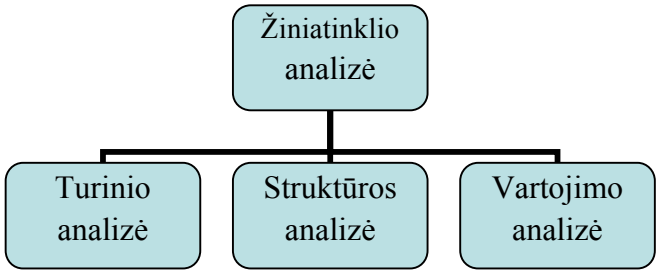
1. DUOMENŲ IŠRIKIMAS INTERNETINIUOSE PUSLAPIUOSE

1. 1 Žiniatinklio duomenų analizės ir nagrinėtojų principai

Neretai kuriant naują projektą, kuriame naudojami dažnai atsinaujinantys duomenys, vienas iš pagrindinių reikalavimų yra kuo didesnis sistemos automatizavimas, kad sistema reikalautų minimalios priežiūros. Dažnai šiam reikalavimui įgyvendinti naudojami duomenų nagrinėtojai (*angl.* parsers) ir žiniatinklio duomenų gavyba (*angl.* web mining) [4–12, 21].

Duomenų nagrinėtojų paskirtis yra analizuoti jiems paduodamą duomenų srautą, siekiant išfiltruoti reikiamus duomenis [10, 11]. Analizuojant duomenis, duomenų nagrinėjimo algoritmai ieško tam tikrų vietų duomenyse, kurios atitiktų iš anksto nustatytus paieškos šablonus. Dažniausiai duomenų nagrinėjimas atliekamas keliais žingsniais. Atlikus kiekvieną žingsnį, analizuojamas duomenų kiekis sumažėja ir pasikeičia ieškomų duomenų šablonas, taip šis procesas vyksta tol, kol galiausiai algoritmas randa ieškomus duomenis arba nebeturi analizuojamų duomenų tolesnei jų analizei.

Žiniatinklio analizė yra sudėtingas procesas, nes reikalauja skirtingų duomenų srautų analizės ir susideda iš kelių dalių (*žr. 1 pav.*): žiniatinklio turinio analizė, žiniatinklio struktūros analizė ir žiniatinklio vartojimo analizė.



1 pav. *Žiniatinklio analizės klasifikavimas[13]*

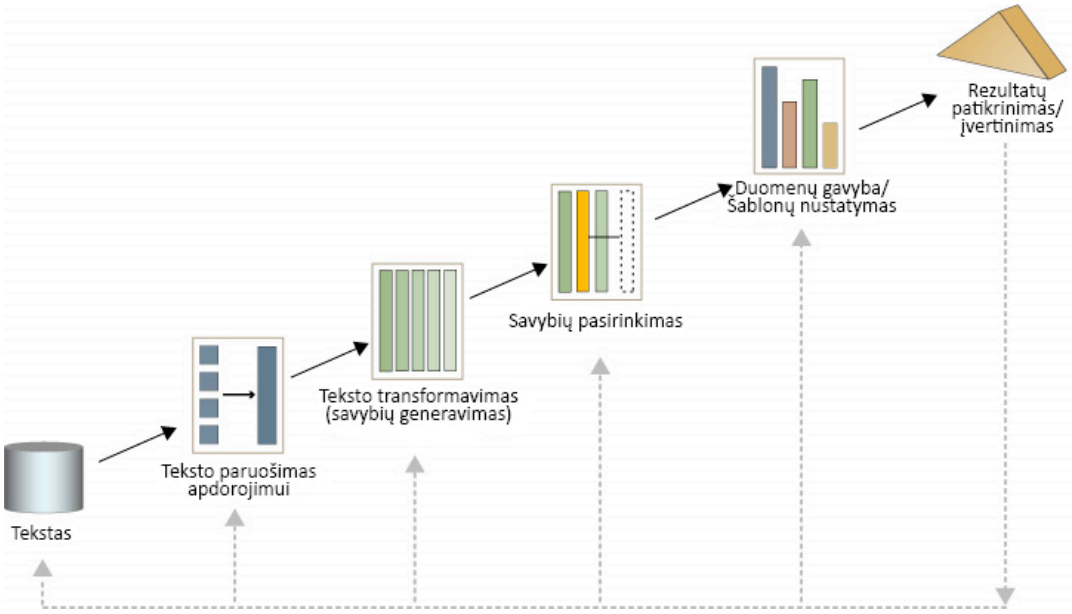
Žiniatinklio turinio analizės metu naudingos informacijos ieškoma iš žiniatinklio turinio, duomenų ir dokumentų [20]. Žiniatinklio duomenys susideda iš: teksto, paveikslėlių, garso, vaizdo, meta-duomenų ir nuorodų.

Maždaug 90 % pasaulinių duomenų yra laikomi nenuosekliuose formatuose [3, 15], todėl reikalinga automatizuota teksto analizės sistema. Teksto analizė tai naujos informacijos atradimas kompiuterio pagalba, automatinis informacijos išgavimas iš dažniausiai didelio kiekio skirtingų nenuoseklių tekstinių šaltinių.

Teksto analizės procesas gali būti iteratyvus ir kartotis atitinkamą skaičių kartų vis generuojant naują informaciją iš jau gautų duomenų (*žr. 2 pav.*).

Tipiško žiniatinklio struktūra panaši į grafus, ji susideda iš tinklalapių kurie yra kaip viršūnės, o nuorodos tarp puslapių kaip briaunos, jungiančios du susijusius puslapius [2]. Žiniatinklio struktūros analizė tai procesas, skirtas atrasti informaciją iš žiniatinklio. Gaunama informacija apie žiniatinklio tinkamumą ir kokybę. Šis analizės tipas gali būti įvykdytas kaip dokumento lygmuo arba kaip hipersaito lygmuo.

Tinklo struktūros analizė nusako bendrai viso tinklapio hierarchiją ir iš esmė gali būti naudojama kaip būdas atrinkti kuri žiniatinklio dalis skiriama navigacijai ar tinklapio statiniams elementams. Tokiu būdu galima susiaurinti esminio teksto gavimo ribas visoje sistemoje.



2 pav. *Teksto analizės procesas[15]*

Taip pat žiniatinklio struktūros pagalba galima išgauti informacija apie tai kokia turinio valdymo sistema naudojama, kas taip pat gali suteikti daugiau informacijos apie to žiniatinklio kodo išdėstymą ir galimas esminės informacijos vietas jame.

Žiniatinklio vartojimo analizė taip pat žinoma kaip žiniatinklio žurnalinių įrašų (*angl.* log) analizė dažniausiai naudojama vartotojų veiksmų sistemoje analizei, tačiau pasitelkus intelektualias sistemas, tokia informacija taip pat gali tarnauti ir kaip tam tikri kriterijai skaitomiausių tinklapio dalių aptikimui.

1. 2 Duomenų išgavimo internetiniuose puslapiuose problematika

Mokslinė literatūra, informatikos srityje, pateikia daug straipsnių apie duomenų gavybos iš internetinių puslapių problemas. 2002 metais buvo pristatytas tyrimas, kur buvo aprašoma sistemų, skirtų duomenų išgavimui iš interneto puslapių klasifikacija [7]:

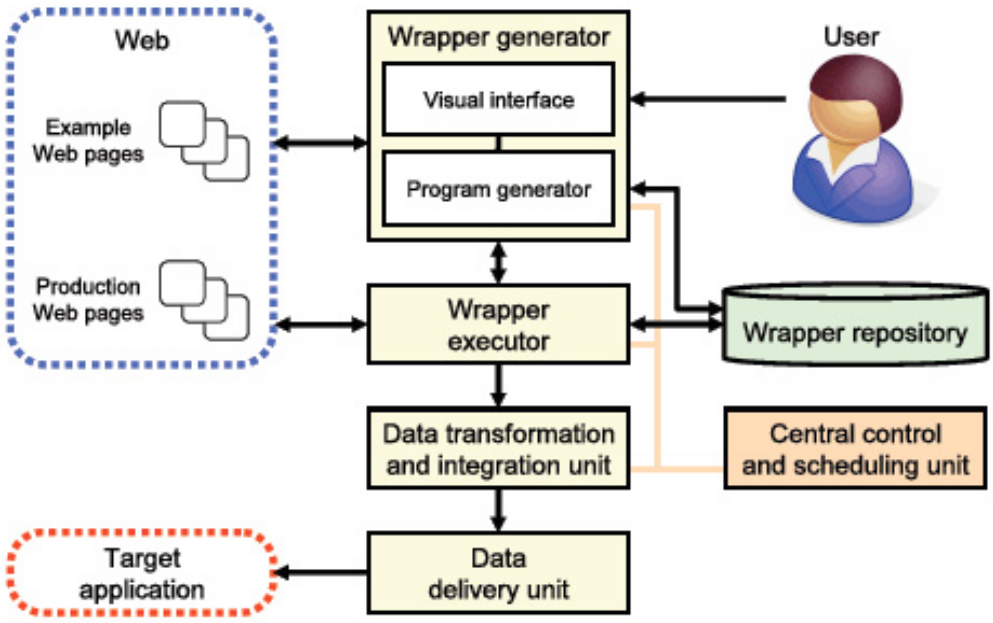
- HTML suprantantys įrankiai (*angl.* HTML-aware) – įrankiai, kurie paremti struktūrinėmis HTML savybėmis. Prieš atliekant duomenų išgavimą, dokumentas paverčiamas medžių, kuris atvaizduoja HTML žymų hierarchiją dokumente.
- Natūralios kalbos apdorojimu paremti įrankiai (NLP-based (*angl.* natural language processing) tools) – įrankiai, naudojami tam siekiant išgauti duomenis iš natūralios kalbos dokumentų. Dažniausiai šie įrankiai naudoja filtravimą, dalinį kalbos žymėjimą (*angl.* part-of-speech tagging), leksinį semantinį žymėjimą (*angl.* lexical semantic tagging) tam, kad nustatytų ryšius tarp frazių ir sakinio elementų. Ryšių dėka nustatomos duomenų išgavimo taisyklės.
- Aplanko įterpimo įrankiai (*angl.* Wrapper induction tools) – įrankiai, kurie sukuria duomenų išgavimo taisykles, paremtas skyrikliais, kurie nustatomi išanalizavus kelis, sistemai apmokyti skirtus duomenų rinkinius. Pagrindinis šių įrankių skirtumas nuo

NLP įrankių yra tas, jog šie įrankiai vadovaujasi ne lingvistiniais požymiais, o formatavimo savybėmis, kuriuos tiksliai nusako kiekvieno ieškomo duomenų fragmento struktūrą.

- Modeliu paremti įrankiai (*angl.* Modeling-based) – tai įrankiai, kurie ieško dokumente fragmentų, kurie atitiktų sistemai žinomą struktūrą. Šių įrankių veikimo algoritmas yra panašus į aplanko įterpimo įrankių algoritmą
- Ontologija paremti įrankiai (Ontology-based) – visi prieš tai aprašyti įrankiai vadovaujasi dokumento pateikimo struktūra, tam kad iš jos sudarytų duomenų išrinkimo taisykles ir šablonus. Tačiau duomenų išgavimas gali būti vykdomas atsižvelgiant būtent į duomenis. Šie įrankiai reikalauja specifinio pritaikymo ir gali būti naudojami tam, kad rastų specifines, duomenyse pateikiamas konstantas, su kuriomis vėliau yra sukuriami duomenų objektai.

Laender darbe buvo pristatytas kriterijų sąrašas ir kokybinė internetinių puslapių duomenų išgavimo sistemų analizė [7]. Tais pačiais metais Kushmerick pristatė savo požiūrį į problemą, įskaitant aplanko įterpimo metodo pritaikymą ir tokios sistemos priežiūrą [14].

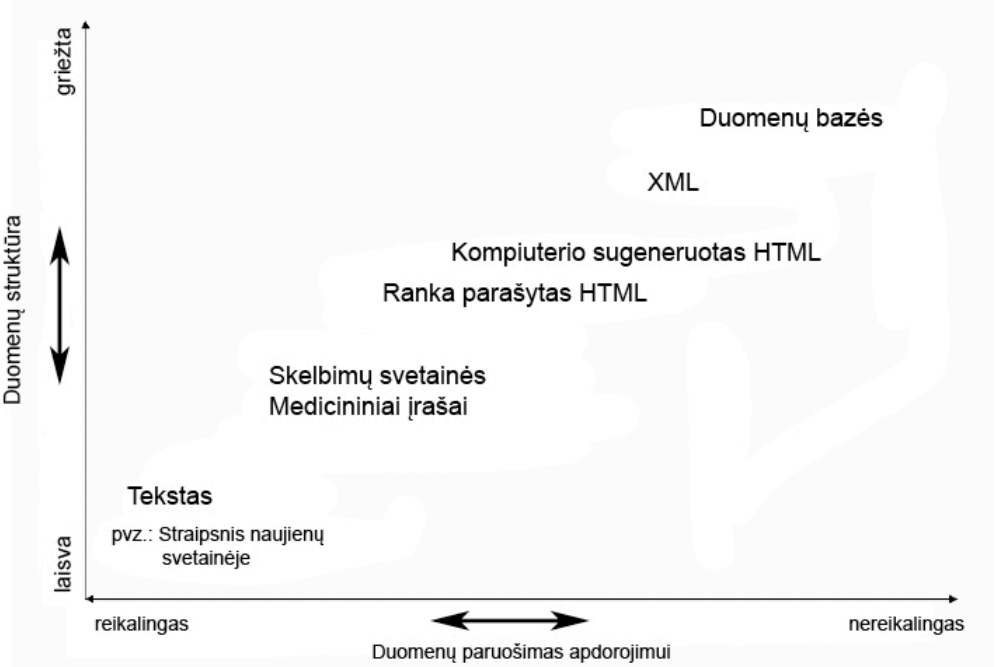
Kuhlins ir Tredwell apžvelgė wrapper generavimo įrankius jau 2003 metais, nors jų tyrimo duomenys ir gali būti pasenę, tačiau tyrimo metodai yra tikrai verti dėmesio, nes apžvelgia didelį skaičių komercinių ir nekomercinių duomenų išgavimo įrankių, bei pateikia jų pagrindines savybes [19]. Taip pat wrapper induction problemos tyrimus atliko Flesca, Kaiser ir Miksch, kurių tyrimai apžvelgė skirtingas metodo panaudojimo galimybes, technologijas ir įrankius [23, 24]. Pastarieji taip pat sumodeliavo reprezentacinę informacijos išrinkimo sistemos architektūrą, kuri pateikta 3 paveikslėlyje. Šis paveikslelis parodo aplanko įterpimo sistemos (*angl.* Wrapper induction) architektūrą. Šioje architektūroje galima išskirti kelias pagrindines dalis – duomenų šaltiniai, sistemos valdymo pultas, aplanko sukurimo ir aplanko duomenų apdorojimo moduliai, aplankų duomenų bazė, sistemos automatizavimo modulis.



3 pav. *Tipinė internetinio puslapio duomenų išrinkimo sistemos architektūra [14]*

Chang, 2006 metais pristatė trijų matmenų (turinio, struktūros ir panaudos duomenų) internetinių puslapių duomenų išrinkimo sistemų kategorizavimą, besiremdamas duomenų išrinkimo užduoties sunkumu, naudojamomis technikomis bei sistemos automatizavimo lygių [13], o Fiurama 2007 metais pritaikė šiuos kriterijus tam, kad galėtų klasifikuoti keturis naujus duomenų išgavimo įrankius [26].

Sarawagi 2008 metais išleido savo darbą apie informacijos išgavimą [31], kuriame apžvelgiami duomenų gavybos tyrimai, atlikti per pastaruosius 2 dešimtmečius. Taip pat šiame darbe pateikiamas duomenų gavybos klasifikavimas pagal duomenų gavybos užduoties pobūdį, naudotus metodus duomenims išgauti, skirtingus analizei skirtu duomenų pateikimo būdus, bei pagal gautų rezultatų tipus. 2009 metais Baumgartner išleido trumpą apžvalgą, kurioje apžvelgė tuometines srities naujoves[30].



4 pav. *Įvairių e-duomenų kategorizavimas[18]*

4 paveikslėlyje pateikiamas įvairių e-duomenų kategorizavimas. Galima pastebėti, kad tie duomenys kurie yra lengvai apdorojami kompiuteriais turi griežtesnę struktūrą negu tie kurie yra sunkiau apdorojami, todėl galima teigti kad didėjant duomenų struktūrizavimui proporcingai didės duomenų apdorojimo galimybės ir tikslumas, kadangi duomenys taps lengviau apdorojami.

2. INTERNETINIO PUSLAPIO DUOMENŲ ATRANKOS METODŲ TYRIMAS

Kuriant naują duomenų išrinkimo iš internetinių puslapių algoritmą, visų pirma siekiama įverti galimų naudoti duomenų atrankos algoritmų savybes ir jomis remiantis formuoti naują duomenų išrinkimo algoritmą.

Todėl šio tyrimo tikslas yra išsiaiškinti kuris iš galimų duomenų atrinkimui HTML kode naudojamų metodų yra optimaliausias norint išgauti skaitinius duomenys iš HTML dokumento.

2. 1 Internetinio puslapio duomenų atrankos metodai

Šiuo metu egzistuoja trys pagrindiniai duomenų atrankos iš internetinių puslapių metodai: DOM, SAX ir regex šablonai.

DOM (*angl.* Document Object Model) tai objektiškai orientuotas interneto svetainės atvaizdavimas. Pasinaudojant DOM, HTML dokumentą galima įsivaizduoti kaip medį (*žr. 5 pav.*), kuriame kiekviena žymė vaizduojama kaip medžio šaka, kuri dar gali dalintis į smulkesnes šakas ir t.t. Tai palengvina HTML kodo struktūros suvokimą ir darbą su duomenimis, svetainės struktūra, nes galima naudoti užklausas, konkrečios šakos gavimui.

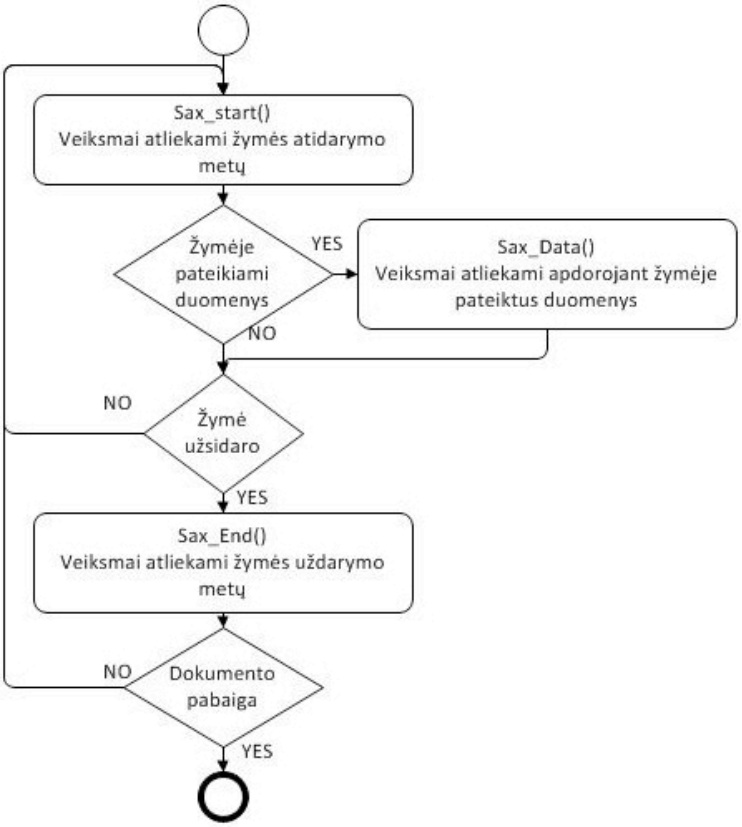
5 pav. *Lentelės fragmentas, pavaizduotas kaip medis*

Regex (*angl.* Regular Expression) šablonai leidžia glaustai ir tiksliai nusakyti ieškomus duomenis. Norint nusakyti ieškomus duomenis, dažniausiai šablonai susideda ne tik iš ieškomų simbolių, bet taip pat iš specifinių paieškos taisyklių. Pavyzdžiui jei bandoma rasti du žodžius, kuriuose skiriasi tik viena raidė, šablonus galima aprašyti keliais budais: *medis|medus* arba *med(i|u)s*. Abu šie šablonai nusako tuos pačius ieškomus žodžius. Taip pat sudarant šablonus gali būti naudojami specialus simboliai, kurie nusako simbolių skaičių prieš arba po ieškomo fragmento, ar ieškomas fragmentas turi būti vieno žodžio, ar jis turi būti eilutės pradžioje ar pabaigoje ir t.t. Panaudojant specialius simbolius galima sukurti kompleksinius šablonus specifinėms duomenims išgauti.

SAX yra įvykiais paremtas duomenų apdorojimo metodas, todėl galima išskirti 3 pagrindinius duomenų apdorojimo etapus:

- Žymės atidarymas (*angl.* opening tag).
- Žymėje talpinami duomenys (*angl.* data).
- Žymės uždarymas (*angl.* closing tag).

6 paveikslėlyje atvaizduojamas SAX algoritmo veikimo principas. Schema skirta suprasti SAX veikimo principą, todėl nėra visiškai detali. Taip pat 6 paveikslėlyje pateiktoje schemoje daroma prielaida, kad analizuojamas duomenų failas yra atitinkantis reikalavimus (*angl.* valid), nes tai yra vienas iš reikalavimų norint apdoroti duomenys SAX metodu.



6 pav. *SAX metodo taikymo principinė schema*

SAX metodo taikymas yra labai lankstus ir gali priklausyti nuo to, kokį uždavinį reikia spręsti. Bendru atveju, kuomet reikia atrinkti konkrečioje žymėje esančius duomenis, SAX metodas veikia taip, kaip pateikta 6 paveikslėlyje:

Sax_start() metodas indikuoja kuomet atveriama žymė, todėl įvertinus ar toje žymėje yra reikiamų duomenų, patys duomenys gaunami Sax_data() metodo pagalba. Siekiant įvertinti kur baigiasi viena žymė ir prasideda kita, būtina sekti ir Sax_end() metodo savybes, kurių pagalba nustatoma kada žymė užsidarė. Toks ciklas vykdomas tol, kol dokumente yra bent viena dar neišnagrinėta žymė.

2. 2 Internetinių puslapių duomenų atrankos metodų tyrimas

Siekiant įvertinti visų trijų internetinių puslapių duomenų atrankai galimų taikyti metodų savybes, atliekamas tyrimas, kurio metu šie trys duomenų atrankos metodai yra pritaikomi PHP programavimo kalboje ir jų pagalba renkami valiutų kursų duomenys iš Swedbank valiutų kursų puslapio (https://ib.swedbank.lt/private/home/more/pricesrates/rates?language=LIT).

Tyrime analizuojamos kelios situacijos, kurių metu iš valiutų kursų pateikimo lentelės internetiniame puslapyje gaunami:

- visi konkrečios eilutės duomenys;
- visi konkretaus stulpelio duomenys;

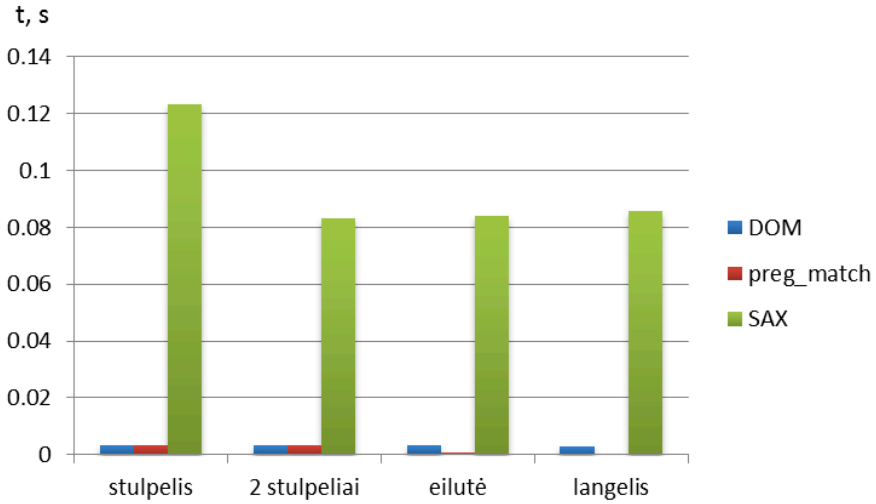
- konkretaus lentelės langelio duomenys.

Kadangi šių naudojamų metodų veikimui turi įtakos HTML kodo atitikimas aprašytam standartui, tai tyrimas bus vykdomas su esamu Swedbank HTML kodu ir su šiek tiek pataisytų (pataisant esamas HTML kodo atitikimo standartui klaidas).

Tyrimo metu bandoma nustatyti šių metodų sunaudojamos atminties kiekį ir duomenų atrinkimui reikalingą laiką. Tame pačiame PHP kode įterpiamas programinis kodas, skirtas kodo vykdymo laiko ir jo metu naudojamos atminties kiekiui įvertinti.

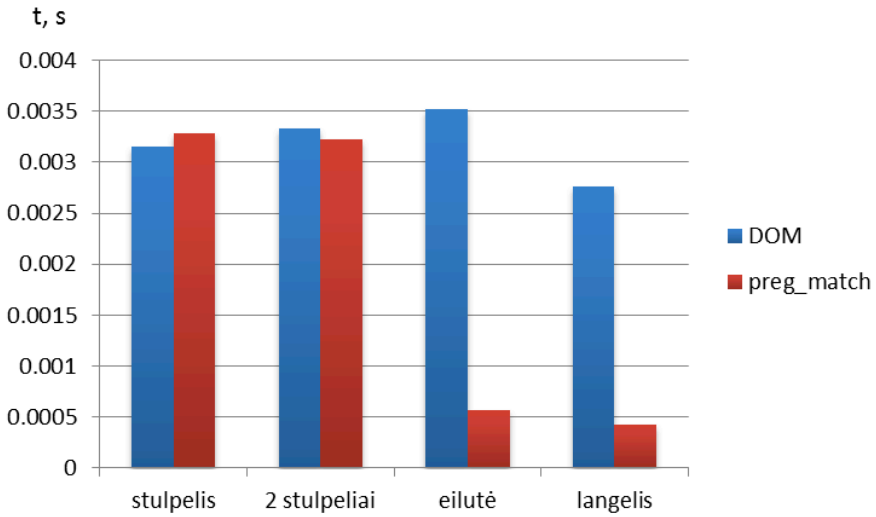
2. 3 Internetinių puslapių duomenų atrankos metodų tyrimo rezultatai

Vienas iš svarbiausiu duomenų apdorojimo rodiklių yra apdorojimo laikas. Iš trijų analizuojamų duomenų išrinkimo metodų SAX turi vieną išskirtini reikalavimą, apdorojant duomenis su SAX apdorojamas dokumentas turi atitikti standarto reikalavimus. Dažniausiai jei apdorojamas dokumentas neatitinka standarto, jo apdoroti su SAX nepavyksta. Būtent dėl šitos priežasties netinkamo HTML kodo apdorojimo laikas su SAX yra žymiai didesnis už DOM ir Regex(preg_match) (*žr. 7 pav.*).



7 pav. *Neatitinkančio standarto HTML kodo apdorojimo laikas*

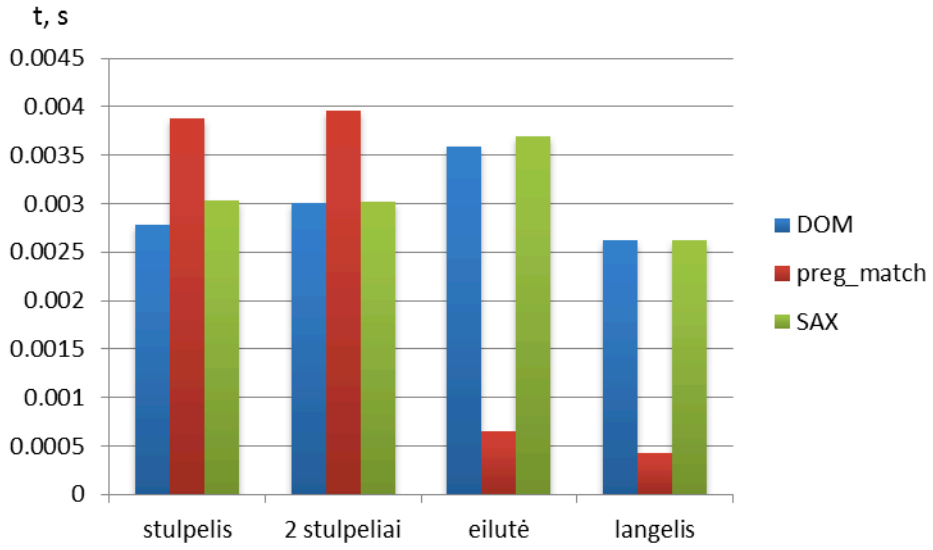
Kadangi neatitinkantis standarto HTML apdorojimo laikas su SAX yra ženkliai didesnis už DOM ir Regex(preg_match), DOM ir Regex(preg_match) neatitinkančio standarto HTML apdorojimo laikas pateikiamas atskirai (*žr.* ***Error! Reference source not found.***.).



8 pav. *Neatitinkančio standarto HTML, apdorojimo laikas (DOM ir preg_match)*

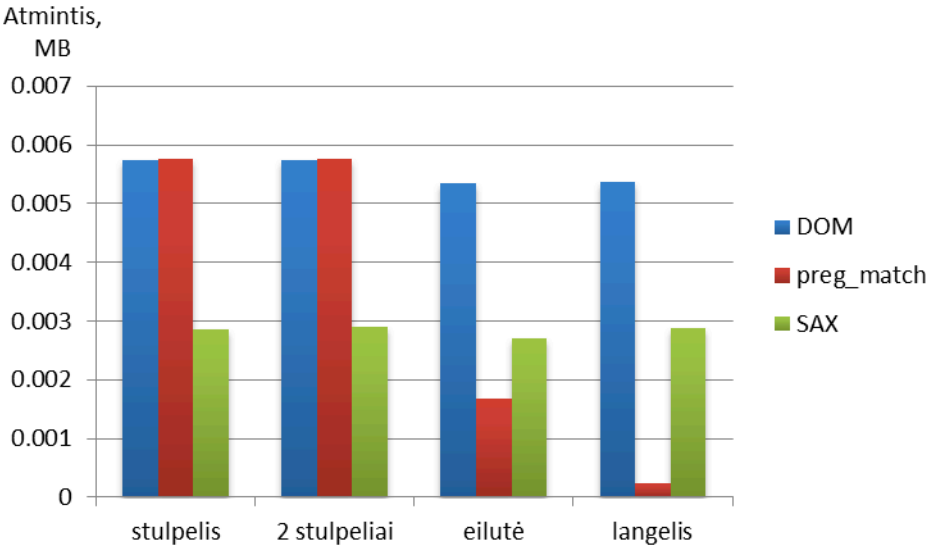
Iš rezultatų matyti, kad apdorojimo laikas su DOM yra beveik stabilus ir svyravimai yra menki. Tuo tarpu naudojant Regex (preg_match) apdorojimo laikas priklauso nuo ieškomų duomenų kiekio, bei ieškomų duomenų pozicijos HTML kode. Dėl šios priežasties dokumento apdorojimo laikas išrenkant eilutės ir langelio duomenis yra daug mažesnis nei išrenkant duomenis iš vieno ar dviejų stulpelių. Gali atrodyti jog laikas, kuris sugaištamas išrenkant duomenis iš eilutės ir stulpelio turėtų sutapti, tačiau taip nėra, nes duomenys, kurie išdėstyti stulpeliais HTML kode yra išdėstyti toliau vienas nuo kito, nenuosekliai, o eilutėje pateikiami duomenys eina nuosekliai vienas po kito, šį skirtumą galima pamatyti 9 paveikslėlyje.

Apdorojant atitinkantį standartą HTML dokumentą DOM ir SAX apdorojimo laikas yra labai panašūs, Regex (preg_match) kaip ir analizuojant neatitinkantį standarto kodą užtrunka mažiau, jeigu ieškomas mažesnis duomenų fragmentas (eilutė, langelis) (*žr. 9 pav.*).



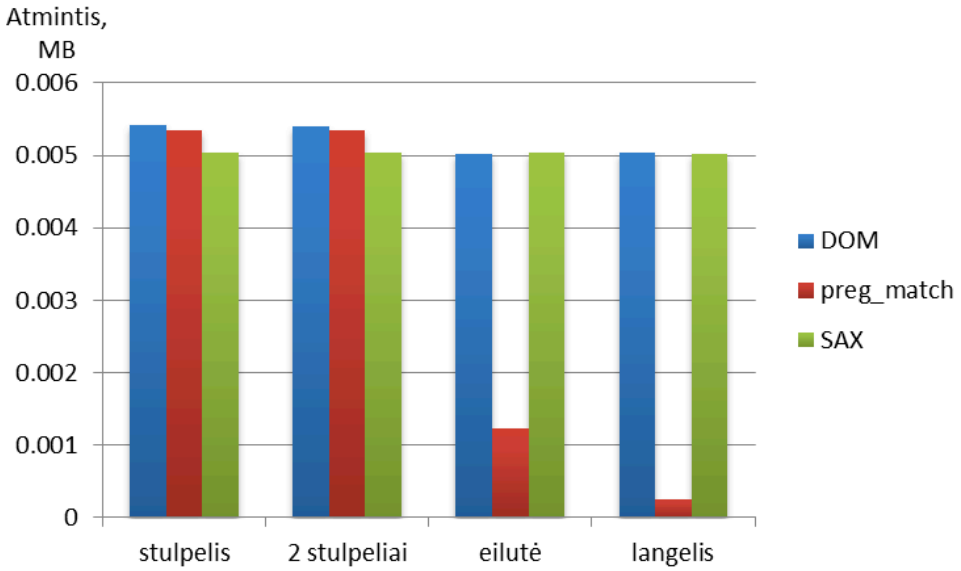
9 pav. *Atitinkantis standartą HTML, apdorojimo laikas*

Apdorojant neatitinkantį standarto HTML kodą, didžiausias atminties sunaudojimas yra DOM ir Regex (preg_match) metodų, tačiau Regex (preg_match) atminties sunaudojimas mažėja priklausomai nuo ieškomų duomenų kiekio, panašiai kaip ir apdorojimo laikas (*žr. 10 pav.*).



10 pav. *Neatitinkančio dtandarto HTML, atminties sunaudojimas*

Atitinkančio standartą HTML kodo apdorojimo metu, kai ieškomas didesnis duomenų kiekis (stulpelis arba 2 stulpeliai) atminties sunaudojimas tarp visų metodų yra labai panašus. SAX sunaudojamas atminties kiekis yra stabiliausias ir labai mažai kinta. DOM sunaudojamas atminties kiekis yra panašus kaip ir SAX, tačiau galima pastebėti didesnius svyravimus. Regex (preg_match) atminties sunaudojimas kaip ir ankstesniuose atvejuose priklausomo nuo ieškomo duomenų kiekio. Ieškant didesnio duomenų kiekio atminties sunaudojimas yra panašus kaip ir kitų metodų, tačiau ieškant mažo duomenų fragmento sunaudojamos atminties kiekis yra daug mažesnis (*žr. 11 pav.*).



11 pav. *Atitinkančio standartą HTML, atminties sunaudojimas*

Išanalizavus tyrimo rezultatus galima pasakyti, kad kiekvienas iš metodų turi tiek pliusų, tiek ir minusų. Todėl renkantis vieną iš šių metodų reikia atsižvelgti į tai, kokie duomenys bus išgaunami. Taip pat būtina atsižvelgti į tai koks yra duomenų šaltinis, ar iš šaltinio gaunamas duomenų failas yra pastovios struktūros ar jis yra atitinkantis standartą ir t.t.

Pagal atlikto tyrimo rezultatus kiekvienam iš metodų galima išskirti šias savybes:

DOM:

- Nereikalauja atitinkančio standarto HTML kodo.
- Paprastesnė užklausa norint išfiltruoti norimus duomenis negu su preg_match.
- Naudoja daugiau atminties negu preg_match.
- Lėtesnis už preg_match.

SAX:

- Paprastesnė užklausa norint išfiltruoti norimus duomenis negu su preg_match.
- Apdorojimo laikas artimas DOM jei dokumentas atitinka standartą.
- Reikalauja, kad analizuojamas kodas būtų atitinkantis standartą.
- Žymiai lėčiau analizuoja neatitinkantį standarto kodą.

preg_match:

- Apdoroja dokumentus greičiau už kitus metodus.
- Sunaudoja mažiau atminties negu kiti metodai.
- Reikalauja šablonų.
- Net smulkus kodo pakeitimai gali pareikalauti naujo paieškos šablono.

Atsižvelgiant į tyrimo rezultatus ir jo apibendrinimus, išskiriami tokie optimaliausi duomenų atrankos metodų pritaikymo scenarijai:

- Nekintančiose sistemose našiausi yra regex duomenų išgavimo metodai.
- DOM ir SAX yra lankstesni nei regex, todėl gali būti lengviau prisitaikantys prie HTML kodo pokyčių.
- SAX užtikrina mažesnį atminties naudojimą nei DOM, tačiau reikalauja atitinkančio standartą HTML kodo.
- DOM metodas nelaikomas našiausiu (lyginant su SAX ir regex), bet pasižymi našumo pastovumu ir pakankamai lengvu taikymu.

2. 4 Skyriaus išvados

Duomenų atrankos metodų tyrimui atlikti buvo pasirinkti trys, duomenų atrinkimo metodai, kurie pasižymi skirtingomis savybėmis – SAX, DOM, preg_match (naudojant regex šablonus). Atlikus našumo tyrimą, rezultatai buvo pateikti grafikuose. Išanalizavus tyrimo rezultatus galima teigti, kad preg_match yra našiausias iš metodu kai duomenų šaltinis yra pastovios struktūros, t.y. nekinta HTML kodo struktūra, o kinta tik duomenys. DOM ir SAX metodų našumas yra panašus, tačiau SAX reikalauja, kad duomenų šaltinio HTLM arba XML kodas būtų atitinkantis standartą, todėl tai lemia mažesnes SAX metodo taikymo galimybes. DOM ir SAX skiriasi ir tuo, kad abu metodai skirtingai interpretuoja analizuojamą dokumentą, DOM dokumentą analizuoja kaip medį, SAX dokumentą analizuoja pažingsniui, atlikdamas numatytus veiksmus kiekvieno žingsnio metu.

3. SIŪLOMO INTERNETINIO PUSLAPIO DUOMENŲ IŠRINKIMO ALGORITMO ARCHITEKTŪRA

3. 1 Dokumento paruošimas apdorojimui

Vienas iš pagrindinių algoritmui keliamų reikalavimų yra toks, kad jis tūri veikti su skirtingos kodo struktūros svetainėmis, kuriuos visos pateikia tuos pačius arba tokio pat tipo duomenis. Dėl šio reikalavimo iškyla keletas problemų realizuojant algoritmą:

- svetainės HTML kodas turi atitikti standartą;
- turi būti nustatyta duomenų pateikimo vieta puslapyje;
- būtina nustatyti duomenų išdėstymą (eilutėmis/stulpeliais) internetiniame puslapyje.

Svetainės HTML kodo atitikimas standartui yra svarbus dėl to, kad su svetainėje pateiktais duomenimis yra lengviau dirbti kai jie yra apdorojami XML pavidalų. Svetainės HTML kodo atitikimą standartui galima pabandyti užtikrinti naudojant kokį nors tam skirta įrankį, šiuo atveju naudojamas HTML Purifier PHP įrankis. HTML Purifier negali užtikrinti 100% kodo atitikimo standartui, tačiau jo dėka neatitinkančių standarto HTML dokumentų kiekis yra ženkliai sumažinamas. HTML Purifier nustatymai gali skirtis priklausomai nuo to kokie duomenys yra ieškomi, šiuo atveju iškarto yra šalinami tokie duomenų blokai kaip: head, script, style, select, input, img, radio, a(tačiau paliekamas nuorodos tekstas), pašalinami visi elementų atributai (id, class, inline CSS ir t.t.).

3. 2 Dokumento apdorojimas su SAX

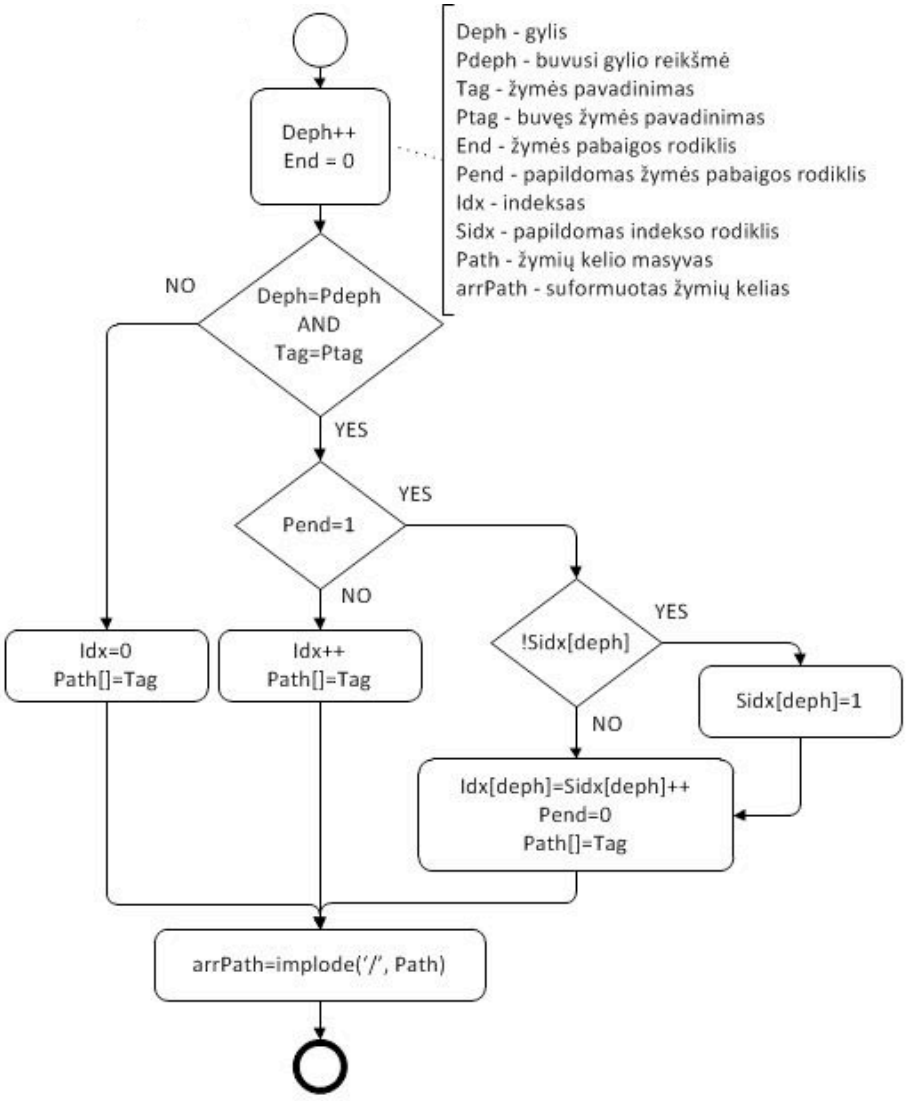
Po dokumento pakeitimo atitikti standartą apdorojimas tęsiamas naudojant SAX metodą. SAX buvo pasirinktas atlikus duomenų atrankos metodų tyrimą. Dirbant su atitinkančiu standartą HTML dokumentu, jo apdorojimo laiko vidurkis naudojant SAX buvo mažiausias, taip pat SAX atminties sunaudojimas yra stabilus, t.y. kinta nedaug nepriklausomai nuo to koks duomenų kiekis yra ieškomas, tai gali užtikrinti sistemos stabilumą dirbant su skirtingo tipo duomenimis.

Žymės atidarymo (*angl.* opening tag) metų atliekami veiksmai:

- Padidinamas žymės pozicijos gylis (index).
- Nustatomas kelias iki žymės (path).

Žymės atidarymo metų pirmiausiai bandomas nustatyti gylis kuriame yra atidaromas elementas. Gylis reikalingas tam, kad būtų galima teisingai suformuoti kelią iki žymės. Gyliui nustatyti taip pat naudojami duomenys kurie yra gaunami įvykdant funkciją žymės uždarymo metų. Nustačius gylį galima pradėti formuoti kelią iki žymės. Kelias reikalingas tam, kad jeigu žymės duomenyse bus rastas ieškomas raktažodis, jis bus panaudotas gauti duomenų bloką, kuriame buvo rastas raktažodis.

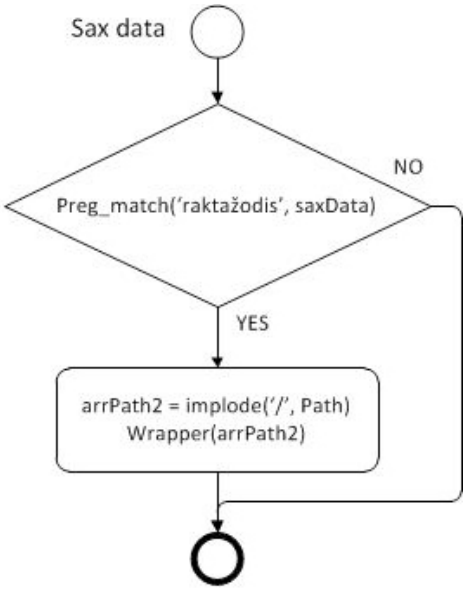
Tikslesnis žymės atidarymo metodo veikimas yra pateiktas 12 paveikslėlyje, kuriame pavaizduota kurie kintamieji yra keičiami tam tikrose situacijose ir kaip nustatomas ieškomų duomenų kelias.



12 pav. *Žymės atidarymo (opening tag) metų atliekami veiksmai*

Apdorojant žymėje pateikiamus duomenys atliekami šie veiksmai (*žr. 13 pav.*):

- Tikrinama ar duomenyse yra ieškomas raktažodis.
- Jei raktažodis rastas paleidžiama funkcija wrapper į kuria perduodamas ankščiau gautas kelias iki žymės(path).

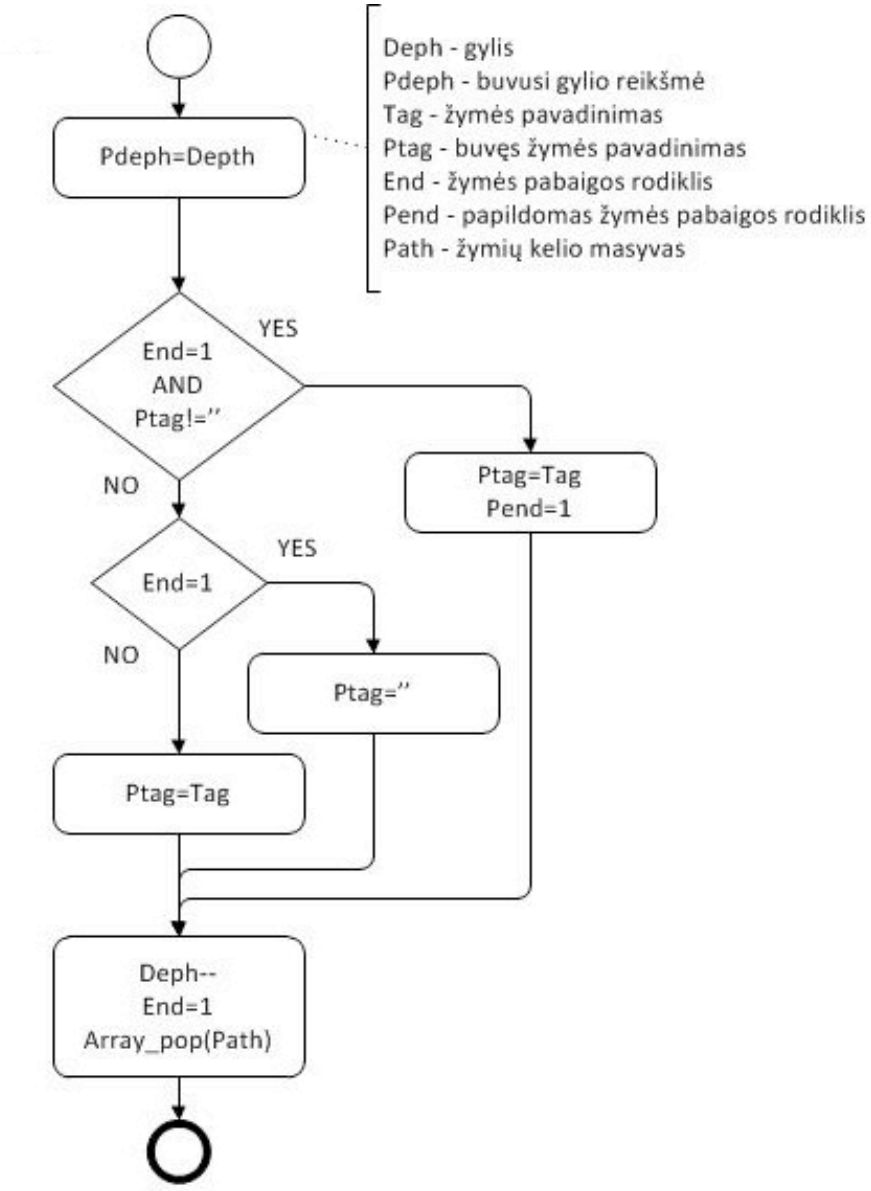


13 pav. *Apdorojant žymėje pateikiamus duomenys atliekami veiksmai*

Kai apdorojami žymėje esantys duomenys, tikrinama ar duomenyse yra ieškomas raktažodis. Raktažodis yra nustatomas kartu su duomenimis kurie vėliau bus naudojami identifikuoti pateikiamus duomenys bei jų išdėstymą. Jei raktažodis randamas, paleidžiama funkcija „wrapper“ į kuria paduodamas žymių kelias (path), kuris buvo sugeneruotas kai buvo vykdoma „sax start“ funkcija.

Žymės uždarymo (angl. opening tag) metų atliekami veiksmai (*žr. 14 pav.*):

- Pašalinama paskutinė žymė(tag) iš kelio (path).
- Sumažinamas žymės pozicijos gylis (index).



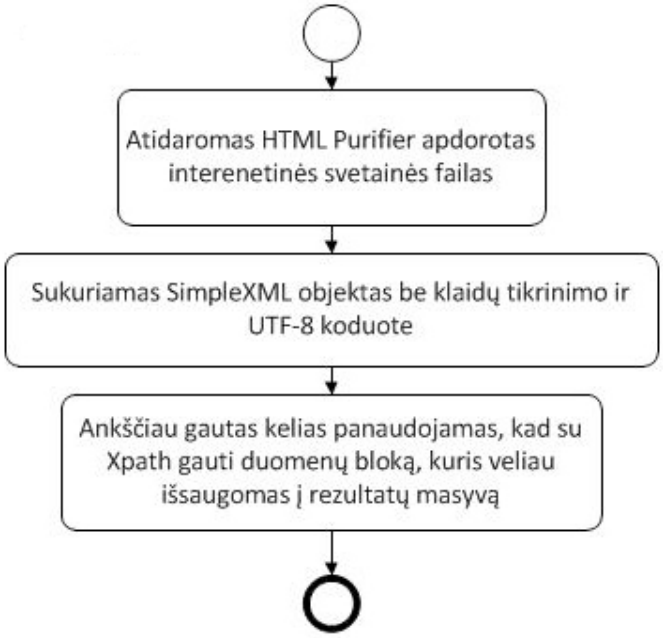
14 pav. *Žymės uždarymo metu atliekami veiksmai*

Žymės uždarymo metu atliekami veiksmai yra susiję su veiksmais atliekamais žymės atidarymo metų. Pagrindiniai veiksmai kurie yra atliekami žymes uždarymo metu yra gylio sumažinimas (deph), uždaromos žymės pašalinimas iš žymų kelio masyvo, bei žymės uždarymo vėliavėles (End) nustatymas.

3. 3 Pirminio duomenų išrinkimo funkcija wrapper

Funkcija wrapper yra skirta tam, kad sumažinti apdorojamų duomenų kiekį, kadangi kelias iki surasto raktažodžio yra žinomas, galima gauti visus duomenys kurie yra tame pačiame lygije kaip raktažodis, tai pagreitins sistemos darbą, bei sumažins sunaudojamos atminties kiekį.Veiksmai atliekami funkcijoje wrapper:

- Ankščiau gautą kelią (path) paduodame i XPATH funkciją, taip gaunamas duomenų blokas.
- Duomenų blokas išsaugomas į daugialypį masyvą.



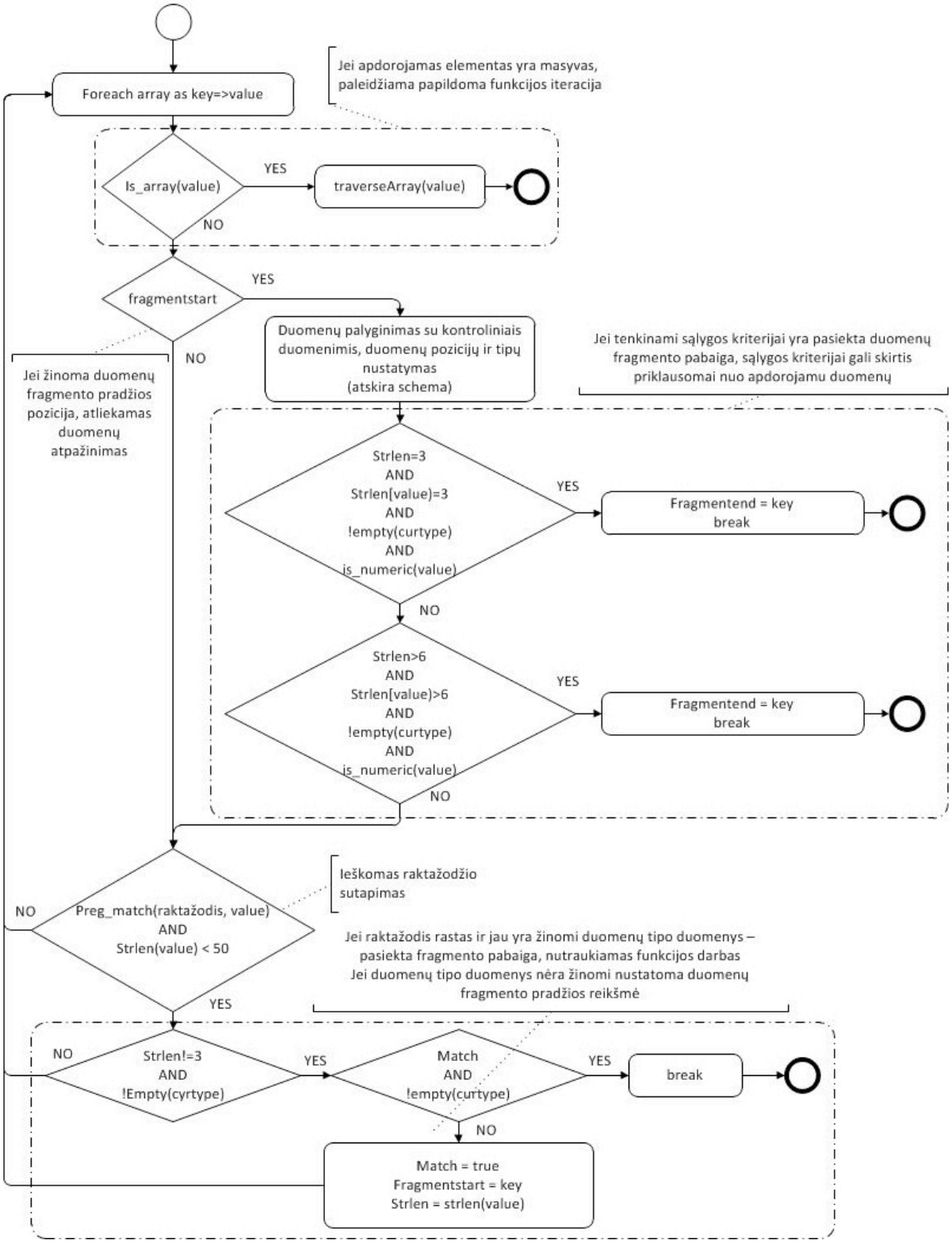
15 pav. *Veiksmai atliekami funkcijoje wrapper*

Kartais gali atsitikti taip, kad ieškomas raktažodis bus rastas keliose vietose, todėl rezultatų masyvas papildomai tikrinamas, kad iškarto atmesti nereikalingus duomenų blokus. Nereikalingi duomenų blokai saugomi rezultatų masyve nustatomi pagal jų dydį. Kiekvienas pirmo lygio masyvas rezultatų masyve yra duomenų blokas, kuriame buvo rastas raktažodis. Atlikus visų pirmo lygio masyvu didžių paliginimą, galima pašalinti nereikalingus arba besikartojančius duomenų blokus.

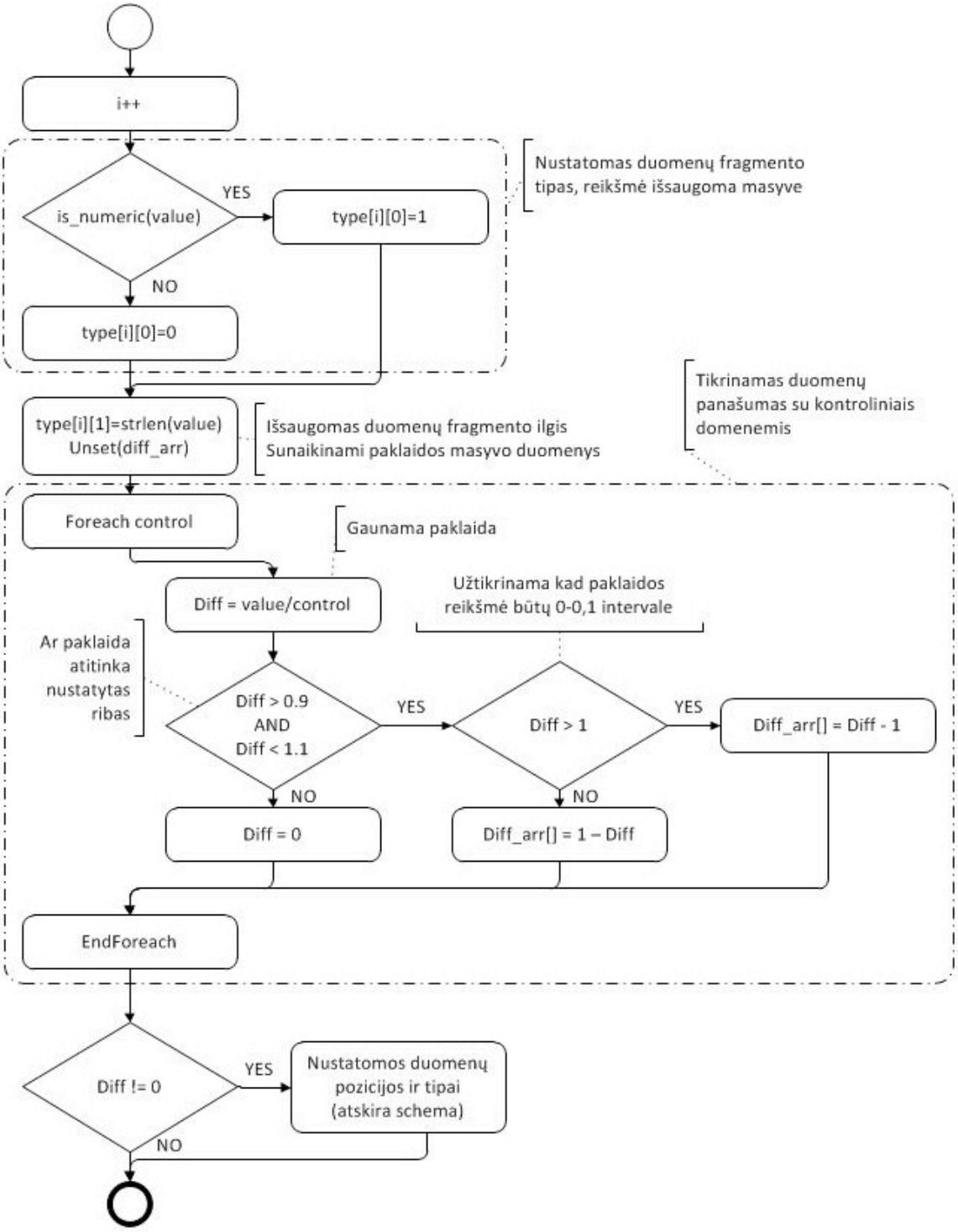
3. 4 Duomenų atpažinimo funkcija traverseArray

Šios funkcijos tikslas yra nustatyti žinomų duomenų pozicijas rezultatų masyve. Pirmiausia bandoma surasti raktažodžio sutapimą, šiuo atveju naudojamas raktažodis EUR. Kai raktažodis yra randamas jo pozicija rezultatų masyve yra išsaugoma kaip duomenų fragmento pradžia.

Ši reikšmė yra reikalinga, kad vėliau butu galima nustatyti bendrą duomenų pradžios poziciją, t.y. nuo kurios vietos rezultatų masyve prasideda mus dominantys duomenys.

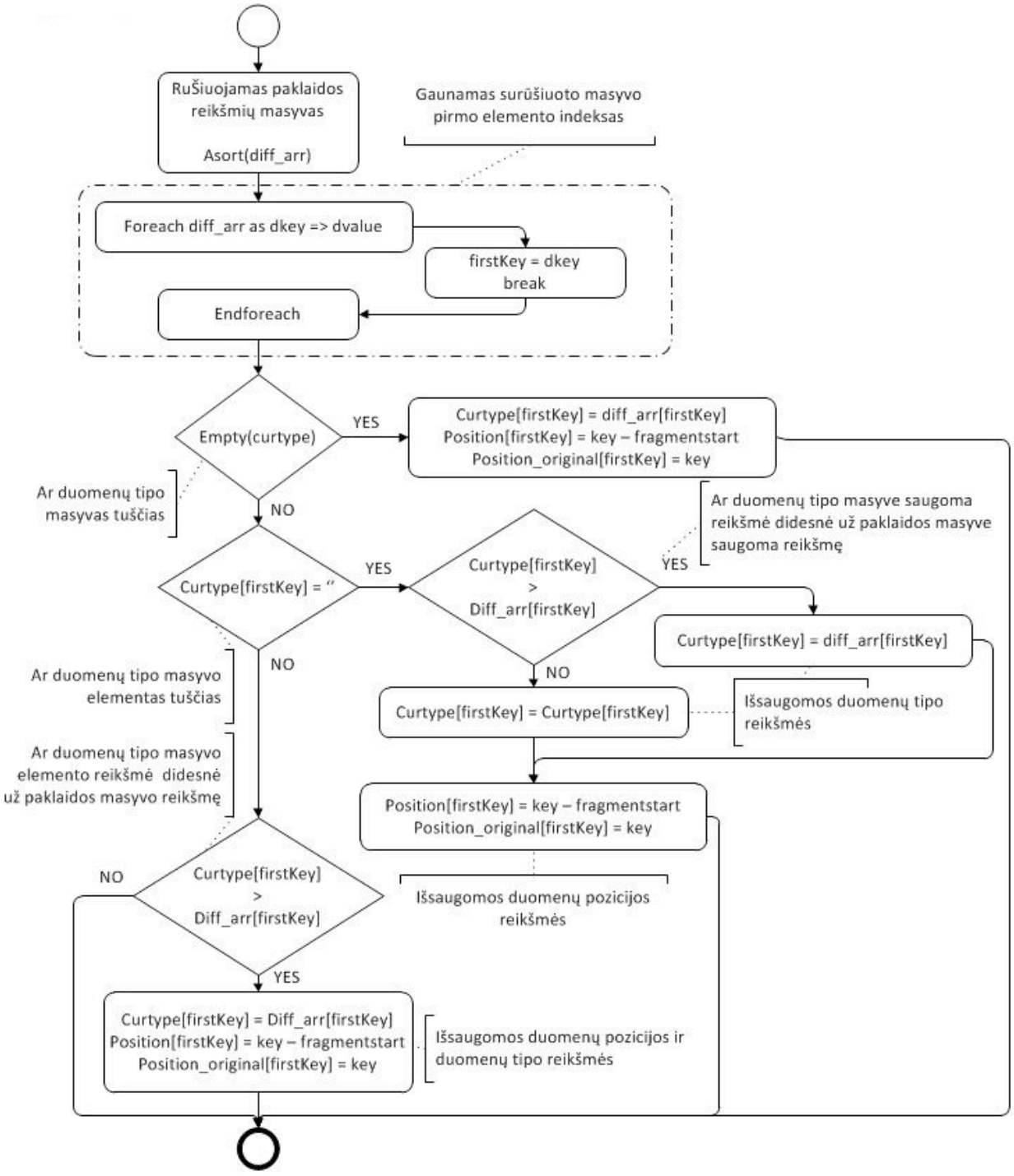


16 pav. *Duomenų atpažinimo funkcija traverseArray*



17 pav. *Funkcija traverseArray, duomenų palyginimas su kontroliniais duomenimis*

Kai duomenų pradžios pozicija yra žinoma, rezultatų masyvo duomenys pradėdami lyginti su iš anksto žinomu duomenų rinkiniu. Taip yra nustatomas duomenų išdėstymas, šiuo atveju nustatoma kurioje pozicijoje pateikiami skirtingi tos pačios valiutos kursai – pardavimas grynais, negrynais, bei pirkimas grynais ir negrynais. Kiekvienas rezultatų masyvo elementas, kuris atitinka iškeltus reikalavimus (šiuo atvejų elementas turi būti skaičiumi) yra lyginamas su kiekvienu elementu iš žinomo duomenų rinkinio, jeigu palyginimo rezultatas tenkina paklaidos kriterijus jis yra išsaugomas į palyginimų rezultatų masyvą.



18 pav. *Duomenų atpažinimo funkcija traverseArray, duomenų pozicijų nustatymas*

Kai gauti visi palyginimo duomenys, tikrinama kuris iš rezultatų turi mažiausia paklaida, tada išsaugoma šio elemento pozicija rezultatų masyve ir jam priskiriamas atitinkamas duomenų tipas (pardavimas grynais, negrynais, pirkimas grynais, negrynais). Paklaidos reikšmė yra išsaugoma tam, kad būtų galima palyginti ją su vėliau gautas rezultatais ir nustatyti ar nauja reikšmė yra tikslesnė. Po funkcijos įvykdymo yra žinomos pradinio duomenų rinkinio reikšmių pozicijos rezultatų masyve, tai leis išgauti visus ieškomus duomenys po to kai bus nustatytas duomenų išdėstymo būdas (eilutėmis, stulpeliais).

3. 5 Duomenų išdėstymo nustatymas

Duomenų išdėstymo atpažinimas atliekamas po to kai yra žinomos duomenų pozicijos po traverseArray funkcijos įvykdymo. Žinodami duomenų pozicijas galime lengvai nustatyti duomenų išdėstymą (ar duomenys pateikiami stulpeliais ar eilutėmis) paprasčiausiai sudėjus duomenų pozicijų indeksus.

```
<table>
  <tr>
    <td>eilute1</td>
    <td>eilute1</td>
    <td>eilute1</td>
  </tr>
  <tr>
    <td>eilute2</td>
    <td>eilute2</td>
    <td>eilute2</td>
  </tr>
  <tr>
    <td>eilute3</td>
    <td>eilute3</td>
    <td>eilute3</td>
  </tr>
</table>
```

```
<table>
  <tr>
    <td>stulpelis1</td>
    <td>stulpelis2</td>
    <td>stulpelis3</td>
  </tr>
  <tr>
    <td>stulpelis1</td>
    <td>stulpelis2</td>
    <td>stulpelis3</td>
  </tr>
  <tr>
    <td>stulpelis1</td>
    <td>stulpelis2</td>
    <td>stulpelis3</td>
  </tr>
</table>
```

eilute1	eilute1	eilute1
eilute2	eilute2	eilute2
eilute3	eilute3	eilute3

stulpelis1	stulpelis2	stulpelis3
stulpelis1	stulpelis2	stulpelis3
stulpelis1	stulpelis2	stulpelis3

19 pav. *Duomenų išdėstymo nustatymas*

Pavyzdžiui, jeigu duomenys yra išdėstyti eilutėmis, tada duomenų pozicijų suma bus lygi:3+4+5=12. Jeigu duomenys yra išdėstyti stulpeliais, tada duomenų pozicijų suma bus lygi:3+8+13=24.

Išanalizavus šiuos duomenis galima padaryti prielaidą, kad kai duomenys yra išdėstyti stulpeliais duomenų pozicijų suma gali būti artima arba didesnė už rezultatų masyvo dydį (šiuo atvejų rezultatų masyvo dydis būtų 17).

3. 6 Skyriaus išvados

Sukurtas algoritmas naudoja kelis duomenų atrankos metodus, kurie buvo ištirti šiame darbe – SAX ir preg_match. SAX naudojamas pradiniam dokumento apdorojimui, tam kad pažingsniui patikrinti kiekvieną duomenų bloką esanti apdorojamame dokumente. Duomenų bloko apdorojimo metu naudojamas preg_match metodas, tam kad nustatyti ar duomenyse yra ieškomas raktažodis.

4. PASIŪLYTO INTERNETINIO PUSLAPIO DUOMENŲ IŠRINKIMO ALGORITMO TYRIMAS

4. 1 Duomenų išrinkimo algoritmų palyginimo kriterijai

Atliekant duomenų išrinkimo algoritmų palyginimą dažniausiais išskiriamos kelios, visiems algoritmams pritaikomos charakteristikos. Tos pačios charakteristikos naudojamos ir kuriant ar testuojant duomenų išrinkimo algoritmus.

Kadangi duomenų gavybos algoritmų paskirtis yra duomenų išgavimas, labai svarbu, kad gaunami duomenys būtų būtent tie, kurių ieškoma. Tam norint nustatyti gaunamų duomenų tikslumą, naudojami keli rodikliai:

- Bendras duomenų fragmentų skaičius.
- Bendras išrinktų duomenų fragmentų skaičius.
- Teisingai išrinktų duomenų fragmentų skaičius (True-Positive).
- Neteisingai išrinktų duomenų fragmentu skaičius (False-Positive), taip pat prie šio rodiklio priskiriami tie duomenų fragmentai kurie turėjo būti išrinkti algoritmo, bet dėl kažkokių priežasčių pasirinkti nebuvo (False-Negative).

Žinant aukščiau aprašytus duomenis galima apskaičiuoti tokius rodiklius kaip ieškomų duomenų dalis tarp visų atrinktų duomenų – tikslumas (*angl.* Precision) ir atrinktų ieškomų duomenų dalis tarp visų ieškomų duomenų – atrinkimas (*angl.* Recall):

Tikslumas (*angl.* Precision) – santykis tarp teisingai išrinktų duomenų fragmentų (True-Positive) ir bendro išrinktų duomenų fragmentų skaičiaus. Šis rodiklis parodo sistemos resursų naudojimo efektyvumą (žr. 1 formulę).

$$\text{Precision} = \frac{tp}{tp + fp} \tag{1}$$

kur *tp* – (angl. true-positive) atrinktų ieškomų fragmentų skaičius; *fp* – (angl. false-positive) klaidingai atrinktų fragmentų skaičius.

Atrinkimas (angl. Recall) – santykis tarp teisingai išrinktų duomenų fragmentų (True-Positive) ir bendro ieškomų duomenų fragmentų skaičiaus. Šis rodiklis parodo algoritmo naudojamų taisyklių duomenų atpažinimui efektyvumą (žr. 2 formulę).

$$\text{Recall} = \frac{tp}{tp + fn} \tag{2}$$

kur *tp* – (angl. true-positive) atrinktų ieškomų fragmentų skaičius; *fn* – (angl. false-negative) ieškomų fragmentų, kurie nebuvo atrinkti skaičius.

Pavyzdžiui, jei duomenų masyve yra 9 ieškomi fragmentai, sistema iš duomenų masyvo išrenka 7 fragmentus, tačiau tik 4 iš jų yra teisingi (True-Positive), likusieji 3 yra klaidingi (False-Positive). Šiuo atveju tikslumas (Precision) yra 4/7, o atrinkimas (Recall) yra 4/9.

Apibendrinus galima pasakyti jog algoritmai su aukštu tikslumo atrenka daugiau teisingų duomenų fragmentų negu klaidingų, o algoritmai su aukštu atrinkimu atrenka didžiąją dalį ieškomų duomenų fragmentų.

F-rodiklis (F-score/F-measure) – apibendrina tikslumą ir atrinkimą, kas leidžia išreikšti algoritmo našumą vienu skaičiumi (žr. 3 formulę).

Šis rodiklis gali būti naudojamas kaip duomenų išrinkimo algoritmų apisendinamasis rodiklis.

4. 2 Duomenų atrankos įrankių lyginamoji analizė

Atliekant duomenų atrankos įrankių lyginamąją analizę analizuojami duomenų išrinkimo rezultatai gauti su trimis komerciniais įrankiais (jų kainų palyginimas pateiktas 1 lentelėje) ir sukurtu pasiūlyto algoritmo prototipu. Nors šiuo metu egzistuoja nemažai įrankių ar duomenų išrinkimo iš internetinių puslapių algoritmų [16, 17, 19, 22, 25–29], tačiau ne visus įrankius, ar aprašomus algoritmus įmanoma ištirti praktiškai dėl jų neatskleidžiamo kodo ar veikimo detalių.

1 lentelė. Duomenų išrinkimo įrankių kaina

	Visual Web Ripper	Helium Scraper	OutWit Hub
Kaina	$299	$99	$59

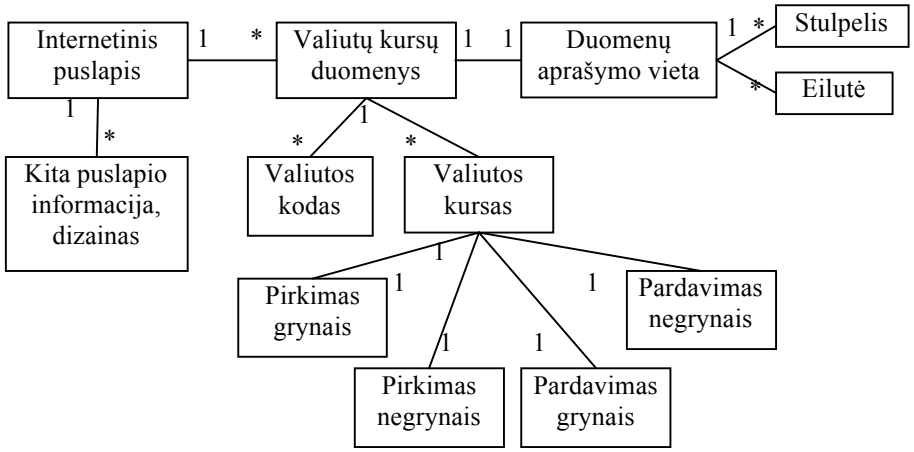
Šio tyrimo tikslas yra palyginti algoritmų lankstumą, t.y. priklausomybę nuo duomenų failo struktūros bei duomenų išdėstymo pakeitimo.

Lyginimui pasirinktos sistemos ir pagal sukurtą duomenų išrinkimo algoritką realizuotas prototipas tiriami siekiant nustatyti jų tikslumo ir atrenkamumo rodiklius esant skirtingiems žiniatinklio duomenų rinkiniams ar sistemos nustatymams.

Šiame tyrime naudojami du pagrindiniai duomenų rinkiniai, kurie testavimo metu bus vadinami „pirma svetainė“ ir „antra svetainė“:

- Pirma svetainė – DNB Banko valiutų kursų puslapio HTML kodas, kuriame aprašyti valiutų kursai yra išdėstyti eilutėmis.
- Antra svetainė – modifikuotas DNB Banko valiutų kursų puslapio HTML kodas, kuriame aprašyti valiutų kursai yra išdėstyti stulpeliais.

Naudojamuose duomenų rinkiniuose HTML kode tarp papildomo dizainui ir kitai informacijai aprašyti reikalingo kodo, pateikiama informacija apie valiutų kursus. Juose bent kartą yra pateikiamas euro kursas, o kiekvienai valiutai nurodomas valiutos kodas ir jo kursas skirtingais atvejais (pirkimas grynais, pirkimas negrynais, aprdavimas grynais, pardavimas negrynais) (*žr. 20 pav.*).



20 pav. *Valiutų kursų internetinio puslapio komponentų struktūra*

Tyrimo metu bandomi išgauti visi valiutų kursų duomenys tame puslapyje, nurodant koks yra kiekvienos valiutos kodas ir kokie yra jų kursai. Visa kita informacija yra ignoruojama.

Šio tyrimo metu testuojamos tokios situacijos, kurios įvertina kaip duomenų išrinkimo tikslumas priklauso nuo to, kokie duomenys ir kur HTML kode keičiasi:

1. Pradinis duomenų išrinkimas (eilutėmis) – sistemai pateikiamas DNB Banko valiutų kursų puslapio HTML kodas, kuriame aprašyti valiutų kursai yra išdėstyti eilutėmis (*žr.* 9 lentelę 1 priede).
2. Pradinis duomenų išrinkimas (stulpeliais) – sistemai pateikiamas modifikuotas DNB Banko valiutų kursų puslapio HTML kodas, kuriame aprašyti valiutų kursai yra išdėstyti stulpeliais (*žr.* 10 lentelę 1 priede).
3. Tikrinamas duomenų surinkimo algoritmų lankstumas, naudojamas antros svetaines nekoreguotas failas, tačiau duomenų išrinkimo nustatymai tokie pat kaip išrenkant duomenys iš pirmo failo (*žr.* 11 lentelę 1 priede).
4. Tikrinamas duomenų surinkimo algoritmų lankstumas, naudojamas pirmos svetaines nekoreguotas failas, tačiau duomenų išrinkimo nustatymai tokie pat kaip išrenkant duomenys iš antro failo (*žr.* 12 lentelę 1 priede).
5. Tikrinamas duomenų surinkimo algoritmų lankstumas įnešus pakeitimų į pirmos svetainės HTML kodą - sukuriamas 0-lygio div elementas (*žr.* 13 lentelę 1 priede).
6. Tikrinamas duomenų surinkimo algoritmų lankstumas įnešus pakeitimų į antros svetainės HTML kodą - sukuriamas 0-lygio div elementas (*žr.* 14 lentelę 1 priede).
7. Tikrinamas duomenų surinkimo algoritmų lankstumas įnešus pakeitimų į pirmos svetainės HTML kodą - sukuriamas n-lygio div elementas (*žr.* 15 lentelę 1 priede).
8. Tikrinamas duomenų surinkimo algoritmų lankstumas įnešus pakeitimų į antros svetainės HTML kodą - sukuriamas n-lygio div elementas (*žr.* 16 lentelę 1 priede).
9. Tikrinamas duomenų surinkimo algoritmų lankstumas įnešus pakeitimų į pirmos svetainės HTML kodą - sukuriamas 0-lygio table elementas (3 eilutės ir 3 stulpeliai) (*žr.* 17 lentelę 1 priede).
10. Tikrinamas duomenų surinkimo algoritmų lankstumas įnešus pakeitimų į antros svetainės HTML kodą - sukuriamas 0-lygio table elementas (3 eilutės ir 3 stulpeliai) (*žr.* 18 lentelę 1 priede).
11. Tikrinamas duomenų surinkimo algoritmų lankstumas įnešus pakeitimų į pirmos svetainės HTML kodą - sukuriamas n-lygio table elementas (3 eilutės ir 3 stulpeliai) (*žr.* 19 lentelę 1 priede).
12. Tikrinamas duomenų surinkimo algoritmų lankstumas įnešus pakeitimų į antros svetainės HTML kodą - sukuriamas n-lygio table elementas (3 eilutės ir 3 stulpeliai) (*žr.* 20 lentelę 1 priede).
13. Tikrinamas duomenų surinkimo algoritmų lankstumas įnešus pakeitimų į pirmos svetainės HTML kodą (pašalinama viena iš lentelės eilučių) (*žr.* 21 lentelę 1 priede).
14. Tikrinamas duomenų surinkimo algoritmų lankstumas įnešus pakeitimų į antros svetainės HTML kodą (pašalinamas vienas iš lentelės stulpelių) (*žr.* 22 lentelę 1 priede).

15. Tikrinamas duomenų surinkimo algoritmų lankstumas įnešus pakeitimų į pirmos svetainės HTML kodą (pašalinama EUR eilutė) (*žr.* 23 lentelę 1 priede).
16. Tikrinamas duomenų surinkimo algoritmų lankstumas įnešus pakeitimų į antros svetainės HTML kodą (pašalinamas EUR stulpelis) (*žr.* 24 lentelę 1 priede).

Šio tyrimo metu gauti rezultatai ir tikslesni parametrai pateikti 9–16 lentelėse 1 priede, o juos apibendrinančių grafikų ir diagramų skaitomumui padidinti, algoritmų žymėjimui naudojami tokie sutrumpinimai:

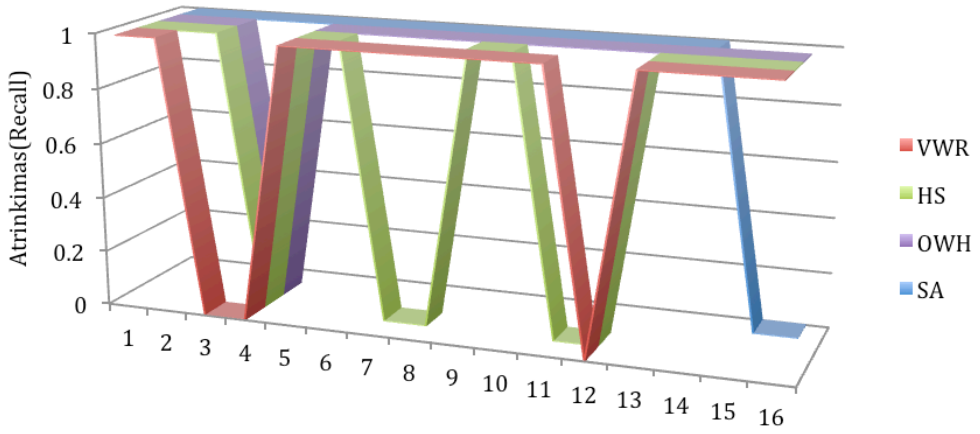
VWR – Visual Web Ripper.

HS – Helium Scraper.

OWH – OutWit Hub.

SA – siūlomo algoritmo modelis.

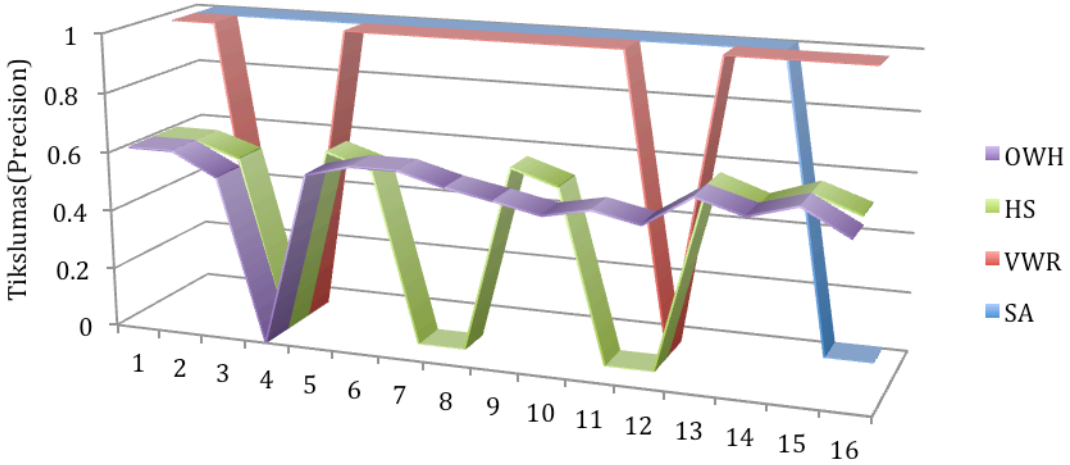
Diagramoje (*žr. 21 pav.*) atvaizduojamas keturių testuojamų įrankių atrankos (*angl.* Recall) rodiklis.



21 pav. *Atrinkimo (angl. Recall) rodiklio kaita, priklausomai nuo testuojamo įrankio ir tyrimo testinės situacijos*

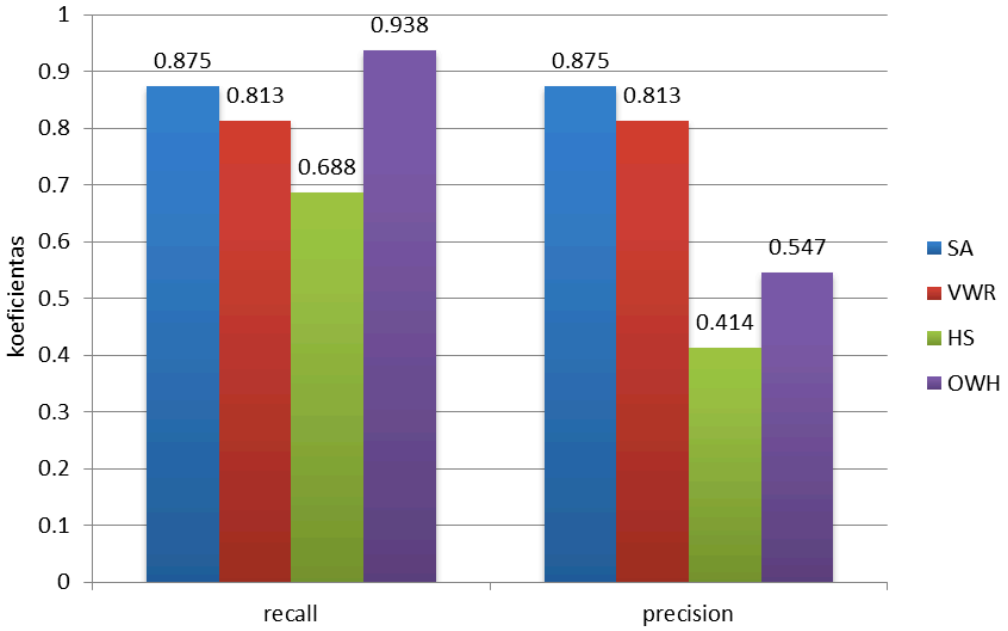
Jis parodo kiek ieškomų duomenų buvo išrinkta kiekvieno testo metu. X ašyje vaizduojami testavimo atvejai, Y ašyje vaizduojamas atrankos rodiklis, kurį galima paversti procentais padauginus gautą reikšmę iš 100. 21 paveikslėlyje matyti, kad visi algoritmai arba išrinkdavo visus ieškomus duomenys arba neišrinkdavo išvis.

Diagramoje (*žr. 22 pav.*) atvaizduojamas keturių testuojamų įrankių tikslumo rodiklis.



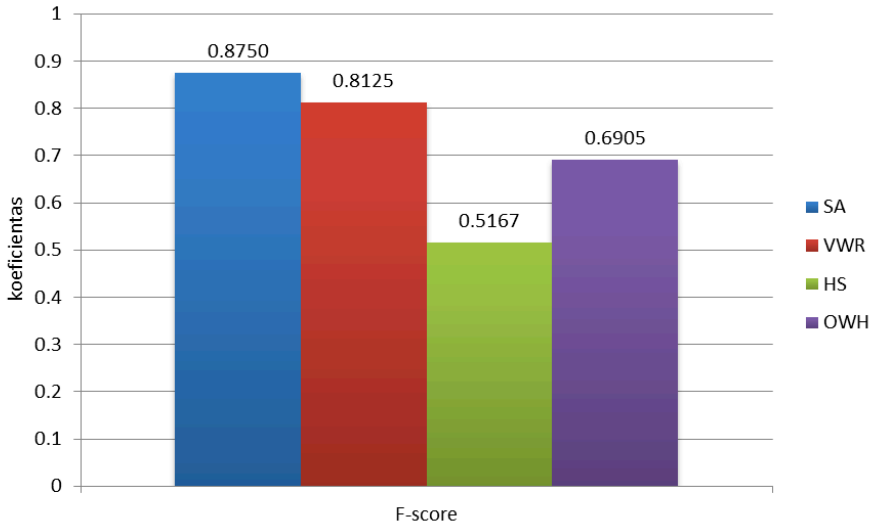
22 pav. *Tikslumo (angl. Precision) rodiklio kaita, priklausomai nuo testuojamo įrankio ir tyrimo testinės situacijos*

Tikslumo rodiklis parodo santykį tarp išrinktų duomenų ir išrinktų ieškomų duomenų, t.y. santykis tarp išrinktų true positive duomenų ir true positive bei false positive duomenų sumos. X ašyje vaizduojami testavimo atvėjai, Y ašyje vaizduojamas tikslumo rodiklis, kuri galima paversti procentais padauginus iš 100. Galima teigti jog Visual Web Ripper ir sukurtas algoritmas ženkliai lenkia kitus du duomenų išrinkimo įrankius (OutWit Hub ir Helium Scraper), pagal šį rodikli galima spręsti apie tai kiek ilgai bus apdorojami duomenys, kadangi žemas tikslumo rodiklis parodo jog algoritmui ieškant norimus išrinkti duomenys teks analizuoti didesni duomenų kieki, negu algoritmams su aukštu tikslumo rodikliu.



23 pav. *Atrinkimo (angl. Recall) ir tikslumo (angl. Precision) rodiklių vidurkiai*

Tam kad būtų paprasčiau analizuoti gautus duomenys, buvo paskaičiuoti visų 16 testinių atvejų atrinkimo ir tikslumo rodiklių vidurkiai. OutWit Hub turi didžiausia atrinkimo reikšmę, jis nežymiai lenkia Visual Web Ripper ir sukurto algoritmo pasiektą rezultatą, tačiau OutWit Hub tikslumo rodiklis yra daug mažesnis už Visual Web Ripper ir sukurto algoritmo. Helium Scraper parodė prasčiausia rezultata iš visų, tiek atrinkimo tiek tikslumo rodikliai yra patys žemiausi.



24 pav. *F-score rodiklis*

Kad būtų paprasčiau įvertinti algoritmo naudingumą, naudojamas papildomas F-score rodiklis, jis apibendrina atrinkimo ir tikslumo duomenys. Sukurtas algoritmas pagal atliktus testus turi aukščiausia F-score, nedaug atsilieka Visual Web Ripper ir OutWit Hub, pagrindine priežastis dėl ko OutWit Hub like trečioje vietoje yra ta, kad šio algoritmo tikslumo rodiklis yra daug žemesnis už sukurto algoritmo ir Visual Web Ripper. Helium Scraper pasirodė prasčiausiai iš visų algoritmu, tačiau tai buvo matoma iškarto kai buvo gautos atrinkimo ir tikslumo reikšmės.

4. 3 Sukurto algoritmo nustatymų universalumo testavimas

Šio testo tikslas yra nustatyti ar algoritme naudojamų nustatymų rinkinys leidžia išgauti duomenys iš skirtingų interneto svetainių nekeičiant nustatymų.

Tam buvo pasirinkti Lietuvoje ir Latvijoje veikiantys bankai ir jų teikiami duomenys apie valiutų kursus. Iš viso tyrimui naudojama 10 žiniatinklio kodų, kur 2 iš jų buvo modifikuoti siekiant padidinti duomenų pateikimo variantų kiekį, bei kardinaliai pakeisti duomenų išdėstymo HTML kodo struktūrą.

Taip pat siekiant įvertinti pasiūlyto algoritmo panaudojimo specifiką, kiekvienam valiutų kursus pateikiančiam žiniatinkliui susisteminta informacija apie jų HTML kodo specifiką, duomenų kiekius ir pagal tai įvertinama koks yra šio algoritmo tikslumo, atrinkimo ir F-score (*žr.* 2–4 lenteles).

2 lentelė. Testuojamas algoritmo nustatymų universalumas

Algoritmo pradiniai duomenys	Raktažodis: eur Kontroliniai duomenys: 3.44200, 3.46300, 3.44450, 3.46100			
Svetaines pavadinimas	Swedbank	SEB (modifikuotas)	DNB	DNB (modifikuotas)
Duomenų pateikimo būdas (table/div/list)	table	div	table	table
Duomenų išdėstymas (eilutės, stulpeliai)	eilutės	eilutės	eilutės	stulpeliai
Bendras duomenų fragmentų skaičius	471	320	586	575
Bendras ieškomų duomenų fragmentų skaičius	95	135	80	80
Išrinktų duomenų fragmentų skaičius	95	135	80	80
Išrinkta ieškomų duomenų fragmentų	95	135	80	80
Atrinkimas (Recall)	1	1	1	1
Tikslumas (Precision)	1	1	1	1
F-score	1	1	1	1

3 lentelė. Testuojamas algoritmo nustatymų universalumas

Algoritmo pradiniai duomenys	Raktažodis: eur Kontroliniai duomenys: 3.44200, 3.46300, 3.44450, 3.46100			
Svetaines pavadinimas	Valiutos.lt	SEB	Siauliu Bankas	Medbank
Duomenų pateikimo būdas (table/div/list)	table *	table	table **	table ***
Duomenų išdėstymas (eilutės, stulpeliai)	eilutės	eilutės	eilutės	stulpeliai
Bendras duomenų fragmentų skaičius	1159	732	925	520

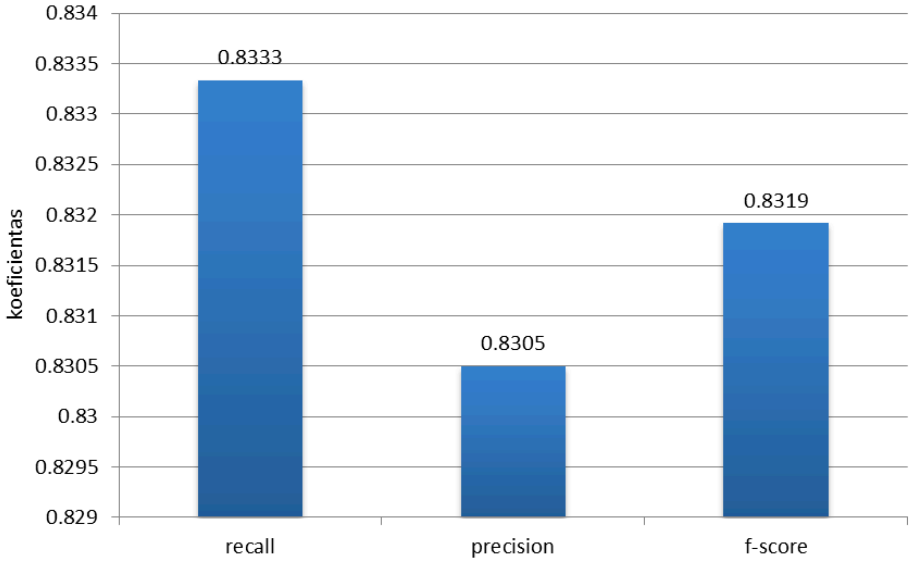
Bendras ieškomų duomenų fragmentų skaičius	140	150	81	65
Išrinktų duomenų fragmentų skaičius	145	150	0	65
Išrinkta ieškomų duomenų fragmentų	140	150	0	65
Atrinkimas (Recall)	1	1	0	1
Tikslumas (Precision)	0,966	1	0	1
F-score	0,983	1	0	1

* valiutos kodas <a> žymėje
** Duomenys yra pateikiami per 2 lenteles ir skirtingu formatu, naudojami nustatymai netinka duomenų atpažinimui nors ir išrenkami visi reikalingi duomenys
*** puslapyje yra 2 lentelės su duomenimis, viena iš jų teisingai atpažįstama kaip false positive

4 lentelė. Testuojamas algoritmo nustatymų universalumas

Algoritmo pradiniai duomenys	Raktažodis: eur Kontroliniai duomenys: 0.694000, 0.707500			
Svetaines pavadinimas	MoneyExpress.lv	Siauliu Bankas	naudasnams.lv	tavex.lv
Duomenų pateikimo būdas (table/div/list)	table *	table **	table ***	table ****
Duomenų išdėstymas (eilutės, stulpeliai)	eilutės	eilutės	eilutės	eilutės
Bendras duomenų fragmentų skaičius	383	925	995	551
Bendras ieškomų duomenų fragmentų skaičius	42	81	99	123
Išrinktų duomenų fragmentų skaičius	42	81	0	123
Išrinkta ieškomų duomenų fragmentų	42	81	0	123
Atrinkimas (Recall)	1	1	0	1
Tikslumas (Precision)	1	1	0	1
F-score	1	1	0	1

* duomenys pateikiami 2-jose lentelėse
** vienoje iš lentelių pateikiamas EUR/LVL santykis
*** valiutos kodas žymėje valiutos kursas žymėje
**** valiutos kodas žymėje



25 pav. *Nustatymų testavimo metu gautos rodiklių reikšmės*

Testavimo metu gauti rodikliai yra labai artimi vienas kitam tai reiškia, kad algoritmas veikia efektyviai. Maksimalus išrenkamų duomenų skaičius sudaro būtent ieškomus duomenų fragmentus, todėl algoritmas veltui negaišta laiko išfiltruodamas nereikalingus duomenys iš rezultatų masyvo. Gali pasirodyti, kad gautos rodikliu reikšmės yra žemos, tačiau taip yra todėl, kad testavimo metu buvo naudojamos svetainės, kuriuos savo struktūra ir duomenų pateikimo būdų ženkliai skyrėsi viena nuo kitos. Šio tyrimo metu gautos rodiklių reikšmės yra kiek žemesnės negu reikšmės kurios buvo gautos atliekant algoritmų palyginimą, tačiau skirtumas nėra didelis ir tai dar kartą parodo, kad algoritmas gali būti naudojamas norint išgauti duomenys iš skirtingų svetainių naudojant tuos pačius nustatymus.

4. 4 Pasiūlyto algoritmo naudojamos panašių duomenų atrankos paklaidos nustatymas

Sukurtas algoritmas remiasi sistemai žinomais duomenimis tam, kad atlikti surinktų duomenų atpažinimą. Kadangi algoritmas buvo testuojamas su valiutų kursų duomenimis, kuriuos pateikia bankai, reikėjo pasirinkti vieną iš valiutų, kuria prekiautų visi Lietuvos bankai, tačiau tai ne vienintelis kriterijus kuri turi atitikti pasirinkta valiuta. Tam kad užtikrinti tikslesni duomenų atpažinimą reikia kad kontroliniai duomenys pateikiami svetainėje kuo mažiau kistų tarp skirtingu duomenų šaltinių, todėl buvo atliktas tyrimas, tam kad nustatyti EUR ir USD valiutų kursų paklaidos ribas lyginant kontrolinius duomenys ir duomenys pateikiamus bankų internetinėse svetainėse (žr. 5–8 lenteles).

5 lentelė. EUR kurso paklaida lyginant su Swedbank pateikiamu kursu

Swedbank pateikiamas EUR kursas	EUR	Grynais		Negrynais	
		Perka	Parduoda	Perka	Parduoda
		3.44200	3.46300	3.44450	3.46100
SEB	Kursas	3,4420	3,4649	3,4442	3,4614
	Paklaida %	0	0.054866	0.00871	0.011557
DNB	Kursas	3,4370	3,4640	3,4443	3,4613
	Paklaida %	0.145264	0.028877	0.005806	0.008668
Siauliu Bankas	Kursas	3,4380	3,4610	3,4442	3,4614
	Paklaida %	0.116212	0.057753	0.00871	0.011557
Nordea	Kursas	3,4415	3,4641	3,4442	3,4614
	Paklaida %	0.014526	0.031764	0.00871	0.011557
Danske	Kursas	3,4400	3,4620	3,4442	3,4614
	Paklaida %	0.058106	0.028877	0.00871	0.011557
Medbank	Kursas	3,4420	3,4630	3,4442	3,4614
	Paklaida %	0	0	0.00871	0.011557
Paklaidos vidurkiai %		0.055685	0.033689	0.008226	0.011557
Bendras paklaidos vidurkis %		0.025079			
Maksimali paklaida %		0.145264	0.057753	0.00871	0.011557

5 lentelėje pateikiamos EUR kurso svyravimo paklaidos lyginant atitinkamo banko pateikiamus valiutų kursus su EUR kursu pateikiamu Swedbank internetinėje svetainėje. Iš rezultatų matosi kad paklaidos yra labai mažos, maksimali paklaidos reikšmė yra 0,15%, o vidutinė tik 0,025%.

6 lentelė. EUR kurso paklaida lyginant su oficialiu Lietuvos banko kursu

Lietuvos Banko oficialus EUR kursas	EUR	3.4528			
SEB	Kursas	3,4420	3,4649	3,4442	3,4614
	Paklaida %	0.31279	0.35044	0.249073	0.249073
Swedbank	Kursas	3,44200	3,46300	3,44450	3,46100
	Paklaida %	0.31279	0.295412	0.240385	0.237488

DNB	Kursas	3,4370	3,4640	3,4443	3,4613
	Paklaida %	0.4576	0.324374	0.246177	0.246177
Siauliu Bankas	Kursas	3,4380	3,4610	3,4442	3,4614
	Paklaida %	0.428638	0.237488	0.249073	0.249073
Nordea	Kursas	3,4415	3,4641	3,4442	3,4614
	Paklaida %	0.327271	0.327271	0.249073	0.249073
Danske	Kursas	3,4400	3,4620	3,4442	3,4614
	Paklaida %	0.370714	0.26645	0.249073	0.249073
Medbank	Kursas	3,4420	3,4630	3,4442	3,4614
	Paklaida %	0.31279	0.295412	0.249073	0.249073
Paklaidos vidurkiai %		0.36037	0.29955	0.247418	0.247005
Bendras paklaidos vidurkis %		0.269347			
Maksimali paklaida %		0.4576	0.35044	0.249073	0.249073

Lentelėje 6, bankuose pateikiami EUR kursai yra lyginami su oficialiu Lietuvos banko EUR kursu, šis palyginimas reikalingas tam, kad nustatyti ar yra tikimybė, kad Lietuvos banko EUR kursas gali būti klaidingai atpažintas kaip vienas iš komercinio banko pateikiamų EUR kursu, t.y. daugelis banku pateikia ne tik savo EUR kursus, bet taip pat ir Lietuvos banko nustatyta oficialų EUR kursą.

7 lentelė. USD kurso paklaida lyginant su Swedbank pateikiamu kursu

Swedbank pateikiamas USD kursas	USD	Grynais		Negrynais	
		Perka	Parduoda	Perka	Parduoda
		2.6329	2.6974	2.6382	2.6907
SEB	Kursas	2.6057	2.6853	2.611	2.68
	Paklaida %	1.033081	0.44858	1.031006	0.397666
DNB	Kursas	2.623	2.668	2.6201	2.6671
	Paklaida %	0.376011	1.089938	0.686074	0.877095
Siauliu Bankas	Kursas	2.616	2.676	2.6184	2.6729
	Paklaida %	0.641878	0.793357	0.750512	0.661538
Nordea	Kursas	2.5721	2.7209	2.58	2.713
	Paklaida %	2.309241	0.871209	2.20605	0.828781
Danske	Kursas	2.604	2.684	2.6191	2.6667
	Paklaida %	1.097649	0.496775	0.723978	0.891961
Medbank	Kursas	2.6231	2.6601	2.6263	2.6624
	Paklaida %	0.372213	1.382813	0.451065	1.051771
Paklaidos vidurkiai %		0.952779	0.845788	0.955197	0.792146
Bendras paklaidos vidurkis %		0.827379			
Maksimali paklaida %		2.309241	1.382813	2.20605	1.051771

Lentelės 7 ir 8 yra analogiškos lentelėms 5 ir 6, skirtumas tik tas, jog jose nurodytos USD kurso svyravimo paklaidos.

8 lentelė. USD kurso paklaida lyginant su oficialiu Lietuvos banko kursu

Lietuvos Banko oficialus USD kursas	USD	2.6203			
SEB	Kursas	2.6057	2.6853	2.611	2.68
	Paklaida %	0.557188	2.480632	0.354921	2.278365
Swedbank	Kursas	2.6108	2.6748	2.6161	2.6682
	Paklaida %	0.362554	2.079915	0.160287	1.828035
DNB	Kursas	2.623	2.668	2.6201	2.6671
	Paklaida %	0.103042	1.820402	0.007633	1.786055
Siauliu Bankas	Kursas	2.616	2.676	2.6184	2.6729
	Paklaida %	0.164103	2.125711	0.072511	2.007404

Nordea	Kursas	2.5721	2.7209	2.58	2.713
	Paklaida %	1.839484	3.839255	1.537992	3.537763
Danske	Kursas	2.604	2.684	2.6191	2.6667
	Paklaida %	0.622066	2.431019	0.045796	1.77079
Medbank	Kursas	2.6231	2.6601	2.6263	2.6624
	Paklaida %	0.106858	1.51891	0.228981	1.606686
Paklaidos vidurkiai %		0.536471	2.327978	0.344017	2.116443
Bendras paklaidos vidurkis %		1.242479			
Maksimali paklaida %		1.839484	3.839255	1.537992	3.537763

Šio tyrimo rezultatai parodo, kad Euro kursas svyruoja maksimaliai 0,14% todėl norint atrinkti duomenis pagal eurą ir jo kursą, reikėtų paklaidą nustatyti ne daugiau kaip 0,2%, taip pat ši paklaidos riba užtikrina, kad oficialus Lietuvos banko EUR kursas nebus klaidingai atpažįstamas kaip ieškomi duomenys, kadangi vidutinė jo paklaida su ieškomais duomenimis yra 0,27% o minimali 0,24%.

Tuo tarpu dolerio kurso svyravimai yra didesni, USD kursas svyruoja maksimaliai 2,31%, o vidutinė paklaida sudaro 0,83%, tai parodo kad USD kursas yra mažiau stabilus negu EUR, todėl norint atrinkti duomenis apie valiutų kursus, kaip pradinius duomenys geriau naudoti EUR, tai leis sumažinti klaidingu duomenų išrinkimo tikimybę ir taip pat sudarys sąlygas spartesniam algoritmo veikimui, kadangi dėl mažos paklaidos ribos bus atmetama didžioji analizuojamų duomenų dalis.

4. 5 Skyriaus išvados

Atlikus sukurto algoritmo prototipo testavimą lyginant jį su kitais duomenų išgavimo įrankiais buvo nustatyta, kad sukurto algoritmo prototipas savo tikslumu nenusileidžia kitiems testuotiems įrankiams, o pagal F-score rodiklį lenkia juos nuo 7,7% iki 69,3%.

Ištestavus algoritmo prototipo gebėjimą prisitaikyti prie skirtingos strukturos duomenų šaltinių, kai naudojami tie patys duomenų išrinkimo nustatymai paaiškėjo, kad algoritmo tikslumo rodikliai smuko neženkliai, lyginant su ankstenių tyrimu. F-score rodiklis sumažėjo 4,8%, todėl galima teigti, kad algoritmas yra mažai priklausomas nuo duomenų šaltinio kodo struktūros, todėl jis gali būti naudojamas ir nuolat besikeičiančių duomenų šaltinių duomenims išgauti.

IŠVADOS

1. Išanalizavus egzistuojančius duomenų atrankos sprendimus, bei atlikus duomenų atrankos metodų tyrimą, buvo nustatytos šių metodų taikymo savybės. Gauti rezultatai leidžia optimizuoti duomenų išrinkimo iš internetinių puslapių algoritmą, tam tikrose jo fazėse taikant atitinkamą duomenų atrankos metodą ir taip išnaudojant jo privalumus.
2. Sukurtas duomenų išrinkimo iš internetinių puslapių algoritmas atrinkdamas duomenis remiasi jų panašumu su sistemai žinomais duomenimis, o ne jų aprašomos duomenų formos panašumu su žinomu šablonu. Tai leidžia sukurtą algoritmą taikyti duomenų išgavimui skirtingos, nuolat besikeičiančios struktūros duomenų šaltiniuose.
3. Išanalizavus testavimo rezultatus, paaiškėjo jog pagal pasiūlytą duomenų atrankos iš internetių puslapių algoritmą sukurto prototipo F-score rodiklis yra didesnis nei kitų testuotų įrankių (nuo 7,7% iki 69,3%). Tai leidžia teigti, jog sukurtas algoritmas ir jo pagrindu realizuotas prototipas leidžia tiksliai išrinkti norimus duomenis iš internetinių puslapių, kas leidžia naudoti vienodus nustatymus norint išgauti to paties duomenis iš skirtingų duomenų šaltinių.
4. Šiuo metu algoritmas yra tinkamas skaitiniams duomenims išgauti. Jį galima panaudoti valiutų kursams, vertybinių popierių indeksams, degalų kainoms ir kitiems panašiems duomenims išgauti. Kadangi algoritmo duomenų atpažinimo ir išrinkimo logika remiasi panašumu į algoritmui žinomus duomenis, jį galima modifikuoti, pritaikyti tekstinių duomenų išrinkimui. Tokio patobulinimo dėka galima ženkliai praplėsti algoritmo panaudojimo galimybes, reikia tik aprašyti kaip turi būti vertinamas tekstinių duomenų tarpusavio panašumas.

LITERATŪRA

1. M. Johnson "Top-Down Parsing" CS143, Handout 09, Prieiga per internetą <http://dragonbook.stanford.edu/lecture-notes/Stanford-CS143/07-Top-Down-Parsing.pdf>
2. "LL versus LR parsing" Prieiga per internetą <http://www.cs.uu.nl/docs/vakken/gont/chapter11.pdf>
3. J. Srivastava "Web Mining : Accomplishments & Future Directions" Prieiga per internetą: <http://www.ieee.org.ar/downloads/Srivastava-tut-pres.pdf>
4. „Web mining" Prieiga per internetą: <http://searchcrm.techtarget.com/sDefinition/0,,sid11_gci789009,00.html>
5. Amir H. Youssefi "Web mining (visual web mining)", 2004 Prieiga per internetą: < http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.58.6478 >
6. Duomenų išrinkimo įrankis „Visual Web Ripper", dokumentacija Prieiga per internetą: < http://www.visualwebripper.com/ >
7. A. Laender „A brief survey of Web data extraction tools" *Newsletter ACM SIGMOD*, Volume 31 Issue 2, June 2002 Psl 84-93 ACM New York, USA Prieiga per internetą: http://dl.acm.org/citation.cfm?id=565137
8. M. Teisseire "Sequential pattern mining: A survey on issues and approaches", *Encyclopedia of Data Warehousing and Mining, Information Science Publishing*, psl 3-29, 2005, Oxford University Press Prieiga per internetą: <http://www.sop.inria.fr/axis/International_Book_Encyclopedia_2005.pdf>
9. Web Scraper Plus+ oficialus puslapis. Prieiga per internetą: <http://www.velocityscape.com/Products/WebScraperPlus.aspx>
10. Web Information Extractor oficialus puslapis. Prieiga per internetą: <http://www.webinfoextractor.com/>
11. John Daintith. „LR parsing", „LL parsing", A Dictionary of Computing. 2004. *Encyclopedia.com* Prieiga per internetą <http://www.encyclopedia.com>
12. C. Chang „A Survey of Web Information Extraction Systems" *Knowledge and Data Engineering, IEEE Transactions on*, Vol. 18, Issue 10, psl 1411-1428, 2006 Prieiga per internetą: <http://130.49.220.23/~chang/265/proj10/sisref/5.pdf>
13. R. Kosala „Web Mining Research: A Survey" *SIGKDD Explorations*, Vol 2, psl 1-15, 200 Prieiga per internetą: <http://www.umiacs.umd.edu/~joseph/classes/enee752/Fall09/survey-2000.pdf>
14. N. Kushmerick „Finite-state approaches to web informatikon extraction" Proc. 3rd Summer Convention on Information Extraction, psl 77-91, 2002, Springer-Verlag Prieiga per internetą: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.100.6406&rep=rep1&type=pdf >
15. M. Castellano, „A Web Text Mining Flexible Architecture" International Journal of Computer Science & Engineering, 2007, Vol. 1 Issue 4, psl252 Prieiga per internetą: <http://www.waset.org/journals/waset/v32/v32-17.pdf>

16. M. Wright „Using Open Source Tools in Text Mining Research“ Prieiga per internetą: <http://www.swdsi.org/swdsi05/Proceedings05/paper_pdf/SWDSI_submission_usingOpenToolsinTextMining%20(T4D2).pdf>
17. R. Ssemmanda „Web Mining-Based University Search Portal“ 2011 *International Conference on Telecommunication Technology and Applications Proc .of CSIT* vol.5, 2011, IACSIT Press, Singapore Prieiga per internetą: <http://www.ipcsit.com/vol5/27-ICCCM2011-A079.pdf>
18. L. Youanyuan. „Research on text mining“ *American Journal of Engineering and Technology Research,* Vol. 11, No.9, 2011 Prieiga per internetą: < http://www.textmining.xpg.com.br/N0318.pdf >
19. S. Kuhlins „Tools for generating wrappers“ *LNCS*, psl 184-198, Springer, 2002 Prieiga per internetą:<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.165.9746&rep=rep1&type=pdf>
20. J. Han. „Research Challenges for Data Mining in Science and Engineering“, *Next Generation of Data Mining*, Chapman & Hall, 2009, Prieiga per internetą: <http://www.cs.uiuc.edu/~hanj/pdf/ngdm09_han_gao.pdf>
21. „Introduction to Web Mining“ Prieiga per internetą: <http://www.worldscibooks.com/etextbook/5832/5832_chap1.pdf>
22. R. Mooney. „Mining Knowledge from Text Using Information Extraction“ *SIGKDD Explorations*. Volume 7, Issue 1, psl 3-10, Prieiga per internetą: <http://www.cs.utexas.edu/~ml/papers/text-kddexplore-05.pdf>
23. S. Flesca „Web wrapper induction: a brief survey”, *Journal AI Communications*, Volume 17 Issue 2, April 2004, psl 57 – 61, IOS Press Amsterdam, The Netherlands. Prieiga per internetą: http://dl.acm.org/citation.cfm?id=1218707&CFID=78266630&CFTOKEN=58418246
24. K. Kaiser. „Information Extraction. a survey“ Prieiga per internetą: http://ieg.ifs.tuwien.ac.at/techreports/Asgaard-TR-2005-6.pdf
25. L. Park. „A Novel Web Text Mining Method Using the Discrete Cosine Transform“ Lecture Notes in Computer Science, Volume 2431, psl 385-397, 2002, Springer-Verlag Berlin Prieiga per internetą: <http://ww2.cs.mu.oz.au/~lapark/PKDD02_Park.pdf>
26. G. Fiumara „Automated Information Extraction from Web Sources : a Survey“, *Proceedings of Between Ontologies and Folksonomies Workshop in 3rd International Conference on Communities and Technology*, 2007, Prieiga per internetą: <http://www.mendeley.com/research/automated-information-extraction-from-web-sources-a-survey/>
27. C. Corley „Text and Structural Data Mining of Influenza Mentions in Web and Social Media“ *International Journal of Environmental Research and Public Health*, 2010, 7, 596-615 ISSN 1660-4601, Prieiga per internetą: <http://www.cs.uwaterloo.ca/~jhoey/teaching/cs793/papers/Text_Data_mining_web_media.pdf>

28. W. Zhang. „Web Text Mining ON XSSC“, *Knowledge and Systems Sciences: toward Knowledge Synt hesis and Creation Proceedings of KSS2006*, LNDS 8, Global- Link, September 22-25, 2006, Beijing, China
Prieiga per internetą: <http://meta-synthesis.iss.ac.cn/xjtang/paper/zhangwen_kss2006.pdf>
29. V. Gupta. „A Survey of Text Mining Techniques and Applications“ *Journal of Emerging Technologies in Web Intelligence*, Vol 1, No 1 (2009), 60-76, Aug 2009 Prieiga per internetą: < http://ojs.academypublisher.com/index.php/jetwi/article/view/01016076 >
30. R. Baumgartner „Scalable Web Data Extraction for Online Market Intelligence“ *VLDB ‘09*, August 24-28, 2009, Lyon, France,
Prieiga per internetą: <http://dc-pubs.dbs.uni-leipzig.de/files/vldb09-1075.pdf>
31. S. Sarawagi „Information Extraction“, *Foundations and Trends in Databases* Vol. 1, No. 3 psl. 261–377, 2007
Prieiga per internetą: <http://osm.cs.byu.edu/CS652s09/papers/Sarawagi.ieSurvey.pdf>

PRIEDAI

1 priedas. Internetinių puslapių duomenų išrinkimo lankstumo rytimo rezultatai

9 lentelė. Pradinis duomenų išrinkimas (eilutėmis)

Testo tikslas:	Pradinis duomenų išrinkimas (eilutėmis)			
Duomenų išdėstymas (eilutėmis, stulpeliais)	eilutėmis (pirmos svetainės failas)		Bendras duomenų fragmentų skaičius	523
Duomenų išdėstymo pakeitimai	nėra			
HTML struktūros pakeitimai	nėra			
	Duomenų išgavimui naudojami įrankiai			
	Sukurto algoritmo prototipas	Visual Web Ripper	Helium Scraper	OutWit Hub
Duomenų išrinkimo taisyklės	Raktažodis „eur“ 4 valiutų kursai duomenų atpažinimui	Naudojant GUI vartotojas sukuria išrinkimo taisykles	Naudojant GUI vartotojas pasirenka norimus išrinkti duomenys	Naudojant GUI vartotojas nustato duomenų išrinkimo ribas (regex)
Išrinktų duomenų fragmentų skaičius	45	45	72	72
Išrinkta ieškomų duomenų fragmentų	45	45	45	45
Atrinkimas (Recall)	1	1	1	1
Tikslumas (Precision)	1	1	0,625	0,625

10 lentelė. Pradinis duomenų išrinkimas (stulpeliais)

Testo tikslas:	Pradinis duomenų išrinkimas (stulpeliais)			
Duomenų išdėstymas (eilutėmis, stulpeliais)	stulpeliais (antros svetainės failas)		Bendras duomenų fragmentų skaičius	519
Duomenų išdėstymo pakeitimai	nėra			
HTML struktūros pakeitimai	nėra			
	Duomenų išgavimui naudojami įrankiai			
	Sukurto algoritmo prototipas	Visual Web Ripper	Helium Scraper	OutWit Hub
Duomenų išrinkimo taisyklės	Raktažodis „eur“ 4 valiutų kursai duomenų atpažinimui	Naudojant GUI vartotojas sukuria išrinkimo taisykles	Naudojant GUI vartotojas pasirenka norimus išrinkti duomenys	Naudojant GUI vartotojas nustato duomenų išrinkimo ribas (regex)
Išrinktų duomenų fragmentų skaičius	45	45	72	77
Išrinkta ieškomų duomenų fragmentų	45	45	45	45
Atrinkimas (Recall)	1	1	1	1
Tikslumas (Precision)	1	1	0,625	0,625

11 lentelė. Algoritmų lankstumo tyrimas, pakeičiamas duomenų šaltinis

Testo tikslas:	Patikrinti duomenų surinkimo algoritmų lankstumą, naudojamas antros svetaines nekoreguotas failas, tačiau duomenų išrinkimo nustatymai tokie pat kaip išrenkant duomenys iš pirmo failo			
Duomenų išdėstymas (eilutėmis, stulpeliais)	stulpeliais (antros svetainės failas)		Bendras duomenų fragmentų skaičius	519
Duomenų išdėstymo pakeitimai	duomenų išdėstymas pakeistas iš eilučių į stulpelius			
HTML struktūros pakeitimai	lentelės struktūra yra pakeičiama, tam kad pasikeistų duomenų išdėstymas			
	Duomenų išgavimui naudojami įrankiai			
	Sukurto algoritmo prototipas	Visual Web Ripper	Helium Scraper	OutWit Hub
Duomenų išrinkimo taisyklės	Raktažodis „eur“ 4 valiutų kursai duomenų atpažinimui	Pirmai svetainei pritaikyti nustatymai	Pirmai svetainei pritaikyti nustatymai	Pirmai svetainei pritaikyti nustatymai
Išrinktų duomenų fragmentų skaičius	45	0	78	82
Išrinkta ieškomų duomenų fragmentų	45	0	45	45
Atrinkimas (Recall)	1	0	1	1
Tikslumas (Precision)	1	0	0,577	0,549

12 lentelė. Algoritmų lankstumo tyrimas, pakeičiamas duomenų šaltinis

Testo tikslas:	Patikrinti duomenų surinkimo algoritmų lankstumą, naudojamas pirmos svetaines nekoreguotas failas, tačiau duomenų išrinkimo nustatymai tokie pat kaip išrenkant duomenys iš antro failo			
Duomenų išdėstymas (eilutėmis, stulpeliais)	eilutėmis		Bendras duomenų fragmentų skaičius	523
Duomenų išdėstymo pakeitimai	duomenų išdėstymas pakeistas iš stulpelių į eilutes			
HTML struktūros pakeitimai	lentelės struktūra yra pakeičiama, tam kad pasikeistų duomenų išdėstymas			
	Duomenų išgavimui naudojami įrankiai			
	Sukurto algoritmo prototipas	Visual Web Ripper	Helium Scraper	OutWit Hub
Duomenų išrinkimo taisyklės	Raktažodis „eur“ 4 valiutų kursai duomenų atpažinimui	Antrai svetainei pritaikyti nustatymai	Antrai svetainei pritaikyti nustatymai	Antrai svetainei pritaikyti nustatymai
Išrinktų duomenų fragmentų skaičius	45	0	0	1
Išrinkta ieškomų duomenų fragmentų	45	0	0	0
Atrinkimas (Recall)	1	0	0	0
Tikslumas (Precision)	1	0	0	0

13 lentelė. Algoritmų lankstumo tyrimas, sukuriamas 0-lygio div elementas

Testo tikslas:	Patikrinti duomenų surinkimo algoritmų lankstumą įnešus pakeitimų į pirmos svetainės HTML kodą (sukuriamas 0-lygio div elementas)			
Duomenų išdėstymas (eilutėmis, stulpeliais)	eilutėmis		Bendras duomenų fragmentų skaičius	524
Duomenų išdėstymo pakeitimai	nėra			
HTML struktūros pakeitimai	sukuriamas 0-lygio div elementas			
	Duomenų išgavimui naudojami įrankiai			
	Sukurto algoritmo prototipas	Visual Web Ripper	Helium Scraper	OutWit Hub
Duomenų išrinkimo taisyklės	Raktažodis „eur“ 4 valiutų kursai duomenų atpažinimui	Pirmai svetainei pritaikyti nustatymai	Pirmai svetainei pritaikyti nustatymai	Pirmai svetainei pritaikyti nustatymai
Išrinktų duomenų fragmentų skaičius	45	45	72	77
Išrinkta ieškomų duomenų fragmentų	45	45	45	45
Atrinkimas (Recall)	1	1	1	1
Tikslumas (Precision)	1	1	0,625	0,584

14 lentelė. Algoritmų lankstumo tyrimas, sukuriamas 0-lygio div elementas

Testo tikslas:	Patikrinti duomenų surinkimo algoritmų lankstumą įnešus pakeitimų į antros svetainės HTML kodą (sukuriamas 0-lygio div elementas)			
Duomenų išdėstymas (eilutėmis, stulpeliais)	stulpeliais		Bendras duomenų fragmentų skaičius	520
Duomenų išdėstymo pakeitimai	nėra			
HTML struktūros pakeitimai	sukuriamas 0-lygio div elementas			
	Duomenų išgavimui naudojami įrankiai			
	Sukurto algoritmo prototipas	Visual Web Ripper	Helium Scraper	OutWit Hub
Duomenų išrinkimo taisyklės	Raktažodis „eur“ 4 valiutų kursai duomenų atpažinimui	Antrai svetainei pritaikyti nustatymai	Antrai svetainei pritaikyti nustatymai	Antrai svetainei pritaikyti nustatymai
Išrinktų duomenų fragmentų skaičius	45	45	78	72
Išrinkta ieškomų duomenų fragmentų	45	45	45	45
Atrinkimas (Recall)	1	1	1	1
Tikslumas (Precision)	1	1	0,577	0,625

15 lentelė. Algoritmų lankstumo tyrimas, sukuriamas N-lygio div elementas

Testo tikslas:	Patikrinti duomenų surinkimo algoritmų lankstumą įnešus pakeitimų į pirmos svetainės HTML kodą (sukuriamas n-lygio div elementas)			
Duomenų išdėstymas (eilutėmis, stulpeliais)	eilutėmis		Bendras duomenų fragmentų skaičius	524
Duomenų išdėstymo pakeitimai	nėra			
HTML struktūros pakeitimai	sukuriamas div elementas, kuris yra tame pačiame lygyje kaip ir lentelė			
	Duomenų išgavimui naudojami įrankiai			
	Sukurto algoritmo prototipas	Visual Web Ripper	Helium Scraper	OutWit Hub
Duomenų išrinkimo taisyklės	Raktažodis „eur“ 4 valiutų kursai duomenų atpažinimui	Pirmai svetainei pritaikyti nustatymai	Pirmai svetainei pritaikyti nustatymai	Pirmai svetainei pritaikyti nustatymai
Išrinktų duomenų fragmentų skaičius	45	45	0	72
Išrinkta ieškomų duomenų fragmentų	45	45	0	45
Atrinkimas (Recall)	1	1	0	1
Tikslumas (Precision)	1	1	0	0,625

16 lentelė. Algoritmų lankstumo tyrimas, sukuriamas N-lygio div elementas

Testo tikslas:	Patikrinti duomenų surinkimo algoritmų lankstumą įnešus pakeitimų į antros svetainės HTML kodą (sukuriamas n-lygio div elementas)			
Duomenų išdėstymas (eilutėmis, stulpeliais)	stulpeliais		Bendras duomenų fragmentų skaičius	520
Duomenų išdėstymo pakeitimai	nėra			
HTML struktūros pakeitimai	sukuriamas div elementas, kuris yra tame pačiame lygyje kaip ir lentelė			
	Duomenų išgavimui naudojami įrankiai			
	Sukurto algoritmo prototipas	Visual Web Ripper	Helium Scraper	OutWit Hub
Duomenų išrinkimo taisyklės	Raktažodis „eur“ 4 valiutų kursai duomenų atpažinimui	Antrai svetainei pritaikyti nustatymai	Antrai svetainei pritaikyti nustatymai	Antrai svetainei pritaikyti nustatymai
Išrinktų duomenų fragmentų skaičius	45	45	0	77
Išrinkta ieškomų duomenų fragmentų	45	45	0	45
Atrinkimas (Recall)	1	1	0	1
Tikslumas (Precision)	1	1	0	0,584

17 lentelė. Algoritmų lankstumo tyrimas, sukuriamas 0-lygio table elementas

Testo tikslas:	Patikrinti duomenų surinkimo algoritmų lankstumą įnešus pakeitimų į pirmos svetainės HTML kodą (sukuriamas 0-lygio table elementas)			
Duomenų išdėstymas (eilutėmis, stulpeliais)	eilutėmis		Bendras duomenų fragmentų skaičius	536
Duomenų išdėstymo pakeitimai	nėra			
HTML struktūros pakeitimai	sukuriamas 0-lygio table elementas			
	Duomenų išgavimui naudojami įrankiai			
	Sukurto algoritmo prototipas	Visual Web Ripper	Helium Scraper	OutWit Hub
Duomenų išrinkimo taisyklės	Raktažodis „eur“ 4 valiutų kursai duomenų atpažinimui	Pirmai svetainei pritaikyti nustatymai	Pirmai svetainei pritaikyti nustatymai	Pirmai svetainei pritaikyti nustatymai
Išrinktų duomenų fragmentų skaičius	45	45	72	81
Išrinkta ieškomų duomenų fragmentų	45	45	45	45
Atrinkimas (Recall)	1	1	1	1
Tikslumas (Precision)	1	1	0,625	0,555

18 lentelė. Algoritmų lankstumo tyrimas, sukuriamas 0-lygio table elementas

Testo tikslas:	Patikrinti duomenų surinkimo algoritmų lankstumą įnešus pakeitimų į antros svetainės HTML kodą (sukuriamas 0-lygio table elementas)			
Duomenų išdėstymas (eilutėmis, stulpeliais)	stulpeliais		Bendras duomenų fragmentų skaičius	532
Duomenų išdėstymo pakeitimai	nėra			
HTML struktūros pakeitimai	sukuriamas 0-lygio table elementas			
	Duomenų išgavimui naudojami įrankiai			
	Sukurto algoritmo prototipas	Visual Web Ripper	Helium Scraper	OutWit Hub
Duomenų išrinkimo taisyklės	Raktažodis „eur“ 4 valiutų kursai duomenų atpažinimui	Antrai svetainei pritaikyti nustatymai	Antrai svetainei pritaikyti nustatymai	Antrai svetainei pritaikyti nustatymai
Išrinktų duomenų fragmentų skaičius	45	45	78	86
Išrinkta ieškomų duomenų fragmentų	45	45	45	45
Atrinkimas (Recall)	1	1	1	1
Tikslumas (Precision)	1	1	0,576	0,523

19 lentelė. Algoritmų lankstumo tyrimas, sukuriamas N-lygio table elementas

Testo tikslas:	Patikrinti duomenų surinkimo algoritmų lankstumą įnešus pakeitimų į pirmos svetainės HTML kodą (sukuriamas n-lygio table elementas)			
Duomenų išdėstymas (eilutėmis, stulpeliais)	eilutėmis		Bendras duomenų fragmentų skaičius	536
Duomenų išdėstymo pakeitimai	nėra			
HTML struktūros pakeitimai	sukuriamas table elementas, kuris yra tame pačiame lygyje kaip ir lentelė			
	Duomenų išgavimui naudojami įrankiai			
	Sukurto algoritmo prototipas	Visual Web Ripper	Helium Scraper	OutWit Hub
Duomenų išrinkimo taisyklės	Raktažodis „eur“ 4 valiutų kursai duomenų atpažinimui	Pirmai svetainei pritaikyti nustatymai	Pirmai svetainei pritaikyti nustatymai	Pirmai svetainei pritaikyti nustatymai
Išrinktų duomenų fragmentų skaičius	45	45	9	81
Išrinkta ieškomų duomenų fragmentų	45	45	0	45
Atrinkimas (Recall)	1	1	0	1
Tikslumas (Precision)	1	1	0	0,555

20 lentelė. Algoritmų lankstumo tyrimas, sukuriamas N-lygio table elementas

Testo tikslas:	Patikrinti duomenų surinkimo algoritmų lankstumą įnešus pakeitimų į antros svetainės HTML kodą (sukuriamas n-lygio table elementas)			
Duomenų išdėstymas (eilutėmis, stulpeliais)	stulpeliais		Bendras duomenų fragmentų skaičius	532
Duomenų išdėstymo pakeitimai	nėra			
HTML struktūros pakeitimai	sukuriamas div elementas, kuris yra tame pačiame lygyje kaip ir lentelė			
	Duomenų išgavimui naudojami įrankiai			
	Sukurto algoritmo prototipas	Visual Web Ripper	Helium Scraper	OutWit Hub
Duomenų išrinkimo taisyklės	Raktažodis „eur“ 4 valiutų kursai duomenų atpažinimui	Antrai svetainei pritaikyti nustatymai	Antrai svetainei pritaikyti nustatymai	Antrai svetainei pritaikyti nustatymai
Išrinktų duomenų fragmentų skaičius	45	0	9	86
Išrinkta ieškomų duomenų fragmentų	45	0	0	45
Atrinkimas (Recall)	1	0	0	1
Tikslumas (Precision)	1	0	0	0,523

21 lentelė. Algoritmų lankstumo tyrimas, pašalinama viena iš lentelės eilučių

Testo tikslas:	Patikrinti duomenų surinkimo algoritmų lankstumą įnešus pakeitimų į pirmos svetainės HTML kodą (pašalinama viena iš lentelės eilučių)			
Duomenų išdėstymas (eilutėmis, stulpeliais)	eilutėmis		Bendras duomenų fragmentų skaičius	514
Duomenų išdėstymo pakeitimai	pašalinami vienos eilutės duomenys			
HTML struktūros pakeitimai	pašalinama viena iš lentelės eilučių			
	Duomenų išgavimui naudojami įrankiai			
	Sukurto algoritmo prototipas	Visual Web Ripper	Helium Scraper	OutWit Hub
Duomenų išrinkimo taisyklės	Raktažodis „eur“ 4 valiutų kursai duomenų atpažinimui	Pirmai svetainei pritaikyti nustatymai	Pirmai svetainei pritaikyti nustatymai	Pirmai svetainei pritaikyti nustatymai
Išrinktų duomenų fragmentų skaičius	40	40	64	64
Išrinkta ieškomų duomenų fragmentų	40	40	40	40
Atrinkimas (Recall)	1	1	1	1
Tikslumas (Precision)	1	1	0,625	0,625

22 lentelė. Algoritmų lankstumo tyrimas, pašalinamas vienas iš lentelės stulpelių

Testo tikslas:	Patikrinti duomenų surinkimo algoritmų lankstumą įnešus pakeitimų į antros svetainės HTML kodą (pašalinamas vienas iš lentelės stulpelių)			
Duomenų išdėstymas (eilutėmis, stulpeliais)	stulpeliais		Bendras duomenų fragmentų skaičius	511
Duomenų išdėstymo pakeitimai	pašalinami vieno stulpelio duomenys			
HTML struktūros pakeitimai	pašalinamas vienas iš lentelės stulpelių			
	Duomenų išgavimui naudojami įrankiai			
	Sukurto algoritmo prototipas	Visual Web Ripper	Helium Scraper	OutWit Hub
Duomenų išrinkimo taisyklės	Raktažodis „eur“ 4 valiutų kursai duomenų atpažinimui	Antrai svetainei pritaikyti nustatymai	Antrai svetainei pritaikyti nustatymai	Antrai svetainei pritaikyti nustatymai
Išrinktų duomenų fragmentų skaičius	40	40	70	69
Išrinkta ieškomų duomenų fragmentų	40	40	40	40
Atrinkimas (Recall)	1	1	1	1
Tikslumas (Precision)	1	1	0,571	0,580

23 lentelė. Algoritmų lankstumo tyrimas, pašalinama EUR eilutė

Testo tikslas:	Patikrinti duomenų surinkimo algoritmų lankstumą įnešus pakeitimų į pirmos svetainės HTML kodą (pašalinama EUR eilutė)			
Duomenų išdėstymas (eilutėmis, stulpeliais)	eilutėmis		Bendras duomenų fragmentų skaičius	514
Duomenų išdėstymo pakeitimai	pašalinami vienos eilutės duomenys			
HTML struktūros pakeitimai	pašalinama EUR eilutė			
	Duomenų išgavimui naudojami įrankiai			
	Sukurto algoritmo prototipas	Visual Web Ripper	Helium Scraper	OutWit Hub
Duomenų išrinkimo taisyklės	Raktažodis „eur“ 4 valiutų kursai duomenų atpažinimui	Pirmai svetainei pritaikyti nustatymai	Pirmai svetainei pritaikyti nustatymai	Pirmai svetainei pritaikyti nustatymai
Išrinktų duomenų fragmentų skaičius	0	40	64	64
Išrinkta ieškomų duomenų fragmentų	0	40	40	40
Atrinkimas (Recall)	0	1	1	1
Tikslumas (Precision)	0	1	0,625	0,625

24 lentelė. Algoritmų lankstumo tyrimas, pašalinamas EUR stulpelis

Testo tikslas:	Patikrinti duomenų surinkimo algoritmų lankstumą įnešus pakeitimų į antros svetainės HTML kodą (pašalinamas EUR stulpelis)			
Duomenų išdėstymas (eilutėmis, stulpeliais)	stulpeliais		Bendras duomenų fragmentų skaičius	511
Duomenų išdėstymo pakeitimai	pašalinami vieno stulpelio duomenys			
HTML struktūros pakeitimai	pašalinamas EUR stulpelis			
	Duomenų išgavimui naudojami įrankiai			
	Sukurto algoritmo prototipas	Visual Web Ripper	Helium Scraper	OutWit Hub
Duomenų išrinkimo taisyklės	Raktažodis „eur“ 4 valiutų kursai duomenų atpažinimui	Antrai svetainei pritaikyti nustatymai	Antrai svetainei pritaikyti nustatymai	Antrai svetainei pritaikyti nustatymai
Išrinktų duomenų fragmentų skaičius	0	40	70	74
Išrinkta ieškomų duomenų fragmentų	0	40	40	40
Atrinkimas (Recall)	0	1	1	1
Tikslumas (Precision)	0	1	0,571	0,541