T A R T U Ü L I K O O L
MATEMAATIKA-INFORMAATIKATEADUSKOND
Arvutiteaduse instituut
Keeletehnoloogia õppetool
Informaatika eriala

Emilia Käsper

Eesti keele lausete automaatne genereerimine

Bakalaureusetöö

Juhendaja: lektor K. Müürisep

Autor: ..... "....." mai 2004
Juhendaja: ..... "....." mai 2004
Õppetooli juhataja: ..... "....." mai 2004

TARTU 2004

Sisukord

Sissejuhatus

Loomuliku keele genereerimise vallas on Eestis siiani tegeldud peamiselt morfoloogiaga. Süntaksi jaoks on olemas vaid analüüsimiseks sobiv grammatika [Roosmaa jt, 2001]; genereerimist see formalism ei võimalda. Samas on süntaksi genereerimine vältimatu etapp mitmete rakenduste puhul, masintõlkest dialoogisüsteemideni. Käesolev bakalaureusetöö tutvustab süntaksi genereerimise probleemi üldiselt ning analüüsib lähemalt probleemipüstitust eesti keele korral. Töö on esimene samm eesti keele süntaksigeneraatori valmimisel.

Bakalaureusetöö koosneb viiest peatükist. Esimene peatükk tutvustab loomuliku keele genereerimise ülesannet ning annab ülevaate genereerimise rakendustest. Kirjeldatakse loomuliku keele generaatori standardarhitektuuri ning uuritakse keeletehnoloogias möödapääsmatut ressursside taaskasutuse küsimust. Teine peatükk on pühendatud süntaksi genereerimisele. Selgitatakse genereerimise lähteülesannet ning analüüsitakse probleemipüstitust eesti keele korral.

Kolmandas peatükis kirjeldatakse laialtlevinud mitmekeelset süntaksigeneraatorit: funktsionaalse unifitseerimise formalismil põhinevat paketti FUF ning sellega ühilduvat inglise keele grammatikat SURGE. Analüüsitakse võimalusi paketi kohandamiseks eesti keelele. Neljas peatükk annab võrdluseks lühikese ülevaate teistest levinumatest süntaksigeneraatoritest.

Viiendas peatükis vaadeldakse põhjalikumalt SURGE nimisõnafraasi käsitlust ja grammatikat. Selgitatakse, milliseid muudatused on eesti keele grammatikas vajalikud, ning koostatakse SURGE põhjal fragment eesti keele

nimisõnagrammatikast.

Lisades 1–5 on esitatud paketi FUF kasutajasessiooni täielik kommenteeritud logi, koostatud eesti keele grammatika ning grammatikaga unifitseerimise näited.

Peatükk 1

Loomuliku keele genereerimine

1.1 Mis on loomuliku keele genereerimine?

Stephan Busemann [Busemann, 1999] esitab McDonaldi definitsiooni, mille kohaselt

> Loomuliku keele genereerimine on protsess, mille käigus eesmärgipäraselt konstrueeritakse spetsiifilistele kommunikatiivsetele eesmärkidele vastav loomuliku keele tekst.

Loomuliku keele genereerimise all võime mõista ka kõne genereerimist, kuid käesolevas töös piirdume kirjaliku teksti genereerimise vaatlemisega. Niisiis, loomuliku keele generaator on arvutitarkvara, mille sisend on mittelingvistiliselt esitatud informatsioon ja väljund on inimese poolt kõneldava keele ehk loomuliku keele tekst. Teksti vormistuslikud aspektid nagu küljenduse, formaatimise ja illustreerivad graafikud jätame edaspidi vaatluse alt välja ning loeme, et generaatori väljund on inimese poolt mõistetav loomuliku keele sõne.

Loomuliku keele genereerimise tarkvara ei ole väärtuslik eraldiseisvana, kuid on mitmete keeletehnoloogia rakenduste lahutamatu osa. Loomuliku keele genereerimist kasutavad süsteemid võib rakenduse järgi jaotada kolme gruppi:

- **Dialoogisüsteemid** ehk tarkvara, mis suhtleb kasutajaga loomulikus

keeles. Sellised on näiteks reisiinfosüsteemid, intelligentsed otsimootorid ja vestluspartnerid (ingl k *chatterbot*). Üheks esimeseks ja ilmselt tuntuimaks katseks imiteerida inimestevahelist dialoogi oli Joseph Weizenbaumi 1960. aastatel loodud "psühhoanalüütik" ELIZA.

- **Teksti genereerimine** selliste rakenduste jaoks nagu automaatne sisukokkuvõtete tegemine, aruannete genereerimine jms.

- **Masintõlge**. Kui masintõlge ei põhine just puhtalt statistilisel lähenemisel või lihtsal lähtekeele sõnadele sihtkeele sõnade vastavusse seadmisel, on genereeriv komponent tõlkesüsteemi lahutamatu osa. Üks tuntumaid ja suuremaid masintõlke projekte on saksa, inglise ja jaapani keele kõnetõlkija Verbmobil [Verbmobil], mille tekstigeneraatori ehitust tutvustame põgusalt alapeatükis 4.2.

1.2 Mallipõhine vs täielik genereerimine

Loomuliku keele automaatses genereerimises eristatakse mallipõhist (ingl k *template-based*) ja täielikku (ingl k *full text*) genereerimist. Mallipõhine ehk madal (ingl k *shallow*) genereerimine seisneb sõnade sobitamises etteantud raamistikku (lihtsustatult on tegu lünkade täitmisega), samas kui täielik ehk sügav (ingl k *deep*) genereerimine toetub lingvistilisele formalismile (grammatikale) ja eeldab seega genereeritava keele ulatuslikku formaalset kirjeldamist.

Mallipõhise genereerimise standardrakendused on kitsa valdkonna ülesanded. Klassikaline näide on ilmateadete automaatne genereerimine ja/või tõlkimine; reaalsetest süsteemidest võib levinuma näitena tuua raamistiku Exemplars [Exemplars], mille baasil on valminud järgmised rakendused:

- reisiinfosüsteem AQUAINT;

- objekt-orienteeritud andmemudelite diagrammide kirjeldamise süsteem ModelExplainer;

- majandusstatistika kokkuvõtete tegemise süsteem LFS;

- ilmateadete genereerimise süsteem MeteoCogent;
- jne.

Exemplars on uue põlvkonna mallipõhine süsteem, mis võimaldab rakenduste arendajatel täiustada malle keerulisemate lingvistiliste mudelitega, kuid ometi näeme, et reaalsed rakendused on piiritletud väga kitsas valdkonnas. Viimasel ajal on mallipõhisele genereerimisele hakatud uuesti suuremat tähelepanu pöörama ning on eksperimenteeritud komplekssemate mallipõhiste süsteemidega. Artiklis [Stenzhorn, 2002] kirjeldatakse XML- ja Java-põhist süsteemi XtraGen, mille mallipõhine lähenemine toetub väitele, et reaalse maailma rakendused vajavad haruharva grammatika täielikku katmist. *De facto* peetakse mallipõhist lähenemist siiski, võrreldes täieliku genereerimisega, valdkonnaspetsiifiliseks ning seega sobimatuks korduvkasutatavate üldiste (doomenikitsenduseta) süsteemide jaoks. Analoogiliselt statistilise masintõlkega võib mallipõhisele lähenemisele suuremat edu prognoosida fikseeritud sõnajärje ja piiratud morfoloogiaga keelte puhul. Et käesoleva töö lõppeesmärgiks on seatud üldise eesti keele süntaksigeneraatori valmimine, jätame mallipõhise genereerimise edaspidi vaatluse alt välja ning analüüsime vaid täieliku genereerimise meetodeid ja süsteeme.

1.3 Loomuliku keele generaatori arhitektuur

Ehud Reiter ja Robert Dale [Reiter jt, 2000] esitavad loomuliku keele generaatori konveiersüsteemina, kus iga eelmise etapi sisend on järgmise etapi väljundiks. Selline esitus on kooskõlas suure osa reaalsete loomuliku keele genereerimise süsteemide arhitektuuriga. Konveieri selgroo moodustavad kolm gruppi:

I sisuplaneerimine (ingl k *text planning*);

II mikroplaneerimine (ingl k *microplanning*) ja

III pindrealisatsioon (ingl k *surface realization*, ka *tactical generation*).

Selline jaotus on ülesande püstituse seisukohalt loomulik. Konveieri I etapp vastab küsimusele "Mida öelda?": lihtsustatult võib öelda, et tegu on

semantilise infoga. Teine etapp vastab küsimusele "Kuidas öelda?", rikastades semantilist infot süntaktilisega. Lause tasandil tähendab see, et lisanduvad süntaktilised kategooriad nagu kõne liik, aeg, fookus jms. Seotud teksti tasandil võib mikroplaneerija otsustada näiteks asesõnade kasutamise üle. Viimane, III etapp, nagu nimigi ütleb, on realisaator, mille väljund on loomulik keel. Pindrealisaatoris võib omakorda eristada süntaktilist ja morfoloogilist komponenti; meie erilises huviorbiidis on neist esimene.

1.4 Ressursside korduvkasutamine. Mitmekeelsed süsteemid

Ressursside taaskasutus on kõigis keeletehnoloogilistes rakendustes oluline nõue. Taaskasutuses võib eristada kaht suunda:

- **Taaskasutus ühe keele piires** omandab suurima tähtsuse just väiksemate ja vähemuuritute keelte jaoks nagu eesti keel. Kui inglise keele jaoks on valminud mitmeid nii morfoloogilisi kui süntaktilisi, analüüsivaid kui genereerivaid komponente, siis eesti keele jaoks on ressursside hulk piiratud ja iga uus tööriist peaks kasutust leidma võimalikult paljudes rakendustes. Käesolev töö juhindub sellest põhimõttest ja on eesmärgiks seadnud korduvkasutatava genereerimiskomponendi loomise.

- **Mitmekeelne taaskasutus** osutub võimalikuks keelest sõltumatute komponentide puhul. Näiteks masintõlkes esindab sellist lähenemist keelest sõltumatu *interlingua* mooduli loomine tähenduse esitamiseks.

Loomuliku keele generaatori konveieri modulaarne ülesehitus võimaldab ressursside taaskasutamist nii ühe keele piires kui mitmekeelsetes rakendustes. Generaatori sisuspetsiifiline töötlus toimub I (ja II) etapis ning keelespetsiifilise informatsiooni töötlus III etapi pindrealisaatoris või ka osalt mikroplaneerimise viimases etapis. Seega on sisuplaneerija taaskasutusvaldkonnaks sisult sarnased (võimalik, et mitmekeelsed) rakendused, samas kui pindrealisaatorit kui keelt töötlevat komponenti saab korduvkasutada ühe keele erinevates rakendustes. Täpsemini, pindrealisaatori tuum võib olla kasutatav

ka erinevates keeltes, nagu näeme peatükis 3, kuid see eeldab sama formalismi kasutamist grammatikate esitamiseks. Pindrealisaatori mitmekeelne korduvkasutus on seetõttu seda efektiivsem ja otstarbekam, mida sarnasemad on vaatlusalused keeled ehk mida paremini on formalism nendele keeltele kohaldatav.

Korduvkasutuse printsiibi illustreerimiseks vaatleme näitena fiktiivset ilmateadete genereerimise süsteemi.

I Sisuplaneerija saab info andmebaasist ja väljastab semantilise tekstispetsifikatsiooni. Sisuplaneerija näitlik sisend ja väljund on kujutatud joonisel 1.1. Siinjuures tuleb tähele panna, et väljund ei ole leksikali-

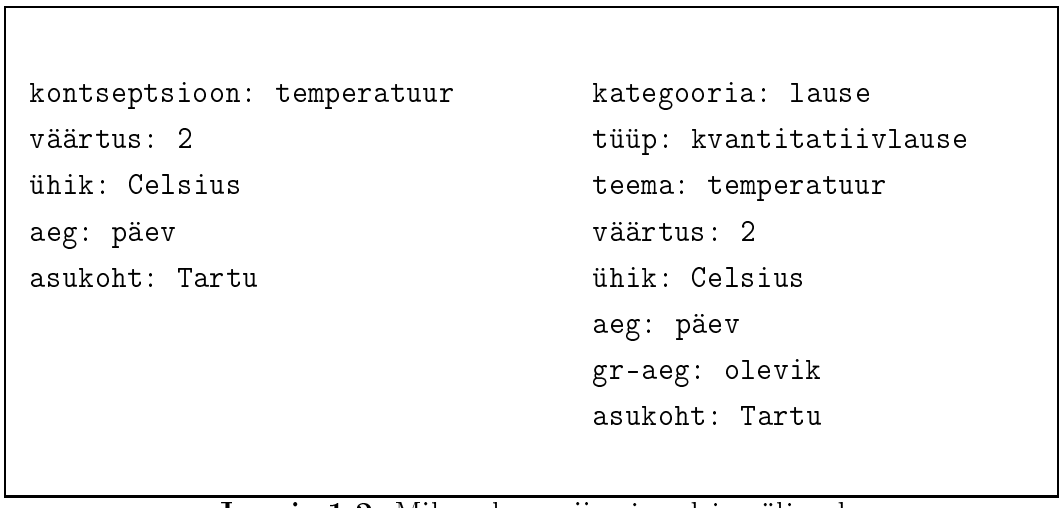
Linn	Öösel ($^\circ C$)	Päeval ($^\circ C$)
Pärnu	0	5
Tallinn	−2	2
Tartu	−4	2

```
kontseptsioon: temperatuur
väärtus: 2
ühik: Celsius
aeg: päev
asukoht: Tartu
```

Joonis 1.1: Sisuplaneerija sisend ja väljund

seeritud, st näiteks semantilisele esitusele
`kontseptsioon: temperatuur`
vastab igas keeles erinev leksikaalne esitus — inglise keeles lekseem "temperature", eesti keeles "temperatuur" jne.

IIa Mikroplaneerija lisab semantilisele informatsioonile süntaktilise. Süntaktilise informatsiooniga rikastatud väljund on kujutatud joonisel 1.2. Mikroplaneerija otsustab kontseptsiooni "temperatuur" järgi, et tegu on kvantitatiivlausega. Lisandunud on ka grammatiline märgend, et lause aeg on olevik. Viimane otsustus võidakse täpsema informatsiooni puudumisel teha vaikimisi.



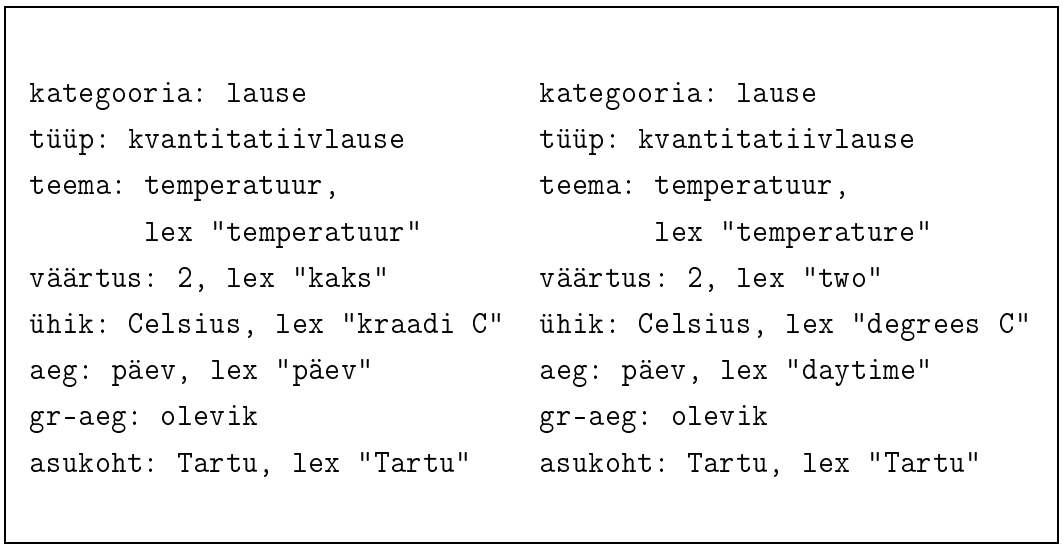
Joonis 1.2: Mikroplaneerija sisend ja väljund

Paneme tähele, et siiani on kõik etapid olnud keelest sõltumatud. Semantilise informatsiooni esitamiseks on küll kasutatud eesti keelt, kuid seda vaid arusaadavuse eesmärgil. Samaväärselt võiks lauseplaan olla esitatud mistahes kokkuleppeliste märgendite abil: lause eesti keeles esitamiseks on vaja vaid leksikoni, mis märgenditele seab vastavusse eestikeelsed lekseemid. Seda etappi vaatlemegi järgmises alapunktis.

IIb Mikroplaneerimise viimases etapis toimub leksikaliseerimine ehk lihtsamalt sõnade valik. Joonisel 1.3 on ära toodud leksikaliseerimise väljund eesti ja inglise keele jaoks. Leksikaliseerimine on esimene keelest sõltuv etapp, kus semantilisele informatsioonile seatakse leksikoni abil vastavusse konkreetse keele lekseemid.

III Pindrealisaator väljastab sisendina saadud semantilise, leksikaalse ja süntaktilise informatsiooni põhjal loomuliku keele lause. Pindrealisatsiooni tulemusena oleksid antud juhul aktsepteeritavad laused
Tartus on temperatuur päeval kaks kraadi C.
ja
Daytime temperature in Tartu is two degrees C.

Niisiis on näites toodud konveieri kaks esimest etappi täielikult genereeritavast keelest sõltumatud. Kolmas etapp — leksikaliseerimine — on keelespetsiifiline. Viimases, pindrealiseerimise etapis, on iga keele jaoks vajalik



Joonis 1.3: Leksikaliseerimise väljund inglise ja eesti keele jaoks

omaette süntaktiline (grammatika) ja morfoloogiline komponent. Korduvalt on võimalik kasutada realisaatori tuuma, mis vastavalt sisendile, grammatikale ja morfoloogilisele komponendile realisatsiooni teostab.

Peatükk 2

Eesti keele süntaksigeneraator: probleemipüstitus

2.1 Korduvkasutatav pindrealisaator

Käesoleva projekti eesmärk on luua eesti keele jaoks korduvkasutatav doomenikitsenduseta pindrealisaator. Millised ülesanded peab selline pindrealisaator lahendama? Michael Elhadad ja Jacques Robin esitavad artiklis [Elhadad jt] ammendava loetelu pindrealisaatori ülesannetest, mis väärib siinkohal äratoomist ja kommenteerimist:

1. **Sisendis antud lausestruktuurile vastavate süntaktiliste rollide määramine**. Ehkki pindrealisaatori sisend võib olla esitatud väga erinevalt (vt alapunkti 2.2), on küllalt levinud seisukoht, et süntaktiline informatsioon tuleb lisada alles pindrealisatsiooni käigus ja pindrealisaatori sisendit peab klientprogramm või kasutaja suutma kirjeldada võimalikult väikese süntaktilise teadmusega.

2. **Süntaktiline parafraseerimine ja varieerimine**. Täiuslikum süsteem peab suutma arvestada rõhuasetusi, näiteks suutma genereerida nii laused "Raamat on laual." (neutraalne) kui "Laual on raamat." (fookus sõnal "raamat"). Ilmselt on sõnade järjekorraga varieerimine oluline, sest vaid teine lause on korrektne vastus küsimusele "Mis on laual?".

3. **Ülegenereerimise välistamine**. Seda punkti võib vaadelda eelmise

punkti kitsendusena. Näiteks, asendades eelmises näites nimisõna "raamat" asesõnaga "see", on korrektne ainult lause "See on laual.", aga mitte "Laual on see.".

4. **Vaikeväärtused**. Pindrealisaator peab suutma anda väljundi ka minimaalselt defineeritud sisendi korral. Vaikimisi grammatiline aeg võib olla näiteks (liht)olevik; vaikimisi sõnade järjekord on neutraalne jne.

5. **Ühilduvuse jälgimine**. Vaatleme taaskord näitelauset. Oletame, et sisendis on öeldud, et "raamat" peab olema ainsuses, kuid verbi kohta pole midagi öeldud. Sel juhul peab realisaator automaatselt otsustama, et verb "olema" peab lauses olema ainsuse kolmandas pöördes.

6. **Asesõnade, abiverbide jms valik**. Pindrealisaatori sisend ei pea sisaldama nn "suletud klassi" sõnu. Näiteks juhul, kui tahame küsida "Kas raamat on laual?" ei pea sõna "kas" sisalduma realisaatori sisendis; küll peab sisend kätkema informatsiooni selle kohta, et soovitakse moodustada "jah-ei"-küsimust.

7. **Lineaarne järjestus**. Pindrealisaator peab määrama sõnade (täpsemini süntaktiliste moodustajate) järjekorra. Näitelause "Raamat on laual." korral on lauseliikmete järjekord `alus < öeldis < määrus`.

8. **Muutelõppude määramine**. Pindrealisaator peab sisaldama muutelõppude valikut ja morfoloogilist sünteesi (näitelauses "raamat — raamat", "olema — on", "laud — laual").

9. **Sõne väljastamine**. Pindrealisaatori väljund on sõne (lause), kus sõnad on õigete muutelõppudega õiges (soovitud) järjekorras.

Viimast punkti tuleb kindlasti täiendada veel kirjavahemärkide ja lause alguse suurtähtede lisamisega. Sõltuvalt ülesandest võib pindrealisaator täita veel mingeid ülesandeid (näiteks teksti küljendamine, väljund html- vms formaadis jne), kuid korduvkasutatava üldise pindrealisaatori väljundiks sobib kõige paremini tavaline tekst.

Ehkki loetelus esitatud nõudmised on püstitatud inglise keele süntaksist ja morfoloogiast lähtuvalt, on need laiendatavad teistele keeltele. Ülesanded varieeruvad ühe ja sama lause genereerimisel erinevates keeltes, kuid loetletud ülesannetegrupp on kinnine. Ülal vaadeldud lause "Raamat on laual." korral peab eesti keele generaator määrama määruse "laual" muutelõpu, samas kui ingliskeelse lause "The book is on the table." korral on generaatori ülesanne valida korrektne eessõna "on". Sisuliselt on tegu analoogsete otsustustega, kus erinevus on vaid väljundis: ühel juhul on väljundiks kääne, teisel juhul eessõna. Eesti keele komplekssе morfoloogiaga seotud probleemid kompenseerib inglise keeles mõneti eessõnade valik.

Pindrealisaatorit võib omakorda vaadelda konveierina, kus on eristatavad süntaktiline ja morfoloogiline komponent: süntaksigeneraator ja morfoloogiline süntesaator. Neist viimane on vastutav vaid sõnade muutelõppude sünteesimise eest ning kasutab selleks leksikoni ja süntaksigeneraatorilt saadud sisendinfot. See tähendab, et muutelõpud *määrab* süntaksigeneraator. Käesolevas peatükis asume vaatlema võimalusi eesti keele süntaksigeneraatori valmimiseks. Sealjuures loeme süntaksigeneraatori ülesandeks ka sõnade järjekorraga manipuleerimise. Kuna eesti keele jaoks on juba valminud morfoloogiline süntesaator [Filosoft, EKI], siis valmiks nende kahe tööriista kombineerimisel eesti keele korduvkasutatav pindrealisaator tulevaste keeletehnoloogiaprojektide jaoks.

Eesti keele süntaksigeneraatori probleemipüstituses võib eristada kolme etappi:

I süntaksigeneraatori sisendi kirjeldamine;

II eesti keele formaalse genereeriva grammatika väljatöötamine ja

III süntaksigeneraatori tuuma valimine või väljatöötamine.

Need etapid ei ole siiski teineteisest sõltumatud. Juhul, kui osutub otstarbekaks mõne eksisteeriva mitmekeelse tuuma taaskasutamine, dikteerib realisaatori ülesehitus sisendi ja seab kitsendused grammatikale.

2.2 Süntaksigeneraatori sisend

Konveierarhitektuur on genereerimisel laialt kasutatav, kuid paradoksaalsel kombel eksisteerib üldtunnustatav kokkulepe vaid selles osas, milline on viimase etapi väljund. Reiteri ja Dale'i [Reiter jt, 2000] järgi koosneb loomuliku keele generaatori sisend nelikust $< k,\ c,\ u,\ d >$, kus k on infoallikas, c on kommunikatiivne eesmärk, u on kasutajamudel ja d on diskursuse ajalugu. Kuna k on ilmselgelt sõltuv konkreetsest rakendusest, on raske või isegi võimatu loomuliku keele generaatori sisendit täpsemalt määratleda. Näiteks masintõlke korral annab lähtekeel küllalt täpselt formuleeritud sisendi, mis sisaldab nii semantilist kui süntaktilist infot; ilmateadete genereerimise korral on sisendinfo abstraktset semantilist laadi, näiteks semantiline raam (ingl k *semantic frame*).

Nii nagu ei ole fikseeritud terve süsteemi sisendi abstraktsioonitase, varieeruvad reaalsetes rakendustes ka konveieri etappide piirid. Meid huvitab käesoleva töö raames, milline võib antud arhitektuuris olla mikroplaneerija väljund ja seega pindrealisaatori sisend. Nagu eelpool mainitud, ei pruugi loomuliku keele genereerimise üksikülesanne piirduda vaid ühe lausega, vaid võib sisaldada ka sidusa teksti planeerimist ja realiseerimist. See planeerimine tehakse aga üldiselt esimeses kahes etapis, nii et pindrealisatsioon toimub lause-haaval: pindrealisaatori sisend on lauseplaan (ingl k *sentence plan*) ning väljund loomuliku keele lause.

Alapeatükis 2.1 esitasime seisukoha, et pindrealisaatori sisend peab olema kirjeldatud kõrgema taseme vahenditega kui süntaktilised kategooriad. See seisukoht ei ole absoluutne: edukat kasutamist on leidnud ka süsteemid, mille sisend on süntaktiline süvastruktuur, näiteks inglise keele pindrealisaator RealPro [RealPro]. RealPro sisend DSyntS, mida on kirjeldatud artiklis [Lavoie jt, 1997], on esitatud süntaktiliselt, kasutades termineid nagu "alus", "sihitis" jm. Semantilise lähenemise korral süntaksitermineid välditakse, kasutades kontseptuaalseid suhteid nagu "agent", "protsess" vm.

Eesti keele süntaksigeneraatori korral on tõenäoliselt siiski otstarbekas

kasutada kõrgemat esitust. Esiteks võib süntaksi märgendite kasutamine osutuda takistavaks asjaoluks mitmekeelsete rakenduste loomise korral — kõrgemalt keelest sõltumatult esituselt süntaktilisele süvastruktuurile üleminekuks tuleb ikkagi luua lisamoodul. Teiseks, mida lihtsam on süntaksigeneraator, seda suurema töö peab igal üksikul juhul ära tegema klientrakenduse mikroplaneerija — eesti keele puhul on tõenäoliselt mõistlik see töö koondada juba pindrealisaatorisse. Kokkuvõttes, kuna loomuliku keele generaatori arhitektuur on modulaarne, taandub sisendi valiku küsimus sellele, kuhu tõmmata piir ehk kui suure osa tööst on pindrealisaator valmis enda kanda võtma.

2.3 Süntaksigeneraatori grammatika

Raamatus [Roosmaa jt, 2001] on kirjeldatud eesti keele formaalne grammatika, mis on kasutusel eesti keele süntaksianalüsaatoris [Müürisep]. Eesti keele jaoks valminud kitsenduste grammatika on formalism pindmiseks morfoloogial põhinevaks süntaktiliseks analüüsiks. Kitsenduste grammatikal põhinev analüüs seisneb analüüsi tulemuse järk-järgulises täpsustamises: analüüsi alguses lisatakse sõnale kõikvõimalikud analüüsivariandid, misjärel asutakse konteksti mittesobivaid eemaldama (kasutades selleks reegleid ehk nö kitsendusi). Kahjuks ei ole kitsenduste grammatika pööratav, mis tähendab, et genereeriva grammatika loomisel tuleb alustada algusest.

Eesti keele kui rikka morfoloogiaga keele korral on lekseemidele õigete muutelõppude määramine äärmiselt keeruline küsimus. Ilmselt on siin vaja mahukat rektsioonide leksikoni. Süntaksigeneraatori loomisel on valida kahe lahenduse vahel:

- sisendi rikastamine leksikonist saadud informatsiooniga toimub süntaksi genereerimise käigus või
- sisendi rikastamine leksikonist saadud informatsiooniga on mikroplaneerimise viimane samm.

Eesti keele süntaksigeneraatori ülesande lahendamisel võib osutuda otstarbekaks viimane variant: leksikaalse info kogumine genereerimisest eralda-

da, nagu seda on edukalt tehtud FUF-interpretaatoril baseeruva pindrealisaatori SURGE [Elhadad jt] korral . Selline lähenemine võimaldab ka üldise mahuka leksikoni puudumisel alustada piiratud doomeni rakenduste loomist väikese kuid rikka leksikoni baasil. Taaskord tuleb rõhutada, et generaator on modulaarne ja pole niivõrd oluline, millised sammud täpselt konkreetse etapi käigus lahendatakse, kui see, et iga lahendus oleks kooskõlas ülejäänutega ja tulevaste klientrakenduste vajadustega.

Peatükk 3

Genereerimine paketiga FUF/SURGE

3.1 Paketi FUF/SURGE üldiseloomustus

FUF [FUF-manual] on akronüüm fraasist "Functional Unification Formalism". Programmeerimiskeelel Common Lisp baseeruv keel on formalism, mida võib võrrelda PROLOG-i sisseehitatud loomuliku keele töötlemise võimalustega. Artiklis [Elhadad jt] tuuakse välja FUF-i kolm olulisemat eelist võrreldes PROLOG-iga:

- **Osalise teadmuse rakendamine**, mis saavutatakse sellega, et informatsiooni kodeerimine ei sõltu ei predikaadi argumentide arvust ega atribuutide järjekorrast;
- **Genereerimisele kohandatus** — FUF-is rakendatakse lingvistilistele moodustajatele ülalt alla laiuti rekursiooni;
- **Sisseehitatud lineariseerija ja morfoloogiline komponent**.

Eesti keele seisukohalt kaotab morfoloogiline komponent kui keelespetsiifiline rakendus tähtsuse. Põhjalikum analüüs formalismi FUF sobivusest eesti keele genereeriva grammatika koostamiseks on toodud alapeatükis 3.4.

Formalismi FUF rakendus on unifitseerimisinterpretaator, mille sisend on osaliselt spetsifitseeritud nn funktsionaalne kirjeldus (ingl k *functional description, FD*) ja väljund on täielikult spetsifitseeritud funktsionaalne kirjeldus. Funktsionaalne kirjeldus kujutab endast atribuutide ja nende väärtuste paare, näiteks FD

```
((cat noun) (proper yes))
```

kirjeldab pärisnime (kategooria "nimisõna", tunnuse "pärisnimi" väärtus positiivne).

Funktsionaalse kirjelduse spetsifitseerimine toimub konkreetsest grammatikast lähtuvalt, kuid protsess ise on keelest sõltumatu. Grammatika kirjutamisel on võimalik vabalt valida sisendi abstraktsiooni tase; võimalik on implementeerida nii lihtsaid fraasistruktuurireegleid kui unifitseerimisgrammatikaid. Järgmises alapeatükis toome näite interpretaatori sisendist ja väljundist ning selgitame lähemalt unifitseerimise iseloomu.

Pakett FUF sisaldab lisaks unifitseerimisinterpretaatorile ka lineariseerijat, mis kujutab endast morfoloogilist süntesaatorit koos sõnade õiges järjekorras väljastamisega. Lineariseerijat saab siiski muutmata kujul kasutada vaid inglise keele jaoks, kuna ta sisaldab keelespetsiifilist morfoloogilist komponenti.

SURGE [Elhadad jt] on inglise keele grammatika, mis on implementeeritud FUF formalismis. SURGE koondab funktsionaalse unifitseerimise raamistikku erinevaid lingvistilisi teooriaid ning on enamlevinumaid inglise keele genereerivaid grammatikaid. Samuti on SURGE-is kasutatud erinevate sisenditüüpide varieerimist. Põhiliselt aktsepteeritakse sisendina hierarhiliselt struktureeritud "protsesse", mis kirjeldavad lause või fraasi temaatilist struktuuri – protsessi, sündmust, seost, olekut. Seega on tegu semantilise/kontseptuaalse lähenemisega. Samas leiavad teatud erijuhtude käsitlemisel rakendust ka leksikaalsed teooriad, kus semantilistele rollidele on juba vastavusse seotud süntaktilised.

Nii interpretaator FUF kui grammatika SURGE on vabavarana saadaval veebiaadressilt [FUF/SURGE].

3.2 Lihtne inglise keele grammatika

Unifitseerimismehhanismi selgitamiseks vaatame lihtsat inglise keele näitegrammatikat, mis on kaasas paketi FUF installatsiooniga [FUF/SURGE]. Grammatika *gr*0 (vt joonist 3.1) sisaldab lause, lihtsa nimisõnafraasi ja verbifraasi kirjeldusi ning genereerib näiteks laused "John loves Mary" või "Mary likes the cat".

Fraasistruktuurireeglitega võib grammatika *gr*0 esitada nii, nagu näidatud joonisel 3.2. Funktsionaalse unifitseerimise formalismi atribuutide-väärtuste paarid aga pakuvad grammatika esitamiseks rikkalikumaid võimalusi.

- Tunnuste lisamine võimaldab süntaktilisi kategooriaid täpsemalt klassifitseerida. Näiteks rida 19

  ```
  19:      ((proper yes)
  ```

  ütleb, et tegu on pärisnimega (ingl k *proper name*).

- Unifitseerimine võimaldab jälgida grammatiliste tunnuste ühildumist. Tunnuse väärtus võib olla absoluutne — (proper yes) —, aga ka relatiivne. Vaatleme näiteks järgmisi ridu:

  ```
  10:   (verb ((cat vp)
  11:         (number {prot number})))
  ```

  Rida 10 ütleb, et temaatilisele rollile "verb" vastab süntaktiline kategooria "verbifraas". Rida 11 ütleb, et verbifraasi arv on sama mis rollile "prot" vastavusse seatud arv. Teisisõnu, öeldise arv peab ühilduma aluse arvuga. Näites 3.4 vaatleme, kuidas toimub unifitseerimine FUF interpretaatoris.

```
 1: ((alt top (
 2:  ;; Grammatikal on alati sama kuju:
 3:  ;; iga moodustajakategooria jaoks on üks haru.
 4:
 5:  ;; Esimene haru
 6:  ;; kirjeldab kategooriat S (lause).
 7:  ((cat s)
 8:   (prot ((cat np)))
 9:   (goal ((cat np)))
10:   (verb ((cat vp)
11:       (number {prot number})))
12:    (pattern (prot verb goal)))
13:
14:   ;; Teine haru: NP (nimisõnafraas).
15:   ((cat np)
16:    (n ((cat noun) (number {^ ^ number})))
17:    (alt (
18:      ;; Pärisnimed ei vaja artiklit.
19:      ((proper yes)
20:       (pattern (n)))
21:      ;; Tavalise nimisõna ette käib artikkel.
22:      ((proper no)
23:       (pattern (det n))
24:       (det ((cat article)
25:             (lex "the")))))))
26:
27:   ;; Kolmas haru: VP (verbifraas).
28:   ((cat vp)
29:    (pattern (v dots))
30:    (v ((cat verb))))
31:
32:   ;; Neljas haru: artikkel
33:   ;; ei tee midagi.
34:   ((cat article)))))))
```


Joonis 3.1: Lihtne inglise keele grammatika $gr0$

```
S  -> NP VP NP
NP -> N
NP -> DET N
VP -> V
```

Joonis 3.2: Grammatika $gr0$ struktuur esitatuna fraasistruktuurireeglite abil

- Hargnemine võimaldab mugavalt käsitleda erinevaid alternatiive ja seada **vaikeväärtusi**. Hargnemine real 17 jaotab nimisõnad kaheks: pärisnimed ja harilikud nimisõnad. Seejuures esimesel juhul artiklit nimisõna ette ei lisata. FUF interpretaator alustab läbimist esimesest harust. Kui etteantud funktsionaalses kirjelduses on määratud, et tegu on hariliku nimisõnaga — (proper no) —, siis unifitseerimine ebaõnnestub. Kui aga on määratud, et tegu on pärisnimega — (proper yes) — **või on nimisõnatüüp määramata**, siis unifikatsioon õnnestub. Seega on antud grammatikas nimisõna tüüp vaikimisi pärisnimi.

- Konstituentide järjekorra määramine koos hargnemisega võimaldab mugavalt määrata lause sõnade järjekorra. Rida 12

  ```
  12:    (pattern (prot verb goal)))
  ```

  ütleb, et lauseliikmed on järjekorras `prot < verb < goal`. Inglise keel on fikseeritud sõnajärjega keel: lausest "John loves Mary." lauseliikmete ümberpaigutamisel saadud lause "Mary loves John." omandab semantiliselt täiesti uue tähenduse. Seevastu eesti keeles on sõnajärg vaba: lausest "Jaan armastab Marit." ja "Marit armastab Jaan." on samatähenduslikud, ainult erineva rõhuasetusega. Seega võimaldab rõhuasetust väljendavate tunnuste lisamine kergesti opereerida sõnade järjekorraga, kasutades lausemustrit (pattern ...) ja hargnemist. Lähema näite toome peatükis 3.4, kus koostame lihtsa eesti keele grammatika.

Järgnevalt vaatame kaht näidet unifitseerimisest. Näited pärinevad reaalsest FUF-sessioonist. Interpretaatori käivitamise täielik logi on toodud Lisas 1.

```
((CAT S) (PROT ((N === JOHN))) (VERB ((V === LOVE))) (GOAL ((N === MARY))))

((CAT S :I) (PROT ((N (# # #)) (CAT NP :E) (PROPER YES :E) (PATTERN (N) :E)) *DONE*)
 (VERB ((V (# #)) (CAT VP :E) (NUMBER {PROT NUMBER} :E) (PATTERN (V DOTS) :E)) *DONE*)
 (GOAL ((N (# # #)) (CAT NP :E) (PROPER YES :E) (PATTERN (N) :E)) *DONE*)
 (PATTERN (PROT VERB GOAL) :E))
```

Joonis 3.3: Lause "John loves Mary" sisend ja väljund

```
((CAT S) (PROT ((N === JOHN) (NUMBER SING))) (VERB ((N === LOVE) (NUMBER PLURAL)))
 (GOAL ((N === MARY))))

 -->Entering alt TOP -- Jump indexed to branch #1: S matches input S
 -->Updating (CAT NIL :E) with NP at level {PROT CAT}
 -->Updating (CAT NIL :E) with NP at level {GOAL CAT}
 -->Updating (CAT NIL :E) with VP at level {VERB CAT}
 -->Fail in trying PLURAL with SING at level {VERB NUMBER}
```

Joonis 3.4: Tunnuse NUMBER (arv) unifitseerimine

Joonisel 3.3 on näha lause "John loves Mary." funktsionaalne kirjeldus enne ja pärast FUF-interpretaatoriga unifitseerimist. Näeme, et unifitseerija lisab sisendile kõik grammatikast saadud atribuutide-väärtuste paarid:

- peategelase (PROT) John korral on tegu nimisõnafraasiga, kus nimisõna on vaikimisi pärisnimi (hargnemise esimene haru);
- verb (VERB) on kategooriast verbifraas, tema arv peab ühilduma peategelase (PROT) arvuga;
- siht (GOAL) unifitseeritakse analoogiliselt esimese punktiga.

FUF-interpretaator võimaldab kontrollida ka sisendi vasturäägivust. Joonisel 3.4 on toodud funktsionaalne kirjeldus, kus aluse arvuks on määratud ainsus ja verbi arvuks mitmus. Kui unifitseerija jõuab verbi unifitseerimiseni, siis osutub süntaktiliselt korrektne funktsionaalne kirjeldus grammatikaga vasturäägivaks ning unifitseerimine ebaõnnestub.

3.3 FUF-põhised rakendused teistes keeltes

Loomuliku keele generaatorite loetelus "The B to Z of Natural Language Generation Systems" [Bateman jt, 2003] on nimetatud FUF-il baseeruvad generaatorid inglise, hispaania ja heebrea keele jaoks. See loetelu ei ole kindlasti täielik — artiklid [Novello jt] ja [Matiasek jt, 1996] kirjeldavad FUF-põhiseid grammatikaid vastavalt itaalia ja saksa keele jaoks. Viimatimainitud artiklid tutvustavad kaht vastandlikku lähenemist: kui saksa keele puhul kohaldati olemasolev saksa keele HPSG-grammatika unifitseerimisformalismile vastavaks, siis itaalia keele projektis võeti aluseks inglise keele grammatika SURGE ning mugandati see itaalia keelele. Et eesti keele jaoks genereerivat grammatikat ei eksisteeri, siis pakub enam huvi just itaalia keele projekt.

Itaalia keele grammatika koostamisel seati eesmärgiks grammatika SURGE sisendi struktuuri võimalikult täpne säilitamine (kui leksikoni puudutav osa välja arvata), et ressursside korduvkasutus mitmekeelse süsteemi korral oleks maksimaalne. Grammatika koostajate kogemus näitas, et paljudel lihtsatel juhtudel saab sisendina kasutada sama funktsionaalset kirjeldust, kus on tehtud vaid pisimuudatusi ja asendatud lekseemid. Joonisel 3.5 on toodud artiklis [Novello jt] esitatud näide: lausete "This car is expensive." ja "Questa macchina e' costosa." sisendite funktsionaalne kirjeldus.

```
"This car is expensive."
((cat clause)
 (proc ((type ascriptive)
        (mode attributive)))
 (partic ((carrier ((lex "car")
                    (cat common)

                    (distance near)))

          (attribute (lex "expensive")
                     (cat ap)))))))
```

```
"Questa macchina e' costosa."
((cat clause)
 (proc ((type ascriptive)
        (mode attributive)))
 (partic ((carrier ((lex "macchina")
                    (cat common)
                    (gender feminine)
                    (distance near)))

          (attribute (lex "costoso")
                     (cat ap)))))))
```

Joonis 3.5: Funktsionaalne kirjeldus: inglise ja itaalia keele võrdlus

Eestikeelne lause "See auto on kallis." vastab siin täpselt ingliskeelse lause sisendi funktsionaalsele kirjeldusele.

Loomulikult ei ole keerulisemate lausete funktsionaalseid kirjeldusi võimalik üksühesesse vastavusse seada ning see tähendab, et mitmekeelse süsteemi korral peab mikroplaneerimise tasandil keelte erinevusi siiski teatud määral arvesse võtma. Kuid itaalia keele grammatika koostajad nendivad:

> Nendes tekstides, mida meie oleme genereerinud, ei ole leidunud lauset, kus muutusi tuleks teha kõrgemal kui mikroplaneerimise tasandil.

3.4 Paketi FUF eesti keelele kohaldatavuse analüüs

Kuna FUF-interpretaator võimaldab grammatika koostamisel rakendada erinevaid formalisme ega sea kitsendusi sisendi abstraktsioonitasemele, võib FUF-i formalismi lugeda keelest sõltumatuks. FUF-i kasutamise peamine eelis avaldub aga ressursside taaskasutusel. Kui võtta eesti keele grammatika koostamisel aluseks inglise keele grammatika SURGE ja seada eesmärgiks tema sisendi ehituse võimalikult täpne säilitamine, on tehtud suur samm mitmekeelse genereeriva süsteemi loomisel — asendamist vajab genereerimise konveierarhitektuuris sel juhul vaid leksikon.

Grammatika kohandamise eksperiment on juba edukalt läbi viidud itaalia keele jaoks [Novello jt], kuid võib eeldada, et eesti keele korral on töö keelte suurema erinevuse tõttu oluliselt mahukam. Suurimad raskused grammatika kirjutamisel ei selgu kahjuks enne kui reaalse töö käigus, kuid kuna SURGE kasutab peamiselt süntaksist kõrgema taseme sisendit, ei ole hetkel põhjust arvata, et eesti keele grammatika koostamine selle sisendi baasil võiks osutuda võimatuks. Isegi kui selgub, et väga palju muutusi tuleb sisse viia juba mikroplaneerimise tasemel, on FUF-i kasutamine siiski samm rakenduste ühtlustamise suunas ning põhjalikult uuritud inglise keele grammatika SURGE teoreetiline baas on eesti keele grammatika koostamisel abiks. Peatükis 5 on alustatud tööd SURGE fragmentide modifitseerimisega ning valminud on lõik eesti keele nimisõnafraasi grammatikast.

Morfoloogiline komponent tuleb eesti keele pindrealisatsiooni korral täielikult asendada, kuid lineariseerijat ehk sõnade järjekorra seadjat on tõenäoliselt võimalik mugandada. Itaalia keele grammatika loomisel modifitseeriti lineariseerijat hinnanguliselt vaid 5% ulatuses.

Näide: lihtne eesti keele grammatika

Selleks, et demonstreerida mõningaid võimalusi, mida unifitseerimise formalism pakub eesti keele jaoks, vaatleme lihtsat eesti keele grammatikat. Joonisel 3.6 toodud grammatika on grammatika *gr*0 (vt joonist 3.1 alapeatükis 3.2) mugandus eesti keelele ja võimaldab genereerida lauseid nagu "Jüri armastab Marit.". Et FUF ei sisalda eesti keele morfoloogilist komponenti, siis on selle grammatika näitel praktiliselt võimalik testida vaid unifitseerimist ja sõnade järjekorraga manipuleerimist.

Toome välja olulisemad erinevused võrreldes analoogilise inglise keele grammatikaga.

- Real 10

  ```
  10:       (person {prot person})))
  ```

  on lisatud isiku ühilduvus, mida grammatika *gr*0 ei sisaldanud, ehkki sellise ühilduvuse kontroll on vajalik ka inglise keeles.

- Kaotatud on pärisnime ja hariliku nimisõna eristamine. Grammatikas *gr*0 oli see vajalik artikli esinemise üle otsustamiseks; eesti keeles moodustatakse laused "Jüri armastab Marit." ja "Jüri armastab raamatuid." analoogiliselt, sihitise nimisõnafraasi käsitluses vahet pole.

- Et demonstreerida eesti keele sõnajärje käsitlust, on lisatud read 11–16:

  ```
  11:   ;; Kaks alternatiivi sõnade järjekorra valikuks
  12:    (alt (
  13:   ;; sõnade järjekord neutraalne
  ```

```
 1: ((alt top (
 2:   ;; Esimene haru:
 3:   ;; kirjeldame kategooriat S (lause).
 4:   ((cat s)
 5:    (prot ((cat np)))
 6:    (goal ((cat np)))
 7:    (verb ((cat vp)
 8:   ;; Arvu ja isiku ühilduvus
 9:      (number {prot number})
10:      (person {prot person})))
11:   ;; Kaks alternatiivi sõnade järjekorra valikuks
12:    (alt (
13:   ;; sõnade järjekord neutraalne
14:         ((focus neutral)(pattern (prot verb goal)))
15:   ;; fookus alusel
16:         ((focus prot)(pattern(goal verb prot))))))
17:
18:   ;; Teine haru: nimisõnafraas
19:   ((cat np)
20:    (n ((cat noun)))
21:    (pattern (n)))
22:
23:   ;; Kolmas haru: verbifraas
24:   ((cat vp)
25:    (pattern (v))
26:    (v ((cat verb)))))))))
```

Joonis 3.6: Lihtne eesti keele grammatika *eesti*1

```
((CAT S) (FOCUS PROT) (PROT ((N === JÜRI) (NUMBER SING) (PERSON THIRD)))
 (VERB ((V === ARMASTAMA))) (GOAL ((N === MARI))))

 ((CAT S :I) (FOCUS PROT :I)
 (PROT ((N (# #)) (NUMBER SING :I) (PERSON THIRD :I) (CAT NP :E) (PATTERN (N) :E)) *DONE*)
 (VERB
 ((V (# #)) (CAT VP :E) (NUMBER {PROT NUMBER} :E) (PERSON {PROT PERSON} :E) (PATTERN (V) :E))
  *DONE*)
 (GOAL ((N (# #)) (CAT NP :E) (PATTERN (N) :E)) *DONE*) (PATTERN (GOAL VERB PROT) :E))
```

Joonis 3.7: Lause "Marit armastab Jüri" sisendi unifitseerimine

```
14:          ((focus neutral)(pattern (prot verb goal)))
15:    ;; fookus alusel
16:          ((focus prot)(pattern(goal verb prot))))))
```

Joonisel 3.7 on toodud lause "Marit armastab Jüri." sisend ja grammatikaga unifitseeritud väljund. Joonise viimasel real näeme, et atribuudi-väärtuse paar (FOCUS PROT) tingib sõnade järjekorra otsustamisel teise haru (rida 16) valiku.

Peatükk 4

Teisi levinumaid pindrealisaatoreid

4.1 KPML

John Batemani ja Michael Zocki loomuliku keele genereerimise süsteemide loetelu sisaldab märtsis 2004 viiteid 426 projektile. Ehkki see loetelu ei ole kindlasti täielik, on märkimisväärne, et vaid 8 neist projektidest on esitatud kui keelest sõltumatud. Mitmekeelsetest süsteemidest vaieldamatult levinuim on KPML (Komet-Penman Multilingual), mille veebilehel [KPML] on loetletud pindrealisatsioonikomponendid 11 keele jaoks.

KPML-põhiste lahenduste suur arv on põhjendatav sellega, et süsteemi loojate pikaajaline eesmärk ongi võimaldada mitmekeelset genereerimist, kasutades üht sisendit ja varieerides vaid grammatikat ja leksikoni. Sellest tulenevalt on KPML sisend väga kõrgel abstraktsioonitasemel — ka Reiteri ja Dale'i [Reiter jt, 2000] hierarhias on KPML sisendi järgi paigutatud kõrgemale kui FUF. Kuigi KPML-i sisend sisaldab grammatilisi atribuutide-väärtuste paare nagu `tense: past` ehk `aeg: minevik`, on tema sisendi põhistruktuur täielikult semantiline.

KPML-i püüd sobida võimalikult paljudele keeltele tähendab, et pindrealiseerija peab igal üksikul juhul ära tegema väga suure hulga tööd. See on ka

peamine argument, miks käesolevas töös on keskendutud FUF keskkonnale — esimese generaatori loomisel on mõistlik alustada väiksemast alamülesandest. Samuti ei ole KPML-põhine generaator otstarbekas, kui üheks lähimaks eesootavaks ülesandeks pidada masintõlkimist. Masintõlkes ei teostata harilikult süvasemantilist analüüsi ning seetõttu ei vaja ka generaator sisendina nii kõrget esitust.

4.2 Verbmobil

Verbmobil on kõnetõlkija inglise, saksa ja jaapani keele jaoks, mille jaoks on välja töötatud kaks sarnast genereerimiskomponenti, VM-GEN ja VM-GIFT [Becker, Becker jt, 2000], koondnimetusega VM-GECO. Uuem komponent VM-GIFT põhineb puude ühendamise grammatikal TAG (ingl k*Tree Adjoining Grammar*). TAG koosneb nn elementaarpuudest (võrreldav harudega FUF-grammatikas) ning generaatori ülesanne on puude kombineerimisel moodustada täielik lausepuu. Erinevalt FUF-põhisest grammatikast SURGE sisaldab Verbmobili pindrealisaatori sisend juba kõiki suletud klassi sõnu peale abiverbide ning seega on mikroplaneerimisel suurem rõhk.

4.3 Genereerimine lõplike automaatidega

Ehkki Noam Chomsky demonstreeris juba 1957. aastal, et loomulik keel ei ole tervikuna lõplike automaatidega kirjeldatav, on lõplikke automaate (ja muundureid) osades valdkondades, näiteks morfoloogia kirjeldamisel, edukalt kasutatud. Käesoleva töö koostamisel uuriti võimalusi süntaksi genereerimiseks lõplike automaatide abil.

Lauri Karttunen demonstreerib artiklis [Karttunen, 2000] küll süntaksi kirjeldamist lõplike automaatide abil, kuid teeb seda väga lihtsa näite — kuupäevade esitus — varal. Karttunen rõhutabki, et lõplikud automaadid sobivad kitsa valdkonna ülesande lahendamiseks.

Fernando Pereira ja Rebecca Wright esitavad artiklis [Pereira jt] üldise-

ma lähenemise, vaadeldes fraasistruktuurigrammatikate lähendusi lõplikele automaatidele. Töös käsitletakse kõnetuvastuse ülesannet ning autorid möönavad, et, kasutades fraasistruktuurigrammatika lähendust lõplikule automaadile, võidetakse arvutusvõimsuses, kuid kaotatakse väljendusvõimsuses. Artiklis kirjeldatud lähendusmeetodi motivatsioon on just reaalajarakenduse ajapiiri arvestamine.

Seega sobivad lõplikud automaadid paremini kitsa valdkonna ülesannete lahendamiseks või reaalajasüsteemide ajakasutuse optimeerimiseks kui üldise korduvkasutatava süntaksigeneraatori loomiseks.

Peatükk 5

Nimisõnafraasi grammatika

Selles peatööükis on vaatluse all nimisõnafraasi süntaks. Vaadeldakse inglise keele grammatikat SURGE ning analüüsitakse, kuidas katab SURGE nimisõnafraasi grammatika eesti keele nimisõnafraasid.

Nimisõnafraas on valitud seetõttu, et see grammatikakomponent on võrdlemisi eraldiseisev ning võimaldab autoril tutvuda FUF süntaksiga ning SURGE modulaarse ülesehitusega.

5.1 Nimisõnafraas inglise keele grammatikas SURGE

Michael Elhadad [Elhadad, 1992] esitab nimisõnafraasi spetsifikatsiooni, millel põhineb SURGE grammatika nimisõnafraasi realisatsioon:

determiner-sequence	describers	classifiers	head	qualifiers
määratleja-jada	kirjeldajad	klassifikaatorid	põhi	täpsustajad
these two	*elderly*	*nuclear*	*scientists*	*with glasses*

Niisiis fraasis "these two elderly nuclear scientists with glasses" (eesti k. "need kaks vanaldast prillidega tuumateadlast"[1]) on "these two" määratlejad,

[1]Inglise keele nimisõnafraasile vastab siinkohal eesti keeles hulgafraas. Kuna üks grammatika katab mõlemad, võidakse edaspidi hulgafraasile viidata kui nimisõnafraasile. Hulgafraasi ja nimisõnafraasi vahekorda ning käsitlust formaalses grammatikas uuritakse lä-

"elderly" kirjeldaja, "nuclear" klassifikaator, "scientists" fraasi põhi ja "with glasses" täpsustaja. Tuleb rõhutada, et tegu on moodustajate järjekorra prototüübiga, mis võib teatud juhtudel varieeruda. Näiteks juhul, kui fraasi põhi on määra-asesõna, asetub põhi kirjeldajast ettepoole ("something green" — eesti k. "midagi rohelist"). Samuti sõltub fraasi põhjast, millised moodustajad üldse võivad esineda. Vastavate kitsenduste seadmine on üks formaalse grammatika ülesandeid. Selles peatükis selgitataksegi järgnevalt määratlejate, kirjeldajate, klassifikaatorite ja täpsustajate semantiliste ja süntaktiliste rollide vahekorda grammatikas SURGE. Järgmine peatükk uurib, kas ja kuidas on eesti keele nimisõnafraasi moodustajad võimalik nendele rollidele vastavusse seada.

Fraasi põhjana eristab grammatika SURGE järgmisi süntaktilisi kategooriaid:

1. harilik nimisõna;
2. pärisnimi;
3. asesõna:
   (a) isikuline asesõna;
   (b) küsiv asesõna;
   (c) siduv asesõna;
   (d) nn määra-asesõna (ingl. k. *quantified pronoun*) - vastab eesti keele käsitluses määratlevale või umbmäärasele asesõnale — nt fraasis "something green" (eesti k. "midagi rohelist");
   (e) näitav asesõna;
4. teonimi.

Määratleja-jada on kompleksne moodustaja, mis SURGE-is on realiseeritud eraldi grammatika komponendina. Määratleja-jada moodustajad võib jagada kahte suurde gruppi:

hemalt peatükis 5.4.

loenduv	possessiiv	täishulk	fraas i.k.	fraas
jah	jah	+	all of my friends	kõik minu sõbrad
jah	jah	-	none of my friends	mitte ükski mu sõpradest
jah	ei	ei	a few people	paar inimest
ei	ei	ei	a little sugar	natuke suhkrut

Tabel 5.1: Deiktikute valik määratleja-jadas

- fraasi põhja hulka määratlevad sõnad, näiteks "*these three* words" — "*need kolm* sõna" või "*the tenth* commandment" — "*kümnes* käsk";

- sõnad, mis määratlevad nimisõnafraasi põhjaks oleva hulga alamhulga. Need on näiteks kogust väljendavad arv- või asesõnad. Fraasis "*some* of the tigers" — "*mõned* tiigritest)" määratleb asesõna "some" hulga tiigrid ("the tigers" — siin artikkel "the" on põhihulga määratleja) alamhulga.

Määratleja-jada võib sisaldada mõlema grupi moodustajaid üheaegselt, nagu näiteks fraasis "*the first five of the ten* commandments" — "*esimesed viis kümnest* käsust". Määratleja-jada moodustamine on erakordselt keeruline seetõttu, et suletud klassi sõnade valik on jäetud grammatika ülesandeks. Grammatikas SURGE kontrollib määratleja-jada moodustamist 12 tunnust, mis määravad näiteks, kas fraasi põhi on loenduv (countable: yes/no), kas fraas väljendab kuuluvust (possessive: yes/no), kas kirjeldatav alamhulk sisaldab kogu põhihulga (total:+/-/no) jne. Tabel 5.1 illustreerib deikikute ehk vaese leksikaalse tähendusega sõnade valikut kolme tunnuse näitel.

Määratleja-jada detailne analüüs jäetakse käesolevas töös esialgu vaatluse alt välja. Küll selgitatakse peatükis 5.4 seda, kuidas määratlejad mõjutavad valikut nimisõna- ja hulgafraasi vahel.

Kirjeldajad ja klassifikaatorid on, nagu termin ütleb, vastavalt kirjeldavad või liiki/laadi näitavad sõnad või fraasid. Kirjeldajate ja klassifikaatorite süntaktilised rollid võivad kattuda. Peatüki alguses toodud näitelauses on nii kirjeldaja "elderly" kui klassifikaator "nuclear" omadussõnad. SURGE eristab kirjeldajaid ja klassifikaatoreid järgmise reegli alusel: kirjeldajad võivad esi-

neda öeldistäitena, klassifikaatorid mitte. Niisiis on grammatiliselt korrektne öelda "the scientists are elderly" — "teadlased on vanaldased", kuid mitte "the scientists are nuclear" — "teadlased on tuuma(uuringute)", ning seetõttu on ühel juhul tegu kirjeldaja, teisel juhul aga klassifikaatoriga. Lisaks omadussõnadele võivad kirjeldajad olla veel nii oleviku kui mineviku kesksõnad või isegi (neid kesksõnu sisaldavad) lühikesed fraasid, nt "a *man eating* tiger" — "*inimsööja* tiiger".

Klassifikaatorid on tavaliselt kas omadussõnad või nimisõnad nagu fraasis "*office* furniture" — "*kontori*mööbel". Kirjeldajate ja klassifikaatorite eristamine on inglise keeles vajalik sõnade järjekorra määramiseks — kirjeldajad eelnevad klassifikaatoritele. Seejuures nii kirjeldajate kui klassifikaatorite omavahelise järjekorra määramine grammatikas realiseeritud ei ole, sisendilt oodatakse järjestatud nimistut.

Inglise keele nimisõnafraasi täpsustaja võib SURGE järgi olla kõrvallause ("the tiger *that ate my hat*" — "tiiger, *kes mu mütsi ära sõi*"), kesksõnafraas ("the bear *dancing on the table*" — "karu, *kes tantsib laua peal*"), eessõnafraas ("the boy *with glasses*" — "*prillidega* poiss") või lisand ("Paris, *the capital of France*" — "Pariis, *Prantsusmaa pealinn*"). SURGE-is eristatakse kitsendavaid ja mitte-kitsendavaid täpsustajaid; lause moodustamisel käituvad nad erinevalt kirjavahemärkide ja siduvate sõnade osas. Kitsendavale kõrvallausele eelneb kas asesõna "that" või jäetakse asesõna üldse ära nagu lauses "the elephant *we saw in the zoo*" — "elevant, *keda me nägime loomaaias*"; kitsendavale täpsustajale ei eelne kirjavahemärki. Mitte-kitsendavad täpsustajad nagu lisandid või nn "wh-kõrvallaused" eraldatakse inglise keeles komaga.

5.2 Grammatika SURGE ja eesti keele nimisõnafraas

Fraasi põhi

Eesti keele nimisõnafraasi põhja võimalikud süntaktilised kategooriad on üldjoontes vastavuses grammatika SURGE klassifikatsiooniga.

Inglise keeles väljendab nn "ing"-vorm nii kestvat olevikku ("I am dancing" — "Ma tantsin"), oleviku kesksõna ("A dancing bear" — "Tantsiv karu") kui teonime ("Dancing is fun" — "Tantsimine on lõbus"). Eesti keeles seevastu kasutatakse teonimele vastavat "mine"-vormi ainult nimisõna funktsioonis. Selgitamist vajab, kas inglise keeles eraldi kategooriana välja toodud teonimi väärib eesti keeles eraldi alajaotust, st kas teonimi käitub fraasi põhjana teisiti kui harilik nimisõna.

M. Elhadad [Elhadad, 1992] toob ühe SURGE laiendamisvõimalusena välja nominalisatsioonide tuletamise (osa)lausetest, nii et näiteks lause "The barbarians destroyed the city" — "Barbarid hävitasid linna" plaanist oleks võimalik genereerida ka vastav nimisõnafraas: "the destruction of the city by the barbarians" — "linna hävitamine barbarite poolt". Nominalisatsiooni korral on fraasi moodustajatele võimalik seada vastavusse nende semantilised rollid lauses; antud näite korral on "barbarid" tegija ja "linn" tegevusobjekt. (Selliste nimisõnafraaside põhi ei ole muidugi alati teonimi, vaid võib olla ka mõni muu verbi tuletis.) Seda arvesse võttes on ka eesti keeles teonime ja hariliku nimisõna eristamine fraasi põhjana otstarbekas, kuna teonimi säilitab alusverbi rektsiooni, nagu fraasides "räägime tööst (*millest?*)" ja "tööst (*millest?*) rääkimine". Seevastu näiteks oleviku kesksõna, mis esineb eesti keeles samuti üksikutel juhtudel nimisõnafraasi põhjana (nt lauses "Jälitatav põikas kõrvaltänavasse"), käitub nagu omadussõna nimisõna funktsioonis.

Omadussõnaline täiend

Nimisõnafraasi omadussõnalised täiendid on

1. omadussõna(fraasi)d, mis täidavad fraasis reeglina kirjeldaja rolli;
2. omadussõnalised asesõna(fraasi)d, mis eelnevad kirjeldajatele ja kuuluvad määratleja-jadasse (nt "*kõik need* ilusad aastad");
3. põhi- ja järgarvsõna(fraasi)d, mis samuti täidavad määratleja rolli (nt hulgafraasis "*esimesed kolm* kollast liblikat");
4. kesksõna- ja *mata*-lühendi(fraasi)d. Siin on liigitamine keerukam. Kui üldiselt täidavad need lühendid kirjeldavat rolli, siis näiteks fraasis "inglise-eesti seletav sõnaraamat" tähistab kesksõna "seletav" sõnaraamatu liiki ja on seega klassifikaatori rollis;
5. harvem tegijanimi, mis täidab kirjeldaja rolli.

Nimisõnaline täiend

Eesti kirjakeele käsiraamat [EKK] ütleb, et "nimisõnaline täiend omastavas käändes märgib sündmuse osalist või asjaolu". Tabel 5.2 toob näiteid sellistest täienditest ja nende võimalikud vasted inglise keeles. Ehkki süntaktiliselt on eesti keeles alati tegu sama tüüpi täiendiga, vastavad need täiendid SURGE liigituse järgi erinevatele moodustajatele. Korrektse sõnade järjekorra realiseerimiseks on selline eristamine vajalik ka eesti keeles. Nimisõnalised täiendid, mis näitavad kuuluvust, vastavad küsimusele *kelle? mille?*, käituvad sõnade järjekorra valikul nagu määratlejad, samas kui liiki/laadi näitavad täiendid käituvad kui klassifikaatorid. Tabel 5.3 illustreerib nimisõnalise täiendi paigutumist nimisõnafraasis.

Kui eesti keele nimisõnalisele täiendile omastavas käändes võib inglise keeles vastata täpsustaja eessõnafraasi *"of..."* näol (nagu näites "olemise kergus" — "lightness of being"), siis vastupidine ei kehti. Niisiis täidab nimisõnaline täiend omastavas käändes määratleja või klassifikaatori rolli, nagu näidatud joonisel 5.1.

Johni auto *kelle?*	*John's* car määratleja
olemise kergus *mille?*	the lightness *of being* täpsustaja
Oxfordi sõnaraamat *milline?*	*Oxford* dictionary klassifikaator
laste mänguväljak *kelle?* *või* *milline?*	*children's* playground määratleja klassifikaator
uudiste agentuur *milline?*	*news* agency klassifikaator

Tabel 5.2: Nimisõnaline täiend omastavas käändes

Johni määratleja	**punane** kirjeldaja	**auto** põhi	punane	Johni	auto
olemise määratleja	**talumatu** kirjeldaja	**kergus** põhi	talumatu	olemise	kergus
Oxfordi	mahukas	sõnaraamat	**mahukas** kirjeldaja	**Oxfordi** klassifikaator	**sõnaraamat** põhi
laste määratleja	**uus** kirjeldaja	**mänguväljak** põhi	**uus** kirjeldaja	**laste** klassifikaator	**mänguväljak** põhi
uudiste	edukas	agentuur	**edukas** kirjeldaja	**uudiste** klassifikaator	**agentuur** põhi

Tabel 5.3: Nimisõnaline täiend määratleja ja klassifikaatori rollis

```
 "the unbearable lightness of being"
 ((cat common)
   (head === "lightness")
   (describer === "unbearable")
   (definite yes) ;; lisab artikli määratleja-jadasse
   (qualifier ((cat pp) ;;  "of" on vaikimisi eessõna
       (restrictive yes) ;; ei eraldata komaga
       (np ((cat nominalized-ing)
    (head === "be"))))))

 "olemise talumatu kergus"
((cat common)
   (head === "kergus")
   (describer === "talumatu")
   ;; info määratlejate valimiseks
   (possessor ((cat nominalized-ing)
                (head === "olema"))))
```

Joonis 5.1: Näide: nimisõnafraasi FD inglise ja eesti keeles

Muus vormis täiendid

Muus vormis täiendid võivad fraasi põhjale nii eelneda kui järgneda ning täita mitmesuguseid semantilisi rolle. Järeltäiend vastab harilikult inglise keele eessõnafraasile ning täidab täpsustaja rolli ("vaade *järvele*" — "a view *to the lake*", "krokodill *Kreekast*" — "a crocodile *from Greece*"). Juba mainitud "of"-fraasi korral ja mõnel teisel juhul kasutatakse eesti keeles eestäiendit: näiteks fraasis ("*prillidega* poiss") täidab sõna "prillidega" kirjeldavat rolli. Et otsustused semantiliste rollide kohta tehakse lauseplaneerimise tasandil, siis jääb nimisõnafraasi grammatika ülesandeks realiseerida korrektne sõnade järjekord ja kirjeldada sobivad vaike-väärtused (nt suletud klassi sõnad, vaikimisi käänded jms). Alapeatükis 5.3 kirjeldatakse fraasiliikmete ühilduvust ja käänete vaikeväärtuste valikut.

Lisand

Lisand, mis tähistab sama objekti kui põhi, on grammatikas SURGE realiseeritud võtmesõna `complex` abil. Joonis 5.2 (näited grammatika SURGE testpaketist [Elhadad jt]) illustreerib ees- ja järellisandi kasutamist. Analoogilist konstruktsiooni kasutades on võimalik eesti keeles realiseerida põhjaga ühilduvad lisandid, näiteks "Eesti Vabariigi pealinn Tallinn" ja "Tallinn, Eesti Vabariigi pealinn" (joonis 5.3). SURGE-is on ka eraldi realiseeritud isikunimede grammatika, mis võimaldab nimele lisada nimetavas käändes lisanditarindeid nagu "kapten Trumm" või "dr. Aivaluson". Muudes käänetes ühildumatute lisandite korrektseks realiseerimiseks tuleb grammatikat täiendada.

Samaväärsed täiendid

Eesti keeles eraldatakse samaväärsed täiendid sidesõna või komaga. Kompleksed täiendid on võimalik realiseerida täpselt samuti nagu lisandid, kasutades tunnust `restrictive` määramaks, kas täiendid eraldatakse ("*naeratav, irvitav* kass") või mitte ("*paks vöödiline* kass").

5.3 Ühilduvus eesti keele nimisõnafraasis. Vaikeväärtused

Eelmise peatüki analüüs näitab, et inglise keele nimisõnafraasi struktuur, nagu see on realiseeritud grammatikas SURGE, on võimalik üldjoontes eesti keelele üle kanda. Nimisõnafraasi eestäiendite järjekorda on võimalik osaliselt kontrollida grammatika-siseselt, eristades määratlejaid, kuuluvust, omadust ning liiki või laadi näitavaid täiendeid. Niisiis moodustavad nimisõnafraasi:

1. fraasi põhi — nimi- või asesõna;
2. määratlejad — kompleksne arv- ja asesõnadest ning kuuluvust näitavatest nimisõna(fraasi)dest koosnev moodustaja;
3. kirjeldaja(d) — harilikult omadussõnaline täiend, aga ka muus vormis täiend, kesksõnafraas vm;

```
"My wife, Carol."
((cat np)
 (complex apposition)
 (restrictive no)
 (distinct ~(((cat common)
              (head === "wife")
              (possessor ((cat personal-pronoun)
                          (person first))))
             ((cat proper)
              (restrictive no)
              (head === "Carol")))))

"My brother Steve."
((cat np)
 (complex apposition)
 (restrictive yes) ;; kirjavahemärgi kasutamine
 (distinct
  ~(((cat common)
     (head === "brother")
     (possessor  ((cat personal-pronoun)
                  (person first))))
    ((cat proper)
     (head === "Steve")))))
```

Joonis 5.2: Ees- ja järellisand inglise keeles

4. klassifikaator(id) — harilikult nimisõnaline täiend omastavas käändes, harvem keksõnafraas vm;

5. täpsustaja(d) — kõrvallause või määruslik täiend.

Inglise keeles ei ühildu kirjeldajad ega klassifikaatorid fraasi põhjaga ning seetõttu ei edastata käände informatsiooni nendele moodustajatele. Eesti keeles tuleb grammatikasse vastavad täiendused teha. Lisas 2 on SURGE grammatika põhjal koostatud fragment eesti keele nimisõnafraasi grammatikast, kuhu on lisatud nimisõnalise määratleja, kirjeldaja ja klassifikaatori käände

```
"Tallinn, Eesti Vabariigi pealinn."
((cat np)
 (complex apposition)
 (restrictive no)
 (distinct ~(((cat proper)
              (head === "Tallinn")))
      ((cat common)
               (head === "pealinn")
               (possessor ((cat proper)
                           (head === "Eesti Vabariik"))))))

"Eesti Vabariigi pealinn Tallinn."
((cat np)
 (complex apposition)
 (restrictive yes) ;; kirjavahemärgi kasutamine
 (distinct
  ~(((cat common)
     (head === "pealinn")
     (possessor  ((cat proper)
                  (head === "Eesti Vabariik"))))
    ((cat proper)
     (head === "Tallinn")))))
```

Joonis 5.3: Ees- ja järellisand eesti keeles

kontroll, samuti on lisatud vaikeväärtused fraasi käände (nimetav) ja arvu (ainsus) valikuks.

Joonisel 5.4 on toodud fraasi "olemise talumatu kergus" funktsionaalne kirjeldus enne ja pärast unifitseerimist. Sisendkirjeldus ei sisalda mingit informatsiooni fraasi käände ega arvu kohta. Väljundis on fraasi põhja ja kirjeldaja käändena määratud nimetav kääne ja arvuna määratud ainsus, samas kui määratleja kääne on omastav (arv on samuti vaikimisi ainsus). Käände ühilduvuse tõttu ebaõnnestub fraasi "mahukaga sõnaraamatuta" kirjelduse

unifitseerimine (näide 4 lisas 3), samas kui fraas "saabastega kassile" (näide 2) unifitseeritakse edukalt, sest siin ei ole tegu omadussõnalise täiendiga. Näide 7 ("Johni venna punases Ferraris") demonstreerib, kuidas määratleja võib omakorda olla rekursiivselt unifitseeritav kompleksne nimisõnafraas ("Johni venna"). Põhjalikumad näited grammatikaga *npest* unifitseerimisest on toodud lisas 3. Kuna FUF-i ei ole hetkel integreeritud eesti keele morfoloogiamoodulit, siis on testi tulemusena võimalik väljastada ainult rikastatud funktsionaalne kirjeldus.

5.4 Nimisõnafraas ja hulgafraas inglise ja eesti keeles

Grammatika SURGE käsitleb hulgafraasi tunnuse `partitive` abil. Hulgafraasi moodustajad on `part, part-of, prep` ehk osa, tervik ja eessõna. Hulgafraasis tuuakse eraldi välja ka tunnus *total*, mille väärtuse "+" korral lisandub hulgafraasi ette asesõna "all". Niisiis on hulgafraasi moodustajate järjekord inglise keeles *(all part prep part-of)*. Kui jätta kõrvale eessõna puudumine, siis kehtib sama reegel ka eesti keeles — fraasi osa on hulgafraasi põhi ning tervik laiend. Asesõna "kõik", kui ta fraasis esineb, on asesõnadest esimene.

Erinevalt inglise keelest jagunevad eesti keele hulgafraasid omakorda osastavas ("seitse pöialpoissi") ja seestütlevas käändes ("kolm seitsmest pöialpoisist") laiendiga hulgafraasideks, vastavalt sellele, kas fraas on nimetav või märgib kindlat osa tervikust. Sealjuures osastavas käändes laiendiga fraasidele, mille põhi on arvsõna, vastab inglise keeles harilik nimisõnafraas ("seitse pöialpoissi" — "the seven dwarves"). Hulgafraasi tüübi vaikimisi väärtust on raske määrata (kas öelda "kaks poissi" või "kaks poistest"?), kuid välja võib tuua järgmise üldise reegli: kui hulgafraasi terviku põhjal on määratleja ("kaks *nendest* poistest"), või on terviku põhi asesõna, siis on hulgafraasi laiend seestütlevas käändes. Sealjuures seestütlevas käändes laiend jääb alati seestütlevasse käändesse (vrdl "seitsme pöialpoisiga" ja "kolmega seitsmest pöialpoisist").

```
((cat common)
 (head ((lex "kergus")))
 (describer  ((cat adj)
              (lex "talumatu")))
 (possessor ((cat common)
             (head ((lex "olemine")))))))


((CAT COMMON :I)
 (HEAD
  ((LEX "kergus" :I) (CAT NOUN :E) (CASE {CASE} :E)
   (NUMBER {NUMBER} :E)))
 (DESCRIBER
;;kirjeldaja ühildub arvus ja käändes põhjaga
  ((CAT ADJ :I) (LEX "talumatu" :I) (NUMBER {HEAD NUMBER} :E)
   (CASE {HEAD CASE} :E)))
 (POSSESSOR
  ((CAT COMMON :I)
   (HEAD
    ((LEX "olemine" :I) (CAT NOUN :E) (CASE {POSSESSOR CASE} :E)
     (NUMBER {POSSESSOR NUMBER} :E)))
;; määratleja kääne on omastav, arv on vaikimisi ainsus
   (CASE GENITIVE :E) (PATTERN (DOTS HEAD) :E) (NUMBER SINGULAR :E)
   (POSSESSOR NONE :E) (DESCRIBER NONE :E) (CLASSIFIER NONE :E))
  *DONE*)
;; moodustajate järjekord: määratleja, kirjeldaja, põhi
;; fraasi vaikimisi kääne nimetav, vaikimisi arv ainsus
 (PATTERN (POSSESSOR DOTS DESCRIBER DOTS HEAD) :E)
 (CASE NOMINATIVE :E)
 (NUMBER SINGULAR :E) (CLASSIFIER NONE :E))
```

Joonis 5.4: Unifitseerimine grammatikaga *npest*

Oluliselt keerulisem on eesti keeles ka hulgafraasi laiendi arvu valik. Põhiarvsõnale, asesõnale ja mõõdunimisõnale järgnev osastavas käändes laiend on ainsuses ("kümme väikest neegrit", "mitu kosilast", "tass teed"), samas kui hulka märkivale nimisõnale järgnev laiend on mitmuses ("kari lehmi"). Määrsõnale järgnev laiend võib olla nii ainsuses kui mitmuses ("palju kannatusi", "palju kannatust"). Seestütlevas käändes laiend jääb ainsusesse, kui laiend ise on hulgafraas (vrdl "need poisid" ja "kaks nendest poistest", "seitse pöialpoissi" ja "kolm seitsmest pöialpoisist").

Lisas 4 on toodud fragment SURGE hulgafraasi grammatikast, mida on modifitseeritud nii, et see vastaks eesti keele osastavas käändes laiendiga hulgafraasile. Grammatika liitmisel grammatikaga *npest* on võimalik nüüd genereerida ka fraase nagu "seitse pöialpoissi", "kari lehmi" jne. Näited unifitseerimisest on toodud lisas 5.

Kokkuvõte

Loomuliku keele generaatori standardarhitektuur on konveier, mille etapid on sisuplaneerimine (semantiline struktuur), mikroplaneerimine (temaatiline või süvasüntaktiline struktuur) ja pindrealiseerimine. Käesolev töö analüüsib võimalusi eesti keele pindrealisaatori loomiseks. Sealjuures on eesmärgiks seatud ressursside korduvkasutatavus — pindrealisaatorit peab saama rakendada muuhulgas nii eesti keele suunalises masintõlkes kui dialoogisüsteemides. Et morfoloogiline komponent on eesti keele jaoks olemas, keskendutakse süntaksi genereerimisele.

Töös antakse ülevaade olemasolevatest loomuliku keele genereerimise süsteemidest, keskendudes mitmekeelsetele pindrealisaatoritele. Lähemalt vaadeldakse funktsionaalse unifitseerimise formalismi FUF [FUF-manual], mille rakendusi demonstreeritakse praktilistelt nii inglise kui eesti keele näitel. Töö viimases etapis on alustatud eesti keele grammatika ühe komponendi, nimisõnafraasigrammatika loomist. Valminud grammatika(fragment) võimaldab genereerida lihtsamaid nimisõnafraase ning jätkata tööd keerulisemate konstruktsioonidega.

Sentence realization for the Estonian language

Term paper

Emilia Käsper

Abstract

The standard architecture of a natural language generation system is a pipeline consisting of text planning (semantic structure), microplanning (thematic or deep syntactic structure) and surface realization. In order to develop generation tools, such as dialogue systems or a machine translation system where the target language is Estonian, a reusable surface realization component for syntax generation is needed. Existing morphology components can be reused, so in this paper, opportunities for sentence realization on the syntactic level are investigated.

This work gives an overview of existing natural language generation systems, focusing on language-independent surface realization components. Opportunities of the FUF package [FUF-manual] based on functional unification formalism are discussed more in detail and an analysis on the adaptivity of FUF to the Estonian language is presented. Actual work has been started on noun phrase grammar. The results allow generation of simple noun phrases and provide a basis for future grammar development.

Kirjandus

[Bateman jt, 2003] J. Bateman, M. Zock. The B to Z of Natural Language Generation Systems, 2003
http://www.fb10.uni-bremen.de/anglistik/langpro/NLG-table/NLG-table-root.htm
— viimati väisatud 07.03.2004

[Becker] T. Becker. Syntactic Generation with a Preprocessed HPSG Grammar
http://www.dfki.de/»becker/Genws/Final/tilman.becker.ps
— viimati väisatud 07.03.2004

[Becker jt, 2000] T. Becker, A. Kilger, P. Lopez, P. Poller. The Verbmobil Generation Component VM-GECO. *W. Wahlster (Ed). Verbmobil: Foundations of Speech-to-Speech Translation.* Springer, 2000. lk 481–496

[Busemann, 1999] S. Busemann. Natural Language Generation. An Overview, 1999
http://www.dfki.de/»busemann/VL-SS99/990415/sld001.htm
— viimati väisatud 05.03.2004

[Elhadad, 1992] M. Elhadad. Using Argumentation to Control Lexical Choice: A Functional Unification Implementation. Doktoriväitekiri. Columbia University, 1992. 360 lk
ftp://ftp.cs.bgu.ac.il/pub/siggen/elhadad-phd.ps.gz — viimati väisatud 08.05.2004

[Elhadad jt] M. Elhadad, J. Robin. An Overview of SURGE: a Reusable Comprehensive Syntactic Realization Component
ftp://ftp.cs.bgu.ac.il/pub/people/elhadad/surge.ps.gz
— viimati väisatud 06.03.2004

[EKI] Morfoloogiline süntesaator. Eesti keele instituut
http://www.eki.ee/tarkvara/vormimoodustus/ — viimati väisatud 06.03.2004

[EKK] Eesti kirjakeele käsiraamat
http://www.eki.ee/books/ekkr/ — viimati väisatud 16.05.2004

[Exemplars] Exemplars. CoGenTex, Inc.
http://www.cogentex.com/technology/exemplars/index.shtml
— viimati väisatud 05.03.2004

[Filosoft] Eesti keele süntesaatori veebidemo. Filosoft
http://www.filosoft.ee/gene_et/ — viimati väisatud 06.03.2004

[FUF-manual] M. Elhadad. FUF: The Universal Unifier User Manual Version 5.2
ftp://ftp.cs.bgu.ac.il/pub/people/elhadad/nlg/fufman.ps
— viimati väisatud 07.03.2004

[FUF/SURGE] Paketi FUF ja grammatika SURGE installatsioonid
http://www.cs.bgu.ac.il/surge/ — viimati väisatud 07.03.2004

[Karttunen, 2000] L. Karttunen. Applications of Finite-State Transducers in Natural-Language Processing, 2000.
www2.parc.com/istl/members/karttune/publications/ciaa-2000/fst-in-nlp/fst-in-nlp.html — viimati väisatud 07.03.2004

[KPML] KPML system
http://www.fb10.uni-bremen.de/anglistik/langpro/kpml/README.html — viimati väisatud 05.03.2004

[Lavoie jt, 1997] B. Lavoie, O. Rambow. A Fast and Portable Realizer for Text Generation Systems. *Proceedings of the Fifth Conference on Applied Natural Language Processing (ANLP97)*. Washington, 1997, lk 265–268. ACL
http://www.cogentex.com/papers/realpro-anlp97.pdf — viimati väisatud 05.03.2004

[Matiasek jt, 1996] J. Matiasek, H. Trost. Implementing HPSG in FUF. An Experiment in the Reusability of Linguistic Resources. *Proceedings of the 16th International Conference on Computational Linguistics (COLING-96)*. Copenhagen, 1996
http://www.oefai.at/cgi-bin/get-tr?paper=oefai-tr-95-14.ps.gz — viimati väisatud 06.03.2004

[Müürisep] K. Müürisep. Eesti keele süntaksianalüsaatori demo
http://www.cs.ut.ee/»kaili/parser/demo/ — viimati väisatud 05.03.2004

[Novello jt] A. Novello, C.B. Callaway. Porting to an Italian Surface Realizer: A Case Study
http://tcc.itc.it/people/callaway/papers/italian.ps — viimati väisatud 06.03.2004

[Pereira jt] F.C.N. Pereira, R.N. Wright. Finite-state approximation of Phrase-Structure Grammars. *E. Roche, Y. Schabes (Ed.). Finite-State Language Processing*. MIT, 1997, lk 149-168

[RealPro] RealPro. CoGenTex, Inc.
http://www.cogentex.com/technology/realpro/index.shtml — viimati väisatud 06.03.2004

[Reiter jt, 2000] E. Reiter, R. Dale. Building Natural Language Generation Systems. Cambridge University Press 2000. 272 lk. *Käsikiri*
`http://www.ling.helsinki.fi/»gwilcock/Tartu/` — viimati väisatud 06.03.2004

[Roosmaa jt, 2001] T. Roosmaa, M. Koit, K. Muischnek, K. Müürisep, T. Puolakainen, H. Uibo. Eesti keele formaalne grammatika. Tartu 2001, 158 lk

[Stenzhorn, 2002] H. Stenzhorn. XtraGen — A Natural Language Generation System Using XML- and Java-Technologies, 2002
`http://www.dfki.de/»holger/publications/stenzhorn-xtragen-nlpxml2002.pdf` — viimati väisatud 05.03.2004

[Verbmobil] Verbmobil. DFKI GmbH
`http://verbmobil.dfki.de/` — viimati väisatud 05.03.2004

Lisad

Lisa 1. Paketi FUF kasutamise logi
Lisa 2. Lihtne eesti keele nimisõnafraasi grammatika *npest*
Lisa 3. Grammatika *npest* testid
Lisa 4. Hulgafraasi grammatika
Lisa 5. Hulgafraaside unifitseerimine

Lisa 1
Paketi FUF kasutamise logi

Käesolevas lisas on toodud kõigi töös kasutatud süsteemi FUF näidete käivitamise logi. Logi on kommenteeritud — töö autori lisatud kommentaariread algavad 5 järjestikuse semikooloniga. Süsteemi FUF testimisel kasutati Allegro Common Lisp Trial Editioni versiooni 6.2 Windowsi platvormil.

```
;;;;; Süsteemi FUF kasutamise logi 03.03.2004.
International Allegro CL Trial Edition 6.2 [Windows] (Jun
26, 2002 11:41) Copyright (C) 1985-2002, Franz Inc., Berkeley, CA,
USA.  All Rights Reserved.

This development copy of Allegro CL is licensed to:
   Emilia, University of Tartu

CG/IDE Version: 1.389.2.105.2.14
;; Optimization settings: safety 1, space 1, speed 1, debug 2.
;; For a complete description of all compiler switches
;; given the current optimization settings
;; evaluate (EXPLAIN-COMPILER-SETTINGS).

[changing package from "COMMON-LISP-USER" to "COMMON-GRAPHICS-USER"]

;;;;; Jooksva kataloogi vahetamine

CG-USER(1): :cd C:/Program Files/fuf
C:\Program Files\fuf\

;;;;; Uue paketi defineerimine

CG-USER(2): (defpackage "FUG5")
#<The FUG5 package>
CG-USER(3): (in-package "FUG5")
#<The FUG5 package>

;;;;; FUG-mootori kompileerimine ja laadimine
```

```
FUG5(4): (compile-file "src/fug5.l")
;;; Compiling file src\fug5.l

;;;;; [Kompileerimisteated eemaldatud]

;;; Writing fasl file src\fug5.fasl
;;; Fasl write complete
#p"src\\fug5.fasl"
T
T
FUG5(5): (load "src/fug5")

;;;;; [Laadimisteated eemaldatud]

    FUF Version 5.3, Copyright (C) 1987-1996 Michael Elhadad.
    FUF comes with absolutely no warranty; for details type (fug5::warranty).
    This is free software, and you are welcome to redistribute it
    under certain conditions, type (fug5::license) for details.
T

;;;;; Näitegrammatika gr0 kompileerimine ja laadimine

FUG5(6): (compile-file "examples/gr0.l")
;;; Compiling file examples\gr0.l
;;; Writing fasl file examples\gr0.fasl
;;; Fasl write complete
#p"examples\\gr0.fasl"
T
NIL
FUG5(7): (load "examples/gr0")
; Fast loading C:\Program Files\fuf\examples\gr0.fasl
T

;;;;; 'grammar-p' kontrollib, kas grammatika süntaks on korrektne

FUG5(8): (grammar-p)
Syntax is correct.
Checking for contradictions and validity of indexes: ......
The grammar is correct.
It contains 0 indexed alts, 0 demo messages and 1 traced alt.
T

;; Funktsionaalne kirjeldus fd1

FUG5(9): (setq fd1 ' ((cat s)
                      (prot ((n === John)))
                      (verb ((v === love)))
                      (goal ((n === Mary)))))
((CAT S) (PROT ((N === JOHN))) (VERB ((V === LOVE))) (GOAL ((N === MARY))))
```

```
;;;;; 'fd-p' kontrollib, kas funktsionaalse kirjelduse süntaks on korrektne

FUG5(10): (fd-p fd1)
Syntax is correct.
No contradictions found.
T
;;;;; 'uni-fd' unifitseerib ilma lineariseerimiseta
;;;;; Väljastatakse rikastatud funktsionaalne kirjeldus

FUG5(11): (uni-fd fd1)

[Used 3 backtracking points - 0 wrong branches - 0 undos]

[Used 3 backtracking points - 0 wrong branches - 0 undos]
((CAT S :I) (PROT ((N (# # #)) (CAT NP :E) (PROPER YES :E) (PATTERN (N) :E)) *DONE*)
 (VERB ((V (# #)) (CAT VP :E) (NUMBER {PROT NUMBER} :E) (PATTERN (V DOTS) :E)) *DONE*)
 (GOAL ((N (# # #)) (CAT NP :E) (PROPER YES :E) (PATTERN (N) :E)) *DONE*)
 (PATTERN (PROT VERB GOAL) :E))

;;;;; 'uni' unifitseerib ja lineariseerib ning väljastab lõpptulemuse

FUG5(12): (uni fd1)

[Used 3 backtracking points - 0 wrong branches - 0 undos]

[Used 3 backtracking points - 0 wrong branches - 0 undos]

;;;;; Lõpptulemus

John loves mary.

;;;;; Selle funktsionaalse kirjeldusega demonstreerime
;;;;; unifitseerimisel toimuvat ühilduvuse kontrolli.
;;;;; Aluse ja öeldise arv ei ühildu.

FUG5(13): (setq fd2 ' ((cat s)
                       (prot ((n === John)(number sing)))
                       (verb ((n === love)(number plural)))
                       (goal ((n === Mary)))))
((CAT S) (PROT ((N === JOHN) (NUMBER SING))) (VERB ((N === LOVE) (NUMBER PLURAL)))
 (GOAL ((N === MARY))))
FUG5(14): (fd-p fd2)
Syntax is correct.
No contradictions found.
T

;;;;; Jälitamise sisselülitamine

FUG5(15): (trace-on)
T
FUG5(16): (uni-fd fd2)
```

```
>

>=======================================
>STARTING CAT S AT LEVEL {}
>=======================================


-->Entering alt TOP -- Jump indexed to branch #1: S matches input S
-->Updating (CAT NIL :E) with NP at level {PROT CAT}
-->Updating (CAT NIL :E) with NP at level {GOAL CAT}
-->Updating (CAT NIL :E) with VP at level {VERB CAT}

;;;;; Unifitseerimine ebaõnnestub

-->Fail in trying PLURAL with SING at level {VERB NUMBER}
:FAIL

;;;;; Eesti keele grammatika kompileerimine ja laadimine

FUG5(17): (compile-file "examples/eesti1.l")
;;; Compiling file examples\eesti1.l
;;; Writing fasl file examples\eesti1.fasl
;;; Fasl write complete
#p"examples\\eesti1.fasl"
T
NIL
FUG5(18): (load "examples/eesti1")
; Fast loading C:\Program Files\fuf\examples\eesti1.fasl
T

;;;;; Süntaksi kontroll

FUG5(19): (grammar-p)
Syntax is correct.
Checking for contradictions and validity of indexes: .....
The grammar is correct.
It contains 0 indexed alts, 0 demo messages and 1 traced alt.
T

;;;;; Funktsionaalne kirjeldus lausele "Jüri armastab Marit"

FUG5(20): (setq fd3 '((cat s)
                      (prot ((n === Jüri)(number sing)(person third)))
                      (verb ((v === armastama)))
                      (goal ((n === Mari)))))
((CAT S) (PROT ((N === JÜRI) (NUMBER SING) (PERSON THIRD))) (VERB ((V === ARMASTAMA)))
 (GOAL ((N === MARI))))

;;;;; Funktsionaalse kirjelduse süntaksi kontroll

FUG5(21): (fd-p fd3)
```

```
Syntax is correct.
No contradictions found.
T

;;;;; Jälitamise väljalülitamine

FUG5(22): (trace-off)
NIL

;;;;; Unifitseerimine ilma lineariseerimiseta
;;;;; Eesti keele lineariseerijat ei ole

FUG5(23): (uni-fd fd3)

[Used 2 backtracking points - 0 wrong branches - 0 undos]

[Used 2 backtracking points - 0 wrong branches - 0 undos]
((CAT S :I)
 (PROT ((N (# #)) (NUMBER SING :I) (PERSON THIRD :I) (CAT NP :E) (PATTERN (N) :E)) *DONE*)
 (VERB
  ((V (# #)) (CAT VP :E) (NUMBER {PROT NUMBER} :E) (PERSON {PROT PERSON} :E) (PATTERN (V) :E))
  *DONE*)
 (GOAL ((N (# #)) (CAT NP :E) (PATTERN (N) :E)) *DONE*) (FOCUS NEUTRAL :E)
 (PATTERN (PROT VERB GOAL) :E))

;;;;; Eelmisel real on näha
;;;;; et vaikimisi valiti neutraalne lauseliikmete järjekord

;;;;; Fookuse muutmise testimine

FUG5(24): (setq fd4 '((cat s)(focus prot)
                      (prot ((n === Jüri)(number sing)(person third)))
                      (verb ((v === armastama)))
                      (goal ((n === Mari)))))
((CAT S) (FOCUS PROT) (PROT ((N === JÜRI) (NUMBER SING) (PERSON THIRD)))
 (VERB ((V === ARMASTAMA))) (GOAL ((N === MARI))))

;;;;; Kontroll ja unifitseerimine

FUG5(25): (fd-p fd4)
Syntax is correct.
No contradictions found.
T
FUG5(26): (uni-fd fd4)

[Used 3 backtracking points - 1 wrong branches - 0 undos]

[Used 3 backtracking points - 1 wrong branches - 0 undos]
((CAT S :I) (FOCUS PROT :I)
 (PROT ((N (# #)) (NUMBER SING :I) (PERSON THIRD :I) (CAT NP :E) (PATTERN (N) :E)) *DONE*)
 (VERB
```

```
  ((V (# #)) (CAT VP :E) (NUMBER {PROT NUMBER} :E) (PERSON {PROT PERSON} :E) (PATTERN (V) :E))
  *DONE*)
 (GOAL ((N (# #)) (CAT NP :E) (PATTERN (N) :E)) *DONE*) (PATTERN (GOAL VERB PROT) :E))

;;;;; Viimasel real on näha, et lauseliikmete järjekord on nüüd GOAL < VERB < PROT
;;;;; Seega moodustaks lineariseerija lause "Marit armastab Jüri."
;;;;; Sessiooni lõpp.
FUG5(27): :exit
```

Lisa 2
Lihtne eesti keele nimisõnafraasi grammatika *npest*

```
;; grammatika npest
;; kasutatud on lihtsustatud lõike grammatikast SURGE

;; fraasi põhi on harilik nimisõna või pärisnimi
;; asesõnu hetkel ei vaatle
;; määratlejatest käsitleme ainult nimisõnu

(define-feature-type np (common proper))

 ((alt np (

;; ap hetkel realiseerimata

  ((cat ap))
  ((cat #(under np))
  (pattern (dots head dots))

;; vaikeväärtused

  (opt ((case nominative)))
```

```
  (opt ((number singular)))

;; fraasi põhi pärib käände ja arvu fraasilt

  (head ((cat noun)
         (case {^2 case})
         (number {^2 number})))

  (:! np-type)

;; määratleja omastavas käändes

  (alt possessor (:index possessor)
    (((possessor given)
      (possessor ((cat #(under np))
                  (case genitive)))
      (pattern (possessor dots head)))
     ((possessor none))
     ((possessor nil)
      (cset ((- possessor))))))

    (:! describer)
    (:! classifier)))))

(def-alt np-type (:index cat)
  (:demo "Kas fraasi põhi on harilik nimisõna või pärisnimi?")

;; harilik nimisõna

   (((cat common))

;; pärisnimel ei või olla klassifikaatoreid

    ((cat proper)
```

```
     (classifier none))))

;; kirjeldaja

 (def-alt describer (:demo "Kas põhjal on kirjeldaja?")(
   ((describer none)(pattern (dots head)))
   ((describer given)
    (pattern (dots describer dots head))
    (describer
     ((alt describer-cat (:index cat)

;; complex ei ole hetkel realiseeritud
;; seega kirjeldaja võib olla 1 omadussõna
;; omadussõna ühildub põhjaga

       (((cat adj)
         (number {^2 head number})
         (:! case-agree))

;; määruslik täiend ei ühildu

((cat ap))
((cat verb)
 (alt verb-ending (:index ending)

;; mineviku kesksõna ei ühildu, oleviku kesksõna ühildub

          (((ending past-participle))
   ((ending present-participle)
            (number {^2 head number})
            (:! case-agree)))))))))))

;; klassifikaator
```

```
 (def-alt classifier (:demo "Kas põhjal on klassifikaator?")
  (((classifier none))
   ((classifier given)(pattern (dots classifier head))
    (classifier
     ((alt classifier-cat (:index cat)

;; kas klassifikaator võib olla terve nimisõnafraas?

       (((cat #(under np))
         (alt case
          (((case genitive))
           ((case given)))))
((cat ap))
((cat verb)
 (ending present-participle)
         (number {^2 head number})
         (:! case-agree)))))))))

;; kui fraas on rajavas, olevas, ilmaütlevas või
;; kaasaütlevas käändes, siis täiend on omastavas käändes

(def-alt case-agree (
 ((control  (or (eq #@{^2 case} 'terminative)
                (eq #@{^2 case} 'essive)
                (eq #@{^2 case} 'abessive)
                (eq #@{^2 case} 'comitative)))
    (case genitive))

;; muudel juhtudel fraasi käändes

  ((case {^2 case}))))
```

Lisa 3
Grammatika *npest* testid

Lisa 3 sisaldab näiteid grammatikaga *npest* unifitseerimisest. Iga fraasi korral on toodud sisendi funktsionaalne kirjeldus ja unifitseerimise tulemus — informatsiooniga rikastatud funktsionaalne kirjeldus või teade unifitseerimise ebaõnnestumisest.

Näide 1. "Olemise talumatu kergus"

```
((cat common)
 (head ((lex "kergus")))
 (describer  ((cat adj)
              (lex "talumatu")))
 (possessor ((cat common)
             (head ((lex "olemine")))))))
```

```
((CAT COMMON :I)
 (HEAD
  ((LEX "kergus" :I) (CAT NOUN :E) (CASE {CASE} :E)
   (NUMBER {NUMBER} :E)))
 (DESCRIBER
  ((CAT ADJ :I) (LEX "talumatu" :I) (NUMBER {HEAD NUMBER} :E)
   (CASE {HEAD CASE} :E)))
 (POSSESSOR
```

```
 ((CAT COMMON :I)
  (HEAD
   ((LEX "olemine" :I) (CAT NOUN :E) (CASE {POSSESSOR CASE} :E)
    (NUMBER {POSSESSOR NUMBER} :E)))
  (CASE GENITIVE :E) (PATTERN (DOTS HEAD) :E) (NUMBER SINGULAR :E)
  (POSSESSOR NONE :E) (DESCRIBER NONE :E) (CLASSIFIER NONE :E))
 *DONE*)
(PATTERN (POSSESSOR DOTS DESCRIBER DOTS HEAD) :E) (CASE NOMINATIVE :E)
(NUMBER SINGULAR :E) (CLASSIFIER NONE :E))
```

Näide 2. "saabastega kassile"

```
((cat common)
          (case allative)
          (head ((lex "kass")
                 (number singular)))
          (describer  ((cat ap)
                       (head ((lex "saabas")))
                       (number plural)
                       (case comitative))))
```

```
((CAT COMMON :I) (CASE ALLATIVE :I)
 (HEAD
  ((LEX "kass" :I) (NUMBER SINGULAR :I) (CAT NOUN :E)
   (CASE {CASE} :E)))
 (DESCRIBER
  ((CAT AP :I) (HEAD ((LEX "saabas" :I))) (NUMBER PLURAL :I)
   (CASE COMITATIVE :I))
  *DONE*)
 (PATTERN (DOTS DESCRIBER DOTS HEAD) :E) (NUMBER SINGULAR :E)
 (POSSESSOR NONE :E) (CLASSIFIER NONE :E))
```

Näide 3. "mahuka Oxfordi sõnaraamatuta"

```
((cat common)
            (case abessive)
            (head ((lex "sõnaraamat")))
            (describer  ((cat adj)
                         (lex "mahukas")))
            (classifier ((cat proper)
                         (head ((lex "Oxford"))))))
```

```
((CAT COMMON :I) (CASE ABESSIVE :I)
 (HEAD
  ((LEX "sõnaraamat" :I) (CAT NOUN :E) (CASE {CASE} :E)
   (NUMBER {NUMBER} :E)))
 (DESCRIBER
  ((CAT ADJ :I) (LEX "mahukas" :I) (NUMBER {HEAD NUMBER} :E)
   (CASE GENITIVE :E)))
 (CLASSIFIER
  ((CAT PROPER :I)
   (HEAD
    ((LEX "Oxford" :I) (CAT NOUN :E) (CASE {CLASSIFIER CASE} :E)
     (NUMBER {CLASSIFIER NUMBER} :E)))
   (CASE GENITIVE :E) (PATTERN (DOTS HEAD) :E) (NUMBER SINGULAR :E)
   (CLASSIFIER NONE :E) (POSSESSOR NONE :E) (DESCRIBER NONE :E))
  *DONE*)
 (PATTERN (DOTS DESCRIBER DOTS CLASSIFIER HEAD) :E)
 (NUMBER SINGULAR :E) (POSSESSOR NONE :E))
```

Näide 4. "mahukaga sõnaraamatuta"

omadussõnaline täiend peab ühilduma põhjaga

```
((cat common)
            (case abessive)
            (head ((lex "sõnaraamat")))
            (describer  ((cat adj)
                         (lex "mahukas")
                         (case comitative))))


:FAIL
```

Näide 5. "uudiste BNS"
pärisnimi ei luba klassifikaatorit

```
((cat proper)
 (head ((lex "BNS")))
 (classifier  ((cat common)
               (head ((lex "uudis")))
               (number plural))))


:FAIL
```

Näide 6. "eduka uudiste agentuuri edutatud töötajatel"

```
((cat common)
            (case adessive)
            (head ((lex "töötaja")
                   (number plural)))
            (describer  ((cat verb)
                         (ending past-participle)
                         (lex "edutama")))
            (possessor ((cat common)
```

```
                        (head ((lex "agentuur")))
                        (describer ((cat adj)
                                    (lex "edukas")))
                        (classifier ((cat common)
                                     (head ((lex "uudis")))


((CAT COMMON :I) (CASE ADESSIVE :I)
 (HEAD
  ((LEX "töötaja" :I) (NUMBER PLURAL :I) (CAT NOUN :E)
   (CASE {CASE} :E)))
 (DESCRIBER
  ((CAT VERB :I) (ENDING PAST-PARTICIPLE :I) (LEX "edutama" :I)))
 (POSSESSOR
  ((CAT COMMON :I)
   (HEAD
    ((LEX "agentuur" :I) (CAT NOUN :E) (CASE {POSSESSOR CASE} :E)
     (NUMBER {POSSESSOR NUMBER} :E)))
   (DESCRIBER
    ((CAT ADJ :I) (LEX "edukas" :I) (NUMBER {POSSESSOR HEAD NUMBER} :E)
     (CASE {POSSESSOR HEAD CASE} :E)))
   (CLASSIFIER
    ((CAT COMMON :I)
     (HEAD
      ((LEX "uudis" :I) (CAT NOUN :E)
       (CASE {POSSESSOR CLASSIFIER CASE} :E)
       (NUMBER {POSSESSOR CLASSIFIER NUMBER} :E)))
     (NUMBER PLURAL :I) (CASE GENITIVE :E) (PATTERN (DOTS HEAD) :E)
     (POSSESSOR NONE :E) (DESCRIBER NONE :E) (CLASSIFIER NONE :E))
    *DONE*)
   (CASE GENITIVE :E)
   (PATTERN (DOTS DESCRIBER DOTS CLASSIFIER HEAD) :E)
   (NUMBER SINGULAR :E) (POSSESSOR NONE :E))
  *DONE*)
```

```
 (PATTERN (POSSESSOR DOTS DESCRIBER DOTS HEAD) :E) (NUMBER PLURAL :E)
 (CLASSIFIER NONE :E))
```

Näide 7. "Johni venna punases Ferraris"

```
((cat proper)
             (case inessive)
             (head ((lex "Ferrari")))
             (describer ((cat adj)
                         (lex "punane")))
             (possessor ((cat common)
                         (head ((lex "vend")))
                         (possessor ((cat proper)
                                     (head ((lex "John")))))))))
```

```
((CAT PROPER :I) (CASE INESSIVE :I)
 (HEAD
  ((LEX "Ferrari" :I) (CAT NOUN :E) (CASE {CASE} :E)
   (NUMBER {NUMBER} :E)))
 (DESCRIBER
  ((CAT ADJ :I) (LEX "punane" :I) (NUMBER {HEAD NUMBER} :E)
   (CASE {HEAD CASE} :E)))
 (POSSESSOR
  ((CAT COMMON :I)
   (HEAD
    ((LEX "vend" :I) (CAT NOUN :E) (CASE {POSSESSOR CASE} :E)
     (NUMBER {POSSESSOR NUMBER} :E)))
   (POSSESSOR
    ((CAT PROPER :I)
     (HEAD
      ((LEX "John" :I) (CAT NOUN :E)
       (CASE {POSSESSOR POSSESSOR CASE} :E)
       (NUMBER {POSSESSOR POSSESSOR NUMBER} :E)))
```

```
    (CASE GENITIVE :E) (PATTERN (DOTS HEAD) :E) (NUMBER SINGULAR :E)
    (CLASSIFIER NONE :E) (POSSESSOR NONE :E) (DESCRIBER NONE :E))
   *DONE*)
  (CASE GENITIVE :E) (PATTERN (POSSESSOR DOTS HEAD) :E)
  (NUMBER SINGULAR :E) (DESCRIBER NONE :E) (CLASSIFIER NONE :E))
 *DONE*)
(PATTERN (POSSESSOR DOTS DESCRIBER DOTS HEAD) :E) (NUMBER SINGULAR :E)
(CLASSIFIER NONE :E))
```

Lisa 4
Hulgafraasi grammatika

```
((cat partitive)
 (pattern (part part-of))

;; seda hetkel ei realiseeri
;; kuna on eesti keeles keerulisem
;; vrdl ‘‘kõik seitse pöialpoissi’’, ‘‘kogu tass teed’’
;; ‘‘kõigi seitsme pöialpoisiga’’

;; (alt all-partitive (:index all)
;;   (((all none)
;;     (total none))
;;    ((total +)
;;     (all ((cat phrase) (lex "kõik"))))))

  (part

;; põhi (arv-, mõõdu- või hulganimisõna) on ainsuses
;; ühildub laiendiga samamoodi
;; nagu omadussõnaline täiend nimisõnafraasi põhjaga
;; määrsõnu ja asesõnu ei vaatle

   ((number singular)
    (:! case-agree)
```

```
    (alt part-cat (:index cat)
  (((cat cardinal))
   ((cat compound-cardinal))
   ((cat common))
   ((cat measure))
   ((cat fraction))))))

;; nimetavas käändes fraasi korral
;; on laiend osastavas käändes
;; muidu fraasi käändes

(part-of ((cat common)
          (alt case (
           (({^2 case} nominative)
            (case partitive))
           ((case {^2 case})))))

;; hulganimisõna korral on laiend mitmuses
;; muidu ainsuses

          (alt number (
           (({^2 part cat} common)
            (number plural))
            ((number singular)))))))
```

Lisa 5
Hulgafraaside unifitseerimine

Lisa 5 sisaldab näiteid hulgafraaside unifitseerimisest. Iga fraasi korral on toodud sisendi funktsionaalne kirjeldus ja unifitseerimise tulemus.

Näide 1. "seitse pöialpoissi"

```
((cat partitive)
 (part ((cat cardinal)
        (lex "seitse")))
 (part-of ((cat common)
           (head === "pöialpoiss")))))
```

```
((CAT PARTITIVE :I)
 (PART
  ((CAT CARDINAL :I) (LEX "seitse" :I) (NUMBER SINGULAR :E)
   (CASE {CASE} :E)))
 (PART-OF
  ((CAT COMMON :I)
   (HEAD
    ((LEX "pöialpoiss" :I) (CAT NOUN :E) (CASE {PART-OF CASE} :E)
     (NUMBER {PART-OF NUMBER} :E)))
   (CASE PARTITIVE :E) (NUMBER SINGULAR :E) (PATTERN (DOTS HEAD) :E)
   (POSSESSOR NONE :E) (DESCRIBER NONE :E) (CLASSIFIER NONE :E))
```

```
  *DONE*)
 (PATTERN (ALL PART PART-OF) :E) (CASE NOMINATIVE :I))
```

Näide 2. "seitsme pöialpoisiga"

```
((cat partitive)
 (case comitative)
 (part ((cat cardinal)
        (lex "seitse")))
 (part-of ((cat common)
           (head === "pöialpoiss"))))
```

```
((CAT PARTITIVE :I) (CASE COMITATIVE :I)
 (PART
  ((CAT CARDINAL :I) (LEX "seitse" :I) (NUMBER SINGULAR :E)
   (CASE GENITIVE :E)))
 (PART-OF
  ((CAT COMMON :I)
   (HEAD
    ((LEX "pöialpoiss" :I) (CAT NOUN :E) (CASE {PART-OF CASE} :E)
     (NUMBER {PART-OF NUMBER} :E)))
   (CASE {CASE} :E) (NUMBER SINGULAR :E) (PATTERN (DOTS HEAD) :E)
   (POSSESSOR NONE :E) (DESCRIBER NONE :E) (CLASSIFIER NONE :E))
  *DONE*)
 (PATTERN (ALL PART PART-OF) :E))
```

Näide 3. "kari lehmi"

```
((cat partitive)
 (part ((cat common)
        (head ((lex "kari"))))))
```

```
 (part-of ((cat common)
           (head ((lex "lehm")))))))


((CAT PARTITIVE :I)
 (PART
  ((CAT COMMON :I)
   (HEAD
    ((LEX "kari" :I) (CAT NOUN :E) (CASE {PART CASE} :E)
     (NUMBER {PART NUMBER} :E)))
   (NUMBER SINGULAR :E) (CASE {CASE} :E) (PATTERN (DOTS HEAD) :E)
   (POSSESSOR NONE :E) (DESCRIBER NONE :E) (CLASSIFIER NONE :E))
  *DONE*)
 (PART-OF
  ((CAT COMMON :I)
   (HEAD
    ((LEX "lehm" :I) (CAT NOUN :E) (CASE {PART-OF CASE} :E)
     (NUMBER {PART-OF NUMBER} :E)))
   (CASE PARTITIVE :E) (NUMBER PLURAL :E) (PATTERN (DOTS HEAD) :E)
   (POSSESSOR NONE :E) (DESCRIBER NONE :E) (CLASSIFIER NONE :E))
  *DONE*)
 (PATTERN (ALL PART PART-OF) :E) (ALL NONE :E) (TOTAL NONE :E)
 (CASE NOMINATIVE :E))
```