

PRIKLAUSOMYBIŲ GRAMATIKA PAGRĮSTŲ LIETUVIŲ KALBOS SINTAKSINIŲ TAISYKLIŲ IŠGAVIMAS IŠ *DABARTINĖS LIETUVIŲ KALBOS TEKSTYNO*

Gintarė Grigonytė
Kauno Technologijos universitetas

Erika Rimkutė
Vytauto Didžiojo universitetas

Straipsnyje aprašoma Priklausomybių gramatika (DG) pagrįstų lietuvių kalbos sintaksinių taisyklių išgavimo iš *Dabartinės lietuvių kalbos teksto* metodologija. Nagrinėjami pagrindiniai taisyklių specifavimo parametrai, aprašomas išgautų taisyklių klasifikavimo procesas.

1. Įvadas

Kompiuterizuojant kalbas, skiriami du pagrindiniai etapai: automatinė morfologinė analizė ir automatinė sintaksinė analizė (ASA). Po šių etapų seka semantinė analizė.

Lietuvių kalbos automatinė morfologinė analizė – jau įvykęs kalbos kompiuterizavimo etapas. Yra sukurtas morfologinis analizatorius *Lemuoklis* [6]; Vytauto Didžiojo universiteto Kompiuterinės lingvistikos centre yra parengtas 1 mln. žodžių pusiau automatiškai morfologiškai anotuotas lietuvių kalbos tekstynas [7].

Tolesnis automatinės lietuvių kalbos analizės etapas – ASA, kurios metu naudojamų taisyklių išgavimą pristatome šiame straipsnyje. Pirmasis ASA etapas apėmė automatinį veiksmažodžių grupių atpažinimą [1]. Ta pati metodologija taikoma kitų kalbos dalių automatinei analizei [2].

Straipsnyje aprašyta, kaip iš 100 mln. žodžių *Dabartinės lietuvių kalbos teksto* buvo išgautos sintaksinės lietuvių kalbos taisyklės (žr. 3 skyrių). Tos taisyklės skirstomos į du lygius: žodžių junginius ir tų junginių kombinacijas aprašančias taisykles (žr. 2 skyrių). Taip pat straipsnyje aprašomi parametrai, reikalingi pritaikant priklausomybių gramatiką lietuvių kalbai.

2. Lietuvių kalbos sintaksinių taisyklių formalizavimas

Lietuvių kalbos sintaksinėms taisyklėms sudaryti buvo panaudoti tokie lingvistiniai resursai: Vytauto Didžiojo universiteto Kompiuterinės lingvistikos centre paruoštas pusiau automatiškai morfologiškai anotuotas lietuvių kalbos tekstynas, susidedantis iš 1 mln. žodžių [7], *Dabartinės lietuvių kalbos tekstynas*, sudarytas iš 100 mln. žodžių [3] ir morfologinis analizatorius *Lemuoklis* [6].

Sintaksinėms taisyklėms aprašyti panaudota priklausomybių gramatika [3]. Remiantis minėta gramatika, yra svarbu nurodyti, kuris žodis yra valdantysis, o kuris valdomasis. Tokie žodžių ryšiai yra išreiškiami rodyklėmis, pvz.: *įdomi* ← *konferencija*. Žodis, iš kurio eina rodyklė (šiuo atveju – *konferencija*), yra pagrindinis, arba valdantysis, o tas žodis, į kurį nukreipta rodyklė (šiuo atveju – *įdomi*), valdomasis, t. y. priklausantis nuo pagrindinio.

2.1 Sintaksinių taisyklių parametrai

Aprašant pagrindinius sintaksinius junginius, neužtenka vien sintaksinių taisyklių, pvz.: *bdvr/dlv/sktv/įvrd* (G, N, C) + *dktv* (G, N, C); *prln* (*su, sulig, ties, po*) + *dktv/bdvr/įvrd/sktv/dlv* I_n^1 . Reikalingi tokie papildomi parametrai: sintaksinė priklausomybė, įsiterpimas, žodžių tvarka ir prioritetas. Toliau pristatysime kiekvieną iš jų.

Priklausomybės žymimos rodyklėmis. Galimi 2 priklausomybės tipai (žr. 1 pav.) a) tipo priklausomybė reiškia, kad *žodis1* valdo *žodis2*, pvz.: *rašyti* → *straipsnį*, o b) tipo priklausomybė, nurodo, kad *žodis2* yra valdantysis, pvz.: *Jono* ← *mašina*.

¹ Pirmoji taisyklė reiškia, kad jei būdvardis, dalyvis, skaitvardis ar įvardis su daiktavardžiu yra suderinti gimine (taisyklėje žymima G), skaičiumi (taisyklėje žymima N) ir linksniu (taisyklėje žymima C), tada tai bus vienas junginys. Antroji taisyklė nurodo, kad prielinksniai *su, sulig, ties* ir *po* jungiasi su daiktavardžių, būdvardžių, įvardžių ar skaitvardžių įnagininku.



1 pav. Priklausomybių tarp žodžių tipai.

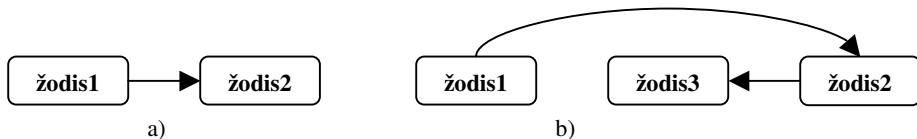
Rašant priklausomybių gramatika pagrįstas taisyklės dar vienas svarbus parametras yra **žodžių tvarka**. Lietuvių kalbos žodžių tvarka yra laisva; tai apsunkina žodžių ryšių nustatinėjimą. Vis dėlto dažniausiai vartojami SVO² tipo sakiniai, pvz.: *Jonas (S) skaito (V) knygą (O)*. 2 pav. matyti, kad priklausomybė išlieka ta pati, t. y. *žodis1* valdo *žodis2*, bet skiriasi žodžių tvarka, pvz.: *mezga → močiutė* vs. *močiutė ← mezga*.



2 pav. Galimi žodžių tvarkos modeliai.

Kitas parametras nustatant sintaksinius junginius yra **įsiterpimas**. Paprastai įsiterpintys žodžiai nepakeičia sintaksinio ryšio – priklausomybės, tačiau modifikuoja pačią struktūrą, todėl svarbu nustatyti, kokie žodžiai ir per kiek pozicijų gali įsiterpti analizuojamame junginyje. Bendra tendencija yra tokia, kad vardažodžių junginiuose yra linkę įsiterpti kiti vardažodžiai irrieveiksmai, niekada neįsiterpia asmenuojamosios vksm formos. Didele formų, žodžių tvarkos ir įsiterpiančių žodžių įvairove pasižymi veiksmažodžių junginiai.

3 pav. matyti, kad *žodis1* valdo *žodis2*, ir ši priklausomybė nesikeičia tarp jų įsiterpus trečiam žodžiui, pvz.: *rinkti → uogas* vs. *rinkti → [saldžias ← uogas]*.



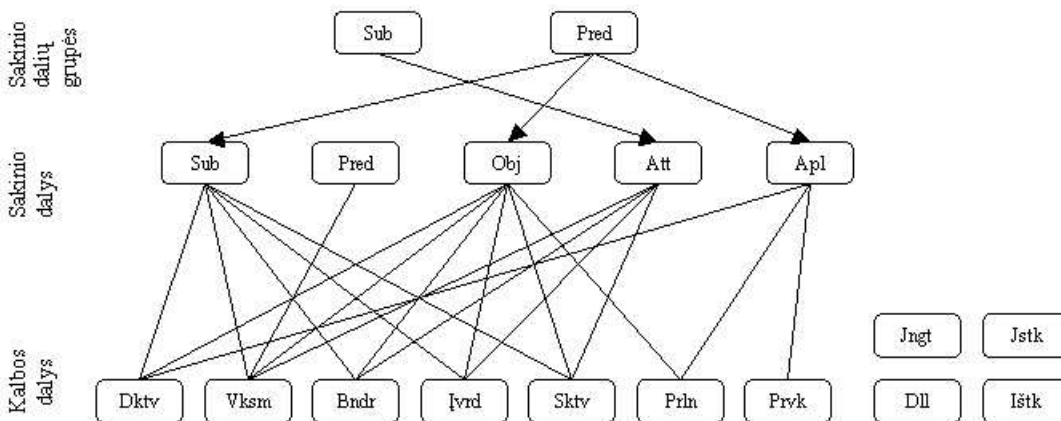
3 pav. Įsiterpimo ir priklausomybės tarp žodžių ryšys.

Ketvirtasis parametras – **prioritetas** – bus svarbus tolesniuose ASA etapuose. Taisyklių prioritetas parodo, kuri tos pačios grupės taisyklė yra svarbesnė. Pavyzdžiui, dažniausiai būdvardis yra vartojamas prepoziciskai, pvz.: *graži diena*, tačiau kartais dėl inversijos gali būti pavartotas postpoziciskai, pvz.: *diena graži*, o priklausomybė išlieka ta pati – šiuo atveju *diena* yra valdantysis žodis. Pagrindine taisykle bus laikoma būdvardis + daiktavardis junginys.

2.2 Sintaksinių taisyklių lygiai

Lietuvių kalbos sintaksinės taisyklės skirstomos į du lygius: žodžių junginių taisyklės (subjekto (Sub.), predikato (Pred.), objekto (Obj.), aplinkybių (Apl.) ir atributo (Att.)) ir tų junginių kombinacijas (subjekto ir predikato ryšys, subjekto ir atributo ryšys, predikato ir objekto ryšys, predikato ir aplinkybių ryšys) (žr. 4 pav.).

² S – subjektas, V – veiksmažodis, O – objektas.



4 pav. Lietuvių kalbos sintaksinių taisyklių lygiai.

3. Taisyklių ištraukimas iš *Dabartinės lietuvių kalbos teksto*

Taisyklių išgavimas iš *Dabartinės lietuvių kalbos teksto* remiasi adaptuotu *pattern recognition* metodu [5]. Automatinė analizė turi šiuos etapus: tekstinių vienetų atpažinimas, automatinė morfologinė žodžių analizė, klasifikavimas. Toliau trumpai aptarsime kiekvieną iš jų.

Tekstiniu vienetu laikome sakinio fragmentą, kuris yra tam tikrą kalbos dalį reprezentuojantis žodžių junginys, pavyzdžiui, daiktavardį, būdvardį ir pan. Tekstiniai vienetai ištraukiami tokiu būdu: skaitomi elektroniniai tekstai, viena kuri nors kalbos dalis laikoma junginio centru ir analizuojami pasirinktu atstumu nuo jos nutolę žodžiai, pavyzdžiui, centras – daiktavardis, atstumas – du žodžiai. Šio etapo tikslas – išrinkti tekstinius vienetus, kurie turi padengti visas vienuolika lietuvių kalbos dalių. Šio etapo pabaigoje turime pradinį taisyklių sąrašą.

Gautas taisyklių sąrašas toliau **analizuojamas morfologiškai**. Morfologinė informacija svarbi taisyklių parametru specifikuojimui, jos metu nustatoma, kokios yra kitos tekstinį vienetą sudarančios kalbos dalys, gaunama informacija apie linksnius, gimines, skaičius ir kitus lietuvių kalbos analizei svarbias morfolgines pažymas, formuojama pirminė taisyklės specifikuojama.

Šio etapo tikslas – išanalizuoti tekstinių vienetų žodžius, naudojant turimą morfolgine informaciją aprašyti tekstinius vienetus.

Morfologinės analizės metu gaunami rezultatai:

<w l="eiti(eina,ėjo)" m="finite verb, infinitive" l="eiti(eina,ėjo)" m="finite verb, participle>eiti<w>

Klasifikavimas – paskutinis automatinio taisyklių ištraukimo iš teksto etapas. Po morfologinės analizės visi tekstiniai vienetai laikomi pirminėmis taisyklėmis, kurios klasifikuojamos pagal tekstinio vieneto ilgį, ir galimus kalbos dalių ryšius: subjekto, predikato, objekto, atributo arba aplinkybės.

Tolesniame taisyklių kūrimo etape junginiai ir juos aprašančios pirminės taisyklės yra peržiūrimos eksperto, netinkami junginiai atmetami, pildoma trūkstama morfolgine informacija. Po eksperto peržiūros pagrindinėse taisyklių grupėse atliekamas papildomas grupavimas. Šis grupavimas skiriasi nuo anksčiau aptarto tuo, kad grupuojamos formalios junginių taisyklės be konkrečių, jas atstovaujančių pavyzdžių, į kuriuos buvo atsižvelgiama pradiniam klasifikavimo etape.

Gauti rezultatai dar kartą tikrinami eksperto. Peržiūros metu atrenkamos galutinės sintaksinės taisyklės, kurios plačiau aptariamos [2].

Išvados

Straipsnyje analizuotas lietuvių kalbos sintaksinių taisyklių išgavimo iš *Dabartinės lietuvių kalbos teksto* metodas remiasi *pattern recognition* būdu. Aprašyta sintaksinių taisyklių specifika, būtinieji parametrai ir klasifikacijos procesas. ASA taisyklių išgavimo metodologija gautos taisyklės bus panaudotos automatiniam sintaksiniame analizatoriuje.

Literatūros sąrašas

- [1] **G. Grigonytė, E. Rimkutė.** Automatinis lietuvių kalbos veiksmažodžių grupių atpažinimas. *Konferencijos „Informacinės technologijos 2005“ pranešimų medžiaga*, 2005, 315–320.
- [2] **G. Grigonytė, E. Rimkutė.** Formal Specifications for a Dependency Grammar of the Lithuanian Language. *Konferencijos „The Second Baltic Conference on Human Language Technologies“ pranešimų medžiaga*, 2005, 237–242.
- [3] **R. Marcinkevičienė.** Tekstynų lingvistika (teorija ir praktika). *Darbai ir Dienos*, 2000, Nr. 24, 7–64.
- [4] **I. Mel'nik.** Dependency Syntax: Theory and Practice. *Albany: State University of New York Press*, 1988.
- [5] **S. Theodoridis, K. Koutroumbas,** Pattern recognition. *Athens: Academic Press*, 1999
- [6] **V. Zinkevičius.** Lemuoklis – morfologinei analizei. *Darbai ir dienos*, 2000, Nr. 24, 245–273.
- [7] **V. Zinkevičius, V. Daudaravičius, E. Rimkutė.** The Morphologically Annotated Lithuanian Corpus. *Konferencijos „The Second Baltic Conference on Human Language Technologies“ pranešimų medžiaga*, 2005, 365–370.

The Extraction of DG Syntactic Rules of the Lithuanian Language

The need of an automatic Lithuanian language syntactic analysis was the background of this research. Concerning Lithuanian language processing, there are some key works on morphology level, but syntactical analysis is still lagging behind. That is the main reason why we consider formal specifications of Dependency Grammar being important for Lithuanian language processing. Our approach is based on corpus-based pattern recognition methods that let us extract, and classify Dependency Grammar rules.