LATVIJAS UNIVERSITĀTE
FIZIKAS UN MATEMĀTIKAS FAKULTĀTE
DATORIKAS NODAĻA

Latviešu valodas vārdformu sintezators

BAKALAURA DARBS

Autors:	Ivars Štrāls
Stud. Apl. Nr.	DatZ 040079
Vadītājs:	Normunds Grūzītis
	Mg.sc.comp.

Rīga, 2008

ANOTĀCIJA

Darbā ir apskatītas latviešu valodas vārdformu sintēzes īpatnības un būtiskākās problēmas. Tāpat tiek aplūkoti jau izstrādāti risinājumi vārdformu darināšanai un vārdu analīzei. Darba rezultātā ir izveidots vārdformu sintēzes modulis, kas tiek balstīts uz iepriekš izstrādāta rīka vārdu morfoloģiskai analīzei un spēj locīt visu vārdšķiru vārdus.

Papildus tam tika arī apskatīti latviešu valodas vārdu sintēzē eksistējoši izņēmumi. To apstrādei bija nepieciešami un arī tika izstrādāti gan datubāzes struktūra, gan efektīvs algoritms šādu vārdu locīšanai. Rezultātā paši lietotāji var papildināt izņēmumus, ja kāds vārds netiek locīts atbilstoši sagaidāmajam rezultātam.

ABSTRACT

This paper explores features and main problems of Latvian language's word form synthesis. Also already developed systems of word form synthesis and word analysis are examined. A synthesizer module which is supported by already developed module of word morphological analysis and can process words of all lexical categories is built as a result of this work.

Additionally exceptions which exist in word form synthesis of Latvian language were explored in this paper too. There were developed both database structure and efficient algorithm for processing such exceptions. Users now can add new exceptions if there are any problems with word form synthesis.

SATURS

IEVADS

Latviešu valoda ir samērā maza valoda, līdz ar to ir maz izstrādātu rīku un sistēmu tās vārdu analīzei un sintēzei. Kā viena no sintēzes sastāvdaļām ir vārdformu darināšana jeb vārda locīšana – tā tad arī tiks sīkāk apskatīta šajā darbā.

Vārdformu sintēze izpaužas kā galotņu piekārtošana vārdiem atkarībā no noteiktiem latviešu valodas likumiem, kuri tiks apskatīti darba sākumā. Tiks apskatīti arī iepriekš veidotie rīki vārdu locīšanai. Kā viena no šādu rīku daļām ir vārdu analīze, jo tā ir nepieciešama, lai pareizi izlocītu vārdu. Autora izstrādātais sintezators ir kā papildinājums uz leksikonu bāzētam vārdu morfoloģiskajam analizatoram, kas ir izstrādāts LU MII semantiskā tīmekļa projekta Kamols ietvaros.

Darba galvenais mērķis tad arī bija izstrādāt ar iepriekš pieminēto analizatoru saderīgu vārdformu sintezatoru, kas, balstoties uz vārda analīzē iegūtās informācijas, spētu pareizi to izlocīt. Tādējādi darbā tiks apskatīts gan šis iepriekš izstrādātais vārdu analizators, gan arī locītāja sadarbība ar to.

Šāda rīka nepieciešamība slēpjas tajā apstāklī, ka tādējādi ir iespējams būtiski samazināt vārdnīcas apjomu, tajā glabājot tikai vārdu pamatformas ar nepieciešamo informāciju pareizai vārdformu darināšanai. Tādējādi šādi var pārbaudīt, vai vārdam eksistē noteikta vārdforma – tas noder pārbaudot vārdu pareizrakstību, lai nepieļautu nekorekta locījuma izmantošanu. Pašlaik šim mērķim pārsvarā izmanto pilnās vārdnīcas pieeju – visi vārdi un to vārdformas tiek glabāti vārdnīcā.

Vēl šāds rīks noderētu dažādos meklētājos – gan interneta, gan arī lokālos, gan dažādās datu bāzēs. Tādējādi ar tā palīdzību būtu iespējams meklēt ne tikai ievadīto vārdu, bet arī dažādus tā locījumus, kas būtiski atvieglotu lietotāja ikdienu, jo nebūtu vairs jādomā, pēc kādas vārdformas vislabāk meklēt.

Latviešu valodā eksistē salīdzinoši daudz izņēmumu, kas tiek locīti savādāk kā pārējie līdzīgie vārdi. Līdz šim šādi gadījumi tika vai nu iestrādāti algoritmos, vai arī paredzēti datubāzē, bet abas šīs metodes nav pietiekami universālas. Līdz ar to kā viens no darba uzdevumiem bija izstrādāt elastīgu risinājumu tieši šo izņēmumu apstrādei, lai tos varētu pievienot ne tikai rīka izstrādātāji. Tas ir nepieciešams tāpēc, ka latviešu valodā jaunu vārdu parādīšanās ir izplatīta parādība, līdz ar to arī var parādīties papildus izņēmumi, kuri tad būtu jāpievieno sarakstam.

Vārdformu sintēzes rīks ir izstrādāts JAVA programmēšanas valodā, līdz ar to ir iespējams to lietot neatkarīgi no operētājsistēmas – sākot ar personālajiem datoriem un beidzot ar mobilajiem telefoniem. Uz pēdējiem īpaši attiecas nosacījums par mazu vārdnīcas lielumu, jo to atmiņas daudzums vēl aizvien ir salīdzinoši mazs.

1. TEORĒTISKAIS PAMATOJUMS

Latviešu valoda pieder indoeiropiešu valodas saimes baltu valodu grupai. Tā ir sarežģīta valoda, salīdzinot ar vienu no izplatītākajām valodām pasaulē – angļu –, jo vārdiem eksistē daudz vairāk locījumu. Nodaļas turpinājumā tiks apskatīts vārdu iedalījums un to īpašības, kas ir būtisks pareizai vārdformu sintēzei.

1.1. Vārdšķiras

Svarīgākais vārdu iedalījums vārdformu sintēzei ir vārdšķiras – pēc tām tad arī nosaka, kā katrs konkrētas vārdšķiras ir jāloka. Pavisam latviešu valodā ir 10 vārdšķiras, bet lokāmas ir tikai daļa no tām. Vispirms apskatīsim šīs lokāmās vārdšķiras:

- Lietvārdi (daļa ir nelokāmi),
- Īpašības vārdi,
- Vietniekvārdi,
- Darbības vārdi,
- Skaitļa vārdi (daļa ir nelokāmi).

Šīs 5 vārdšķiras tad arī ir tās, kuras tiks sīkāk apskatītas darba turpinājumā. To locīšana būtiski atšķiras savā starpā, līdz ar to arī izmantotie vārdformu darināšanas algoritmi ir dažādi.

Pārējās 5 nelokāmās vārdšķiras ir šādas:

- Saikļi,
- Prievārdi,
- Apstākļa vārdi,
- Partikulas,
- Izsauksmes vārdi.

Tādējādi šo vārdu locīšana nav iespējama, kas arī būtu jāparedz veidojot rīku vārdu locīšanai. Pretējā gadījumā var rasties situācija, ka tiek darinātas nekorektas vārdformas.

1.2. Sīkāks vārdšķiru iedalījums

Lai gan vārda piederība kādai konkrētai vārdšķirai lielā mērā nosaka to, kā šis vārds tiks locīts, tā tomēr nav pietiekams nosacījums korektai vārdformu sintēzei. Tāpēc ir nepieciešams vēl detalizētāks iedalījums, kas apvienotu vārdus ar vienādām locīšanas īpašībām. Tādējādi nepieciešams apskatīt visu šo piecu lokāmo vārdšķiru sīkāku sadalījumu un galvenās īpašības, ko pēc tam izmantotu sintēzes algoritmos.

1.2.1. Lietvārdi

Lietvārdi ir būtiskākā vārdšķira, jo tie ir visizplatītākie latviešu valodā. Tas ir tāpēc, ka lietvārdi teikumos var parādīties jebkura teikuma locekļa vietā, tādējādi to izmantojums ir ļoti plašs. Tātad ir svarīgi izstrādāt pilnīgus algoritmus tieši lietvārdu locīšanai.

Kopā lietvārdiem ir septiņi locījumi, no kuriem divi ir vienādi – vienskaitļa Akuzatīvs un Instrumentālis un daudzskaitļa Datīvs un Instrumentālis, atšķiras tikai ar vārdu *ar*, tāpēc turpmāk darbā netiks aplūkota Instrumentāļa vārdforma. 1.1 tabulā var redzēt visus šos locījumus, kā arī to, kā ir izlocīts lietvārds *kāja*.

1.1. tabula **Lietvārda kāja locīšana**

Locījums	Vienskaitlis	Daudzskaitlis
Nominatīvs (Kas?)	kāja	kājas
Ģenitīvs (Kā?)	kājas	kāju
Datīvs (Kam?)	kājai	kājām
Akuzatīvs (Ko?)	kāju	kājas
Instrumentālis (Ar ko?)	ar kāju	ar kājām
Lokatīvs (Kur?)	kājā	kājās
Vokatīvs (bez jautājuma)	kāja!	kājas!

Viens no šīs vārdšķiras iedalījumiem ir sadalīšana sugas vārdos un īpašvārdos, bet vārdformu sintēzei tas nav pietiekams. Vēl lietvārdus iedala pa dzimtēm – sieviešu un vīriešu – un skaitļiem – vienskaitlis un daudzskaitlis -, bet arī šāds sadalījums ir nepilnīgs. Savukārt, dalījums deklinācijās, kas daļēji ietver sevi arī dalījumu pa dzimtēm, ir piemērots, jo apvieno vārdus ar ļoti līdzīgām īpašībām un līdz ar to arī to vārdformu darināšana ir līdzīga.

Pavisam izšķir sešas deklinācijas [1, 5.-9. lpp.]:

1. deklinācija – vīriešu dzimtes lietvārdi, kas vienskaitļa nominatīvā beidzas vai nu ar galotni *–s*, vai ar *–š* (piemēram, *koks, mežs,*);
2. deklinācija – vīriešu dzimtes lietvārdi
    - kas vienskaitļa nominatīvā beidzas ar galotni *–is* (*lācis*),
    - kam vienskaitļa ģenitīvs un nominatīvs ir vienāds (*akmens, zibens*),
    - kā arī vārds *suns*;
3. deklinācija – vīriešu dzimtes lietvārdi, kas vienskaitļa nominatīvā beidzas ar galotni *–us* (*medus, ledus*);
4. deklinācija – sieviešu dzimtes lietvārdi
    - kas vienskaitļa nominatīvā beidzas ar galotni *–a* (*meita, kleita*),
    - vīriešu dzimtes lietvārdi, kas beidzas vienskaitļa nominatīvā arī beidzas ar galotni *–a* (*puika*);

5. deklinācija – sieviešu dzimtes lietvārdi
    - kas vienskaitļa nominatīvā beidzas ar galotni –*e* (*egle, ēvele*),
    - vīriešu dzimtes lietvārdi, kas beidzas vienskaitļa nominatīvā arī beidzas ar galotni –*e* (*Egle*);
6. deklinācija – sieviešu dzimtes lietvārdi, kas beidzas ar galotni –*s*.

Vēl eksistē arī vārdi, kas parasti tiek lietoti vienskaitlī, jeb vienskaitlinieki – *maize, malka* -, kā arī attiecīgi daudzskaitlinieki – *putraimi, kvieši*. Tādējādi šo vārdu locīšanā jāievēro, ka nedrīkst tikt veidotas attiecīgi daudzskaitļa un vienskaitļa vārdformas.

1.2.2. Darbības vārdi

Darbības vārdi ir vēl viena būtiska vārdšķira latviešu valodā, jo gandrīz katrā teikumā kā izteicējs tiek pielietots darbības vārds. Tādējādi arī bez šīs vārdšķiras sintēzes nav iespējams iztikt. Tāpat kā lietvārdiem arī darbības vārdus iedala pa skaitļiem – vienskaitlis un daudzskaitlis, vēl klāt nāk iedalījums pa personām – pirmā, otrā un trešā persona –, kā arī sešos laikos – vienkāršā tagadne, vienkāršā pagātne, vienkāršā nākotne, saliktā tagadne, saliktā pagātne un saliktā nākotne.

Darbības vārdi ir visgrūtāk lokāmi vārdi latviešu valodā, jo vārdformu darināšanai jāņem vērā pat trīs celmi, kamēr citām vārdšķirām pilnīgi pietiek tikai ar vienu. Šie celmi tad ir šādi [2, 72.-73. lpp.]:

- nenoteiksmes celms – to iegūst, atmetot nenoteiksmes galotni –t vai -ties: mes-t, mes-ties, runā-t, runā-ties. No nenoteiksmes celma veido
    - īstenības izteiksmes nākotnes formu – met-īšu, runā-šu,
    - atstāstījuma izteiksmes nākotnes formu – met-īšot, runā-šot,
    - vēlējuma izteiksmes formu – mes-tu, runā-tu,
    - divdabjus ar -dams, -ts –mez-dams, runā-dams,
- tagadnes celms – to iegūst, atmetot tagadnes vienskaitļa 1. personas galotni -u vai daudzskaitļa 1. personas galotni -am (-ām): met-u (met-am), runāj-u (runāj-am). No tagadnes celma veido:
    - īstenības izteiksmes tagadnes formas – met-u, met, met, met-am, met-at; runāj-u, runā, runā, runāj-am, runāj-at;
    - pavēles izteiksmes formas – met, lai met, met-iet; runā, lai runā, runāj-iet;
    - vajadzības izteiksmes formu – jāmet, jā-runā,
    - atstāstījuma izteiksmes tagadnes formas – met-ot, runāj-ot,
    - divdabjus
        - ar -ošs – runāj-ošs,

- ar -ams, -āms – met-ams, runāj-ams,
- ar -am, -ām – met-am, runāj-am,
- ar -ot – met-ot, runāj-ot.

- pagātnes celms – to iegūst atmetot pagātnes vienskaitļa 1. personas galotni -u vai daudzskaitļa 1. personas galotni -ām: met-u (met-ām), runāj-u (runāj-ām). No pagātnes celma veido:
  - īstenības izteiksmes pagātnes formas – met-u, met-i, met-a, met-ām, met-āt; runāj-u, runāj-i, runāj-a, runāj-ām, runāj-āt;
  - divdabi ar -is – met-is, runāj-is.

Vārdformu darināšanu sarežģī arī tas, ka šai vārdšķirai ir darbības vārdu izteiksmes, kas atspoguļo darbības attiecinājumu pret īstenību. Tādējādi tās visas ir jāloka tikai tajos laikos, kas tām var eksistēt. Pavisam ir piecas šādas izteiksmes [2, 70.-72. lpp.]:

a) īstenības izteiksme – tā izsaka, ka darbība patiešām notiek, notika vai notiks, tātad šai izteiksmei ir iespējami visi seši laiki;
b) pavēles izteiksme – tā izsaka rosinājumu (aicinājumu, lūgumu, pavēli), šai izteiksmei savukārt nav laiku un tās formām ir tagadnes nozīme;
c) vēlējuma izteiksme – tā izskata vēlēšanos, lai darbība tiktu īstenota, vai arī norāda, ka darbība ir iespējama, šai izteiksmei ir divi laiki – vienkāršā tagadne un saliktā tagadne;
d) atstāstījuma izteiksme – tā izsaka darbību, par kuru runātājs neatbild, šai izteiksmei ir četri laiki – vienkārša tagadne, vienkārša nākotne, saliktā tagadne un saliktā nākotne;
e) vajadzības izteiksme – tā izskata vajadzību (nepieciešamību) īstenot darbību, šai izteiksmei tāpat kā īstenības ir iespējami visi seši laiki.

Līdzīgi kā lietvārdi arī darbības vārdi iedalās lielās grupās ar līdzīgām īpašībām – šajā gadījumā tās ir vārdu formu maiņas laikos, personās un skaitļos. Šādu maiņu sauc par konjugāciju, to darbības vārdam var noteikt pēc iepriekš minētajām 3 pamatformām – nenoteiksmes, tagadnes vienskaitļa 1. vai 3. personas formas un pagātnes vienskaitļa 1. vai 3. personas formas. Pavisam ir 3 konjugācijas [2,73.-76. lpp.]:

- pirmā konjugācija – pie tās pieder darbības vārdi, kam nenoteiksmē, tagadnes un pagātnes celmā ir tikai viena zilbe (neskaitot priedēkli), piemēram, *mest, aizmest, likt, glābt*;
- otrā konjugācija – pie tās pieder darbības vārdi ar piedēkļiem *–ā-, -ē-, -o-, -ī-, -alē-, -aļā-, -elē-, -uļo-*, kam tagadnes un pagātnes celmā ir vienāds zilbju skaits, bet nenoteiksmē ir vismaz 2 zilbes, piemēram, *runāt, rīkot*;

- trešā konjugācija – pie tās pieder darbības vārdi ar piedēkļiem –ā-, -ē-, -ī-, -inā-, kam nenoteiksmē ir vismaz 2 zilbes un pagātnes celmā par vienu zilbi vairāk nekā tagadnes celmā, piemēram, *vēdināt, gaidīt*.

Darbības vārdi konjugācijās iedalās vēl sīkāk – grupās –, bet to neapskatīsim, jo tas nav nepieciešams vārdformu sintēzei. Savukārt, trīs darbības vārdi – *būt, iet, dot* – neiederas nevienā konjugācijā.

1.2.3. Īpašības vārdi

Tāpat kā lietvārdus arī īpašības vārdus iedala pēc skaitļa un dzimtes. Papildus tam īpašības vārdiem var būt divu veidu galotnes – nenoteiktā un noteiktā. Piemēram, īpašības vārdi ar nenoteikto galotni *zaļš koks, zaļa zāle* (atkarībā no dzimtes), bet ar noteikto – attiecīgi *zaļais koks, zaļā zāle*.

Arī locījumi ir tie paši septiņi – nominatīvs, ģenitīvs, datīvs, akuzatīvs, instrumentālis, lokatīvs un vokatīvs. Īpašības vārdus ar nenoteikto galotni loka tāpat kā I deklinācijas lietvārdus (ja īpašības vārds ir vīriešu dzimtē) vai arī kā IV deklinācijas lietvārdus (sieviešu dzimtē), kā arī tiem neeksistē vokatīva locījums. Savukārt, šīs vārdšķiras vārdus ar noteikto galotni loka īpatnēji – tiem ir paplašinātā galotne vienskaitļa datīvā un lokatīvā, daudzskaitļa datīvā un lokatīvā, kā arī eksistē vokatīva locījums [3].

Īpašības vārdiem ir salīdzināmās pakāpes – pamata, pārākā un vispārākā. Pārākajā pakāpē tiek pievienots piedēklis *-āk-*, piemēram, *zaļš* pārākajā pakāpē būs *zaļāks*. Savukārt, vispārākajā pakāpē tiek pievienota morfēma *vis* pārākās pakāpes priekšā, piemēram, *zaļākais* būs *viszaļākais*. Jāievēro, ka vispārākā pakāpe neeksistē īpašības vārdiem ar nenoteikto galotni. Tātad, veidojot īpašības vārdu vārdformas, ir jāsintezē arī šīs salīdzināmās pakāpes. Tabulā Nr. 1.2. var apskatīt vārda *liels* visu pakāpju un dzimtu vārdformas vienskaitļa Nominatīvā.

1.2. tabula **Īpašības vārda liels Nominatīva vārdformas**

Pakāpe	Nenoteiktā galotne		Noteiktā galotne	
	vīriešu dzimte	sieviešu dzimte	vīriešu dzimte	sieviešu dzimte
Pamata	liels	liela	lielais	lielā
Pārākā	lielāks	lielāka	lielākais	lielākā
Vispārākā	neeksistē	neeksistē	vislielākais	vislielākā

1.2.4. Skaitļa vārdi

Skaitļa vārdu locīšana ir līdzīga īpašības vārdu locīšanai – tāpat tos iedala skaitļos – vienskaitlī un daudzskaitlī –, kā arī dzimtēs – sieviešu un vīriešu. Atšķirībā no iepriekš pieminētās vārdšķiras skaitļa vārdi iedalās pamata un kārtas skaitļos, tātad nav sadalījuma pakāpēs (pārākā, vispārākā). Attiecīgi pamata skaitļa vārdi, piemēram, ir *viens, pieci, deviņi, abi*, bet kārtas – *pirmais, otrais, piektais*.

Pamata skaitļa vārdus tāpat kā īpašības vārdus ar nenoteikto galotni vīriešu dzimtē tos loka kā I deklinācijas lietvārdus un sieviešu dzimtē – kā IV deklinācijas lietvārdus. Savukārt, kārtas skaitļa vārdformas sintezē līdzīgi īpašības vārdiem ar noteikto galotni, tiesa, neeksistē vokatīva locījums nevienā no abiem skaitļiem. Izņēmums ir vārds trīs, ko loka īpatnēji, kā arī vārds tūkstotis ir jāloka kā II deklinācijas lietvārds. Tāpat ir jāievēro, ka daļa skaitļa vārdu ir nelokāmi, piemēram, *desmit, simt* un visi citi, kuriem nav galotnes. [2, 48.-54.lpp]

1.2.5. Vietniekvārdi

Vietniekvārdus tāpat kā lietvārdus iedala pēc dzimtes un skaitļa, līdz ar to arī locīšana ir līdzīga lietvārdu vārdformu darināšanai. Vienīgi vietniekvārdiem nav vokatīva locījums, tādējādi ir par vienu locījumu mazāk.

Lielāko daļu vietniekvārdu dala kā I vai IV deklinācijas lietvārdus (atkarībā no dzimtes). Savukārt, dažus loka īpatnēji, piemēram, *es, tu, mēs, jūs, sevis, kas, tas, šis, pats*. Kā arī dažiem vietniekvārdiem iespējama nenoteiktā un noteiktā galotne, piemēram, *tavējs – tavējais, savējs – savējais*, līdz ar to šos vārdus loka līdzīgi kā pamata īpašības vārdus ar noteikto vai nenoteikto galotni [2, 59.-60. lpp.].

1.3.Mijas

Līdzskaņu mija ir celma pārmaiņas, kas rodas darinot noteikta locījuma vārdformas. Tās galotnes priekšā ir II, V un VI deklinācijas lietvārdiem. II deklinācijai mijas ir vienskaitļa ģenitīvā un visos daudzskaitļa locījumos, bet V un VI deklinācijai tās ir tikai daudzskaitļa ģenitīvā [2, 18.-20. lpp.]. Tabulā Nr. 1.3 var aplūkot, kādas mijas var būt attiecīgajās deklinācijās.

Jāievēro, ka daļai vārdu šāda mija nav, lai gan beidzas ar attiecīgo līdzskani. Piemēram, *viesis – viesa, mute – mutu*. Šādi gadījumi darba turpinājumā tiks apskatīti kā izņēmumi.

1.3. tabula **Lietvārdu līdzskaņu mijas**

Celma beigu līdzskanis (-ņi)	Mija	Deklinācija	Piemērs
b	bj	II, V	gulbis – gulbja
c	č	II, V	lucis – luča
d	ž	II, V	briedis – brieža
l	ļ	II, V, VI	brālis – brāļa
m	mj	II, V	kurmis – kurmja
n	ņ	II, V, VI	līnis – līņa
p	pj	II, V	ūpis – ūpja
s	š	II, V, VI	lasis – laša
t	š	II, V, VI	zutis – zuša
v	vj	II, V, VI	cirvis – cirvja
z	ž	II, V, VI	āzis – āža
sn	šņ	II, V, VI	slieksnis – sliekšņa
zn	žņ	II, V	lauznis – laužņa
sl	šļ	II	kāpslis – kāpšļa
zl	žļ	II	zizlis – zižļa
ln	ļņ	II	alnis – aļņa
ll	ļļ	II, V	zellis – zeļļa
kst	kš	V, VI	pāksts – pākšu

2. ESOŠĀ SITUĀCIJA

Vārdformu sintēze latviešu valodai kā jau mazai valodai nav daudz pētīta. Vairāk uzmanības tiek veltīta gan teksta analīzei, gan vārdu morfoloģiskajai analīzei, tāpēc bieži vien vārdu locīšana ir tapusi tikai kā papildinājums pārējai sistēmai.

Lielākajā daļā šādu sistēmu tiek pielietota viena no divām pieejām – pilnā vārdnīca vai pamatformu vārdnīca. Pirmajā gadījumā tiek glabāt visi vārdi un to vārdformas, līdz ar to sintēze nav nekas vairāk, kā lokāmā vārda un attiecīgi tā locījumu atrašana. Veiktspējas ziņā šāds risinājums ir ļoti ātrs, jo ir jāveic tikai meklēšana. Tā kā tiek glabātas pilnīgi visas vārdformas, tad vārdnīca var kļūt ļoti apjomīga, un tas var novest pie būtiska ātrdarbības krituma vārda meklēšanā.

Savukārt, otra pieeja ir glabāt tikai vārdu pamatformas un pārējās vārdformas darināt ar likumu palīdzību. Tā kā šajā gadījumā tiek glabāta vien vārda pamatforma un informācija par tā locīšanu, tad vārdnīca ir daudz mazāka nekā pilnās vārdnīcas gadījumā. Šādas pieejas lielākais ieguvums ir tāds, ka, papildinot vārdnīcu, ir jāpievieno tikai vārds tā pamatformā, nevis katra vārdforma kā iepriekšējā pieejā.

2.1. Izstrādātas sistēmas vārdformu sintēzei

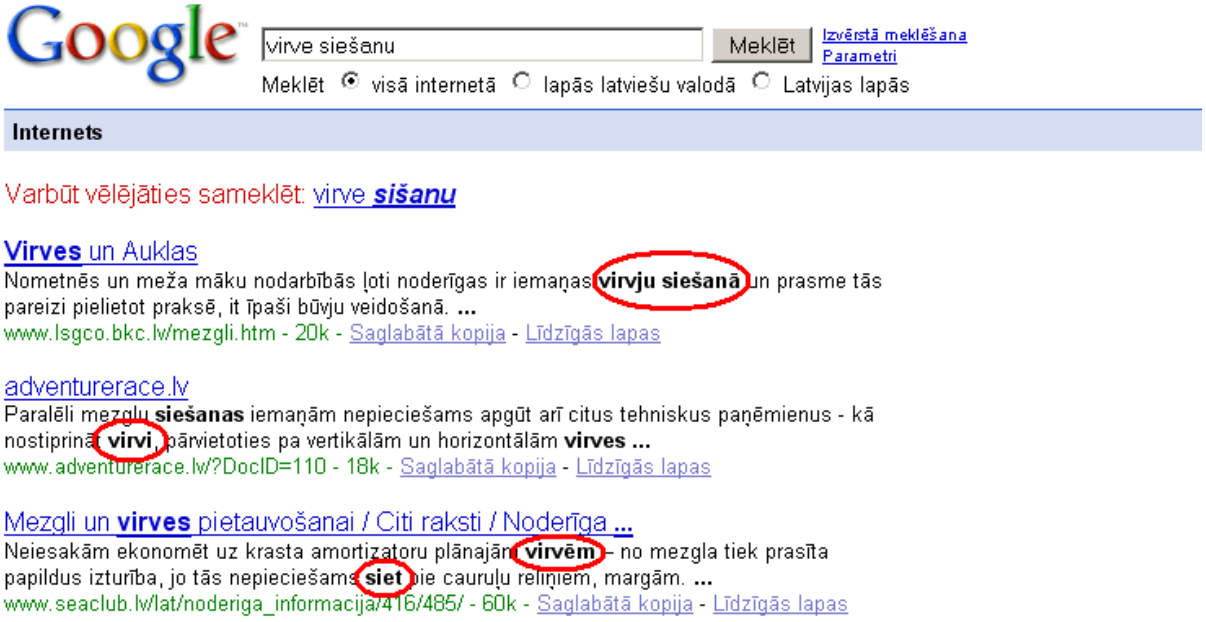
Lielāka daļa šādu sistēmu vai rīku ir izstrādātas Latvijas Universitātes Matemātikas un informātikas institūta Mākslīgā intelekta laboratorijā. Kā pirmo var minēt Anda Cirša izveidoto vārdu locīšanas sistēmu, kas tapusi sadarbībā ar Ingunu Greitāni. Tajā bija iekļauta lietvārdu, darbības vārdu. divdabju un arī īpašības vārdu locīšana. Tā kā vārdi bija iedalīti tikai vārdšķirās, tad pareizai vārdformu sintēzei bija nepieciešama apjomīga datubāze, kas norādītu, kā katrs vārds būtu jāloka.

Kā nākošo var minēt Uģa Sarkana izstrādāto sistēmu „LatvDict", kas bija gan skaidrojošā vārdnīca, gan arī vārdformu sintezators. Šajā gadījumā tika izmantota pilnās vārdnīcas metodes pieeja, jo visi locījumi tika glabāti datubāzē, kas, protams, būtiski, palielina tās apjomu un līdz ar to tās uzturēšana ir laikietilpīga.

Pēdējo šādu sistēmu 2002. gadā sāka izstrādāt Kārlis Goba. Tā atbalstīja daļēju lietvārdu un darbības vārdu sintēzi – netika ņemti vērā izņēmumi, kā arī ne visas deklinācijas un konjugācijas tika locītas. Pēc tam šo sistēmu uzlaboja Mārcis Pinnis [4]. Tika pilnveidota lietvārdu un darbības vārdu vārdformu sintēze, tika pievienota daļēja izņēmumu apstrāde, kā arī papildus izveidoti algoritmi īpašības vārdu locīšanai. Salīdzinot ar šo sistēmu, autora izstrādātajā papildus vēl ir skaitļa vārdu, vietniekvārdu un divdabju locīšana, kā arī elastīgāka izņēmumu apstrāde.

Pašlaik ārpus LUMII izstrādātu vārdformu sintēzes sistēmu, kas nodrošinātu vairāku vārdšķiru locīšanu, nav daudz. Kā vienu no tādām var minēt Tildes izstrādāto vārdu analizatoru[1]. Tas nodrošina ievadītā vārda visu locījumu parādīšanu lietotājam. Spriežot pēc pieejamā sistēmas apraksta, tiek izmantota līdzīga pieeja kā šajā darbā – vārdnīca satur vārdu pamatformas celmus un automātiski tiek uzģenerētas visas vārdformas, izmantojot latviešu valodas likumus. Šīs sistēmas lielākā problēma ir tāda, ka tā nav izmantojama citām automātiskām sistēmām, jo neeksistē interfeiss datu apmaiņai. Tādējādi lietotājam šis vārdformu sintezators ir pieejams tikai pārlūkprogrammas logā.

Vēl arī Google[2] meklētājā ir iespējams vērot, ka rezultātos tiek atgriezts ne tikai meklētais vārds, bet arī dažādi tā locījumi. Tiesa, tā nav pilnīga vārdformu sintēze, jo ne visiem iespējamajiem locījumiem atgriež meklēšanas rezultātus, bet šādi var redzēt, kā informācijas atrašanu atvieglo ne tikai konkrētās ievadītās frāzes saturošo rezultātu parādīšana. Attēlā Nr. 2.1. var redzēt piemēru, kurā meklēšanas rezultātos tiek atgriezta ne tikai meklētā simbolu virkne, bet arī dažādas vārdformas.



2.1. att. **Meklēšana pēc sinonīmiem google.lv meklētājā**

2.2. Vārdu morfoloģiskais analizators

Tā kā vārdu locīšana nav iespējama bez vārdu morfoloģiskās analīzes, tad ir nepieciešams rīks vai sistēma, kas patvaļīgam vārdam atgrieztu vārdformas ģenerēšanai nepieciešamo informāciju – vārdšķiru, sakni (vai saknes, ja tādas ir vairākas), kā arī, protams vārda pamatformu. Tas ir viss nepieciešamais, lai varētu pareizi izlocīt ievadīto vārdu.

[1] http://www.letonika.lv/groups/default.aspx?g=5&r=1100&q=
[2] http://www.google.lv

Darbā apskatītais vārdformu sintezators ir tapis kā papildinājums Pētera Paikena izstrādātajam automatizētās morfoloģiskās un sintaktiskās analīzes rīkam [5], kas tapis SemtiKamola[3] ietvaros. Tas veic vārdu morfoloģisko analīzi, līdz ar to vārdformu darināšanu var balstīt uz iegūtajiem rezultātiem, jo šādas analīzes rezultātā tiek iegūta sintēzei nepieciešamā informācija.

Šis rīks tiek balstīts uz vārdnīcu, kas satur vārdu pamatformas, bet pārējās formas tiek iegūtas ar likumu palīdzību. Tā satur *leksēmas* – morfoloģiskās analīzes abstraktas pamatvienības. Ar to apzīmē *vārdu* un atbilstošās *vārdformas* kopā ar tā nozīmi. Tādējādi vienam un tam pašam vārdam var atbilst divas vai vairākas leksēmas ar attiecīgo jēdziena apzīmējumu. Piemēram, vārdam *loku* var atbilst divas dažādas leksēmas – *lakt* (es loku pienu) un *loks* (es šauju ar loku). Tātad katrā gadījumā vārdam ir pavisam cita nozīme. Šādu vārdnīcu, kas sastāv no *leksēmām*, sauc par *leksikonu*. 2.2. attēlā var redzēt kā savstarpēji tiek sasaistītas šādas klases morfoloģiskā analizatorā.

Vēl tiek glabātas arī lielākā daļā iespējamo galotņu latviešu valodā, lai katram vārdam varētu piekārtot atbilstošo un pēc tā iegūt informāciju par locījumu. Tas ir iespējams tādēļ, ka vārdu locījumi mainās pārsvarā piekārtojot tikai citu galotni. Līdz ar to arī vārdformu sintēze balstās uz šo galotņu piekārtošanu vārda celmam.

Lai vārdu morfoloģiskā analīze būtu precīza, leksēmas ir iedalītas vārdgrupās – grupās ar līdzīgām locīšanas īpašībām. Atbilstoši vārdgrupai tad arī tiek piekārtotas galotnes, kas, zinot leksēmu un attiecīgo galotni, ļauj noteikt vārda locījumu. Kopā tiek izmantotas 24 dažādas vārdgrupas:

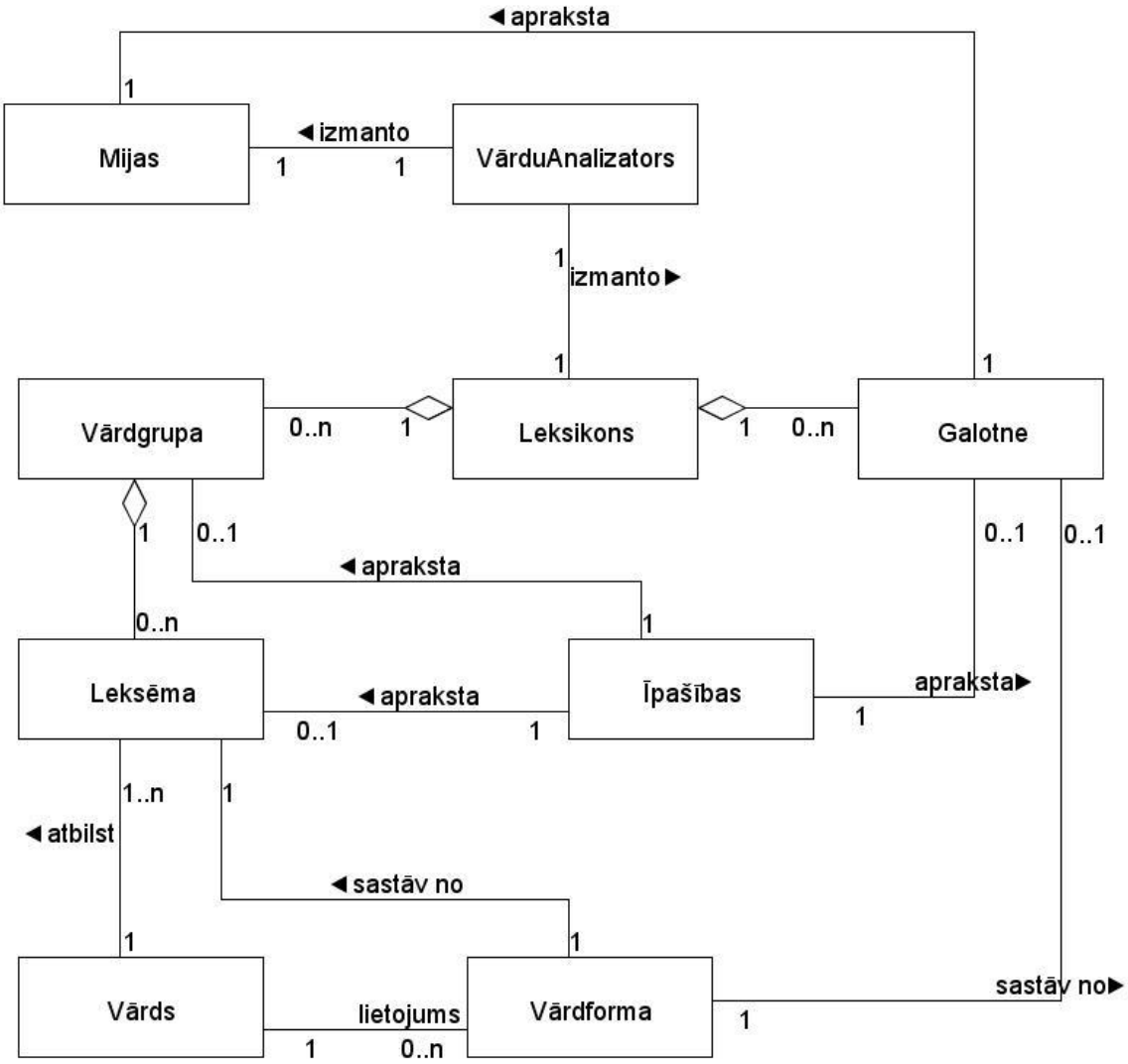
1. Lietvārds 1. deklinācija *–s*,
2. Lietvārds 1. deklinācija *–š*,
3. Lietvārds 2. deklinācija *–is*,
4. Lietvārds 2. deklinācija *–s*,
5. Lietvārds 3. deklinācija *–us*,
6. Lietvārds 4. deklinācija *–a* vīriešu dzimtē,
7. Lietvārds 4. deklinācija *–a* sieviešu dzimtē,
8. Lietvārds 5. deklinācija *–e* vīriešu dzimtē,
9. Lietvārds 5. deklinācija *–e* sieviešu dzimtē,
10. Lietvārds 6. deklinācija *–s*,
11. Nelokāmie lietvārdi,
12. Īpašības vārdi *–s*,
13. Īpašības vārdi *–š*,

[3] http://www.semti-kamols.lv/

14. Darbības vārdi 1. konjugācija tiešie,
15. Darbības vārdi 2. konjugācija tiešie,
16. Darbības vārdi 3. konjugācija tiešie,
17. Darbības vārdi 1. konjugācija atgriezeniskie,
18. Darbības vārdi 2. konjugācija atgriezeniskie,
19. Darbības vārdi 3. konjugācija atgriezeniskie,
20. Apstākļa vārdi,
21. Kārtas skaitļa vārdi,
22. Pamata skaitļa vārdi – vienskaitļa,
23. Pamata skaitļa vārdi – daudzskaitļa,
24. *Hardcoded* vārdformas: prievārdi, partikulas, saikļi un vietniekvārdi.

Tā kā rīks izstrādāts JAVA vidē un tā datubāze ir XML formātā, tad tas vairs nav piesaistīts noteiktai platformai, kā tas bija iepriekš izstrādātajiem rīkiem. Tāpat arī XML formāts nodrošina to, ka to ir iespējams lietot ne tikai tiešsaistē.

Attēlā Nr. 2.2. var apskatīt šī rīka konceptuālo klašu diagrammu. Turpinājumā visas attēlotās formālās diagrammas tiks attēlotas kā *Unified Modeling Language* jeb UML [6] diagrammas, lai tās būtu veidotas pēc vienota standarta.



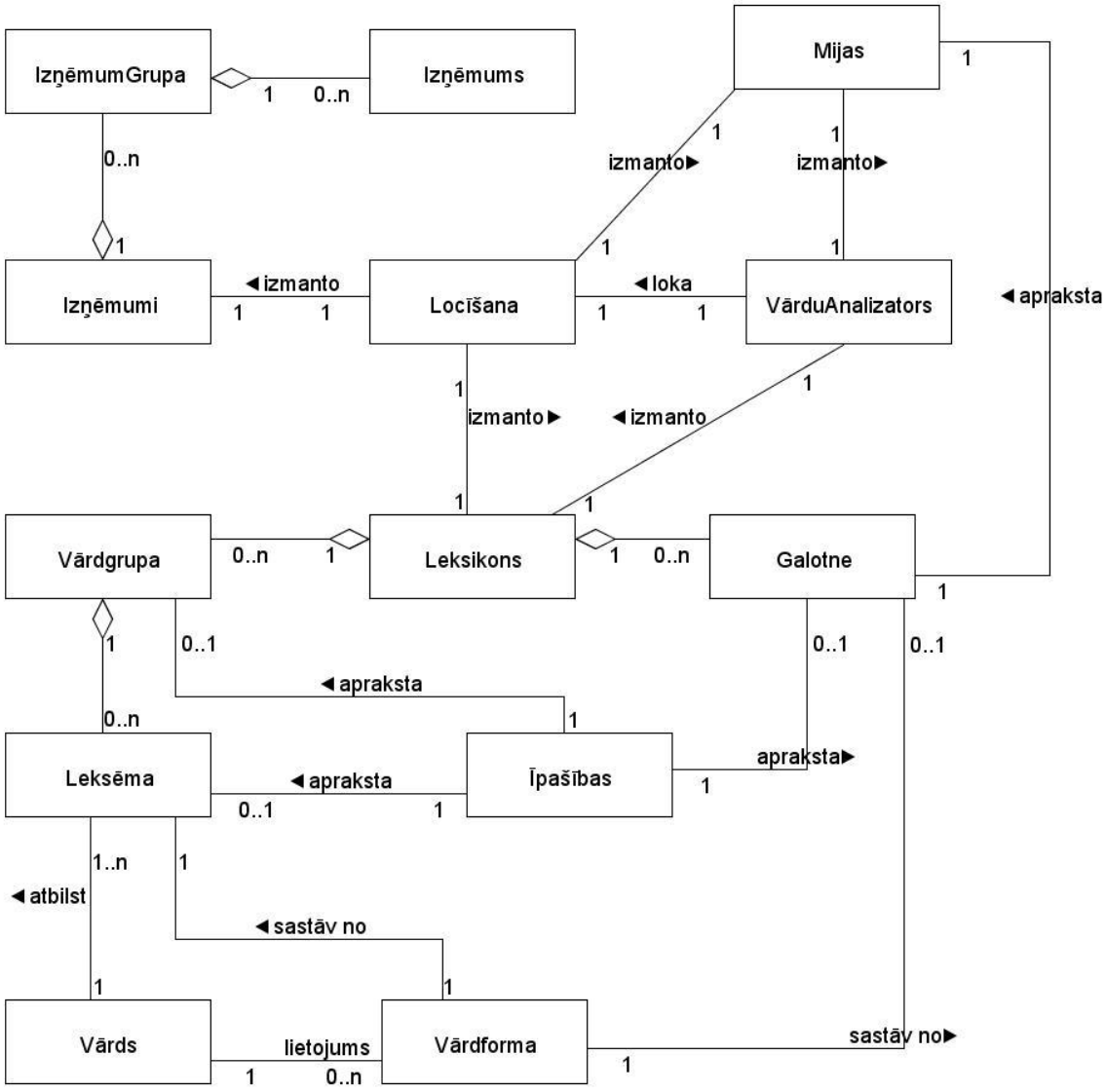
2.2. att. **Konceptuāla klašu diagramma morfoloģiskajam analizatoram**

3. TEHNISKAIS RISINĀJUMS

Kā jau iepriekš tika minēts, tad autora izstrādātais vārdformu sintezators balstās uz jau izstrādāta morfoloģiskā analizatora. Tā kā tas ir izstrādāts JAVA [7] programmēšanas valodā, tad arī sintezators tiek izstrādāts, izmantojot to pašu programmēšanas valodu.

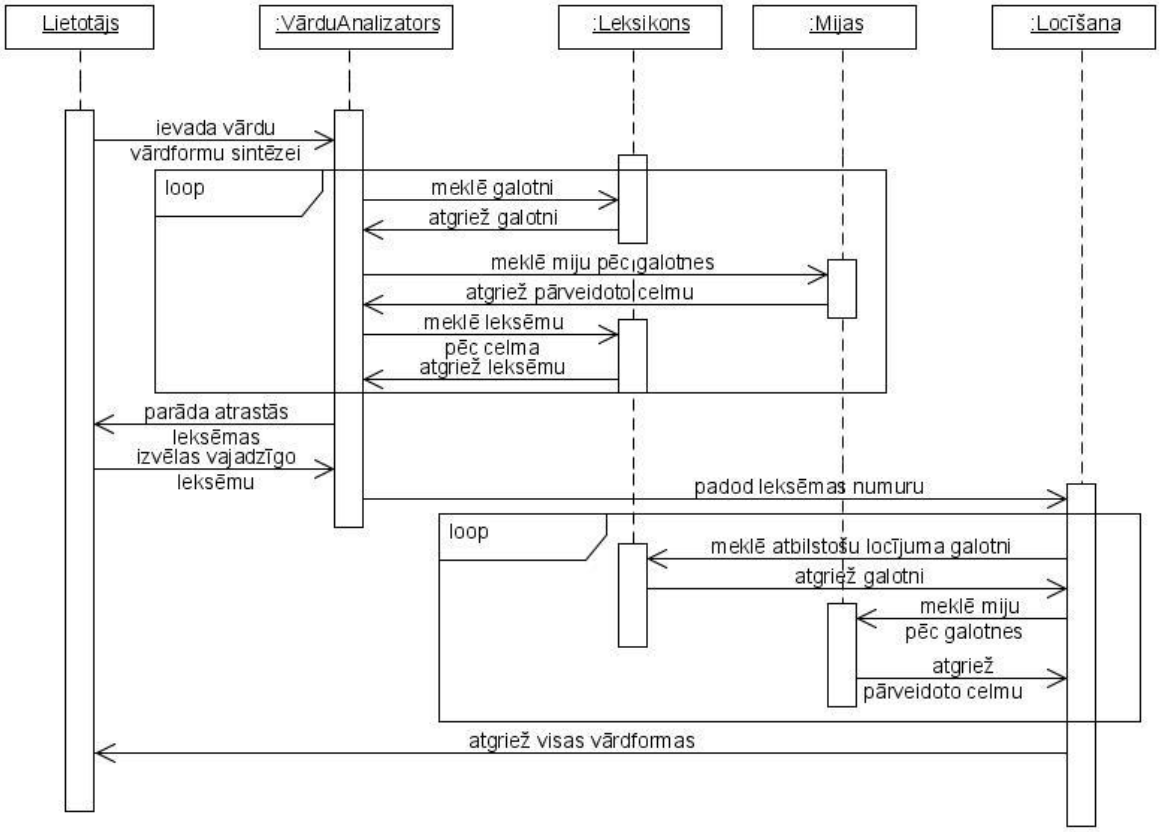
Tādējādi no morfoloģiskā analizatora tiek iegūta nepieciešamā informācija par lokāmo vārdu – pamatforma, vārdšķira un arī vēl sīkāks iedalījums (piemēram, konjugācija darbības vārdam vai deklinācija lietvārdam).

Lai šāda informācijas apmaiņa būtu efektīva ir nepieciešama atbilstoša datu struktūra, kas saturētu vajadzīgo minimumu. Tas tiek panākts ar leksēmas palīdzību, kas tad arī satur visus vajadzīgos datus. 3.1. attēlā var aplūkot klašu diagrammu vārdformu sintezatoram kopā ar morfoloģisko analizatoru. Tajā ir iekļauta arī izņēmumu apstrāde, kas tiks apskatīta nākošajā nodaļā.



3.1. att. **Konceptuāla klašu diagramma morfoloģiskajam analizatoram kopā ar vārdformu sintezatoru**

3.2. attēlā var redzēt kā vispārīgā gadījumā notiek vārda locīšana. Tajā var redzēt, kā notiek iesaistīto klašu sadarbība gan savstarpēji, gan arī ar lietotāju, lai rezultātā atrieztu izvēlētā lokāmā vārda visas vārdformas. Tiesa, uzskatāmības labad nav iekļauta izņēmumu apstrāde.



3.2. att. **Vārda locīšanas secību diagramma**

Tā kā pavisam kopā ir piecas lokāmās vārdšķiras, tad turpinājumā tiks apskatīti visi pieci algoritmi to locīšanai.

3.1.Lietvārdu locīšana

Lietvārdu locīšana ir samērā vienkārša, jo ir tikai septiņi locījumi un divi skaitļi, kas kopā veido četrpadsmit dažādus locījumus, bet tā kā Instrumentāļa locījumu var iegūt no citiem locījumiem, tad tas netiek iekļauts sintēzē. Rezultātā vispārīgā gadījumā ir tikai jāpiekārto galotnes, kas atbilst konkrētajai deklinācijai un skaitlim.

Precīzākai galotņu piekārtošanai lietvārdi ir sadalīti vienpadsmit dažādās vārdgrupās:

1. Lietvārds 1. deklinācija *–s*;
2. Lietvārds 1. deklinācija *–š*;
3. Lietvārds 2. deklinācija *–is*;
4. Lietvārds 2. deklinācija *–s*;

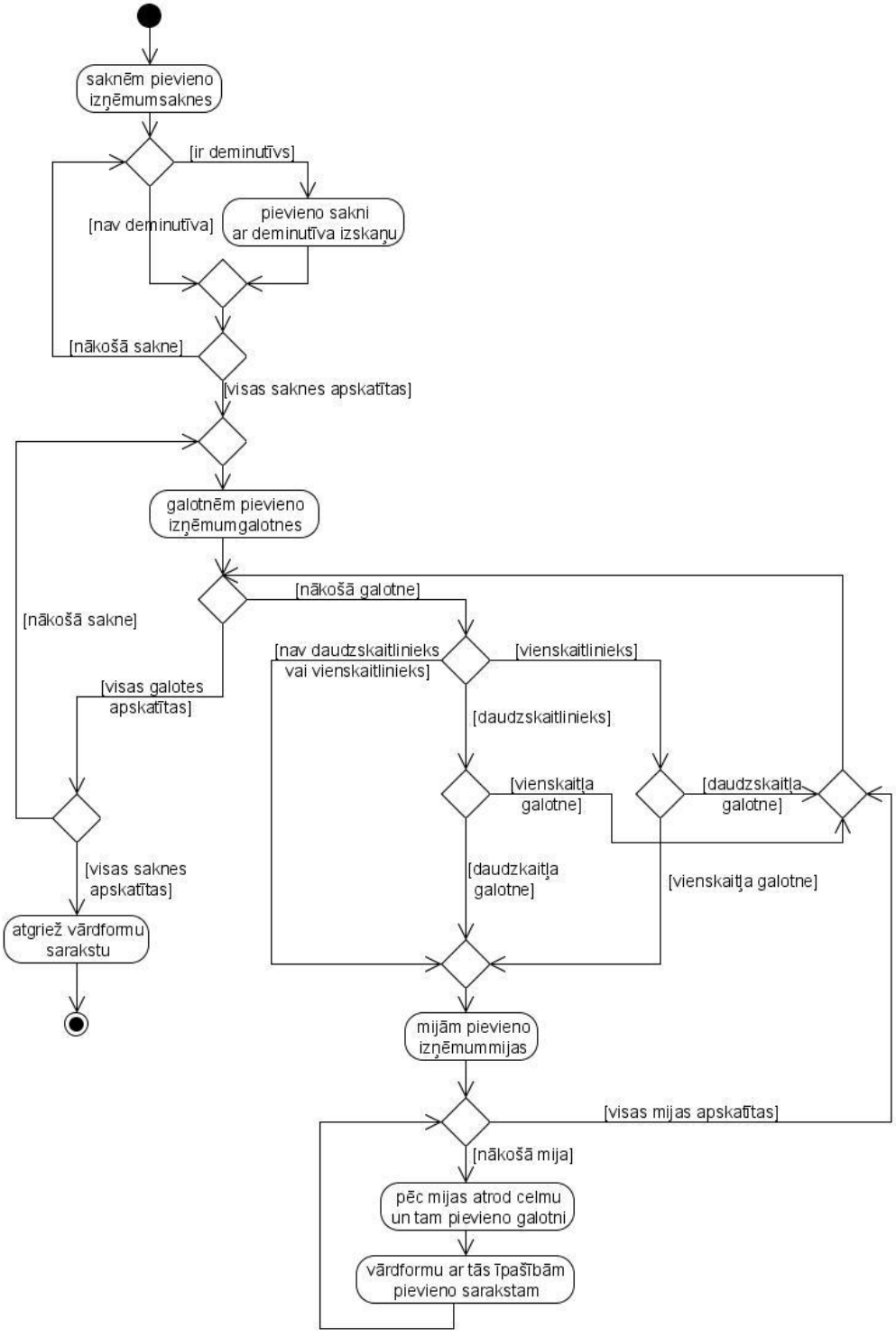
5. Lietvārds 3. deklinācija *–us*;
6. Lietvārds 4. deklinācija vīriešu dzimtē *–a*;
7. Lietvārds 4. deklinācija sieviešu dzimtē *–a*;
8. Lietvārds 5. deklinācija vīriešu dzimtē *–e*;
9. Lietvārds 5. deklinācija sieviešu dzimtē *–e*;
10. Lietvārds 6. deklinācija *–s*;
11. Nelokāmie lietvārdi.

Tādējādi vispirms jānoskaidro, vai lietvārds vispār ir lokāms – ja nav, tad tiek atgriezta vārda pamatforma. Pretējā gadījumā šīs vārdgrupas tiek iedalītas divas lielās grupās – vārdos, kuriem nav līdzskaņu mija atsevišķos locījumos, un vārdos, kuriem ir celma beigu izmaiņas. Līdzskaņa mijas ir 2., 5. un 6. deklinācijas lietvārdiem – sīkāk tās tika apskatītas iepriekšējā darba nodaļā, tāpēc otrreiz netiks aplūkots, kuriem celmu beigu līdzskaņiem attiecīgajā locījumā tiek pielietota mija.

Īpaši ir jāapskata vienskaitļa vokatīva locījums – parasti tiem ir tāda pati galotne kā vienskaitļa nominatīvā vai arī īpaša galotne (2. un 3. deklinācijas lietvārdiem), bet deminutīviem visās deklinācijās nav galotnes. Savukārt, daudzskaitļa vokatīva galotne ir tāda pati kā daudzskaitļa nominatīvā. Vēl ir jāievēro, ka 1., 2. 4. un 5. deklinācijas lietvārdiem var vēl būtiskāk atšķirties vokatīva galotnes. Tiem var būt šādas vārda beigas šajā locījumā:

- galotne *-s*, piemēram, vārdiem, *koks*;
- bez galotnes, piemēram, *arāj*;
- galotne *-s* un bez galotnes, piemēra, *tēvs* un *tēv*.

Tā kā lielākajai daļai šo deklinācijas lietvārdu vienskaitļa vokatīva galotne neatšķiras no vienskaitļa nominatīva galotnes, tad pārējie vārdi tiks iekļauti pie izņēmumgadījumu apstrādes, kas tiks apskatīta nākošajā nodaļā. Tāpat tajā tiks apskatīti arī gadījumi, kad noteiktiem lietvārdiem nav līdzskaņu mija. Vēl īpaši jāapskata daudzskaitlinieki un vienskaitlinieki, lai tiem netiktu ģenerētas vārdformas nepareizajā skaitlī. Rezultātā tiek iegūta 3.3. attēlā redzamā UML aktivitāšu diagramma.



3.3. att. **UML aktivitāšu diagramma lietvārdu locīšanai**

3.2.Darbības vārdu locīšana

Kā jau iepriekš tika minēts, tad darbības vārdiem ir sarežģītākais locīšanas algoritms. Tāpēc tie ir sadalīti šādās sešās vārdgrupās:

1. Darbības vārdi 1. konjugācija tiešie;

2. Darbības vārdi 1. konjugācija atgriezeniskie;
3. Darbības vārdi 2. konjugācija tiešie;
4. Darbības vārdi 2. konjugācija atgriezeniskie;
5. Darbības vārdi 3. konjugācija tiešie;
6. Darbības vārdi 3. konjugācija atgriezeniskie.

Tāpat darbības vārdiem eksistē bezpersonas forma – divdabis –, kuru gan loka līdzīgi īpašības vārdiem, bet jāietver darbības vārdu vārdformas sintēzes algoritmā. Šī paša iemesla dēļ darbības vārdu leksēmām nav norādīta vārdšķira – to norādot tiks locīts vai nu darbības vārds, ja to norādīs, vai arī divdabis. Savukārt, galotnēm ir norādīts, kas ar to var beigties (vai nu divdabis, vai arī darbības vārds), līdz ar to tādējādi tiek panākta šī atsevišķā vārdformu sintēze darbības vārdam un divdabim, ja nepieciešams.

Papildus grūtības rada tas, ka 1. konjugācijas darbības vārdiem var būt trīs dažādas saknes, lai pareizi tiktu darinātas nākotnes, pagātnes un tagadnes vārdformas. Šī problēma tiek atrisināta ar saknes numura norādīšanu leksikonā katrai galotnei. Tādējādi visām galotnēm – līdz ar to arī attiecīgajiem locījumiem – tiks piekārtota tikai viena pareizā sakne, kas arī ir nepieciešams.

Tā kā tiešie un atgriezeniskie darbības vārdi ir dažādās vārdgrupās, tad sīkāk nav jāapskata, kurā gadījumā piekārtot tiešo vai atgriezenisko galotni, jo pareizās tiek piekārtotas pēc vārdgrupas. Tā kā darbības vārdu saliktās tagadnes, saliktās pagātnes un saliktās nākotnes laiki savā starpā atšķiras tikai ar to priekšā piekārtoto darbības vārda ir formām, tad tiek sintezēti tikai vienkāršie laiki.

Darbības vārdu locīšanu būtiski ietekmē to konjugācija, tāpēc nosacījumus, kas jāievēro vārdformu sintēzē apskatīsim katras konjugācijas ietvaros. Tādējādi vieglāk būs uztverams, vai un kādas mijas ir konkrētos locījumos, kā arī citas nepieciešamās celma izmaiņas.

Darbības vārdu 1. konjugācija

Tā kā šīs konjugācijas darbības vārdiem ir trīs pamatformas, tad tas savā ziņā vienkāršo algoritmu, jo mijas un celma pārmaiņas ir ietvertas šajās trīs pamatformās. Pilnībā iztikt bez mijām gan nav iespējams, jo tādas eksistē tagadnes 2. personai, kā arī dažiem nākotnes locījumiem.

Tagadnes 2. personai atšķirībā no pārējām tagadnes personām ir nepieciešama līdzskaņu mija – piemēram, vārda izciest tagadnes celms ir izcieš (*es izciešu*), bet 2. personā ir *š* -> *t* mija (*tu izciet*). Jāievēro, ka šī mija nav viennozīmīgi pielietojama, jo var būt arī š -> s, piemēram, vārdam plēst – es plēšu un tu plēst. Tāpēc ir nepieciešams zināt pagātnes formu, jo pēc tās var izsecināt, kura mija jāpielieto konkrētā gadījumā. Līdzīgi ir arī ar *ž* -> *z* (es blenžu

un tu blenz) un *ž -> d* (es spriežu un tu spried) mijām. Pārējās mijas, piemēram, ļ->l un bj->b, ir viennozīmīgi pielietojams, tāpēc tās sīkāk netiks apskatītas.

Savukārt, nākotnē celma pārmaiņas ir tikai vārdiem, kam nenoteiksmes pamatforma beidzas ar līdzskani *s*. Arī šajā gadījumā jāskatās uz pagātnes pamatformas pēdējo burtu, tādējādi pavisam ir trīs iespējas:

- *s->šī*, piemēram, *satumst* un *es satumsīšu*;
- *s->dī*, piemēram, *sviest* un *es sviedīšu*;
- *s->tī*, piemēram, *mest* un *es metīšu*.

Tā kā no nenoteiksmes pamatformas veido arī citas darbības vārda vārdformas, kurām nav nepieciešama šīs celma pārmaiņas, piemēram, vēlējuma izteiksmi, tad tās nav iespējams iekļaut šajā pamatformā. Tāpēc šādas celma izmaiņas ir jāapskata atsevišķi.

Darbības vārdi 2. konjugācija

Šajā konjugācijā nav ne miju, ne kādu savādāku celmu būtisku pārmaiņu. Vienīgi visos vienkāršās pagātnes un tagadnes locījumos, izņemot vienkāršās tagadnes vienskaitļa otro personu, kā arī abu skaitļu trešo personu, parādās papildus celmu beigu līdzskanis *j*. Tā kā šāda īpašība ir lielākajai daļai 2. konjugācijas darbības vārdu, tad burts *j* tiek pievienots attiecīgajām locījumu galotnēm, līdz ar to nekāda īpaša celma apstrāde vairs nav vajadzīga.

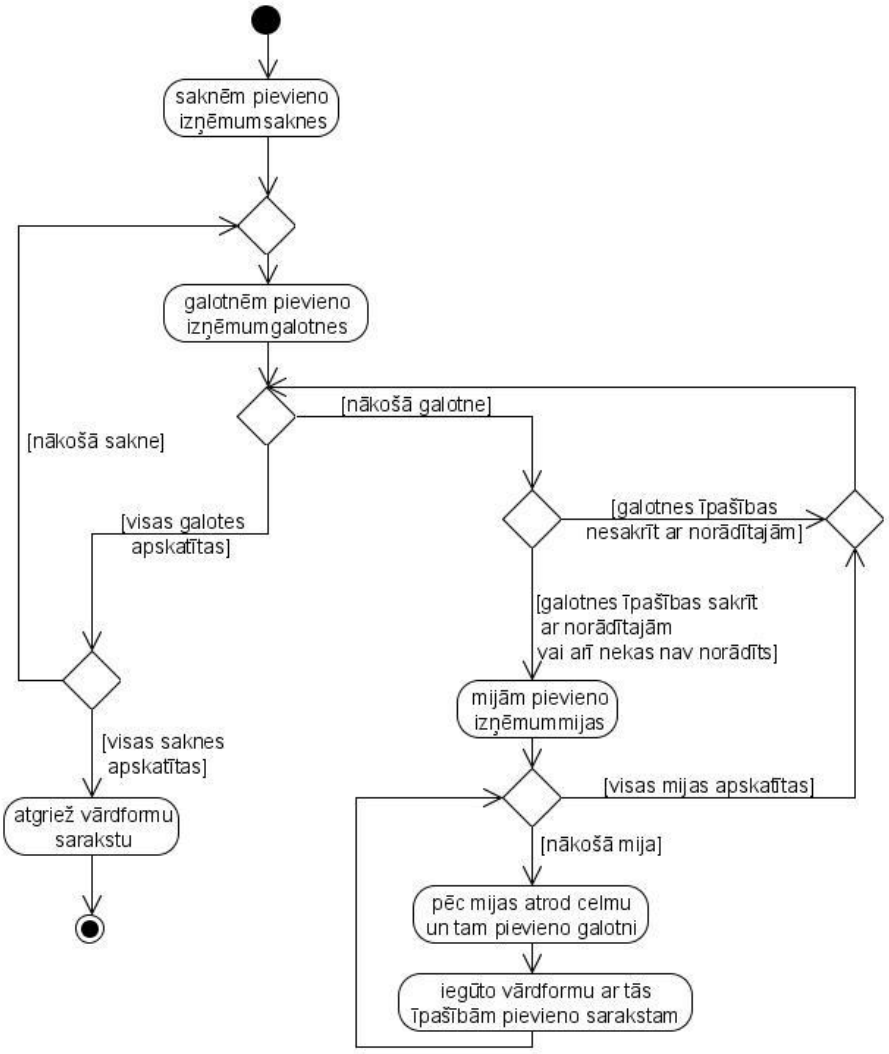
Darbības vārdi 3. konjugācija

Savukārt, šajā konjugācijā ir visvairāk dažādu celma pārmaiņu. Tās ir gan īstenības izteiksmes tagadnē un nākotnē, kā arī vajadzības izteiksmē.

3. konjugācijas vārdiem tagadnē celmam tiek noņemts pēdējais patskanis no saknes, lai pēc tam pievienotu galotni. Jāievēro, ka vārdiem, kas nenoteiksmē beidzas ar *–īt, -īties, -ināt* un *–ināties* ir atšķirīgas galotnes no pārējiem darbības vārdiem īstenības izteiksmes daudzskaitļa 1. un 2. personā – jālieto attiecīgi *-ām, -āt* vai *-āmies, -āties*. Tāpat eksistē arī līdzskaņa mija *c->k*, piemēram, *sacīt* un *es saku*.

Līdzīgi arī vajadzības izteiksmē šiem vārdiem ar attiecīgajām četrām galotnē tiek pievienots galotne –a, kamēr citiem 3. deklinācijas darbības vārdiem galotne nav. Tieši tas pats ir arī īstenības izteiksmes tagadnes 3. personā.

Visām trim konjugācijām kopīga ir priedēkļa *jā* pievienošana vārda priekšā vajadzības izteiksmē. Apvienojot visus šos nosacījumus, iegūst darbības vārdu locīšanas algoritmu, kuru var apskatīt 3.4. attēlā. Visas iepriekš minētās celmu pārmaiņas notiek solī *pēc mijas atrod celmu un tam pievieno galotni.*



3.4. att. **UML aktivitāšu diagramma darbības vārdu locīšanai**

3.3.Īpašības vārdu locīšana

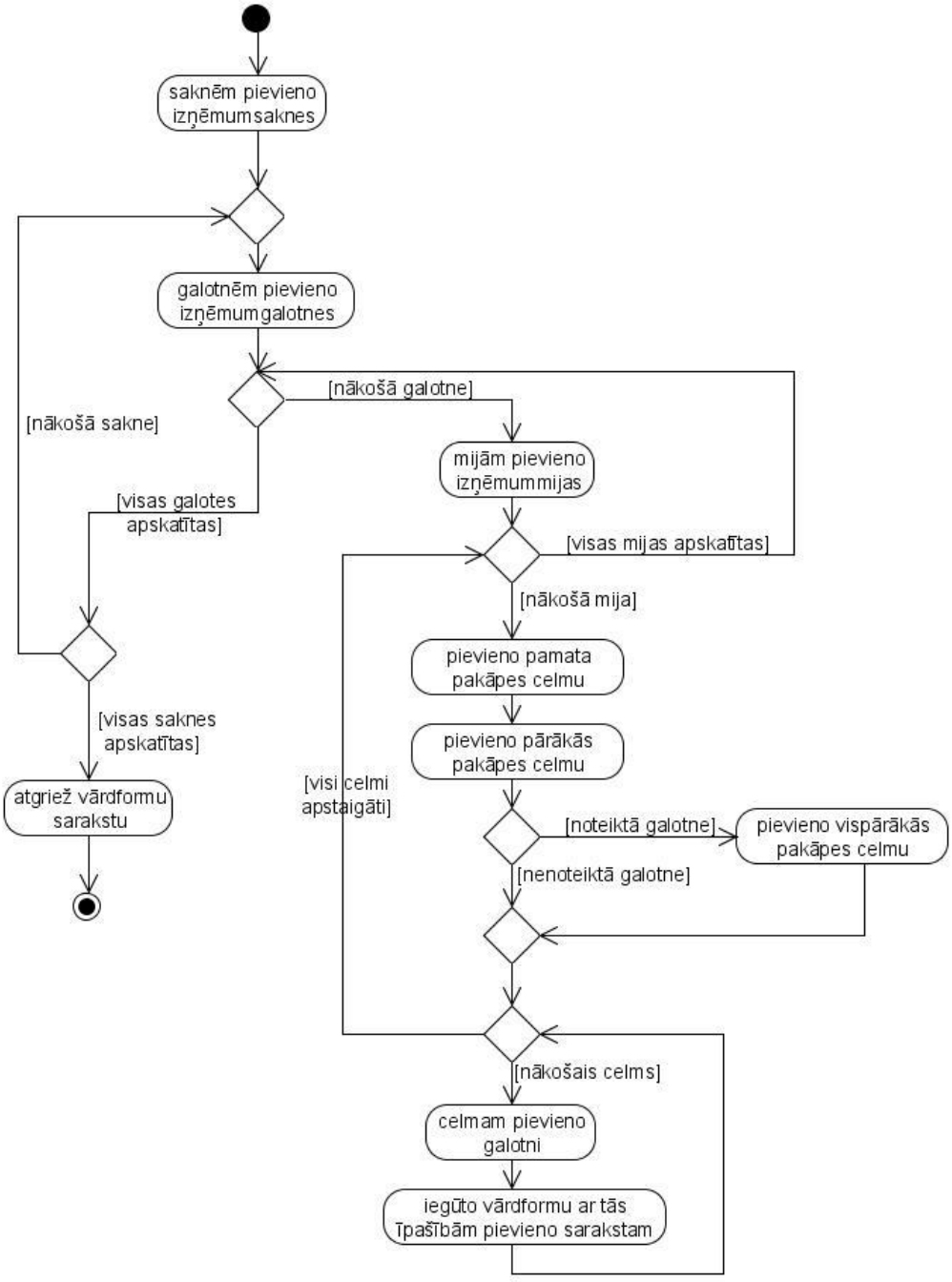
Īpašības vārdu locīšana ir viena no vienkāršākajām, jo tiem nav ne vārdu miju, ne neviennozīmīgu galotņu. Tie sīkāk tiek iedalīti šādās vārdgrupās:

- Īpašības vārds ar galotni *–s*,
- Īpašības vārds ar galotni *–š*.

Vienīgā atšķirība starp tām ir tāda, ka vienskaitļa nominatīvā galotne ir vai nu *–s* vai arī *–š*. Visos pārējos locījumos un skaitļos tās ir identiskas, līdz ar to locīšanas ziņā nav nekādu atšķirību.

Jāņem vērā, ka īpašības vārdiem ir divu veidu galotnes – nenoteiktā un noteiktā –, tātad uzģenerēto vārdformu būs divreiz vairāk. Lai vārdformu darīšana būtu pilnīga, ir jāģenerē arī īpašības vārdi pārākajā un vispārākajā pakāpēs (vispārākā pakāpe neeksistē vārdiem ar nenoteikto galotni). Visās trīs pakāpēs ir vienas un tās pašas galotnes atbilstošajos locījumos,

vienīgi, ja pamata pakāpes vienskaitļa nominatīvā beidzās ar *–š*, tad pārējās pakāpēs būs galotne *–s*. Vārdiem ar noteikto galotni vienskaitļa datīvā un lokatīvā, kā arī daudzskaitļa datīvā un lokatīvā ir paplašināta galotne, bet tā nav paplašināta daļai vārdiem ar šādām izskaņām *–ējais*, *-ējā*. Attēlā Nr. 3.5. var redzēt īpašības vārdu UML aktivitāšu diagrammu.



3.5. att. **UML aktivitāšu diagramma īpašības vārdu locīšanai**

Tā kā viens un tas pats īpašības vārds var būt gan sieviešu, gan vīriešu dzimtē, tad celmam tiek piekārtotas abu veidu galotnes, tādējādi kopā tiek iegūtas 82 vārdformas. Tabulā Nr. 3.1. var redzēt pilnībā izlocītu vārdu *zaļš*.

3.1. tabula **Īpašības vārda zaļš visi locījumi**

	Vīriešu dzimte						
		Nenoteiktā galotne			Noteiktā galotne		
Vienskaitlis		Pamata	Pārākā	Vispārākā	Pamata	Pārākā	Vispārākā
	Nominatīvs	zaļš	zaļāks	neeksistē	zaļais	zaļākais	viszaļākais
	Ģenitīvs	zaļa	zaļāka	neeksistē	zaļā	zaļākā	viszaļākā
	Datīvs	zaļam	zaļākam	neeksistē	zaļajam	zaļākajam	viszaļākajam
	Akuzatīvs	zaļu	zaļāku	neeksistē	zaļo	zaļāko	viszaļāko
	Lokatīvs	zaļā	zaļākā	neeksistē	zaļajā	zaļākajā	viszaļākā
	Vokatīvs	neeksistē	neeksistē	neeksistē	zaļais	zaļākais	viszaļākais
		Nenoteiktā galotne			Noteiktā galotne		
Daudzskaitlis		Pamata	Pārākā	Vispārākā	Pamata	Pārākā	Vispārākā
	Nominatīvs	zaļi	zaļāki	neeksistē	zaļie	zaļākie	viszaļākie
	Ģenitīvs	zaļu	zaļāku	neeksistē	zaļo	zaļāko	viszaļāko
	Datīvs	zaļiem	zaļākiem	neeksistē	zaļajiem	zaļākajiem	viszaļākajiem
	Akuzatīvs	zaļus	zaļākus	neeksistē	zaļos	zaļākos	viszaļākos
	Lokatīvs	zaļos	zaļākos	neeksistē	zaļajos	zaļākajos	viszaļākajos
	Vokatīvs	neeksistē	neeksistē	neeksistē	zaļie	zaļākie	viszaļākie
	Sieviešu dzimte						
		Nenoteiktā galotne			Noteiktā galotne		
Vienskaitlis		Pamata	Pārākā	Vispārākā	Pamata	Pārākā	Vispārākā
	Nominatīvs	zaļa	zaļāka	neeksistē	zaļā	zaļākā	viszaļākā
	Ģenitīvs	zaļas	zaļākas	neeksistē	zaļās	zaļākās	viszaļākās
	Datīvs	zaļai	zaļākai	neeksistē	zaļajai	zaļākajai	viszaļākajai
	Akuzatīvs	zaļu	zaļāku	neeksistē	zaļo	zaļāko	viszaļāko
	Lokatīvs	zaļā	zaļākā	neeksistē	zaļajā	zaļākajā	viszaļākajā
	Vokatīvs	neeksistē	neeksistē	neeksistē	zaļā	zaļākā	viszaļākā
		Nenoteiktā galotne			Noteiktā galotne		
Daudzskaitlis		Pamata	Pārākā	Vispārākā	Pamata	Pārākā	Vispārākā
	Nominatīvs	zaļas	zaļākas	neeksistē	zaļās	zaļākās	viszaļākās
	Ģenitīvs	zaļu	zaļāku	neeksistē	zaļo	zaļāko	viszaļāko
	Datīvs	zaļām	zaļākām	neeksistē	zaļajām	zaļākajām	viszaļākajām
	Akuzatīvs	zaļas	zaļākas	neeksistē	zaļās	zaļākās	viszaļākās
	Lokatīvs	zaļās	zaļākās	neeksistē	zaļajās	zaļākajās	viszaļākajās
	Vokatīvs	neeksistē	neeksistē	neeksistē	zaļās	zaļākās	viszaļākās

3.4.Skaitļa vārdu locīšana

Kārtas skaitļa vārdu locīšana ir līdzīga lietvārdu un īpašības vārdu locīšanai. Tā kā izšķir divus skaitļa vārdu veidus – kārtas un pamata – un tas ietekmē vārdformu sintēzi, tad tie tiek šādai sadalīti:

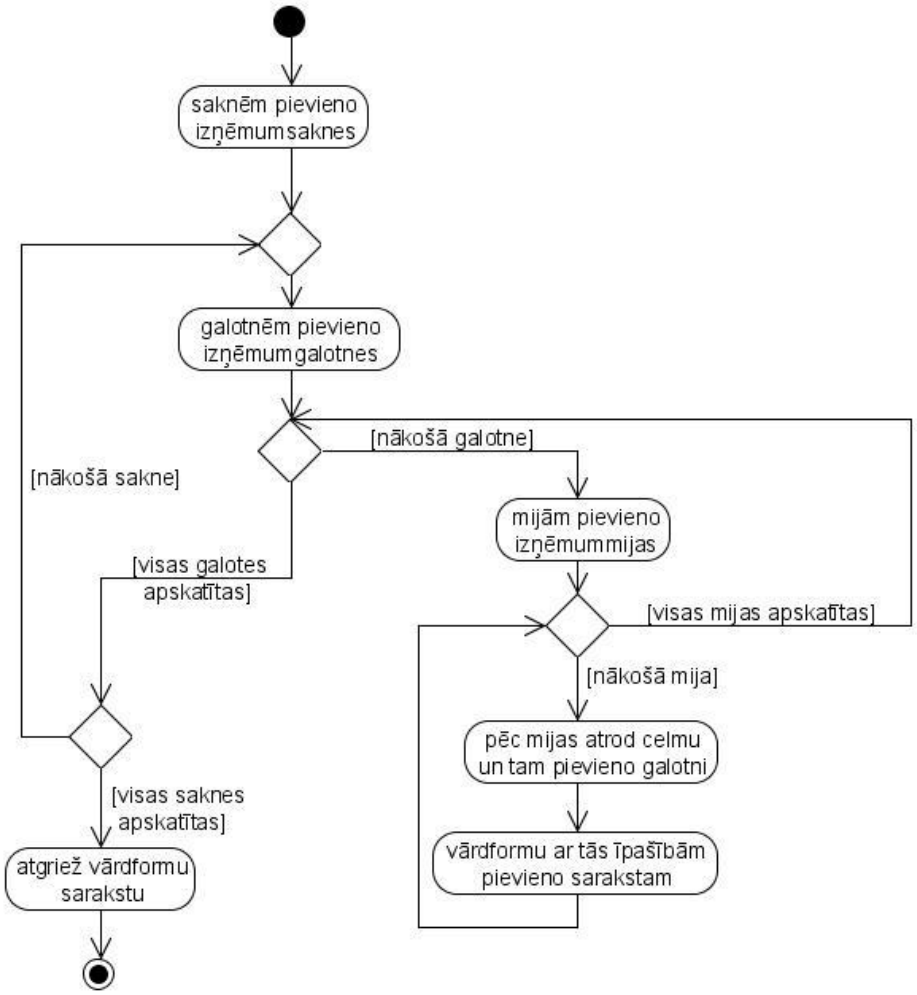
- Kārtas skaitļa vārdi,
- Pamata skaitļa vārdi – vienskaitļa,
- Pamata skaitļa vārdi – daudzskaitļa.

Līdz ar to šo vārdiem piekārtotās galotnes ir atkarīgas no tā, kurā vārdgrupā tie ietilpst. Īpatnēji tiek locīts vārds *trīs*, tāpēc tas tiks apskatīts nākošajā nodaļā izņēmumgadījumu apstrādē. 3.2. tabulā var apskatīt izlocītu vārdu *pieci*.

3.2. tabula **Skaitļa vārda pieci visi locījumi**

Locījums	Vīriešu dzimte	Sieviešu dzimte
Nominatīvs	pieci	piecas
Ģenitīvs	piecu	piecu
Datīvs	pieciem	piecām
Akuzatīvs	piecus	piecas
Lokatīvs	piecos	piecās

Rezultātā iegūtais locīšanas algoritms ir viens no vienkāršākajiem. 3.6. attēlā to tad arī var aplūkot.



3.6. att. **UML aktivitāšu diagramma skaitļa vārdu locīšanai**

3.5. Vietniekvārdi

Vietniekvārdus ir vissarežģītāk sadalīt grupās ar līdzīgām īpašībām, jo to iespējamās galotnes ir ļoti dažādas – daļu loka kā I un IV deklinācijas vārdus, daļu kā pamata īpašības vārdus, bet pārējos loka īpatnēji. Līdz ar to katrs vietniekvārds tiek pievienots pie jau esošās vārdgrupas, bet, ja tas nav iespējams, tad to pievieno pie pēdējās vārdgrupas *Hardcoded.*

Protams, pie īpašībām papildus jānorāda vārdšķira, lai to atšķirtu no pārējiem grupas vārdiem, bet pašu locīšanu tas neietekmē, jo galotnes tiek piekārtotas, balstoties uz vārdgrupu. Savukārt, pie pēdējās grupas piederošie vietniekvārdi tiks locīti ar nākošajā nodaļā apskatīto izņēmumu apstrādes palīdzību, jo, piemēram, vārdu *es, šis, jūs* visas vārdformas nav iespējams aprakstīt ar likumu palīdzību. Tabulā Nr. 3.3. to var labi redzēt.

3.3. tabula **Īpatnēji lokāmo vietniekvārdu vārdformas**

Locījums	*es*	*šis*	*jūs*
Nominatīvs	es	šis	jūs
Ģenitīvs	manis	šī, šā	jūsu
Datīvs	man	šim	jums
Akuzatīvs	mani	šo	jūs
Lokatīvs	manī	šai, šajā, šinī	jūsos
Vokatīvs	neeksistē	neeksistē	neeksistē

4. IZŅĒMUMU APSTRĀDE

Lai vārdformu būtu pēc iespējas precīzāka ir jāņem vērā dažādi izņēmumi, kad kādus vārdus no vārdgrupas loka savādāk kā pārējos. Tā kā šādi izņēmumgadījumi laika gaitā var mainīties, piemēram, valodā ienāk jauni vārdi, tad ir nepieciešams elastīgs risinājums. Tas vajadzīgs, lai bez iejaukšanās sistēmas izejas kodā būtu iespējams pievienot jaunus izņēmumus.

Iepriekšējās vārdformu sintēzes sistēmās šāda iespēja netika paredzēta, un izņēmumgadījumi tika iestrādāti vai nu algoritmos, vai datubāzē kā jauna vārdgrupa. Pirmā varianta lielākā problēma ir tāda, ka laika gaitā būtu nepieciešams modificēt sistēmu, tādējādi sistēmas uzturēšana tiek apgrūtināta. Otra varianta gadījumā tiek sarežģīta datubāzes struktūra, jo sākotnējais vārdgrupu sadalījums tiek sadalīts sīkāk un ne vienmēr tas būtu nepieciešams, ja kopējais līdzīgo izņēmumgadījumu skaits ir neliels. Arī šajā gadījumā var būt nepieciešama iejaukšanās sistēmas izejas kodā, ja visu vārdgrupu apstrāde nav pietiekami universāla.

Šo iemeslu dēļ tika izvēlēts veidot mehānismu, kas datubāzē glabātos izņēmumgadījumus apstrādātu pēc atbilstošajām to īpašībām, tādējādi nepieciešamības gadījumā vajadzētu tikai pievienot jaunu izņēmumu datubāzei kopā ar nosacījumiem, kas norādītu, kā tas atšķiras no pārējiem vārdiem. Tādējādi nebūtu nepieciešama ne iejaukšanās datubāzes struktūrā, ne izejas koda modificēšana. Lai šādu mehānismu izveidotu, vispirms ir nepieciešams apskatīt, kādi tad šie izņēmumi var būt.

4.1. Izņēmumu veidi

Latviešu valodā izņēmumi ir daudz un tos ir nepieciešams sagrupēt grupās ar līdzīgām īpašībām, lai varētu izveidot pietiekami universālu izņēmumu apstrādi. Pavisam kopā tādu ir trīs šādu grupu.

Kā pirmo grupu apskatīsim izņēmumus, kad noteiktiem vārdiem neeksistē mija vai arī eksistē cita mija noteiktos locījumos atšķirībā no pārējiem vārdgrupas vārdiem. Šāda situācija ir izplatīta starp lietvārdiem, kad vārdiem ar noteiktām vārda beigām nav līdzskaņu mijas, lai gan visiem citiem tāda ir jālieto pareizai vārdformu darināšanai. Savukārt, dažiem VI deklinācijas lietvārdiem vienā un tajā pašā locījumā ir iespējams vārdformas gan ar miju, gan arī bez tās. Līdz ar to arī šādu situāciju ir jāspēj apstrādāt. Atsevišķos gadījumos šādi izņēmumi ir tikai vārdiem ar noteiktu zilbju skaitu, piemēram, *Valdis*, kuram ir 2 zilbes, tiek locīts kā izņēmumvārds bez mijas, bet *Visvaldis* tiek locīts kā parasts II deklinācijas lietvārds ar miju *d->ž*.

Vēl ir iespējama situācija, kad kādā noteiktā locījumā ir vairākas galotnes vai arī cita galotne atšķirībā no pārējiem vārdgrupas vārdiem. Piemēram, dažiem lietvārdiem vokatīvā ir viena savādāka galotne vai arī šī papildus galotne veido divas iespējamas vārdformas.

Trešā un sarežģītākā grupa ir vārdi, kuru pārmaiņas celmā ir atšķirīgas no pārējiem vārdgrupas vārdiem. Spilgts piemērs tam ir vārda *trīs* locīšana. Tabulā Nr. 4.1. var apskatīt tā un arī vārda *četri*, kas ietilpst tajā pašā vārdgrupā, locīšanu. Var redzēt, ka dažos locījumos abiem ir kopīgas galotnes, bet celma pārmaiņas ir dažādas, kā arī iespējamas papildus vārdformas vārdam *trīs* dažos locījumos.

4.1. tabula **Skaitļa vārda trīs visas vārdformas**

	Vīriešu dzimte		Sieviešu dzimte	
Nominatīvs	trī**s**	četr**i**	trī**s**	četr**as**
Ģenitīvs	trij**u**	četr**u**	trij**u**	četr**u**
Datīvs	tri**m**, trij**iem**	četr**iem**	tri**m**, trij**ām**	četr**ām**
Akuzatīvs	trī**s**	četr**us**	trī**s**	četr**as**
Lokatīvs	trij**os**, trī**s**	četr**os**	trij**ās**, trī**s**	četr**ās**
Vokatīvs	neeksistē	neeksistē	neeksistē	neeksistē

4.2. Realizācija

Tā kā izņēmumi veidi savā starpā ir atšķirīgi, tad gan datubāzei, gan algoritmiem jābūt pietiekoši universāliem, lai spētu tos apstrādāt. Pretējā gadījumā tie var kļūt pārāk sarežģīti un līdz ar to arī to veiktspēja var izrādīties pārāk zema. Tāpēc turpinājumā tiks apskatīts kā tas tika realizēts, lai efektīvi darbotos kopā ar izstrādāto latviešu valodas vārdformu sintezatoru, kas tika apskatīts iepriekšējā nodaļā.

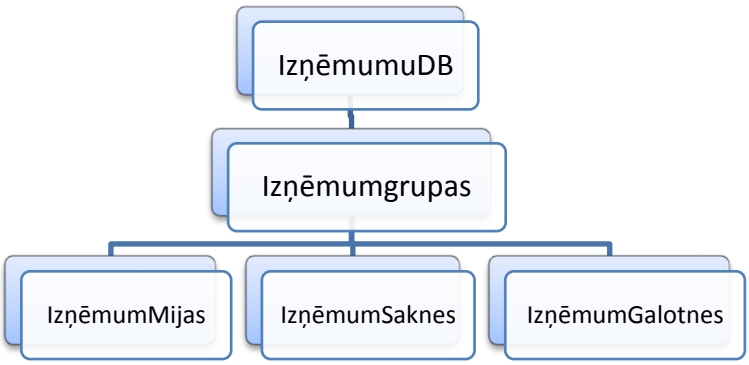
4.2.1. Datu bāzes struktūra

Vispirms tiks apskatīta datubāzes struktūra, kādā šie izņēmumi tiks glabāti, un tās veiksmīga izvēle ir par pamatu efektīvu algoritmu izveidei. Izņēmumi tiks glabāti XML (*eXtensible Markup Language*) failā šādu iemeslu dēļ:

- var tikt lietoti gan tiešsaistē, gan arī lokāli,
- tas ir izplatīts formāts, līdz ar to var tikt pielāgotas jau esošas dažādās programmēšanas valodās izstrādātas sistēmas,
- XML shēma nodrošina to, ka lietotie XML faili būs atbilstoši nosacījumiem.

Tā kā izņēmumi iedalās grupās ar līdzīgām īpašībām, tad arī datubāzē tie tiks glabāti tādās pašās grupās. Attēlā Nr. 4.1. tad var redzēt šo iedalījumu. Tādējādi katrā izņēmumgrupā tiks glabāti tikai attiecīgie izņēmumi. Šāda iedalījuma pa grupām viens no ieguvumiem ir

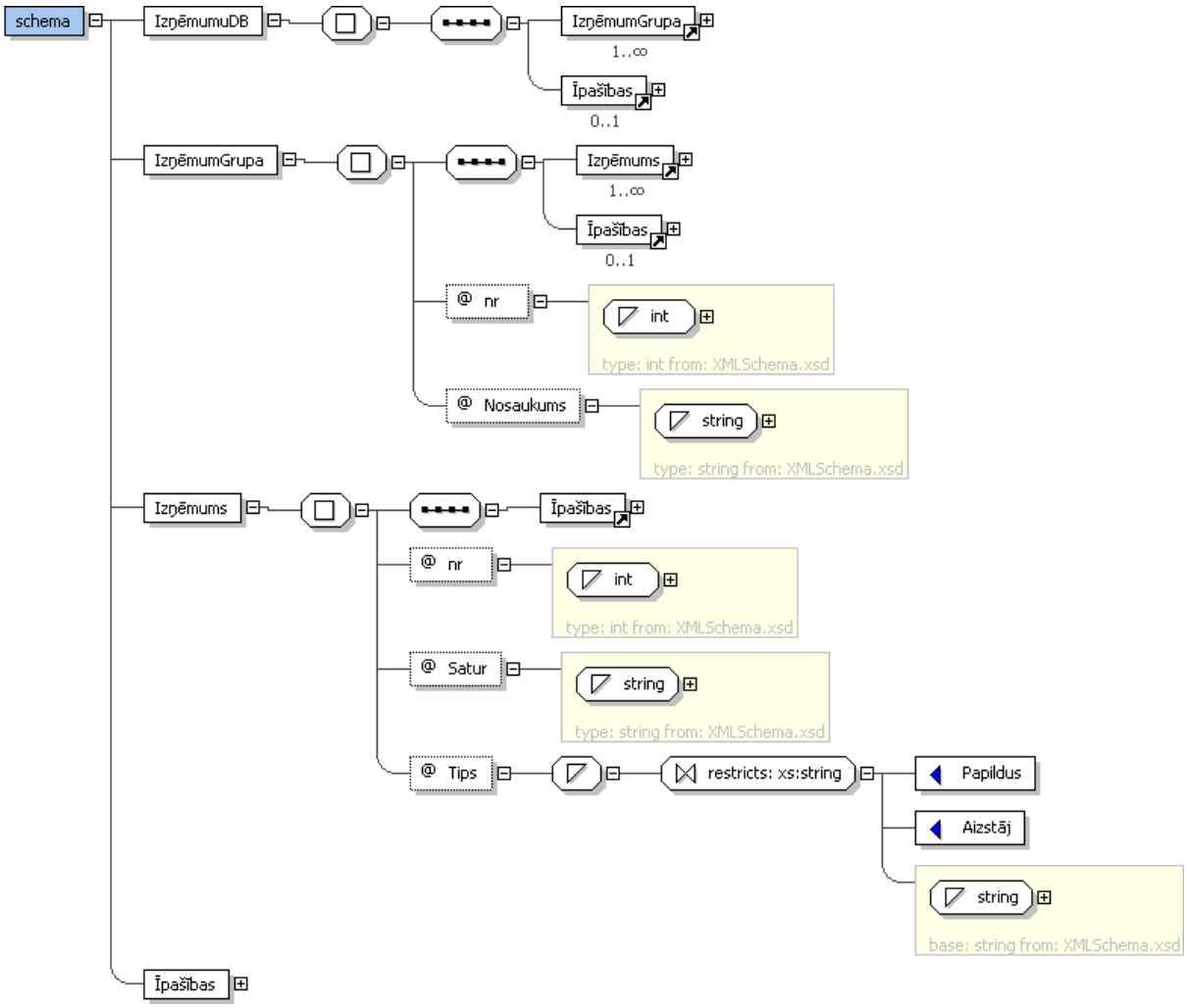
tāds, ka var meklēt tikai nepieciešamā veida izņēmumus, kuri ir vajadzīgi konkrētajā brīdī. Tāpat arī lietotājam ir vieglāk uztvert pēc kāda nosacījuma strukturētu informāciju, kas atvieglo jaunu izņēmumu pievienošanu datubāzei. Tas ir svarīgi, jo izņēmumgadījumu apstrāde tiek veidota ar mērķi, lai paši lietotāji varētu vienkārši pievienot jaunus vai rediģēt esošos izņēmumus. Tiesa, to vēl būtiskāk ietekmē grafiskā saskarne, bet tā vēl nav pilnībā izstrādāta un līdz ar to netiks šajā darbā apskatīta.



4.1. att. **Izņēmumu iedalījums pēc to grupām**

Pielikumā Nr. 1 var apskatīt XML shēmu [8], kurai jāatbilst izņēmumu datubāzes XML failiem. Pēc tās tad var vienkārši pārbaudīt, vai izveidotā datne satur atļautās birkas un atribūtus, kā arī to iespējamās vērtības. Attēlā Nr. 4.2. var aplūkot vizuālo reprezentāciju daļai XML shēmas. Tajā var redzēt, ka *IzņēmumuDB* ir XML dokumenta sakne, kurai attiecīgi var norādīt īpašības, ja nepieciešams. Savukārt, nākošajā līmenī var būt neierobežota daudzuma *IzņēmumGrupas*, kas satur atribūtus *nr* un *Nosaukums* (attiecīgi izņēmumgrupas numurs un nosaukums). Vēl zemākā līmenī atrodas *Izņēmumi*, kas satur atribūtus *nr*, *Satur* un *Tips*, kā arī zem sevis nākošajā līmenī satur birku *Īpašības*. Tāpat vēl var redzēt arī atribūtu tipus un to iespējamās vērtības. Savukārt, *Īpašību* atribūtus var apskatīt iepriekš pieminētajā pielikumā, jo to skaits ir pārāk liels, lai attēlotu šādā tipa diagrammā.

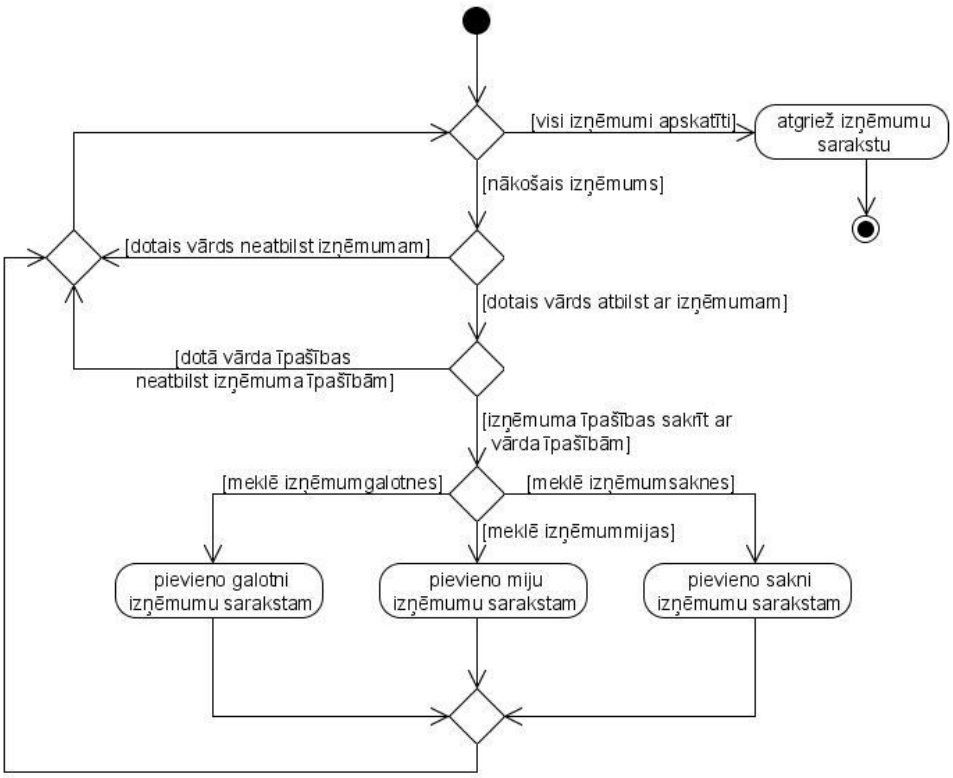
Tas, vai vārds ir izņēmums vai arī nav, tiks pārbaudīts pēc tā vai vārds beidzas ar datubāzē ievietoto simbolu virkni (atribūts *Satur*). Simbolu virkne ir izvēlēta tāpēc, ka ar to var beigties vairāki vārdi, bet leksēmu gadījumā būtu jānorāda vairākas, ja tām ir vienādas vārda beigas, kas apgrūtinātu jaunu izņēmumu pievienošanu. Savukārt, atribūts *Tips* nodrošina to, vai izņēmums aizstās standarta locījumu vai arī veidos papildus iespējamo vārdformu (attiecīgi vērtībām tad jābūt *Aizstāj* vai *Papildus*). Tas nepieciešams, piemēram, vārda *trīs* sintēzes gadījumā, kad tiek veidots papildus celms *trij* atsevišķos locījumos.



4.2. att. **Izņēmumu datubāzes XML shēma**

4.2.2. Izņēmumu apstrādes algoritms

Izņēmuma sakne, galotne vai mija tiek norādīta pie īpašībām un tiek izmantotas tikai tādā gadījumā, ja pārējās īpašības atbilst konkrētajam vārda locījumam. Tās var saturēt vārda skaitli, dzimti, vārdgrupas numuru u.c. vārdformu aprakstošu informāciju, tādējādi nodrošinot, ka pārējos gadījumos tiktu pielietota standarta vārdformu darināšana. Līdz ar to tikai norādītajā locījumā vai locījumos vārds tiks apstrādāts kā izņēmums, pievienojot norādīto sakni, galotni vai miju atgriežamajam sarakstam. 4.3. attēlā var redzēt vienkāršotu UML aktivitāšu diagrammu vispārīgai izņēmumgadījumu apstrādei.



4.3. att. **UML aktivitāšu diagramma izņēmumu apstrādei**

Galarezultātā tiek atgriezts saraksts ar mijām, galotnēm vai saknēm atkarībā no tā, kas tika meklēts. Tādējādi tiek panākts, ka pēc tam locīšanā tiks apskatīti visi izņēmumi, kas atbilst dotajam vārdam.

Izņēmumu meklēšanas apstrādes ātrdarbību ietekmē visbūtiskāk, tāpēc nepieciešams pēc iespējas efektīvāks un ātrāks risinājums. Vienkāršākais variants būtu apskatīt katru izņēmumu un pārbaudīt, vai vārds ar to beidzas. Tādējādi sliktākajā gadījumā meklēšana būtu lineārā laikā *O(n)*, līdz ar to ļoti atkarīga no izņēmumu skaita. Laika gaitā tas var būtiski pieaugt, tāpēc šāda pieeja nav īpaši efektīva.

Cits variants būtu izņēmumu glabāšana *heštabulā* (*hash table*), kurā meklēšana parasti notiek konstantā laikā *O(1)*. Tiesa, sliktākajā gadījumā tā var notikt lineārā laikā kā iepriekšējā variantā, bet šāda situācija ir maz ticama un liecina par nekorektu *hešfunkcijas* darbību. Izņēmumu apstrādē tiek izmantots šis variants, jo meklēšana konstantā laikā pēc izņēmumvārda normālos gadījumos ir ātrāka kā iepriekšējais variants. Tā kā izņēmumos tiek glabātas vārda beigas, tad meklēšana jāveic ne tikai pēc pilnā vārda, bet arī pēc visām simbolu virknēm, ar kurām tas var beigties. Tādējādi šis variants vēl ir atkarīgs arī no vārda garuma, bet tā kā tas ir daudz mazāks par visu iespējamo izņēmumu skaitu, tad veiktspēja būs labāka nekā pilnās pārlases gadījumā.

Pēc atbilstoša izņēmuma atrašanas vispirms tiek pārbaudīts, vai tā vārdgrupa ir tāda pati kā vārdam, kam tika meklēts šis izņēmums. Ja šis nosacījums izpildās, tad pēc tam tiek pārbaudīts, vai vārda zilbju skaits ir atbilstošs, ja vien tas ir norādīts pie izņēmuma īpašībām. Dalīšana zilbēs un līdz ar to arī to skaitīšana tiek nodrošināta šādā vienkāršā veidā:

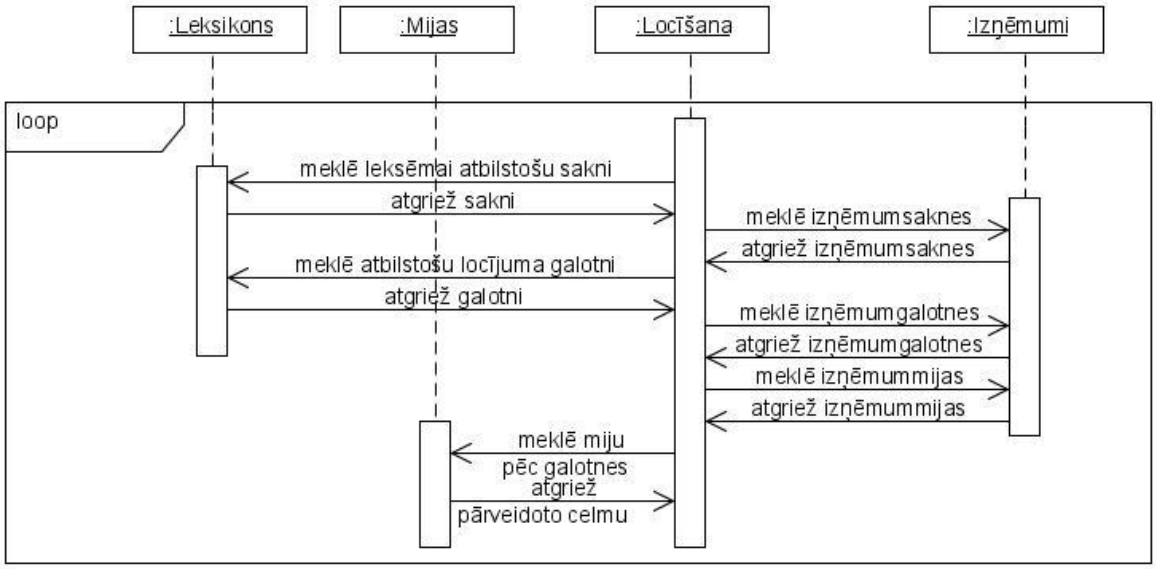
- atdalot priedēkļus;
- ja pēc patskaņa ir viens līdzskanis, tad dalījums ir starp abiem burtiem, piemēram, *kā–ja*;
- ja pēc patskaņa ir divi līdzskaņi, tad dalījums ir starp abiem līdzskaņiem, piemēram, *mar–ga*;
- ja pēc patskaņa ir trīs līdzskaņi, tad dalījums ir starp pirmo un pārējiem līdzskaņiem, piemēram, *kāp–slis*;
- jā pēc patskaņa ir četri līdzskaņi, tad dalījums ir pēc pirmajiem diviem līdzskaņiem, piemēram, *pulk–ste–nis*.

Ja arī zilbju skaits ir atbilstošs, tad visbeidzot tiek pārbaudīts, vai galotnes īpašības sakrīt ar tām īpašībām, kas ir norādītas pie izņēmuma. Tā kā katra šāda īpašība satur *nosaukums -> vērtība* simbolu virkņu pārus, tad tiek salīdzināts tikai tās īpašības, kurām sakrīt *nosaukums* (gan starp katras galotnes, gan katra izņēmuma īpašībām nevar būt vairāk kā viens šāds pāris ar vienu un to pašu *nosaukumu*). Ja tāds pāris ar noteiktu *nosaukumu* ir tikai vienā no abiem salīdzināmajiem īpašību sarakstiem, tad tas netiek ņemts vērā salīdzināšanas rezultātā. Tātad, ja eksistē kaut viens šāds gadījums, kad pāra *nosaukumi* sakrīt, bet *vērtības* nesakrīt, tad izņēmums neatbilst konkrētajam locījumam un netiek atgriezts.

4.2.3. Rezultāts

Rezultātā iegūtais risinājums nodrošina pietiekami elastīgu izņēmumu apstrādi, lai ar tā palīdzību varētu korekti sintezēt vārdformas visiem vārdiem, kuri netiek locīti atbilstoši vispārējiem likumiem. Līdz ar to vēl būs nepieciešams izstrādāt grafisko saskarni ērtākai izņēmu ievadei.

4.4. attēlā var apskatīt, kā vārdformu sintēzē tiek iesaistīta izņēmu apstrāde, pieņemot, ka locīšanai ir padota kāda noteikta leksēma. Var redzēt, kādā secībā notiek izņēmumu un parasto gadījumu meklēšana – vispirms tiek sameklēts parastais, bet pēc tam atkarībā no tā tiek meklēts izņēmums, tādējādi nodrošinot, ka, piemēram, izņēmumsakne tiks meklēta atkarībā no iepriekš atrastās saknes.



4.4. att. **Secību diagramma izņēmumu apstrādei**

Savukārt, attēlā Nr. *4.5*. var apskatīt paraugu izņēmumu XML failam. Var redzēt, ka tas satur visas trīs izņēmumgrupas, kā arī pa vienam izņēmumam katrā grupā. Turpinājumā neliels ieskats, kā šādi izņēmumi tiks apstrādāti.

```
<?xml version="1.0" encoding="UTF-8"?>
<IzņēmumuDB>
<IzņēmumGrupa nr="1" nosaukums="IzņēmumMijas">
<Izņēmums nr="1" Satur="valdis" Tips="Aizstāj">
<Īpašības Vārdgrupas_nr="3" Zilbes="2" Mijas_nr="0"/>
</Izņēmums>
</IzņēmumGrupa>
<IzņēmumGrupa nr="2" nosaukums="IzņēmumGalotnes">
<Izņēmums nr="2" Satur="trī" Tips="Aizstāj">
<Īpašības Vārdgrupas_nr="23" Galotne="s" Locījums="Lokatīvs"/>
</Izņēmums>
</IzņēmumGrupa>
<IzņēmumGrupa nr="3" nosaukums="IzņēmumSaknes">
<Izņēmums nr="3" Satur="trī" Tips="Papildus">
<Īpašības Vārdgrupas_nr="23" Sakne="trij" Locījums="Lokatīvs"/>
</Izņēmums>
</IzņēmumGrupa>
</IzņēmumuDB>
```

4.5. att. **Paraugs izņēmumu XML failam**

Kā pirmā tiks apskatīta izņēmummija, kas satur izņēmumu vārdiem, kas beidzas ar *valdis*. Ar šādu simboli virkni beidzas II deklinācijas lietvārdi, kuriem eksistē mija vienskaitļa ģenitīvā un daudzskaitļa ģenitīvā un lokatīvā. Tā kā divzilbju vārdiem, kas beidzas ar *valdis,* attiecīgās mijas nav, tad arī pie īpašībām tiek norādīts, ka vārda zilbju skaitam jābūt 2, tāpat vārdgrupas numurs jābūt 3 (II deklinācijas lietvārdi). Visbeidzot tiek arī norādīts mijas numurs, kur 0 apzīmē to, ka mijas nav.

Pārējās divās izņēmumgrupās tiek norādīti izņēmumi vienam un tam pašam vārdam. Var redzēt, ka lokot vārdu *trīs*, tā lokatīvā tiks veidota gan papildus sakne *trij*, gan arī papildus galotne *s,* kas gan aizstāj parasto galotni. Vispirms tiek meklēti izņēmumi saknēm, līdz ar to lokatīva locījumā šādā gadījumā būs divas saknes – *trī* un *trij*. Pēc tam tiek pārbaudīts, vai galotnēm neeksistē izņēmumi. Tā kā saknei *trī* ir izņēmumgalotne *s*, tad tiek aizstāta standarta galotne un iegūta vajadzīgā vārdforma. Otras saknes *trij* gadījumā arī tiek meklētas izņēmumgalotnes, bet tā kā tādu nav, tad tiek piekārtota standarta galotne *os,* iegūstot vēl vienu vārdformu *trijos*. Jāievēro, ka šajā gadījumā skaitļa vārdi netika dalīta pa dzimtēm, jo abās lokatīva locījums ir *trīs*, bet otrai vārdformai sakne sakrīt – *trijos, trijās*.

REZULTĀTS UN SECINĀJUMI

Latviešu valodas vārdformu sintēze pirmajā brīdī šķiet ļoti vienkārša – atliek tikai vārdam piekārtot attiecīgās galotnes un visi locījumi ir iegūti. Tāpēc darbā tika apskatītas gan dažādu vārdšķiru vārdu, gan arī daudzo izņēmumu locīšanas raksturīgākās iezīmes, kas manāmi sarežģī šo procesu. Līdz ar to lasītājs var iepazīties ar visām latviešu valodas vārdformu sintēzes īpatnībām.

Vispirms tika apskatīta vārda locīšana atkarībā no vārdšķiras, parādot kādas ir to būtiskākās iezīmes vārdformu darināšanai. Rezultātā, izmantojot šos apskatītos algoritmus, tika izveidots sintezators, kas ir saderīgs ar jau iepriekš izstrādāto latviešu valodas automatizēto morfoloģisko analizatoru. Tādējādi ir iegūta sistēma, kas ļauj gan analizēt vārdu, gan arī sintezēt tam atbilstošās vārdformas, izmantojot analīzē iegūto informāciju, kas arī bija darba galvenais mērķis.

Savukārt, viens no darba galvenajiem uzdevumiem bija izstrādāt elastīgu izņēmumu apstrādi, kas ļautu pareizi sintezēt tos vārdus, kas netiek locīti atbilstoši vairumam pārējo attiecīgās vārdgrupas vārdu. Tādējādi darbā tika apskatīta gan datubāzes struktūra, kas sevī iekļāva arī tās XML shēmas izstrādi, gan arī izveidots efektīvs risinājums šādu gadījumu locīšanai pēc norādītājām īpašībām. Rezultātā tika izstrādāts papildinājums latviešu valodas vārdformu sintezatoram, kas apstrādā lietotāju pievienotus izņēmumus. Tātad šis darba uzdevums ir sasniegts.

Rezultātā izstrādātais modulis kopā ar morfoloģisko analizatoru ļauj lietotājam veikt vārdu analīzi un pēc tam veikt tā vārdformu sintēzi. Ja tiek atklāts, ka vārds netiek locīts pareizi kā izņēmumgadījums, tad pats lietotājs var ievadīt izņēmumu ar atbilstošām īpašībām, un tādējādi nav nepieciešamas izmaiņas moduļa izejas kodā. Tas arī ir būtiskākais šī darba ieguvums, jo visās iepriekš izstrādātajās sistēmās šādā gadījumā būtu jāveic vai nu izmaiņas datubāzes struktūrā vai jālabo pati sistēma, kas bieži vien var izrādīties ļoti sarežģīti un laikietilpīgi.

Turpmākās iespējas

Kā svarīgākais no turpmākajiem darbiem ir izņēmumgadījumu apstrādes grafiskās saskarnes izveide, kas ļautu lietotājam ātri un ērti pievienot jaunus izņēmumus un meklēt starp esošajiem. Tas ir būtiski, jo pašlaik darbā izstrādājam risinājumam pietrūkst lietojamības ērtuma, jo tas vairāk tika veidots ar mērķi iekļaut elastīgu izņēmumu apstrādi vārdformu sintēzē.

Tādējādi nākošais solis būtu izņēmumu datubāzes papildināšana, jo pašlaik tā satur tikai pašus nepieciešamākos, kas atbild par miju neesamību konkrētiem vārdiem, kā arī iepriekš darbā apskatītos spilgtākos izņēmumus. Rezultātā vārdformu sintēzes precizitāte tiks būtiski uzlabota, jo pašlaik šādi vārdi tiek locīti kā standarta gadījumi, ja nav norādīts, ka tie ir jādarina kā savādāk.

IZMANTOTĀ LITERATŪRA

1. **Rubīna Ai.** *Latviešu valodas rokasgrāmata*. Rīga : Zvaigzne ABC, 2005. 187 lpp.
2. **Ceplīte, B., Ceplītis, L.** *Latviešu valodas praktiskā gramatika*. Rīga : Zvaigzne ABC, 1997. 232 lpp.
3. *Valodas uzziņas* [tiešsaiste] Rīga : SIA Tilde, 2002 – [atsauce 02.05.2008]. Pieejams: http://www.letonika.lv/groups/default.aspx?g=5
4. **Pinnis, M.** Uzlabota latviešu valodas analīzes un sintēzes sistēma : kursa darbs. LU Fizikas un matemātikas fakultāte. Rīga : Latvijas Universitāte, 2007. 62.lpp.
5. **Paikens, P.** Lexicon-based morphological analysis of Latvian language **In:** *3rd Baltic Conference on Human Language Technologies*, Kaunas, Lithuania, October 4-5, 2007. 6 p.
6. *UML 2.1.2 Specification* [tiešsaiste] - [atsauce 27.04.2008.]. Pieejams: http://www.omg.org/spec/UML/2.1.2/Infrastructure/PDF
7. *Java 2 Platform, Standard Edition, v1.4.2 API Specification* [tiešsaiste] – [atsauce 30.04.2008.]. Pieejams: http://java.sun.com/j2se/1.4.2/docs/api/
8. *XML Schema.* [tiešsaiste] - [atsauce 21.04.2008.]. Pieejams: http://www.w3.org/XML/Schema

PIELIKUMS 1 – XML SHĒMA IZŅĒMUMU GLABĀŠANAI

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
   <xs:element name="IzņēmumuDB">
      <xs:complexType>
         <xs:sequence>
            <xs:element ref="IzņēmumGrupa" maxOccurs="unbounded"/>
            <xs:element ref="Īpašības" minOccurs="0" maxOccurs="1"/>
         </xs:sequence>
      </xs:complexType>
   </xs:element>
   <xs:element name="IzņēmumGrupa">
      <xs:complexType>
         <xs:sequence>
            <xs:element ref="Izņēmums" maxOccurs="unbounded"/>
            <xs:element ref="Īpašības" minOccurs="0" maxOccurs="1"/>
         </xs:sequence>
         <xs:attribute name="nr" type="xs:int"/>
         <xs:attribute name="Nosaukums" type="xs:string"/>
      </xs:complexType>
   </xs:element>
   <xs:element name="Izņēmums">
      <xs:complexType>
         <xs:sequence>
            <xs:element ref="Īpašības" minOccurs="1" maxOccurs="1" />
         </xs:sequence>
         <xs:attribute name="nr" type="xs:int"/>
         <xs:attribute name="Satur" type="xs:string"/>
         <xs:attribute name="Tips">
            <xs:simpleType>
               <xs:restriction base="xs:string">
                  <xs:enumeration value="Papildus"/>
                  <xs:enumeration value="Aizstāj"/>
               </xs:restriction>
            </xs:simpleType>
         </xs:attribute>
      </xs:complexType>
   </xs:element>
   <xs:element name="Īpašības">
      <xs:complexType>
         <xs:attribute name="Vārdšķira">
            <xs:simpleType>
               <xs:restriction base="xs:string">
                  <xs:enumeration value="Apstākļa vārds"/>
                  <xs:enumeration value="Darbības vārds"/>
                  <xs:enumeration value="Divdabis"/>
                  <xs:enumeration value="Izsauksmes vārds"/>
                  <xs:enumeration value="Lietvārds"/>
                  <xs:enumeration value="Partikula"/>
                  <xs:enumeration value="Pieturzīme"/>
                  <xs:enumeration value="Prievārds"/>
```

```
            <xs:enumeration value="Saiklis"/>
            <xs:enumeration value="Skaitļa vārds"/>
            <xs:enumeration value="Vietniekvārds"/>
            <xs:enumeration value="Īpašības vārds"/>
         </xs:restriction>
      </xs:simpleType>
   </xs:attribute>
   <xs:attribute name="Vietniekvārda_tips">
      <xs:simpleType>
         <xs:restriction base="xs:string">
            <xs:enumeration value="Personu"/>
         </xs:restriction>
      </xs:simpleType>
   </xs:attribute>
   <xs:attribute name="Transitivitāte">
      <xs:simpleType>
         <xs:restriction base="xs:string">
            <xs:enumeration value="Nepārejošs"/>
            <xs:enumeration value="Pārejošs"/>
         </xs:restriction>
      </xs:simpleType>
   </xs:attribute>
   <xs:attribute name="Skaitļa_vārda_tips">
      <xs:simpleType>
         <xs:restriction base="xs:string">
            <xs:enumeration value="Kārtas"/>
            <xs:enumeration value="Pamata"/>
         </xs:restriction>
      </xs:simpleType>
   </xs:attribute>
   <xs:attribute name="Skaitlis">
      <xs:simpleType>
         <xs:restriction base="xs:string">
            <xs:enumeration value="Daudzskaitlis"/>
            <xs:enumeration value="Vienskaitlis"/>
         </xs:restriction>
      </xs:simpleType>
   </xs:attribute>
   <xs:attribute name="Saikļa_tips">
      <xs:simpleType>
         <xs:restriction base="xs:string">
            <xs:enumeration value="Pakārtojuma"/>
            <xs:enumeration value="Sakārtojuma"/>
         </xs:restriction>
      </xs:simpleType>
   </xs:attribute>
   <xs:attribute name="Persona" type="xs:int"/>
   <xs:attribute name="Novietojums">
      <xs:simpleType>
         <xs:restriction base="xs:string">
            <xs:enumeration value="Pirms"/>
            <xs:enumeration value="Pēc"/>
```

```
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="Noteiktība">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="Nenoteiktā"/>
        <xs:enumeration value="Noteiktā"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="Lokāmība">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="Nelokāms"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="Locījums">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="Akuzatīvs"/>
        <xs:enumeration value="Datīvs"/>
        <xs:enumeration value="Lokatīvs"/>
        <xs:enumeration value="Nominatīvs"/>
        <xs:enumeration value="Vokatīvs"/>
        <xs:enumeration value="Ģenitīvs"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="Lietvārda_tips">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="Īpašvārds"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="Leksēmas_nr" type="xs:int"/>
  <xs:attribute name="Laiks">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="Nākotne"/>
        <xs:enumeration value="Pagātne"/>
        <xs:enumeration value="Tagadne"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="Konjugācija" type="xs:int"/>
  <xs:attribute name="Izteiksme">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="Atstāstījuma"/>
```

```
                <xs:enumeration value="Nenoteiksme"/>
                <xs:enumeration value="Pavēles"/>
                <xs:enumeration value="Vajadzības"/>
                <xs:enumeration value="Īstenības"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
          <xs:attribute name="Galotnes_nr" type="xs:int"/>
          <xs:attribute name="Dzimte">
            <xs:simpleType>
              <xs:restriction base="xs:string">
                <xs:enumeration value="Sieviešu"/>
                <xs:enumeration value="Vīriešu"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
          <xs:attribute name="Deklinācija">
            <xs:simpleType>
              <xs:restriction base="xs:string">
                <xs:enumeration value="1"/>
                <xs:enumeration value="2"/>
                <xs:enumeration value="3"/>
                <xs:enumeration value="4"/>
                <xs:enumeration value="5"/>
                <xs:enumeration value="6"/>
                <xs:enumeration value="Nelokāms"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
          <xs:attribute name="Atgriezeniskums">
            <xs:simpleType>
              <xs:restriction base="xs:string">
                <xs:enumeration value="Jā"/>
                <xs:enumeration value="Nē"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
          <xs:attribute name="Vārdgrupas_nr" type="xs:int"/>
          <xs:attribute name="Mijas_nr" type="xs:int"/>
          <xs:attribute name="Galotne" type="xs:string"/>
          <xs:attribute name="Sakne" type="xs:string"/>
        </xs:complexType>
      </xs:element>
    </xs:schema>
```

Bakalaura darbs

Ar savu parakstu apliecinu, ka pētījums veikts patstāvīgi, izmantoti tikai tajā norādītie informācijas avoti un iesniegtā darba elektroniskā kopija atbilst izdrukai. Piekrītu sava darba publicēšanai internetā.

Autors: _____
(Autora paraksts)

Ar savu parakstu apliecinu, ka esmu lasījis augšminēto bakalaura darbu un atzīstu to par **piemērotu/nepiemērotu** (nevajadzīgo svītrot) aizstāvēšanai Latvijas Universitātes datorzinātņu bakalaura studiju programmas gala pārbaudījuma komisijas sēdē.

Darba vadītājs(-ja): _____
(Vadītāja paraksts)

Darbs iesniegts Datorikas nodaļā _____.
(Iesniegšanas datums)

Ar šo es apliecinu, ka darba elektroniskā versija ir augšupielādēta LU informatīvajā sistēmā.
Metodiķe: _____.
(Metodiķes paraksts)

Recenzents: _____
(Recenzenta paraksts)

Darbs aizstāvēts bakalaura darbu gala pārbaudījuma komisijas sēdē

_____ prot. Nr. _____, vērtējums _____
(Darba aizstāvēšanas datums)

Komisijas sekretārs: _____
(Sekretāra paraksts)