



VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
INFORMATIKOS INSTITUTAS
KOMPIUTERINIO IR DUOMENŲ MODELIAVIMO KATEDRA

Bakalauro baigiamasis darbas

Svetimų kalbų asmenvardžių perrašymo lietuviškai algoritmai

Atliko:

Adomas Bazinys

parašas

Vadovas:

dr. Pijus Kasparaitis

Vilnius
2020

Turinys

Santrauka	4
Summary	5
Įvadas	6
1. Svetimų kalbų žodžių perrašymo lietuviškai principai	7
1.1. Transkripcija	7
1.2. Transliteracija	7
1.3. Adaptavimas	8
2. Vokiškų asmenvardžių sulietuvinimas	9
2.1. Vokiečių kalbos abėcėlė	9
2.2. Lietuvinimo taisyklės	9
2.3. Samplaikų skilimas	9
2.4. Priebalsėms taikomos taisyklės	10
2.4.1. Besąlyginiai keitimai	10
2.4.2. Sąlyginiai keitimai	11
2.4.3. Keitimai žodžio pradžioje	11
2.5. Balsėms taikomos taisyklės	12
2.5.1. Balsių trumpumo taisyklės	12
2.5.2. Balsių ilgumo taisyklės	12
2.5.3. Išimtys balsių ilgumo nustatyme	13
2.6. Galūnių nustatymas	13
2.6.1. Vyriškos galūnės	14
2.6.2. Moteriškos galūnės	14
3. Algoritmo realizacija	15
3.1. Formalių taisyklių transkribavimo metodas	15
3.2. Pseudokodas	16
3.3. Kontekstų tikrinimas	17
3.4. Taisyklių aprašymas	18
3.4.1. Eiliškumo problema	18
3.4.2. Našumo vertinimas	19
3.4.3. Taisyklių praleidimas	19
3.4.4. Taisyklių kiekio sumažinimas	20
3.4.5. <i>If - Else</i> principas	21
4. Rezultatai	21
4.1. Klaidų analizė	21
4.2. Klaidų atsiradimo priežastys	22
4.3. Algoritmo tikslumo vertinimas	22
5. Programos grafinė sąsaja	23
Išvados ir rekomendacijos	24

Ateities tyrimų planas	25
Literatūros šaltiniai	26
Priedai	27
A. Testavimui naudotų asmenvardžių sąrašas	28

Santrauka

Šiomis dienomis visuomeniškai baigia įsitvirtinti praktika pusiau adaptuoti asmenvardžius užrašant originaliąsias jų formas, pritaikant lietuvišką galūnę bei vartojant nelietuviškos abėcėlės raides. Svetimvardžių rašymas publicistiniuose straipsniuose originalo kalba mažina gimtosios kalbos dvasią, kurią išsaugoti yra labai svarbu. Baigiamajame darbe nagrinėjamas sukurtas algoritmas, sulietuvinantis vokišką asmenvardį pagal kamieno perteikimo atvejus vokiečių kalbai. Taisyklėms taikyti naudojamas formalių taisyklių transkribavimo metodas, kurio veikimas paremtas kairiojo ir dešiniojo konteksto tikrinimu. Darbe aprašytos balsių ilgumo ir trumpumo, priebalsėms taikomos taisyklės bei išimtys. Remiantis testinių duomenų masyvu ir algoritmo išvedamais sulietuvintais asmenvardžiais įvertintas algoritmo tikslumas ir greitimeika. Atlikta klaidų analizė ir aptarti potencialūs būdai algoritmo tikslumui ir spartai pagerinti.

Summary

Algorithms for Rewriting Foreign Names into Lithuanian

Currently, the practice of half-adapting personal names by writing their original forms, adapting the Lithuanian ending and using the letters of the non-Lithuanian alphabet is a common practice. The spelling of foreign words in journalistic articles in the original language reduces the spirit of the native language, which is very important to preserve. This bachelor's thesis analyzes the developed algorithm which is able to lithuanize the German personal name according to the rules for lithuanizing German names. The rules use a formal rules transcription method which works by checking the left and right context. The length and shortness of vowels, rules and exceptions to consonants are described. According to the test data array and algorithmic output for a German personal names the accuracy and speed of the algorithm is tested. An error analysis was performed and potential ways to improve the accuracy and speed of the algorithm is discussed.

Įvadas

Žiniasklaida kasdien publikuoja gausybę straipsnių, naujienų iš kitų pasaulio šalių ir žemynų. Pačios aktualiausios ir svarbiausios žinios akimirksniu atsiduria įvairių pasaulio valstybių naujienų portaluose. Neretai tose žiniose minimi ir svetimvardžiai, kuriuos, vadovaujantis visuomenėje priimtomis etiketo normomis, versti į gimtąją kalbą taisyklingai yra privaloma. Šiuolaikinėje visuomenėje teisingas, gramatinėmis taisyklėmis paremtas užsienio kalbų asmenvardžių sulietuvinimas yra vienas iš kalbos kultūros, pagarbos asmenvardžio savininkui rodiklių. Šiomis dienomis publicistikoje įprasta palikti pusiau adaptuotus asmenvardžius prie svetimvardžio pridėjus tik lietuvišką galūnę. Šiek tiek rečiau šalia pusiau adaptuoto svetimžodžio skliaustuose parašomas sulietuvintas svetimžodžio variantas arba atvirkščiai - originalas pateikiamas skliaustuose [4]. Asmenvardžių perrašymo į kitą kalbą svarbumą pagrindžia keletas priežasčių. Viena iš jų - tam, kad tolimesnės kartos, galėtų lengviau atsekti savo šaknis, išsibarsčiusias po visą pasaulį. Kalbininkai ne visada taisyklingai perrašydavo užsienietiškus asmenvardžius į lietuvių kalbą. Tai vykdavo Lietuvos okupacijos metais, kuomet norint išversti vardą ar pavardę, pirmiausia jie buvo išverčiami į rusų, o tik paskui į lietuvių kalbą. Dėl to atsirado daug iškraipytų užsienietiškų vardų bei pavardžių [6]. Kalbotyros specialistai tiksliai asmenvardžius gali perrašyti tik vadovaudamiesi tam tikrais sulietuvinimo taisyklių rinkiniais. Šiame technologijų amžiuje tokio tipo perrašą periminėja programiniai algoritmai, kurie puikiai atlieka bet kokias formaliomis taisyklėmis paremtas operacijas.

1985 m. buvo išleistas nelietuviškų tikrinių vardų rašymo taisyklių rinkinys su mintimi, kad tie patys vardai leidiniuose būtų rašomi taip pat, nes iki tol įvairiuose leidiniuose minimi nelietuviški tikriniai vardai buvo rašomi labai skirtingai. Būtent šiuo taisyklių rinkiniu paremta bakalauro baigiamojo darbo praktinė dalis. Jos tikslas yra realizuoti algoritmą, kuris sulietuvintų vokišką vardą ir pavardę pagal formalias taisykles. Svarbu paminėti, kad algoritmas niekaip nebus susijęs su *tradiciniais tikriniais vardais*, kurių lietuviškoji forma turi vartojimo tradiciją ir formalių taisyklių neatitinka [8]. Toks algoritmas galėtų būti potencialus sprendimas prieš tai paminėtai problemai spręsti. Toks algoritmas jau yra realizuotas lenkiškiems bei angliškiems asmenvardžiams, todėl dalis funkcionalumo bus paimta iš jau egzistuojančių šaltinių. Algoritmas bus realizuotas *JavaScript* programavimo kalba.

Nei vienas iš darbe minimų procesų tiksliai nenusako to, kas bus įgyvendinama praktinėje darbo dalyje. Šiame darbe pirmiausia supažindinama būtent su tais procesais - transliteracijos, transkripcijos ir adaptavimo, kurie tiksliausiai ir atspindės darbo praktinę dalį, bus paaiškintos pačios lietuvinimo taisyklės. Bus paruoštas vokiškų asmenvardžių masyvas ir realizavus algoritmą analizuojami vokiškų asmenvardžių sulietuvinimo rezultatai. Atsiradus klaidingai sulietuvintų asmenvardžių - bus atlikta ir klaidų analizė apie tai, kas padarė įtaką klaidų atsiradimui ir kaip jų būtų galima išvengti algoritmą tobulinant.

1. Svetimų kalbų žodžių perrašymo lietuviškai principai

Nelietuviškų tikrinių vardų rašymas turi bent apytiksliai perteikti originalo tarimą, kuo mažiau nukrypstant nuo originalo rašybos. Jų perraša ir transkribavimas į kitą kalbą susideda iš dviejų etapų: asmenvardžio perrašymo į kitą rašto sistemą ir asmenvardžio kamieno bei galūnės sulietuvinimo. Pirmajame etape asmenvardis yra perrašomas iš originalo rašto sistemos į kitą, tai reiškia, kad asmenvardis yra *transliteruojamas*. Paskui, remiantis kamieno perteikimo atvejais ir gramatinimo taisyklėmis, lietuvinamas asmenvardžio kamienas ir galas [8]. Nei vienas šiame skyriuje paminėtas procesas tiksliai nenusako praktiniame darbe automatizuojamo proceso, tačiau visi šie procesai yra glaudžiai susiję su praktiniu darbu.

1.1. Transkripcija

Transkripcija - kalbos žodžių ir garsų perteikimas rašytine forma. Šio proceso metu yra nustatoma, kaip žodis turi būti tariamas. Tam, kad nustatyti žodžio tarimą, pirmiausiai reikia jį suskiemuoti bei sukirčiuoti, nustatyti, kokį fonetinį vienetą žodžio garsai atitinka. Fonetiniai vienetai nusako, kaip turi būti tariamas tam tikras žodžio skiemuo [8].

Transkribavimas yra naudojamas teksto - kalbos sintezės procese. Šio proceso metu žodis, pagal tam tikrus taisyklių rinkinius, skirtingus kiekvienai kalbai, yra taisyklingai ištariamas girdimąja forma. Procesas yra paremtas formaliomis taisyklėmis ir naudingas tuo, kad jo dėka gali būti išaiškintas, tarkime, iki šiol nežinoto, nevartoto ar naujo žodžio taisyklingas tarimas. Šiame procese žodis suskaidomas į skiemenis. Tai padarius randamas kirčiuotas skiemuo ir nustatomas jo ilgumas, tiksliau - kurio tariamojo žodžio skiemens raidė išsiskiria iš kitų savo tarimo intensyvumu ir skiemens tono aukščio kaita - priegaide. Tada visi skiemenys, įskaitant kirčiuotąjį, transkribuojami - tekstas verčiamas į fonetinius vienetus - garsus, ko pasekoje prieinama prie paskutinio šio proceso etapo - garso sugeneravimo [5].

1.2. Transliteracija

Transliteracija - teksto perrašymas, naudojant kitą kalbos rašto sistemą, sakykime, rusiškos kirilicos raides perrašant į lotynišką abėcėlę (žiūrėti 1 pavyzdį). Šis procesas naudojamas transkribuojant ar lietuvinant tam tikrus žodžius, kurie originaliai užrašyti kita rašto sistema. Taip pat procesas yra skirtas tiems, kurie nesupranta originalaus teksto užrašymo, tačiau turi poreikį skaityti žodžius originalo kalba [8].

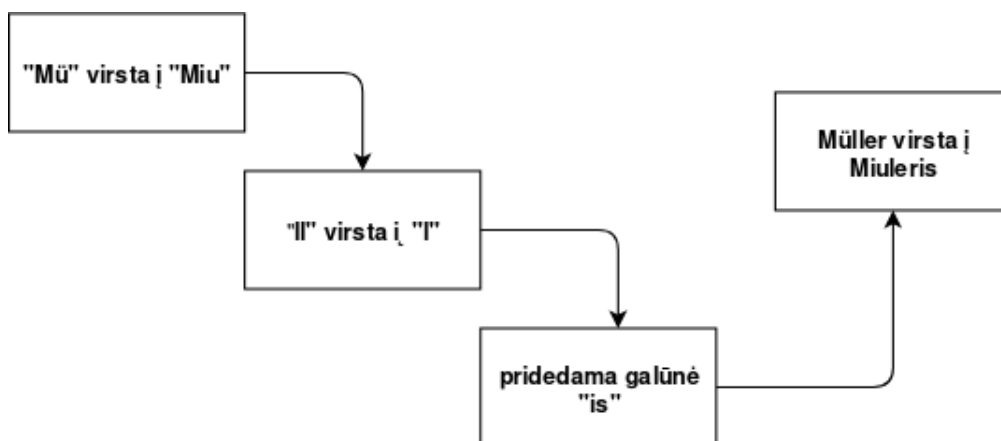
ТРАНСЛИТЕРАЦИЯ → *TRANSLITERACIJA*

1 pav. Transliteracija

1.3. Adaptavimas

Tai procesas, kurio metu svetimų kalbų asmenvardžiams yra pritaikomos lietuviškos galūnės. Šis procesas remiasi visoms kalboms bendromis taisyklėmis, padedančiomis nustatyti svetimvardžio galūnę, o dalį iki galūnės perteikti kuo mažiau nukrypstant nuo originalo rašybos, tomis pačiomis lietuviškomis raidėmis, bei atsižvelgiant į susidariusią tradiciją ir vadovaujantis kalbos jausmu. Adaptavimas reikalauja kalbos, iš kurios kilęs asmenvardis, mokėjimo, tam, kad būtų aišku, kaip jis tariamas ir kirčiuojamas [4].

Vokiška pavardė **Müller** sulietuvintai skambėtų kaip **Miuleris**, pritaikius taisykles, pateiktas apačioje (žiūrėti 2 paveikslėlį). Šis procesas gali pasirodyti panašus į procesą, įgyvendinamą praktinėje dalyje, tačiau jis nėra paremtas jokiais formaliomis taisyklėmis.



2 pav. Adaptavimas

2. Vokiškų asmenvardžių sulietuvinimas

Asmenvardžių sulietuvinimas pagrįstas taisyklėmis, kurios taikomos kiekvienai einamajai raidei arba jų junginiui, padarant įvairias išvadas apie tai, ar, tarkime, einamasis balsis yra ilgas ar trumpas, ar samplaika skyla į skirtingus skiemenis, ar ne - pritaikius tokias ir panašias taisykles yra išvedamas vokiško asmenvardžio atitikmuo lietuvių kalba [5]. Apytiksliai egzistuoja apie 56 vokiškų asmenvardžių sulietuvinimo taisyklės [8].

2.1. Vokiečių kalbos abėcėlė

Didelių skirtumų tarp vokiečių ir lietuvių kalbų abėcėlių nėra. Abi abėcėlės turi tas pačias lotyniškas raides, tačiau yra keletas skirtumų. Vienas akivaizdžiausių simbolių, padedančių atpažinti vokiečių kalbą, yra papildomos raidės su vadinamuoju umliautu. Jos atpažįstamos iš dvitaškio viršuje. Vokiečių kalba iš viso turi 29 raides įskaitant raides su umliautais ir escetą (žiūrėti 1 lentelę).

1 lentelė. Vokiečių kalbos abėcėlė

A a	O o	U u	I i	B b	C c	D d	F f	G g	H h	J j	K k	L l	M m	N n
Ä ä	Ö ö	Ü ü	E e	P p	Q q	R r	S s	T t	V v	W w	X x	Y y	Z z	ß

Raidės pažymėtos umliautais tariamos minkštai, prie lotyniško raidės varianto pridedant balse [i]. Pavyzdžiui, vokiškos raidės [ö] garsinis atitikmuo lietuvių kalboje bus [io].

2.2. Lietuvinimo taisyklės

Lietuvinimo taisyklių aibė taikoma raidžių junginiams iš priebalsių ir balsių, vien iš balsių ar priebalsių, taip pat atskirai pavienėms priebalsėms ir balsėms. Balsių taisykles įgyvendinti yra kur kas sudėtingiau nei priebalsių, nes kiekvienai iš jų visuomet reikia pritaikyti vokiečių tikrinių vardų balsių ilgumo nustatymo taisykles (skaityti 2.5 skyrelį). Tai pagrindinės taisyklės, kuriomis paremtas lietuvinimo algoritmo veikimas. Jose aprašytos visos vokiečių abėcėlės raidės bei dvigarsiai, jų atitikmenys lietuvių kalboje ir atitikimo sąlygos. Praktiniame darbe realizuojamas algoritmas automatizuos šių taisyklių pritaikymą.

Teisingai pritaikius šias taisykles vokiškam asmenvardžiui bus gaunama jo sulietuvinta forma. Šiuose atvejuose dominuoja taisyklės, susijusios su žodžių samplaika, balsių ilgumu bei trumpumu ir tam tikrų dvigarsių atitikmenimis lietuvių kalboje. Šiame skyrelyje bus plačiau aptartos kiekvienos taisyklių grupės ir jų taikymo pavyzdžiai.

2.3. Samplaikų skilimas

Samplaika yra garsų junginys, pasireiškiantis tam tikrų garsų junginio skilimo į du atskirus skiemenis metu. Dažniausiai tai galima pastebėti vokiškuose žodžiuose, turinčiuose savyje dvibalsiai [ay], [ai], [ey], [ei], [äu] ir [eu]. Lietuvinant galimi du su samplaika susiję atvejai: kai samplaika lieka viename skiemenyje arba samplaika skyla į skirtingus skiemenis [8].

Žemiau pavaizduoti keli samplaikos pasilikimo viename skiemenyje ir samplaikos skilimo į skirtingus skiemenis atvejai, atitinkamai 3 ir 4 pavyzdžiuose.

1	Aybar	→	Aibaras		[ay]
2	Daiderich	→	Daiderichas		[ai]
3	Meybohm	→	Meibomas		[ey]
4	Weimar	→	Veimaras		[ei]
5	Bräuning	→	Broiningas		[äu]
6	Euler	→	Oileris		[eu]

3 pav. Samplaika lieka viename skiemenyje

1	Altay	→	Altajus		[ay]
2	Peyer	→	Pejeris		[ey]
3	Maier	→	Majeris		[ai]
4	Steiermark	→	Štejermarkas		[ei]
5	Bräuer	→	Brojeris		[äu]
6	Heuer	→	Hojeris		[eu]

4 pav. Samplaika skyla į skirtingus skiemenis

2.4. Priebalsėms taikomos taisyklės

Priebalsėms taikomos taisyklės yra gerokai paprastesnės negu balsėms, todėl, kad joms labiau būdingos daug paprastesnės sąlygos nei balsėms. Dažniausiai tai būna tam tikrų priebalsių junginių pakeitimas į kitą esant labai paprastoms ir aiškioms arba nesant jokioms sąlygoms.

2.4.1. Besąlyginiai keitimai

Vienas dažniausių atvejų yra dvejų tokių pat priebalsių einančių viena šalia kitos apjungimas į vieną priebalsę lietuviškoje formoje (žiūrėti 5 pavyzdį). Ši taisyklė taikoma beveik visoms raidėms, išskyrus raides su umliautais ir raides, kurios lietuvių rašytinėje kalboje išvis nevartojamos, kaip **[w]**, **[B]**, **[x]** ir **[q]**. Toliau seka tam tikrų priebalsių junginių (dvigarsių ir pan.) pakeitimas į jų atitikmenis tinkančius lietuvių kalbai (žiūrėti 6 pavyzdį) [8]. Svarbu atkreipti dėmesį į tai, kad būtent šios taisyklės lietuvinimo algoritme yra pritaikomos paskutinėje vietoje, kadangi jos yra besąlyginės.

1	Feddermann	→	Federmanas		[dd]
2	Eggenweiler	→	Egenveileris		[gg]
3	Keppler	→	Kepleris		[pp]

5 pav. Dvejų priebalsių kitimas į vieną

1 Kretzschmann	→ Krečmanas	[tزش]
2 Schneider	→ Šneideris	[sch]
3 Uhlberg	→ Ūlbergas	[uh]

6 pav. Paprasti keitimai

2.4.2. Sąlyginiai keitimai

Taip pat priebalsėms būna taikomos sąlyginės taisyklės, kurių pritaikymas arba nepritaikymas priklauso nuo kairiųjų ir dešiniųjų kontekstų. Viena iš įdomesnių ir paprastesnių taisyklių yra taikoma priebalsei **[c]** - jeigu prieš ją eina **[e]**, **[i]** arba **[y]**, tada lietuviškoje formoje paliekama **[c]**, kitu atveju, prieš visas priebalses, taip pat prieš **[a]**, **[o]**, **[u]** arba **[c]** raidei esant žodžio gale - lietuvinant ji keičiama į **[k]** (žiūrėti 7 pavyzdį) [8].

1 Cyprian	→ Cyprianas	[c]
2 Clostermeyer	→ Klostermejeris	[c]
3 Marc	→ Markas	[c]

7 pav. Priebalsėms taikomų taisyklių pavyzdžiai

2.4.3. Keitimai žodžio pradžioje

Vokiečių kalboje yra keletas atvejų, kuomet tam tikriems garsų junginiams taisyklė taikoma tik žodžio pradžioje (**[st]**, **[sp]**, **[üh]** - žiūrėti 8 pavyzdyje) arba po priešdėlio, tačiau priešdėliai asmenvardžiuose vokiečių kalboje, lygiai taip pat kaip ir lietuvių kalboje yra labai retas atvejis, todėl algoritme bus daroma išimtis ir taisyklės priešdėliams nebus taikomos [8].

1 Steinmiller	→ Šteinmileris	[st]
2 Springholz	→ Špringolcas	[sp]
3 Ührchen	→ Yrchenas	[üh]

8 pav. Priebalsės kitimas

2.5. Balsėms taikomos taisyklės

Egzistuoja penkios formalios taisyklės balsių ilgumui ir dvi balsių trumpumui nustatyti, taisyklės pavaizduotos žemiau, atitinkamai 2 ir 3 lentelėse [8].

2.5.1. Balsių trumpumo taisyklės

2 lentelė. Balsių trumpumo taisyklės lietuvinant vokišką asmenvardį

Balsis trumpas, jeigu..	
1. po jo eina bb, ck, dd, ff, ll, mm, nn, pp, rr, ss, tt, tz, pf, x .	2. po jo eina keli priebalsiai priklausantys tai pačiai morfemai.
Ebber [e] - Eberis	Achermann [a] - Achermanas
Ammann [a] - Amanas	Sternberg [e] - Šternbergas
Ullritz [u] - Ulricas	Bernard [e] - Bernardas

2.5.2. Balsių ilgumo taisyklės

3 lentelė. Balsių ilgumo taisyklės lietuvinant vokišką asmenvardį

Balsis ilgas, jeigu..				
1. po jo eina netariamoji h , po kurios eina balsė arba l, m, n, r .	2. jis pažymėtas dviguba balse.	3. jis yra kirčiuoto skiemens gale.	4. šalia jo yra netariamoji e , einanti samplaikoje ie .	5. jis yra kirčiuotame skiemenyje, kuris tampa atviru, žodį kaitant.
Kuhlmann [u] - Kūlmanas	Aaron [a] - Aronas	Cremer [e] - Krėmeris	Biertz [i] - Byrcas	Gut [u], die Güte [ü] - Gūtas
Falandas [a] - Fahland	Schuurman [u] - Šūrmanas	Rubel [u] - Rūbelis	Diebald [i] - Dybaldas	
Friehling [i] - Frylingas	Bloom [o] - Blomas	Berend [e] - Bėrendas	Zwiebler [i] - Cvybleris	

2.5.3. Išimtys balsių ilgumo nustatyme

Pasitaiko atvejų, kai balsių ilgumo nustatyti nepavyksta, tokiu atveju verčiant lietuvinant vokišką asmenvardį einamąją balsę laikome trumpa - rašome ir tariame trumpai (žiūrėti 4 lentelę) [8].

4 lentelė. Išimtys

Balsis gali būti tiek trumpas, tiek ilgas, jeigu..
1. prieš jį eina ß, st, ch, r + dantinis, sch, tsch .
Schuster [u] - Šusteris
Faustmann [a] - Faustmanas ([a] - ilgas)

2.6. Galūnių nustatymas

Vokiečių kalboje moteriškos ir vyriškos pavardės nesiskiria (*Ernst Müller* ir *Sophia Müller*), tačiau lietuvinant yra taikomos tam tikros galūnės tiek moteriškai, tiek vyriškai lyčiai. Yra du būdai pritaikyti vokiškų asmenvardžių galūnes lietuvių kalbai. Pirmasis - lietuvišką galūnę pridėti prie užsienietiško asmenvardžio galūnės. Antrasis būdas - pakeisti perrašomo asmenvardžio galūnę, kuri baigiasi balse, į lietuvišką galūnę. Toks lietuviškų galūnių pridėjimas taikomas tuomet, kai nėra svarbus tikslumas, o siekiama paprastumo [2]. Asmenvardžio galūnė lietuviškoje formoje priklauso nuo kairiojo konteksto. Taip pat egzistuoja asmenvardžių galūnės, kurios lietuviškoje formoje išlieka tokios pat kaip vokiečių kalboje, tarp jų **[a]**, **[e]**, **[as]**, **[is]** (žiūrėti 5 lentelę) [8].

5 lentelė. Galūnių adaptavimas

Galūnė	Atitikmuo lietuvių kalboje
[a]	[a]
[e]	[ė]
[as]	[as]
[is]	[as], [is]

2.6.1. Vyrishkos galūnės

Lietuvių vyrishkos giminės galūnės yra - **[as]**, **[is]** ar **[(i)us]**. Vyrishkuose asmenvardžiuose galūnės dažniausiai pridedamos arba pakeičiamos, nors praktikoje dažnai pasitaiko atvejų, kai jokia lietuviška galūnė nėra pridedama [1].

Pavyzdžiui, vyrishkam asmenvardžiui baigiantis balsiu arba dvibalsiu lietuviškoje formoje dažniausiai galūnė nėra pridedama. Vokiečių vyrishkų vardų galūnės, kurios įprastos lietuvių kalboje kaip **[as]**, **[is]**, **[(i)us]** lietuvinant nėra keičiamos ir galimas tokių galūnių kaitaliojimas linksniais. Lietuviška galūnė pridedama prie asmenvardžių, kurie baigiasi priebalsiu. Užsienietiškos pavardės, kurių tarimo metu yra girdima galūnės priebalsė, dažniausiai lietuvinamos pridedant lietuvišką galūnę **[as]**. Galūnė **[is]** pridedama po priebalsių **[l]** ir **[r]** bei po minkštojo priebalsio. Užsienietiškos pavardės, kurios baigiasi priebalsiu **[j]** pakeičiamos lietuviška galūne **[(i)us]** [3].

1 Bernd → Berndas [as]
2 Schäffer → Šeferis [is]
3 Nay → Najus [us]
4 Tschirma → Čirma [a]
5 Thomas → Tomas [as]

9 pav. Vyrishko asmenvardžio galūnės adaptavimas

2.6.2. Moterishkos galūnės

Lietuviškos moterishkos galūnės yra **[a]**, **[ė]**. Perrašant užsienietišką moterishką pavardę į lietuvišką, moterishkos giminės pavardėse šeiminė padėtis neatsispindi. Taigi, lietuvinant moterishką pavardę nėra atsižvelgiama į tai, ar moteris ištėkėjusi, ar ne. Prie lietuvinamų pavardžių nėra pridedamos galūnės, jei jos baigiasi priebalse. Tai yra vienintelis būdas įsitikinti, kad lietuviškoje formoje omeny yra turima moterishka, o ne vyrishka pavardė. Minėtos moterishkų pavardžių galūnės yra pridedamos tik tuo atveju, jei jos baigiasi balse. Užsienietiškų moterishkų vardų lietuvinimas yra platesnis. Galima pakeisti užsienietiškų vardų galūnes skirtingomis moterishkomis galūnėmis, o kai kuriais atvejais jų visai ne pridėti - bet kuris variantas bus teisingas [2].

1 Elisabeth → Elizabeta [a], Elizabetė [ė]
2 Isabell → Izabela [a], Izabelė [ė]
3 Anne → Ana [a], Anė [ė]

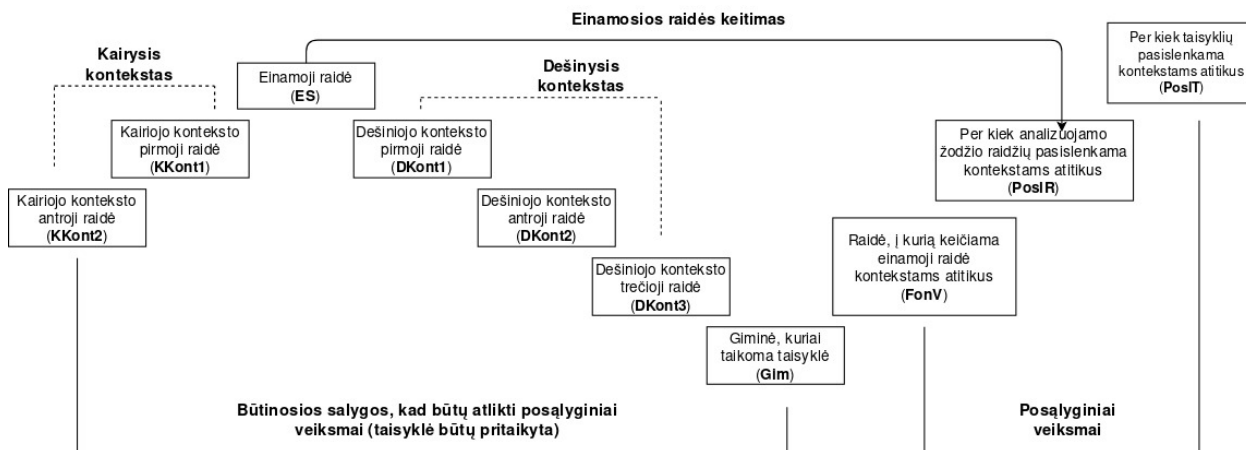
10 pav. Moterishko asmenvardžio galūnės adaptavimas

3. Algoritmo realizacija

Praktinio darbo tikslas - parašyti algoritmą, kuris gebėtų priimti vokišką asmenvardį ir jį su-lietuvinti remiantis aptartomis formaliomis taisyklėmis. Lengviausiai informatikoje pritaikomas yra formalių taisyklių transkribavimo metodas, kuriam nėra reikalingi aukštesnio lygio sprendimai kaip neuroniniai tinklai ar dirbtinis intelektas, todėl būtent šiuo metodu ir bus paremtas algoritmo veikimas.

3.1. Formalių taisyklių transkribavimo metodas

Tai metodas, kurio veikimas paremtas oficialiomis tam tikros kalbos garsų transkribavimo taisyklėmis. Tikrasis šio metodo tikslas yra nustatyti žodžio tarimą. Metodo veikimo principas yra toks, kad programa iteruoja per kiekvieną transkribuojamo (lietuvinamo) žodžio raidę ir, atsižvelgdama į kairėje bei dešinėje esantį raidžių kontekstą, pakeičia ją į atitinkamus garsus, remiantis būtent šiuo kontekstu. Šis metodas yra labai tinkamas programinei realizacijai, palyginus su kitais egzistuojančiais transkribavimo metodais. Žemiau pateikta metodo veikimo schema (žiūrėti 11 paveikslėlį) [5].



11 pav. Formalių taisyklių transkribavimo metodo schema.

Kairiuosius bei dešiniuosius kontekstus sudaro raidės prieš ir raidės po einamosios raidės, atitinkamai. Šių kontekstų tikslas yra nustatyti, kokios taisyklės turi būti pritaikytos einamajai raidei. Į papildomus veiksmus įeina du parametrai, vienas iš jų nurodo, per kiek tiriamojo žodžio raidžių pasislinkti einamąją taisyklę pritaikius, pradedant nuo einamosios raidės, o kitas - per kiek taisyklių pasislinkti ieškant kitos taisyklės praeitai netikus. Tačiau pastarieji du parametrai yra labiau aktualūs programiškai įgyvendinant šį metodą. Kuriamoje programoje bus įgyvendinta kontekstų susidedančių iš dviejų prieš ir trijų raidžių po einamosios raidės analizės. [5].

3.2. Pseudokodas

1 algoritmas. Lietuvinio algoritmas

```
TaisyklIDE[] = [{"", "", 'Ä', "B", "B", "", 3, "E", 1, 1},  
["", "", 'E', "I", "EAEIOUÖÜ", "", 3, "EJ", 2, 1],  
["", "", 'Ö', "B", "B", "", 3, "E", 1, 1], ...}
```

function lithuanizeDE(input, Gim)

var output;

var j = 0;

var i = 2;

input = ("__"+input.toUpperCase() + "____0 ");

while i < input.length + 1 **do**

 // netikusių taisyklių praleidimas

if (TaisyklIDE[j].ES != input[i]) **then**

 | j += TaisyklIDE[j].PoslT;

end

 // kontekstų tikrinimas, jis aptartas 3.3 skyrelyje

if (TaisyklIDE[j].KKont2!="" ? (TaisyklIDE[j].KKont2).indexOf(input[i-2])+1 : 1) AND
 ((TaisyklIDE[j].KKont1!="" ? (TaisyklIDE[j].KKont1).indexOf(input[i-1])+1 : 1) AND
 ((TaisyklIDE[j].DKont1!="" ? (TaisyklIDE[j].DKont1).indexOf(input[i+1])+1 : 1) AND
 ((TaisyklIDE[j].DKont2!="" ? (TaisyklIDE[j].DKont2).indexOf(input[i+2])+1 : 1) AND
 ((TaisyklIDE[j].DKont3!="" ? (TaisyklIDE[j].DKont3).indexOf(input[i+3])+1 : 1) AND
 (TaisyklIDE[j].Gim Gim)) **then**

 // taisyklių taikymas

 i += TaisyklIDE[j].PoslR;

 output = output.concat(TaisyklIDE[j].FonV);

 j = 0;

else

 // perėjimas prie sekančios taisyklės nepritaikius praeitos

 j++;

end

 // grįžimas prie pirmosios taisyklės praitejavus pro visas
 taisykles

if (j > TaisyklIDE.length - 1) **then**

 | j = 0; i++;

end

end

return output;

Algoritmą sudaro du pagrindiniai programiniai komponentai: masyvas, kuris kaskart paleidžiant programą užkrauna aprašytas transkribavimo taisykles ir metodas, kuris tas taisykles taiko, pavadinti jie atitinkamai *TaisyklIDE[]* ir *TranskrDE(input,gim)*. Algoritmo esmė yra kairiojo ir dešiniojo konteksto tikrinimas - 2 raidės į kairę ir 3 raidės į dešinę nuo einamosios raidės, tai aptarta 3.3 skyrelyje.

Visų pirma į metodo įvesties parametą paduodama įvesties reikšmė - vartotojo įvestas vokiškas asmenvardis. Visos raidės yra paverčiamos į didžiąsias, prie jos priekio pridedama "_", prie galo "____0" - tam, kad būtų galima atpažinti ir taikyti taisyklės žodžio pradžiai arba pabaigai pažymint kairįjį ar dešinįjį kontekstą simboliu "_". Pavyzdžiui, į algoritmą paduodamas asmenvardis **Baedeker** bus apdorojamas ir paverčiamas į "**__BAEDEKER__0**".

Paskui yra paleidžiamas *while* ciklas, kuris sukasi (*asmenvardžio ilgis*)+1 lygų kartų skaičių. Iš karto tikrinama, ar einamosios taisyklės (taisyklės elementą žymi kintamasis *j*) analizuojamoji raidė nėra lygi analizuojamojo žodžio einamajai raidei (jos poziciją nurodo kintamasis *i*), sąlygai esant teisingai kintamojo *j* reikšmė yra padidinama einamosios taisyklės *PosIT* reikšme (kad būtų pereita prie kitos taisyklės tikrinimo).

Pakitusi *j* reikšmė yra naudojama sekančiame *if* cikle, kur yra tikrinama taisyklė ir jos tinkamumas einamajai analizuojamojo žodžio raidei. Tikrinami kairieji ir dešinieji analizuojamojo žodžio kontekstai. Einamieji taisyklės kontekstai lyginami su *eil[i -+ x]*, kur *x* nurodo analizuojamojo žodžio konteksto raidės padėtį. Jei taisyklės ir analizuojamojo žodžio kontekstai bei giminė, kuriai taikoma taisyklė sutampa, ji yra pritaikoma asmenvardžiui, *i* reikšmė yra padidinama einamosios taisyklės *PosLR* reikšme (pasislenkama per analizuojamojo žodžio raides), prie *output* kintamojo yra pridedama einamosios taisyklės *FonV* reikšmė - su kiekviena šia teisinga sąlyga gaunama dalis asmenvardžio lietuviškos formos. Galiausiai *j* reikšmė yra nunulinama, kad sekančiai einamajai raidei būtų ieškoma tinkamos taisyklės pradedant nuo nulinio *TaisyklIDE[j]* elemento (pirmosios taisyklės). Žemiau pateiktoje 6 lentelėje aprašytos iteracijos bei kintamųjų kaita paleidžiant algoritmą su įvesties parametru **Müller** (iteracijos duomenys užfiksuoti radus tinkamą taisyklę).

6 lentelė. Vokiško asmevardžio *Müller* lietuvinimo proceso iteracijos

Iteracija	1	2	3	4	5
Gim	1	1	1	1	1
i	2	3	4	6	8
j	196	312	3	164	380
TaisyklIDE[j].KKont2					
TaisyklIDE[j].KKont1					LR
TaisyklIDE[j].ES	M	Ü	L	E	—
TaisyklIDE[j].DKont1		L	L	R	
TaisyklIDE[j].DKont2		L		—	
TaisyklIDE[j].DKont3					
TaisyklIDE[j].FonV	M	IU	L	ER	IS
TaisyklIDE[j].PosLR	1	1	2	2	1
TaisyklIDE[j].PosIT	1	10	1	13	3
output	M	MIU	MIUL	MIULER	MIULERIS

3.3. Kontekstų tikrinimas

Kontekstų tikrinimas nustato kontekstus ir randa taisyklę, kuri pritaikoma einamajai raidei. *If* sąlygoje yra paaimama konkreti taisyklė bei nuskaitomos jos kontekstų reikšmės, jos lyginamos su lietuvinamo žodžio einamosios raidės kairiuoju bei dešiniuoju kontekstais. Kontekstams ir giminei, kuriai skirta taisyklė, atitikus, sąlyginis kodas yra įvykdomas ir taisyklė pritaikoma.

3.4. Taisyklių aprašymas

Algoritme yra deklaruotas masyvas **TaisyklIDE[]**, talpinantis beveik 400 asmenvardžių sulietuvinimo taisyklių. Kiekviena taisyklė turi 10 parametrų, kurie nusako išankstines būtinas sąlygas ir posąlyginės elgsenos veiksmus, į ką turi būti keičiama einamoji raidė ar jų junginys, kokiai gimiinei (vyriškai, moteriškai ar abejoms) taisyklė yra taikoma, per kiek analizuojamojo žodžio raidžių pasislenkama ir kiek taisyklių praleidžiama taisyklę sėkmingai pritaikius. Parametrai eilės tvarka nurodyti 7 lentelėje.

7 lentelė. Taisyklės kintamieji

Kintamojo pavadinimas	Kintamojo reikšmė
<i>KKont2</i>	Dvi raidės į kairę nuo einamosios raidės
<i>KKont1</i>	Viena raidė į kairę nuo einamosios raidės
<i>ES</i>	Einamoji raidė
<i>DKont1</i>	Pirma raidė į dešinę nuo einamosios raidės
<i>DKont2</i>	Antra raidė į dešinę nuo einamosios raidės
<i>DKont3</i>	Trečia raidė į dešinę nuo einamosios raidės
<i>Gim</i>	Giminė, kuriai taikoma taisyklė
<i>FonV</i>	Raidė, į kurią keičiame einamąją raidę
<i>PoslR</i>	Per kiek raidžių pasislenkama, pritaikius taisyklę
<i>PoslT</i>	Per kiek taisyklių pasislenkama, pritaikius taisyklę

8 lentelė. Taisyklės kintamieji

Nr.	KKont2	KKont1	ES	DKont1	DKont2	DKont3	Gim	FonV	PoslR	PoslT
1.	"_"	"J"	'Ü'	"H"	"L"	""	3	"Ū"	1	1
2.	""	""	'E'	"B"	"B"	""	3	"E"	1	1
3.	""	"BCL"	"_"	""	""	""	1	"AS"	1	2

Pirmoji 8 lentelės taisyklė bus taikoma tik asmenvardžio pradžioje, jei einamoji raidė [Ü] iš kairės ir iš dešinės turės taisyklėje aprašytus kontekstus, ji bus pakeista į [Ū]. Pagal šią taisyklę **Jühlicher** virst į **Jülicheris**.

Pagal antrąją taisyklę **Ebbertz** virst į **Ebercas** - šiuo atveju balsis [E] yra trumpas, nes po jo eina "BB" (pagal 2.5.1 pateiktą taisyklę).

Trečiojoje taisyklėje kintamojo **KKont1** reikšmė yra raidžių masyvas. Jis visais atvejais reiškia tai, kad einamajame analizuojamo žodžio kontekste privalo būti raidė, priklausanti tam masyvui, kad ji galėtų būti pritaikyta. Tiksliau - jeigu lietuvinamas žodis baigiasi raide, priklausančia masyvui "BCL", tai lietuviškoje formoje bus pridėdama galūnė [as]. Pagal šią taisyklę asmenvardis **Marc** virsta į **Markas**.

3.4.1. Eiliškumo problema

Taisyklių masyve labai svarbų vaidmenį vaidina jų eiliškumas. Kai kurios taisyklės būtinai turi būti pritaikytos pirmiau nei kitos. Pirmiausiai atliekami sąlyginiai keitimai, aptarti 2.4.2 skyrelyje, toliau beveik visos taisyklės taikomos abėcėlės tvarka. Išimtytys taikomos situacijose, kai tenka pritaikyti kelias labai panašias taisykles - tarkime, kai dviejų taisyklių einamoji raidė yra ta pati,

dešinės pusės kontekstas nesiskiria, vienoje iš taisyklių nurodytas kairysis kontekstas, o kitoje ne - tokiu atveju prioritetas turi būti konkretesnei taisyklei (turinčiai daugiau kontekstų).

```

1 Taisyklės[0] = ("", "_", 'S', "P", "", "", 3, "ŠP", 2, 1);
2 Taisyklės[1] = ("", "", 'S', "P", "", "", 3, "SP", 2, 1);
3 Taisyklės[2] = ("", "", 'E', "I", "BCDFGHIJKLMNOPRTVZ", "", 3, "EI", 2, 1);
4 Taisyklės[3] = ("", "D", ' ', "", "", "", 3, "D", 1, 1);
5 Taisyklės[4] = ("", "E", ' ', "", "", "", 3, "E", 1, 1);
6 Taisyklės[5] = ("", "L", ' ', "", "", "", 3, "L", 1, 1);
7 Taisyklės[6] = ("", "LR", ' _', "", "", "", 3, "IS", 1, 1);

```

12 pav. Taisyklių prioritetizavimo pavyzdys

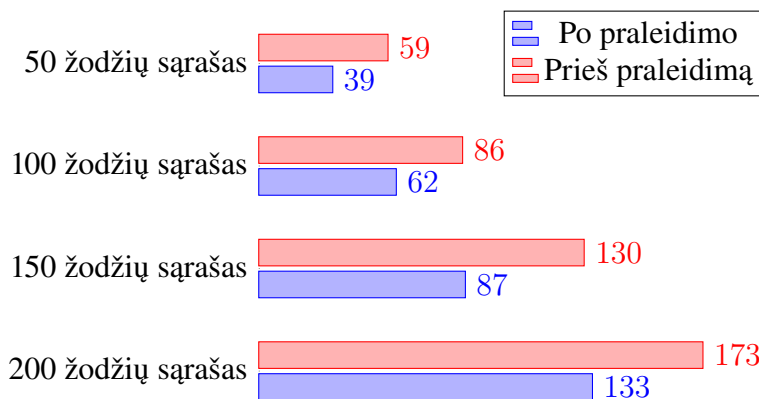
Pagal 12 pavyzdyje pavaizduotas taisykles ir jų eiliškumą, asmenvardis **Speidel** lietuviškoje formoje turėtų pavirsti į **Špeidelis**, tačiau apkeitus pirmąsias dvi taisykles vietomis ir jas pritaikius tas pats asmenvardis lietuviškoje formoje virstų į **Speidelis**, o tai nėra teisingas asmenvardžio atitikmuo lietuvių kalboje.

3.4.2. Našumo vertinimas

Tam, kad algoritmas veiktų sklandžiai reikėtų pagalvoti ir apie jo greitaveiką. Algoritmas iteruoja per taisykles (**asmenvardžio ilgis + 7**) (pagal 3.2 pateiktą algoritmo logiką) lygų kartų skaičių. Norint procesą pagreitinti reikia būdo sumažinti taisyklių skaičių arba būdo sumažinti tikrinamų taisyklių skaičių. Būtent dėl to kiekvienoje iš taisyklių yra nurodoma **PosIT** reikšmė.

3.4.3. Taisyklių praleidimas

Tai vienas iš būdų algoritmo greitaveikai padidinti, kuomet vienai raidei sėkmingai pritaikius taisyklę visos toliau einančios tos raidės taisyklės yra praleidžiamos. Šis metodas leidžia sumažinti tikrinamų taisyklių skaičių, atitinkamai didėja ir algoritmo veikimo greitis, nes nemaža dalis taisyklių yra tiesiog praleidžiama. Akivaizdų algoritmo pagreitėjimą galima pastebėti 13 pavyzdyje, iš jo galima spręsti, jog įgyvendinus nereikalingų taisyklių praleidimą algoritmas veikia apytiksliai 30 procentų greičiau ir algoritmo tikslumas nepakinta.



13 pav. Greitaveikos vertinimas prieš ir po taisyklių praleidimo (laikas milisekundėmis)

3.4.4. Taisyklių kiekio sumažinimas

Akivaizdus algoritmo greitimeikos pagreitinėjimas pasijustų sumažinus lietuvinimo taisyklių kiekį. Yra keletas būdų tai padaryti, vienas efektyviausių būdų įvedamą asmenvardį optimizuoti **[ä]**, **[ö]**, **[ü]** pakeičiant atitinkamai į **[ae]**, **[oe]**, **[ue]** [7].

Praktiniame darbe šis būdas nebus taikomas, nes dėl minėtų pakeitimų kai kurie asmenvardžiai bus sulietuvinami klaidingai, nes pasitaiko atveju, kai minėtų dvibalsių dvi raidės atsiduria skirtingose samplaikose, to pasekoje pritaikomos klaidingos taisyklės ir asmenvardis sulietuvinamas neteisingai. Norint įgyvendinti šį algoritmo pagreitinimo būdą minėtus dvibalsius reiktų pakeisti tik tuo atveju, jei jie priklauso tai pačiai samplaikai.

Įgyvendinus minėtų garsų sukeitimą ir turint asmenvardį **Feuerbach**, pagal suprogramuotas taisykles jis virstų į **Feiurbachas**, nors taisyklinga asmenvardžio lietuviška forma yra **Fojerbachas** (žiūrėti 14 ir 15 pavyzdžius).

```
1 ...
2 Taisyklės[0] = ("", "", 'F', "", "", "", 3, "F", 1, 1);
3 Taisyklės[1] = ("", "", 'E', "U", "", "", 3, "OJ", 1, 1);
4 Taisyklės[2] = ("", "", 'E', "", "", "", 3, "E", 1, 1);
5 Taisyklės[3] = ("", "", 'R', "", "", "", 3, "R", 1, 1);
6 Taisyklės[4] = ("", "", 'B', "", "", "", 3, "B", 1, 1);
7 Taisyklės[5] = ("", "", 'A', "", "", "", 3, "A", 1, 1);
8 Taisyklės[6] = ("", "", 'C', "H", "", "", 3, "CH", 1, 1);
9 Taisyklės[7] = ("", "H", '_', "", "", "", 1, "AS", 1, 7); //" Fojerbachas "
10 ...
```

14 pav. Taisyklių eiliškumas netaikant samplaikai keitimo

```
1 ...
2 Taisyklės[0] = ("", "", 'F', "", "", "", 3, "F", 1, 1);
3 Taisyklės[1] = ("", "", 'E', "", "", "", 3, "E", 1, 1);
4 Taisyklės[2] = ("", "", 'Ü', "R", "B", "", 3, "IU", 1, 1);
5 Taisyklės[3] = ("", "", 'R', "", "", "", 3, "R", 1, 1);
6 Taisyklės[4] = ("", "", 'B', "", "", "", 3, "B", 1, 1);
7 Taisyklės[5] = ("", "", 'A', "", "", "", 3, "A", 1, 1);
8 Taisyklės[6] = ("", "", 'C', "H", "", "", 3, "CH", 1, 1);
9 Taisyklės[7] = ("", "H", '_', "", "", "", 1, "AS", 1, 1); //" Feiurbachas "
10 ...
```

15 pav. Neteisingas taisyklių taikymas **[ä]**, **[ö]**, **[ü]** pakeitus į **[ae]**, **[oe]**, **[ue]**

Svarbu paminėti, kad vokiečių kalboje yra įprasta praktika kaitalioti **[ä]**, **[ö]**, **[ü]** atitinkamai į **[ae]**, **[oe]**, **[ue]** ir atvirkščiai, nepaisant to, kad algoritme tai nėra taikoma.

3.4.5. If - Else principas

Kitas būdas sumažinti taisyklių yra naudoti **If - Else** principu aprašant taisykles. Tai reiškia, kad aprašant tam tikrą sąlyginę taisyklę yra aprašoma tik viena iš dviejų taikymo sąlygų bei veiksmų jai esant teisingai, kitu atveju taisyklė pritaikoma be jokios sąlygos. Labiausiai šiam būdui tinka **[C]** raidei taikoma taisyklė. Žemiau pateiktame 16 pavyzdyje pašalinus antrąją taisyklę šis principas bus pritaikytas (taisyklę, kuriai taikomas principas galima rasti skaityti 2.4.2 skyrelyje). Svarbu paminėti, kad tai galima taikyti tik toms raidėms, kurios turi lygiai du taikymo atvejus. Pritaikius šį principą tikėtina, kad gali sumažėti algoritmo tikslumas.

```
1 ("", "", 'C', "EIY", "", "", 3, "C", 1, 1) // pašalinus principas taikomas
2 ("", "", 'C', "AOUPBTDFVSZJXLMNR", "", "", 3, "K", 1, 1)
3 ("", "", 'C', "", "", "", 3, "C", 1, 1)
```

16 pav. If - Else principas

4. Rezultatai

9 lentelė. Klaidingai sulietuvinti asmenvardžiai

Vokiškai	Lietuviškai	Išvestis
Köln	Kelnas	Kėlnas
Seghers	Zėgersas	Zegersas
Seghers	Zėgers	Zegers
Karlsruhe	Karlsruhė	Karlsruhė
Strigel	Štrigelis	Štrygelis
Jena	Jėna	Jena
Cyprian	Ciprianas	Cyprianas
Alzey	Alcėjus	Alcejus
Schwerin	Šverynas	Švėrynas
Ludwigshafen	Liūdvigshafenas	Liudvigshafenas
Störing	Šteringas	Štėringas

4.1. Klaidų analizė

Iš rezultatų matyti, kad daugelis klaidų yra susijusios su balsių ilgumu, tačiau to ir buvo galima tikėtis, nes kirčiuojamas visur buvo pirmasis skiemuo. Buvo susitaikyta, kad dėl šios išimties algoritmo tikslumas šiek tiek sumažės. Iš klaidų matyti, kad garsų **[ö]**, **[e]** ir **[i]** ilgumas kartais nustatomas klaidingai. Iš rezultatų akivaizdu, kad balsių **[a]**, **[o]** ilgumas visuomet nustatomas teisingai ir tai - logiška, nes lietuvių rašytinėje kalboje **[a]** ir **[o]** balsių ilgumas neatsispindi nors ilgosios tiek **[a]**, tiek **[o]** egzistuoja.

4.2. Klaidų atsiradimo priežastys

Formalios lietuvinimo taisyklės turi tam tikrų išimčių balsių ilgumo nustatyme. Tai galėjo padaryti įtaką daugeliui klaidingai sulietuvinėtų atvejų. Visų pirma, algoritme nebuvo taikyta 2.5.2 skyrelyje 3 lentelėje aprašyta penktoji taisyklė, kuomet kirčiuotame skiemenyje esanti balsė tampa ilga, kai tas kirčiuotas skienuo tampa atviru žodį kaitant. Šios taisyklės taikymui galima būtų turėti duomenų bazę, talpinančią giminingus žodžius, turinčius bendrą žodžio šaknį bei realizuojant algoritmą, kuris juos kaitaliotų bei nustatytų, ar egzistuoja bendrą šaknį turinčių žodžių, kuriuose ties tam tikra balse kirčiuotas skienuo tampa atviru. 2.5.1 skyrelyje 2 lentelės antrosios taisyklės netikslus pritaikymas taip pat turėjo įtakos lietuvinimo rezultatams. Algoritme ši taisyklė taikoma visuomet, kai po balsės eina keli bet kokie priebalsiai tokiu atveju darant prielaidą, kad jie priklauso tai pačiai morfemai. Dėl to tam tikra dalis atvejų galėjo būti verčiami klaidingai - pavyzdžiui, vietoj [ė] paliekama [e].

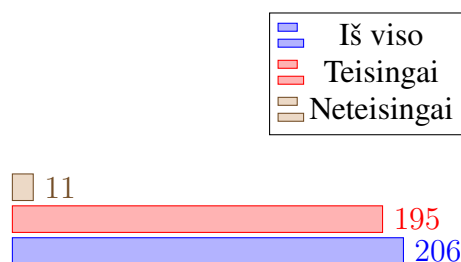
```
1 Wislmar -> Vismaras | [i] // nepriklauso tai pačiai morfemai  
2 Sternlberg -> Sternberg | [rn] // priklauso tai pačiai morfemai
```

17 pav. Priebalsių priklausymas tai pačiai morfemai

2.5.3 skyrelyje pavaizduota išimtis, kai balsė gali būti tiek ilga, tiek trumpa. Ši taisyklė algoritme visiškai nebuvo taikoma dėl pabrėžtinai daug išimčių ir netikslumų bei informacijos apie taisyklę šaltiniuose trūkumo.

4.3. Algoritmo tikslumo vertinimas

9 lentelėje pavaizduota dalis testinių asmenvardžių, iš viso jų yra 206, iš kurių net 195 sulietuvinata teisingai. Testiniai asmenvardžiai su vokiškomis ir lietuviškomis jų formomis imti iš knygos *Nelietuviškų tikrinių vardų rašymas „Lietuviškojoje tarybinėje enciklopedijoje“*, paskui lyginti su algoritmo išvesties rezultatais [8]. Pagal 18 duomenis, pasiektas algoritmo tikslumas yra 95 proc.



18 pav. Teisingai ir neteisingai sulietuvinėtų asmenvardžių skaičius

5. Programos grafinė sąsaja

Language: Gender:

Input: Text in Lithuanian:

No file selected.

19 pav. Programos grafinė sąsaja

Išvados ir rekomendacijos

Akivaizdu, kad lietuvinimo algoritmui buvo artimi tiek transkribavimo, tiek transliteravimo, tiek adaptavimo procesai. Žodžių transkribavimui skirtu metodu buvo paremtas pats algoritmo principas (kontekstai), tik tiriamas buvo ne žodžio tarimas, o tam tikrų rašytinių raidžių atitikmenys tarp dviejų kalbų, transliteruojant vokiški garsai buvo perteikiami lietuviška forma (sulietuvinami escetai ir raidės su umliautais), adaptuojamos buvo asmenvardžių galūnės.

Vokiečių kalboje egzistuoja daug įvairių vardų ir pavardžių - vokiškos ir ne vokiškos kilmės pavadinimų. Egzistuoja nemažai su asmenvardžių lietuvinimu susijusių išimčių nustatant balsių ilgumą, kurias teisingai suvaldyti yra gana sudėtinga. Visgi pastebėta, kad bene didžiausią įtaką algoritmo tikslumui turi taisyklių išdėstymas ir prioritetiškumas. Kadangi taisyklės stengtasi sudėlioti kuo praktiškesne ir teisingesne tvarka, algoritmo tikslumas remiantis testiniais duomenimis ir rezultatais pasiekė 95 proc. Taikant taisyklių kiekio sumažinimą arba nereikalingų taisyklių praleidimą, pagal darbe atlikto bandymo rezultatus, algoritmo sparta padidėjo 30 proc.

Norint algoritmą paversti tikslesniu ar galbūt visiškai tikslu rekomenduotina įgyvendinti papildomą programinį komponentą (metodą) nustatantį tikrąjį kirčiuotą skiemenį ir balsio ilgumą, atsisakyti idėjos besąlygiškai kirčiuoti pirmą skiemenį lietuvinant. Vokiško asmenvardžio balsių ilgumo nustatymui naudoti naujesnius ar patobulintus taisyklių rinkinius, kadangi algoritmo veikimas paremtas 1985 m. išleistais taisyklių rinkiniais, logiška, kad per 35 metus asmenvardžių lietuvinimo tradicijos galėjo ženkliai pakisti.

Ateities tyrimų planas

Pagrindinis tikslas ateityje yra kiek įmanoma labiau didinti algoritmo tikslumą. Visų pirma, norima papildyti lietuvinimo algoritmo funkcionalumą šalia sulietuvinto asmenvardžio pridedant ir fonetinę perrašą, kuri asmenvardžius perrašytų tarptautine fonetine abėcėle tam, kad naudotojai galėtų matyti ne tik asmenvardžio rašybą, kas jau yra įgyvendinta, tačiau ir tarimą. Taip pat norima įgyvendinti automatinį taisyklių generavimą, kuris pagal žinomus asmenvardžius ir jų sulietuvin-tas formas išvedinėtų ir išsaugotų teisingas lietuvinimo taisykles. Tarkime, žinoma, jog vokiško asmenvardžio **Schuster** lietuviškas variantas yra **Šusteris**. Tuomet, taisykles generuojantis algo-ritmas automatiškai sugeneruos ir pridės į masyvą taisyklę, reiškiančią, kad balsė **u**, prieš kurią eina junginys **[Sch]** bus laikoma trumpa. Įgyvendinus šiuos planus tikimasi akivaizdaus algorit-mo tikslumo padidėjimo - pakankamo, kad algoritmo rezultatais būtų galima visiškai pasitikėti ir vartoti jo sugeneruotus rezultatus publicistiniuose straipsniuose ir leidiniuose.

Literatūros šaltiniai

- [1] Adaptuotų asmenvardžių ir vietovardžių galūnių rašymas.
<http://www.vlkk.lt/aktualiausios-temos/svetimvardziai/adaptavimas>.
- [2] A. Drukteinis. Daugiau galimybių, bet ir daugiau sumanumo.
- [3] A. Drukteinis. Rekomendacija dėl autentiškų asmenvardžių gramatinimo.
<http://www.vlkk.lt/aktualiausios-temos/svetimvardziai/gramatinimas>.
- [4] L. Kalėdienė. Svetimžodžių perraša lietuviškame tekste.
- [5] P. Kasparaitis. Transcribing of the lithuanian text using formal rules. *Informatica*, 10(4):367–376, 1999.
- [6] A. Teišerskis. Ar raidės gali sunaikinti kalbą?
<http://pasauliolietuvis.lt/arunas-teiserskis-ar-raides-gali-sunaikinti-kalba/?print=pdf>.
- [7] Unicode. Characters and combining marks. http://unicode.org/faq/char_combmark.html#18.
- [8] Vyriausioji enciklopedijų redakcija. *Nelietuviškų tikrinių vardų rašymas „Lietuviškojoje tarybinėje enciklopedijoje“*. 1986.

Priedai

Dokumentą sudaro vienas priedas: A priede yra pavaizduoti asmenvardžiai, kurie buvo naudoti algoritmo testavimui. Raudona splava paryškinti asmenvardžiai sulietuvinti nesėkmingai, visi kiti - sėkmingai. Pirmas stulpelis nurodo žodžį vokiškai, o kitas - algoritmo sugeneruotą sulietuvinimą.

A. Testavimui naudotų asmenvardžių sąrašas

Vokiškai	Sulietuvinta
Baden	Badenas
Brahms	Bramsas
Läwitz	Lėvicas
Laeven	Lėvenas
Schäffer	Šeferis
Altranstädt	Altranštetas
Wächter	Vechteris
Kähler	Kėleris
Krähe	Krėhė
Maybach	Maibachas
Faißt	Faistas
Mayer	Majeris
Nay	Najus
Bräuning	Broiningas
Säuritz	Zoiricas
Bräuer	Brojeris
Brandenburg	Brandenburgas
Lübben	Liubenas
Celle	Celė
Cyprian	Cyprianas
Cainsdorf	Kainsdorfas
Cremer	Krėmeris
Marc	Markas
Beckmann	Bekmanas
Klopstock	Klopštokas
Bruch	Bruchas
Dachau	Dachau
Chemnitz	Chemnicas
Fuchs	Fuchsas
Ochsensaale	Ochsenzalė
Dessau	Desau
Hiddense	Hidenzė
Arndt	Arntas
Eichrodt	Eichrotas
Bremen	Brėmenas
Weser	Vėzeris
Engels	Engelsas
Leezen	Lėcenas
Behrens	Bėrensas
Weimar	Veimaras
Lorelei	Lorelei
Freyburg	Freiburgas
Meyr	Meiras
Beier	Bejeris
Steiermark	Štejermarkas
Alzey	Alcejus
Ley	Lėjus

Vokiškai	Sulietuvinta
Freudenstadt	Froidenštatas
Euler	Oileris
Feuerbach	Fojerbachas
Heuer	Hojeris
Frankfurt	Frankfurtas
Krefeld	Krėfeldas
Waffenrode	Vafenrodė
Hoffman	Hofmanas
Wolfgang	Volfgangas
Göggingen	Gegingenas
Elsnigk	Elsnikas
Seghers	Zegersas
Lohengrin	Lohengrynas
Karlsruhe	Karlsruhė
Jhering	Jėringas
Thälmann	Tėlmanas
Ibach	Ybachas
Widumann	Vydumanas
Schwerin	Švėrynas
Strigel	Štrygelis
Ingolstadt	Ingolštatas
Ribnitz	Ribnicas
Rainer	Raineris
Viersen	Fyrzenas
Riehl	Rylis
Nietzsche	Nyčė
Diesel	Dyzelis
Ihlenburg	Ylenburgas
Knihke	Knykė
Jena	Jena
Kamentz	Kamencas
Frank	Frankas
Bekker	Bekeris
Ekkehard	Ekehardas
Ludwigshafen	Liudvigshafenas
Luther	Liuteris
Langenfeld	Langenfeldas
Loburg	Loburgas
Halle	Halė
Keller	Keleris
Templin	Templynas
Kammhof	Kamhofas
Memmingen	Memingenas
Nürnberg	Niurnbergas
Nonnenwerth	Nonenvertas
Rostock	Rostokas
Mohr	Moras

Vokiškai	Sulietuvinta
Köthen	Kėtenas
Störing	Štėringas
Oeser	Ėzeris
Goethe	Gėtė
Görlitz	Gerlicas
Köln	Kėlnas
Böhme	Bėmė
Köhler	Kėleris
Oehme	Ėmė
Oidtweiler	Oitveileris
Nestroy	Nestrojus
Hoyerswerda	Hojsersverda
Penne	Penė
Zeppelin	Cepelynas
Krupp	Krupas
Pfalz	Pfalcas
Pforzheim	Pforcheimas
Rumpf	Rumpfas
Philippi	Filipis
Nymphenburg	Nimfenburgas
Stoph	Štofas
Quedlinburg	Kvėdlinburgas
Queller	Kveleris
Mur	Mūras
Rhein	Reinas
Sparrenbusch	Šparenbušas
Alfons	Alfonsas
Dresden	Dresdenas
Wiesbaden	Vysbadenas
Wismar	Vismaras
Saale	Zalė
Mosel	Mozelis
Elsa	Elza
Schiller	Šileris
Büschdorf	Biušdorfas
Spandau	Špandau
Speidel	Špeidelis
Kassenbohm	Kasenbomas
Lessing	Lesingas
Meißen	Meisenas
Pößneck	Pesnekas
Strauß	Štrausas
Staffelstein	Štafelšteinas
Teltow	Teltovas
Todt	Totas
Potsdam	Potsdamas
Zittau	Citau

Vokiškai	Sulietuvinta
Thalheim	Talheimas
Bützow	Biucovas
Lortzing	Lorcingas
Götnitz	Getnicas
Tschirma	Čirma
Bartsch	Barčas
Brauchitsch	Brauchičas
Gruber	Grūberis
Ruderer	Rūdereris
Burg	Burgas
Luckenwalde	Liukenvaldė
Kufstein	Kufšteinas
Überlingen	Yberlingenas
Uebermann	Ybermanas
Üchtland	Ichtlandas
Ueckermünde	Ikermiundė
Jülich	Jūlichas
Jüterbog	Jüterbogas
Jünger	Jungeris
Lübeck	Liūbekas
Grünewald	Griūnevaldas

Vokiškai	Sulietuvinta
Günzburg	Giuncburgas
Müller	Miuleris
Uhlberg	Ūlbergas
Uhland	Ūlandas
Huhn	Hūnas
Ührchen	Yrchenas
Dühring	Diūringas
Kühne	Kiūnė
Vogtland	Fogtlandas
Havel	Hafelis
Gustav	Gustavas
Wurzen	Vurcenas
Ewe	Ėvė
Pasewalk	Pazevalkas
Hexenberg	Heksenbergas
Marx	Marksas
Mylau	Mylau
Naunyn	Naunynas
Meyer	Mejeris
Ay	Ajus
Ypsheim	Ipsheimas

Vokiškai	Sulietuvinta
Zack	Cakas
Zürich	Ciūrichas
Würzburg	Viurcburgas
Zschokke	Čokė
Zschopau	Čopau
Aare	Arė
Celle	Celė
Göhrde	Gėrdė
Hiddense	Hidenzė
Zella	Cela
Seghers	Zegers
Saale	Zalė
Agatha	Agata
Blume	Bliūmė
Oehme	Ėmė
Penne	Penė
Ochsensaale	Ochsenzalė
Knihke	Knykė
Halle	Halė
Luckenwalde	Liukenvaldė
Ueckermünde	Ikermiundė
Kühne	Kiūnė
Zschokke	Liukenvaldė