TARTU ÜLIKOOL
MATEMAATIKA-INFORMAATIKATEADUSKOND
Arvutiteaduse instituut
Informaatika eriala

Krista Liin

Reeglipõhine komavigade tuvastaja eestikeelsetele tekstidele

Magistritöö (20 AP)

Juhendaja: Kaili Müürisep
Keeletehnoloogia vanemteadur

TARTU 2008

Sisukord

Sissejuhatus

Käesoleva töö eesmärgiks on luua reeglite baas eesti keele grammatikakorrektori prototüübile. Töös keskendun komavigade tuvastamisele, jättes kõrvale ühildumis-, kokku-lahkukirjutamis- ning muud grammatikavead. Eesti keele komakasutus on küllaltki rangelt ette määratud, kuid sageli eksitakse kirjutamisel just selles valdkonnas.

Reeglite loomisel oli eesmärgiks luua võimalikult täpsed reeglid ehk püüda vältida valealarmide hulka nii palju kui võimalik. Eelistatavam on, et keerulistes lausetes jääb mõni viga märkimata, kui et tegelikult korrektne lause tõstetakse vigasena esile. Viimasel juhul võivad ebakindlamad kasutajad, näiteks keeleõppijad või lapsed, usaldada pigem keeletarkvara kui iseennast ja oma laused valeks parandada.

Eesti keelele on loodud kirjutaja abivahenditena kasutatamiseks õigekirjakontrollija ehk speller ja tesaurus. Grammatikakorrektor oleks järgmine samm, mis leiaks vead üles ka lausetasandil ning aitaks kirjutamisel vältida grammatiliselt ebakorrektseid lauseid. Viimane oleks lisaks tavakasutajatele oluline ka näiteks keeleõppeprotsessis.

Grammatikakorrektori loomise vajadust toonitatakse muuhulgas ka riiklikus programmis „Eesti keele keeletehnoloogiline tugi (2006-2010)". Grammatikakorrektori ning muu süntaktilist infot kasutava keeletarkvara loomiseks on selle programmi raames käivitatud projekt „Süntaksianalüüsil põhinev keeletarkvara ning selle arendamiseks vajalikud keeleressursid”.

Eesti keelele on grammatikakorrektori prototüüpi loodud varem kahel korral, Lauri Lundi ja Kristiina Kure bakalaureusetööde raames. Mõlemal juhul oli tegu programmeerimiskeeles Java realiseeritud testversioonidega, mida treeniti ja katsetati küllaltki väikesel korpusel, kõigest paarikümnel konstrueeritud lausel. Antud töös loodud grammatikakorrektor erineb eelmistest katsetest nii kasutatava korpuse kui ka aluseks oleva formalismi poolest.

Antud töö eesmärgiks polnud koostada mitte niivõrd kontrollija, vaid tuvastaja, mis leiab küll vead üles, kuid ei soovita parandatud, korrektset lauset. Samas on võimalik olemasolevate reeglite baasil luua ka korrektor.

Komavigade tuvastamisel kasutatakse nii morfoloogilist kui süntaktilist infot, mille märgendasid eeltöötluses kasutatud morfoloogiline märgendaja ja ühestaja ning süntaksianalüsaator. Baastekstina on reeglite loomisel kasutusel internetikommentaaridest kogutud komavigadega laused.

Grammatikakorrektor on realiseeritud kitsenduste grammatika reeglitena. Antud formalismi valikule suunas asjaolu, et eesti keele morfoloogiline ühestaja ning süntaksianalüsaator on välja arendatud kitsenduste grammatikat kasutades, seega oleks lihtne lisada järgmise moodulina grammatikakorrektor. Samuti on seda formalismi kasutades loodud grammatikakorrektoreid teistele keeltele, sealhulgas Põhjamaades rootsi, taani ja norra keelele. Lisaks on kitsenduste grammatika tööpõhimõte võrdlemisi kergesti mõistetav ning selle reegleid koostada ning testida on suhteliselt lihtne.

Töö koosneb kolmest peatükist. Esimene peatükk annab ülevaate olemasolevatest lähenemistest grammatikavigade tuvastamiseks teiste keelte jaoks, keskendudes kirsenduste grammatika formalismi kasutavatele süsteemidele, ja varasematest katsetustest eesti keele grammatikakorrektori loomiseks. Teises peatükis tutvustatakse lähemalt kitsenduste grammatikat kui veatuvastusgrammatika formalismi, valitud vigade märgendamisviisi ning annan süstemaatilise ülevaate koostatud reeglitest. Viimases peatükis kirjeldatakse kasutatud korpusi ja nende omapärasid, analüüsitakse koostatud grammatikakorrektori pool tehtud vigu ning antakse hinnang tulemustele.

Tööle on lisatud komavigade tuvastaja kasutusjuhend, korpuse eeltöötlusel kasutatud skript, grammatikakorrektori reeglifail, näited sisendist ja väljundist ning töös kasutatud korpuse käsitsi märgendatud variandid.

1. Ülevaade grammatikakorrektoritest

1.1. Grammatikakorrektorites kasutatavatest formalismidest

Nagu muudeski keeletehnoloogia valdkondades, on ka grammatikakorrektorite loomisel kasutatud nii reeglipõhiseid kui statistilisi lähenemisi, samuti mõlema kombinatsioone. Valdavalt on kasutusel reeglipõhised meetodid, küllaltki levinud on ka masinõppemeetoditel koostatud grammatikakorrektorid.

Samuti erinevad korrektorid selle poolest, missugust lingvistilist infot nad sisendilt eeldavad, kas vastava info lisamine on grammatikakorrektoriga integreeritud või tuleb seda eelnevalt teha. Strateegiaid võib jagada ka korrektori ning süntaksianalüsaatori suhte järgi: kas alustada vea otsimist siis, kui süntaksianalüsaator lause vigaseks tunnistab ning analüüsida ei suuda, või püüda võimalikke vigu tuvastada juba enne teksti süntaktilist analüüsi või selle käigus [Kuboň jt., 1997, lk 4].

Osadel juhtudel märgib korrektor vaid, et sisendlause on vigane, osadel soovitatakse kasutajale sobivat parandust või korrigeeritakse tekst automaatselt, osadel antakse võimalike paranduste paremusjärjestus. Enamasti on grammatikakorrektori puhul tõesti tegu korrektoriga, s.t. iga leitud vea korral pakutakse välja ka korrigeeritud lause, millele enne tekstis asenduse tegemist küsitakse kasutajapoolset kinnitust.

Lisaks on grammatikakorrektorid välja töötatud erinevate tekstitüüpide abil. Sagedaseimaks on ajalehetekstid, kuid on kasutatud ka ilukirjandustekste ning võrdlemisi suure vigade osakaaluga laste, võõrkeeleõppijate või düsgraafikute kirjutatud tekste.

Kontrollitavate veatüüpide lõikes tuvastatakse ortograafia-, süntaksi- ning stiilivigu. Sagedasim sihtgrupp on ühildumisvead, parandatakse ka punktuatsioonivigu, kokku-lahkukirjutamist, puuduvaid või üleliigseid sõnu jms. Mõned korrektorid püüavad katta kõiki vealiike, teised keskenduvad mingit kindlat tüüpi vigadele.

Võib öelda, et küllaltki sageli tegeldakse eraldi kas ühilduvus- või komavigade tuvastusega. Grammatikakorrektorit looma asudes valitakse välja üks neist vealiikidest ning jäetakse teise veatüübi kontroll hilisemaks, seda tihti arvestusega, et tolle sisendis on esimest liiki vead juba parandatud. Sealjuures lähevad lahku teadlaste arvamused,

kas komavead tuleks korrigeerida enne ühilduvusvigadega tegelema hakkamist või vastupidi. Bick toob välja, et komavigu tuleks tuvastada eelnevalt ühildumisvigade suhtes korrigeeritud, grammatilisemas kontekstis [Bick, 2006 lk 9]. Samas Aldezabal jt. argumenteerivad, et peale komavigade tuvastust oleks lihtsam osalausepiire määrata ning seega lihtsam muid süntaksivigu tuvastada ja parandada [Aldezabal jt., 2003, lk 1]. Kuid on ka lähenemisi, kus ühildumis- ja komavead on koondatud üheaegselt samasse grammatikakorrektorisse, näiteks baski [Ansa jt., 2004] või taani keele [Hardt, 2001] puhul.

Eelpooltoodu näitlikustamiseks on järgnevalt toodud mõningate lähenemiste lühiiseloomustused, pöörates tähelepanu mainitud aspektidele. Välja on toodud neli reeglipõhist korrektorit, üks reeglipõhist ja statistilist lähenemist kombineeriv korrektor ning lõpetuseks kaks masinõppemeetoditel loodud grammatikakorrektorit.

SCARRIE projekti raames norra keelele loodud reeglipõhine grammatikakorrektor [De Smedt, Rosén, 1999] tuvastab ühilduvus- ning stiilivigu. Veakaalud on integreeritud süntaksianalüsaatori reeglitesse, nii leitakse analüüsi käigus võimalikud vead ning suudetakse neile parandused välja pakkuda. Stiilivead parandatakse eelnevalt leksikoni põhjal eraldi, kuna sobimatu sõna asendus võib kaasa tuua ühildumisvigu.

Mõnevõrra sarnaselt toimib tšehhi keele grammatikakorrektor, kus samuti toimub grammatikavigade tuvastus ning korrigeerimine samaaegselt süntaksianalüsaatori tööga. Erinevalt Scarriest on tšehhi keele puhul tegu sõltuvusgrammatikaga. Grammatikakorrektor ning süntaksianalüsaator käivitatakse ühe protsessina, korrektori reegleid rakendatakse vaid lausetele, millele süntaksianalüsaator korrektseid projektiivseid süntaksipuid koostada ei suuda. Ebakorrektsetest või mitteprojektiivsetest (alam)puudest tuvastatakse ühilduvusvigu [Kuboň, 1997].

Pisut erinev on baski keelele loodud puude transformatsioonil põhinev grammatikakorrektor [Díaz de Ilarraza jt., 2005], millega tuvastatakse küll samuti ühildumisvigu, ent seda peale süntaksianalüsaatori töö lõpetamist. Esimese sammuna analüüsitakse lauset morfoloogiliselt ning süntaktiliselt baski keele sõltuvusgrammatika analüsaatori abil. Seejärel rakendatakse saadud sõltuvuspuu(de)le grammatikareegleid. Viimased on kirjutatud spetsifikatsioonikeeles, mis võimaldab sõltuvusgrammatika puudest otsida kirjeldatud tüüpi lauseid ning neid etteantud viisil transformeerida. Lisaks vea tuvastusele ning parandamisele märgib reegel ära vigased puuosad ning

annab loomulikus keeles veakirjelduse.

Baski keele jaoks on loodud ka teine reeglipõhine grammatikakorrektori versioon [Ansa jt., 2004, lk 7-8]. Enne grammatikakorrektori reeglite kirjutama asumist koostati ühes lingvistide ning erialaekspertidega lingvistiliste vigade korpus ERREUS. Plaanis on tuvastada ortograafiavigu, morfosüntaktilisi, semantilisi ning kirjavahemärgivigu. Välja on toodud võimalikud veatüübid ning jaotatud need rühmadesse vastavalt sellele, missuguse lingvistilise info põhjal on võimalik neid tuvastada: ilma lisainfota (nt. mõningad kirjavahemärgivead), pindsüntaktilisel analüüsil tuvastatavad (nt. järelsõnavead) ning süvasüntaktilise analüüsi põhjal tuvastatavad (nt. ühildumisvead). Esialgu on realiseeritud vaid MS Wordi API regulaaravaldistel põhinevaid esimese rühma vigu tuvastavaid reegleid.

Rootsi keele jaoks loodud Granska grammatikakorrektor [Carlberger et al., 2004] ühendab endas statistilisi ning reeglipõhiseid meetodeid. Ühte korrektorisse on integreeritud ortograafia-, ühildumis- ning kokku-lahkukirjutamise vigade analüüs, samuti viimase kahe vealiigi tuvastamisele eelnev morfoloogiline märgendamine. Korrektor ise moodustab osa keeleõppesüsteemist. Reeglid on formuleeritud spetsiaalses keeles, grammatikareegli põhikujus on määratud muster, millele reeglit rakendada, parandus ning kasutajale edastatav veakirjeldus. Lisaks kasutatakse ka abireegleid (kasutusel näiteks fraasipiiride tuvastuseks – nii saab määrata, et mõnd reeglit rakendada üksnes teatud tüüpi osalausel) ja erandeid välja sorteerivaid aktsepteerimisreegleid. Abireeglite kasutus muutvat antud keele võimsamaks tavapärastest regulaaravaldistel põhinevatest grammatikakorrektoritest. Kohad tekstis, millele reegleid rakendada, määratakse statistiliselt. Ajalehetekstidel saavutati täpsus 53% ja saagis 52%[1].

Hardt kasutas Brilli märgendajat taani keele grammatikakorrektori prototüübi loomiseks [Hardt, 2001]. Meetodi sobivuse katsetamiseks tuvastati ning korrigeeriti teatud tüüpi komavigu ning üht tüüpi ühildumisviga. Ühildumisvea puhul kasutati morfoloogiliselt märgendatud sisendteksti, komavigade puhul vähendati märgendeid üksnes sõnaliigile ja käändele. Tuvastatavaks ühildumisveaks oli üks artikli-nimisõna ühilduvuse juht, treenimiseks ning testimiseks vahetati omavahel kahe artikli esinemisi tekstis. Selle veatüübi puhul saavutati täpsus 95% ja saagis 85%. Komadest tuvastati selliseid

[1]Täpsuse all mõistetakse korrektselt määratud märgendite arvu ja kõigi määratud märgendite arvu suhet. Saagis on korrektselt määratud märgendite arv jagatud kõigi korrektsete märgendite arvuga.

juhtumeid, kus koma tekstis oli üleliigne. Treenimiseks ning testimiseks lisati teksti juhuslikke komasid. Tulemuseks saadi täpsus 91% ja saagis 77%.

Alegria jt püüdsid masinõppemeetodite abil luua baski keelele korrektorit, mis tuvastaks nii kohad, kust koma on puudu, kui ka need, kus koma ülearu [Alegria jt., 2006]. Igale korpuse sõnale lisati pindsüntaktiliselt analüsaatorilt saadud märgendid, info tema paigutuse kohta lauses ning märge, kas talle järgneb koma. Samuti märgiti ära, kas sõna kuulub 300 sagedasema pärast koma asetseva sõna või trigrammi hulka. Katsetati kolme masinõppemeetodit: Naiivse Bayesi meetodil, tugivektoritel ning otsustuspuudel põhinevaid klassifitseerijaid, parim tulemus saadi otsustuspuude meetodiga. Juhtudel, kus tekstis esines liigne koma, saavutati 96% täpsus ja 98% saagis, puuduoleva koma tuvastamisel 70% täpsus ja 49% saagis.

1.2. Kitsenduste grammatikal põhinevad grammatikakorrektorid

Kitsenduste grammatika põhjal on loodud grammatikakorrektorid rootsi Arppe, 2000, norra Hagen et al., 2001a, taani Bick, 2006 ja baski keelele Aldezabal et al., 2003. Kõigil nimetatud keeltel on olemas kitsenduste grammatika põhine süntaksianalüsaator. Järgnevalt antakse lühiülevaade nimetatud keelte korrektorite puhul kasutatud lähenemisviisist, korpustest, kitsenduste grammatika reeglite osast kogu süsteemis ning tuvastatavatest veatüüpidest.

Aastast 1997 arendab firma LingSoft rootsi keelele kitsenduste grammatikal põhinevat grammatikakorrektorit Grammatifix. Kuna samal firmal on rootsi keele jaoks juba välja arendatud speller ning kitsenduste grammatikal põhinev süntaksianalüsaator, siis loodi grammatikakorrektor spelleriga koos kasutamiseks mõeldud järgmise keeleabivahendi etapina. Korrektori väljatöötamine toimus Microsoft Wordi integreerituna. Reegleid testiti ajalehtedest, raamatutest pärit ning ka korrektori loojate poolt koostatud morfoloogiliselt ühestatud ning süntaktiliselt analüüsitud tekstide põhjal. Olenevalt tuvastatavast veatüübist kasutati kolme erinevat formalismi: süntaksivigade leidmiseks kitsenduste grammatika reegleid, kirjavahemärkide ja numbrite õigekirja kontrolliks ning lauselõpu kirjavahemärgi määramiseks regulaaravaldisi, stiilivigade tuvastamiseks lisati leksikonis sõnadele stiilimärgendid. Komavigu kui selliseid ei tuvastatud. Lõpp-produkti jäeti alles vaid need reeglid, mille täpsus oli vähemalt 67%. (Korrektori keskmine täpsus on 70% [Hagen jt., 2001a].) Süntaksivigade tuvastamiseks koostati üle

650 reegli, millega tuvastati 26 veatüüpi. Iga reegliga oli seotud ka vastav parandussoovitus, mis kasutajale edastati. Kuna süntaksianalüsaator ei suutnud grammatiliselt ebakorrektsetes lausetes fraasipiire korrektselt määrata, siis kirjeldati reeglites täpsemalt, mida loetakse fraasideks, mille raames siis näiteks ühilduvust kontrollida. [Arppe, 2000]

Oslo ülikooli teadlased koostöös firmaga LingSoft töötasid 2001. aastaks välja kitsenduste grammatikal põhineva grammatikakorrektori NGC norra kirjakeelele [Hagen jt. , 2001a]. Reeglid koostati ning täpsustati enam kui 4 miljonil sõnal, testiti 890000-sõnalisel ajalehekorpusel [Hagen jt. , 2001b]. Kõik reeglid on kirjeldatud kitsenduste grammatikas, morfoloogiliselt analüüsitud ning ühestatud tekst antakse ette grammatikakorrektori reeglitefailile. Kuna saadud süsteem moodustas terviku, siis kirjutati grammatikakorrektori loomise käigus ümber seal kasutatav morfoloogiline ühestaja. Nimelt eemaldati osad ühestamisreeglid, lubades alles jätta osad grammatiliselt korrektses lauses sobimatud märgendid ning lahendades nõnda suure osa morfoloogilisest ühestajast tulenevatest probleemidest. Samuti kaaluti spelleri muutmist, lisades leksikoni sagedasemad valesti kirjutatud sõnad, märkega 'ebakorrektne' ('Incorr'). Seega oleks grammatikakorrektor suuteline õigesti leidma vigu ka seal, kus sõna on kirjutatud valesti, kuid väär vorm kattub mõne muu sõna korrektse vormiga. Tuvastati ühilduvuse, sõnajärje ja mõningaid sõnavaliku probleeme, samuti finiitsete verbide arvu lauses ning ka üht komaviga: norra keeles peab kahe omadussõna vahel olema koma või sidesõna. Kokku on kasutusel üle 700 reegli, korrektori täpsus on 75% [Hagen jt., 2001a; Hagen jt., 2001b]. Sama grammatikakorrektorit testiti võõrkeeleõppijate ning kurtide laste kirjutatud esseedel, süntaktiliste ja morfoloogiliste vigade suhtes saavutati täpsus 95%, ent saagis jäi alla 40% [Hagen jt., 2002].

Eckhard Bick Lõuna-Taani ülikoolist on taani keelele välja arendanud kitsenduste grammatikal põhineva grammatikakorrektori OrdRet. Tegu on eraldiseisva programmiga, mis koosneb spellerist, morfoloogilisest analüsaatorist ja korrigeerijast, süntaksipõhisest ühestajast ning veatuvastajast-korrektorist. Sõnatasemel vigade tuvastusel kasutatakse nii reeglipõhist kui statistilist parandamist. Eraldi tähelepanu pöörati asjaolule, et grammatiliselt korrektsetel tekstidel üles ehitatud morfoloogiline ühestaja eemaldab grammatiliselt ebakorrektse teksti puhul töö käigus sageli tegelikult korrektse märgendi, jättes korrektorile analüüsiks vaid ebakorrektse märgendiga sõna. Lahendamaks seda probleemi, rakendati osa reegleid (kõige täpsemad) ka enne

morfoloogilist ühestamist – nii võeti arvesse ka neid märgendeid, mis muidu oleks ühestamise käigus eemaldatud. Korpuseks olid suure vigade osakaaluga düsgraafikute kirjutatud tekstid, kokku üle 32000 sõna. Tuvastati ühilduvusvigu, vale sõnakasutust jms. Muuhulgas soovitab korrektor ka punkte, kuna düsgraafikud jätavad sageli lause lõppu punkti panemata ning kirjutavad järgmise lause algustähe väiksena. Samas komavigu korrektor veel ei tuvasta, seda soovitatakse realiseerida teise etapina, juba muus osas korrigeeritud tekstil. Veatuvastusreeglid lisavad vigasele kohale veatüüpi näitava märgendi, kasutajale esitatakse pakutavate korrektuuride pingerida. OrdRet tuvastas 68% tekstis olevatest vigadest, täpsus oli samuti kõrge, 92% [Bick, 2006].

Aldezabal jt. tegelevad grammatikakorrektori loomisega baski keele komavigade tuvastamiseks. Kuna baski keelel ei olnud aastaks 2003 veel komareeglid ametlikult normeeritud, siis oli grammatikakorrektori loomisel esimeseks sammuks koostöös lingvistidega täpsustada juhud, mil koma peaks lauses esinema. Esialgu on loodud mõningad kitsenduste grammatika reeglid, mis tuvastavad puuduvad komad sidesõnade ümbruses, lisades vastavale sõnale märgendi, mis osutab, kuhu tuleks koma lisada. Lisaks kitsenduste grammatika reeglitele on Microsoft Wordi APIs kirjutatud mõned lihtsamad regulaaravaldistel põhinevad grammatikareeglid lause alguse suurtähtede ning kirjavahemärkide õigekirja kontrolliks. Edaspidi plaaniti ka valminud kitsenduste grammatika veatuvastusreeglid MS Wordi integreerida. [Aldezabal jt., 2003]

1.3. Eesti keele grammatikakorrektorid

Eesti keelele on varem kahel korral proovitud grammatikakorrektorit luua Lauri Lundi [Lunt, 2003] ja Kristiina Kure [Kurg, 2005] bakalaureusetööde raames. Mõlemal juhul kirjutati programmeerimiskeeles Java mõningate grammatikareeglite kontroll, kus sisendtekst kontrolliti iga veatüübi kohta eraldi lause-lauselt läbi. Kure töös on mainitud, et väljundis lisati ebakorrektseks määratud lausetele veatüüpi kirjeldav märkus; Lunt pole seda oma töö tekstis küll selgesõnaliselt välja toonud, ent programmi koodi uurimisel selgus, et ka tema annab väljundis kasutajale infot veatüübi kohta. Reegleid testiti bakalaureusetööde autorite endi poolt koostatud lausetel, mis olid eelnevalt automaatselt süntaktiliselt märgendatud. [Lunt, 2003, lk 15-21, 31; Kurg, 2005, lk 19-22, 28]

Lunt tuvastas oma töös loodud korrektori abil nii komavigu kui ka ühilduvusvigu: aluse

ja öeldise, omadussõnalise täiendi ja fraasipõhja vahelist ühilduvust ning verbide rektsiooni. Komavigadest kontrolliti koma olemasolu sidesõnade ees ja järel. Koma mittenõudvate sidesõnade (ja, ning, ega, ehk, või) puhul keelati koma nii sõna ees kui järel. Koma nõudvate sidesõnade (aga, vaid, kuid, ent) puhul nõuti koma olemasolu sidesõna ees, välja arvatud juhul, kui ühes kahest sidesõnaga eraldatud osalausest puudus öeldis. Töö käigus prooviti luua reegleid ka järeldus- ning seletusseoste ees olevate komade kontrolliks, kuid need reeglid jäid töö lõppversioonist välja. Saagis oli töö lõpptulemusel küllaltki suur, ent see-eest anti palju valepositiivseid tulemusi. Kuna Lundi korrektor tuvastas lauseliikmed süntaksianalüsaatori märgendite põhjal, siis andis grammatikakorrektor sageli väära tulemuse just valesti määratud süntaktiliste või morfoloogiliste märgendite tõttu. Et nii töös kasutatud morfoloogiline ühestaja kui ka süntaksianalüsaator on üles ehitatud grammatiliselt korrektsete lausete põhjal, siis andsid mõlemad grammatiliselt ebakorrektsele sisendile tihti ka ebakorrektse analüüsi. [Lunt, 2003, lk 4-10, 21, 24-33]

Kure koostatud grammatikakorrektor tuvastas sarnaselt Lundi tööga ühilduvus- ning rektsioonivigu, kuid komavead jäeti vaatluse alt välja. Töö protsess oli võrdlemisi sarnane Lundile, reeglid arvestasid samuti morfoloogiliste ning süntaktiliste märgenditega. 24 testlausel saadud tulemused jäid Lundi tööle lähedaseks. Ka Kurg tõi välja, et vähene täpsus tuleneb suuresti sellest, et süntaksianalüsaator on loodud analüüsima grammatiliselt korrektseid lauseid. [Kurg, 2005, lk 20-28]

Käesolev töö erineb eelnevatest eesti keelele loodud grammatikakorrektoritest nii kasutatava formalismi kui testimiseks võetud korpuse poolest. Sarnaselt Lundi ja Kure tööle kasutatakse sisendiks eesti keele süntaksianalüsaatori poolt märgendatud teksti, kuid korrektor on realiseeritud mitte Java programmi, vaid kitsenduste grammatika reeglitefailina ning reeglite koostamisel ja testimisel kasutatakse internetist kogutud mõnevõrra suuremat korpust.

2. Reeglid

2.1. Kitsenduste grammatika

Kitsenduste grammatika formalismi on välja töötanud Fred Karlsson Helsingi Ülikoolist, mõeldes sealjuures eelkõige süntaktilisele analüüsile. Hiljem on seda formalismi kasutatud ka paljudes muudes valdkondades. Kitsenduste grammatika süntaksianalüüsi põhieesmärk seisneb piiramata morfoloogiliselt analüüsitud teksti pindsüntaktilises analüüsis. Analüsaator teostab kolme baastüüpi operatsioone: ühestamise kitsenduste abil toimuv kontekstitundlik ühestamine, osalause piiride määramine vastavate märgendite lisamise abil ning morfosüntaktiliste märgendite lisamise ja süntaktiliste kitsenduste abil toimuv pindsüntaktiliste funktsioonide märgendamine [Karlsson jt., 1995, lk 42]. Kitsenduste grammatika üldpõhimõte seisneb seega konteksti arvestades sõnadele analüüside sõnadele võimalike analüüside analüüside määramises ja seejärel nende järk-järgulises eemaldamises (kitsenduste rakendamises), kuni alles jäävad vaid korrektsed analüüsid. Reeglid määravad oma rakendumise konteksti mustrite abil.

Antud töös kasutasin Lõuna-Taani Ülikoolis Visual Interactive Syntax Learning (VISL) projekti [World of VISL] raames realiseeritud kitsenduste grammatika analüsaatorit VISL CG-3 [VISL CG-3 Development Information].

Antud süsteem eeldab teataval kujul sisendit, mistõttu eesti keele süntaksianalüsaatorilt saadud morfoloogiliselt ja süntaktiliselt märgendatud teksti tuli enne vislcg3-süsteemile etteandmist töödelda. Seejuures, kuna antud süsteem on arendamisel ja võrdlemisi mittetäieliku dokumentatsiooniga, tuli aktsepteeritav sisendi kuju leida katsete käigus. Eesti keele süntaksianalüsaatorilt saadud väljundi teisendamiseks koostasin skripti (saadaval lisas 2). Näiteks on lause „Ilusad tüdrukud lähevad mööda“ süntaksianalüsaatori väljundis kujul:

```
"<$LA$>"
   #### ???
"<Ilusad>"
   ilus+d // _A_ pos pl nom  #cap // **CLB  @AN>
"<tüdrukud>"
   tüdruk+d // _S_ com pl nom  //  @SUBJ
"<lähevad>"
   mine+vad // _V_ main indic pres ps3 pl ps af  #FinV #Intr #ill #all //  @+FMV
"<mööda>"
```

```
   mööda+0 // _A_ pos //  @<AN
   mööda+0 // _D_ //  @ADVL
"<$.>"
   . // _Z_ Fst //
"<$LL$>"
   #### ???
```

Enne grammatikakorrektori reeglite failile sisendiks andmist viisin eelpool toodud näide skriptiga järgmisele kujule:

```
"<$LA$>"
   "####" ???
"<Ilusad>"
   "ilus+d" // _A_ pos pl nom  #cap // **CLB  AN>
"<tüdrukud>"
   "tüdruk+d" // _S_ com pl nom  //  SUBJ
"<lähevad>"
   "mine+vad" // _V_ main indic pres ps3 pl ps af  #FinV #Intr #ill #all //  +FMV
"<mööda>"
   "mööda+0" // _A_ pos //  <AN
   "mööda+0" // _D_ //  ADVL
"<$.>"
   "." // _Z_ Fst //
"<$LL$>"
   "####" ???
```

Nagu näha, on igale sõnale antud eraldi ridadel üks või mitu analüüsi, lause algusesse ja lõppu on lisatud vastavad märgendid, millele samuti on antud analüüsid. Reeglifail arvestab sõna koos kõigi tema analüüsiridadega: uusi märgendeid lisatakse ja eemaldatakse analüüsiridadele, ent võimalik on ka eemaldada terveid analüüsiridu, jättes alles vaid valitud märgenditega analüüsid. Kunagi ei eemaldata viimast allesjäänud märgendit või rida.

Koostatud reeglifailis leidub 3 kujul reegleid ning reeglitele eelnevad definitsioonid. Defineeritud on lauseeraldajad (Delimiters), antud juhul "<LA>" ja "<LL>", ning hulgad (List), tänu millele saab hulgas sisalduvatele märgenditele reeglites lühemalt hulganimega viidata. Reeglitüüpidest olen kasutanud märgendite lisamiseks Map käsku, Remove eemaldab vastava märgendi, Select jätab selle ainsana alles. Võimalik oleks olnud kasutada ka muid tüüpi käske, kuid selleks polnud vajadust.

Reegli põhikujuks on *'Käsk Märgend Sihtgrupp (Kontekstitingimused);*'. Järgnevalt on toodud mõningad definitsioonide ja reeglite näited.

- Lause alguse ja lõpu märgendid – arvestada neid lausete eraldajatena.

  DELIMITERS = "<LA>" "<LL>";

- Hulk 'Ja' koosneb koma mittenõudvatest sidesõnadest *ja*, ning, ehk, ega ja *või*, kusjuures *ja* ning *või* puhul arvestatakse vaid sidesõnana esinemisi. Viimane on

vajalik, kuna näiteks võib sõna *või* esineda ka nimisõnana.

List Ja = ("ja+0" // _J_) ("ning+0") ("ehk+0") ("ega+0") ("või+0" // _J_); #Koma mittenõudvad sidesõnad

- Lisa kõigile hulka Ja kuuluvatele sõnadele (täpsemalt vastavatele sõna analüüsidele) märgendid @ERR ja @OK.

  MAP (@ERR @OK) Ja;

- Kui sõna kuulub hulka Ja, siis jäta talle alles märgend @OK, kustuta ülejäänud märgendid (antud juhul @ERR). Juhul, kui sõnal on mitu analüüsi, millest osadel on üheselt märgend @OK, teistel vastav märgend puudub, siis kustutab ülejäänud analüüsiread.

  SELECT (@OK) (0 Ja);

- Eemalda märgend @OK nendel sõna *kui* esinemistel, millele otseselt eelneb sõna *siis*.

  REMOVE (@OK) (0 („kui+0“)) (-1 („siis+0“)); #õige on 'siis, kui'

Viimases reeglis sõna *siis* kohta käiv sulg on kontekstitingimus. Vastavaid tingimusi võib esineda ühes reeglis ka rohkem kui üks, sel juhul igaüks eraldi sulgudes. Järgnevalt on välja toodud variandid, milliseid kontekstitingimusi saab määrata. Kõik positsioonid on defineeritud võrreldes nullpositsioonis asuva sõnaga, reegli sihtgrupiga.

- 1 (A) – järgneval, s.t. lauses ühe koha võrra tagapool asuval sõnal on märgend A
- -2 (A) – sõna märgendiga A asub lauses kaks kohta eespool
- NOT 1 (A) – järgneval sõnal ei esine märgend A
- 1*(A) – sõna märgendiga A asub lauses vähemalt ühe koha võrra tagapool
- 1C (A) – järgnev sõna on üheselt analüüsitud märgendiga A
- 1* (A) BARRIER (Com) – järgneb sõna märgendiga A nii, et kahe sõna vahel ei leidu koma
- 1* (A) LINK 1 (Com) - järgneb sõna märgendiga A, millele järgneb vahetult koma

Grammatikakorrektori töö käigus käivitatakse kõigepealt märgendite lisamise käsud, seejärel rakendatakse kitsenduste reegleid järjest, s.t. iga reegel võib sõna märgendeid

muuta vaid siis, kui talle eelnevad reeglid antud sõna juba üheselt märgendanud pole. Seega on reeglid järjestatud täpsemalt ning kindlamalt üldisemale, erandid enne reegleid, nii et üldisemad reeglid rakenduvad neil juhtudel, mille eelmised reeglid analüüsimata on jätnud.

2.2. Kasutatud märgendus

Reeglifaili algul on defineeritud sõnade rühmad (List) hilisema viitamise lihtsustamiseks. Seega ei pea ühtmoodi käituvaid sõnu iga reegli juures eraldi välja tooma, vaid võib kõigile neile viidata alguses määratletud hulganimega. Olen loonud eraldi hulgad enamasti komata (hulk Ja), enamasti komaga (hulk Et) kasutatavatele sidesõnadele, samuti potentsiaalselt võrdlusena kasutatavatele sidesõnadele *kui* ja *nagu* (hulk Kui) jms.

Reeglifaili teine pool koosneb ühest sektsioonist, kus lisatakse sõnadele märgendid (MAP) ning kitsendusreeglite abil valitakse märgenditest sobiv välja (SELECT, REMOVE). Olemasolev sektsioon tegeleb komavigadega, hiljem võib lisada sektsioonid teiste veatüüpide tuvastamiseks. Kuna igas sektsioonis saab nii märgendeid lisada kui eemaldada, on mõttekas selguse huvides eri veatüüpide reegleid eraldi sektsioonides hoida.

Esimese sammuna lisan kaks märgendit, @ERR ja @OK, vastavalt vigase ning korrektse koha märgendamiseks, kõigile sidesõnadele ja verbi pöördelistele vormidele. Alternatiivina võiks lisada märgendid kõigile sõnadele sisendtekstis, ent see raskendaks otsustamist, millise sõna juures osutada puuduvale komale. Üks võimalus siinkohal oleks jätta märgend @ERR alati selle sõna juurde, mis asub puuduva koma ees (või teise võimalusena taga). Kuid sageli on raske automaatselt otsustada, kus täpselt peab koma asuma, seda eriti juhul, kui vastavat kohta ei märgi ei ükski sidesõna ega ka näiteks kaks järjestikust verbivormi. Teine alternatiiv oleks lisada märgendid otsustamisreeglitega ehk jätta ära märgendite lisamine sektsiooni algul ning muuta olemasolevaid reegleid nii, et mitte ei valitaks sobivat märgendit kahest olemasolevast, vaid lisatakski üksnes sobiv märgend. Nii on olukord lahendatud näiteks baski ja taani keele puhul, viimasel küll lisatakse mõnel juhul mitu märgendit, millest hiljem sobiv välja valitakse [Aldezabal et al., 2005, lk 3; Bick, 2006, lk 5-6]. Ent sel juhul oleks tunduvalt raskem tuvastada neid kohti, mida reeglid ei kata, teisisõnu kus mõni

sihtgrupi sõna (sidesõna, verbivorm) on jäänud märgendamata ehk siis mitmeseks. Seega on valitud lähenemise eelisteks hea jälgitavus ning kuigi märgend ei pruugi alati viidata just puuduva koma konkreetset asukohta, osutab ta põhjusele, konkreetsele sõnale, mis komavajaduse tekitab.

Siinkohal võiks anda erinevad märgendid sidesõnadele, mille puhul @ERR märgendiga sõna osutab, et vahetult (ühend)sidendi ees peaks asuma koma, ning verbivormidele, mille puhul @ERR märgend näitab vaid, et koma peaks asuma lauses kusagil eespool, kuna kaks vastavakujulist verbivormi tuleks üksteisest eraldada kas koma või koma mittenõudva sidesõnaga. Niisiis oleks paranduse soovitamise mõttes kasulikum, kui neid kahte eri juhtu ka eri viisil märgendataks. Teisest küljest osutab ka verbivormi @ERR märgend just probleemsele kohale – sinnamaani ei ole midagi, mis oleks nõudnud koma olemasolu, seega kuni tolle verbivormini on lause põhimõtteliselt selles suhtes korrektne. Hetkel kasutan mõlema juhu jaoks samu märgendeid, kuid nende muutmine oleks soovi korral võrdlemisi lihtne.

Õige märgendi valik toimub kas Select- või Remove-tüüpi reeglite abil. Mõlemal puhul tekkis probleeme mitmeste analüüsidega sõnadega, millel märgendid lisati vaid ühe sõnaliigi analüüsidele. Seda alles siis, kui mõni eelnev reegel on kahest märgendist, @ERR ja @OK, juba õige välja valinud. Valdavalt olen selliste sõnade puhul kasutanud Select-tüüpi reegleid, mis eemaldavad kõik ülejäänud (märgenditeta) analüüsid. See tähendab, et näiteks sõnal, millel morfoloogilisel ühestamisel olid jäetud alles nii sidesõna kui ka määrsõna märgendid, kaotatakse tema analüüs määrsõnana. Seevastu Remove-tüüpi reeglite puhul võib juhtuda, et kui üks reegel on eemaldanud sõna analüüsilt @OK märgendi, osutades seega vigasele kohale, võib mõni hilisem ja üldisem reegel, mis eemaldab @ERR märgendid, vastava analüüsirea tervikuna kustutada. Sel juhul jääks tõenäoline grammatikaviga tuvastamata, seega olen püüdnud vältida Remove-tüüpi reeglite kasutust mitmeste analüüsidega sõnade korral.

2.3. Reeglite koostamise alused

Grammatikakorrektori reeglite koostamisel lähtusin eelkõige grammatikakäsiraamatus [Erelt, 2006] toodud reeglitest, valides välja toodud komakasutuse reeglitest need, mille automaatne kontroll teostatav tundus. Olles vastavad reeglid kirja pannud, katsetasin nende täpsust korpusel. Kui leidsin korpusepõhisel vaatlusel juhtumeid, mida

käsiraamatu põhjal koostatud reeglid ei katnud, lisasin tekstides esinenud lausete põhjal täpsemaid reegleid.

Käesoleva töö eesmärgiks oli koostada grammatikakorrektor, mis tegeleb üksnes komavigade tuvastamisega. Seega jätsin korrektorist välja kõik muude kirjavahemärkide kohta käivad reeglid. Samuti jäid antud rakendusest kõrvale sellised grammatikareeglid, mille rakendamisel tuleks lähtuda semantilisest või pragmaatilisest teadmisest, mida antud korrektor ei võimalda. Nii näiteks ei jälgi korrektor komakasutust samaliigiliste ja eriliigiliste määruste korral, kus esimesel juhul tuleks kahe määruse vahele koma panna, teisel juhul aga mitte.

Grammatikakorrektoris keskendusin reeglitele, mis määravad komakasutuse üheselt ära. Võimalik on lisada ka stilistilisi juhtnööre, soovituslikku komakasutust väljendavad reeglid, mida siis tuleks märgendada ka erinevalt otsestest grammatikavigadest.

Mitte kõik grammatikareeglid, mida oleks võimalik antud süsteemis väljendada, ei ole sisse võetud. Näiteks on veel realiseerimata komakasutus kiilu korral. Samuti leidub korrektoris reegleid, mis ei ole sajaprotsendiselt õiged. Samas saab viimast tüüpi reegleid täpsustada, lisades olemasoleva reegli ette reegleid nendele juhtudele, mil olemasolev reegel ei kehti. Kuna reegleid rakendatakse järjest, siis eelpoolasetsevad täpsemad reeglid analüüsivad ära laused, milles hilisem reegel eksiks, ning hilisemat, üldisemat reeglit ei rakendata, seega pole teda vaja ka muuta.

Et nii morfoloogiline ühestaja kui süntaksianalüsaator on loodud toimima korrektsetel lausetel, siis grammatiliselt ebakorrektse sisendi puhul võivad mõlemad sõnadele anda ka tegelikkusele mittevastavaid märgendeid. Seega keskendusin sidesõnade kohta käivatele reeglitele, kuna neid on valdavalt kergem tuvastada ning nende vormidel on suhteliselt ühesed analüüsid. Niisiis, isegi kui puuduva komaga lauses on verbid valesti analüüsitud ning nende põhjal viga tuvastada ei saa, on võimalik komaviga leida sidesõnade abil. Kuigi süntaksianalüsaator märgendab ka osalausete piire, mis on komakohtade määramisel sageli otsustava tähtsusega, määrab süntaksianalüsaator need piirid just kirjavahemärkide, sidesõnade ja verbide alusel [Roosmaa jt., 2001, lk 48], seega pole need grammatiliselt ebakorrektses lauses suure tõenäosusega kuigi usaldusväärsed. Seega olen keskendunud pigem morfoloogilistele kui süntaktilistele märgenditele.

Reeglite koostamisel jagasin märgendatavad sõnaliigid – pöördelised verbivormid,

küsisõnad ja sidesõnad – eri hulkadesse, millele eraldi reegleid koostasin. Sidesõnad jagasin edasi nelja rühma, arvestades eraldi ka võrdlussõnadena kasutatavaid sõnu *kui* ja *nagu*, sageli määrsõnadena esinevaid sõnu *siis* ja *mitte*, valdavalt koma mittenõudvaid sidesõnu (*ja, ning, ega, ehk, või*) ja ülejäänud, koma nõudvaid sidesõnu. Täpsemate reeglite kirjutamisel, näiteks ühendite *olgugi et* vms korral, arvestasin vajadusel ka konkreetse sõnaga.

Kontekstitingimustes kasutasin sageli kirjavahemärke, koma mittenõudvaid sidesõnu ja lause lõppu tõenäolise osalause piiri märkimiseks ehk siis näiteks määramaks, kust kaugemal ei tuleks enam öeldist otsida. Võimaliku öeldisena kontekstitingimustes võtsin arvesse pöördelisi verbivorme, da-infinitiive, mis teatud tingimustel võivad esineda öeldisena, ja nud-, tud-kesksõnu, mille puhul võib öeldise teine pool, tegusõna 'on', olla osalausest välja jäetud.

Järgmises alampeatükis on ära toodud töö käigus koostatud 90 reegli kirjeldused nimetatud hulkade kaupa. Iga reeglite rühma järel on märgitud, mis ridadel reeglifailis (lisas 3) vastava selgituse kohta käivad reeglid asuvad.

2.4. Reeglite kirjeldus

2.4.1. Üldised reeglid

- Kui märgendatav sõna paiknes lauses esimesel positsioonil, määrasin ta korrektseks. [rida 45]
- Kui kaks sidesõna asuvad järjest, siis nende vahele koma ei panda. Nt *Tuleks tõrelda, kuid et ta on nii tundlik, siis piisab ehk märkusest* [Erelt, 2006, lk 142]. [read 65, 67] Juhul, kui tegu oleks loeteluga (Nt *Sidesõnade ja, ning, ega, ehk, või ette enamasti koma ei panda*.), siis tuleks nende vahele koma panna. Selline juht on küllaltki harv ja et see ka kasutatud korpuses ei esinenud, siis pole seda reeglites arvestatud.
- Kui küsisõnale eelneb sidesõna, siis nende vahele koma ei panda. Nt *Ja mis see sinu asi on*? [read 65, 67]

2.4.2. Sõnade *siis* ja *mitte* reeglid

- Kui sõnale *siis* järgneb vahetult sõna *kui*, siis nõuda nende vahele koma. *Nt Tulen siis, kui tahan.* [rida 50]
- Kui sõnale *siis* eelneb lauses sõna *kui* ning järgneb võimalik öeldis, siis juhul, kui sõnade kui ja siis vahel leidub võimalik öeldis, nõuda *siis* ette koma. Nt *Kui teha, siis teha hästi.* Vastasel juhul koma mitte panna. Nt *Ja kui siis hakkas sadama*. [read 51, 52]
- Kui järjest esinevad sõnad *kui*, *siis* ja küsisõna, siis nõuda *siis* ette koma. Nt *Ja kui, siis mida?*[rida 53]
- Kui sõnale *siis* või sõnale *mitte* eelneb vahetult sõna *ka*, siis lugeda määrsõnana esinemiseks ning tema ette koma mitte nõuda. Nt *Liikluses ei maksa unistama jääda ka siis mitte, kui oled juba harjunud korralikult liiklema.*[rida 72] Antud reegel pole ilmselt küll grammatiliselt kõige korrektsem, kuid kasutatud korpuses vigu ei põhjustanud.
- Kui sõnale *siis* või sõnale *mitte* eelneb mõni sõnadest *kui*, *kuna*, *et* nii, et nende kahe sõna vahel ei leidu koma, kuid leidub võimalik öeldis, siis nõuda sõna *siis* (või vastavalt *mitte*) ette koma. Nt *Kui teha, siis teha hästi.* [rida 73]
- Kui sõna *siis* või sõna *mitte* asub lause lõpus või talle järgneb kirjavahemärk (punkt, küsimärk, sulud..), määrata ta õigeks. Nt *Või siis mitte. Kes see võttis siis? Kui jääd ette uimama, siis ...* [rida 74]
- Kui sõnale *siis* või sõnale *mitte* ei järgne samas oletatavas osalauses võimalikku öeldist, siis määrata ta õigeks. Nt *Tuli siis värsket verd ja ükspäev oli päevnikuks üks Narva poiss.* [rida 75]
- Kui sõna *siis* või sõna *mitte* asub lauses vahetult aluse ja verbivormi vahel, siis määrata ta õigeks. Nt *Nii me siis magasime hommikuni.* [rida 76]
- Kui sõnale *siis* või sõnale *mitte* eelneb modaal- või abiverb ja nende kahe sõna vahel ei asu võimalikku öeldist, siis määrata ta õigeks. Nt *Aga kus sa saad siis vahetust teha.* [rida 77]

2.4.3. Sõnade *kui* ja *nagu* reeglid

- Mitmest sõnast koosnevatel ühenditel *siis kui, enne kui, juhul kui, nii palju kui, samal ajal kui* võib koma olenevalt rõhuasetusest panna nii ühendi kui ka sidesõna *kui* ette. [read 90-93, 98-105]
- Mitmest sõnast koosneva ühendi *isegi kui* puhul pannakse koma ühendi ette. [read 115, 116]
- Ühendis *nii ... kui ka* kui ette koma ei panda. Sealjuures võib osa ühendi liikmeid (näiteks *ka*) ära jätta. [read 133, 135]
- Kui sõnale *kui* või sõnale *nagu* eelneb ning järgneb ilma osalause eraldajata (koma või koma mittenõudev sidesõna) pöördeline verbivorm, siis märkida ta vigaseks. Nt *Miks sõltub riskikofitsent sellest* ***kui(@ERR)*** *kaua ma mingit autot oman?* Kui sellisel juhul on vastava sidesõna ees koma, märkida ta õigeks. [read 136-139]
- Kui sõnale *kui* või sõnale *nagu* järgneb enne mõnda võimalikku öeldist side- või küsisõna, siis määrata ka komata kasutus õigeks. Nt *Sa oled mõttetu raiskaja nagu mu sõber, kes ei oska raha hinnata.* [read 140, 141]

2.4.4. Koma mittenõudvate sidesõnade reeglid

Koma mittenõudvate sidesõnade ette tuleb koma panna kahel juhul. Esiteks, kui talle eelneb teise taseme kõrvallause vms, mis niikuinii tuleks muust lausest komadega eraldada. Teiseks, kui selle sidesõnaga algav osalause on eelnevaga võrreldes väga uudse või vastandliku sisuga. Kuna mõlemat on olemasoleva info põhjal raske otsustada ja sellised juhtumid, kus koma mittenõudva sidesõna ette oleks tulnud koma panna, moodustasid korpuses olevatest koma mittenõudvate sidesõnade esinemistest vaid 3,6%, siis hetkel valminud reeglites määratakse kõik selliste sidesõnade esinemised korrektseks. [rida 150] Hiljem võib üldise reegli ette lisada täpsemaid reegleid määramaks, millal tuleb koma panna ja millal mitte.

2.4.5. Koma nõudvate sidesõnade reeglid

- Mitmest sõnast koosnevatel ühenditel *selleks et, sellepärast et, selle asemel et, eeldusel et* võib koma olenevalt rõhuasetusest panna nii ühendi kui ka sidesõna *kui* ette. [read 86-89, 94-97]

- Ühend *nii et* käitub sarnaselt eelmise reegli ühenditega, aga püsiühendites võib ka päris ilma komata olla Nt *Valetab nii et suu suitseb*. [read 107-110]
- Mitmest sõnast koosnevatel ühenditel *olgugi et, ainult et, vaevalt et, peaasi et, ilma et, mitte et* puhul pannakse koma ühendi ette. [read 113, 114]
- Kui sõnale *et* eelneb vahetult sõna *see*, siis panna *et* ette koma. Nt *See, et oled raha kätte saanud, ei tähenda veel, et sellega asi lõppenud on.* [read 118, 119]
- Muul juhul peab koma nõudva sidesõna ees peab olema koma, nt *Järgmine päev soovitati minust teha kapral, et ma nii eeskujulikult tõin auto tagasi*, v.a. juhul, kui lauses temast tagapool ei leidu võimalikku öeldist, nt *Mina aga mitte.* [read 124-127]

2.4.6. Küsisõnade reeglid

- Kui mitu küsisõna esineb lauses järjest ning neile ei järgne lauses võimalikku öeldist, siis eeldada, et tegu on loeteluga, ja nõuda nende vahele koma. Nt *Kes, kus, mis, millal?* [read 57, 58]
- Kui mitu küsisõna esineb lauses järjest ning neile järgneb võimalik öeldis, siis ei pea koma olema (tõenäoliselt on tegu sama verbi kohta käivate küsimustega). Nt *Kes mida väärtustab. Sõltub sellest, mis asjad kus kohas müügil on.* [rida 60]
- Kui sõna ’mitu’ on kasutusel eestäiendina, siis tema ette koma ei pea panema. Nt *Ei suuda mitut asja korraga teha. Olen müünud ka mitmeid teisi masinaid.* [read 158-162]
- Kui küsisõnale eelneb „ma ei tea“ ning järgneb nimisõna, siis lubada küsisõna ette koma mitte panna. Nt *Loete siin ma ei tea mida kokku.* Juhul, kui järgneb ka võimalik öeldis, ilma et vahepeal leiduks kirjavahemärke või sidesõna, siis nõuda küsisõna ette koma panekut. Nt *Ma ei tea, mis see on.* [read 165, 166]
- Kui küsisõna on relatiivlause algul, siis peab tema ees koma olema. Nt *Pilk, millega teda kostitan, muutub seepeale eriti ükskõikseks.* [read 168, 169]
- Kui küsisõnale eelneb vahetult komaga sidesõna, siis lugeda küsisõna esinemine korrektseks. Nt *Kui ma olen kindel et ostan endale automaatkastiga auto, siis miks mitte.* [rida 171]

2.4.7. Pöördeliste verbivormide reeglid

- Kui otse vasakul on '*ei*' ja verb ise ei ole eitav, siis panna nende vahele koma. Nt *Aga ei, pidin minema.* [rida 180]
- Kui otse vasakul asub infinitiiv, mis ei ole sihitise funktsioonis, siis nende kahe vahel peaks olema koma. Nt *Selleks et paika panna, pidin parema käe taha ja üles painutama.* Juhul, kui tegu on modaal- või abiverbiga, lubada komata kasutus. Nt *Ei ole mõtet teha lube, millega ainult automaadiga sõita tohib.* Juhul, kui infinitiivi puhul võib tegu olla objektiga, lubada komata kasutus. Nt *Ta on juba 10 korda enne sodiks sõidetud, kui mina seal möödasõitu teha jõuan.* [read 183-185]
- Alati eraldatakse komaga järeltäiendina esinevad v-, tav-, nud-, tud-lühendid. Nt *Kaup, osalt kastidesse pakitud, osalt pakkimata, oli jäetud riiulitele vedelema* [Erelt, 2006, lk 150]. [rida 195]
- Alati eraldatakse komaga määruslikud nud- ja tud-lühendid. Nt *Asi aetud, tuli postile tagasi.* [rida 197]
- Kunagi ei eraldata komaga da-, ma-, mas-, mast-, vat- ja nuna-lühendit. Nt *Ma tahaksin koju minna* [Erelt, 2006, lk 151]. [rida 199]
- Kui vasakul leidub veel teinegi pöördeline verbivorm, ilma et nende kahe vahel koma oleks, siis märkida teine esinemine vigaseks. Nt *Aga kui kusagil kaugemal on kodu olemas ja seal elamise eest maksma ei pea* ***on(@ERR)*** *ju mõtekas seal elada.* [read 202-206]

3. Tulemuste analüüs

3.1. Korpusest

Antud magistritöös kasutatud korpuseks on Heli Uibo poolt Delfi internetiportaali kommentaaride hulgast kogutud grammatiliselt ebakorrektsed laused (kättesaadavad veebis aadressil http://www.ut.ee/~heli_u/vigadekorpus/). Kasutatud 298 lauses on pea igaühes üks või mitu koma puudu. Kuna grammatilist korrektsust või ebakorrektsust tähistav märgend lisatakse konkreetsele sõnale (täpsemalt küsisõnadele, sidesõnadele ja verbi pöördelistele vormidele), mitte tervele lausele, siis on sama teksti korrektsed osalaused piisavaks korpuseks, mille põhjal vältida liigset valepositiivsete tulemuste arvu.

Käesolevate reeglite väljatöötamisel pole kasutatud tervet kogutud Delfi kommentaaride korpust, vaid üksnes osa sellest. Edaspidi suurema korpuse põhjal testides on tõenäoline reeglite katvust ning ka täpsust parandada, seega võiks kindlasti edaspidisesse arendustöösse haarata ka ülejäänud korpuse.

Grammatikakorrektori edasiarendamisel võib kasutada ka kättesaadavat õppijavigade korpust[2], kus on toodud eesti keele võõrkeelena õppijate esseedest kogutud vead ja ka vastavad parandusettepanekud. Enne selle korpuse kasutuselevõttu oleks siiski vajalik kõigepealt seal esinevad vead jagada veatüübi alusel ning seejärel korpus märgendada.

Seega on tekstid, mida kasutasin grammatikakorrektori reeglite koostamisel, sarnased pigem taani keele grammatikakorrektoris kasutatud düsgraafide [Bick, 2006] ja norra keele grammatikakorrektori testimisel kasutatud võõrkeeleõppijate tekstidega [Hagen jt., 2002] kui näiteks rootsi keele puhul kasutatud ajalehekorpusega [Arppe, 2000]. Kuigi ka eesti ajalehtedest võiks kindlasti leida grammatiliselt ebakorrektseid lauseid ning koostada grammatikaanalüüs seda tüüpi tekstide põhjal, on just keelereeglite õppijad see sihtgrupp, kellele grammatikakorrektorit rohkem vaja läheks. Samuti esinevad kõrgtasemel keelekasutajate tehtavad veatüübid tõenäoliselt ka käesolevas töös kasutatud, kõrge veasisaldusega tekstides.

Kuigi internetibrauserites on võimalik kasutada ka spellerit, pole kasutatud korpuse tekstide autorid tõenäoliselt seda kas üldse teinud, või siis pole nad õigekirjavigadele tähelepanu pööranud. Igal juhul esineb antud tekstides mitmeidki vigu, mille speller

[2]http://www.keeletehnoloogia.ee/projektid/veebipohine-keeleope

tuvastada suudaks, näiteks valesti kirjutatud või kokku kirjutatud sõnad. Kuna grammatikakorrektori kasutamise eelduseks on, et kasutaja võtab arvesse ka eelneva lingvistilise kontrolli tulemusi, siis ei ole püütud grammatikakorrektori reeglitesse lisada nende vigade tuvastamist, mida juba sõnavormide leksikonis leiduvuse kontrolliga tuvastada suudaks. Samas peaks grammatikakorrektor olema suuteline tuvastama probleeme, mis tekivad siis, kui valesti kirjutatud sõna ühtib mõne leksikonis olemasoleva sõnavormiga (näiteks 'kindlasti' asemel on kirjutatud 'kindalasti'), mistõttu konteksti kasutamata seda eelnevalt üles leida ning parandada ei suudeta. Ühest küljest tekib sarnane probleem sellest, kui mõnel sõnal määratakse vale sõnaliik või muul moel väär morfoloogiline analüüs – lausesse jääb sinna mittesobiva märgendiga sõna. Ent teisest küljest on selliseid vigu on arvatavasti kergem leida ühilduvuse kui komavigade kontrollil.

3.2. Eelnev analüüs, probleemid ja võimalikud lahendused

Sisendi eelnevaks töötluseks kasutasin programmi Estcgparser, millesse on koondatud Heikki-Jaan Kaalepi loodud morfoloogiline analüsaator ning kitsenduste grammatika formalismil põhinevad Tiina Puolakaineni loodud morfoloogiline ühestaja ning Kaili Müürisepa loodud süntaksianalüsaator. Põhimõtteliselt võiks programme Estcgparser ja VISL CG-3 kasutada ühe käsuna, saades tekstifailile grammatikakorrektori analüüs ilma vahetulemusteta, ent seda takistab asjaolu, et Estcgparser eeldab Windowsi keskkonda, VISL CG-3 töötab aga operatsioonisüsteemis Linux.

Morfoloogiline analüsaator Estmorf põhineb sõnavormide leksikonis olevate lekseemide kombinatsioonidega võrdlemisel, analüsaator märgendab eestikeelse teksti 97% täpsusega [Kaalep, 1999, lk 26]. Järgmise etapina toimub morfoloogiline ühestamine, mis eemaldab 87% täpsusega mittesobivad morfoloogilised analüüsid, jättes ligi 10-15% sõnadele mitmesed analüüsid [Roosmaa jt., 2001, lk 92]. Viimase sammuna leiab aset süntaktiline analüüs ning ühestamine, mille tulemuseks automaatselt morfoloogiliselt ühestatud tekstidel on 78% täpsus, mitmese süntaktilise analüüsiga jääb 17% sõnadest [Roosmaa jt., 2001, lk 104].

Grammatikakorrektori sisendiks on seega süntaksianalüsaatori poolt märgendatud tekst, kus igale sõnale on määratud üks või mitu morfoloogilist analüüsi, millel igal on ka süntaktilised märgendid. Et nii morfoloogiline ühestaja kui süntaksianalüsaator

eeldavad grammatiliselt korrektset sisendit, siis ebakorrektsete lausete analüüs ei ole sageli õige. Näiteks võib olla verbil võimalikest märgenditest ühestamise käigus alles jäetud üksnes nimisõna või omadussõna märgend, mistõttu grammatikakorrektoril pole teada, et tegu on hoopis verbiga.

Üheks võimalikuks lahenduseks antud situatsioonis oleks muuta morfoloogilise ühestaja ja süntaksianalüsaatori reegleid nii, et jäetaks alles rohkem, ka grammatiliselt õiges lauses mittesobivaid märgendeid. Kuna selline tegevus nõuaks lisaks juba integreeritud valmiskujul programmide muutmisele sügavamat arusaama nii nimetatud programmide ülesehitusest, reeglikombinatsioonidest kui ka võimalikest grammatiliselt ebakorrektsete lausete mustritest, siis jätsin selle variandi antud töös kõrvale.

Teine võimalus oleks täpsustada grammatikakorrektori reegleid selliselt, et arvestataks ka võimalike ebakorrektsete märgenditega. Kuna see tooks sageli kaasa ka korrektsete lausete määramise ebakorrektseteks, siis pole ma olemasolevates reeglites sellist lähenemist kasutanud.

3.3. Hindamise metoodika

Reeglite koostamisel kasutasin võrdluseks käsitsi märgendatud 298 lauset (saadaval kaasapandud CD-l, lisas 6). Sealjuures verbide kohta käivaid reegleid koostasin toetudes vaid osale korpusest, 173 lausest koosnevale osale (fail komad2.txt), sidesõnade kohta käivate reeglite väljatöötamisel kasutasin lisaks 125 lausest koosnevat osa (fail komad1.txt).

Faili komad1 tekstis oli kokku 3023 sõna, neist 432 grammatikakorrektori poolt märgendatavad, sealhulgas 323 sidesõna ja 109 küsisõna. Faili komad2 tekstis oli sõnu 3818, @ERR või @OK märgenditega neist 1028. Viimastest 489 olid verbide pöördelised vormid, 131 küsisõnad ja 408 sidesõnad. Märgendatavate sõnade sagedus tekstides, koos väärate ja õigete esinemiste osakaaludega on ära toodud tabelis 1.

Kõigil juhtudel peale verbide ning koma mittenõudvate sidesõnade oli rohkem vigaseid esinemisi (selliseid, kus vastava sõna ees oleks pidanud olema koma) kui korrektseid. Verbide puhul tuleneb suurem korrektsete esinemiste osakaal asjaolust, et lause esimene verbi pöördeline vorm määrati alati õigeks. Sellest hoolimata olid neist üle 35% märgitud vigaseks. Koma mittenõudvad sidesõnad said märgendi @ERR vaid 3,6%

esinemistest, seda kas juhtudel, mil neile eelnes madalama taseme kõrvallause, või juhtudel, kui vastava sidesõnaga alanud lause oli eelmisele vastanduva sisuga.

Hoolimata asjaolust, et tekstides oli sageli kasutatud osalausete eraldamiseks koma mittenõudvaid sidesõnu (rohkem kui 26% kõigist sidesõnadest), said vigaseks märgitud ligi pooled märgendatavatest sõnadest. Sams võib öelda, et hoolimata suurest vigade osakaalust tekstis, leidub siiski piisavalt ka korrektseid esinemisi, mille põhjal vältida ekslikku komavea tuvastamist.

Tabel 1: Märgendite sagedus sõnatüüpide lõikes

	Komad1.txt			Komad2.txt			Kokku		
	@ERR	@OK	Kokku	@ERR	@OK	Kokku	@ERR	@OK	Kokku
Verbid (#FinV)	-	-	-	174	315	489	174	315	489
Küsisõnad	73	36	109	87	44	131	160	80	240
Sidesõnad kokku	134	189	323	136	272	408	270	461	731
Koma mittenõudvad sidesõnad	4	85	89	3	102	105	7	187	194
Koma nõudvad side-sõnad	66	25	91	65	66	131	131	91	222
Kui, nagu	18	43	61	31	37	68	49	80	129
Siis, mitte	46	36	82	37	67	104	83	103	186
Kokku	207	225	432	397	631	1028	604	856	1460

Faile komad1.txt ja komad2.txt kasutasin reeglite kirjutamise järgus grammatikakorrektori reeglite korrigeerimiseks. Käsitsi märgendamisel lasin kõigepealt grammatikakorrektori märgendite lisamise sektsioonil sisendi märgendada ning seejärel kustutasin ebakorrektsed märgendid. Seega olid käsitsi märgendatud vaid need sõnade esinemised, mida ka grammatikakorrektor arvestas. Sellest tulenevalt võisid tulemuste analüüsist jääda välja vead, kus grammatikakorrektor märgendamist vajava sõna esinemist üles ei leia.

Kuigi koostasin reeglid eespooltoodud tekstide põhjal, ei suutnud need kõiki sealesinevaid vigu korrektselt tuvastada. Alapeatükis 3.4 on lähemalt kirjeldatud, mis põhjustel vead leidmata jäid ning kuidas võiks olukorda parandada.

Valminud reeglitefaili testisin lisaks 20 lausest koosneval tekstil, mida poldud reeglite koostamisel kasutatud. Võis eeldada, et sellisel tekstil ilmnevad ka need grammatikakorrektori reeglite vead, mis reeglite koostamisel kasutatud tekstil välja ei tule. Testimise tulemusi kirjeldan lähemalt alapeatükis 3.5.

3.4. Reeglite koostamisel allesjäänud vigade analüüs

Reeglite koostamisel kasutatud korpuseosa 6841 sõnal, millest 1460 märgendatud, ei suutnud reeglid valida korrektset märgendit 37 sõnal. Samuti esines käsitsi märgendatud korpusega võrreldes kahte liiki erinevusi, mis olid põhjustatud reeglite rakendamise tööviisist. Nimelt eemaldasid reeglid kümnel juhul osad sõnade analüüsid, jättes alles vaid märgendiga analüüsi. Põhjused, miks reeglid allesjäänud erinevusi tuvastada ei suutnud, võib jagada üheksaks erinevaks tüübiks.

Üheksa vea puhul oli tegu ellipsiga või verbita lauselühendiga, nt allpool toodud näitelauses (näide 1) anti sõnale ’tegi’ vale märgend, kuna eespool oli öeldis osalausest välja jäetud. Selle probleemi võiks lahendada, lastes reeglites lisaks eksplitsiitselt välja toodud verbivormidele otsida vaadeldavast osalausest teiste osalausete paralleelina ka võimalikku väljajäetud öeldist.

Näide 1: Verbita lauselühendist tulenev viga

*Meil nägu hämmingus peas aga süda rahul kui leitnant 90 kraadi kannaka **tegi** ja jooksuga sinna pani.*

Nagu eelpool mainitud, võis oodata ka seda, et morfoloogiline ühestaja jätab sõnale alles vaid tegelikult ebakorrektse märgendi. Seda tüüpi ühestamised põhjustasid grammatikakorrektori vea seitsmel korral, enamusel juhtudest oli verb analüüsitud nimisõnana. Nii ei suutnud grammatikakorrektor ka näites 2 toodud lause puhul märkida sõna ’kust’ vigaseks, kuna järgnev verb on üheselt määratud nimisõnaks. Samas tekkis erinevus käsitsi märgendatud tekstiga ka just vastupidisest olukorrast, kus näites 3 toodud lauses ühel sõnal (’täis’) oli alles jäetud võrdlemisi palju analüüse, millest üks verbi pöördelise vormina. Seetõttu märkis grammatikakorrektor vigaseks lause viimase verbi, ’ronis’, kuna sellele leiti ilma eraldajateta eelnevat tegusõna pöördeline vorm.

Näide 2: Morfoloogilisest ühestamisest tulenev viga

*Ise nad ju ütlesid et otsige süüa sealt **kust saate**("saade+0" // _S_ com sg gen ADVL).*

Näide 3: Liiga mitmesest analüüsist tulenev viga

*Üks tüüp kes oli pidev hüppeskäia oli juba paar kuud teenistuse lõpetanud, kui juua **täis** peaga üle väeosa aia **ronis**.*

Sõna *täis* analüüsid:
"täis+0" // _A_ pos AN> PRD
"täis+0" // _D_ ADVL
"täis+0" // _S_ com sg nom SUBJ
"täi+s" // _S_ com sg in ADVL NN>
"täi+s" // _V_ main indic impf ps3 sg ps af #FinV #InfP +FMV

Kuna kasutatavate tekstide autorid tõenäoliselt spellerit ei kasutanud, siis olid kahel juhul korpusesse sisse jäänud ka vead, mille õigekirjakontroll tuvastanud oleks. Nimelt olid komade järele tühikud kirjutamata jäänud, seega analüüsiti näites 4 toodud viisil kahte sõna koos nende vahel asuva komaga kui liitsõna. Sellisest märgendusest grammatikakorrektor koma üles leida ei suutnud, seega tekkis vigu reeglites, mis arvestavad kontekstis esinevate komadega. Antud vea leidmine ning parandamine oleks võrdlemisi lihtne morfoloogilisele analüüsile eelneva kirjavahemärkide õigekirja kontrolliga, kus vaadataks üle tühikute olemasolu, alustavate ja lõpetavate sulgude sobivus jms.

Näide 4: Trükiveast tulenev grammatikakorrektori viga/Spelleriga tuvastatav viga

*Alguses öeldi kohe et väljaload unustage **ära,kui** just midagi pakilist **pole**.*

Mõnevõrra raskem on tuvastada järgmist tüüpi viga, kus erinevused põhjustas asjaolu, et sõnad *vaid* ja *aga* olid kasutusel määrsõnadena, ent grammatikakorrektori reeglid arvestasid neid sidesõnadena, kuna sõnadel esinesid mõlemad märgendid. Nii näiteks ei tuvastatud allpool toodud lauses (näide5) puuduvat koma sõna ’mis’ ees, kuna talle eelnes sidesõna. Samas määrati vigaseks sõna ’vaid’, nii et vähemalt märgiti ära, et lausesse tuleks lisada koma, isegi kui selle täpne asukoht tuvastati valesti. Kuna sageli oli sidesõnana esinemine märgendatud üheselt määrsõnaks, siis võeti arvesse kõik sõna esinemised, mitte üksnes need, millel sidesõna märgend. Üheks võimalikuks lahenduseks antud olukorras oleks sidesõna analüüsi eemaldamine määrsõnana esinemisi tuvastavate reeglitega enne komakontrolli alustamist. Ent sageli on raske automaatselt otsustada, kumba sõnaliigiga on tegu. Vastasel juhul oleks see tõenäoliselt juba morfoloogilise ühestamise etapis täpsustatud.

Näide 5: Viga, mis tuleneb sõnade sidesõna kasutusest määrsõnana

*Ta ju küsis **vaid mis** vahe on automaat- ja poolautomaatkäigukastiga autol?*

Mõnes mõttes sarnane probleem tekkis sidesõnade *ja* ning *või* kahesest rollist: nad võivad esineda nii osalausete kui ka koondlauses loetelu osade eraldajana, nagu näites 6. Teisel juhul kumbki neist sidesõnadest osalause piire ei märgi, kuid kuna mõlemad variandid olid märgendatud ühtemoodi, siis lugesid grammatikakorrektori reeglid iga *ja* esinemise osalause piiriks. Lahendusena võiks sarnaselt *vaid* ja *aga* juhtumiga muuta komakontrollile eelnevalt ühe *ja* rolli märgendeid. Mõeldav oleks proovida korduvaid lauseliikmeid otsides tuvastada *ja* rolli koondlause liikmete eraldajana ning lisada sellistele *ja* esinemistele vastav märgend, mida hiljem komakontrolli reeglites arvestataks.

Süntaksianalüsaator on selle probleemi osaliselt lahendanud, märkides osalause piire – seega on osadel *ja* esinemistel märgend, mis osutab osalause piiri või võimalikku osalause piiri. Kuna osalausete piiride märgendamine toimub aga grammatiliselt korrektse õigesti märgendatud lause eeldusel, siis ei pruugi vastavad märgendid *ja* esinemistel alati õiged olla. Sellegipoolest võiks muuta reegleid selliselt, et *ja* esinemisi, millel puudub igasugune osalause piiri märge, ei arvestataks piirina. Kuid märgendatud tekstide vaatlemisel tundub, et sellised *ja* esinemised grammatikakorrektoris probleeme ei tekitanudki.

Näide 6: Koondlause eraldajana esinevast sidesõnast tulenev viga

*Kui keegi möödub ohutult **ja** sind segamata mis selles halba **on** kui sa ise takistav tegur olid.*

Probleeme tekitasid ka samamoodi reeglites osalausete eraldajana arvestatavad kirjavahemärgid. Nimelt on olemasolevates reeglites kõik kirjavahemärgid sarnaseks loetud, ent arvesse tuleks võtta, et sulud, mõttekriipsud ja jutumärgid omavad lauses mõnevõrra erinevat rolli. Nii näiteks peaks näites 7 toodud lauses sõna ’mille’ kohta käiv reegel vaatama ka jutumärkides olevale sõnale („tahtmine“) eelnevat lauseosa ning seal leiduvaid verbivorme. Seega tuleks viia reeglitesse sisse muudatus, et komakontrollil arvestataks vastavate kirjavahemärkide vahel olevale eelnevat lauseosa.

Näide 7: Viga, mis tuleneb kirjavahemärkide määramisest eraldajana

*Iseenesest on see juba "tahtmine**" mille** eest tulebki maksta.*

Seitsmel juhul ei suutnud grammatikakorrektor korrektseid otsuseid vastu võtta olukordades, mis oleksid nõudnud semantilisi teadmisi lause kohta, s.t. küsimus seisnes kas osalausete taseme määramisel, kus eri tasemel osalausete puhul tuleks *ja* ette koma panna, sama taseme osalausete puhul aga mitte, või ka lause sisu uudsuse määramises ehk siis otsustamises, kas *ja*-ga algav osalause on piisavalt uudne või vastandlik, et selle ette koma tuleks panna.

Nii on näites 8 viimane osalause eelnevale vastandatud, seega tuleks seal sidesõna *ja* ette koma panna. Näites 9 on välja toodud lause, kus *ja*-le eelneb madalama taseme kõrvallause, mistõttu tuleks samuti *ja* ette koma panna. Kuna reeglid lugesid iga sel moel eelneva osalause *ja*-tüüpi sidesõnale järgnevaga võrdsel tasemel olevaks, siis sel juhul koma *ja* ette ei soovitatud.

Näide 8: *Ja* uudse sisuga osalause ees

*Tagasi tulles, et päev oleks ikka korda läinud nägin samasugust vaatepilti, kus vanemate kallis maimuke tagaistmel istus **ja** täiesti lahti.*

Näide 9: Osalause taseme valesti määramisest tulenev viga

*Täna siis üks õnnetu opeli juht ei leidnud oma suunatule kangi üles kui reastus **ja** ühe kaubiku juhil puudus ka lisavarustuses suunatule kang.*

Reeglite koostamise etapil tuvastamata jäänud vigade leidmata jäämise põhjustest on ülevaade tabelis 2. Nagu näha, tulenes üle 27% grammatikakorrektori vigadest asjaoludest, mida saaks eelneva töötlusega (spelleripoolsete korrektuuride, morfoloogilise ühestamisega) parandada. Samas ligi 22% vigadest oleks võimalik eemaldada grammatikakorrektorisisese eelneva märgendamisega (sidesõnade rollid määrsõna või koondlause osade eraldajana, hõlmavate kirjavahemärkide parem arvestamine). Peaaegu pooled vead (ellips, semantilise info alusel toimivad grammatikareeglid) vajavad suuremat tööd, ent tõenäoliselt oleks võimalik näiteks heuristikuid kasutades ka neid parandada, isegi kui pole võimalik reeglitel saavutada sajaprotsendist täpsust.

Tabel 2: Treeningkorpuses vigade mittetuvastamise põhjused

	Komad1.txt	Komad2.txt	**Kokku**
Ellips või lauselühend	2	7	9

Morfoloogilise ühestamise viga	2	5	7
Liiga mitmene analüüs	0	1	1
Koma sõna keskel	1	1	2
Vaid, *aga* määrsõnana	0	3	3
Ja koondlause eraldajana	0	2	2
Kirjavahemärk eraldajana	2	1	3
Semantilist teadmist vajav otsus	5	2	7
Muu	2	1	3
Kokku	14	23	37

Tabelis 3 on toodud ülevaade reeglite koostamisel allesjäänud vigadest reeglites kasutatud sõnahulkade lõikes.

Verbid moodustasid kõigist tuvastamata jäänud vigadest üle veerandi. Põhjused, miks puuduvaid komakohti ei leitud, olid mitmesugused, jagunedes seitsme eelpooltoodud tüübi vahel.

Küsisõnadest jäid tuvastamata suhteliselt vähesed komavead. Küsisõnad, mille ette oleks tulnud panna koma, jäid leidmata kolmel juhul, sealjuures kahel korral seetõttu, et neile loeti eelnevaks kas sidesõna või kirjavahemärk. Kolmandal juhul oli tegu ellipsiga.

Kuna sidesõnu oli märgendatavatest sõnadest kõige rohkem, jäi neis ka kõige rohkem vigu alles. Sidesõnade *kui* ja *nagu* puhul olid vead tingitud osalausest puuduvast verbist, kas siis oli tegu ellipsiga või, enamusel juhtudest, oli morfoloogilisel ühestamisel verb mõneks muuks sõnaliigiks määratud. Seevastu sõna *siis* puhul olid vigade tuvastamata jäämisel erinevad põhjused. Sõnale *mitte* määras grammatikakorrektor alati õige märgendi.

Koma nõudvatest sidesõnadest tekkis probleem kahel korral sõna *vaid* esinemisest määrsõnana. Ülejäänud kahest korrast oli ühel puhul tegu koma järele tühiku kirjutamata jätmisest, teisel puhul aga sellest, et sõna *et* järgnes vahetult kirjavahemärkides tsitaadile.

Korpuses esines 7 lauset, kus koma mittenõudva sidesõna ette oleks tulnud kas osalause vastandlikkuse või eelneva madalama taseme kõrvallause tõttu koma panna. Kõik 7

vigast juhtu jäid ka tuvastamata, kuna nende jaoks reegleid pole välja töötatud (see nõuaks lähemat analüüsi ja teoreetiliselt semantilise info kasutamist). Samas moodustasid vigased juhud kõigist vastavat tüüpi sidesõna esinemistest vaid 3,6%.

Tabel 3: Treeningkorpuses tuvastamata jäänud vead sõnatüüpide lõikes

	Komad1.txt	Komad2.txt	Kokku
Verbid (#FinV)	-	10	10
Küsisõnad	1	2	3
Sidesõnad	13	11	24
Koma mittenõudvad sidesõnad	5	2	7
Koma nõudvad sidesõnad	2	2	4
Kui, nagu	0	6	6
Siis, mitte	6	1	7
Kokku	14	23	37

3.5. Testimine tundmatu tekstiga

Reeglifaili testisin samast Delfi korpusest pärit 30 lausel (575 sõnal), mida reeglite koostamise faasis vaadanud ei olnud. Testkorpuses kuulusid märgendamisele 124 sõna, neist 41% osutasid vigastele kohtadele. Verbi pöördelisi vorme oli märgendatavate hulgas 71, @ERR märgendiga neist 32%.

Grammatikakorrektor tegi testkorpusel märgendite valimisel vea kokku üheksas kohas ehk 7,3% juhtudest, lisaks eemaldas kahel korral mitmese analüüsiga sõnadelt määrsõna analüüsid. Kuuel juhul jäi komaviga vastavas sõnas tuvastamata, kolmel korral aga määrati viga sinna, kus seda tegelikult ei esinenud. Vead tekkisid kuuel juhul verbide märgendites, lisaks sõnade *kas*, *siis* ja *nagu* märgendamisel.

Viiel juhul oli põhjuseks mõni juba treeningkorpusel vigade sissejäämist põhjustanud asjaoludest. Nimelt ühel lausel oli välja jäetud verbi *on* vorm, kahel korral oli tegu koondlause liikmete eraldajana esineva *ja* arvamisega osalausete eraldajate hulka. Ühel juhul olid sõna kirjutamisel tähemärgid vahetusse läinud ning sõna *ühesnada* (mõeldud oli *üheksanda*) analüüsiti verbivormina, teisel juhul tuli viga verbivormi *saan*

analüüsimisest nimisõnana.

Ülejäänud neljal juhul seevastu tulenesid grammatikakorrektori eksimused reeglites leiduvatest vigadest, mida oleks sarnaste juhtudega arvestamisel kerge parandada. Valdavalt oli tegu juhtudega, mida reeglid ei katnud, s.t. sõna märgendi määras kas viimane, vaikimisi reegel või jäidki sõnale mõlemad märgendid alles. Nii jäi lauses „*On olemas selline jook www.teeccino.com mis väidetavalt* ***nagu*** *kohvi ja ergutab sind.“* sõna *nagu* mitmese analüüsiga. Sõna *siis* märgendatakse viimase, vaikimisi õigeks määramisega valesti lauses „*Kui vee värvuse, lõhna, maitse ja hägususe kohta pole piirnorme kehtestatud* ***siis*** *ei hakkagi paljud kraanivett jooma“*. Sõna *kas* puhul tuleks sisse viia reegel *kas ... või* kasutuse kohta, antud juhul märgiti tegelikult korrektne osalause vigaseks. Üks reegel, reeglifailis real 185, osutus valeks, kuna juhtude täpsustamisel on kogemata kontekstitingimusse märgitud infinitiivi asemel nii infinitiiv kui verbi pöördeline vorm. Seega jääb tuvastamata puuduv koma lauses *„Kui ununeb pean ostma“*.

Korrektselt tuvastatud komavigade osakaal kõigist grammatikakorrektori poolt antud komavigade märgenditest ehk grammatikakorrektori täpsus testkorpusel on 93,8%. Saagis ehk leitud komavigade suhe korpuses leidunud (ja käsitsi märgendatud) komavigadesse on 94,1%. Siinkohal tuleb mainida, et kõrge saagis tuleneb asjaolust, et selle arvutamisel ei arvestatud vigadega, mis esinesid lausetes, kus neid kasutusel olnud märgendisüsteemi järgides (s.t. märgendades ainult side-, küsisõnu ja verbi pöördelisi vorme) märkida polnud võimalik.

Testkorpuse 30 lausest esines komaviga 27 lauses. Grammatikakorrektor leidis vea 24 lauses, seega eksiti kokku kolme lause juures, nii et komavigade suhtes jäi esile tõstmata tervelt 10% testkorpuse lausetest. Muudes lausetes esinenud vigade korral leiti vea tuvastamata jäämisel komaviga mõne teise sõna abil, näiteks verbi asemel sidesõna abil või vastupidi, või valepositiivse tulemuse korral oli mõni lause muu osa grammatiliselt ebakorrektne.

Teistele keeltele kitsenduste grammatika alusel loodud grammatikakorrektoritega võrreldes on tulemused üsnagi head. Viimaste täpsused jäid vahemikku 70-95%. Kuid sellisel võrdlusel tuleks pöörata tähelepanu kahele olulisele erinevusele antud töö raames koostatud grammatikavigade tuvastaja ning võrreldavate grammatikakorrektorite vahel. Esiteks, komavigade tuvastamisega tegeles vaid baski

keele grammatikakorrektor, mille tulemused polnud artikli avaldamise ajaks veel selgunud, teiste keelte puhul olid sihtgrupiks aga mitmed muud veatüübid. Teiseks tuleks välja tuua, et rootsi, norra ja taani keele puhul oli tegu tõesti korrektoriga, s.t. lisaks vigade tuvastamisele pakuti välja ka parandatud laused, mille korrektsust ning asjakohasust süsteemi hindamisel arvesse võeti.

Kokkuvõte

Käesoleva magistritöö eesmärgiks oli luua eesti keelele komakasutust kontrolliv grammatikavigade tuvastaja. Töö käigus koostati kitsenduste grammatika reeglitefail, mis sisaldab 90 reeglit.

Töös kasutati enam kui 6800-sõnalist osa (298 lauset) internetiportaalist kogutud komavigu sisaldavate lausete korpusest. Enne grammatikakorrektori töö algust analüüsiti korpus morfoloogiliselt ning süntaktiliselt programmi Estcgparser abil.

Vigade tuvastamiseks märgendati tekstis esinevad verbi pöördelised vormid ning side- ja küsisõnad, seejärel rakendati kitsenduste grammatika reegleid, mis valisid vastavalt sõna kontekstile lauses märgenditest välja kas komavea või korrektset kohta märkiva märgendi. Vigade osakaal korpuses oli suur, komaveale osutava märgendi said 41% arvestatavatest sõnadest.

Reeglite koostamisel kasutatud korpuses jäi tuvastamata ligi 4,5% vigadest. Neist üle veerandi puhul tulenes eksimus asjaoludest, mida saaks eelneva analüüsiga parandada, näiteks õigekirjakontrolliga või morfoloogilise ühestaja kohaldamisega grammatiliselt ebakorrektsetele lausetele sobivamaks. Üle 40% allesjäänud eksimustest olid põhjustatud automaatselt raskesti analüüsitavast kontekstist, kus komakasutuse otsustab osalause tähendus või kus osa lauseliikmeid oli osalausest välja jäetud. Ülejäänud eksimused oleksid edasise tööga mõnevõrra kergemini kõrvaldatavad.

Komavigade tuvastajat testiti 30-lauselisel tekstil (575 sõna) sama korpuse reeglite väljatöötamisel kasutamata osast. Testkorpusel eksisid reeglid üheksal korral, sellest kolmel korral määrati vigaseks tegelikult korrektse märgendiga sõna. Eksimustest ligi pooled olid põhjustatud juba reeglite koostamisel lahendamata jäetud asjaoludest, pooli eksimusi oleks olnud võimalik vältida kasutades reeglite kirjutamisel suuremat korpust. Seevastu ühtegi lauset ekslikult vigaseks ei märgitud, 27 komavigu sisaldavast lausest jäid tuvastamata kolm lauset. Märgendatavatel sõnadel oli komavigade leidmise täpsus 93,8%, saagis 94,1%.

Võib öelda, et saadud tulemus on võrreldav teistele keeltele loodud grammatika-korrektoritega. Samas tuleks kindlasti eesti keele grammatikakorrektorit edasi arendada, reeglite väljatöötamisel suuremat korpust kasutades on lootust tulemusi parandada. Samuti tuleks lisaks komavigadele haarata korrektorisse ka ühilduvus-, kokku-

lahkukirjutamis- ja muud grammatikavead, siis oleks suure tõenäosusega olulisem roll ka süntaktilisel infol. Mõeldes lõppkasutajale, peaks arvestama ka parandusettepanekute lisamisega korrektorisse. Lisaks otseste vigade tuvastamisele võiks samuti tegelda stiilivigade leidmise ning parandamisega.

Rule-based grammar checker for detecting comma mistakes in Estonian texts

Master thesis
Krista Liin
Abstract

The purpose of this thesis was to create an automatic grammar checker that would detect comma mistakes in written Estonian. As of yet, the checker does not suggest corrections, but that function could be added to the existing system. The goal was to achieve as good a precision as possible. This was the third attempt to create a grammar checker for Estonian, but the method and corpora used were different from the previous attempts.

The grammar checker rules are based on the Constraint Grammar Formalism that was developed at University of Helsinki and has been used to build grammar checkers in several other languages. The motivation for using this formalism was that the morphological unification as well as the syntactic analyzer for Estonian are based on the same formalism, so the grammar checker could be easily integrated in the existing systems.

The corpus used for rule development and testing consists of grammatically uncorrect sentences gathered from an Internet site. 298 sentences (over 6800 words) were first morphologically and syntactically analyzed, then manually tagged for comma error detection and used for comparison in the development phase. Finite verb forms, interrogative words and conjugations were tagged and marked as correct or incorrect, depending on whether there was a comma mistake before those words.

About 4,5% of mistakes could not be correctly detected in the training corpora. The problems were mainly caused by incorrect spelling, previous tagging, or situations where the usage of comma depends on sematic information. The comma usage in Estonian is rather strictly regulated, but in many cases knowledge about the meaning of the sentence is needed to decide, whether or not to use a comma.

The 90 constraint rules were tested on a 30-sentence test corporus of the same text type. A precision of 93,8% and recall of 94,1% was achieved on tagged words. About half of the grammar checker's mistakes were caused by the same kind of problems as on the training corpus, the rest could have been avoided with trainning on larger corpora.

In conclusion, the results achieved are comparable to other grammar checkers. The grammar checker for Estonian should be further developed using larger corpora and targeting also other error types, such as agreement mistakes.

Kasutatud kirjandus

1. Aldezabal I., Aranzabe M., Arrieta B., Maritxalar M., Oronoz M. Toward a punctuation checker for Basque. ATALA workshop of punctuation (Paris), 2003. Arvutivõrgus kättesaadav: http://ixa.si.ehu.es/Ixa/Argitalpenak/-Artikuluak/1069080468/publikoak/Toward-a-punctuation-checker-for-Basque.pdf (26. mai 2008[3])
2. Alegria I., Arrieta B., Díaz de Ilarraza A., Izagirre E., Maritxalar M. Using Machine Learning Techniques to Build a Comma Checker for Basque. Coling-ACL. Sydney. Australia. lk.1-8, 2006. Arvutivõrgust kättesaadav: http://ixa.si.ehu.es/Ixa/Argitalpenak/Artikuluak/1150185248/publikoak/komak-ML.pdf (26. mai 2008)
3. Ansa O., Arregi X., Arrieta B., Ezeiza N., Fernandez I., Garmendia A., Gojenola K., Laskurain B., Martínez E., Oronoz M., Otegi A., Sarasola K., Uria L. Integrating NLP Tools for Basque in Text Editors. Workshop on International Proofing Tools and Language Technologies, University of Patras (Greece), 2004. Arvutivõrgus kättesaadav: http://ixa.si.ehu.es/Ixa/Argitalpenak/Artikuluak/1088503737/-publikoak/04_ProofingTools.doc (26. mai 2008)
4. Arppe, Antti. Developing a Grammar Checker for Swedish. Nordgård, Torbjørn (ed.) Proceedings from the 12th *Nordiske datalingvistikkdager*, Trondheim, December 9-10, 1999. Department of Linguistics, Norwegian University of Science and Technology (NTNU), Trondheim, Norra, 2000. Arvutivõrgus kättesaadav: http://www.ling.helsinki.fi/~aarppe/Publications/Nodalida-99.pdf (16. mai 2008)
5. Arppe, Antti; Birn, Juhani; Westerlund, Fredrik. Lingsoft's Swedish Grammar Checker. http://www2.lingsoft.fi/doc/swegc/ (26. mai 2008)
6. Bick, Eckhard. A Constraint Grammar Based Spellchecker for Danish with a Special Focus on Dyslexics. SKY Journal of Linguistics, vol 19:2006. Arvutivõrgus kättesaadav: http://www.ling.helsinki.fi/sky/julkaisut/-SKY2006_1/1.6.1.%20BICK.pdf (26. mai 2008)
7. Carlberger, Johan; Domeij, Rickard; Kann, Viggo, Knutsson, Ola. The Development and Performance of a Grammar Checker for Swedish: A Language Engineering Perspective. 2004. Arvutivõrgus kättesaadav: http://www.csc.kth.se/tcs/projects/granska/rapporter/granskareport.pdf (26. mai 2008)
8. De Smedt, Koenraad; Rosén, Victoria. Automatic proofreading for Norwegian: The challenges of lexical and grammatical variation. Nordgård, T. (ed.) NODALIDA '99: Proceedings from the 12th "Nordiske datalingvistikkdager", Trondheim, 9-10 December, 1999 (pp. 206-215). Trondheim: NTNU, 1999. Arvutivõrgus kättesaadav: http://ling.uib.no/~desmedt/papers/nodalida99pub.html (26. mai 2008)
9. Díaz de Ilarraza A., Gojenola K., Oronoz M. Design and Development of a System for the Detection of Agreement Errors in Basque. Computational Linguistics and Intelligent Text Processing. Lecture Notes in Computer Science, Vol. 3406 ISBN: 3-540-24523-5 pp793-803. Springer-Verlag GmbH, CICLing-2005, Sixth International Conference on Intelligent Text Processing and

[3] Veebiaadresside taga sulgudes on vastava aadressi viimatise külastuse kuupäev

Computational Linguistics, Mexico City, Mexico, 2005. Arvutivõrgust kättesaadav: http://ixa.si.ehu.es/Ixa/Argitalpenak/Artikuluak/1101134041/-publikoak/CICLing2005.pdf (26. mai 2008)
10. Erelt, Mati. Lause õigekeelsus. Juhatused ja harjutused. Bookmill, 2006.
11. Hagen, Kristin; Lane, Pia. "Det er fort gjort og skrive feil." En presentasjon av en automatisk grammatikkontroll for bokmål. Foredrag på Mons, Oslo, 2001. Arvutivõrgus kättesaadav: http://www.hf.uio.no/tekstlab/prosjekter/Mons.gr-sjekker.htm (26. mai 2008)
12. Hagen, Kristin; Johannessen, Janne Bondi; Lane, Pia. Some problems related to the development of a grammar checker. Foredrag på NoDaLiDa, Uppsala, 2001. Arvutivõrgus kättesaadav: http://www.hf.uio.no/tekstlab/prosjekter/-NoDaLiDa_gram.html (26. mai 2008)
13. Hagen, Kristin; Johannessen, Janne Bondi; Lane, Pia. The performance of a grammar checker with deviant language input. Proceedings of the 19th international conference on Computational linguistics - Volume 2. Taipei, Taiwan, 2002. Arvutivõrgust kättesaadav: http://portal.acm.org/citation.cfm-?id=1071884.1071894 (26. mai 2008)
14. Hardt, Daniel. Transformation-Based Learning of Danish Grammar Correction. Proceedings of RANLP 2001. Arvutivõrgus kättesaadav: http://www.id.cbs.dk/~dh/papers/ranlp.pdf (26. mai 2008)
15. Kaalep, Heiki-Jaan. Eesti keele ressursside loomine ja kasutamine keeletehnoloogilises arendustöös. *Dissertationes philologicae estonicae Universitatis Tartuensis*. Tartu Ülikooli Kirjastus, 1999. Arvutivõrgust kättesaadav: http://www.cl.ut.ee/yllitised/hkaalep_dr.pdf (26. mai 2008)
16. Karlsson, Fred; Voutilainen, Atro; Heikkila, Juha; Anttila, Arto. Constraint Grammar: A Language-independent System for Parsing. Berlin and New York, Walter de Gruyter, 1995. Arvutivõrgus kättesaadav: http://books.google.com/-books?hl=en&lr=&id=70IvVPlH63cC&oi=fnd&pg=PP10&dq=Constraint+Grammar:+A+Language-independent+System+for+Parsing&ots=mA5pxnqGGb&-sig=ajX5aV_JUpdNp5FmhqpscP9UhIY (26. mai 2008)
17. Kuboň, Vladislav; Holan, Tomáš, Plátek, Martin. A Grammar-checker for Czech. ÚFAL Technical Report TR-1997-02. Universitas Carolina Pragensis, 1997.
18. Kurg, Kristiina. Katsetused eesti keele grammatikakorrektori loomisel. Bakalaureusetöö. Juhendaja Heli Uibo. Tartu Ülikool, Matemaatika-informaatikateaduskond, 2005.
19. Lunt, Lauri. Eesti keele grammatikakontrollija: esimesed sammud. Bakalureusetöö. Juhendaja Heli Uibo. Tartu Ülikool, Matemaatika-informaatika-teaduskond, 2003.
20. Riiklik programm „Eesti keele keeletehnoloogiline tugi (2006-2010)" http://www.keeletehnoloogia.ee/ (26. mai 2008)
21. Roosmaa, Tiit; Koit, Mare; Muischnek, Kadri; Müürisep, Kaili; Puolakainen, Tiina; Uibo, Heli. Eesti keele formaalne grammatika. Tartu Ülikooli Kirjastus, Tartu, 2001
22. VISL CG-3 Development Information, http://visl.sdu.dk/cg3.html (26. mai 2008)
23. World of VISL, http://visl.sdu.dk/ (26. mai 2008)

Lisad

Lisa 1. Kasutusjuhend
Lisa 2. Sisendi eeltöötluse skript
Lisa 3. Reeglifail
Lisa 4. Komavigade tuvastaja sisend
Lisa 5. Komavigade tuvastaja väljund
Lisa 6. CD

Lisa 1. Kasutusjuhend

Grammatikakorrektori kasutamine eeldab programmide Estcgparseri ja VISL CG-3 olemasolu.

Estcgparser on kättesaadav aadressilt http://lepo.it.da.ut.ee/~kaili/nptool/

VISL CG-3 on kättesaadav aadressilt http://beta.visl.sdu.dk/cg3.html, installeerimis-juhend samal leheküljel asuvast manuaalist.

Sisendi eeltöötlus:

cat tekstifail.txt | estscparser.exe | convert.sh > tekstifail.snx

Komavigade tuvastaja käivitamine:

cat tekstifail.snx | vislcg3 –g komavead.rle > margendatud.fail

Juhul, kui Estcgparser ja VISL-CG3 on installeeritud samas operatsioonisüsteemis, saab komavigade tuvastaja käivitada ühe käsuna:

cat tekstifail.txt | estscparser.exe | convert.sh | vislcg3 –g komavead.rle > margendatud.fail

Soovi korral võib määrata trace parameetri, et näha, mis reeglid täpsemalt kus rakenduvad:

cat tekstifail.txt | estscparser.exe | convert.sh | vislcg3 –g komavead.rle --trace > margendatud.fail

Lisa 2. Sisendi eeltöötluse skript

```
#!/bin/sh
#Eemaldada @ märgid (segab MAP käsku)
#Viia märgendite vahele sisse tühikud
#Viia analüüsi algul olev sõnavorm jutumärkidesse

tr -d '@' \
| tr '\015' '@' \
| tr -d '@' \
| sed 's/^\([^ ].*\)$/"<\1>"/' \
| sed 's/^    \([^ ]*\)/    "\1"/' \
| sed 's#//_#// _#'
```

Lisa 3. Reeglifail

```
 1: DELIMITERS = "<$LA$>" "<$LL$>";
 2:
 3: LIST >>> = >>>;
 4: LIST <<< = <<<;
 5:
 6: SECTION
 7:
 8: #Koma nõudvad sidesõnad
 9: LIST Et = ("et+0") ("ent+0") ("kuid+0") ("aga+0" // _J_) ("vaid+0"
    // _J_) ("sest+0") ("kuna+0") ("kuigi+0"); #NB! Sageli valesti: (Aga
    _D_)
10: #Komata kasutatavad sidesõnad
11: List Ja = ("ja+0" _J_) ("ning+0") ("ehk+0") ("ega+0") ("või+0" //
    _J_);
12: #Võrdlustes kasutatavad sidesõnad
13: List Kui = ("kui+0") ("nagu+0");
14: #Küsisõnade list - mida, kuhu jne . Kuna alati ei ole _P_ inter,
    siis välja toodud
15: List Mis = ("mis+0") ("mis+da") ("mis+le") ("mis+l") ("mis+lt")
    ("mis+sse") ("mis+s") ("mis+st") ("mis+ks") ("mis+ni") ("mis+na")
    ("mis+ta") ("mis+ga");
16: List Kes = ("kes+0") ("kes+da") ("kes+le") ("kes+l") ("kes+lt")
    ("kes+sse") ("kes+s") ("kes+st") ("kes+ks") ("kes+ni") ("kes+na")
    ("kes+ta") ("kes+ga");
17: List Kysi = (_P_ inter) ("kus+0") ("kust+0") ("kuhu+0") ("millal+0")
    ("miks+0") ("kuidas+0") ("kas+0") ("mis_pärast+0") ("mis_moodi+0")
    ("kus_kohas+0") ("mitu+0");#kes != mis
18: #Tihti määrsõnadena
19: List Siis = ("siis+0") ("mitte+0"); #NB! (_J_)
20: #Kirjavahemärk (koma, koolon..), lause algus - koma nõudvatele
    sidesõnadele
21: List Eraldaja = (_Z_) ("\#\#\#\#") (>>>);
22: # Need, mis võivad olla nii '_, _' kui ka ', _ _'
23: List EtEtteMitmene = ("see+ks") ("selle_pärast+0") ("eeldus+l")
    ("eeldus+l");
24: List KuiEtteMitmene = ("enne+0") ("juht+l") ("siis+0");
25: #olgugi et, ainult et, vaevalt et, peaasi et, ilma et, mitte et.
    Õige: _, mitte et
26: List EtEesKomaga = ("olgugi+0") ("ainult+0") ("vaevalt+0")
    ("pea_asi+0") ("ilma+0") ("mitte+0");
27:
28: List Verb = (_V_ main inf) (\#FinV);#Öeldiseks võib olla pöördeline
    vorm, da-infinitiiv
29: List Verbivorm = (_V_ main inf) (_V_ main partic past) (\#FinV); #-
    tud/nud -da pöördeline
30:
31:
32: MAP (@ERR @OK) Et;
33: MAP (@ERR @OK) Ja;
34: MAP (@ERR @OK) Kui;
35: MAP (@ERR @OK) Siis;
36: MAP (@ERR @OK) Kysi;
37: MAP (@ERR @OK) Mis;
38: MAP (@ERR @OK) Kes;
39: MAP (@ERR @OK) (\#FinV);
40:
41: #-------------------------------------------------------------
```

```
42: #Üldised reeglid
43: #------------------------------------------------------------
44: #Lause algul on kõik õige
45: SELECT (@OK) (0 Et OR Ja OR Kui OR Kysi OR Mis OR Kes) (-1
    ("\#\#\#\#") OR ("(") OR (>>>));
46:
47: #-------
48: #Siis reeglid - kaks sidesõna järjest
49: #siis, kui; kui siis vs kui, siis - tuleb täpsustada enne 'kaks
    sidesõna kõrvuti' reeglit
50: REMOVE (@OK) (0 ("kui+0")) (-1 ("siis+0")); #'siis, kui' peab olema
51: REMOVE (@OK) (0 ("siis+0")) (-1 ("kui+0")) (1 Verbivorm) (-1*
    Verbivorm BARRIER Eraldaja OR Ja); #Kui teha, siis teha hästi.
52: REMOVE (@ERR) (0 ("siis+0")) (-1 ("kui+0")) (1 Verbivorm) (NOT -1*
    Verbivorm BARRIER Eraldaja OR Ja); #Ja kui siis hakkas sadama.
53: REMOVE (@OK) (0 ("siis+0")) (-1 ("kui+0")) (1 Kysi OR Kes OR Mis);
    #Ja kui, siis mida?
54:
55: #Kysi reeglid - mitu küsisõna järjest
56: #kes, kus, mis, millal?
57: SELECT (@ERR) (0 Kysi OR Mis OR Kes) (-1 Kysi OR Mis OR Kes) (NOT 1*
    Verbivorm BARRIER Eraldaja OR (Com));
58: SELECT (@OK) (0 Kysi OR Mis OR Kes) (-1 (Com)) (-2 Kysi OR Mis OR
    Kes) (NOT 1* Verbivorm BARRIER Eraldaja OR (Com));
59: #kes mida väärtustab
60: SELECT (@OK) (0 Kysi OR Mis OR Kes) (-1 Kysi OR Mis OR Kes);
61:
62: #-------
63: #Üldised reeglid:
64: #mitu sidesõna kõrvuti: koma ainult esimese ees
65: SELECT (@OK) (0 Et OR Ja OR Kui OR Siis OR Kysi OR Mis OR Kes) (-1
    Et OR Ja OR Kui);
66: #Mürakarud: ta ütles, et ... ja_,_ et ...
67: SELECT (@ERR) (0 Et OR Ja OR Kui OR Siis OR Kysi OR Mis OR Kes) (-1
    (Com)) (-2 Et OR Ja OR Kui);
68:
69: #-------------
70: #Siis reeglid - jätkub
71: #-------------
72: REMOVE (@ERR) (0 Siis) (-1 ("ka+0")); #jääda ka siis mitte kui oled
    - ADVL
73: REMOVE (@OK) (0 ("siis+0")) (-1* ("kui+0") OR ("kuna+0") OR ("et+0")
    BARRIER Eraldaja LINK 1* Verbivorm BARRIER Eraldaja OR Ja OR Kysi OR
    Kes OR Mis ); #Kui teha, siis teha hästi.
74: REMOVE (@ERR) (0 Siis) (1 Eraldaja); #Või siis mitte. Kes see võttis
    siis?#Pärast kui'sid: Kui jääd ette uimama, siis ...
75: REMOVE (@ERR) (0 Siis) (NOT 1* Verbivorm BARRIER Eraldaja OR Ja OR
    Et OR Kui OR Siis OR Kysi OR Kes OR Mis); #Tuli siis värsket verd
    ja...
76: REMOVE (@ERR) (0 Siis) (-1 (SUBJ)) (1 Verbivorm); #Nii me siis
    magasimegi
77: REMOVE (@ERR) (0 Siis) (-1* (_V_ aux) OR (_V_ mod) BARRIER
    Verbivorm); #Kus sa saad siis vahetust teha
78: REMOVE (@ERR) (0 Siis);
79:
80:
81: #------------------------------------------------------------
82: #Et & Kui Erijuhud
83: #------------------------------------------------------------
84: #selleks et, sellepärast et, selle asemel et, eeldusel et, nii et,
    siis kui, enne kui, juhul kui, nii palju kui, samal ajal kui
```

```
 85: #Õiged: _, sellepärast et;_ sellepärast, et
 86: SELECT (@ERR) (0 ("et+0")) (-1 EtEtteMitmene) (NOT -2 Eraldaja); #
     selleks et
 87: SELECT (@ERR) (0 ("et+0")) (-1 (Com)) (-2 EtEtteMitmene) (-3 (Com));
     #, selleks, et
 88: SELECT (@OK) (0 ("et+0")) (-1 EtEtteMitmene) (-2 Eraldaja); #,
     selleks et
 89: SELECT (@OK) (0 ("et+0")) (-2 EtEtteMitmene) (-1 (Com)); # selleks,
     et
 90: SELECT (@ERR) (0 ("kui+0")) (-1 KuiEtteMitmene) (NOT -2 Eraldaja); #
     enne kui
 91: SELECT (@ERR) (0 ("kui+0")) (-1 (Com)) (-2 KuiEtteMitmene) (-3
     (Com)); #, enne, kui
 92: SELECT (@OK) (0 ("kui+0")) (-1 KuiEtteMitmene) (-2 Eraldaja); #,
     enne kui
 93: SELECT (@OK) (0 ("kui+0")) (-2 KuiEtteMitmene) (-1 (Com)); # enne,
     kui
 94: SELECT (@ERR) (0 ("et+0")) (-1 ("asemel+0")) (-2 ("see+0")) (NOT -3
     Eraldaja); # selle asemel et
 95: SELECT (@ERR) (0 ("et+0")) (-1 (Com)) (-2 ("asemel+0")) (-3
     ("see+0")) (-4 (Com)); #, selle asemel, et
 96: SELECT (@OK) (0 ("et+0")) (-1 ("asemel+0")) (-2 ("see+0")) (-3
     Eraldaja); #, selle asemel et
 97: SELECT (@OK) (0 ("et+0")) (-1 (Com)) (-2 ("asemel+0")) (-3
     ("see+0")); # selle asemel, et
 98: SELECT (@ERR) (0 ("kui+0")) (-1 ("palju+0")) (-2 ("nii+0")) (NOT -3
     Eraldaja); # nii palju kui
 99: SELECT (@ERR) (0 ("kui+0")) (-1 (Com)) (-2 ("palju+0")) (-3
     ("nii+0")) (-4 (Com)); #, nii palju, kui
100: SELECT (@OK) (0 ("kui+0")) (-1 ("palju+0")) (-2 ("nii+0")) (-3
     Eraldaja); #, nii palju kui
101: SELECT (@OK) (0 ("kui+0")) (-1 (Com)) (-2 ("palju+0")) (-3
     ("nii+0")); # nii palju, kui
102: SELECT (@ERR) (0 ("kui+0")) (-1 ("aeg+l")) (-2 ("sama+l")) (NOT -3
     Eraldaja); # samal ajal kui
103: SELECT (@ERR) (0 ("kui+0")) (-1 (Com)) (-2 ("aeg+l")) (-3
     ("sama+l")) (-4 (Com)); #, samal ajal, kui
104: SELECT (@OK) (0 ("kui+0")) (-1 ("aeg+l")) (-2 ("sama+l")) (-3
     Eraldaja); #, samal ajal kui
105: SELECT (@OK) (0 ("kui+0")) (-1 (Com)) (-2 ("aeg+l")) (-3
     ("sama+l")); # samal ajal, kui
106: #nii et - võib ka päris ilma olla (ainult teine juht vigane) -
     püsiühendites jms
107: SELECT (@OK) (0 ("et+0")) (-1 ("nii+0")) (NOT -2 Eraldaja);# nii et
108: SELECT (@ERR) (0 ("et+0")) (-1 (Com)) (-2 ("nii+0")) (-3 (Com)); #,
     nii, et
109: SELECT (@OK) (0 ("et+0")) (-1 ("nii+0")) (-2 Eraldaja); #, nii et
110: SELECT (@OK) (0 ("et+0")) (-2 ("nii+0")) (-1 (Com)); # nii, et
111:
112: #isegi kui, olgugi et, ainult et, vaevalt et, peaasi et, ilma et,
     mitte et - koma ees
113: SELECT (@ERR) (0 ("et+0")) (-1 EtEesKomaga) (NOT -2 Eraldaja);
114: SELECT (@OK) (0 ("et+0")) (-1 EtEesKomaga) (-2 Eraldaja);
115: SELECT (@ERR) (0 (kui+0")) (-1 ("isegi+0")) (NOT -2 Eraldaja);
116: SELECT (@OK) (0 ("kui+0")) (-1 ("isegi+0")) (-2 Eraldaja);
117:
118: SELECT (@ERR) (0 Et) (-1 ("see+0")); # see et - õige: see, et
119: SELECT (@OK) (0 Et) (-1 (Com)) (-2 ("see+0")); # õige: see, et
120:
121: #----------------------------------------
122: #Et ülejäänud
```

```
123: #----------------------------------------
124: SELECT (@ERR) (0 Et) (NOT -1 Eraldaja) (*1 (\#FinV)) (*-1 (\#FinV));
     #Et ees peab olema koma
125: SELECT (@ERR) (0 Et) (-1 (Com)) (NOT *1 (\#FinV)); #Kui paremal
     öeldist pole, siis Et ette koma pole vaja ('mina aga mitte')
126: SELECT (@ERR) (0 Et) (NOT -1 Eraldaja);
127: SELECT (@OK) (0 Et); #kui vigu ei leitud, on korras
128:
129: #------------------------------------------------------------
130: #Kui reeglid
131: #------------------------------------------------------------
132: #nii ... kui ka - koma ei ole
133: SELECT (@OK) (0 ("kui+0")) (1 ("ka+0")) (-1* ("nii+0") BARRIER
     Eraldaja); #mõned osad võib ka ära jätta
134: #Link Verb, kuna: venelased räägivad nii millegipärast kui on...
135: SELECT (@OK) (0 ("kui+0")) (-2* ("nii+0") BARRIER Eraldaja LINK NOT
     -1 Verb);
136: #Kui järgneb ja eelneb FinV -> koma
137: SELECT (@ERR) (0 Kui) (NOT -1 Eraldaja) (*-1 (\#FinV) BARRIER
     Eraldaja OR Ja) (*1 Verbivorm BARRIER Eraldaja OR Et OR Ja OR Kysi
     OR Kes OR Mis OR Siis OR Kui);
138: SELECT (@ERR) (0 Kui) (NOT -1 Eraldaja) (*1 Verbivorm BARRIER
     Eraldaja OR Et OR Ja OR Kysi OR Kes OR Mis OR Siis OR Kui);
139: SELECT (@OK) (0 Kui) (-1 Eraldaja) (*1 Verbivorm BARRIER Eraldaja OR
     Et OR Ja OR Kysi OR Kes OR Mis OR Siis OR Kui);
140: SELECT (@OK) (0 Kui) (NOT -1 Eraldaja) (*1 Eraldaja LINK 1 Kes OR
     Mis OR Kysi OR Et OR Siis OR Kui BARRIER Verbivorm); #nagu see, mis;
     kui nii, siis
141: SELECT (@OK) (0 Kui) (NOT -1 Eraldaja) (*1 Kes OR Mis OR Kysi OR Et
     OR Siis OR Kui OR Eraldaja BARRIER Verbivorm); # nagu/OK mu sõber
     kes/ERR
142:
143:
144: #----------------------------------------
145: #Ja
146: #----------------------------------------
147: #Ja: tavaliselt koma pole
148: #Võib olla komaga, kui järgnev mõte on väga uus või vastandlik
149: #SELECT (@ERR)
150: SELECT (@OK) (0 Ja);
151:
152: #----------------------------------------
153: #Kysi
154: #----------------------------------------
155: #sõltub sellest, mis meetmed missuguse omadusega puhkudel kasutusel
     on - mitu küsimust järjest
156: SELECT (@OK) (0 Kysi OR Kes OR Mis) (-1* Kysi OR Kes OR Mis LINK NOT
     1* Verbivorm);
157: #neid on päris mitu tehtud
158: SELECT (@OK) (0 ("mitu+0") OR ("kaua+0")) (-1 (_A_));# See on nüüd
     vist korpuse ainsa lause peal tehtud
159: #ei suuda mitut asja korraga teha;  #1 (_S_) OR (_A_) OR (dem)
160: SELECT (@OK) (0 (_P_ inter rel indef)) (1 (_S_) OR (_A_) OR (dem))
     (NOT 1* (\#FinV) BARRIER Eraldaja OR Ja OR (_V_ main inf));
161: #olen müünud ka mitmeid teisi masinaid
162: SELECT (@OK) (0 (_P_ inter rel indef)) (0 (NN>)) (NOT 1* (\#FinV)
     BARRIER Eraldaja OR Ja OR (_V_ main inf)); #seda tuleks edasi mõelda
163: #ma ei tea mis asi, ma ei tea kus kohas - seal ees ei ole koma,
     käitub põhimõtteliselt omadussõnana. Seda siis, kui lisaverb on
     olemas. Võib olla ka Ma ei tea, mis see on.
164: #Loete siin _ma ei tea mida_ kokku
```

```
165: SELECT (@OK) (0 Kysi OR Kes OR Mis) (-3 ("mina+0")) (-2 ("ei+0")) (-
     1 ("tead+0")) (NOT 1* Verbivorm BARRIER Eraldaja OR Ja);
166: SELECT (@ERR) (0 Kysi OR Kes OR Mis) (-1 (_S_)); #jawaga, millel
167: #Küsisõnad: kui relatiivlause alguses, siis ikka komaga
168: SELECT (@ERR) (0 Kysi OR Mis OR Kes) (1* Verbivorm BARRIER Eraldaja
     OR Ja) (NOT -1 Eraldaja);
169: SELECT (@OK) (0 Kysi OR Mis OR Kes) (1* Verbivorm BARRIER Eraldaja
     OR Ja) (-1 Eraldaja);
170: #, siis miks mitte
171: SELECT (@OK) (0 Kysi OR Mis OR Kes) (-1 (_J_)) (-2 (Com));
172: SELECT (@OK) (0 Kysi OR Mis OR Kes) (-1 (Com)); #Lõpus üldine
     reegel: Kui koma on ees, on õige
173: SELECT (@ERR) (0 Kysi OR Mis OR Kes); #Kui viga pole, on korras
174:
175: #----------------------------------------
176: #  Verbivormid
177: #----------------------------------------
178: #Kui otse vasakul on 'ei' ja verb ise ei ole eitav
179: #aga ei, pidin minema
180: REMOVE (@OK) (0 (\#FinV)) (-1 ("ei+0")) (NOT 0 (neg));
181: #Kui otse vasakul on infinitiiv - nende kahe vahel peaks olema koma
182: #selleks et paika panna, pidin;
183: REMOVE (@OK) (0 (\#FinV)) (-1 (_V_ main inf)) (-1 (\#NGP) OR (\#NGP-
     P)) (NOT -1 (OBJ));
184: #millega ainult automaadiga sõita tohib:
185: REMOVE (@ERR) (0 (_V_ aux) OR (_V_ mod)) (-1 Verb) (NOT 1* Verb
     BARRIER Eraldaja OR Ja);
186: #,kui mina seal möödasõitu teha jõuan
187: REMOVE (@ERR) (0 Verb) (0 (\#Inf) OR (\#InfP)) (-1 Verb) (NOT 1*
     Verb BARRIER Eraldaja OR Ja);
188:
189: #----
190: #Lauselühendid
191: #
192: #Alati eraldatakse komaga järeltäiendina esinevad v-, tav-, nud-,
     tud- ja mata-lühendid.
193: #Põhimõtteliselt   ei   esine   korpuses   eriti   verbivormidest
     järeltäiendeid
194: #OR (sup ps abes) - kuidas mata-vormil järeltäiendi märgend välja
     näeb?
195: REMOVE (@OK) (0 Verb) (-1* (<VN) BARRIER Eraldaja OR Ja);
196: #Alati eraldatakse komaga määruslikud nud- ja tud-lühendid. E.g.
     söönud kõhu täis, heitsin magama.
197: REMOVE (@OK) (0 Verb) (-1* (_V_ partic ADVL) BARRIER Eraldaja OR
     Ja);
198: #Kunagi ei eraldata komaga  da-, ma-, mas-, mast-, vat- ja nuna-
     lühendit.
199: REMOVE (@OK) (0 Verb) (1 (Com)) (2* (<INF_N) BARRIER Et OR Ja OR Kui
     OR Kysi OR Mis OR Kes);
200:
201: #Kui vasakul samas osalauses on ka pöördeline verbivorm
202: REMOVE (@ERR) (0 Verb) (NOT -1* Verb BARRIER Eraldaja OR Ja);
203: SELECT (@ERR) (0C Verb) (-1* Verb BARRIER Eraldaja OR Ja);
204: #da-infinitiiv tekitab probleeme
205: REMOVE (@OK) (0 Verb) (-1* Verb BARRIER Eraldaja OR Ja);
206: SELECT (@OK) (0C Verb);
207:
208: END
```

Lisa 4. Komavigade tuvastaja sisend

```
"<$LA$>"
    "####" **CLB ???
"<Kui>"
    "kui+0" // _J_ crd sub #cap // **CLB  J
"<inglane>"
    "inglane+0" // _S_ com sg nom  //  SUBJ
"<ükskord>"
    "üks_kord+0" // _D_ //  ADVL
"<väljub>"
    "välju+b" // _V_ main indic pres ps3 sg ps af  #FinV #Intr //
+FMV
"<küsib>"
    "küsi+b" // _V_ main indic pres ps3 sg ps af  #FinV #NGP-P #abl //
+FMV
"<ameeriklane>"
    "ameeriklane+0" // _S_ com sg nom  //  ADVL
"<$:>"
    ":" // _Z_ Col //
"<$">"
    """ // _Z_ Quo //  **CLB
"<Miks>"
    "miks+0" // _D_ #cap // **CLB  ADVL
"<sul>"
    "sina+l" // _P_ pers ps2 sg ad  //  ADVL
"<nii>"
    "nii+0" // _D_ //  ADVL
"<kaua>"
    "kaua+0" // _D_ //  ADVL
"<läks>"
    "mine+s" // _V_ main indic impf ps3 sg ps af  #FinV #Intr #ill
#all //  +FMV
"<$?>"
    "?" // _Z_ Int  //
"<$LL$>"
    "####" **CLB-C ???
"<$LA$>"
    "####" ???
"<$">"
    """ // _Z_ Quo //
"<Kuna>"
    "kuna+0" // _D_ #cap //  **CLB-C ADVL
    "kuna+0" // _J_ sub #cap //  **CLB-C J
"<kaelkirjakut>"
    "kael_kirjak+t" // _S_ com sg part  //  SUBJ OBJ
"<ei>"
    "ei+0" // _V_ aux neg  //  NEG
"<ole>"
    "ole+0" // _V_ main indic pres ps neg  #FinV #Intr //  +FMV
"<siis>"
    "siis+0" // _D_ //  ADVL
    "siis+0" // _J_ //  J
"<läheb>"
    "mine+b" // _V_ main indic pres ps3 sg ps af  #FinV #Intr #ill
#all //  +FMV
"<mees>"
    "mees+0" // _S_ com sg nom  //  SUBJ
    "mesi+s" // _S_ com sg in  //  ADVL NN>
```

```
"<lehmaga>"
    "lehm+ga" // _S_ com sg kom  //  ADVL NN> <NN
"<naabri>"
    "naaber+0" // _S_ com sg gen  //  P>
"<juurde>"
    "juurde+0" // _K_ post #gen //  ADVL
"<ja>"
    "ja+0" // _J_ crd // **CLB  J
"<naaber>"
    "naaber+0" // _S_ com sg nom  //  SUBJ
"<küsib>"
    "küsi+b" // _V_ main indic pres ps3 sg ps af  #FinV #NGP-P #abl //
+FMV
"<kuhu>"
    "kuhu+0" // _D_ //  ADVL
"<sa>"
    "sina+0" // _P_ pers ps2 sg nom  //  ADVL
"<selle>"
    "see+0" // _P_ dem sg gen  // **CLB-C  NN>
"<seaga>"
    "siga+ga" // _A_ pos sg kom  //  ADVL
    "siga+ga" // _S_ com sg kom  //  ADVL NN>
"<tuled>"
    "tuli+d" // _S_ com pl nom  //  SUBJ OBJ
    "tule+d" // _V_ main indic pres ps2 sg ps af  #FinV #Intr //  +FMV
"<$?>"
    "?" // _Z_ Int  //
"<$LL$>"
    "####" ???
```

Lisa 5. Komavigade tuvastaja väljund

```
"<$LA$>"
	"####" **CLB ???

"<Kui>"
	"kui+0" // _J_ crd sub #cap **CLB J @OK
"<inglane>"
	"inglane+0" // _S_ com sg nom SUBJ
"<ükskord>"
	"üks_kord+0" // _D_ ADVL
"<väljub>"
	"välju+b" // _V_ main indic pres ps3 sg ps af #FinV #Intr +FMV @OK
"<küsib>"
	"küsi+b" // _V_ main indic pres ps3 sg ps af #FinV #NGP-P #abl +FMV @ERR
"<ameeriklane>"
	"ameeriklane+0" // _S_ com sg nom ADVL
"<$:>"
	":" // _Z_ Col
"<$">"
	"""" // _Z_ Quo **CLB
"<Miks>"
	"miks+0" // _D_ #cap **CLB ADVL @OK
"<sul>"
	"sina+l" // _P_ pers ps2 sg ad ADVL
"<nii>"
	"nii+0" // _D_ ADVL
"<kaua>"
	"kaua+0" // _D_ ADVL
"<läks>"
	"mine+s" // _V_ main indic impf ps3 sg ps af #FinV #Intr #ill #all +FMV @OK
"<$?>"
	"?" // _Z_ Int
"<$LL$>"
	"####" **CLB-C ???

"<$LA$>"
	"####" ???

"<$">"
	"""" // _Z_ Quo
"<Kuna>"
	"kuna+0" // _D_ #cap **CLB-C ADVL @OK
	"kuna+0" // _J_ sub #cap **CLB-C J @OK
"<kaelkirjakut>"
	"kael_kirjak+t" // _S_ com sg part SUBJ OBJ
"<ei>"
	"ei+0" // _V_ aux neg NEG
"<ole>"
	"ole+0" // _V_ main indic pres ps neg #FinV #Intr +FMV @OK
"<siis>"
	"siis+0" // _D_ ADVL @ERR
	"siis+0" // _J_ J @ERR
"<läheb>"
	"mine+b" // _V_ main indic pres ps3 sg ps af #FinV #Intr #ill #all +FMV @ERR
"<mees>"
	"mees+0" // _S_ com sg nom SUBJ
	"mesi+s" // _S_ com sg in ADVL NN>
"<lehmaga>"
```

```
        "lehm+ga" // _S_ com sg kom ADVL NN> <NN
"<naabri>"
        "naaber+0" // _S_ com sg gen P>
"<juurde>"
        "juurde+0" // _K_ post #gen ADVL
"<ja>"
        "ja+0" // _J_ crd **CLB J @OK
"<naaber>"
        "naaber+0" // _S_ com sg nom SUBJ
"<küsib>"
        "küsi+b" // _V_ main indic pres ps3 sg ps af #FinV #NGP-P #abl +FMV @OK
"<kuhu>"
        "kuhu+0" // _D_ ADVL @ERR
"<sa>"
        "sina+0" // _P_ pers ps2 sg nom ADVL
"<selle>"
        "see+0" // _P_ dem sg gen **CLB-C NN>
"<seaga>"
        "siga+ga" // _A_ pos sg kom ADVL
        "siga+ga" // _S_ com sg kom ADVL NN>
"<tuled>"
        "tuli+d" // _S_ com pl nom SUBJ OBJ
        "tule+d" // _V_ main indic pres ps2 sg ps af #FinV #Intr +FMV @ERR
"<$?>"
        "?" // _Z_ Int
"<$LL$>"
        "####" ???
```