

UNIVERZITET U BEOGRADU
FILOLOŠKI FAKULTET

Bojana P. Đorđević

**IZRADA OSNOVA FORMALNE GRAMATIKE
SRPSKOG JEZIKA UPOTREBOM METAGRAMATIKE**

doktorska disertacija

Beograd, 2017

UNIVERSITY OF BELGRADE
FACULTY OF PHILOLOGY

Bojana P. Đorđević

**CONSTRUCTION OF A FORMAL GRAMMAR OF
SERBIAN USING A METAGRAMMAR**

Doctoral Dissertation

Belgrade, 2017

Mentor:

dr Cvetana Krstev, redovni profesor, Univerzitet u Beogradu, Filološki fakultet

Članovi komisije:

dr Gordana Pavlović-Lažetić, redovni profesor, Univerzitet u Beogradu, Matematički fakultet

dr Balša Stipčević, docent, Univerzitet u Beogradu, Filološki fakultet

Datum odbrane: _____

Ovaj rad nikada ne bi ugledao svetlost dana da nije bilo moje mentorke dr Cvetane Krstev i profesora dr Duška Vitasa. Neizmerno im hvala na dobrim idejama, podršci, strpljenju i poverenju koje su mi pružali svih ovih godina.

Posebno hvala Janiku Parmentjeu (Yannick Parmentier) i Simonu Petižanu (Simon Petitjean) sa Univerziteta u Orleanu u Francuskoj, koji su me uveli u formalizme TAG gramatike i metagramatike XMG i dali početni vetar u leđa za rad na razvoju ovih formalizama za srpski jezik.

Veliku zahvalnost dugujem i svim kolegama saborcima iz grupe za jezičke tehnologije JERTEH na saradnji na projektima iz računarske lingvistike, i uopšte, na lepom druženju. Hvala i mojim kolegama sa letnje škole EGG, oni su mi pokazali koliko lingvistika može biti zabavna, kao i to da se istrajnost isplati.

Zahvaljujem se i svojim kolegama i direktorki škole Azbukum, kao i svim svojim studentima koji su me s puno razumevanja podržavali.

Za Acu

Izrada osnova formalne gramatike srpskog jezika upotrebom metagramatike

Rezime

Ovaj rad predstavlja proces izrade osnova FBLTAG gramatike srpskog jezika, a zatim i proces izrade njene metagramatike, čija primena na osnovne modele rečenica u srpskom jeziku omogućava njihovu automatsku sintaksičku analizu.

Prvo poglavlje rada daje uvod u polje obrade prirodnih jezika, navodeći istorijat discipline i njene podgrane. Veća pažnja posvećuje se automatskoj obradi srpskog jezika, gde se daje pregled dosad ostvarenih rezultata, počevši od analize fonetike i fonologije, pa sve do analize sintakse. Za svaku od navedenih sfera navode se i konkretni alati i resursi koji su dosad razvijeni za srpski jezik.

Drugo poglavlje donosi pregled pojma formalna gramatika, da bi se zatim usmerilo na oblast unifikacionih gramatika kao modela koji čini okvir ovog rada. U nastavku se detaljno iznosi struktura unifikacione gramatike koja će u nastavku rada biti primenjena na srpski jezik – FBLTAG. U drugom delu poglavlja uvodi se pojam metagramatike, kao i konkretne metagramatike koja se u ovom radu koristi za sažeto predstavljanje gramatike FBLTAG – XMG. XMG se obrađuje detaljno, pri čemu se opisuje njegova struktura, namena i princip funkcionisanja, kao i perspektive za primenu na srpski jezik.

Treće poglavlje predstavlja prvo ključno poglavlje rada, kako se u njemu iznose osnove FBLTAG gramatike srpskog jezika nazvane SrpTAG. Polazi se od izbora morfosintaksičkog opisa, kao i izbora deskriptivne gramatike srpskog jezika koja se koristi kao osnov pri izradi formalne gramatike. Kroz strukturalna poređenja između FBLTAG gramatike i deskriptivne gramatike srpskog jezika vrše se i izbori za izradu same SrpTAG. Nakon toga se navode konkretne analize delova rečenične strukture – predikata i glagolskih argumenata. U poslednjem delu poglavlja je data detaljna struktura rečeničnih modela unutar SrpTAG, prema modelima koji su definisani u izabranoj deskriptivnoj gramatici.

U četvrtom poglavlju je predstavljena metagramatika definisana za SrpTAG, kao mehanizam koji olakšava izradu SrpTAG, njeno nadograđivanje, izmenu i primenu. Osnov ovog poglavlja su rezultati trećeg poglavlja, pri čemu se detaljno, po klasama i rečeničnim

modelima, formira apstrahovan i sažet pregled SrpTAG, koji će se dalje koristiti za direktnu primenu prilikom automatske obrade srpskog jezika.

Peto poglavlje se bavi načinom na koji je moguća primena metagramatike napravljene u četvrtom poglavlju. S više detalja predstavljaju se leksikoni koji su neophodan deo automatske obrade, a zatim se u delu o evaluaciji daju konkretni rezultati dobijeni pri automatskoj obradi srpskog jezika primenom metagramatike napravljene u četvrtom poglavlju.

U šestom poglavlju daje se zaključak rada, prikazuju njegovi generalni i spacificni rezultati, i daju perspektive za dalji rad.

Poslednje, sedmo poglavlje sadrži nekoliko dodataka: sintaksički i morfološki leksikon koji se trenutno koriste za automatsku obradu teksta, shemu (DTD) koja se koristi za definisanje leksikona i pregled modela zavisnih rečenica kakav je ponuđen u deskriptivnoj gramatici srpskog jezika.

Možemo zaključiti da je rad ispunio svoju svrhu – na osnovu metagramatike napravljene u ovom radu moguće je generisati formalnu gramatiku za srpski jezik. Ova gramatika obrađuje osnovne strukture srpskog jezika, onako kako su definisane u rečeničnim modelima korišćene deskriptivne gramatike, kao i njihove osnovne transformacije (negacija, pitanje, pasiv, obezličene forme). Napravljena gramatika se može koristiti za prepoznavanje i analizu teksta, ali se potencijalno može koristiti i za generisanje teksta. U svom trenutnom stanju ona ispravno generiše i prepoznaje obrađene strukture srpskog jezika, ali i uspešno odbacuje negramatične realizacije obrađenih struktura.

Ključne reči: formalna gramatika srpskog jezika, FBLTAG, SrpTAG, XMG, metagramatika, parsiranje, automatska sintaksička analiza, obrada prirodnih jezika

Naučna oblast: računarska lingvistika

Uža naučna oblast: obrada prirodnih jezika

UDK broj: _____

Construction of a Formal Grammar of Serbian Using a Metagrammar

Abstract

This paper presents the process of creating the basis of FBLTAG grammar of the Serbian language, followed by the process of building its metagrammar, whose application on basic sentence models of Serbian allows for their automatic syntactic analysis.

The first chapter of the thesis gives an introduction to the field of natural language processing by outlining the history of the discipline and its subfields. The chapter focuses on automatic processing of the Serbian language, providing an overview of the results achieved so far, spanning from the analysis of phonetics and phonology to parsing. Tools and resources that have been developed for the Serbian language are listed for each of the stated fields.

The second chapter offers an overview of the concept of formal grammar, only to focus on the area of unification grammars as a framework for the thesis. The structure of the unification grammar that will later be applied to Serbian – FBLTAG – is presented in detail. The second part of the chapter introduces the concept of metagrammar, as well as the specific metagrammar – XMG – used in the thesis in order to describe FBLTAG in an abstract way. XMG is presented in detail, through describing its structure, purpose and its principles, as well as prospects for its use on the Serbian language.

The third chapter is the key chapter of the thesis since it presents the basics of FBLTAG grammar for Serbian named SrpTAG. The chapter starts with the choice of the morphosyntactic description, as well as the choice of the descriptive grammar of Serbian language, used as a guide for the development of the formal grammar. Through structural comparisons between FBLTAG and the descriptive grammar of Serbian language, choices are made for the construction of SrpTAG. This part is followed by detailed analyses of specific parts of the sentence structure – predicate and verbal arguments. The final part of the chapter gives a detailed structure of sentence models within SrpTAG, according to the models defined in the selected descriptive grammar.

The fourth chapter presents the metagrammar defined for SrpTAG, as a mechanism that facilitates the creation of SrpTAG, as well as its upgrading, modification, and implementation. The basis of this chapter are the results of the third chapter, where the

abstracted and concise overview of SrpTAG is built through detailed classes and sentence models, which will continue to be used for direct application in automatic processing of Serbian language.

The fifth chapter deals with the way in which the application of metagrammar constructed in the fourth chapter is possible. Lexicons needed for the automatic processing are presented in more detail, while the section on evaluation provides concrete results obtained during parsing of Serbian using the given metagrammar.

The sixth chapter gives the conclusion of the thesis, showing its general and specific results, and provide perspectives for future work.

The thesis contains a few appendices – morphological and syntactic lexicon that are currently used for parsing, DTD used to define the lexicons, and an overview of dependent clause model as offered in the mentioned descriptive grammar of the Serbian language.

We can conclude that the thesis has fulfilled its purpose – based on the metagrammar made in this thesis it is possible to generate a formal grammar for Serbian language. This grammar handles basic structures of Serbian, as defined in the sentence models used in the descriptive grammar, as well as their basic transformations (negation, question, passive, depersonalized forms). The constructed grammar can be used to identify and analyze text, but can potentially also be used to generate it. In its current state, it correctly generates and recognizes the mentioned structures of Serbian language, but also successfully rejects ungrammatical realizations of the given structures.

Key words: formal grammar of Serbian, FBLTAG, SrpTAG, XMG, metagrammar, parsing, automatic syntax analysis, NLP

Scientific field: computational linguistics

Scientific area: natural language processing

UDC number: _____

Sadržaj

1	UVOD	1
1.1	Obrada prirodnih jezika.....	2
1.2	Automatska obrada srpskog jezika – resursi i alati	8
1.2.1	Korpusi srpskog jezika	8
1.2.2	Elektronski rečnici i MSO za srpski jezik	10
1.2.3	Obeležavanje teksta i tageri	14
1.2.4	Plitka sintaksička analiza i srpski jezik.....	15
1.2.5	Ostali alati.....	16
2	FORMALNE GRAMATIKE	18
2.1	Definicija formalne gramatike.....	19
2.2	Formalna gramatika i prirodni jezik.....	21
2.3	Unifikacione gramatike	28
2.4	Gramatika adjungovanih stabala (TAG)	35
2.4.1	Definicija gramatike	35
2.4.2	Stabla u TAG	37
2.4.3	Operacije u TAG.....	39
2.4.4	Leksička pravila i porodice stabala	43
2.4.5	Tipovi TAG.....	46
2.4.5.1	FBLTAG.....	48
2.4.6	Relevantnost FBLTAG za obradu prirodnih jezika.....	52
2.5	Metagramatika	54
2.5.1	XMG.....	55
2.5.1.1	Teorijski principi	55
2.5.1.2	Definicija metagramatike	57
2.5.1.3	Mehanizam nasleđivanja	59
2.5.1.4	Dodavanje specifičnih ograničenja	60
2.5.1.5	Parsiranje TAG pomoću metagramatike.....	62
3	IZRADA OSNOVA FORMALNE GRAMATIKE SRPSKOG JEZIKA	66

3.1	Morfosintaksički opis.....	68
3.2	O izboru deskriptivne gramatike srpskog jezika	74
3.3	Elementarno stablo i rečenica srpskog jezika.....	76
3.4	Pojam i sastav predikata.....	87
3.4.1	Modalni i fazni glagoli u TAG	91
3.4.2	Kopulativni i semikopulativni glagoli u TAG	91
3.4.3	Pomoćni glagoli u TAG.....	93
3.4.4	Pasivni glagoli u TAG.....	95
3.4.5	Negacija	96
3.4.6	Refleksivni glagoli	97
3.5	Glagolski argumenti	101
3.5.1	Realizacija argumenata glagola.....	101
3.5.2	Funkcije glagolskih argumenata.....	111
3.5.2.1	Gramatički subjekat.....	111
3.5.2.2	Logički subjekat.....	113
3.5.2.3	Dopune.....	114
3.6	Rečenični modeli	117
3.6.1	Leksička pravila	122
3.6.1.1	Pravila redistribucije.....	122
3.6.1.2	Pravila izmeštanja	127
3.6.1.3	Pravila realizacije	130
3.6.2	Subjekatsko-predikatski modeli.....	133
3.6.3	Bezlični modeli.....	139
3.7	Neobrađeni slučajevi.....	141
4	METAGRAMATIKA ZA SRPTAG.....	143
4.1	Metodologija.....	144
4.2	Jezik za opis metagramatike	148
4.3	Generalizovanje rečeničnih segmenata u klase i organizovanje klasa u funkcije	155
4.3.1	Glagoli	156
4.3.2	Glagolski argumenti.....	165

4.3.2.1	Subjekat.....	165
4.3.2.2	Nesubjekatski argumenti.....	171
4.4	Definisanje klasa koje opisuju promenu dijateze i porodice stabala	184
4.4.1	Subjekatsko-predikatski modeli.....	184
4.4.1.1	Klase prvog rečeničnog modela	184
4.4.1.2	Klase drugog i trećeg rečeničnog modela.....	185
4.4.1.3	Klase četvrtog rečeničnog modela	186
4.4.1.4	Klase petog rečeničnog modela.....	187
4.4.1.5	Klase šestog rečeničnog modela	188
4.4.1.6	Klase sedmog rečeničnog modela	189
4.4.1.7	Klase osmog rečeničnog modela.....	189
4.4.1.8	Klase devetog rečeničnog modela.....	189
4.4.1.9	Klase desetog rečeničnog modela	190
4.4.2	Bezlični modeli	190
4.4.2.1	Klase prvog bezličnog modela	190
4.4.2.2	Klase drugog bezličnog modela.....	191
4.4.2.3	Klase trećeg bezličnog modela	191
4.4.2.4	Klase četvrtog bezličnog modela	191
4.5	Dodatne klase	191
5	PRIMENA METAGRAMATIKE	196
5.1	Leksikoni	197
5.2	Evalvacija.....	204
6	ZAKLJUČAK I DALJI RAD	210
6.1	Zaključak.....	211
6.2	Dalji rad.....	212
7	LITERATURA.....	214
7.1	Korišćena literatura.....	215
7.2	Referentne veb lokacije.....	223
8	PRILOZI.....	225
8.1	DTD za leksikon lema i morfološki leksikon.....	226

8.2	Leksikon lema.....	228
8.3	Morfološki leksikon.....	240
8.4	Pregled tipova zavisnih rečenica i njihovih konstituentских vrednosti.....	280
9	BIOGRAFIJA AUTORA.....	282
10	IZJAVE O DISERTACIJI.....	284

1 UVOD

1.1 Obrada prirodnih jezika

Ovaj rad započinjemo kratkim uvodom u obradu prirodnih jezika¹. Termin *obrada prirodnih jezika* (nadalje OPJ) ćemo koristiti u najširem mogućem značenju da označimo obradu pisanog ili govornog prirodnog jezika pomoću računara, ali ćemo svejedno na ovom mestu detaljnije objasniti svaki od njegovih segmenata.

Najpre, pod terminom *prirodni jezik* podrazumeva se svaki ljudski jezik nastao prirodnim procesom. Nasuprot njemu stoji termin veštački jezik, koji s jedne strane označava takozvani međunarodni ili pomoćni jezik, veštački *lingua franca* stvoren za komunikaciju među ljudima koji govore različitim jezicima. Primer takvog jezika je esperanto. S druge strane, veštački jezik je i bilo koji programski jezik (Java ili C++...), koji se koristi za komunikaciju između računara i čoveka.

Sledeća stavka termina OPJ je *obrada prirodnog jezika*. Obrada prirodnog jezika podrazumeva jezičku analizu, ali isto tako i generisanje jezika, odnosno određenog segmenta jezika. Obrada se može vršiti na svim jezičkim nivoima – počevši od fonetike i fonologije (prepoznavanje audio signala i raspoznavanje reči od kojih se taj signal sastoji, kao i generisanje audio signala od niza reči), preko morfologije (prepoznavanje i generisanje odgovarajućih formi reči zavisno od konteksta) i sintakse (prepoznavanje i generisanje struktura koje se sastoje od grupa reči u određenom redosledu), do semantike (prepoznavanja značenje reči i njihovih kombinacija), pragmatike (prepoznavanje i generisanje jezika s odgovarajućom komunikacijskom svrhom) i nivoa diskursa (prepoznavanje i generisanje kohezivnog i koherentnog teksta). Tako aplikacije za proveravanje pravopisa u programima za obradu teksta rade na nivou fonologije i morfologije, uzimajući u obzir i segmente sintakse. Aplikacije koje se koriste za obeležavanje vrsta reči rade na morfosintaksičkom nivou. Na sintaksičkom nivou rade programi koji mogu vršiti analizu na „plitkom“ nivou, uzimajući u obzir samo segmente rečenica (npr. sintagme, složene glagolske oblike, vremenski izraze), i na „dubokom“ nivou na kome se vrši analiza strukture čitave rečenice.

¹ Neki naučnici kao hiperonime za ovu oblast koriste termine *računarska lingvistika* (Mitkov, 2005) ili *obrada govora i jezika* (Jurafsky & Martin, 2000)

Potreba da se jezička obrada prepusti računaru prvi put se javila paralelno sa prvim računarima – sredinom prošlog veka. Ti napori su u početku bili usmereni na automatsko tj. mašinsko prevođenje i bili su direktno vezani za tadašnje političke prilike u svetu – potrebu da se u vreme hladnog rata na brz način i automatski prevodi između ruskog i engleskog. Nakon memoranduma Vorena Vivera (*Warren Weaver*) iz 1949. godine, u kome se iznosi stav da bi mašinsko prevođenje moglo biti moguće, napori počinju da se usmeravaju na tu stranu. Prvi zvanični sistem za mašinsko prevođenje, Džordžtaun-IBM sistem, predstavljen 1954. godine², sastojao se od 250 reči i 6 pravila, dovoljnih za prevod 60 odabranih rečenica sa ruskog na engleski jezik. Presentacija ovog događaja izazvala je značajnu pažnju u javnosti i ostavila jak utisak, kako među laicima tako i među naučnicima, da će u relativno kratkom vremenskom periodu biti napravljen sistem koji će davati visokokvalitetan prevod. Narednih deset godina značajne sume novca su ulagane u razvoj mašinskog prevođenja, međutim, nakon deset godina, rezultati izveštaja o napretku na tom polju bili su razočaravajući. Prvi problemi na koje su naišli naučnici baveći se mašinskim prevođenjem jesu problemi koji su gorući i dan danas, i s jedne strane se mogu pripisati problemu dvosmislenosti u jeziku, kao posledici homomorfности i idiomatičnosti, a s druge strane strukturnim razlikama među jezicima, uključujući i red reči.

Ovaj događaj je preusmerio pažnju naučnika sa mašinskog prevođenja na druge sfere u kojima su zajednički napori programera i lingvиста mogli da daju značajnije rezultate. Oblast OPJ je danas veoma razgranata i sastoji se od velikog broja postupaka za obradu jezika i različitih disciplina, od kojih su neke, pored već pomenutog mašinskog prevođenja, i pronalaženje informacija, ekstrakcija informacija, odgovaranje na pitanja, sumarizacija teksta, ekstrakcija termina, automatsko indeksiranje teksta, učenje jezika uz pomoć računara, prepoznavanje i generisanje govora, i mašinsko učenje.

Generalno posmatrano, pristup OPJ može biti statistički ili zasnovan na pravilima. Statistički pristupi podrazumevaju automatsko uočavanje obrazaca u strukturama na različitim nivoima analize i izođenje pravila na osnovu tih obrazaca. Za taj proces obično je potrebno postojanje određenog korpusa materijala koji se koristi za obuku sistema. Dati sistem nadalje na osnovu naučenih obrazaca može sam vršiti odgovarajući tip analize. Ovi

² Za detaljan pregled javnog predstavljanja Džordžtaun-IBM sistema 1954. godine pogledati (Hutchins, 2005)

sistemi minimizuju ljudski, ručni doprinos analizi, iako nikada ne mogu biti potpuno samostalni i zahtevaju proveru u finalnim fazama procesa odlučivanja. Pristupi zasnovani na pravilima najčešće podrazumevaju postojanje tima ljudi koji izrađuje pravila za određeni nivo jezičke strukture, a na osnovu kojih se zatim vrši jezička analiza na tekstu. Ovaj pristup je temeljniji, precizniji, i tek se zahvaljujući njemu mogu definisati npr. odnosi među nesusednim delovima rečenice. Ova dva sistema se često kombinuju kako bi se poboljšao učinak oba pristupa, dajući takozvani hibridni metod analize.

Ovo nas dovodi do sledećeg segmenta koji je od ključnog značaja za OPJ – resursi. Temelj automatske jezičke analize čine tri vrste resursa: korpus, rečnici i gramatike. Neophodni alati potrebni za obradu ovih resursa jesu programi za obeležavanje teksta (tageri), koji se koriste za obeležavanje morfosintaksičkih informacija u tekstu, i programi za sintaksičku analizu (parseri) koji se koriste za prepoznavanje i obeležavanje struktura različitog nivoa. Pogledajmo svaku od pomenutih stavki s malo više pažnje.

Uopšteno govoreći, **korpus** predstavlja kolekciju tekstova u elektronskoj formi koja je reprezentativna po svojoj veličini, tematici tekstova, funkcionalnom stilu i vremenskom periodu u kome su tekstovi nastali. To šta je reprezentativno za određeni korpus zavisi od namene korpusa³. Korpusi mogu biti jednojezični i dvojezični ili višejezični. Višejezični korpusi se obično sastoje od tekstova od kojih jedan predstavlja original, a drugi njegov prevod, pri čemu je za različita jezička istraživanja od posebne koristi to da tekstovi budu međusobno upareni na nivou rečenice.

Korpus se može sastojati od tekstova u sirovom obliku, odnosno tekstova koji ni na koji način nisu obeleženi. Međutim, da bi korpus mogao efikasno da se koristi – kako za pretraživanje i ekstrakciju informacija, tako i za statističku obradu, potrebno je obeležiti (anotirati, etiketirati) tekst. Tekstovi se obeležavaju primenom programa za obeležavanje teksta ili tagera.

Tageri su programi koji vrše obeležavanje određenih informacija u tekstu. Obeležavanje podrazumeva dodavanje informacija različitog tipa, od opštih bibliografskih informacija kao što su autor, funkcionalni stil ili informacije o procesu digitalizacije teksta,

³ Za podrobnije raščlanjenje definicije korpusa i pojma reprezentativnosti korpusa videti (Utvić, 2013, pp. 14–23)

preko informacija o logičkoj strukturi teksta – početku i kraju poglavlja, paragrafa i rečenica, do jezičkih informacija, kao što su morfosintaksičke kategorije reči u tekstu, njihova funkcija ili značenje. Pod obeležavanjem korpusa u terminima OPJ se ipak najčešće podrazumeva obeležavanje morfosintaksičkim informacijama⁴, koje se može vršiti ili na osnovu informacija u rečnicima, o kojima će biti reči nešto niže, ili na osnovu statističkih metoda. Obeležavanje korpusa se u nedostatku rečnika u početnoj fazi obrade vrši ručno, što postaje materijal na kome se tageri koji vrše statističko obeležavanje teksta obučavaju.

U procesu obeležavanja, rečima⁵ u korpusu se pridodaju informacije na osnovu odedenog tagseta, odnosno standarda za morfosintaksički opis (MSO).

Tagset predstavlja skup svih oznaka (tagova, anotacija ili etiketa) koje se u procesu obeležavanja morfosintaksičkim (i semantičkim) informacijama može dodeliti rečima jednog jezika. Način na koji je svaka oznaka unutar tagseta organizovana se može zamisliti kao šablon po kome se za reči jednog jezika definišu informacije koje su od značaja za njihovo identifikovanje. Taj šablon tipično polazi od tekstualne reči – forme reči koja se pojavljuje u tekstu, a koja se zatim vezuje za lemu reči, odnosno njen osnovni oblik (nominativ jednine za imenice, infinitiv za glagole...). Šablon može da sadrži i morfosintaksičke, a ponekad i semantičke karakteristike reči. Informacije o kategorijama i karakteristikama reči u tagsetu su po pravilu kodirane – obeležene slovom, brojem ili skraćenicom. Oznake koje se javljaju u tagsetu mogu biti intuitivne (m – muški rod, 3 – treće lice), ali ne moraju (x – prvo lice, y – drugo lice, z – treće lice), i zato se vrlo često uz tagset prilažu i uputstva za dekodiranje njihovog sadržaja. Imajući sve ovo u vidu, može se reći da tagset predstavlja standard za rečnički unosak, koji se doduše od standardnog rečničkog unosa po mnogo čemu razlikuje. Ovo nas dalje vodi na temu rečnika, ili preciznije rečeno – elektronskog rečnika.

Rečnici o kojima se govori u domenu OPJ su takozvani **elektronski** ili **mašinski čitljivi rečnici**. Ovo ime, pored toga što treba da naznači da se rečnici nalaze u elektronskom formatu, koristi se da bi se naznačilo da se razlikuju od rečnika namenjenih

⁴ engl. *Part-of-Speech tagging* ili *PoS tagging*.

⁵ Napominjeno da u ovom kontekstu (i nadalje u tekstu) pod pojmom reči podrazumevamo takozvanu *korpusnu reč* – nisku koja se sastoji isključivo od slovnih karaktera. U skladu sa tom definicijom, reč *veb-sajt* se sastoji od dve korpusne reči.

ljudskoj upotrebi. Naime, ove rečnike čitaju, odnosno koriste, skoro isključivo računari. Druga značajna karakteristika elektronskih rečnika jeste što su to rečnici upotrebe, i u najjednostavnijem slučaju rečnici teksta, odnosno određenog korpusa. Objasnimo šta pod time podrazumevamo. Dok je osnovna jedinica tradicionalnog rečničkog unoska osnovni oblik reči ili lema, i dok se uz nju najčešće navode samo nepravilne forme te reči, u elektronskom rečniku je osnovna jedinica forma reči, i to, zavisno od rečnika, ne samo jedna forma reči, već ili sve njene forme ili one koje se pojavljuju u određenom tekstu, odnosno korpusu. Uz formu reči se dalje definiše njena lema, a onda i sve ostale morfosintaksičke, a ponekad i semantičke, karakteristike – kako same leme, tako i konkretne forme reči.

Rečnicima se, pored tagera, koriste i gramatike, kojima se mogu obeležiti strukture različitog nivoa složenosti. Ovde ćemo govoriti o dva osnovna tipa gramatika – gramatikama koje se koriste za plitku sintaksičku analizu (tzv. *plitko parsiranje*) i gramatikama za opsežnu sintaksičku analizu (tzv. *duboko parsiranje*).

Plitka sintaksička analiza je proces u kome se u tekstu prepoznaju određeni, ali ne nužno i najčešće ne svi segmenti rečenice. To su mahom niske reči, u čiju se strukturu i funkciju u rečenici ne ulazi. Zadatak plitke sintaksičke analize podrazumeva tzv. *komadanje* rečenice⁶, čime se ona razdvaja na segmente koji se često mogu poistovetiti sa sintagmama. Obrasci koji se koriste za prepoznavanje ovih rečeničnih segmenata mogu se prikazati pomoću regularnih izraza – niske simbola i karaktera koja predstavlja nisku ili šablon koji se traži. Regularni izrazi su sasvim zadovoljavajući za ovaj zadatak ukoliko se traže segmenti koji se sastoje od dve ili tri reči, za čiji odnos je potrebno postaviti minimum uslova. Međutim, za analizu komplikovanijih segmenata koji se sastoje od više reči čiji je odnos potrebno precizno definisati koristi se predstava u vidu konačnih automata (ili konačnih transduktora ukoliko automat pored prepoznavanja generiše i odgovarajući izlaz). Konačni automat je graf koji se sastoji od početnog stanja, određenog broja međustanja i završnog stanja. Stanja su međusobno povezana lukovima, koji predstavljaju uslove za prelazak u novo stanje. Jedan segment teksta se smatra prepoznatim ukoliko graf iz početnog stanja, čitajući reči unutar segmenta, dođe do završnog stanja i istovremeno prepozna sve reči u redosledu i obliku u kome se javljaju u datom segmentu. Ovako

⁶ engl. *chunking*.

prepoznati segmenti mogu se zatim iskoristiti u procesu duboke sintaksičke analize. Niz grafova konstruisanih za određeni zadatak organizovan je u lokalne gramatike. Unutar lokalne gramatike grafovi mogu biti organizovani u takozvane *kaskade*. Kaskada podrazumeva organizovanje grafova prema prioritetu, pri čemu se oni koji prepoznaju jednostavnije i nedvosmislenije jedinice primenjuju prvi nad tekstom, da bi se na njihovim rezultatima – određenoj leksičkoj oznaci koja se unosi oko prepoznatog izraza, primenili složeniji grafovi. Ovo je metod koji značajno povećava preciznost i brzinu sistema, smanjujući pritom i dvosmislenost.

Pored zadatka plitke sintaksičke analize, konačni automati se koriste i u procesu pretraživanja i ekstrakcije informacija. Ovi zadaci najčešće podrazumevaju dve glavne oblasti: 1) Prepoznavanje imenovanih entiteta – prepoznavanje vlastitih imena (ljudi, organizacija, lokacija), vremenskih izraza (datum, vreme, vremenski intervali) i numeričkih izraza (izraza za količinu i meru, procenata, novčanih vrednosti, kardinalnih brojeva) i 2) Ekstrakciju termina – pronalaženje termina u korpusu koji su specifični za određenu oblast.

Duboka sintaksička analiza podrazumeva raščlanjivanje rečenice na njene sastavne delove, uz određivanje njihove funkcije u rečenici i njihove interne strukture. Gramatike koje vrše ovakvu analizu su obično vrlo opsežne i prave se s težnjom da pokriju što veći broj struktura u jeziku. Da bi se napravila razlika između ovih gramatika i gramatika namenjenih ljudskoj upotrebi, često se za ovaj tip gramatika koristi naziv *formalna gramatika*, što je naziv koji ćemo i mi koristiti u ovom radu. Kao i elektronski rečnici, i ovaj tip gramatika je napravljen za računar, a naziv *formalna* između ostalog dolazi i od metoda izrade ove gramatike, koji se zasniva na principima logike i matematike. Ovakve gramatike mogu biti ručno pisane, što je zadatak koji zahteva puno truda i vremena, ili statistički izvedene na osnovu obrazaca pojavljivanja koje odgovarajući program – parser – uoči na obeleženom tekstu. I pored količine posla koju ručno pisanje formalne gramatike zahteva, ovakva gramatika je uvek preciznija, temeljnija i najčešće kvalitetnija od gramatike nastale statističkim putem, iako je po opsegu često značajno manja. Ručno pisana gramatika ovog tipa je tema ovog rada i o njoj će biti više reči počevši od poglavlja 2.

1.2 Automatska obrada srpskog jezika – resursi i alati

U ovom poglavlju ćemo dati pregled resursa i alata razvijenih za srpski jezik, a koji su od direktnog ili indirektnog značaja za razvoj i primenu formalne gramatike koju opisujemo u ovom radu. Treba napomenuti da dok za srpski jezik postoji značajna podrška u smislu resursa koji omogućavaju morfosintaksičku i plitku sintaksičku analizu, gramatika koja je tema ovog rada predstavlja pionirski rad na izradi sistema koji omogućava potpunu ili duboku sintaksičku analizu.

1.2.1 Korpusi srpskog jezika

Prvi korpus na ovim prostorima, napravljen sa ciljem razvoja jezičkih tehnologija za prepoznavanje govora i mašinsko prevođenje (Dobrić, 2012) je dijahronijski korpus srpskog jezika Đorđa Kostića (A. Kostić, 2003), čije je pravljenje započeto 1957. godine. Ovaj korpus sadrži tekstove počevši od 12. veka pa sve do trenutka trajanja projekta, 1962. godine, i sastoji se od oko 11 miliona reči. Digitalizovan je i ručno lematizovan. Danas se ovaj korpus koristi primarno u svrhe kvantitativnog opisa srpskog jezika (Đ. Kostić, 2001).

Trenutno najveći korpus savremenog srpskog jezika, *SrpKor* (Krstev & Vitas, 2005; Utvić, 2013), razvijen je na Matematičkom fakultetu u Beogradu. Njegova izrada je počela 1981. godine, a njegovo dopunjavanje traje i danas. Ovaj korpus sadrži književne i umetničke tekstove pisaca 20. i 21. veka, naučne tekstove iz oblasti društvenih i prirodnih nauka, administrativne tekstove i tekstove iz dnevnih novina, časopisa i magazina počevši od 1991. godine. Manji deo korpusa čine prevodi književnih, umetničkih i novinskih tekstova. Njegova trenutna veličina je 122 miliona reči. Tekstovi u korpusu su obeleženi bibliografski i morfološki. Bibliografska oznake nose informacije o autoru, registru teksta, kao i izvornom jeziku teksta, dok morfološke oznake podrazumevaju da je svakoj reči pridružena lema i njena vrsta reči. Ovaj korpus je slobodno dostupan na vebu za korišćenje u naučne svrhe.

Od manjih korpusa mogu se pomenuti *Rastko projekat*, započet 1997. godine, koji sadrži kolekciju tekstova počevši od srednjovekovnih pa sve do savremenih tekstova. Tu je i elektronski korpus dela Laze Kostića, čija je izrada započeta 2009. godine pod okriljem

Matice srpske. Na kraju ćemo pomenuti i korpus *Antologija srpske književnosti*, razvijen na Učiteljskom fakultetu u Beogradu. Ovaj korpus se sastoji od 130 dela stare i savremene književnosti.

Što se paralelnih korpusa tiče, može se pomenuti šest korpusa. Jedan od njih je paralelizovani francusko-srpski korpus *SrpFrankKor* (Vitas & Krstev, 2006; Vitas, Krstev, & Laporte, 2006) sastavljen od književnih i novinskih tekstova, koji sadrži nešto manje od milion reči po jeziku. Tu je zatim i englesko-srpski korpus *SELFEH*, koji se sastoji od književnih, novinskih i tekstova iz domena ekonomije, prava i zdravstva, sa milion reči po jeziku, kao i srpsko-engleski korpus *SrpEngKor* (Krstev & Vitas, 2009), sa tekstovima iz domena beletristike, novina, naučnih časopisa, zdravstva, prava, udžbenika i filmskih titlova, sa nešto više od dva miliona reči po jeziku. Sva tri korpusa su poravnata na nadrečeničnom nivou, kao i na nivou rečenice. Četvrti korpus je paralelni francusko-englesko-srpski korpus *ParCoLab* (Balvet, Stošić, & Miletić, 2014). Korpus se sastoji od nešto manje od četiri miliona reči za sva tri jezika, i sadrži mahom književna dela originalno napisana na francuskom, a prevedena na preostala dva jezika.

Srpski jezik se nalazi u još dva višejezična korpusa. Jedan od njih je korpus koji se sastoji od originala i prevoda *1984* Džordža Orvela na preko 10 jezika (bugarski, češki, engleski, estonski, francuski, mađarski, rumunski, slovački, slovenački, ukrajinski, persijski, makedonski, poljski, ruski, srpski, litvanski i dijalekat slovenačkog jezika, rezijski), i predstavlja deo projekta MULTEXT-East na izgradnji multilingvalnih resursa (korpusa, rečnika, MSO)⁷. Ovaj korpus se sastoji od oko 100.000 reči i obeležen je kako strukturnim oznakama (poglavlja, paragrafi, rečenice...), tako i morfosintaksičkim oznakama⁸. Na osnovu ovog korpusa izrađen je elektronski rečnik za svaki od jezika, o kome će biti više reči u odeljku o rečnicima srpskog jezika.

Drugi višejezični korpus je takođe korpus sastavljen od originala i prevoda dela, ovoga puta *Puti oko sveta za 80 dana* Žila Verna. Korpus se sastoji od verzija teksta na francuskom i 16 prevoda (engleski, nemački, španski, portugalski, italijanski, rumunski, ruski, srpski, hrvatski, bugarski, makedonski, poljski, mađarski, grčki, danski i slovenački).

⁷ Više informacija o projektu može se naći na: <http://nl.ijs.si/ME/>

⁸ Korpus je dostupan preko platforme Clarin: <https://www.clarin.si/repository/xmlui/handle/11356/1043>

Tekst je uparen između 12 jezika. Korpus je obeležen strukturno i morfološki, primenom elektronskih rečnika (Vitas, Koeva, Krstev, & Obradović, 2008).

1.2.2 Elektronski rečnici i MSO za srpski jezik

Rad na morfološkim elektronskim rečnicima takođe je započet na Matematičkom fakultetu u Beogradu (Krstev, 2008), a njihovo dopunjavanje neprekidno traje. Ovi rečnici se nalaze u takozvanom LADL⁹ formatu, prema imenu laboratorije tvorca rečnika, Morisa Grosa (*Maurice Grosse*), koji se koristi i za niz drugih jezika (francuski, engleski, grčki, portugalski, ruski, arapski, korejski, italijanski, španski, norveški, nemački, poljski i bugarski). Format ovog rečnika podrazumeva postojanje dve izdvojene grupe rečnika – rečnika prostih reči DELAS i njihovih flektivnih oblika DELAF s jedne strane (Tabela 1), i rečnika višečlanih reči¹⁰ DELAC i njihovih flektivnih oblika DELACF (Tabela 2).

Tabela 1 Izgled unoska prostih reči (DELAS) i njihovih oblika (DELAF) u elektronskom rečniku srpskog jezika

DELAS		DELAF
Obrazac	lema.kodKlase[+sinSem]*	oblikReči.lema.vrstaReči[+sinSem]*[:kategorije]*
Primer	devojka.N618+Hum+Ek	devojci,devojka.N+Hum+Ek:fs3v:fs7v
Objašnjenje oznaka	N618: kod imeničke klase Hum: semantička oznaka za ljudsko biće Ek: ekavski izgovor *: nula ili više pojavljivanja	N: imenica f: ženski rod s: singular 3: dativ 7: lokativ v: živo biće

Rečnik DELAS sastoji se od leme, koda flektivne klase, koja predstavlja tip promene date reči, i sintaksičkih, semantičkih i derivacionih svojstava leme. Kod flektivne klase je zapravo ime odgovarajućeg konačnog transduktora koji je izrađen ručno tako da od leme generiše sve njene morfološke oblike. Svi ovi oblici nalaze se u rečniku DELAF. Unosak u

⁹ fran. *Laboratoire d'Automatique Documentaire et Linguistique*

¹⁰ engl. *multi-word units* (MWE)

DELAF-u sastoji se od oblika proste reči i njoj pripadajuće leme, kao i morfosintaksičkih karakteristika leme i datog morfološkog oblika.

Tabela 2 Izgled unoska višečlanih reči (DELAC) i njihovih oblika (DELACF) u elektronskom rečniku srpskog jezika

DELAC	
Obrazac	lemaVišečlaneReči.KodKlaseVišečlaneLeme[+SinSem]* lemaVišečlaneReči = Oblik1(lema1.kodKlase[:kategorije]) oblik2(lema2.kodKlase[:kategorije]) oblik3(lema3.KodKlase[:kategorije]) ...
Primer	drveni(drven.A6:adms1g) duvački(duvački.A2:adms1g) instrument(instrument.N29:ms1q), NC_AXAXN+Conc
Objašnjenje oznaka	A6, A2: kodovi pridevskih klasa N29: kod imeničke klase NC_AXAXN: kod klase višečlane reči a: pozitivan oblik d: određen vid 1: nominativ g: animatnost ne utiče na promenu q: neanimatan Conc: oznaka za konkretan pojam
DELACF	
Obrazac	OblikVišečlaneReči,lemaVišečlaneReči.kodVišečlaneKlase[:kategorije]+
Primer	drvenim duvačkim instrumentom,drveni duvački instrument.NC_AXAXN+Conc:ms6q
Objašnjenje oznaka	6: instrumental

Rečnici DELAC i DELACF prate format rečnika DELAS i DELAC, s tim što u sastav leme i morfološkog oblika, kako su u pitanju višečlane reči, ulaze i karakteri koji se mogu

naći između reči, kao što su praznina, crtica ili apostrof¹¹. Slika 1 daje izvod iz rečnika DELAF za srpski jezik.

```
adut, .N:ms1q
adut, .N:ms4q
aduta, adut.N:mw2q
aduta, adut.N:mw4q
aduta, adut.N:mp2q
aduta, adut.N:ms2q
advokatskom, advokatski.A+PosQ:adms3g
advokatskom, advokatski.A+PosQ:adms7g
advokatskom, advokatski.A+PosQ:adns3g
advokatskom, advokatski.A+PosQ:adns7g
advokatskom, advokatski.A+PosQ:aefs6g
afera, .N:fp2q
afera, .N:fs1q
Africi, Afrika.N+NProp+Top+Kont:fs3q
Africi, Afrika.N+NProp+Top+Kont:fs7q
```

Slika 1 Izvod iz DELAF rečnika srpskog jezika

Pored rečnika opšteg sadržaja, razvijen je i niz specijalnih rečnika, među kojima rečnici vlastitih imena: DELA-Top i DELAC-Top – rečnici prostih i višečlanih geografskih imena, DELA-First i DELA-Last – rečnici imena i prezimena stanovnika Srbije, DELA-EN-First i DELA-EN-Last – rečnici transkribovanih engleskih imena i prezimena, i DELA-Enc – rečnik prostih imena koja spadaju u enciklopedijsko znanje. Ovi rečnici su od posebnog značaja pri obeležavanju imenovanih entiteta.

Rečnik se trenutno sastoji od 140.000 prostih i 18.000 višečlanih reči, pri čemu se taj broj skoro svakodnevno uvećava. Može se koristiti unutar sistema za obradu korpusa Unitex¹² i Nooj¹³. Unitex je istovremeno i platforma u okviru koje je razvijena većina resursa za srpski jezik, kao što će se videti u daljem tekstu.

Drugi elektronski rečnik koji postoji za srpski jezik je rečnik teksta Orvelove 1984 razvijen u okviru MULTEXT-East projekta (Krstev, Vitas, & Erjavec, 2004). Ovaj rečnik sadrži 20.294 unosaka, od toga 16.907 različitih oblika reči, 8.392 lema i 906 MSO. Rečnik u

¹¹ Detaljniji pregled oznaka može se naći u (Đorđević, 2014; Krstev, 2008).

¹² Unitex <http://www-igm.univ-mlv.fr/~unitex/>

¹³ Nooj: <http://www.nooj-association.org>

ovom obliku postoji za sve jezike koji su učestvovali u formiranju korpusa sastavljenog od originala i prevoda pomenutog dela. Slika 2 prikazuje srpsku verziju teksta obeleženu na ovaj način.

```
<s xml:id="Osr.1.2.2.1">
<w xml:id="Osr.1.2.2.1.1" lemma="biti" ana="#Vmps-smn-n---p">Bio</w>
<w xml:id="Osr.1.2.2.1.2" lemma="jesam" ana="#Va-p3s-an-y---p">je</w>
<w xml:id="Osr.1.2.2.1.3" lemma="vedar" ana="#Afpmsnn">vedar</w>
<w xml:id="Osr.1.2.2.1.4" lemma="i" ana="#C-s">i</w>
<w xml:id="Osr.1.2.2.1.5" lemma="hladan" ana="#Afpmsnn">hladan</w>
<w xml:id="Osr.1.2.2.1.6" lemma="aprilski" ana="#Aopmpn">aprilski</w>
<w xml:id="Osr.1.2.2.1.7" lemma="dan" ana="#Ncmsn--n">dan</w>
<c xml:id="Osr.1.2.2.1.8">;</c>
<w xml:id="Osr.1.2.2.1.9" lemma="na" ana="#Spasa">na</w>
<w xml:id="Osr.1.2.2.1.10" lemma="časovnik" ana="#Ncmsa--n">časovnicima</w>
<w xml:id="Osr.1.2.2.1.11" lemma="jesam" ana="#Va-p3s-an-y---p">je</w>
<w xml:id="Osr.1.2.2.1.12" lemma="izbijati" ana="#Vmps-smn-n---e">izbijalo</w>
<w xml:id="Osr.1.2.2.1.13" lemma="trinaest" ana="#Mc---1">trinaest</w>
<c xml:id="Osr.1.2.2.1.14">.</c>
</s>
```

Slika 2 Deo teksta srpske verzije Orvelove 1984 anotiran etiketama u MULTEXT-East formatu

Unosak ovog rečnika (Tabela 3) sastoji se od tekstualnog oblika reči, njegove leme i MSO, koji se nalaze unutar XML etikete¹⁴. Ovaj MSO je pozicioni – unutar svake vrste reči postoji unapred propisan broj pozicija koje zauzimaju morfosintaksička svojstva te reči. Vrednost svakog svojstva obeležava se pojedinačnim karakterom. Kako su svojstva definisana tako da pokrivaju veliki broj strukturno različitih jezika, ne moraju svi jezici imati popunjene sve pozicije unutar MSO – u takvom slučaju na mestu nepostojećeg obeležja stoji samo crtica.

Tabela 3 Izgled unoska u elektronskom rečniku srpskog jezika u MULTEXT-East formatu

MULTEXT-East MSO	
Obrazac	<w xml:id="ID" lemma="konkretnaLema" ana="MSO">oblikReči</w>
Primer	<w xml:id="Osr.1.2.2.1.7" lemma="dan" ana="Ncmsi--n">danom</w>
Objašnjenje oznaka	w i /w: početna i završna XML etiketa za reč xml:id="Osr.1.2.2.1.7": identifikator rečenice (Osr.1.2.2.1) i redni broj reči u rečenici (7)

¹⁴ Detaljan opis MULTEXT-East MSO može se naći na: <http://nl.ijs.si/ME/V4/msd/html/>

	N: imenica c: zajednička imenica m: muški rod s: singular i: instrumental -: ne postoji dato obeležje (u ovom slučaju - određenost, klitika) n: neanimatan
--	--

1.2.3 Obeležavanje teksta i tageri

Ne postoje tageri koji su razvijeni konkretno za primenu na srpskom jeziku, međutim postoji niz tagera koji su dosad korišćeni u ove svrhe.

(Popović, 2010) navodi rezultate evaluacije pet tagera dostupnih u nekomercijalne ili akademske svrhe – TnT¹⁵, Tree Tagger¹⁶, Rule Based Tagger¹⁷, MXPOST¹⁸ i SVMTool¹⁹. Obeležavanje pomoću ovih tagera nije vršeno na osnovu pravila, već zahvaljujući statističkim metodama. Projekat je podrazumevao postojanje obeleženog korpusa na kome je vršena obuka programa, koji su se dalje koristili u svrhe obeležavanja. Korpusi koje je autor koristio, a koji su bili sastavljeni u ove svrhe, bili su obeleženi MULTEXT-East oznakama. Rezultati autora su pokazali da TnT i TreeTagger daju najbolje rezultate, sa 94,39% kao najboljom ocenom performansi u prvom slučaju i 86,18% u drugom slučaju.

U (Utvić, 2011, 2013) mogu se naći postupak i rezultati anotiranja korpusa srpskog jezika SrpKor. Autor navodi da je pri izboru tagera za automatsko obeležavanje tekstova birao između tri programa za anotaciju. Dva su programi koji su dali najbolje rezultate u prethodno pomenutoj evaluaciji, a treći je okruženje Unitex, kao okruženje u okviru koga se mogu direktno koristiti elektronski rečnici srpskog jezika. Izbor autora je na kraju bio TreeTagger, kao jedini od ranije pomenutih tagera koji rečima u korpusu mogu da pridruže i lemu.

¹⁵ <http://www.coli.uni-saarland.de/~thorsten/tnt/>

¹⁶ <http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/>

¹⁷ <http://www.ling.gu.se/~lager/mogul/brill-tagger/>

¹⁸ http://www.inf.ed.ac.uk/resources/nlp/local_doc/MXPOST.html

¹⁹ <http://www.cs.upc.edu/~nlp/SVMTool/>

Još jedan primer tagiranja koji se zasniva na statističkim metodama je onaj predstavljen u (Gesmundo & Samardžić, 2012). Autori navode postignute rezultate prilikom lematizacije teksta na srpskom jeziku korišćenjem programa Btagger²⁰, a koji ukazuju na čak 99,51% tačnosti pri lematizaciji poznatih i 84,98% pri lematizaciji nepoznatih reči.

(Constant, Krstev, & Vitas, 2015) predstavljaju primere obeležavanja imenovanih entiteta i višočlanih reči primenom hibridnog metoda. Ovaj metod podrazumeva unošenje informacija koje dolaze iz postojećih elektronskih rečnika i lokalnih gramatika za srpski jezik u statistički model, obučen na obeleženom korpusu. U zaključku, autori navode da primena leksičkih resursa – rečnika i gramatike – značajno povećava preciznost tagera, koja dostiže 99,09% za neke od primenjenih tehnika ovog metoda.

1.2.4 Plitka sintaksička analiza i srpski jezik

Za srpski je dosad razvijeno nekoliko sistema za pretraživanje i ekstrakciju određene vrste rečeničnih segmenata. Svi sistemi koji će biti pomenuti baziraju se na pravilima definisanim kroz automate konačnih stanja, odnosno transduktore, implementirane unutar sistema Unitex. Primena svih navedenih sistema direktno zavisi od faze predobrade teksta u Unitexu, koja podrazumeva podelu teksta na rečenice i prepoznavanje oblika reči u tekstu, na osnovu čega se formira rečnik teksta koji transduktori dalje koriste u obradi.

U domenu sintakse razvijen je sistem za prepoznavanje složenih glagolskih oblika (Vitas & Krstev, 2003; Đorđević, 2012), koji prepoznaje sva složena glagolska vremena i načine uz pomoć konačnih transduktora organizovanih u lokalne gramatike. Kao rezultat prepoznavanja, sistem oko prepoznate jedinice, a unutar teksta, unosi odgovarajuće etikete. Sistem razvijen u (Vitas & Krstev, 2003) napravljen je za Intex, preteču Unitexa, koji se od njega razlikuje po nemogućnosti rada sa Unicode kodnim rasporedom. Drugonavedeni sistem izrađen je u Unitexu i trenutno se koristi kao jedna od pripremnih faza obrade teksta pri pretraživanju imenovanih entiteta.

U domenu prepoznavanja višočlanih imenovanih entiteta, razvijeno je nekoliko sistema.

²⁰ <http://clcl.unige.ch/SOFTWARE.html>

Sistem za prepoznavanje i ekstrakciju vlastitih imena (Gucul-Milojević, 2010) omogućava prepoznavanje imena i prezimena ljudi, uz niz mogućih dodataka (titula, drugo prezime, nadimak, puno ime oca ili inicijal imena oca). Metodologija napravljena u okviru ovog rada može se koristiti za različite vrste analiza pojavljivanja određenih ličnosti u korpusu, što dalje vodi automatskoj analizi stava, tj. proceni tona teksta u odnosu na datu osobu, ili za analizu razlika u zastupljenosti rodova u korpusu, kao što je učinjeno u (Krstev, Utvić, & Jaćimović, 2015).

Sistem za prepoznavanje vremenskih izraza (Jaćimović, 2016) definisan je preko transduktora konačnih stanja organizovanih u kaskade. Ovaj sistem prepoznaje apsolutno (*17.marta 2011. godine*), kao i relativno vreme (*nekoliko dana*). Sistem istovremeno omogućava i normalizaciju vremenskih izraza, koja podrazumeva prilagođavanje i izmenu pronađenih vremenskih izraza prema unapred definisanom obrascu usklađenim sa međunarodnim standardom.

Oblast kojoj se trenutno posvećuje značajna pažnja jeste ekstrakcija višočlanih termina (Krstev, Stanković, Obradović, & Lazić, 2015; Stankovic, Krstev, Obradović, Lazić, & Trtovac, 2016), a koja se zasad vrši za oblasti bibliotekarstva (Trtovac & Andonovski, 2014) i rudarstva i geologije (Stanković, Obradović, & Utvić, 2014). Pored ekstrahovanja, ovi sistemi istovremeno automatski i opisuju pronađene jedinice i omogućavaju njihovo dodavanje u rečnik višočlanih reči.

1.2.5 Ostali alati

Na samom kraju pomenućemo alat razvijen najpre pod imenom WS4LR²¹ (Krstev C., Stanković R., Vitas D., 2006), danas pod nazivom LeXimir. LeXimir je jezički nezavisan softverski alat napravljen sa idejom da objedini resurse napravljene za srpski jezik i olakša njihovo korišćenje, kao i održavanje i dopunjavanje. LeXimir zasad objedinjuje niz resursa i ima niz funkcionalnosti, od kojih će ovde biti pomenute samo tri koje su od značaja za naš rad:

²¹ engl. *Workstation for Lexical Resources*

- 1) Konvertovanje resursa, uglavnom iz starijeg formata Intex u Unitex i Unitex formata u NooJ, što podrazumeva konverziju rečnika i sa njima povezanih lokalnih gramatika;
- 2) Olakšano korišćenje i održavanje sistema morfoloških rečnika DELAS i DELAC – pretraživanje rečnika na osnovu leme i parametara dodeljenih unoscima, pregledanje i ispravljanje automata povezanih sa njima, unošenje novih lema i proveru automatski dodeljenih automata za izvođenje njihovih flektivnih oblika, što trenutno nije lako izvodljivo unutar Unitex-a.
- 3) Mogućnost izgradnje i pretrage paralelizovanih tekstova, i uopšte, paralelni rad sa više tekstova.

2 FORMALNE GRAMATIKE

2.1 Definicija formalne gramatike

Formalna gramatika, kao gramatika izražena sredstvima matematike i logike, čini sastavni deo formalne teorije jezika. Da bismo mogli da govorimo o definiciji i terminima formalne gramatike, neophodno je poći od određenja osnovnih pojmova formalne teorije jezika.

Kao i u prirodnom jeziku, osnovnu, nedeljivu jedinicu formalnog jezika čini **azbuka** ili **alfabet**. Alfabet, koji se po konvenciji obeležava grčkim slovom Σ predstavlja konačan skup svih slova – nedeljivih simbola ili karaktera jednog jezika. Slova formalnog jezika se obeležavaju malim slovima abecede (a, b, c). U terminima prirodnih jezika alfabet se sastoji od svih malih i velikih slova, kao i od svih karaktera koji se mogu javiti u tekstu, uključujući prazninu. U formalnom smislu alfabet je bilo koji skup karaktera i ne mora striktno sadržati slova. Primer jednog formalnog alfabeta je $\{a, b\}$ – ovaj alfabet se sastoji od dva mala slova abecede.

Nad alfabetom se definiše operacija konkatencije ili dopisivanja, koja pridodaje slova formalnog jezika jedna na druga. Rezultat operacije konkatencije je **reč** ili **niska**. Reč se označava simbolom x i predstavlja konačan niz simbola $x = (a_1, a_2, \dots, a_n)$ takav da je $n \geq 0$ i $a_i \in \Sigma$. Ukoliko je dužina reči 0, odnosno ukoliko se reč ne sastoji ni iz jednog slova, govori se o **praznoj reči** (ϵ). Niska u terminima prirodnih jezika može odgovarati pojmu reči, ali i pojmu sintagme, odnosno bilo kom skupu reči, kao i pojmu rečenice. Na primeru našeg malog alfabeta možemo formirati sledeće niske: $a, b, aa, ab, aab, bbb, \dots$

Skup Σ^* naziva se i Klinijevo zatvorenje skupa Σ i predstavlja skup svih reči nad alfabetom Σ , uključujući i praznu reč. U teoriji formalnih jezika generalno, znak $*$ označava neobavezno pojavljivanje, odnosno 0 ili više pojavljivanja simbola ili grupe simbola u čijem se superskriptu nalazi. Skup Σ^+ predstavlja skup svih nepraznih reči nad alfabetom Σ . Znak $+$ u teoriji formalnih jezika označava obavezno pojavljivanje, odnosno 1 ili više pojavljivanja simbola ili grupe simbola u čijem se superskriptu nalazi. **Jezik** (L) nad alfabetom Σ je bilo koji podskup skupa Σ^* , što se formalno beleži kao $L \subseteq \Sigma^*$. Skup već pomenutih niski $a, b, aa, ab, aab, bbb, \dots$ tako predstavlja jedan jezik. Taj jezik se može formalno opisati sledećom formulom: $L = \{a^n b^m \mid n, m \geq 0\}$ koja kaže da će se sve niske našeg jezika sastojati od nula ili

više pojavljivanja slova a , za kojim sledi nula ili više pojavljivanja slova b , pri čemu broj pojavljivanja slova a i b u niskama ne mora biti isti. Tako je niska a opisana formulom a^1b^0 , niska bbb formulom a^0b^3 itd.

Ukoliko niska x pripada jeziku L : $x \in L$, kažemo da ona predstavlja **rečenicu** jezika L . Za jezik koji se sastoji od prebrojivog broja niski ili rečenica kažemo da je konačan jezik; u suprotnom, jezik je beskonačan. Ukoliko je jezik konačan i sastoji se od malog broja rečenica, to da li jedna rečenica pripada datom jeziku možemo proveriti prostim upoređivanjem sa skupom rečenica koje mu pripadaju. Međutim, za jezike koji se sastoje od velikog broja rečenica ili su beskonačni, potrebno je osmisliti drugačiji mehanizam provere. Mehanizam koji može da pokaže da li jedna rečenica pripada datom jeziku naziva se **gramatika**. Za gramatiku ujedno kažemo i da generiše jezik koji opisuje.

Formalno, gramatika G je uređena četvorka (Σ, N, P, S) gde je:

- Σ završni alfabet, koji se sastoji od završnih (terminalnih) simbola. Za potrebe našeg rada, pod ovim pojmom ćemo podrazumevati kategorije reči (N – imenica, V – glagol itd.);
- N nezavršni alfabet, koji se sastoji od nezavršnih (pomoćnih, neterminalnih) simbola takav da je $\Sigma \cap N = \emptyset$. Za potrebe našeg rada, pod ovim pojmom ćemo podrazumevati sintagme ili fraze (NP – imenička fraza, VP – glagolska fraza itd.)
- S početni (rečenični, startni) simbol, takav da $S \in N$; S će u našem radu označavati rečenicu.
- P konačni skup pravila takav da je $P \subseteq (\Sigma \cup N)^* N (\Sigma \cup N)^* \times (\Sigma \cup N)^*$. Pravilo se sastoji od leve i desne strane pravila (npr. $S \rightarrow NP VP$), pri čemu leva strana pravila mora sadržati makar jedan element skupa N .

Sve niske koje se sastoje od terminalnih simbola, a koje su izvedene od početnog simbola određene gramatike, pripadaju jeziku koji je opisan tom gramatikom. Pogledajmo kako ovo izvođenje izgleda na primeru našeg malog jezika. Gramatika koja definiše i generiše niske našeg jezika je sledeća:

$$G = (\{a, b\}, \{A, B\}, S, P), \text{ gde je } P = \{S \rightarrow AB, A \rightarrow Aa, A \rightarrow \varepsilon, B \rightarrow Bb, B \rightarrow \varepsilon\}$$

Definicija ove gramatike kaže da se završni alfabet sastoji od slova a i b , da su nezavršni simboli koji se koriste pri izvođenju A i B , da je početni simbol S , i da su pravila ona koja su navedena. Pravila su data u formi takozvanih *pravila prepisivanja* (produkcije, proizvodnje, izvođenja), a pokazuju u šta se simbol sa leve strane pravila transformiše, što je prikazano s desne strane pravila. Pravila prepisivanja su rekurzivna ukoliko se isti neterminalni simbol sa leve strane pravila nađe i sa desne strane pravila (pravila tipa $A \rightarrow Aa$ i $B \rightarrow Bb$)²².

Koristeći ovu gramatiku, možemo proveriti da li niska $abbb$ pripada našem jeziku, odnosno da li može biti generisana ili izvedena pravilima naše gramatike. Tom prilikom polazimo od početnog simbola i dalje primenjujemo pravila koja bi potencijalno mogla da generišu ovakvu nisku. Rezultat je sledeći:

$$S \rightarrow AB \rightarrow AaBb \rightarrow \varepsilon aBbb \rightarrow aBbbb^{23} \rightarrow a\varepsilon bbb \rightarrow abbb$$

Kako smo primenom postojećih pravila uspeli da generišemo datu nisku, zaključujemo da ona pripada jeziku opisanom našom gramatikom. Isto tako bismo primenom pravila naše gramatike mogli da vidimo da niska $abaa$ ne pripada našem jeziku, kako u gramatici našeg jezika ne postoji pravilo koje može da generiše slovo ili slova a posle slova b .

2.2 Formalna gramatika i prirodni jezik

Ideja da se jezik može izraziti formalnim (matematičkim i logičkim) sredstvima datira sa početaka dvadesetog veka. 1914. godine Aksel Tju (*Axel Thue*), norveški matematičar, predstavlja formalizovan prikaz transformacija među niskama, koji je u osnovi bio sistem prepisivanja niski čiji smo primer mogli da vidimo u prethodnom poglavlju. Teorijom prepisivanja nastavlja da se bavi Emil Post (*Emil Post*), čiji rad (Kracht, 2003, p.

²² Rekurzija kao metod podrazumeva pozivanje samog sebe.

²³ Dopisivanje prazne reči ε ne utiče na reč kojoj se dopisuje: $\varepsilon a = a\varepsilon = a$

65) navodi kao inspiraciju za klasifikaciju formalnih jezika koju je dao Noam Čomski (*Noam Chomsky*).

U radu *Tri modela za opis jezika*²⁴ (Chomsky, 1956) Čomski navodi podelu formalnih gramatika na četiri grupe prema strukturi pravila prepisivanja koju koriste, odnosno kompleksnosti same gramatike. Svaka od ovih gramatika opisuje (generiše) tip jezika koji prepoznaje odgovarajući tip automata. Primarni interes koji Čomski navodi pri definisanju gramatika jeste pronalaženje gramatike koja bi generisala sve ispravne i samo ispravne rečenice engleskog jezika, u nadi da takva gramatika može voditi otkrivanju generalne teorije jezičke strukture tj. univerzalne gramatike svih prirodnih jezika. Četiri tipa formalnih gramatika uređeno je u hijerarhiju od najjednostavnije do najkompleksnije i po svom autoru nose naziv *hijerarhija Čomskog*:

1. **Tip 3** ili **regularne** (engl. *regular*), još zvane i **Klinijeve gramatike**. Struktura pravila ovog tipa gramatika je takva da sadrže jedan pomoćni (nezavršni) simbol s leve strane pravila, dok se desna strana pravila sastoji od najviše jednog pomoćnog simbola i bilo kog broja terminalnih (završnih) simbola. U zavisnosti od toga s koje strane se na desnoj strani pravila nalazi pomoćni simbol, ove gramatike mogu biti desno-linearne (pravilo tipa $A \rightarrow aB$) ili levo-linearne (pravilo tipa $A \rightarrow Ba$). Ovo je ujedno i najjednostavniji tip gramatika u hijerarhiji.

Regularne gramatike opisuju regularne jezike i prepoznaju se konačnim automatima²⁵. Prema tipu automata koji ih prepoznaje ovi jezici se ponekad nazivaju i jezicima konačnih stanja. Ove jezike je moguće opisati i regularnim izrazima.

2. **Tip 2** ili **kontekstno slobodne** (engl. *context-free*), još poznate i kao **algebarske gramatike**. Pravila ovih gramatika sadrže jedan pomoćni simbol s leve strane i bilo koji broj terminalnih ili neterminalnih simbola s leve strane: $A \rightarrow \beta$, $A \in N$, a $\beta \in (\Sigma \cup N)^*$.

Kontekstno slobodne gramatike opisuju kontekstno slobodne jezike, a prepoznaju se nedeterminističkim potisnim automatom. Mala gramatika koju smo definisali u prethodnom poglavlju pripada ovom tipu gramatika.

²⁴ engl. *Three Models for the Description of Language*

²⁵ Kako teorija automata prevazilazi okvire ovog rada, za detaljan opis strukture i funkcionisanja automata koji će biti pomenuti u ovom poglavlju upućujemo na (Chomsky, 1963) ili na srpskom jeziku (Vitas, 2006). Konačni automati pomenuti u poglavlju **Error! Reference source not found.** su zapravo ova vrsta automata.

3. **Tip 1** ili **kontekstno osetljive gramatike** (engl. *context-sensitive*). Pravila ovog tipa gramatika ne smeju imati više simbola s leve strane pravila nego s desne strane, i sledećeg su tipa: $uXv \rightarrow uyv, X \in N, y \neq \varepsilon, u, v \in (\Sigma \cup N)^*$.

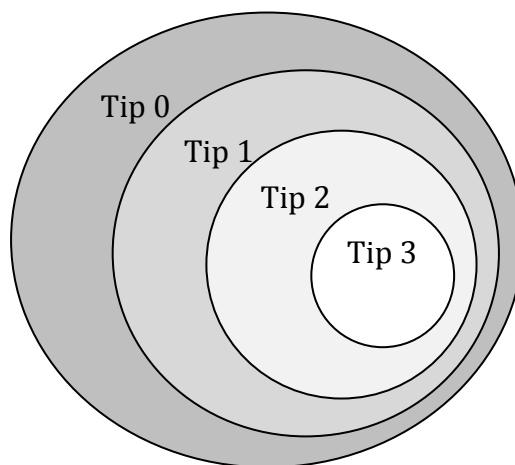
Ove gramatike opisuju kontekstno osetljive jezike, a prepoznaju se linearno ograničenom nedeterminističkom Turingovom mašinom. Nazivaju se kontekstno osetljivima jer su to prve gramatike koje pri prepisivanju uzimaju u obzir kontekst (u navedenom pravilu – u i v).

4. **Tip 0** ili **gramatike bez ograničenja** (engl. *unrestricted*). Jedina ograničenja na strukturu pravila ove gramatike jesu da leva strana pravila ne sme biti prazna i da mora sadržati makar jedan pomoćni simbol.

Ove gramatike opisuju jezike bez ograničenja ili rekurzivno prebrojive jezike, a prepoznaju se Turingovom mašinom. Gramatike ovog tipa su ujedno i najkompleksnije formalne gramatike.

Gramatike tipa 1, 2 i 3 su odlučive, što znači da za svaku nisku terminalnih simbola mogu da odrede da li pripadaju datom jeziku. U slučaju gramatika tipa 0, to nije uvek slučaj.

Svaka od ovih gramatika predstavlja podskup sebi nadređene gramatike, pa je tako, na primer, svaka regularna gramatika je istovremeno i kontekstno slobodna, dok obrnuto ne važi. Slika 3 daje grafički prikaz ove zakonitosti.



Slika 3 Čomskijeva hijerarhija formalnih gramatika

Pogledajmo sada kakav je odnos ovih gramatika i prirodnih jezika. U već pomenutom radu iz (1956), Čomski kao uslov adekvatnosti lingvističke teorije, odnosno njene gramatike, navodi sledeće: „... ukoliko je određena rečenica S dvosmisljena, možemo proveriti adekvatnost date lingvističke teorije postavljajući pitanje da li ili ne najjednostavnija gramatika izgradiva u terminima ove teorije za dati jezik automatski obezbeđuje različite načine za generisanje rečenice S“²⁶. Pojmove čiji se začetak vidi u ovoj definiciji Čomski definiše u (Chomsky, 1965) kao *slab* i *jak generativni kapacitet gramatike*. Slab generativni kapacitet gramatike odnosi se na ukupan skup rečenica (niski) koje jedna gramatika može da generiše. Jak generativni kapacitet gramatike odnosi se na skup strukturnih opisa koje jedna gramatika može da dodeli rečenicama (niskama) koje generiše. Adekvatna gramatika prirodnog jezika tako ne samo da treba da generiše sve gramatične i samo gramatične rečenice jednog jezika, već treba da im pripiše i odgovarajuće strukture, po kojima bi se dvosmisljene rečenice međusobno mogle razlikovati. Ideja da odgovarajuća gramatika *generiše* rečenice datog jezika polazeći od konačnog vokabulara i konačnog broja pravila dala je naziv *generativna* čitavoj klasi gramatika, kao i pravcu u lingvističkoj teoriji, čijim rodonačelnikom se smatra upravo Čomski.

U skladu sa navedenim parametrima, Čomski (1956) ispituje kapacitet regularne i kontekstno slobodne gramatike za opisivanje prirodnog jezika i zaključuje da složenost prirodnog jezika prevazilazi slab i jak generativni kapacitet obe gramatike, odnosno da postoji veći broj rečenica i pojava u jeziku nego što ove dve gramatike mogu da opišu²⁷. Štaviše, regularne gramatike ne dodeljuju strukturu rečenicama koje prepoznaju. Za njih, rečenica je niz reči, a jezik niz rečenica te se u slučaju regularnih gramatika može govoriti samo o slabom generativnom kapacitetu. Imajući to u vidu, ne postoji način da regularne gramatike na različite načine opišu dvosmisljene strukture.

Kontekstno slobodne gramatike, s druge strane, dodeljuju određenu strukturu svojim rečenicama, i to strukturu koju Čomski naziva *fraznom strukturom*. Naime, ove gramatike predviđaju postojanje složenih struktura, fraza, koje obuhvataju reč ili grupe

²⁶ Prevod autorke. U originalu: ... *if a certain sentence S is ambiguous, we can test the adequacy of a given linguistic theory by asking whether or not the simplest grammar constructible in terms of this theory for the language in question automatically provides distinct ways of generating the sentence S.*

²⁷ Za detalje dokaza upućujemo na sam rad (Chomsky, 1965).

susednih reči, čijim se daljim kombinovanjem dolazi do rečenice. Izvođenje jedne ovakve strukture počevši od startnog simbola rečenice *S* moguće je prikazati preko pravila prepisivanja, koja smo videli u prethodnom poglavlju, ili grafički, preko stabala frazne strukture²⁸. Tako strukturu dvosmislene rečenice *Marko gleda scenu na brodu*, možemo ilustrativno opisati sledećom gramatikom:

$$G = (\{N, V, Prep\}, \{NP, VP, PP\}, S, P)$$

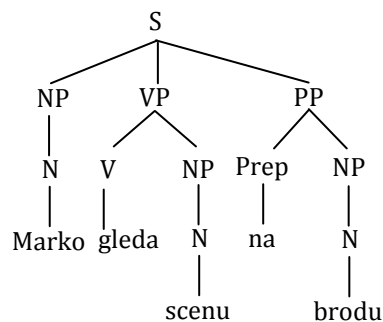
$$P = \{S \rightarrow NP VP, S \rightarrow NP VP PP, NP \rightarrow N, NP \rightarrow N PP, VP \rightarrow V NP, PP \rightarrow Prep NP, PP \rightarrow \varepsilon, N \rightarrow Marko|^{29}scenu|brodu, V \rightarrow gleda, Prep \rightarrow na \}$$

Izvođenje date rečenice na osnovu ove gramatike možemo predstaviti na dva različita načina u zavisnosti od čitanja:

1) Marko se nalazi na brodu i na njemu gleda neku scenu:

$$S \rightarrow NP VP PP \rightarrow N V NP Prep NP \rightarrow Marko gleda N na N \rightarrow Marko gleda scenu Prep NP \rightarrow Marko gleda scenu na N \rightarrow Marko gleda scenu na brodu$$

Slika 4 prikazuje stablo frazne strukture za ovo čitanje.



Slika 4 Stablo frazne strukture za prvo tumačenje rečenice *Marko gleda scenu na brodu*.

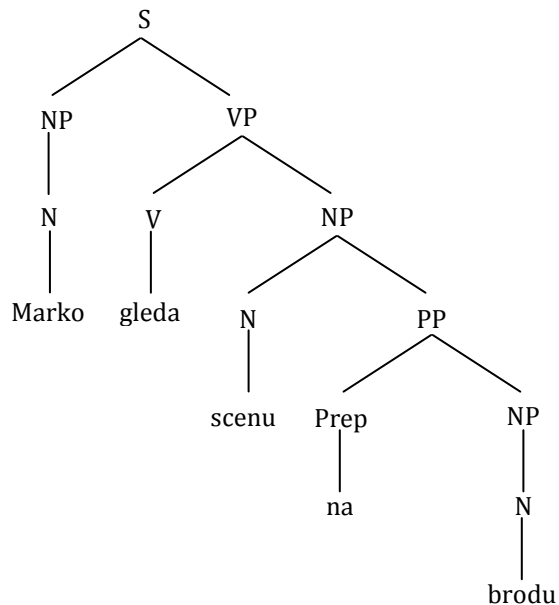
²⁸ Ali, između ostalog, i tabelarno ili obeležavanjem pomoću zagrada.

²⁹ Uspravna crta “|” označava logički operator ILI – kategorija *N* se može realizovati kao *Marko* ili *scenu* ili *brod*.

2) Marko gleda scenu koja se odvija na brodu:

$S \rightarrow NP VP \rightarrow N V NP \rightarrow Marko\ gleda\ N\ PP \rightarrow Marko\ gleda\ scenu\ Prep\ NP \rightarrow Marko\ gleda\ scenu\ na\ N \rightarrow Marko\ gleda\ scenu\ na\ brodu$

Slika 5 prikazuje stablo frazne strukture za ovakvo čitanje.



Slika 5 Stablo frazne strukture za drugo tumačenje rečenice *Marko gleda scenu na brodu*

Ono sa čim gramatika frazne strukture ipak ne može da izađe na kraj jesu rečenice ili fraze koje su po strukturi identične, ali predstavljaju drugačiji tip dvosmislenosti. Primer ove pojave bile bi imeničke fraze *zavijanje mačaka* i *uzgajanje cveća*³⁰. Njihova struktura je identična – u pitanju su imeničke fraze koje se prepisuju kao imenica (u nominativu) koja je praćena imenicom (u genitivu), međutim, u slučaju prve fraze, *mačke* predstavljaju aktivnog vršioca, odnosno rečenica koja se nalazi u njoj osnovi jeste rečenica *Mačke zavijaju*. U drugom primeru, *cveće* trpi radnju, što se može videti kada se fraza transformiše u rečenicu *(Neko) uzgaja cveće*.

³⁰ Fraze su inspirisane primerima iz (Chomsky, 1965)

Da bi strukture ovog i sličnog tipa³¹ mogle biti ispravno analizirane u jeziku, Čomski u istom radu u kontekstno slobodne gramatike uvodi operaciju transformacije. Ova operacija podrazumeva postojanje dve strukture u jeziku – dubinske, koja se sastoji od bazičnih³² rečenica jezika (*Mačke zavijaju.*) i površinske u kojoj se iste rečenice nalaze u obliku u kakvom ih srećemo u svakodnevnoj upotrebi (*zavijanje mačaka*). Transformacije su operacije koje povezuju ove strukture tako što transformišu bazične rečenice u njihove strukturne varijante (kao što su npr. i pasiv ili pitanje).

Prva formulacija transformacija u teoriji Čomskog podrazumevala je da one mogu vršiti brisanje rečeničnih delova, njihovo dodavanje, zamenu i menjanje njihovog mesta. Ova rana definicija je kritikovana kao izrazito nerestriktivna (Müller, 2016). Zbog mogućnosti da segmente rečenice zamenjuju drugim proizvoljnim segmentima, transformacije su značajno povećavale generativni kapacitet gramatike, čineći je čak formalizmom tipa 0 (Peters & Ritchie, 1973). Isto tako, nije bilo definisano na koji način transformacije međusobno deluju – u kom redosledu i u kom broju, posebno uzevši u obzir operaciju brisanja. Kao reakciju na ove kritike, u sledećim verzijama svoje teorije (Chomsky, 1981) ograničava moć transformacija, uvodeći između ostalog i zahtev da u površinskoj strukturi za obrisanim ili izmeštenim elementom mora ostati *trag*, realizovan kao niska bez fonološke realizacije (prazna niska)³³.

Tvrđnja Čomskog da prirodni jezici ne mogu biti opisani kontekstno slobodnom gramatikom bila je opšteprihvaćena preko dvadeset narednih godina. Ranih osamdesetih (Pullum & Gazdar, 1982) sakupili su na jednom mestu sve argumente koji su se u toku godina navodili u prilog ovoj tezi. Argumenata je bilo ukupno pet³⁴, međutim za svaki od njih autori navode odgovarajuće kontraargumente. Kao pokušaj da dâ još jednu potporu za svoju tvrdnju da je prirodne jezike moguće opisati ovim tipom gramatika, Gezdar (*Gerald Gazdar*) krajem sedamdesetih godina razvija gramatiku za engleski jezik sa kontekstno

³¹ Čomski (1956) kao primer navodi tretman pasiva, ali nećemo ovde rekonstruisati njegovu analizu.

³² engl. *kernel*

³³ (Pelletier, 1980, pp. 29–33) navodi još nekoliko metodoloških pokušaja različitih autora da ograniče snagu transformacija.

³⁴ Za argumente i njihova opovrgavanja upućujemo direktno na (Pullum & Gazdar, 1982)

slobodnom osnovom – *generalizovanu gramatiku frazne strukture* ili GPSG³⁵(Gazdar, Klein, Pullum, & Sag, 1985)³⁶.

S druge strane, na osnovu dokaza iz germanskih jezika, (Shieber, 1985) i nešto kasnije (Miller, 1991) navode da ni slab ni jak generativni kapacitet kontekstno slobodnih gramatika nisu dovoljni da opišu prirodne jezike. Krajem šezdesetih i početkom sedamdesetih godina se već pojavio niz autora koji uz tvrdnju da su prirodni jezici umereno kontekstno osetljivi, razvijaju formalizme koji se po slabom generativnom kapacitetu svrstavaju između kontekstno slobodnih i kontekstno osetljivih gramatika. Jedan od ovih formalizama je i *gramatika adjungovanih stabala* – TAG³⁷ (Joshi, Levy, & Takahashi, 1975). Osnovni objekti kojima operiše TAG više nisu niske, ali ni frazne strukture, već strukturirani objekti – stabla.

Još jedna klasa gramatika koja se u literaturi navodi kao odgovarajuća za predstavljanje prirodnih jezika jeste takozvana *indeksna gramatika* (Aho, 1968). Indeksna gramatika prema slabom generativnom kapacitetu spada između kontekstno slobodnih i kontekstno osetljivih gramatika, i obuhvata umereno kontekstno osetljive gramatike. (Gazdar, 1988) daje moguće primene ove gramatike na prirodne jezike, navodeći da je moguće da postoje segmenti prirodnih jezika koje čak ni indeksna gramatika ne može da opiše. Kao korak još dalje, (Kaplan & Bresnan, 1982) definišu *leksičko-funkcionalnu gramatiku* ili LFG³⁸, koja spada u klasu kontekstno osetljivih gramatika. Kao zaključak, navodimo tvrdnju iz (Walter J. Savitch, 1987) po kojoj nije poznat dokaz da je bilo koja struktura prirodnog jezika van dometa kontekstno osetljivih gramatika, i to je stav koji je prihvaćen i danas danas.

2.3 Unifikacione gramatike

Počevši od sedamdesetih godina prošlog veka, sa različitih strana dolazi kritika transformacione gramatike (Abeillé, 2007). Osnovna grupa zamerki dolazi od interpretacije

³⁵ engl. *Generalized Phrase Structure Grammar*

³⁶ U literaturi (Berwick, 1984) postoje naznake da dodaci uvedeni u GPSG ipak čine ovu gramatiku jačom od kontekstno slobodne.

³⁷ engl. *Tree-Adjoining Grammar*

³⁸ engl. *Lexical Functional Grammar*

dokaza u (Peters & Ritchie, 1973) koja govori u prilog tome da transformacije generišu jezik tipa 0, za koji se smatra da ima previše jak generativni kapacitet za OPJ. Druga grupa zamerki stiže od psiholingvisti koji u svojoj praksi nisu pronašli dokaze za transformacije kao mentalne operacije. S druge strane, upotreba stabala frazne strukture nije se pokazala pogodnom za prikazivanje rečenične strukture *nekonfiguracionih* jezika, odnosno jezika sa slobodnim redom reči, zbog čega se javila potreba za drugačijim reprezentacijama strukture. Dalje, transformacioni model nije bio praktičan za primenu u automatskoj analizi teksta, kako primenjuje suprotan postupak od onog od koga se u analizi polazi – polazi od dubinske strukture umesto od površinske. Iz tog razloga transformaciona gramatika je i danas najpre lingvistička teorija. Međutim, možda najžustrija kritika transformaciono-generativne teorije uopšte data je u radu Morisa Grosa (Gross, 1979), i bazirana je na Grosovom iskustvu pri pravljenju gramatike francuskog jezika na transformacionim osnovama (Gross, 1975). Osnovna zamerka izneta u ovom radu jeste to što se generativna teorija usmerava na definisanje apstraktnih pravila univerzalne gramatike, odbijajući da se posveti konkretnim zadacima – pravljenju gramatika konkretnih jezika, i pre svega temeljnoj klasifikaciji reči u leksikonu, koji za Grosu predstavlja osnov svih strukturnih razlika među konstrukcijama višeg reda. Po Grosu, generalizovanjem reči na opšte kategorije (*NP, VP* itd.), generativna gramatika propušta da primeti da karakteristike reči unutar iste klase nisu nužno iste, i da su zapravo češće vrlo različite, te da je izbor struktura u kojima se mogu naći za svaku leksemu mnogo ograničeniji nego što to generativna gramatika pretpostavlja³⁹.

Kao rezultat navedene kritike, počevši od sredine sedamdesetih godina javlja se nova vrsta formalizama koji mogu biti grupisani pod naziv *unifikacione gramatike*. Prema (Shieber, 2003; Abeillé, 2007) unifikacione gramatike se karakterišu time što:

- **integrišu leksikon, sintaksu i semantiku u jedinstvenu strukturu.**

Unifikacione gramatike su mahom leksikalizovane, što znači da su gramatičke strukture direktno povezane sa određenim leksemama (ili skupom leksema). Takođe, uz

³⁹ Grosovo pridavanje značajne uloge leksikonu pri definisanju sintakse uobličeno je u formalizam nazvan *leksikon-gramatika* (engl. *Lexicon-Grammar*). Ovaj formalizam je zapravo nastao u toku izgradnje pomenute gramatike na transformacionim osnovama (Gross, 1975). Na temelju ove teorije definisan je i LADL format elektronskih rečnika o kome je bilo reči u poglavlju 1.2.2.

svaku leksikalizovanu strukturu paralelno se daje i njena semantička struktura, pre svega za potrebe razrešavanja višeznačnosti i tretmana složenih izraza kakvi su idiomi.

- **direktno opisuju konkretnu površinsku realizaciju niske.**

Unifikacione gramatike se vraćaju staroj tradiciji konstituentske analize. Budući da ne podrazumevaju postojanje dubinske strukture, gubi se potreba za mehanizmom transformacija.

- **odnos između struktura opisuju leksičkim pravilima.**

Unifikacione gramatike pokušavaju da operaciju transformacije učine potpuno suvišnom. U te svrhe se definišu takozvana *leksička pravila (metapravila)*, koja predstavljaju svojevrsan pandan transformacijama, s tim što se od njih razlikuju na dva značajna načina. Pre svega, dok se transformacije primenjuju na rečenice (niske), leksička pravila se primenjuju na lekseme. Imajući u vidu da je u ovim gramatikama leksema dominantni nosilac sintaksičke informacije, leksičkim pravilima se osnovne strukture jedne lekseme povezuju sa svim drugim strukturama u kojima se ona može naći. S druge strane, transformacije predstavljaju proces, tj. definišu način na koji se od jedne strukture kroz niz koraka izvodi druga strukture. Leksička pravila su najčešće obrazac na osnovu koga se proverava da li je određena struktura gramatična.

- **dodeljuju svojim niskama (rečima, frazama i rečenicama) određene informacije u vidu obeležja.**

Obeležje predstavlja par koji se sastoji od atributa (npr. rod) i njegove vrednosti (muški|ženski|srednji). Da bi fenomen kao što je slaganje, recimo između prideva i imenice, mogao da se definiše preko klasičnih pravila prepisivanja gramatika frazne strukture, potrebno je za svaku kombinaciju kategorija u kojima se pridev i imenica slažu napraviti posebno pravilo ($NP_1 \rightarrow A_1 N_1$, $NP_2 \rightarrow A_2 N_2$ itd. gde A_1 predstavlja pridev u nominativu jednine muškog roda, a N_1 imenicu s istim kategorijama, A_2 predstavlja pridev u genitivu jednine muškog roda a N_2 imenicu s istim kategorijama i tako dalje za svaku kombinaciju odgovarajućih kategorija). Umesto umnožavanja pravila, ovakva vrsta informacija se kod unifikacionih gramatika unosi u strukturu kroz obeležja za koje se definiše međusobno slaganje ($NP \rightarrow A N$, $\langle A \text{ rod} \rangle = \langle N \text{ rod} \rangle$ $\langle A \text{ padež} \rangle = \langle N \text{ padež} \rangle$ $\langle A \text{ broj} \rangle = \langle N \text{ broj} \rangle$). Obeležja se dodeljuju i leksemama, čime se uslovljavaju strukture u

koje date lekseme mogu ući (npr. <pasiv>=+|- u slučaju glagola označava da li dati glagol može ući u pasivnu konstrukciju). Struktura obeležja se često predstavlja u vidu matrice, i kao takvu ćemo je nadalje koristiti u ovom uvodnom delu.

Struktura obeležja može biti atomična – kada atribut sadrži samo proste vrednosti. U slučaju prideva, to može biti obeležje tip s vrednostima opisni|gradivni|prisvojni. Za pridev *lep* imaćemo ovakvu matricu:

$$\begin{bmatrix} \text{KATEGORIJA} & \text{A} \\ \text{TIP} & \text{opisni} \end{bmatrix}$$

Ipak, inovacijom unifikacionih gramatika smatra se složena struktura obeležja⁴⁰, gde se vrednost sastoji od nove strukture obeležja, u ovom slučaju kod obeležja SLAGANJE:

$$\begin{bmatrix} \text{KATEGORIJA} & \text{A} \\ \text{SLAGANJE} & \begin{bmatrix} \text{ROD} & \text{m} \\ \text{BROJ} & \text{sing} \\ \text{PADEŽ} & \text{nom} \end{bmatrix} \end{bmatrix}$$

Primer slaganja između prideva i imenice mogao bi se pomoću matrice predstaviti ovako:

$$\begin{bmatrix} \text{KATEGORIJA} & \text{A} \\ \text{SLAGANJE} & \begin{bmatrix} \text{ROD} & \text{m} \\ \text{BROJ} & \text{sing} \\ \text{PADEŽ} & \text{nom} \end{bmatrix} \end{bmatrix} \begin{bmatrix} \text{KATEGORIJA} & \text{N} \\ \text{SLAGANJE} & \begin{bmatrix} \text{ROD} & \text{m} \\ \text{BROJ} & \text{sing} \\ \text{PADEŽ} & \text{nom} \end{bmatrix} \end{bmatrix}$$

Činjenica da pridev i imenica dele vrednosti obeležja SLAGANJE može biti prikazana i na sledeći način:

$$\begin{bmatrix} \text{A} \\ \text{N} \end{bmatrix} \begin{bmatrix} \text{SLAGANJE} & \begin{bmatrix} \text{ROD} & \text{m} \\ \text{BROJ} & \text{sing} \\ \text{PADEŽ} & \text{nom} \end{bmatrix} \\ \text{[SLAGANJE]} & \begin{bmatrix} \text{ROD} & \text{m} \\ \text{BROJ} & \text{sing} \\ \text{PADEŽ} & \text{nom} \end{bmatrix} \end{bmatrix}$$

⁴⁰ Uvedena u (Kay, 1984).

Simbol broja u kućici – $\boxed{1}$ koristi se da označi deljene ili takozvane *uvučene*⁴¹ strukture obeležja.

- **gramatičnost definišu deklarativno, a ne proceduralno.**

Pristup generativne gramatike po kome je određena rečenica gramatična ukoliko je moguće generisati je kroz niz propisanih pravila prepisivanja je primer proceduralne definicije gramatičnosti. Unifikacione gramatike, s druge strane, definišu gramatičnost preko skupa ograničenja koje niska mora da zadovolji⁴². Skup ograničenja može biti lokalni, kao onaj za slaganje između prideva i imenice, ili generalan i primenjiv na sve strukture u jeziku, oličen u takozvanim *principima dobre formiranosti*. Određivanje gramatičnosti prema tome da li niska zadovoljava uslove koje postavlja određena gramatika naziva se deklarativnim pristupom.

- **definišu novu operaciju pod imenom unifikacija.**

Unifikacija je operacija koja vrši proveru usklađenosti (kompatibilnosti) među strukturama obeležja pri izvođenju rečeničnih struktura. Rezultat unifikacije dve strukture obeležja X i Y, koji se označava kao XUY, je minimalna struktura koja sadrži sve informacije sadržane u X i u Y. Ukoliko nije moguće formirati takvu strukturu, unifikacija se ne može izvršiti (što se obeležava znakom \perp). Pogledajmo kako izgledaju različiti primeri unifikacije na našim primerima prideva i imenice⁴³:

$$1) \left[\begin{array}{c} \text{KATEGORIJA} \\ \text{A} \end{array} \right] \cup \left[\begin{array}{c} \text{KATEGORIJA} \\ \text{N} \end{array} \right] = \perp$$

$$2) \left[\begin{array}{c} \text{KATEGORIJA} \\ \text{A} \end{array} \right] \cup \left[\begin{array}{c} \text{SLAGANJE} \\ \text{[ROD m]} \end{array} \right] = \left[\begin{array}{c} \text{KATEGORIJA} \quad \text{A} \\ \text{SLAGANJE} \quad \text{[ROD m]} \end{array} \right]$$

$$3) \left[\begin{array}{c} \text{KATEGORIJA} \quad \text{N} \\ \text{SLAGANJE} \quad \text{[ROD m]} \end{array} \right] \cup \left[\begin{array}{c} \text{SLAGANJE} \\ \text{[BROJ sing]} \end{array} \right] = \left[\begin{array}{c} \text{KATEGORIJA} \quad \text{N} \\ \text{SLAGANJE} \quad \text{[ROD m]} \\ \text{[BROJ sing]} \end{array} \right]$$

$$4) \left[\begin{array}{c} \text{KATEGORIJA} \quad \text{N} \\ \text{SLAGANJE} \quad \text{[ROD m]} \\ \text{[BROJ sing]} \end{array} \right] \cup \left[\begin{array}{c} \text{SLAGANJE} \\ \text{[ROD m]} \end{array} \right] = \left[\begin{array}{c} \text{KATEGORIJA} \quad \text{N} \\ \text{SLAGANJE} \quad \text{[ROD m]} \\ \text{[BROJ sing]} \end{array} \right]$$

$$5) \left[\begin{array}{c} \text{KATEGORIJA} \quad \text{N} \\ \text{SLAGANJE} \quad \text{[ROD m]} \\ \text{[BROJ sing]} \end{array} \right] \cup \left[\begin{array}{c} \text{SLAGANJE} \\ \text{[BROJ plur]} \end{array} \right] = \perp$$

⁴¹ engl. *reentrant*

⁴² Zbog čega se ova grupa gramatika često definiše kao bazirana na ograničenjima (engl. *constraint-based*), videti (Müller, 2016, pp. 488–497)

⁴³ Primeri unifikacije, kao i generalno primeri u ovom poglavlju, inspirisani su primerima u (Abeillé, 2007)

Unifikacija nije mogla da se izvrši u prvom i petom slučaju, u prvom kako strukture obeležja i ne postoje, a u petom kako strukture nisu kompatibilne. Unifikacija u primerima dva i tri dovodi do usložnjavanja strukture, dok se u četvrtom primeru već postojeće obeležje ne udvaja, već struktura ostaje ista.⁴⁴.

- **zasnivaju se na logičkim i matematičkim modelima za koje su definisane programske metode.**

Tri najranije unifikacione gramatike, iako konstruisane za potrebe OPJ, zapravo su u velikoj meri i dalje bile pre matematički konstrukti nego teorije sintaksičke strukture. U pitanju su PATR-II (Shieber, Uszkoreit, Pereira, Robinson, & Tyson, 1983), funkcionalna unifikaciona gramatika – FUG⁴⁵ (Kay, 1984) i gramatika DCG (Pereira & Warren, 1980). Ono što (Abeillé, 2007, pp. 24–25) navodi kao nedostatke ova tri formalizma u smislu OPJ jeste to što ne mogu da postignu deskriptivnu adekvatnost za jezik koji opisuju drugačije nego umnožavanjem pravila i principa koje koriste. Ne zasnivaju se na opštim jezičkim principima ili metodu koji bi se mogao uspešno primeniti i na druge jezike, i kao takvi ne mogu se iskoristiti za konstruisanje opsežnih gramatika prirodnih jezika, niti se mogu nazvati sintaksičkim teorijama. Ipak, mogu se vrlo uspešno koristiti u krajnjim fazama primene gramatika, kao formalizmi koji opisuju same formalne gramatike (takozvane *metagramatike*)⁴⁶.

Tri dominantne unifikacione gramatike, koje su istovremeno i formalizmi za OPJ i sintaksičke teorije, su LFG (Kaplan & Bresnan, 1982), HPSG⁴⁷ (Pollard & Sag, 1994) i TAG (Joshi et al., 1975).

LFG nastavlja tradiciju Čomskog u smislu da teži da definiše univerzalnu gramatiku ljudskog jezika, iako na potpuno drugačijim osnovama. LFG na velika vrata vraća sintaksičku analizu u terminu sintaksičkih funkcija (koju naziva *funkcionalna struktura*), čije kombinovanje sa analizom u terminima sintaksičkih kategorija (koju naziva *konstituentna struktura*) omogućava uspešnu analizu nekonfiguracionih jezika. Pod

⁴⁴ U terminima logike, za operacije sa ovim svojstvom kaže se da su idempotentne.

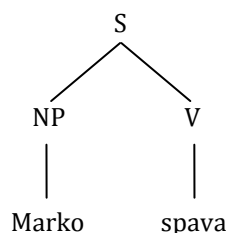
⁴⁵ engl. *Functional Unification Grammar*

⁴⁶ O pojmu metagramatike govorićemo u poglavlju 2.5

⁴⁷ engl. *Head-driven Phrase Structure Grammar* ili gramatika preokrenute frazne strukture. GPSG, o kome je bilo reči u poglavlju 2.2 predstavlja preteču ove gramatike.

uticajem FUG, koji kao alternativu stablima frazne strukture prvi uvodi predstavljanje rečenice u vidu strukture obeležja (u formi ranije navedenih matrica), LFG prihvata ovaj metod za predstavljanje funkcionalne strukture. Za predstavljanje konstituentne strukture LFG i dalje zadržava predstavu u vidu stabala frazne strukture (Slika 6).⁴⁸

Konstituentna struktura



Funkcionalna struktura

	[PRED ⁴⁹	'Marko']
		ROD	m
		BROJ	sing
		PADEŽ	nom
]	
	PRED	'spavati'⟨SUBJ⟩ ⁵⁰	
	VREME	prez	
	BROJ	sing	
	LICE	3	

Slika 6 Struktura rečenice *Marko spava.* u formalizmu LFG

HPSG je formulisan je sa idejom jasne integracije fonologije, sintakse i semantike. Ova ideja je realizovana u formi složenih struktura obeležja pridodatih svakoj leksemi unutar leksikona, čineći HPSG istovremeno i primerom izrazito leksikalizovanog formalizma. Predstava rečenica koja se koristi u ovoj gramatici podseća na stabla frazne strukture, s tim što su čvorovi stabla zapravo strukture obeležja. Svaka struktura obeležja sastoji se od atributa FON (fonološka predstava), SINTAKSA i SEMANTIKA u okviru kojih se dalje definiše data jedinica. Grane stabla obeležene su prema funkciji jedinice do koje vode (S – subjekat, G – glava⁵¹, odnosno upravni član jedinice) (Slika 7). Kao i LFG, i HPSG pretenduje na to da bude formalizam koji može da predstavi svojstva svih jezika.⁵²

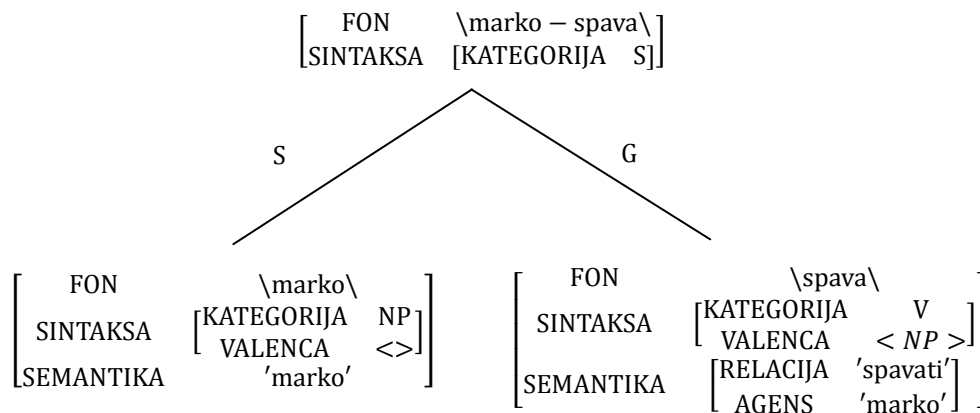
⁴⁸ Za više detalja o LFG upućujemo na (Abeillé, 2007; Dalrymple, 2001; Kaplan & Bresnan, 1982)

⁴⁹ PRED kao funkcija označava glagolski predikat, a kao obeležje postoji u svakoj frazi i za vrednost uzima leksemu koja nosi datu frazu.

⁵⁰ Unutar šiljastih zagrada, neposredno iza glagola, daje se njegov supkategorizacioni okvir. U slučaju glagola *spavati* u taj okvir ulazi samo subjekat.

⁵¹ engl. *head*

⁵² Za više detalja o HPSG upućujemo na (Abeillé, 2007; Levine & Meurers, 2006; Pollard & Sag, 1994)



Slika 7 Struktura rečenice *Marko spava.* u formalizmu HPSG

Formalizam TAG, kao treća dominantna unifikaciona gramatika, je tema ovog rada, i kao takav će biti detaljno predstavljen u narednom poglavlju.

2.4 Gramatika adjungovanih stabala (TAG)

2.4.1 Definicija gramatike

Gramatika adjungovanih stabala TAG začeta je sedamdesetih godina prošlog veka, najpre kao matematički model (Joshi et al., 1975). Leksikalizacijom, odnosno povezivanjem formalno definisanih struktura s leksikonom, ova gramatika je postala upotrebljiva za opisivanje konkretnih prirodnih jezika. Leksikalizovana verzija gramatike ili LTAG (Schabes, Abeillé, & Joshi, 1988; Schabes, 1990) je ono na šta nadalje u radu mislimo kada koristimo naziv TAG.

Specifičnost ove gramatike ogleda se u činjenici da TAG ne operiše niskama, kao ostale gramatike opisane u ovom radu, već stablima. Iz tog razloga postoji potreba da damo njenu formalnu definiciju.

Formalno, TAG je kvintet (Σ, NT, I, A, S) gde je:

- Σ konačan skup terminalnih simbola
- NT konačan skup neterminalnih simbola takav da $\Sigma \cap NT = \emptyset$

- I konačan skup konačnih stabala koja se nazivaju *inicijalna stabla*, sa sledećim karakteristikama:
 - njihovi unutrašnji čvorovi su obeleženi neterminalnim simbolima;
 - čvorovi na granici inicijalnih stabala su terminalni ili neterminalni; neterminalni čvorovi su obeležni za *zamenu*⁵³ strelicom nadole (\downarrow), sem *sidra*⁵⁴ koje se obeležava romбом (\diamond).

Inicijalna stabla se konvencionalno obeležavaju simbolom α (Slika 8).

- A konačan skup konačnih stabala koja se nazivaju *pomoćna stabla*, sa sledećim karakteristikama:
 - njihovi unutrašnji čvorovi su obeleženi neterminalnim simbolima
 - čvorovi na granici pomoćnih stabala su terminalni ili neterminalni; neterminalni čvorovi su obeleženi za zamenu, sem *priključnog čvora*⁵⁵ koji se obeležava asteriskom (*). Oznaka priključnog čvora mora biti identična oznaci korenog čvora stabla.

Inicijalna stabla se konvencionalno obeležavaju simbolom β . (Slika 8)

- S je istaknuti neterminalni simbol takav da je $S \in NT$.

Slika 8 prikazuje model inicijalnog stabla za rečenicu i pomoćnog stabla za pridevsku frazu u svom neleksikalizovanom obliku. Neleksikalizovan oblik podrazumeva da stablo nije povezano sa konkretnom leksemom, odnosno da stablo predstavlja samo opštu shemu date konstrukcije.

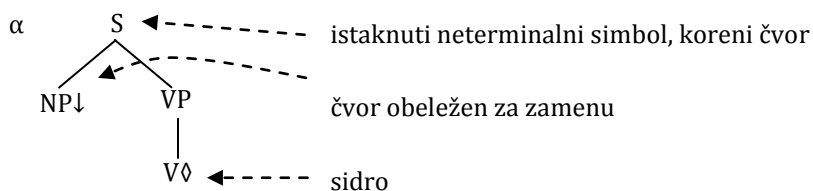
Definicija ove gramatike donosi niz potpuno novih pojmova – inicijalna i pomoćna stabla, zamena, priključivanje, sidro i priključni čvor – koje ćemo jedan po jedan razjasniti u narednim poglavljima.

⁵³ engl. *substitution*

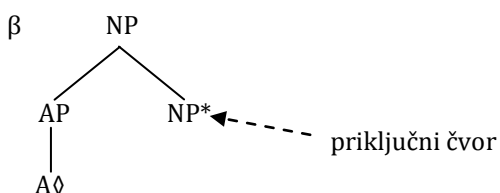
⁵⁴ engl. *anchor*

⁵⁵ engl. *foot node*

Inicijalno stablo



Pomoćno stablo

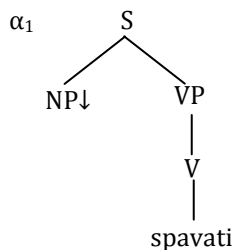


Slika 8 Inicijalno i pomoćno stablo LTAG gramatike

2.4.2 Stabla u TAG

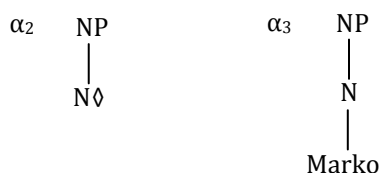
TAG pravi razliku između dva tipa stabala:

1. Inicijalna stabla odgovaraju minimalnim nerekurzivnim jezičkim strukturama. Za stablo čije je sidro glagol u funkciji predikata, minimalna jezička struktura se sastoji od svih obaveznih argumenata datog glagola. Kako primer inicijalnog stabla (Slika 8) sadrži samo jedan dodatni čvor pored glagolskog čvora, ovo stablo je ispravna reprezentacija isključivo neprelaznih glagola kakav je glagol *spavati* (Slika 9).



Slika 9 Inicijalno stablo za glagol *spavati*

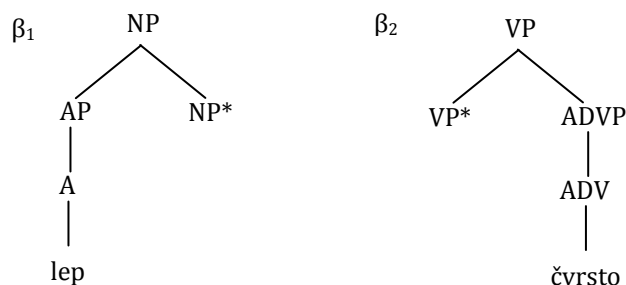
Inicijalna stabla se koriste i za predstavljanje manjih struktura od rečenice, odnosno za predstavljanje fraza. Slika 10 prikazuje stabla za imeničku frazu. Stablo α_2 je neleksikalizovano, i kao takvo šablon za imeničku frazu, dok je stablo α_3 leksikalizovano – povezano sa konkretnom leksemom *Marko*.



Slika 10 Inicijalna stablo za imenice – α_2 opšte stablo, α_3 stablo za imenicu *Marko*

To da inicijalna stabla odgovaraju nerekurzivnim konstrukcijama znači da se npr. nikada ne mogu javiti u funkciji modifikatora u rečenici.

2. Pomoćna stabla odgovaraju minimalnim rekurzivnim jezičkim strukturama i najčešće se koriste kao modifikatori. Slika 11 prikazuje pomoćno stablo β_1 za pridev u funkciji modifikatora (nastalo leksikalizacijom stabla β , Slika 8) i stablo β_2 za prilog u funkciji modifikatora⁵⁶.



Slika 11 Pomoćno stablo β_1 za pridev *lep* i pomoćno stablo β_2 za prilog *čvrsto*.

⁵⁶ Prilog se nalazi sa desne strane glagola umesto tipične, leve, da bi se pokazalo da postoje različiti tipovi pomoćnih stabala.

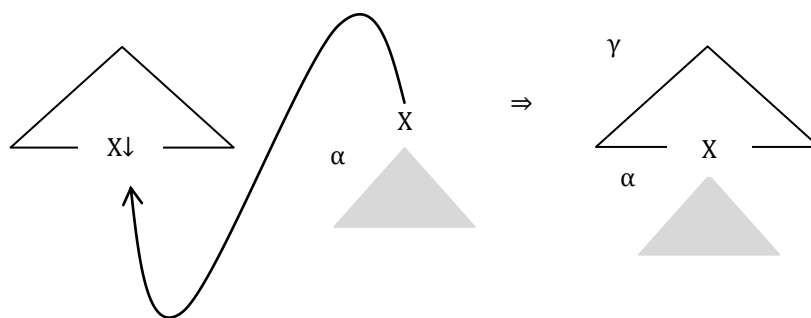
Stablo β_1 na istoj slici je primer levog pomoćnog stabla, kako se njegov priključni čvor nalazi sa desne strane, dok je stablo β_2 primer desnog pomoćnog stabla, kako se njegov priključni čvor nalazi sa leve strane.

Skup svih inicijalnih i pomoćnih stabala ($I \cup A$) čini *elementarna stabla*. Svako stablo se može imenovati po svom korenom čvoru; tako su stabla α i α_1 stabla tipa S, stabla $\alpha_2, \alpha_3, \beta$ i β_1 stabla tipa NP, i stablo β_2 stablo tipa VP.

2.4.3 Operacije u TAG

Kombinovanje stabala se može izvršiti pomoću jedne od dve operacija koje propisuje TAG:

1. Zamena predstavlja proizvod leksikalizacije TAG i kao takva, zadužena je za unošenje leksičkih elemenata u stablo. Može se odvijati samo na neterminalnim čvorovima na granici stabala, koji po konvenciji nose oznaku \downarrow ⁵⁷. Na jednom čvoru se zamena može izvršiti samo jednom⁵⁸, pri čemu se on se zamenjuje korenim čvorom stabla koje kao svoje sidro ima leksemu⁵⁹. Vrsta čvorova koji se spajaju mora biti ista – čvor $NP\downarrow$ može se zameniti jedino stablom čiji je korenim čvor NP. Za zamenu se mogu iskoristiti samo inicijalna stabla ili ona koja su izvedena od inicijalnih stabala. Na čvoru koji je obeležen za zamenu nije moguće izvršiti drugu TAG operaciju – pripajanje. Rezultat zamene je takođe inicijalno stablo, koje se još naziva i izvedeno stablo i po konvenciji obeležava grčkim slovom γ (Slika 12).



Slika 12 Shematski prikaz operacije zamene.

⁵⁷ Na čvorovima obeleženim rombom, sidrima, ne vrši se zamena. Stabla sa sidrom koriste se samo kao opšte sheme konkretnih leksikalizovanih stabala.

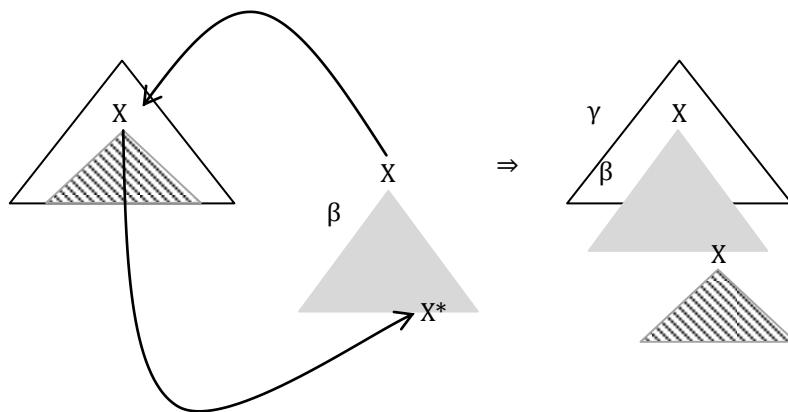
⁵⁸ Odnosno operacija zamene je nerekurzivna.

⁵⁹ Za konkretan primer operacije zamene na srpskom jeziku videti nastavak testa (Slika 15).

Gramatika koja se sastoji samo od inicijalnih stabala i operacije zamene je po slabom generativnom kapacitetu ekvivalentna kontekstno slobodnim gramatikama. Ono što ovu gramatiku čini umereno kontekstno osetljivom, i samim tim prikladnom za OPJ, je sledeća operacija – operacija pripajanja.

2. Pripajanje je operacija specifična za TAG koja ovoj gramatici obezbeđuje rekurzivnost. Pripajanje podrazumeva spajanje pomoćnog stabla i bilo kog drugog stabla, uključujući i pomoćno. Već je napomenuto da je oznaka korenog čvora i priključnog čvora u pomoćnim stablima ista. Da bi pripajanje moglo da se izvrši, u stablu na kome će se izvršiti pripajanje mora postojati čvor koji nosi istu takvu oznaku. Pored tog osnovnog uslova, kao mehanizam ograničenja rekurzije, za svaki čvor je moguće definisati i ograničenja na pripajanje. Tako pripajanje može biti:

- 1) **Selektivno (SP)** – samo članovi skupa $T \subseteq A$ (podskup skupa pomoćnih stabala) mogu biti pripojeni na datom čvoru. Pripajanje u ovom slučaju nije obavezno.
- 2) **Nulto (NP)** – zabranjuje bilo kakvo pripajanje na datom čvoru.
- 3) **Obavezno (OP)** – pomoćno stablo koje je član skupa $T \subseteq A$ mora biti pripojeno na datom čvoru.

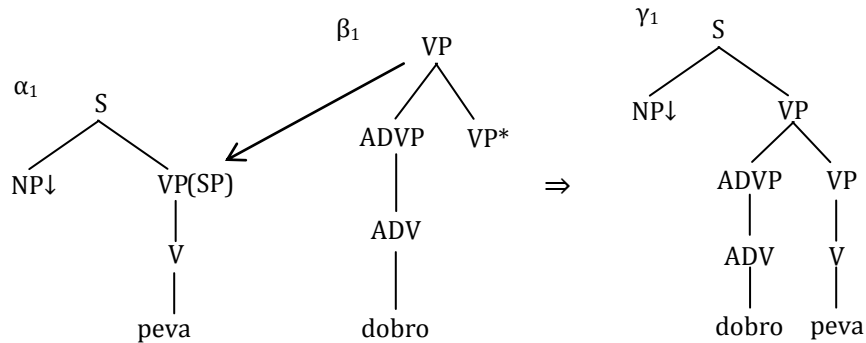


Slika 13 Shematski prikaz operacije pripajanja

Pripajanje može biti izvedeno na bilo kom čvoru stabla sem na onom koji je obeležen za zamenu ili obeležen oznakom NP (nulto pripajanje). Rezultat pripajanja je takođe izvedeno stablo, takođe označeno grčkim slovom γ (Slika 13).

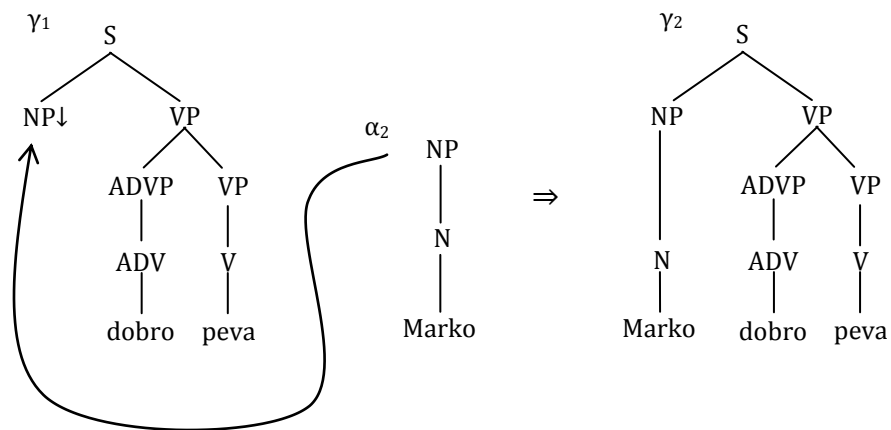
Slika 14 predstavlja primer operacije pripajanja za srpski jezik. U pitanju je pripajanje pomoćnog stabla za prilog β_1 na rečenično stablo α_1 . β_1 je ovde primer levog

pomoćnog stabla. U rečeničnom stablu α_1 čvor VP je obeležen za selektivno pripajanje⁶⁰, što znači da mu se može, ali ne mora, pripojiti odgovarajuće pomoćno stablo, odnosno da je rečenica kompletna i bez njega. Rezultat pripajanja je izvedeno inicijalno stablo γ_1 .



Slika 14 Primer operacije pripajanja na srpskom jeziku

Slika 15 daje primer za operaciju zamene. Ovde je zamena izvršena u prethodno izvedenom stablu γ_1 (Slika 14). Rezultat zamene je izvedeno stablo γ_2 .



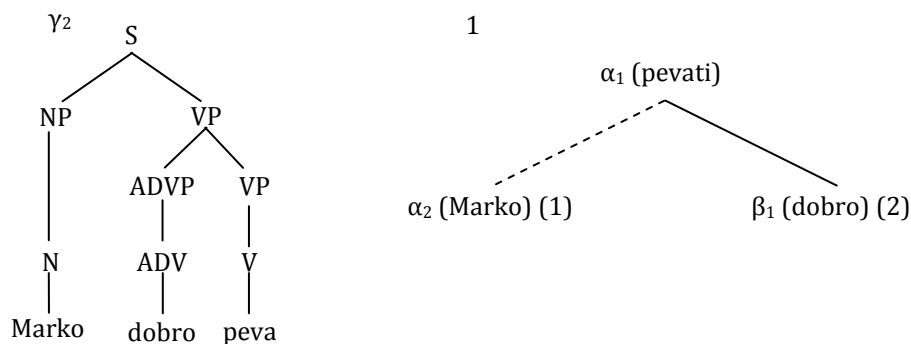
Slika 15 Primer operacije zamene na srpskom jeziku

U TAG se pravi jasna razlika između *izvedenih stabala* i *stabala izvođenja*. Kako iz izvedenog stabla – stabla nastalog spajanjem elementarnih stabala – nije moguće zaključiti tačno kako je do njegovog izvođenja došlo, za svako izvedeno stablo postoji i odgovarajuće

⁶⁰ Oznaka SP se ovde koristi samo u ilustrativne svrhe i zapravo se nikad zaista ne sreće u stablima. Ograničenja na pripajanje će nadalje biti definisana drugačijim sredstvima. Za detalje videti poglavlje o FBLTAG (2.4.5.1)

stablo izvođenja. Kao što i ime kaže, stabla izvođenja su takođe stabla, ali njihovi čvorovi predstavljaju stabla od kojih je izvedena data rečenica, dok grane predstavljaju operacije kojima su data stabla uneta u konačnu strukturu. Konvencija koja se obično prati jeste da se u stablima izvođenja zamena prikazuje isprekidanom linijom a pripajanje punom linijom.

Čvorovi stabala izvođenja obeleženi su adresama stabala, koje se sastoje od tri elementa – imena stabla (α_x , β_x ili γ_x), imena čvora (leksema), i broja čvora. Broj čvora se određuje počevši od korenog čvora izvedenog stabla, koji je obeležen kao 0. Čvorovi neposredno ispod korenog čvora obeležavaju se prema svom linearnom rasporedu (1, 2...) gledano s leva na desno, a čvorovi ispod svakog od njih brojem svog natčvora i brojem koji označava njihov linearni redosled unutar datog čvora (1.1., 1.2., 2.1., 2.2....).

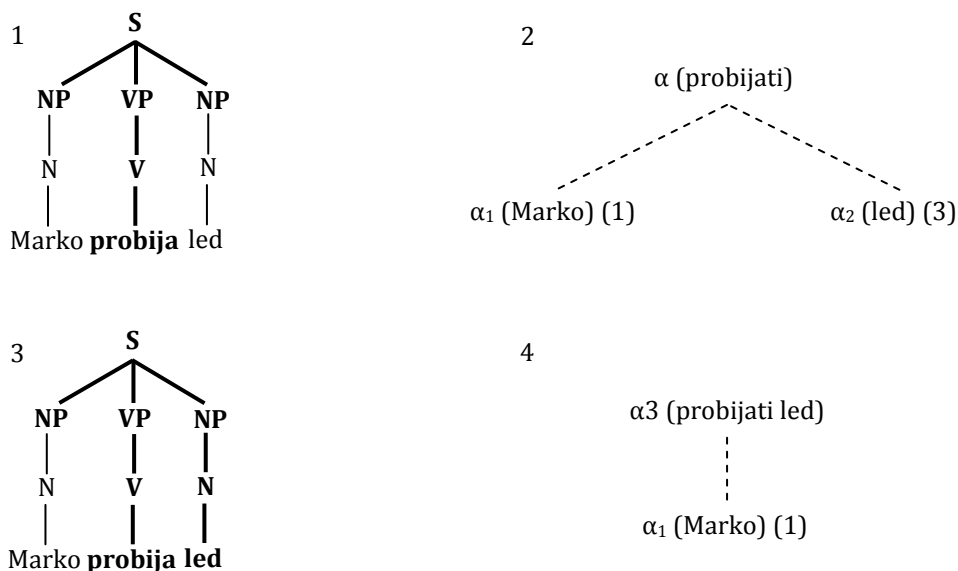


Slika 16 Izvedeno stablo i njegovo stablo izvođenja

Gledajući stablo γ_2 (Slika 16), njegovo stablo izvođenja, obeleženo brojem 1, pokazuje da je početno stablo na kome su izvođenja izvršena bilo stablo α_1 (Slika 14), i da su nad njime izvršene operacije pripajanja na adresi 2 stabla α_1 (čvor VP), i zamene na adresi 1 stabla α_1 (čvor NP). Stablo izvođenja ne definiše redosled operacija.

Stabla izvođenja mogu se iskoristiti za uspešno razrešavanje semantičke dvoznačnosti u situacijama kada je struktura stabala ista, kao kod idioma. Naime, idiomi se unutar TAG tretiraju kao jedinstvena jedinica u leksikonu. U slučaju rečenice koja sadrži idiom – *Marko probija led*. – u osnovnom čitanju su u inicijalno stablo *probija*, ovde predstavljeno kao podebljani deo na već izvedenom stablu 1 (Slika 17) (nazovimo ovo stablo α) zamenom uneta inicijalna stabla za *Marko* (α_1) i *led* (α_2), što je struktorno

prikazano izvedenim stablom 1 i odgovarajućim stablom izvođenja (stablo 2). U slučaju idiomatičnog čitanja, u inicijalno stablo idioma (podebljani deo izvedenog stabla 3 nazovimo ovo stablo α_3) zamenom je uneto samo inicijalno stablo za *Marko* (α_1). Ovakva struktura stabla za idiome – jedno inicijalno stablo sa dva sidra, po jedno za svaku leksemu idioma (*probija* i *led*) umesto odvojenih stabala za svaku od ovih leksema, predložena je u (Abeillé, 1990)⁶¹.



Slika 17 Izvedena stabla i stabla izvođenja za dva tumačenja rečenice *Marko probija led*.

Stabla izvođenja omogućavaju jasno prikazivanje odnosa zavisnosti među rečeničnim delovima, čak i onda kada se čvorovi u izvedenom stablu više ne nalaze neposredno jedan do drugog (Slika 16, direktan odnos između *Marko* i *peva*).

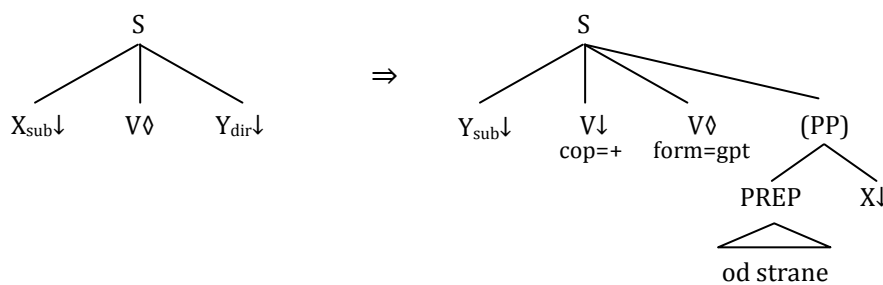
2.4.4 Leksička pravila i porodice stabala

U poglavlju o unifikacionim gramatikama pomenuli smo leksička pravila kao mehanizam osmišljen da zameni operaciju transformacije. Na ovom mestu ćemo bliže

⁶¹ Kako se u ovom radu nećemo detaljnije baviti idiomima, upućujemo na ovaj rad za njihovu detaljniju analizu.

objasniti njihovu konkretnu realizaciju unutar TAG. Leksička pravila su prvi put primenjena na TAG u (Vijay-Shanker & Schabes, 1992).

Kao što je pomenuto, leksička pravila povezuju osnovnu strukturu u koju može ući jedna leksema (ili grupa leksema) sa svim njenim strukturnim varijantama. Pogledajmo šta ovo znači u praksi. Uzmimo za primer glagol *kupovati*. Osnovna struktura u koju ulazi ovaj glagol je subjekat–predikat–pravi objekat (*Žena kupuje kafu.*), kako *kupovati* zahteva informaciju o onome ko kupuje i onome što se kupuje. U upotrebi, ovaj glagol se može javiti u različitom tipu alternativnih struktura, kao što je recimo pasivna, u obliku participskog pasiva (*Kafa je kupovana.*), pasivna, u obliku reflektivnog pasiva (*Kafa se kupuje.*) ili upitna (*Šta kupuje?*). Ovom prilikom može biti promenjen oblik glagola (kao u pasivu), neki od argumenata mogu biti izostavljeni (participski pasiv, pitanje) ili potpuno uklonjeni iz konstrukcije (refleksivni pasiv), ili im pak može biti promenjena funkcija (participski pasiv). Leksička pravila se koriste da povežu sve ovakve alternativne strukture i istovremeno opišu strukturne promene karakteristične za svaku od njih. Za svaku od alternacija u strukturi definiše se zasebno leksičko pravilo, pri čemu je opšta struktura leksičkog pravila uvek ista. Svako leksičko pravilo se tako sastoji od leve strane pravila, gde je dato stablo osnovne, polazne strukture, i desne strane pravila, gde je dato stablo u kome je došlo do alternacije. Sva leksička pravila su deklarativnog tipa, što znači da predstavljaju model na osnovu koga se procenjuje da li je određena struktura ispravno formirana, a ne proces na osnovu koga se strukture izvode. Slika 18 prikazuje ilustrativno pravilo za participski pasiv⁶².



Slika 18 Model leksičkog pravila za pasiv u srpskom jeziku

⁶² Za detalje potrebne za razumevanje strukture desne strane ovog pravila upućujemo na poglavlje 3.6.1.1. Na ovom mestu ćemo samo reći da ako gledamo pasivnu rečenicu *Kafa je kupovana (od strane žene)*, na desnoj strani pravila čvor V koji je obeležen za prepisivanje stoji na mestu pomoćnog glagola *je* i nosi obeležje *cop* – kopula. Sidro stabla je glagol *kupovana*, s obeležjem oblika glagola: *form* i njegovom vrednošću: *gpt* – glagolski pridev trpni. Agentivna odredba je obeležena kao fakultativna stavljanjem PP u zagradu, dok je složeni predlog *od strane* označen trouglom, što znači da njegova struktura neće biti detaljno prikazana.

Sva stabla koja su međusobno povezana leksičkim pravilima organizovana su u porodicu stabala. Na taj način je pri definisanju rečeničnih modela dovoljno navesti strukturu predstavnika porodice tj. osnovnu rečeničnu strukturu, i leksička pravila koja se mogu primeniti na tu strukturu, čime se dobijaju i svi ostali članovi porodice.

Leksička pravila se mogu podeliti na dva tipa koja se značajno razlikuju:

1. **Pravila redistribucije** utiču na supkategorizacioni okvir glagola – dodaju ili brišu broj argumenata glagola, menjaju njihove kategorije, kao i njihove funkcije. Ova pravila mogu dodati i nova sidra i izmeniti obeležja originalnog sidra. Primeri pravila redistribucije su:

- pravila za pasiv:
 - participski – *Žena kupuje kafu.* ⇒ *Kafa je kupljena.*
 - refleksivni – *Žena kupuje kafu.* ⇒ *Kafa se kupuje.*
- pravilo za obezličavanje – *Ljudi putuju* ⇒ *Putuje se.*
- refleksivizacija – *Žena češlja bebu.* ⇒ *Žena se češlja.*
- relativizacija – *Vidim čoveka.* ⇒ *Čovek koga vidim.*

U poseban tip pravila redistribucije spadaju **pravila izmeštanja**, kojima se opisuje odnos između stabala u kojima je došlo do izmeštanja argumenata sa njihove uobičajene pozicije. Pravila izmeštanja se koriste da opišu strukturu pitanja:

- pitanje putem izmeštanja dopune – *Žena kupuje kafu.* ⇒ *Šta žena kupuje?*
- subjekatsko pitanje – *Žena kupuje kafu.* ⇒ *Ko kupuje kafu?*

Iako su oba tipa pitanja svrstana u ovu grupu, njihova struktura se međusobno značajno razlikuje, posebno zato što, tehnički, u slučaju subjekatskog pitanja izmeštanje i ne postoji. Pitanja u koja je uneta upitna reč – *Zašto žena kupuje kafu?* – ne potpadaju pod leksička pravila⁶³.

2. **Pravila realizacije** ne menjaju supkategorizacioni okvir glagola, kao ni funkcije glagolskih argumenata, već utiču samo na njihov redosled u rečenici, na to da li će argumenti biti realizovani ili na njihov pojavni oblik, i definišu vrednosti njihovih već postojećih obeležja. Primeri pravila realizacije su pravila za obradu:

- varijacija u redu reči – *Žena kupuje kafu.* ⇒ *Kupuje žena kafu.*

⁶³ Strukturu pravila izmeštanja detaljnije obrađujemo u poglavlju 3.6.1.2.

- izostavljanja argumenata – *Žena kupuje kafu.* ⇒ *Kupuje kafu.*
- kliticizacije – *Žena kupuje kafu.* ⇒ *Žena je kupuje.*⁶⁴

Kao pravila koja određuju broj argumenata u izmenjenoj strukturi, pravila redistribucije se moraju primeniti prva, pre pravila realizacije. Pravila realizacije se primenjuju na pojedinačne argumente, dok pravila redistribucije utiču na čitavu strukturu ili više elemenata istovremeno. Za razliku od pravila redistribucije, koja podležu strogim leksičkim ograničenjima, pravila realizacije nemaju veliki broj izuzetaka.

Dobro formirano elementarno stablo je rezultat od nula do dva leksička pravila redistribucije i od nula do dva leksička pravila realizacije za svaki od argumenata sa rečeničnom funkcijom⁶⁵. Elementarno stablo na koje nije primenjeno nijedno leksičko pravilo i koje se sastoji od kanonske realizacije svih argumenata predikata, u njihovoj kanonskoj funkciji, zove se *kanonsko elementarno stablo*.

2.4.5 Tipovi TAG

Na ovom mestu ćemo napraviti kratak pregled tipova TAG gramatika, kako je nastavak rada izrađen upravo na jednoj od tih varijanti. U toku godina razvijen je niz gramatika koje dosad predstavljenu osnovu TAG gramatika neznatno menjaju ili nadograđuju. Neke od tih verzija su:

1. **FBLTAG**⁶⁶ (Vijay-Shanker & Joshi 1988; Vijay-Shanker & Joshi 1991).

LTAG zasnovana na obeležjima specifična je po tome što svakom čvoru u stablu pridodaje strukturu obeležja. Ova struktura obeležja se od onih koja smo mogli da vidimo u poglavlju 2.3 razlikuje po tome što je dupla, pa tako sadrži gornja obeležja, koja pokazuju odnos datog čvora prema sebi nadređenoj strukturi i donja obeležja, koja pokazuju odnos datog čvora prema sebi podređenoj strukturi. FBLTAG propisuje operaciju unifikacije koja odobrava ili zabranjuje spajanje stabala s obzirom na

⁶⁴ Za strukturni prikaz ovih leksičkih pravila upućujemo na poglavlje 3.6.1.3.

⁶⁵ Ovaj broj se razlikuje od jezika do jezika, tako da su ovo podaci za srpski. Maksimalan broj primena pravila redistribucije javlja se u slučaju primene pravila za pitanje na pravilo za pasiv. Kod pravila realizacije, maksimalan broj primena odnosi se na primenu pravila za promenu mesta reči nakon kliticizacije.

⁶⁶ engl. *Feature-Based Lexicalized Tree Adjoining Grammar*

usklađenost njihovih obeležja. Ove karakteristike FTAG će biti detaljno prikazane u narednim poglavljima kako je to gramatika koju ćemo koristiti za izradu formalne gramatike srpskog jezika.

FTAG gramatike postoje za engleski (XTAG Research Group, 2001), francuski (Abeillé, 2002), italijanski (Candito, 1999), korejski (Han, Yoon, Kim, & Palmer, 2000) i nemački (kao varijanta FBLTAG) (Gerdes, 2002)

2. **MCTAG**⁶⁷ (Weir, 1988).

Specifičnost multikomponentne TAG ogleda se u tome što grupiše stabla u setove stabala koja se pri pripajanju sva pripajaju odjednom. Postoji nekoliko tipova MCTAG od kojih se dva osnovna razlikuju po tome da li se pripajanje seta stabala vrši na jednom stablu: TL-MCTAG⁶⁸ ili na setu stabala: SL-MCTAG⁶⁹. Dok TL-MCTAG generiše jezik koji je istog slabog i jakog generativnog kapaciteta kao TAG, i slab i jak generativni kapacitet SL-MCTAG je jači nego TAG⁷⁰. MCTAG i posebno njene varijante s jakim generativnim kapacitetom pokazale su se pogodne za tretman specifičnog ponašanja glagola u nemačkom jeziku.

Gramatike bazirane na MCTAG postoje za nemački (Kallmeyer, Lichte, Maier, Parmentier, & Dellert, 2008; Kallmeyer & Yoon, 2004; Rambow & Lee, 1994) i korejski (Rambow & Lee, 1994).

3. **Sinhrona TAG**⁷¹ (Shieber & Schabes 1990).

Sinhrona TAG je napravljena sa ciljem da poveže određeni sintaksički opis predstavljen pomoću TAG sa njegovom semantičkom reprezentacijom ili sa njegovim prevodnim ekvivalentom na nekom drugom jeziku, koji je takođe dat u formi TAG. Ta dva opisa su međusobno sinhronizovana.

4. **Probabilistička TAG** (Resnik, 1992).

Ova TAG gramatika povezuje verovatnoću sa svakim stablom u gramatici, na osnovu čega izračunava verovatnoću određene derivacije.

⁶⁷ engl. *Multi-Component Tree Adjoining Grammar*

⁶⁸ engl. *tree-local*

⁶⁹ engl. *set-local*

⁷⁰ Za pregled opštih karakteristika MCTAG i pomenutih tipova, upućujemo dalje na (Schuler, Chiang, & Dras, 2000).

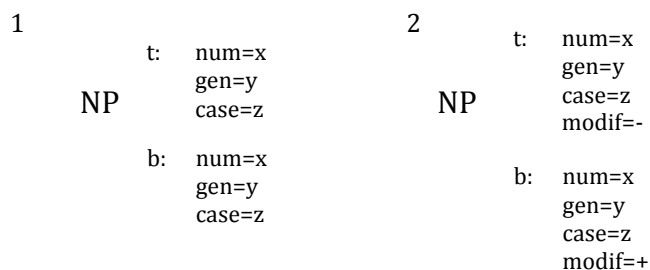
⁷¹ engl. *Synchronous TAG*

2.4.5.1 FBLTAG

FBLTAG je TAG proširena strukturom obeležja i stavljena u unifikacioni okvir. U ovom poglavlju ćemo se baviti implementacijom ova dva pojma u TAG gramatiku. Postoje dva glavna razloga za unošenje obeležja u TAG:

- stavljanje jasnijih ograničenja na izbor struktura u koje mogu ući lekseme. U FBLTAG svaka leksema nosi obeležja koja ograničavaju strukture u koje može ući. Tako se za prelazan glagol (kao npr. *znati*), za koji se tradicionalno vezuje prelazna struktura, može pomoću obeležja *pasiv=-* označiti da ne ulazi u pasivnu konstrukciju, odnosno da se na njega ne može primeniti leksičko pravilo za participski pasiv.
- stavljanje ograničenja na izvođenja koja se mogu izvršiti. U strukturu obogaćenu obeležjima mogu biti uneta samo stabla koja su po obeležjima kompatibilna sa datim stablom. Pomoću obeležja je isto tako moguće formalizovati ograničenja na priključivanje. U nastavku teksta predstavimo taj mehanizam.

U FBLTAG struktura obeležja se javlja kao dodatna informacija uz čvorove u elementarnom stablu. FBLTAG ne koristi složenu strukturu obeležja⁷², ali kao svoju specifičnost uvodi dve strukture obeležja za čvorove na kojima se može vršiti pripajanje (Slika 19)⁷³.



Slika 19 Dve strukture obeležja na primeru imeničke fraze

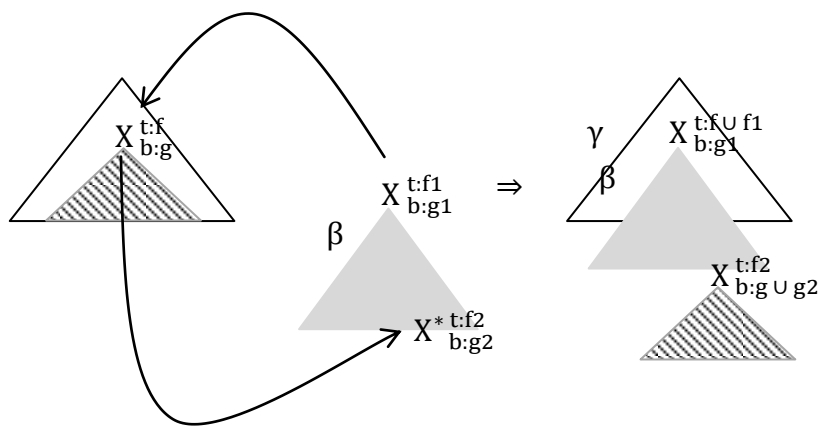
Postojanje dve strukture obeležja zasniva se na činjenici da svaki čvor ima:

⁷² O složenoj strukturi obeležja bilo je reči u poglavlju 2.3.

⁷³ Nadalje u tekstu ćemo za obeležja koristiti oznake koje mahom koristimo i u samoj gramatici (detaljnije u poglavlju 3.1). Na slici 19, *num* označava broj, *gen* rod, *case* padež a *modif* zahtev za modifikatorom („+“ označava zahtev za modifikatorom). Vrednosti *x*, *y* i *z* koriste se samo da označe jednakost obeležja, koja god da je njihova vrednost. Od svih obeležja jedino je obeležje *modif* fiktivno i koristimo ga samo u ilustrativne svrhe.

- Odnos prema nadstablu, koji predstavlja pogled na čvor odozgo. Ovaj odnos je prikazan gornjom strukturom obeležja koju ćemo obeležavati sa t^{74} .
- Odnos prema podređenim čvorovima, koji predstavlja pogled na čvor odozdo. Ovaj odnos je prikazan donjom strukturom obeležja i obeležavaćemo ga sa b^{75} .

Ipak, osnovni razlog za primenu duple strukture obeležja leži u prirodi operacije pripajanja. Ono što se prilikom pripajanja dešava jeste da se čvor na kome se vrši pripajanje (Slika 20, čvor $X_{b:g}^{t:f}$) deli na dva dela – jedan koji se spaja sa korenim čvorom pomoćnog stabla ($X_{b:g_1}^{t:f_1}$) i drugi koji se spaja sa priključnim čvorom pomoćnog stabla ($X_{b:g_2}^{t:f_2}$). Dvostruka struktura obeležja na tom čvoru omogućava mu ovakvo deljenje. Tom prilikom se gornje obeležje čvora na kome se vrši priključivanje ($t:f$) spaja (unifikuje)⁷⁶ sa gornjim obeležjem korenog čvora pomoćnog stabla ($t:f_1$), a donje obeležje čvora na kome se vrši priključivanje ($b:g$) spaja sa donjim obeležjem priključnog čvora ($b:g_2$). Da bi do spajanja moglo da dođe, potrebno je proveriti da li su vrednosti navedenih obeležja međusobno kompatibilne, odnosno iste. Ukoliko jesu, dolazi do njihove unifikacije.



Slika 20 Shematski prikaz unifikacije struktura obeležja pri pripajanju

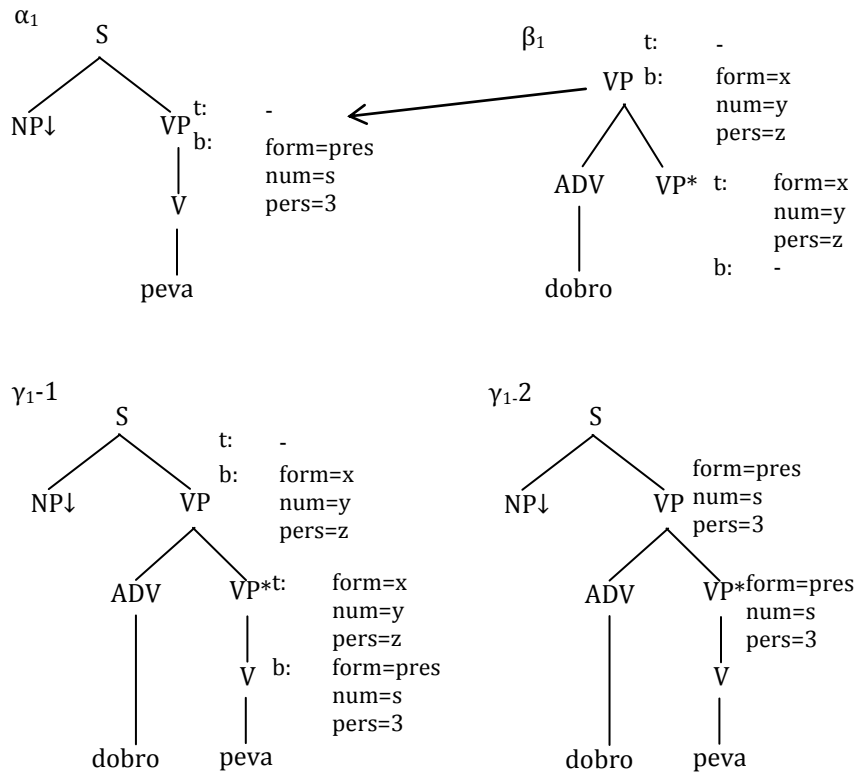
Na čvorovima na kojima u toku izvođenja ne dođe do pripajanja, t i b svojstva datog čvora se međusobno unifikuju. U konačnom, izvedenom stablu svi čvorovi su međusobno unifikovani i ne postoji više nijedan čvor sa duplom strukturom obeležja.

⁷⁴ t od engl. *top* (vrh)

⁷⁵ b od engl. *bottom* (dno)

⁷⁶ Pojam unifikacije koristimo u terminima opisanim u poglavlju 2.3.

Slika 21 prikazuje strukture obeležja za čvorove VP u inicijalnom (α_1) i pomoćnom stablu (β_1) za srpski jezik. Stablo γ_1-1 predstavlja prvi korak nakon pripajanja na čvor VP u stablu α , u kome još uvek nije došlo do unifikacije obeležja. Stablo γ_1-2 prikazuje izgled izvedenog stabla na kome su unifikovana sva obeležja na čvorovima VP.

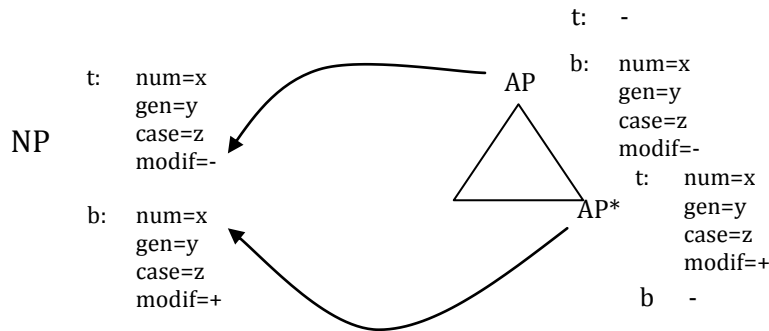


Slika 21 Primer za unifikaciju pri pripajanju

Način na koji se formalizuju ograničenja na pripajanje je sledeći. Ako se vratimo na sliku 19, možemo videti da čvor 1 ima kompatibilna gornja i donja obeležja. Ovo omogućava da se, ukoliko ne dođe do pripajanja na ovom čvoru, njegova t i b obeležja unifikuju. Ipak, ukoliko se pojavi stablo čiji koreni i priključni čvor nose obeležja s istim vrednostima (num=x, gen=y, case=z), ono se može pripojiti na ovom čvoru. Na taj način se formalno reguliše ograničenje selektivnog pripajanja (SP).

U slučaju stabla 2, možemo videti da njegova obeležja nisu kompatibilna – vrednost obeležja modif se razlikuje. Kako obeležje modif ima pozitivnu vrednost u donjoj strukturi obeležja, to označava da leksema zahteva modifikaciju u obliku neke vrste atributa. Negativna vrednost u gornjoj strukturi obeležja označava da modifikator ne postoji u datoj

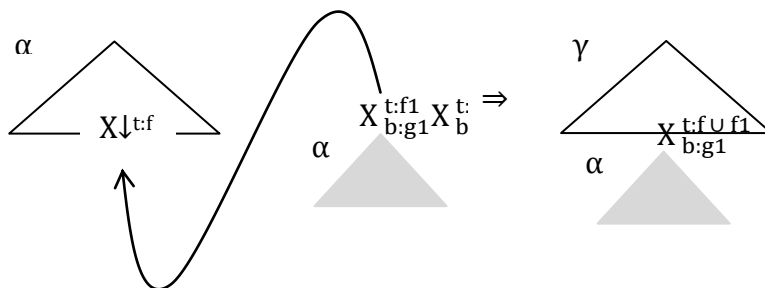
strukturi. Ova obeležja se zbog razlike u vrednostima ne mogu unifikovati, tako da se na ovom čvoru mora pripojiti stablo koje će zadovoljiti oba uslova. Slika 22 prikazuje takvo stablo, zajedno sa stablom 2 sa slike 19. Na ovaj način, međusobnom neusklađenošću obeležja na istom čvoru regulisano je ograničenje obaveznog pripajanja (OP).



Slika 22 Formalizacija ograničenja obaveznog pripajanja na primeru pripajanja pridevske na imeničku frazu

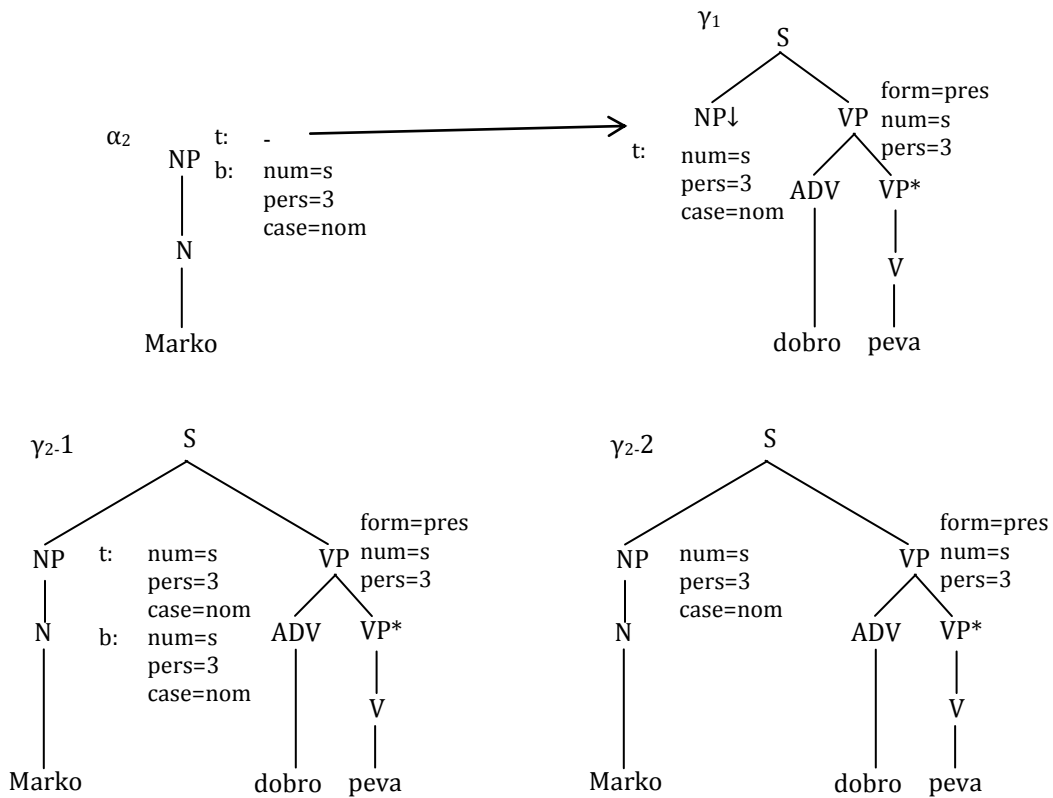
Čvorovima koji nose obeležje NP (nulto pripajanje) kao što je priključni čvor, dodeljuje se samo gornja struktura obeležja, čime se sprečava njegovo deljenje, neophodno za proces pripajanja.

Čvoru obeleženom za zamenu se tako takođe dodeljuje se samo gornja struktura obeležja (Slika 23, obeležje t:f). S druge strane, koreni čvor stabla koje služi kao zamena ima i obeležje t i b. Prilikom zamene, potrebno je izvršiti unifikaciju t obeležja oba stabla (t:f i t:f₁), a zatim i međusobnu unifikaciju t i b obeležja (t:f ∪ f₁ i b:g₁).



Slika 23 Shematski prikaz unifikacije struktura obeležja pri zameni

Slika 24 prikazuje proces unifikacije pri operaciji zamene na konkretnom primeru. Stablo γ_2-1 prikazuje izgled izvedenog stabla u kome je čvor NP zamenjen inicijalnim stablom α_2 , ali u kome još uvek nije izvršena unifikacija gornjih i donjih obeležja. U stablu γ_2-2 su unifikovana sva gornja i donja obeležja.



Slika 24 Primer za unifikaciju pri zameni

2.4.6 Relevantnost FBLTAG za obradu prirodnih jezika

TAG ima tri obeležja koja je čine prikladnom za OPJ, posebno za nekonfiguracione jezike kakav je srpski:

1. Prošireni domen lokalnosti⁷⁷

Ovo svojstvo proizlazi iz same prirode TAG kao formalizma koji za osnovnu jedinicu ima elementarno stablo. Dobro formirano predikatsko elementarno stablo zahteva da se unutar njega nalaze svi argumenti predikata. Prateći ovaj princip, u TAG se rečenice sa

⁷⁷ engl. *Extended domain of locality*

izmeštenim argumentima definišu ne kao izvedena, već kao primarna elementarna stabla. Na ovaj način se postiže to da su odnosi među svim argumentima glagola, bez obzira na to gde se nalaze u odnosu na svoju osnovu poziciju, i dalje definisani lokalno, odnosno u okviru istog stabla. Iz ovog razloga u TAG ne postoji potreba za operacijom transformacije koja bi generisala izmenjene niske i tražila način da izmeštenoj niski dodeli odgovarajuća obeležja.

2. Rekurzija čuva domen zavisnosti⁷⁸

Rekurzija u TAG obezbeđuje se primenom operacije pripajanja. Pripajanje se vrši na elementarnom stablu (lokalnom domenu zavisnosti) u koje se na taj način unosi nov element. Budući da je zavisnost već jasno definisana među argumentima elementarnog stabla, unošenje novog stabla ni na koji način ne menja njihov odnos – skup njihovih deljenih obeležja i dalje je onaj iz početne elementarne strukture. U tom smislu se zapravo i ne može govoriti o udaljenosti argumenata, koji su i pored linearne udaljenosti uvek lokalni. Rečenica *Šta Marko misli da Ivan voli*, u kojoj se argument *šta* tipično smatra izmeštenim iz zavisne rečenice *Ivan voli*, u TAG se rekonstruiše kao elementarna rečenica *Šta Ivan voli*, u koju je zatim pripajanjem uneta „glavna” rečenica *Marko misli da+zavisna rečenica*.

3. Leksikalizovanost

Zahtev LTAG gramatika da svako stablo bude povezano sa leksikonom čini ovaj formalizam u potpunosti leksikalizovanim. Na ovaj način morfosintaksička i sintaksička svojstva svake lekseme definišu se direktno u leksikonu (Slika 25).

$$\text{probiti} \left[\begin{array}{cc} \text{model} & n0Vn1dir^{79} \\ MS^{80} \text{ obeležja} & \begin{bmatrix} \text{ref} & - \\ \text{asp}^{81} & \text{perf} \end{bmatrix} \\ \text{leksička pravila} & \begin{bmatrix} \text{ppasiv} & + \\ \text{refpasiv} & + \end{bmatrix} \end{array} \right] \text{ led} \left[\begin{array}{cc} \text{model} & n^{82} \\ MS \text{ obeležja} & \begin{bmatrix} \text{gen} & m \\ \text{anim} & - \end{bmatrix} \\ \text{leksička pravila} & \begin{bmatrix} \text{klit} & + \end{bmatrix} \end{array} \right]$$

Slika 25 Leksički unosci za reči *probiti* i *led*

⁷⁸ engl. *Factoring recursion from the domain of dependencies*

⁷⁹ Model obuhvata: imenicu u funkciji subjekta (n_0) i imenicu u funkciji pravog objekta (n_{1dir}). Ovaj glagol može ući u još modela, ali ovde koristimo jedan u ilustrativne svrhe.

⁸⁰ Morfosintaksička

⁸¹ Glagolski vid

⁸² Neglagolske lekseme takođe imaju svoje modele, koji se zasebno prave za svaku od vrsta reči (imenice, pridevi i prilozii). U ovom slučaju je u pitanju model definisan za imenice.

Leksikalizovanost obezbeđuje međusobno razdvajanje leksema koje inače pripadaju istoj porodici. Takođe, leksikalizovanost omogućava olakšano predstavljanje idioma i složenih izraza, koji za svoje metaforičko (ili složeno) značenje imaju odvojen unosak, u okviru kojih su grupisani u isto stablo (Slika 26).

probiti led	model	[n0VN] ⁸³
	MS obeležja V	[ref -]
	MS obeležja N	[asp perf]
	leksička pravila	[gen m]
		[num s]
		[case acc]
		[anim -]
		[ppasiv +]
		[refpasiv +]
		[klit +]

Slika 26 Leksički unosak idioma *probiti led*

Složenost leksikona, kao jedna od ključnih karakteristika koje TAG čine pogodnom za opis i obradu prirodnih jezika, ipak ima i svoje mane. TAG generalno, bez obzira na svoju varijantu, nosi sa sobom značajnu količinu redundantnosti. Unutar leksikona, za svaku leksemu je vezano najmanje jedno, a najčešće niz stabala koja predstavljaju sve obrasce stabala u kojima se leksema može naći. Takođe, ovi isti obrasci stabala se ponavljaju uz sve lekseme koje u njih mogu ući. Na ovaj način, leksikon TAG gramatika vrlo brzo postaje prevelik i skoro nemoguć za održavanje ili dopunjavanje. Zapravo, samo generisanje TAG gramatike pod ovim uslovima predstavlja pretežak zadatak. Zato se u ove svrhe, kao i svrhe održavanja i menjanja TAG gramatika koristi metagramatika, o kojoj će biti više reči u narednom poglavlju.

2.5 Metagramatika

Metagramatika u domenu OPJ predstavlja formalizam koji za cilj ima generalizovano predstavljanje određene formalne gramatike. Iako je metagramatika suštinski formalna gramatika, to nije sintaksička teorija, već logički ili matematički konstrukt napravljen sa

⁸³ U okviru modela veliko slovo označava sidro modela, tj. porodice stabala. U ovom slučaju, sidra su i glagol *probiti* i imenica *led*.

ciljem da sažeto opiše odgovarajuću formalnu gramatiku. U većini slučajeva, ona podrazumeva postojanje dve komponente:

- formalnog jezika kojim se opisuje ciljna formalna gramatika;
- programskog dela (kompilatora) koji na osnovu metagramatike generiše strukture formalne gramatike.

Prva metagramatika za TAG razvijena je za potrebe automatskog generisanja TAG za francuski i italijanski (Candito, 1996, 1999). Na njenim postavkama definisana je metagramatika koju ćemo koristiti za generisanje TAG gramatike srpskog jezika, a koju ćemo predstaviti u narednom poglavlju.

2.5.1 XMG

2.5.1.1 Teorijski principi

Proširiva metagramatika^{84,85} je formalizam kreiran i opisan u radu (Crabbé, 2005) kao jezik za predstavljanje gramatika baziranih na stablima, kao i kompilator za taj jezik. XMG definiše TAG gramatike zahvaljujući njihovoj generalizaciji, koja je dvojaka:

- generalizacija struktura;
- povezivanje alternativa.

Objasnimo šta znači svaki od ovih pojmova.

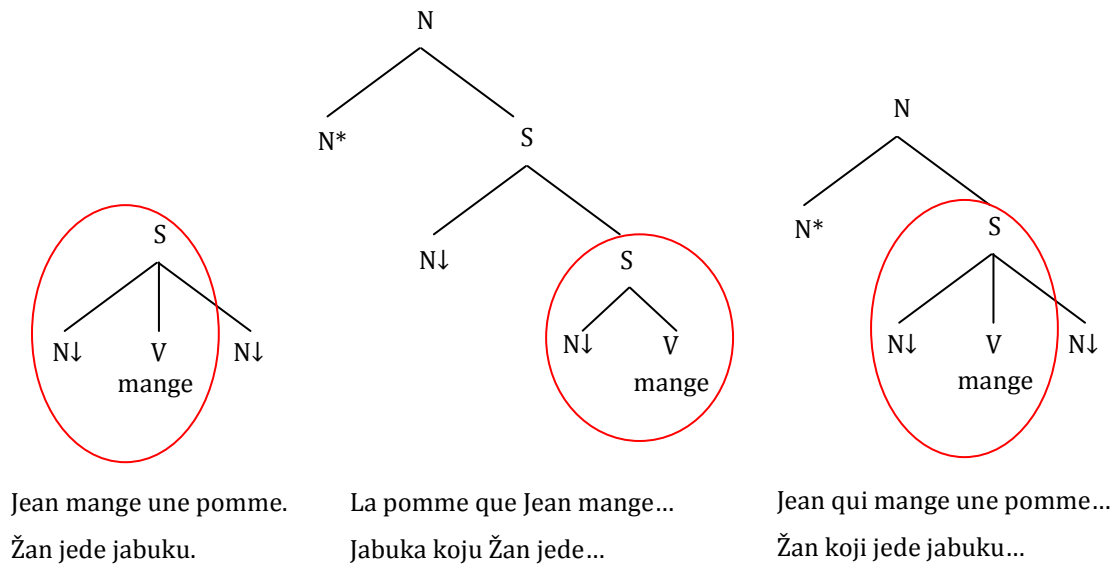
1. Generalizacija struktura

Već pomenutu redundantnost XMG rešava uočavanjem još jedne specifičnosti stabala, što je činjenica da se različita stabla sastoje od istih segmenata. Slika 27 tako prikazuje segment koji označava subjekat. Kako bi omogućio uočavanje obrazaca i regularnosti u strukturama, kao i definisanje daljih generalizacija, XMG operiše ovakvim fragmentima kao svojim osnovnim jedinicama.

⁸⁴ engl. *eXtensible MetaGrammar* – XMG

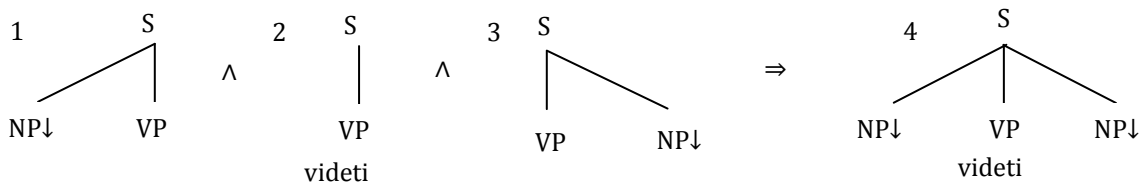
⁸⁵ XMG je dizajniran za Linux operativni sistem, ali se na Windows operativnim sistemima može koristiti unutar virtuelne mašine. Uputstvo za instalaciju u oba slučaja nalazi se na adresi:
http://dokufarm.phil.hhu.de/xmg/?animal=xmg#getting_started

Jezgro metagramatike XMG čine osnovni rečenični fragmenti (kao što su subjekat, predikat, pravi ili nepravi objekat) koji su po svojim karakteristikama organizovani u grupe koje se nazivaju *klase*. Kombinovanjem klasa pomoću dve operacije – konjunkcije⁸⁶ i disjunkcije⁸⁷, dobijaju se sva stabla TAG gramatike.



Slika 27 Uočavanje istog segmenta u različitim stablima (slika je uz neznatne izmene preuzeta iz (Le Roux, 2005))

Slika 28 prikazuje povezivanje tri segmenta stabla organizovana u klase – *Subjekt* (fragment 1), *Aktivni predikat* (fragment 2) i *Objekat* (fragment 3) – operacijom konjunkcije, čiji je rezultat klasa *Aktivna prelazna konstrukcija* (stablo 4)⁸⁸.



Slika 28 Kombinovanje klasa operacijom konjunkcije

⁸⁶ Logička operacija koja je ekvivalent veznika *i* u prirodnom jeziku. Obeležava se simbolom \wedge .

⁸⁷ Logička operacija koja je ekvivalent veznika *ili* u prirodnom jeziku. Obeležava se simbolom \vee .

⁸⁸ Nazivi klasa koji se koriste ovom prilikom su ilustrativni i upotrebljeni u svrhe olakšanog prikaza koncepta.

2. Povezivanje alternativa

Drugi nivo generalizacije podrazumeva povezivanje različitih realizacija iste argumentske strukture. Ovakve alternative, kakve su, na primer, aktivna i pasivna konstrukcija, se u TAG povezuju leksičkim pravilima, dok se u XMG za ove potrebe koristi samo disjunkcija, pomoću koje se alternativne strukture udružuju u natklase.

Ranije definisanu klasu *Aktivna prelazna konstrukcija* (Slika 28) možemo prikazati i na sledeći način:

$$\textit{Aktivna prelazna konstrukcija} \rightarrow \textit{Subjekat} \wedge \textit{Aktivni predikat} \wedge \textit{Objekat}$$

Da bismo definisali alternaciju aktiv-pasiv, potrebno je da definišemo još jednu klasu, *Pasivna prelazna konstrukcija*. Ova klasa će izgledati ovako:

$$\textit{Pasivna prelazna konstrukcija} \rightarrow \textit{Subjekat} \wedge \textit{Pasivni predikat} \wedge \textit{Agentivna odredba}$$

Alternacija zatim može biti definisana u okviru natklase pod imenom *Prelazna konstrukcija*, kao disjunkcija prethodno definisanih klasa:

$$\textit{Prelazna konstrukcija} \rightarrow \textit{Aktivna prelazna konstrukcija} \vee \textit{Pasivna prelazna konstrukcija}$$

2.5.1.2 Definicija metagramatike

XMG se formalno definiše sledećim formulama:

A *Klasa* ::= *Ime* \rightarrow *Sadržaj*

B *Sadržaj* ::= *Ime* | *Sadržaj* \vee *Sadržaj* | *Sadržaj* \wedge *Sadržaj* | *Opis*

C *Opis* ::= $n_1 \rightarrow n_2$ | $n_1 \rightarrow^+ n_2$ | $n_1 \rightarrow^* n_2$ | $n_1 < n_2$ | $n_1 <^+ n_2$ | $n_1 <^* n_2$ | $n[f:E]$ | $n(p:E)$

D *Upit* ::= *Ime*

Deo A opisuje klasu kroz ime i sadržaj klase. U primeru *Aktivna prelazna konstrukcija* \rightarrow *Subjekat* \wedge *Aktivni predikat* \wedge *Objekat* – deo s leve strane strelice je ime klase (*Aktivna prelazna konstrukcija*), a deo s desne, sadržaj klase (*Subjekat* \wedge *Aktivni predikat* \wedge *Objekat*).

Deo B predstavlja sadržaj klase. On može biti:

- Ime: Ime potklase.

Subjekat → **Imenički subjekat**

Subjekat → **Zamenički subjekat**

- Sadržaj ∨ Sadržaj: Disjunkcija ili izbor sadržaja.

Subjekat → **Imenički subjekat** ∨ **Zamenički subjekat**

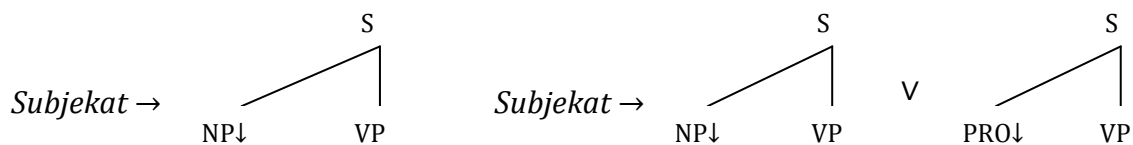
Prelazan → **Prelazan sa subjektom** ∨ **Prelazan bez subjekta**

- Sadržaj ∧ Sadržaj: Konjunkcija ili spoj sadržaja.

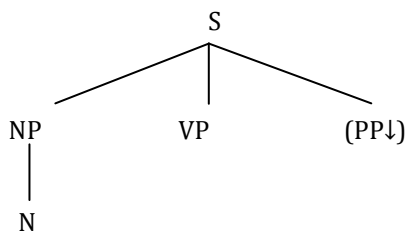
Prelazan sa subjektom → **Subjekat** ∧ **Predikat** ∧ **Pravi objekat**

Prelazan bez subjekta → **Predikat** ∧ **Pravi objekat**

- Opis: Konkretno TAG stablo (ili stabla) na koje referiše ime date klase. U primeru je to stablo imeničkog subjekta, odnosno disjunkcija stabala imeničkog i zameničkog subjekta.



Deo C definiše odnos između dva bilo koja čvora u stablu. Uzmimo za primer sledeće stablo:



- → označava odnos direktne nadređenosti (S i NP ili S i VP); →+: indirektna nadređenosti (S i N); →*: nadređenosti ili jednakosti (S i fakultativna PP);

- < označava neposredno linearno prethođenje (NP i VP); <+: posredno linearno prethođenje (NP i PP); <*: posredno linearno prethođenje ili jednakost (VP i PP);
- $n[f=E]$ se odnosi na obeležje f čvora n sa vrednošću E (na primer NP[gen=m])
- $n(p=E)$ označava svojstvo⁸⁹ p čvora n sa vrednošću E (na primer PP(mark=subst)).

Deo D predstavlja ime natklase od koje se polazi pri kompilaciji metagramatike. Ako je za kompilaciju označena klasa *Prelazna konstrukcija*, kompilator će kompilirati sve klase od kojih se ona sastoji, počevši od klasa *Aktivna prelazna konstrukcija* i *Pasivna prelazna konstrukcija*, do svih klasa za subjekat, predikat i objekat. Pozivanjem klasa određenog nivoa može se ograničiti broj klasa koji se poziva pri kompilaciji.

2.5.1.3 Mehanizam nasleđivanja

U XMG se za upućivanje na segmente klase, tj. konkretne čvorove, koriste promenljive. Promenljive su lokalne, tj. vezane su za klasu unutar koje su definisane, međutim, XMG podržava njihovu upotrebu i van date klase zahvaljujući mehanizmu izvoza⁹⁰. Za svaku klasu je moguće definisati sve promenljive za koje je u planu da se koriste nelokalno. Mehanizam uvoza⁹¹ podrazumeva ili uvoženje svih izvezenih promenljivih jedne klase, ili uvoženje pojedinih promenljivih na koje se može referisati preko imena klase i naziva promenljive. Tako ukoliko klasa pod imenom *Subjekat* sadrži čvorove *V* i *NP*, i ukoliko su u toj klasi date promenljive definisane za izvoz (formulom *export V NP*), na te čvorove se u drugim klasama može referisati kao *Subjekat.V* i *Subjekat.NP*. Druga mogućnost je da se pozove čitava klasa *Subjekat* (formulom *import Subjekat*) kada će u klasi u kojoj je klasa *Subjekat* pozvana moći da se koristite sve promenljive te klase koje su definisane za izvoz (u ovom slučaju *V* i *NP*).

Mehanizam nasleđivanja omogućava definisanje apstraktnih klasa (npr. *Argument glagola*) koje definišu opšte nadstrukture i koje se zatim koriste kao model za definisanje odgovarajućih potklasa (*Subjekat*, *Pravi objekat*, *Nepravi Objekat*). Isto tako, omogućava

⁸⁹ XMG definiše dva tipa svojstava za čvorove: 1) **tip čvora** (subst – čvor za zamenu, foot – priključni čvor, flex – terminalni čvor tj. leksema, nadj – čvor na kome pripajanje nije dozvoljeno, anchor – sidro, coanchor – dodatno sidro) i 2) **boja čvora** (crvena, bela i crna). Za sistem upotrebe boja pogledati poglavlje 2.5.1.4.

⁹⁰ engl. *export*

⁹¹ engl. *import*

upotrebu istih naziva čvorova u klasama, među kojima se razlika pravi na osnovu imena klase u okviru koje se nalaze (*Subjekt.NP* i *PraviObjekat.NP*).

2.5.1.4 Dodavanje specifičnih ograničenja

1. Boje

Da bi se olakšalo pisanje gramatike i proces kombinovanja fragmenata učinio poluautomatskim, uveden je princip ograničavanja na kombinovanje čvorova putem boja (Crabbé & Duchier, 2005). Kombinovanje ili identifikacija je vrsta unifikacije koja prateći matricu (Tabela 4) određuje koji fragmenti se mogu spajati sa kojim drugim fragmentima⁹². U upotrebi su tri boje – crvena, bela i crna.

Beli čvorovi obavezno podležu identifikaciji – mogu se identifikovati i sa drugim belim čvorovima, ali na kraju se uvek moraju identifikovati sa nekim crnim čvorom. Crni čvorovi se mogu identifikovati sa belim čvorovima, ali ne moraju, odnosno mogu ostati crni. Crveni čvorovi se nikad ne identifikuju ni sa jednom vrstom čvorova.

Jedno stablo opisano na ovaj način je prihvatljivo samo ako su svi njegovi čvorovi posle identifikacije obeleženi crnom ili crvenom bojom⁹³.

Tabela 4 Logika sistema boja u XMG

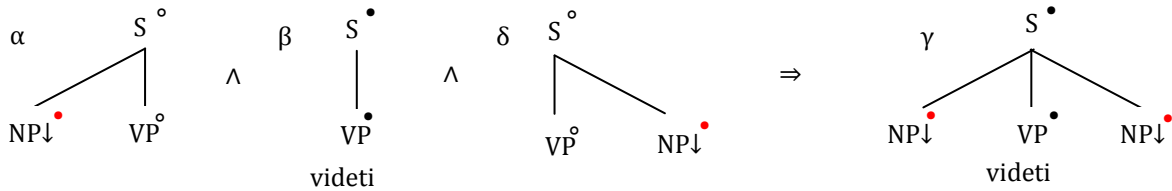
	•B	•R	◦W
•B	⊥	⊥	•B
•R	⊥	⊥	⊥
◦W	•B	⊥	◦W

Slika 29 daje primer koji smo videli ranije, ali ovoga puta obeležen bojama. Bela boja na čvorovima S i V stabala za subjekat i objekat pokazuje da se oni moraju identifikovati sa

⁹² Ovo je istovremeno i zamena za unifikaciju koja se unutar ove metagramatike ne vrši.

⁹³ Primena ovog sistema na konkretnim klasama za srpski može se videti u poglavlju 4.

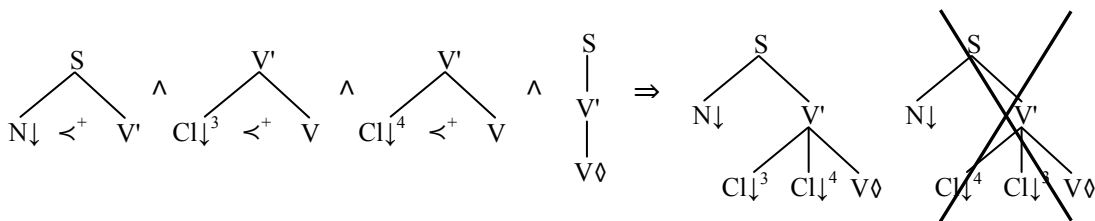
odgovarajućim čvorovima obeleženim crnom bojom, u ovom slučaju čvorovima S i V glagola. Čvor N je na svim stablima označen crvenom bojom, kako ne treba, i ne sme da se identifikuje ni sa jednim od čvorova u datim stablima.



Slika 29 Primer stabala obeleženih bojama i njihova identifikacija

2. Redosled enklitika

XMG sadrži zaseban modul, razvijen za francuski jezik (Crabbé, 2005), koji omogućava automatsko raspoređivanje enklitika u pravilnom redosledu. On se zasniva na dva principa: *principu jedinstvenosti čvora* i *principu redosleda čvora*. Svakom čvoru koji nosi oznaku *enklitika* (cat=cl) dodeljuje se obeležje ranga sa celobrojnom vrednošću n koje će imati onoliko vrednosti koliko ima pozicija za enklitike u datom jeziku. Princip jedinstvenosti čvora obezbeđuje da u jednom stablu ne postoje dva čvora sa istom vrednošću svojstva rang. Princip redosleda čvorova dalje zahteva da u finalnom stablu čvorovi budu linearno raspoređeni prema vrednosti njihovog ranga (Slika 30).



Slika 30 Raspoređivanje enklitika u francuskom (prema (Crabbé et al., 2013))

Prema tradicionalnom gramatičkom opisu, srpski ima pet pozicija za enklitike, koje se, u slučajevima kada se javljaju istovremeno, javljaju u sledećem redosledu: 1) enklitika

*li*⁹⁴, 2) enklitički oblici pomoćnih glagola (sem oblika *je*), 3) enklitički oblici ličnih zamenica u dativu, 4) enklitički oblici ličnih zamenica u akuzativu ili genitivu i 5) rečca *se* ili pomoćni glagol *je*. Tako će maksimalna vrednost *n* za srpski biti 5. Svako od enklitika se dodeljuje odgovarajući rang u skladu sa njenom pozicijom (npr. enklitika dativ [rank=3], enklitika akuzativ [rank=4]), tako da će se pri njihovom istovremenom javljanju u rečenici, rasporediti u odgovarajućem redosledu.

2.5.1.5 Parsiranje TAG pomoću metagramatike

Postoji nekoliko parsera koji su razvijeni za TAG – parser za FBLTAG koji je deo gramatike XTAG za engleski (Doran, Egedi, Hockey, Srinivas, & Zaidel, 1994), parser za LTAG razvijen za francuski pod imenom LLP2 (Lopez, 2000), parser za FBLTAG razvijen za francuski pod imenom DyALog (de La Clergerie, 2005) i parser za FBLTAG i MCTAG razvijen za nemački i francuski pod imenom TuLiPA (Kallmeyer, Lichte, Maier, Parmentier, Dellert, et al., 2008). Za potrebe ovog rada biće korišćen parser TuLiPA⁹⁵.

TuLiPA⁹⁶ je okruženje za parsiranje napravljeno sa ciljem da omogući:

- parsiranje većeg broja formalizama, i to ne nužno gramatika stabala
- grafički interfejs prilagođen potrebama korisnika
- integrisanu obradu semantičkih informacija.

Multiformalizam, i ujedno modularnost ovog okruženja, omogućen je time što je gramatika konkatenacije opsega ili RCG⁹⁷ uzeta za njegov centralni formalizam. Sve gramatike koje su manjeg slabog generativnog kapaciteta i koje se mogu konvertovati u RCG mogu potencijalno biti analizirane ovim parserom, a sve što je potrebno je da se u okruženje

⁹⁴ U ovoj fazi enklitiku *li* ipak obrađujemo samo u okviru upitne reči *da li*, koju kao celinu tretiramo kao rečcu. Ovo rešenje je, iako praktično, kruto za buduću primenu, kada će biti potrebe da se rečca *li* izdvoji kao kategorija nezavisna od ove upitne reči.

⁹⁵ Od engleskog *Tübingen Linguistic Parsing Architecture*

⁹⁶ Uputstvo za instalaciju za Windows/Linux/Mac može se naći na adresi: <https://sourcesup.renater.fr/tulipa/overview.html>

⁹⁷ engl. *Range Concatenation Grammar* – gramatika sa većim slabim generativnim kapacitetom od blago kontekstno osetljivih formalizama, koja ih sadrži u sebi kao svoj podskup.

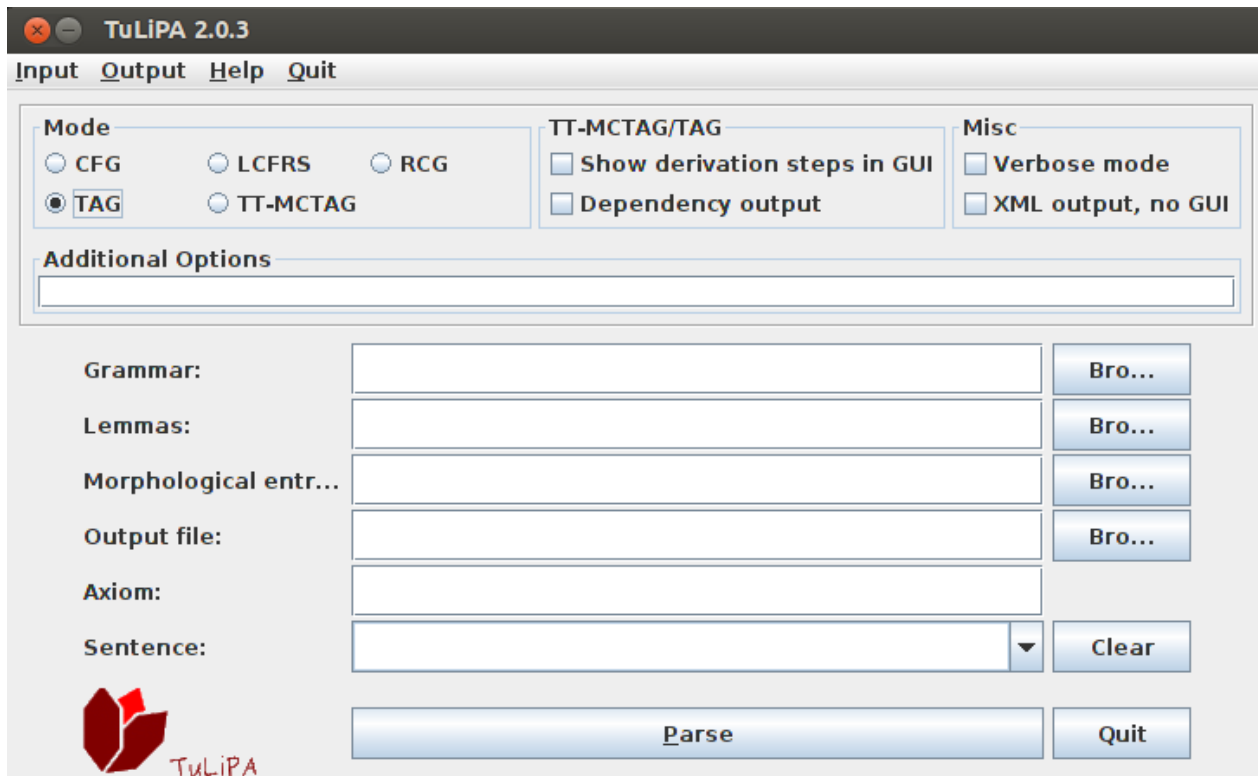
doda modul koji vrši konverziju iz njih i u njih. Proces parsiranja u parseru TuLiPA odvija se tako u tri faze:

1. Ulazna metagramatika se konvertuje u RCG,
2. RCG očitava nisku unetu u parser i analizira je,
3. Rezultat parsiranja se prevodi iz RCG u ciljni formalizam.

Da bi mogla da obavi parsiranje, TuLiPA mora da dobije četiri vrste informacija:

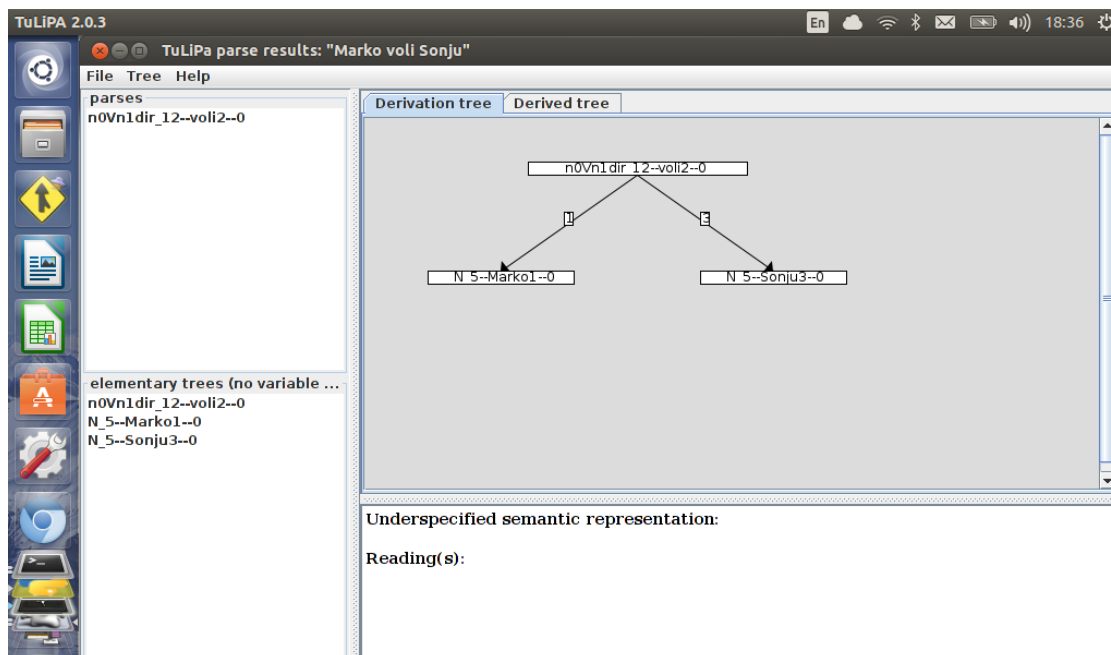
1. Metagramatiku u .xml formatu koja je ručno formirana putem formalizma XMG. Metagramatika obezbeđuje opis shema stabala.
2. Leksikone napravljene po ugledu na dvoslojni leksikon XTAG (2001), a koji se sastoje od:
 - morfološkog dela, koji sadrži forme reči, njihove leme i morfološka obeležja
 - leksičkog dela, u kome se definišu lema, sintaksička kategorija leme, semantičke informacije, naziv porodice stabala čije je ova lema sidro, filteri koji sadrže obeležja koja definišu ograničenja na primenu stabala iz date porodice, izjednačenja koja definišu imena za čvorove sa određenim karakteristikama, i potencijalna dodatna sidra za složene reči i idiome.
3. Takozvani aksiom, kojim se definiše tip niske koji će biti unet u parser. Dostupni aksiomi su oni koji su definisani kao neterminalne kategorije u metagramatici, pa tako unošenje S kao aksioma znači da će niske zadate za parsiranje biti rečenice, dok NP znači da će biti unete imeničke fraze i sl.

Slika 31 prikazuje grafički interfejs parsera TuLiPA.

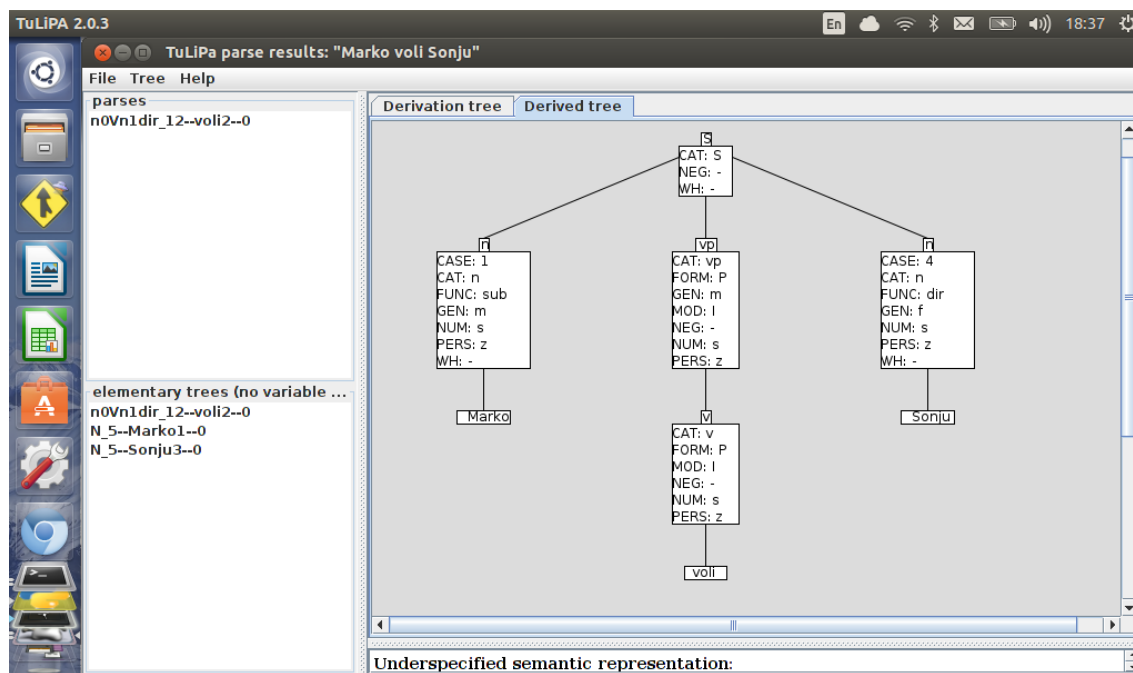


Slika 31 Grafički interfejs parsera TuLiPA

Rezultat rada parsera su leksikalizovana stabla izvođenja (Slika 32) i izvedena stabla (Slika 33) za unetu nisku. Niska koja je parsirana za ovu priliku je niska *Marko voli Sonju*. Ukoliko je definisana semantička dimenzija za datu gramatiku, dobija se i semantička reprezentacija niske.



Slika 32 Izgled stabla izvođenja za srpski jezik u parseru TuLiPA



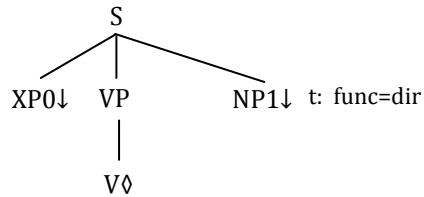
Slika 33 Izgled izvedenog stabla za srpski jezika u parseru TuLiPA

3 IZRADA OSNOVA FORMALNE GRAMATIKE SRPSKOG JEZIKA

Na ovom mestu ćemo predstaviti osnove formalne gramatike koju smo izradili za srpski jezik, a koju smo nazvali SrpTAG. Za izradu ovog segmenta koristili smo formalizam FBLTAG, predstavljen u poglavlju 2.4.5.1. Pored karakteristika same gramatike, koja omogućava olakšano prikazivanje zavisnosti među rečeničnim delovima, ovu gramatiku smo odabrali i zbog primene strukture obeležja, koja omogućava njeno povezivanje sa već postojećim elektronskim rečnicima. Kao uzor za FBLTAG gramatiku koristili smo mahom formalnu gramatiku francuskog jezika, detaljno opisanu u (Abeillé, 2002). Data gramatika se u velikoj meri bazira na rečeničnim modelima opisanim u leksikon-gramatici Morisa Grosa (Gross, 1975), a na čijim principima je izrađen i sam LADL format elektronskih rečnika. Na taj način gramatika koju izrađujemo i rečnici koji su već razvijeni za srpski jezik čine jednu organsku celinu.

Za osnovni, kanonski red reči u srpskom jeziku uzimamo red SVO ili subjekat predikat objekat. Subjekt, objekat i priloške dopune smatramo argumentima glagola ili rečeničnim konstituentima koji su određeni valentnošću glagola⁹⁸. U tom smislu, argumente glagola u stablima, i kasnije u modelima, obeležavaćemo brojčano, prema njihovom linearnom redosledu u rečenici s osnovnim redom reči (gledano s leva). Slika 34 daje ilustrativan prikaz stabla koje je označeno na ovaj način. Broj 0 označava subjekat, a ostali brojevi redom sledeće argumente koji se pojave u strukturi. Argumenti koji mogu biti označeni istim brojem u različitim strukturama (npr. pravi i nepravi objekat), međusobno se razlikuju po obeležju funkcije. Oznaku X ili XP u stablima (ista slika) korišćemo da označimo da se dati čvor može realizovati kao više različitih kategorija, najčešće kao imenička fraza ili zamenica. Na taj način sažeto predstavljamo dva stabla koja se razlikuju samo po tom čvoru, ali ovo sažimanje koristimo samo u tekstu – u gramatici svejedno moraju postojati dva, odnosno odgovarajući broj stabala s različitim kategorijama.

⁹⁸ Valentnost glagola ili valenca, pojam koji se vezuje za Lisjena Tenijera (Lucien Tesnière) i dependencijalnu gramatiku (Tesnière, 1959), označava opštu sposobnost glagola da za sebe veže određen broj argumenata, odnosno dopuna. Prema valentnosti, glagoli u srpskom mogu biti minimalno nulte valentnosti (*sevati*), a maksimalno trovalentni (*davati*).



Slika 34 Primer stabla za prelazni glagol

U narednim poglavljima ćemo predstaviti zasebno svaki od delova značajnih za SrpTAG: morfosintaksičke odluke vezane za oznake u gramatici (poglavlje 3.1), izbor deskriptivne gramatike srpskog jezika (poglavlje 3.2), odnos između elementarnog stabla SrpTAG i rečenice srpskog jezika (poglavlje 3.3), odnos i sastav pojma predikata u SrpTAG i izabranoj deskriptivnoj gramatici (poglavlje 3.4), prikaz izgleda glagolskih argumenata u SrpTAG (poglavlje 3.5) i prikaz porodica stabala SrpTAG i njihovog odnosa prema rečeničnim modelima deskriptivne gramatike (poglavlje 3.6).

3.1 Morfosintaksički opis

Kao polazište za morfosintaksički opis (MSO) gramatike SrpTAG, a samim tim i njene metagramatike, uzeli smo MSO koji se koristi u elektronskim rečnicima u LADL formatu opisanim u (Krstev, 2008), a o kojima je bilo reči u poglavlju 1.2.2. Kako je ovaj rečnik već prilagođen za potrebe automatske obrade teksta, format njegovih obeležja bio je sasvim prikladan za MSO naše gramatike. Ovoj grupi obeležja dodali smo obeležja koja su proizašla iz potreba same formalne gramatike.

Tabela 5 daje pregled kategorija koje smo koristili u formalnoj gramatici. Prvih deset terminalnih kategorija u tabeli preuzeto je, zajedno sa oznakama, iz LADL rečnika. Kategoriju CONJ (veznik) iz ovih rečnika, a koja se u njima koristi za označavanje naporednih (koordinativnih) i subordinativnih veznika, za potrebe formalne gramatike koristimo samo za obeležavanje naporednih veznika⁹⁹. Za subordinativne veznike koristimo novounetu kategoriju C. Kategoriju PAR trenutno koristimo samo za označavanje upitne konstrukcije *da li*, dok kategoriju Cl uvodimo za označavanje enklitičkih oblika zamenica,

⁹⁹ Naporedni odnosi ipak ostaju van okvira ovog rada te se u daljem tekstu nećemo njima baviti.

kao i za rečcu *se*¹⁰⁰. Za potrebe formalne gramatike, rečcu *ne* ne svrstavamo u kategoriju PAR, već u zasebnu kategoriju pod imenom NEG.

U ovoj fazi izrade formalne gramatike ne koristimo nijednu semantičku oznaku koja je u upotrebi u elektronskim rečnicima¹⁰¹.

Što se frazalnih kategorija tiče, definisane su svuda gde terminalna kategorija dozvoljava neku vrstu modifikacije, odnosno može da formira frazu ili sintagmu. Na dnu tabele nalazi se oznaka za samu rečenicu. Iako je oznaka za oba tipa rečenica (zavisne i nezavisne) ista, u stablima se zavisna rečenica razlikuje od nezavisne po tome što nosi oznaku broja glagolskog argumenta uz sebe (npr. S0 – subjekat), dok je nezavisna rečenica uvek samo – S.

Tabela 5 Terminalne i frazalne kategorije u formalnoj gramatici srpskog jezika

	TERMINALNE KATEGORIJE		FRAZALNE KATEGORIJE	
1.	N	imenica	NP	imenička fraza
2.	PRO	zamenica	/	/
3.	V	glagol	VP	glagolska fraza (predikat)
4.	A	pridev	AP	pridevska fraza
5.	ADV	prilozi	ADVP	priloška fraza
6.	PREP	predlozi	PP	predložka fraza
7.	NUM	brojna reč	NUMP	brojna fraza
8.	PAR	rečca	/	/
9.	CONJ	naporedni veznik	/	/
10.	INT	uzvik		
11.	C	subordinativni veznik	/	/
12.	Cl	enklitika	/	/
13.	NEG	negacija	/	/
14.	/	/	S	nezavisna rečenica zavisna rečenica

¹⁰⁰ Zameničke enklitike, zajedno sa rečcom *se*, izdvojene su u zasebnu kategoriju između ostalog i da bi u jednom trenutku bila moguća primena mehanizma za raspored enklitika u metagramatici. Potpuna primena tog mehanizma bi ipak podrazumevala da i pomoćni glagoli nose ovu oznaku, ali i to da se rečca *se* i pomenuti pomoćni glagoli ne nalaze unutar glagolske fraze (VP), već na istom nivou kao i zameničke enklitike, odnosno direktno ispod čvora S.

¹⁰¹ Oznake za tip imenice (apstraktna, konkretna, gradivna, koja označava osobu itd.), vrstu priloškog značenja, vrstu vlastitih imena (imena ljudi, toponima...), oblasti iz koje je data reč (matematika, vojska, bibliotekarstvo...), tip priloškog značenja (spacijalno, direktivno, ablativno...), jezičku varijantu (ekavski, ijekavski) itd.

Tabela 6 daje pregled obeležja koja smo koristili u gramatici. Kako elektronski rečnik najšešće ne sadrži nazive za ono što u tabeli zovemo atributima¹⁰², nazive smo definisali prema (Abeillé, 2002) ili intuitivno u skladu sa nazivom samog obeležja.

Vrednosti atributa i njihove oznake mahom smo preuzeli iz LADL rečnika. Vrednosti “+” i “-“ u tabeli označavaju posedovanje ili neposedovanje vrednosti datog atributa¹⁰³. U koloni *nosilac* nalaze se kategorije za koje je dato obeležje relevantno. Ukoliko obeležje mogu nositi i terminalna i frazalna kategorija, u tabelu smo uneli samo terminalnu¹⁰⁴.

Tabela 6 Obeležja i njihove vrednosti u formalnoj gramatici srpskog jezika¹⁰⁵

	ATRIBUT	ZNAČENJE	VREDNOST	NOSILAC
1.	gen	rod	m – muški rod f – ženski rod n – srednji rod MG – prirodni muški rod ¹⁰⁶ FG – prirodni ženski rod ¹⁰⁷ NG – prirodni srednji rod ¹⁰⁸	N, A, PRO, V, Cl, S
2.	num	broj	s – jednina p – množina PL – semantička množina ¹⁰⁹	N, A, PRO, V, Cl, S
3.	case	padež	1 – nominativ 2 – genitiv 3 – dativ 4 – akuzativ 5 – vokativ 6 – instrumental	N, A, PRO, Cl, PREP ¹¹⁰

¹⁰² Uz retke izuzetke kao što su *wh* za upitne reči, *aux* za pomoćne glagole i *ref* kao oznaku za refleksivnost. U unoscima u rečniku se direktno beleže vrednosti atributa, uglavnom prikazane kao jedan karakter (slovo ili broj).

¹⁰³ “+” i “-“ se još nazivaju *bulovski atributi*, a date vrednosti *bulovske vrednosti*.

¹⁰⁴ Na primer, obeležje roda koristi se kao obeležje i terminalne kategorije N i njene frazalne kategorije NP, za razliku od obeležja funkcije, koje u ovom slučaju nosi samo NP.

¹⁰⁵ Format tabele preuzet iz (Abeillé, 2002)

¹⁰⁶ Za imenice koje su gramatičkog ženskog a prirodnog muškog roda: *trojica, Nikola, tata*.

¹⁰⁷ Za imenice gramatičkog ženskog roda koje označavaju osobe oba prirodna pola: *osoba, budala, pristalica*.

¹⁰⁸ Za zbirne imenice gramatičkog ženskog roda: *deca, braća*

¹⁰⁹ Za zbirne imenice kao što su *braća* ili *deca* koje su gramatički u obliku jednine.

¹¹⁰ Kod predloga, obeležje padeža označava padež regirane imenice ili zamenice.

			7 – lokativ	
4.	def	određenost	d – određen k – neogređen	A
5.	pers	lice	x – prvo lice y – drugo lice z – treće lice	V, PRO, N ¹¹¹ , S
6.	neg	negacija	+, -	PRO, VP, S
7.	aux	pomoćni glagol	+, -	V
8.	wh	upitni oblik	+, -	S, C, PRO, N
9.	ref	refleksivni oblik	+, -	V, CI
10.	ord	redni broj	+, -	A
11.	asp	vid glagola	imperf – nesvršeni vid perf – svršeni vid	V
12.	mod	modus glagola	W – infinitiv I – indikativ Y – imperativ G – radni glagolski pridev T – trpni glagolski pridev S – glagolski prilog sadašnji X – glagolski prilog prošli	V, S
13.	form	prosta i složena glagolska vremena i složeni načini u indikativu	P – prezent F – futur A – aorist I – imperfekat R – prošlo vreme C – potencijal L - pluskvamperfekat D – futur II	V
14.	cop	kopulativni glagol	+, -	V
15.	func	funkcija	sub – subjekat lsub – logički subjekat dir – pravi objekat indir – nepravi objekat	S, NP, AP, PRO, PP, ADVP, NUMP, CI

¹¹¹ Obeležje lica koristimo i kod imenica za potrebe kongruencije sa glagolom, iako ima podrazumevanu vrednost trećeg lica.

			pdop – priloška dopuna	
16.	subg	slaganje sa subjektom u rodu	m – muški rod f – ženski rod n – srednji rod MG – prirodni muški rod FG – prirodni ženski rod NG – prirodni srednji rod	S, V
17.	subp	slaganje sa subjektom u licu	x – prvo lice y – drugo lice z – treće lice	S, V
18.	subn	slaganje sa subjektom u broju	s – jednina p – množina PL – semantička množina ¹¹²	S, V
19.	modal	modalni glagol	+, -	V
20.	fazni	fazni glagol	+, -	V

Obeležja od broja 12 do 20 su ona koja smo uneli za potrebe formalne gramatike. Obeležje **mod** uneli smo da označimo razliku između glagola u ličnom glagolskom obliku (označenom kao I – indikativ) od načina (imperativ) ili glagola u neličnim glagolskim oblicima. Za glagole koji nose vrednost “indikativ” dalje razlikujemo vremena i složene glagolske oblike (vremena i načine), koje označavamo kroz obeležje **form**.

Oznaku **cop** koristimo da označimo kopulativne glagole¹¹³. U stablima koristimo i funkcije, koje unosimo kroz obeležje **func**. Izbor funkcija napravljen je prema gramatici (Stanojčić & Popović, 1997), s tim što za potrebe ovog rada koristimo isključivo funkcije koje se navode unutar rečeničnih modela u datoj gramatici, kao i one koje unutar formalne gramatike imaju konstituentku tj. argumentsku vrednost¹¹⁴. To istovremeno znači da ne obeležavamo funkcije glagolskih odredbi.

¹¹² Za zbirne imenice kao što su *braća* ili *deca* koje su gramatički u obliku jednine.

¹¹³ Za to koji se sve glagoli smatraju kopulativnima u SrpTAG videti 3.4.2.

¹¹⁴ Na taj način su isključene dve dopune koje se pominju u gramatici (Stanojčić & Popović, 1997) – aktuelni kvalifikativ, kako nisu navedeni rečenični modeli u kojima se pojavljuje, i predikativ, kako u formalnoj gramatici nema konstituentku vrednost. Za detalje videti 3.5.2.3.

Obeležja **subg**, **subp** i **subn** koristimo da naznačimo da subjekat zavisne rečenice deli kategorije broja, lica i roda sa subjektom glavne rečenice.¹¹⁵

Za označavanje modalnih i faznih glagola koristimo oznake **modal** i **fazni** s bulovskom vrednošću.

Tabela 7 Obeležja koja ograničavaju primenu leksičkih pravila

NAZIV	ZNAČENJE	VREDNOST	NOSILAC
pasiv	pasivizacija je moguća (participski pasiv)	+,-	V
bez1	izostavljanje prve dopune	+,-	V
bez2	izostavljanje druge dopune	+,-	V

Tabela 7 prikazuje obeležja koja se definišu unutar leksikona i odnose se na čitavu rečeničnu porodicu. Ova obeležja ograničavaju primenu odgovarajućih leksičkih pravila – pasivizacije i izostavljanja argumenata. Obeležje **pasiv** s negativnom vrednošću koristi se one glagole koji ne dozvoljavaju pasivizaciju (npr. *znati*) i obrnuto, s pozitivnom vrednošću uz glagole koji je dozvoljavaju (npr. *voleti*). Oznake **bez** se koriste da se označi da dopune datog glagola mogu ili ne mogu biti izostavljene¹¹⁶.

Obeležja unutar formalne gramatike i metagramatike koristimo u formi *atribut=vrednost*, kao na primer *case=7* ili *pasiv=+*.

Uz stabla koja koristimo kao ilustracije u radu stajaće minimum obeležja potreban da se prikaže pojava o kojoj je reč. Isto tako, nećemo beležiti gornja i donja obeležja uz čvorove sem ako se ona međusobno ne razlikuju. Tako, kada nadalje u stablima u tekstu uz čvor postoji samo jedna grupa obeležja, to znači da gornja i donja obeležja tog čvora imaju iste vrednosti, odnosno da su potpuno kompatibilna. Razlog za to je prosto nedostatak prostora.

¹¹⁵ Kao u primeru *Marko želi da dođe.*, u kome je subjekat zavisne rečenice nerealizovan. Za detalje o tretmanu nerealizovanog subjekta u SrpTAG videti 3.5.2.1.

¹¹⁶ Na primer, kod glagola sa dve dopune – u akuzativu i dativu, obično je moguće izostavljanje dopune u dativu, ali ne i izostavljanje dopune u akuzativu: *Marko_o je poklonio (Ivani)₁ *(mačku)₂*. Data porodica glagola će tako imati sledeća obeležja: *bez1=+*, *bez2=-*.

U stablima gramatike koja se koriste za parsiranje se, ipak, uz svaki čvor stabla nalazi kompletan skup obeležja koji odgovara kategoriji datog čvora.

3.2 O izboru deskriptivne gramatike srpskog jezika

Gramatike namenjene ljudskoj upotrebi generalno se smatraju nezadovoljavajućim za primenu u formalne svrhe usled nedovoljne eksplicitnosti, nedovoljne pokrivenosti jezičkih pojava, teškoća pri implementaciji u formalne svrhe, a katkad i netačnosti gramatičkog opisa (Erbach & Uszkoreit, 1990). Ipak, budući da je formalna gramatika koju pravimo prva formalna gramatika srpskog jezika i da nemamo na raspolaganju drugu formalizovanu predstavu srpskog jezika ovog tipa, verujemo da je neophodno da se u ovom poslu oslonimo na neki tip tradicionalne gramatike. Pod ovime ne mislimo da ćemo vršiti bilo kakav direktan automatski tretman ovakve gramatike, već na to da nam ovakva gramatika služi kao vodič i uvid u strukturu rečenice srpskog jezika, kao i vrsta podsetnika i svojevrsna lista zadataka pri građenju SrpTAG.

Pri izboru takve gramatike, imali smo u vidu sledeće parametre:

- 1) Gramatika treba da bude gramatika savremenog srpskog jezika;
- 2) Treba da pruža pregled kompletnog sintaksičkog sistema srpskog jezika;
- 3) Treba da daje iscrpan i detaljan pregled rečeničnih modela, oblika predikata i njegovih argumenata;
- 4) Treba da bude organizovana pregledno.

Nismo mogli da očekujemo da metod koji koristi bilo koja od deskriptivnih gramatika srpskog jezika bude sličan metodu koji se primenjuje u TAG, ali svaka bliskost sa ovim metodom bila je, naravno, poželjna.

Nekoliko nama dostupnih gramatika je zadovoljilo većinu kriterijuma – *Sintaksa savremenog srpskog jezika* (Piper et al., 2005), *Gramatika srpskog jezika – udžbenik za I, II, III i IV razred srednje škole* (Stanojčić & Popović, 1997) i *Gramatika srpskog jezika za strance* (Mrazović, 2009). Svaka od ovih gramatika sasvim bi dobro poslužila u svrhe koje smo naveli. Ipak, smatrali smo da bi pri predstavljanju formalizma koji je manje poznat našoj

javnosti bilo najbolje za referentnu gramatiku uzeti najčešće korišćenu i najšire poznatu gramatiku srpskog jezika. Uzevši i taj kriterijum u obzir, naš izbor je bez dileme pao na gramatiku Živojina Stanojčića i Ljubomira Popovića. Ova gramatika ne samo da se koristi kroz čitavu srednju školu, već se koristi i kao glavni referentni udžbenik na Filološkom fakultetu, i u tom smislu u potpunosti ispunjava naš zahtev. Pored rasprostranjenosti njene upotrebe, postoji još nekoliko stvari koje ovu gramatiku čine pogodnom za naše potrebe:

- Pri opisu struktura bogato koristi grafički prikaz (u vidu dijagrama (Slika 35) ili tabelaran (Tabela 8)), koji pruža direktan i jasan uvid u izložene strukture;
- Popis i pregled rečeničnih modela je opsežan i temeljan, potkrepljen primerima i odgovarajućom grafičkom predstavom;
- Rečenični modeli su bazirani na funkcijama (umesto na kategorijama), što je od posebnog značaja pri izradi metagramatike za TAG, koja se i sama bazira na konceptu funkcije;
- Glagol nema po svaku cenu dominantnu ulogu u ovom modelu (kao npr. u kopulativnoj konstrukciji gde dominantnu ulogu ima predikativ, a ne kopula), što ovu gramatiku čini umereno verbocentričnom, kakva je i gramatika TAG.

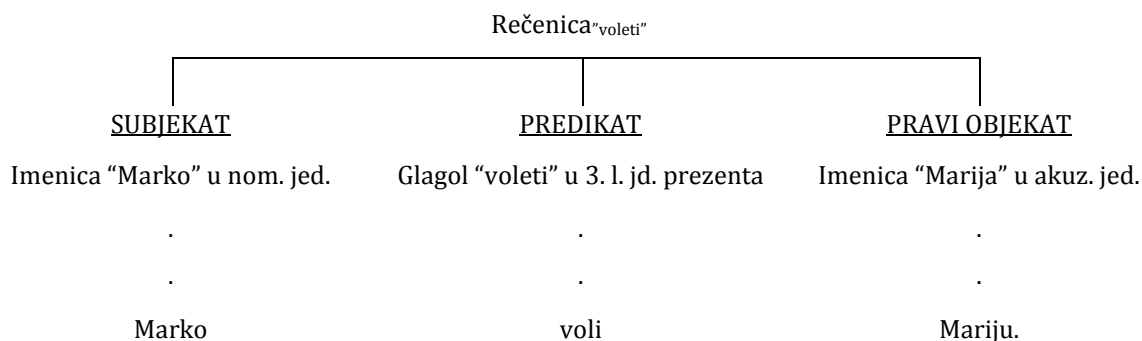


Tabela 8 Tabelarna analiza rečenice *Marko voli Mariju*.

Slika 35 Analiza rečenice *Marko voli Mariju* pomoću dijagrama

Konstituentna jedinica	Tip leksičke jedinice i njeno leksičko jezgro	Konstituentni oblik	Sintaksička funkcija i distribucija
Marko	Imenica "Marko"	nominativ jednine	SUBJEKAT u rečenici "voleti"
voli	Glagol "voleti"	3. l. jd. prezenta	PREDIKAT u rečenici "voleti"
Mariju	Imenica "Marija"	akuzativ jednine	PRAVI OBJEKAT u rečenici "voleti"

U sledećim poglavljima ćemo koristiti ovu gramatiku kao referentnu literaturu za srpski jezik pri definisanju struktura u SrpTAG. Formalna gramatika SrpTAG koju izrađujemo u ovom radu zamišljena je kao minimalna gramatika srpskog jezika. U tom smislu, u ovoj gramatici ćemo obraditi osnovne rečenične argumente (gramatički subjekat, logički subjekat, pravi objekat, nepravi objekat, priloška dopuna) u svom minimalnom obliku i kanonskom redosledu, kao i osnovne subjekatsko-predikatske i bezlične rečenične modele navedene u (Stanojčić & Popović, 1997).

Za sve jedinice i pojave koje opisujemo navodićemo definiciju njihove realizacije prema (Stanojčić & Popović, 1997), a zatim i njihov tretman u gramatici TAG, uključujući tretman koji smo mi odlučili da im damo.

3.3 Elementarno stablo i rečenica srpskog jezika

Pogledajmo sada u kakvom odnosu stoje pojmovi elementarnog stabla i rečenice srpskog jezika.

Najpre, prema definiciji koju smo mogli videti u poglavlju 2.4.2, pojam elementarnog stabla u TAG odgovara pojmu rečenice prirodnog jezika, ali isto tako i pojmu fraze ili sintagme, kao i na još nižem nivou – pojmu reči. U ovom poglavlju ćemo pojam elementarnog stabla svesti na onaj u kome označava rečenicu koja sadrži glagol¹¹⁷.

U podeli elementarnih stabala koju smo naveli u poglavlju 2.4.2 mogli smo videti da se inicijalno elementarno stablo koristi za predstavljanje minimalnih nerekurzivnih struktura, dok se pomoćno elementarno stablo koristi za definisanje minimalnih rekurzivnih struktura. Pored ovog osnovnog određenja i strukturnih obeležja koja ova stabla nose, za elementarna stabla su definisani takozvani principi dobre formiranosti stabala (Kroch & Joshi, 1985), koji oblikuju sve odluke vezane za njihov sastav:

1. **Princip leksičkog usidrenja.** Prema ovom principu, svako elementarno stablo TAG (LTAG) gramatike mora imati makar jedno leksičko sidro, odnosno mora biti direktno povezano sa leksikom. Takođe, ovo sidro mora biti fonetski realizovano i, u slučaju stabala koja predstavljaju rečenice – punoznačno. U skladu sa ovim principom,

¹¹⁷ Glagol umesto predikat, što je osnovno određenje predikatske rečenice srpskog jezika prema (Stanojčić & Popović, 1997), ovde koristimo s namerom. Videti poglavlje 3.4 koje govori o ovom odnosu.

nerealizovani argumenti nikada ne mogu biti sidra elementarnih stabala, kao što to ne mogu biti ni funkcionalne reči poput predloga, iako mogu predstavljati dodatno sidro u rečeničnom stablu.

- Princip zajedničkog javljanja predikata i njegovih argumenata.** Ovaj princip zahteva da se svako elementarno stablo sastoji minimalno od predikata i svih njegovih argumenata, odnosno da se predikat i njegovi argumenti moraju javiti zajedno unutar istog elementarnog stabla. Ovaj princip je ujedno i ključni princip TAG gramatike i značajno utiče na odluke donete u vezi sa oblikom i načinom predstavljanja elementarnih stabala¹¹⁸.

Naspram ove podele i određenja, stoji ona za rečenice srpskog jezika kao predikatske rečenice date u (Stanojčić & Popović, 1997, pp. 194)¹¹⁹. Predikatske rečenice, kao prototip rečenica srpskog jezika, određene su kao rečenice koje sadrže glagol u ličnom glagolskom obliku i koji predstavlja njihovo leksičko jezgro. Predikatske rečenice su dalje podeljene na nezavisne rečenice (obaveštajne, upitne, zapovedne, željne i uzvične) i zavisne (izrične, odnosne, mesne, vremenske, uzročne, uslovne, dopusne, namerne, poredbene i posledične). Dok nezavisne predikatske rečenice imaju komunikativnu funkciju i mogu stajati samostalno, zavisne rečenice imaju konstituentsku funkciju i koriste se za građenje složenih predikatskih rečenica i sintagmi.

Kao osnovni tip predikatske rečenice (Stanojčić & Popović, 1997, p. 188) definišu subjekatsko-predikatsku rečenicu, tj. rečenicu koja se sastoji od glavnih rečeničnih konstituenata – subjekta i predikata. Osim toga, određeni glagoli zahtevaju dopune, što su u tom slučaju konstituenti koji se javljaju u rečenici zajedno sa subjektom i predikatom. Određene zavisne rečenice mogu vršiti funkciju ovakvih dopuna, ali isto tako i funkciju subjekta u predikatskoj rečenici.

¹¹⁸ Postoje još dva principa dobre formiranosti elementarnih stabala – princip koji zahteva da svako sintaksičko elementarno stablo ima odgovarajuće semantičko stablo, kao i princip kompozacionalnosti koji zahteva da uvek postoji samo jedno takvo stablo. Mi se u ovom radu nećemo baviti semantičkom komponentom TAG gramatike, tako da ove principe izostavljamo i o njihovim implikacijama neće biti reči u daljem radu.

¹¹⁹ Osnovna podela rečenica u datoj gramatici jeste na komunikativne rečenice (rečenice u širem smislu), kao jedinice komunikacije, i predikatske i specijalne rečenice (rečenice u užem smislu), od kojih nezavisne predikatske rečenice i specijalne rečenice služe za realizovanje komunikativnih rečenica. Za predikatske rečenice videti nastavak teksta. Specijalne rečenice nisu predikatske tj. ne sadrže glagol u ličnom glagolskom obliku. Primeri specijalnih rečenica su *Auto!* ili *Evo ga!*. U radu se nećemo dalje baviti ovim tipom rečenica.

Rečenice srpskog jezika se prema složenosti dalje dele na proste i složene. Prosta rečenica se sastoji od jedne nezavisne predikatske rečenice, dok se složena rečenica sastoji od najmanje dva dela – takozvane glavne ili upravne rečenice (Stanojčić & Popović, 1997, p. 292) i zavisne rečenice.

Jedno od najpre приметnih nepoklapanja pojmova elementarnog stabla i predikatske rečenice odnosi se na oblik glagola unutar njih. Dok je za predikatske rečenice srpskog jezika jasno određeno da se glagol mora nalaziti u ličnom glagolskom obliku da bi formirao predikat bilo nezavisne bilo zavisne rečenice, nigde u određenju elementarnih rečenica tako nešto nije definisano. Tako predikat u elementarnom stablu može biti i glagol koji se nalazi u neličnom glagolskom obliku, naime, glagol u infinitivu (Abeillé, 2002, p. 101), što u krajnjoj instanci proizlazi iz principa 2¹²⁰.

Ako dalje upoređujemo pojam srpske predikatske rečenice i elementarnog stabla, možemo se zapitati kakav je odnos između zavisne i nezavisne predikatske rečenice i inicijalnih i pomoćnih elementarnih stabala. Iako određeno poklapanje između ovih pojmova postoji (nezavisne rečenice = inicijalna stabla, zavisne rečenice = pomoćna stabla), ono ipak nije potpuno. Naime, dok su proste nezavisne rečenice uvek predstavljene kao inicijalna stabla, zavisne rečenice će ponekad biti predstavljene kao inicijalna stabla, a ponekad kao pomoćna, što nije određeno njihovom funkcijom u rečenici, već upravo principom 2. Ovo važi i za segment složene rečenice koji smo definisali kao glavnu rečenicu. Pogledajmo detaljnije ovaj odnos.

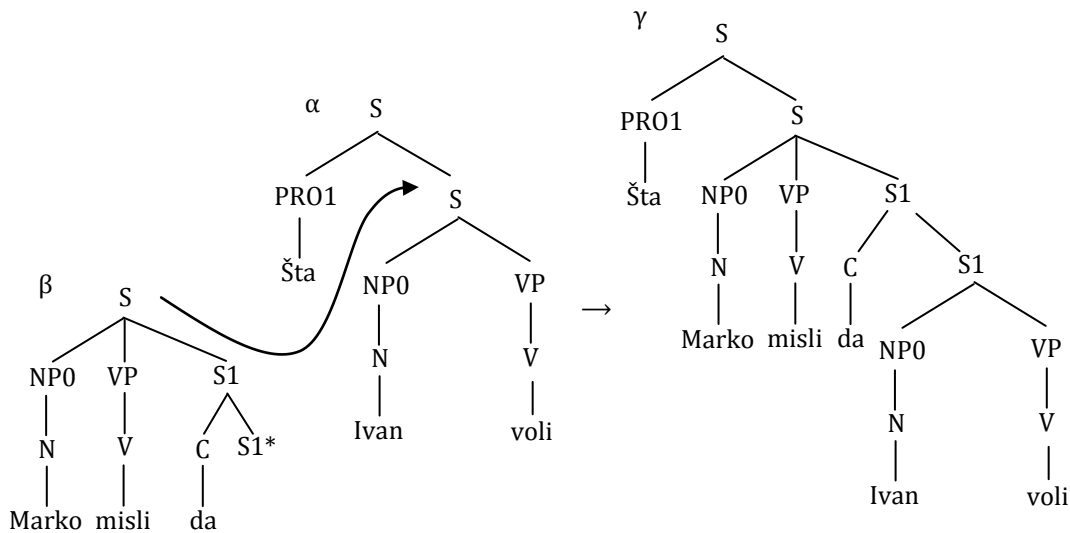
U poglavlju 2.4.6 smo pomenuli da je jedna od glavnih karakteristika FBLTAG (i TAG gramatika generalno) to da pripajanje ne utiče na domen zavisnosti. Ovo je karakteristika TAG koja direktno proizlazi iz navedenog principa 2. Objasnimo kako.

Glagol koji za svoju dopunu bira zavisnu rečenicu može biti sidro inicijalnog stabla s korenim čvorom S, ili pak pomoćnog stabla s korenim čvorom S. Ova razlika se pravi prema tome da li je moguće “izmestiti” argumente iz zavisne rečenice i napraviti pitanje kakvo je, na primer pitanje u rečenici 1, nastalo od „neizmeštene” rečenice kakva je rečenica 2:

¹²⁰ Za demonstraciju delovanja principa 2 u ovom slučaju pogledati isto poglavlje niže.

1. *Šta_i Marko misli da [Ivan voli t_i¹²¹].*
2. *Marko misli da Ivan voli jabuke.*

U ovakvom pitanju, ukoliko se *Šta Marko misli* analizira kao glavna rečenica, kao što naša tradicija analize složenih rečenica nalaže, argument *šta* glagola *voleti*, kao njegov pravi objekat, nalazi se van domena njegove elementarne rečenice (zavisne rečenice *voleti*). Ovakva analiza je, dakle, u neskladu sa ranije navedenim principom po kome svi argumenti predikata moraju zajedno s predikatom biti u istoj elementarnoj rečenici. Analiza koja se za „izmeštanje” iz dopunske zavisne rečenice koristi je tako ona koja podrazumeva postojanje stabla *Šta Ivan voli* kao bazičnog inicijalnog stabla, u koje se pripajanjem unosi segment *Marko misli da* u obliku pomoćnog stabla (Slika 36).



Slika 36 Izvođenje stabla za rečenicu *Šta Marko misli da Ivan voli?*

Na odnos glavnog i zavisnog dela rečenice se iz ovog razloga drugačije gleda u okviru TAG. Međutim, nije tretman svih zavisnih rečenica isti u TAG, i ponovo, vođen je istim polaznim principom. Nasuprot prvom navedenom primeru nalazi se onaj u kome nije

¹²¹ Oznaka *t* se koristi unutar transformacione gramatike da označi trag, odnosno poziciju sa koje je izmešten određeni argument. Trag je sa izmeštenim argumentom povezan indeksom *i*. Ovu notaciju koristimo samo da bi bilo jasnije odakle je argument izmešten, u TAG se ne koristi trag niti zaista postoji izmeštanje u tom smislu reči. Videti nastavak teksta za objašnjenje.

moгуće izmeštanje iz dopunske rečenice. Takav slučaj predstavljaju zavisnoupitne izrične rečenice¹²²:

3.

- a) *Marko pita [zašto Ivan voli jabuke].*
- b) **Šta_i Marko pita [zašto Ivan voli t_i].*

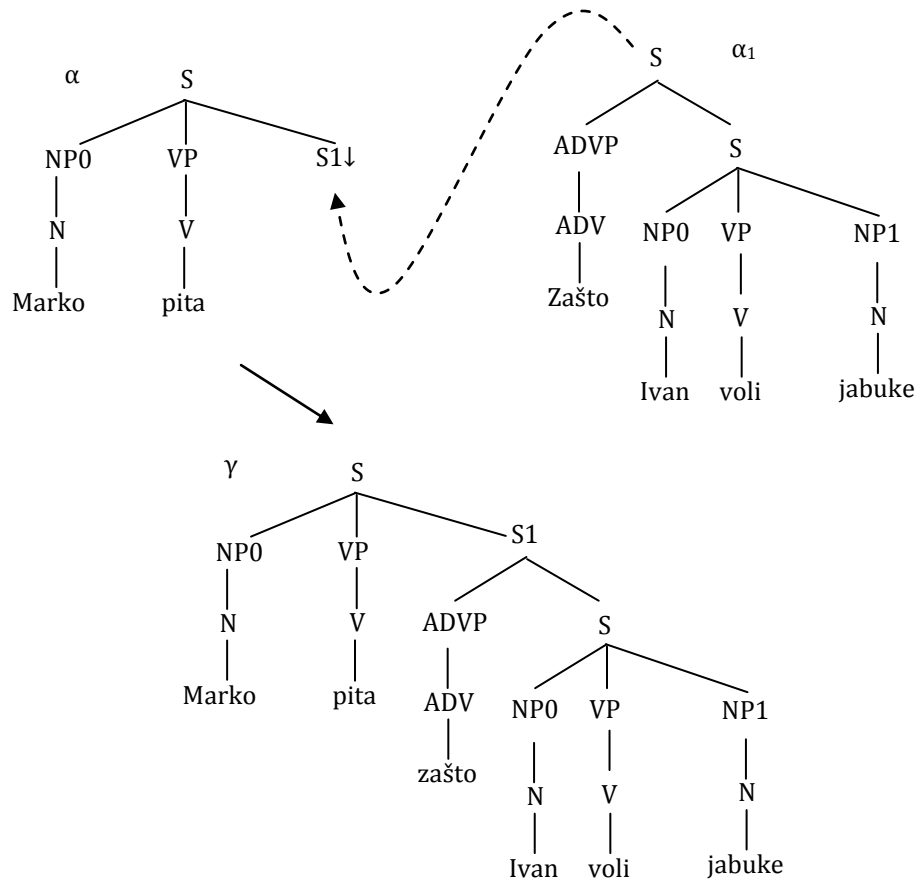
4.

- a) *Marko pita da li Ivan voli Mariju.*
- b) **Koga_i Marko pita [da li Ivan voli t_i].?*

Kako kod ovakvih rečenica nije moguće umetanje pomoćne rečenice *Marko pita zašto/da li* unutar upitne elementarne rečenice *Šta Ivan voli?*, nema potrebe da se zavisna rečenica analizira kao glavna. U ovom slučaju se ona analizira kao i bilo koji drugi argument glagola koji se u strukturu unosi kroz zamenu, te je predstavljena kao inicijalno stablo s korenim čvorom S, kao uostalom i struktura u kojoj se zamenjuje (glavna rečenica u našoj gramatičkoj tradiciji) (Slika 37).

Ako obratimo pažnju stablo β (Slika 36), možemo videti da u dato stablo kao deo strukture ulazi i veznik, tj. subordinator *da*. Ovo je još jedno mesto razmimoilaženja naša dva pristupa. Naime, prema (Stanojčić & Popović, 1997, p. 293) glavno obeležje zavisnih rečenica jeste njihov subordinator (zavisni veznik ili veznički spoj – *da, pre nego što...*, rečce – *li, da li*, odnosne i upitne zamenice, pridevi i prilozii – *ko, šta, gde, koliko, zašto...*), koji obavezno čini njihov sastavni deo. Subordinator bez sumnje ostaje obeležje zavisne rečenice i u TAG, ali se stuktorno ne mora nalaziti uz svoju zavisnu rečenicu. Tako se u pomenutom stablu on nalazi unutar “glavne” rečenice, dok se u stablu α_1 (Slika 37) nalazi unutar stabla zavisne rečenice. Ova odluka je direktna posledica već navedenog principa.

¹²² Termin zavisnoupitna izrična rečenica je termin gramatike (Stanojčić & Popović, 1997). Za punu klasifikaciju rečenica u ovoj gramatici, pogledati tabelu 19 u dodacima (Tabela 15).



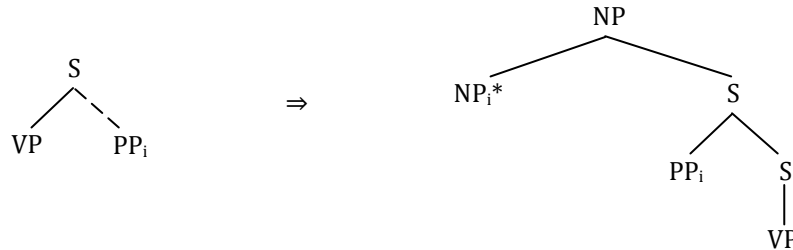
Slika 37 Izvođenje stabla za rečenicu *Marko pita zašto Ivan voli jabuke*.

Još jedna razlika u pristupu zavisnim rečenicama, a ponovo vezana za princip 2, odnosi se na tretman odnosnih rečenica. Prema (Stanojčić & Popović, 1997, p. 301) odnosne ili relativne rečenice predstavljaju vrstu zavisnih rečenica koje svoj sadržaj pripisuju imeničkim pojmovima na koje se odnose (ili nekoj rečenici), a koje se prepoznaju po takozvanim *relativizatorima* (*koji, kakav, čiji, gde, kuda...*). Ovakve konstrukcije se u TAG gramatici nazivaju *relativima* i smatraju se rezultatom leksičkog pravila izmeštanja (Abeillé, 2002, p. 220). Slika 38 prikazuje ovo pravilo. Njime se prikazuje odnos između inicijalnog elementarnog stabla kakvo je ono u primeru 5, i stabla s imeničkom frazom kao svojim korenim čvorom i umetnutom zavisnom rečenicom s istim glagolom (primer 6)¹²³. Jedna odnosna konstrukcija tako nije predstavljena kao S, već zajedno sa imeničkom frazom na koju se odnosi, čini deo veće imeničke fraze.

¹²³ Na mestu fraze PP s leve strane pravila (Slika 38) može se naći i imenička fraza. U tom slučaju će PP čvor s desne strane pravila biti zamenjen komplementizatorom (C) *koji* u odgovarajućem padežu (*Marko voli ženu → Žena koju voli*).

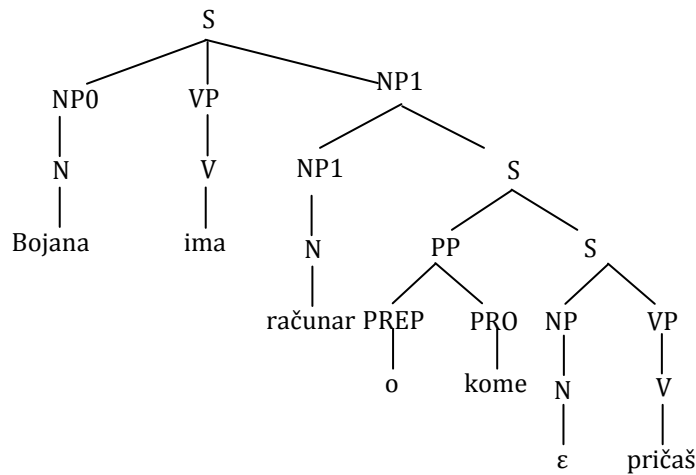
5. *Pričaš o računaru.*

6. *računar o kome pričaš*



Slika 38 Leksičko pravilo za relativizaciju

Isprekidana linija u stablu s leve strane pravila na istoj slici označava da će taj segment stabla biti izmešten. Slika 39 prikazuje izgled stabla za rečenicu *Bojana ima računar o kome pričaš.* dobijenu relativizacijom. U rečenici predstavljenoj stablom na datoj slici subjekat predikata *pričaš* je nerealizovan. Međutim, sve dok određeni argument postoji u argumentskoj strukturi glagola, on mora postojati i u elementarnom stablu, prateći isti princip dobre formiranosti. Nerealizovani argumenti glagola, bilo kog tipa, obeležavaju se oznakom za praznu reč – ϵ .



Slika 39 Elementarno stablo za rečenicu *Bojana ima računar o kome pričaš.*

Tretman infinitiva u TAG se, ponovo na osnovu istog principa, razlikuje od tretmana infinitiva u našoj referentnoj gramatici. Prema (Stanojčić & Popović, 1997, pp. 326–327)

infinitiv je centralni član jedinice pod nazivom *glagolska sintagma*. Prema daljem određenju, ovakvi glagoli nemaju svoj subjekat, te se uz njih javljaju samo njihove dopune. Glagolske sintagme dalje mogu imati imeničku konstituentsku vrednost i funkciju subjekta (primer 7), pravog objekta (primer 8) i ciljne (finalne) dopune (primer 9), kao i funkciju dopunskog dela predikata uz modalne i fazne glagole (primer 10).

7. ***Spavati je divno.***

8. *Ivan je želeo poći kući.*

9. *Otišao je spavati.*

10. *Hteo je/počeo je čitati knjigu.*

Ukoliko bismo u TAG ovu jedinicu analizirali kao glagolsku sintagmu (VP), naišli bismo najpre na nesklad u odnosu na tretman glagolskih dopuna generalno. Naime, za SrpTAG smo, prateći analizu i postavke date u (Abeillé, 2002, p. 80), kao i prateći analizu datu u (Stanojčić & Popović, 1997), dopune glagola ostavili van sintagme VP, kao argumente sa zasebnim frazalnim čvorom, kako bismo omogućili "izmeštanje" njihovih delova, kao i njihovo slobodno raspoređivanje u odnosu na glagol. Potrebno je, dakle, zarad uniformnosti analize u sistemu, ostaviti dopune ovakvih glagola van same VP fraze. Međutim, kako onda među njima uspostaviti vezu? TAG daje svoj specifičan odgovor na ovo pitanje.

U TAG, infinitiv se analizira kao sidro inicijalnog stabla s korenim čvorom S, koje, pored odgovarajućih dopuna, uz sebe ima i svoj subjekat. Ovakva analiza infinitiva, kao i svojevrsno opravdanje da se infinitiv može analizirati na ovaj način, usko su povezani sa analizom modalnih i faznih ili aspektualnih glagola. Naime, prema analizi datoj u (Stanojčić & Popović, 1997, p. 248), modalni i fazni glagoli, kao glagoli nepotpunog značenja, traže dopunu u vidu punoznačnog glagola u obliku *da*+prezent ili u obliku infinitiva. Prema daljoj klasifikaciji, modalni i fazni glagoli u ovakvoj konstrukciji čine modalni deo predikata, dok glagol u infinitivu ili konstrukciji *da*+prezent čine dopunski deo predikata, a zajedno formiraju konstrukciju pod nazivom *složeni predikat*.

Uvid u specifično ponašanje određenih modalnih i faznih glagola daje (Moskvljević Popović, 2007, p. 89) svrstavajući ih u glagole koji zahtevaju takozvano *obligatorno*

*subjekatsko podizanje subjekta*¹²⁴. Ni fazni ni modalni glagoli, navodi Moskovljević, ne dodeljuju svom subjektu semantičku ulogu, već subjekat uvek dele sa subjektom glagola uz koji se javljaju, odnosno – njihov subjekat je zapravo subjekat punoznačnog glagola uz koji stoje (primeri 11-13). U skladu sa time, kada punoznačni glagol u funkciji dopunskog dela predikata nema subjekat u svojoj strukturi, neće ga biti ni u rečenici sa modalnim ili faznim glagolom (primeri 14-16)¹²⁵.

11. *Marija peva.*

12. *Marija počinje da peva. / Marija počinje pevati.*

13. *Marija mora da peva. / Marija mora pevati.*

14. *°Grmi i seva.*

15. *°Počelo je da grmi i seva. / Počelo je grmeti i sevati.*

16. *°Posle onakve sparine, moralo je da grmi i seva. / Posle onakve sparine, moralo je grmeti i sevati.*

17. **Marija počinje da ona peva.*

18. **Marija mora da ona peva.*

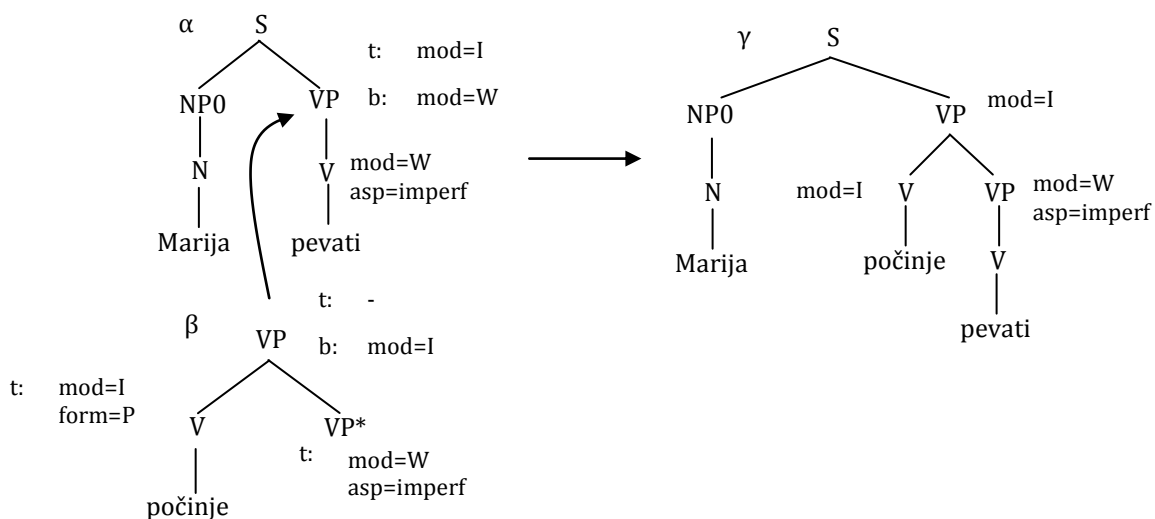
Ukoliko pokušamo da unesemo subjekat ispred punoznačnog glagola, rezultat će biti negramatična rečenica (primeri 17 i 18), što još jednom pokazuje da je subjekat u strukturi već prisutan, ali da je izmešten, odnosno „podignut” sa pozicije subjekta punoznačnog glagola.

Kako smo već videli u jednom od ranije navedenih primera, pojave izmeštanja argumenata iz originalne strukture u TAG zapravo nisu izmeštanje već proizvod pripajanja unutar inicijalne rečenice. Rečenica *Marija počinje pevati.* se tako tumači kao rezultat pripajanja pomoćnog stabla glagola *počinje* (Slika 40, stablo β) na inicijalno stablo *Marija pevati* (ista slika, stablo α). Stablo *Marija pevati*, obeleženo kao stablo α na datoj slici, zahteva pripajanje odgovarajućeg pomoćnog stabla, što je definisano neskladom između

¹²⁴ Ovi glagoli se nazivaju *glagolima podizanja* (engl. *raising verbs*, fran. *verbes „à monté”*) zbog efekta „podizanja” subjekta u strukturi, što je termin transformaciono-generativne gramatike. Pri „podizanju” se subjekat „pomera” levo u odnosu na poziciju u kojoj se nalazi, odnosno u „glavnu” rečenicu. Modalni glagoli koji su ovog tipa su *moći, morati, smeti i trebati*.

¹²⁵ Primeri obeleženi kružićem preuzeti su direktno iz (Moskovljević Popović, 2007, p. 89)

gornjeg i donjeg obeležja na čvoru VP glagola *pevati* – glagolska fraza je ovde obeležena istovremeno i kao indikativ (I) i kao infinitiv (W). Ovaj nesklad može biti razrešen jedino pripajanjem stabla koje na korenom čvoru nosi oznaku indikativa, a na priključnom čvoru oznaku infinitiva.



Slika 40 Pripajanje faznog glagola na stablo s glagolom u infinitivu

Po istoj logici, rečenica *Marija počinje da peva* trebalo bi da predstavlja rezultat pripajanja pomoćnog stabla *počinje da* na rečenicu *Marija peva*. Ipak, dok je ovakva analiza prikladna za rečenice s infinitivom, ona nije moguća za rečenice s prezentom. Naime, da bi pomoćno stablo *počinje da* došlo na ispravno mesto u rečenici – unutar VP fraze glavne rečenice, ono mora biti glagolsko stablo, odnosno njegov koreni i priključni čvor moraju nositi oznaku VP. Međutim, glagolska fraza ne može sadržati komplementizator u svojoj strukturi, te tako pomoćno stablo sa komplementizatorom *da*, kakvo je stablo *počinje da*, može biti samo rečenično stablo. Ne postoji, dakle, način da se rečenice s glagolima podizanja i glavnim glagolom u obliku *da+prezent* u srpskom jeziku analiziraju ovim mehanizmom i zasad ostaju nerešiv slučaj za ovu teoriju¹²⁶.

S druge strane, kod glagola koji u svojoj argumentskoj strukturi imaju subjekat a primaju dopunu u obliku konstrukcije *da+prezent* ili u obliku infinitiva, kod takozvanih

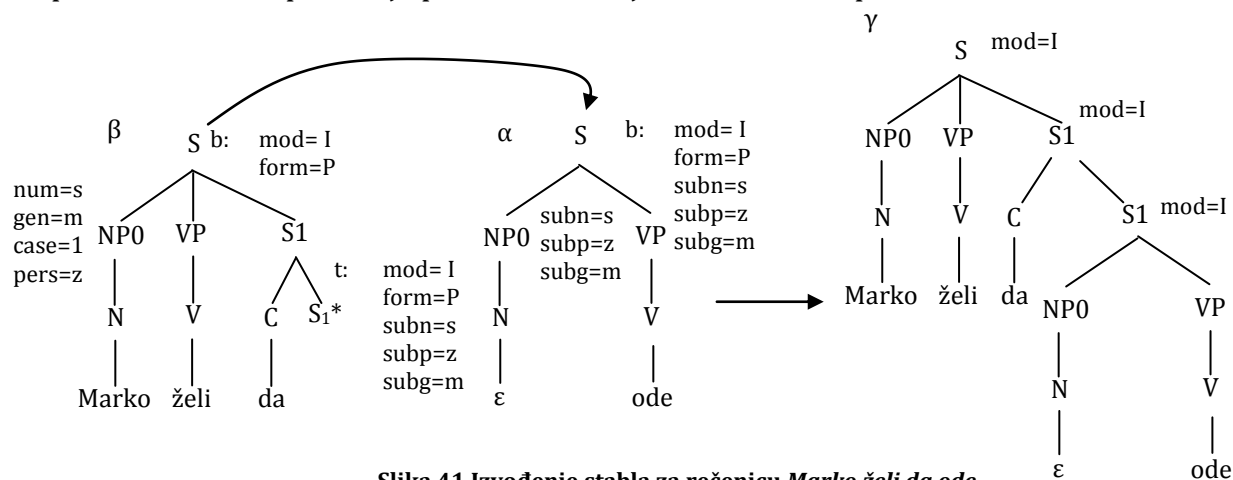
¹²⁶ Ukoliko želimo da ostanemo verni jezičkim principima.

glagola kontrole¹²⁷, subjekat ove dopunske rečenice (stablo α , Slika 41 i Slika 42) je nerealizovan i deli kategorije sa subjektom glagola kontrole (stablo β , Slika 40 i Slika 41) preko obeležja subn, subp i subg. Primeri 19 i 20 su primeri rečenica s glagolom kontrole.

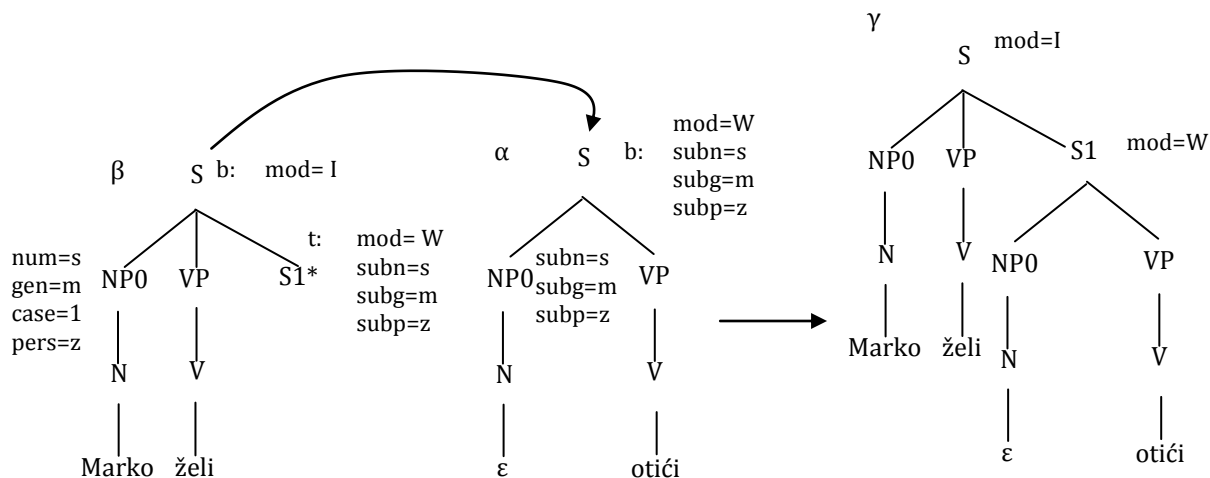
19. Marko_i želi da e_i ode.

20. Marko_i želi e_i otići.

Stablo koje predstavlja glagole kontrole je pomoćno stablo koje se u oba slučaja pripaja na stablo dopune. Slika 41 prikazuje primer izvođenja strukture s dopunom u obliku da+prezent. Slika 42 prikazuje primer izvođenja strukture s dopunom u obliku infinitiva.



Slika 41 Izvođenje stabla za rečenicu *Marko želi da ode*

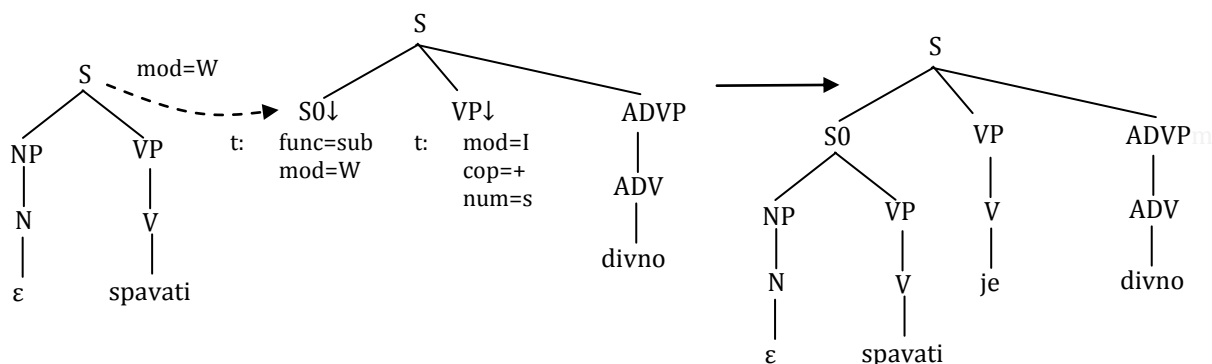


Slika 42 Izvođenje stabla za rečenicu *Marko želi otići*

¹²⁷ engl. *control verbs*, fran. *verbes „à contrôle“*. Neki od tih glagola su *otići*, *doći*, *nameravati*, *omogućiti*, *bojati se*, *želeti*, *hteti*, *mrzeti*, *pokušati*... (Moskvljević Popović, 2007, p. 95-105)

Osnovni princip kojim se možemo voditi pri određivanju načina na koji se kombinuju stabla jeste – ukoliko želimo da omogućimo da se delovi stabla jave udaljeno (kao u primeru *Marko misli da Ivan voli jabuke.* → *Šta Marko misli da Ivan voli.*), tretiraćemo ih kao inicijalna stabla na koja se vrši potencijalno pripajanje glavnog stabla (*Marko misli da*). Ukoliko pak želimo da sprečimo da se iz određenog segmenta vrši izmeštanje, tretiraćemo takva stabla kao inicijalna, koja se u strukturu glavne rečenice unose zamenom.

Infinitivne konstrukcije u funkciji subjekta se po analogiji sa prethodnim primerom takođe mogu prikazati kao rečnično stablo sa nerealizovanim subjektom, međutim ovog puta se unose zamenom u inicijalno stablo glavne rečenice, kako ne želimo da dozvolimo bilo kakvo izmeštanje iz ovog segmenta (Slika 43).



Slika 43 Izvođenje stabla za rečenicu *Spavati je divno*

3.4 Pojam i sastav predikata

(Stanojčić & Popović, 1997, p. 212) sve predikate dele na:

1. Glagolski
2. Kopulativni, i to:
 1. imenski kopulativni (kopula + imenski predikativ)
 2. priloški kopulativni (kopula + priloški predikativ)

Glagolski predikat se u pomenutoj gramatici definiše kao glagol u “ličnom (=finitnom) obliku, potvrđnom ili odričnom, kongruentnom sa subjektom” (Stanojčić &

Popović, 1997, p. 213). Podela glagola koja se u daljem tekstu pravi bazirana je na njihovoj valentnosti, odnosno sposobnosti da uz sebe vežu subjekat (jednovalentan glagol) i dopune (jedna dopuna – dvovalentan glagol, dve dopune – trovalentan glagol). Ova podela je u vezi sa glagolskom kategorijom koju autori nazivaju glagolski rod u širem smislu (Stanojčić & Popović, 1997, pp. 99–100), a po kojoj svi glagoli mogu biti:

- **prelazni** (zahtevaju objekat u obliku akuzativa bez predloga, što ih čini najmanje dvovalentnim glagolima):
 21. *Marko zna **Ivana**.*
 22. *Marija je dala Ivanu **jabuku**.*
 23. *Ivan je stavio **knjigu** na sto.*¹²⁸
- **neprelazni** (ne zahtevaju objekat u obliku akuzativa, što ih čini najmanje glagolima s nultom valentnošću):
 24. *Sviće.*
 25. *Spavam.*
 26. *Ivan pomaže **bratu**.*
 27. *Boli me zub.*
 28. *Stanujemo **ovde**.*
 29. *Stanujemo **u gradu**.*
- **povratni** (u rečniku se pojavljuju sa rečcom *se*)
 30. *Smrkava **se**.*
 31. *Šalim **se**.*
 32. *Radujem **se** pismima.*
 33. *Nalazim **se** u centru.*

Kopulativni predikat se definiše kao konstrukcija kojom se subjektu pripisuje određeno svojstvo, kojom se on identifikuje ili mu se dodeljuje izvestan priloški sadržaj (Stanojčić & Popović, 1997, p. 216). Realizuje se kao kopula – glagol *jesam/bit* u ličnom obliku, i imenski ili priloški deo predikata (predikativ):

¹²⁸ Podvučeni delovi rečenica su takođe dopune. Nećemo ih ovde dalje definisati, o svakoj od njih će biti više reči detaljno u narednim poglavljima.

- **Imenski kopulativni predikativ** se javlja kao imenička jedinica¹²⁹ u nominativu bez predloga (34), ili kao pridevska jedinica¹³⁰ u nominativu (35). Pridevi koji razlikuju vid stoje u neodređenom vidu. Zajedno sa kopulom obrazuje imenski predikat.

34. *Marko je **kuvar**. / To je **on**.*

35. *Dan je **lep**. / Taj telefon je **moj**. / Marko je **prvi**.*

- **Priloški kopulativni predikativ** se javlja kao priloška jedinica¹³¹ (36), imenička jedinica u zavisnom padežu (37) ili kao predložko-padežna konstrukcija s priloškim značenjem (38).

36. *Centar je **daleko**. / Centar je **jako daleko**.*

37. *Utakmice su **sredom**.*

38. *Marija je **kod kuće**.*

Na prelazu između ove dve grupe glagola, kao vrsta koja sadrži obeležja i jedne i druge grupe, jesu takozvani *nepravi kopulativni glagoli* (Stanojčić & Popović, 1997, p. 227). Nepravi kopulativni ili semikopulativni glagoli su prelazni ili neprelazni glagoli koji sadržaj svoje dopune – dopunskog predikativa, pripisuju svom subjektu (kada su neprelazni) ili objektu (kada su prelazni). Dopunski predikativ se može realizovati kao imenička ili pridevska jedinica u nominativu (39) ili instrumentalu bez predloga (40). Može se realizovati i kao predložko-padežna konstrukcija *za+akuzativ* (41). Autori još naglašavaju da ukoliko pridev razlikuje vid, u nominativu i akuzativu jednakom nominativu stoji u neodređenom vidu.

39. *Marko je postao **predsednik**. / Vreme je postalo **hladno**.*

40. *Marka su proglasili **pobednikom**. / Nazvao ga je **ludim**.*

41. *Marija je izrabrala Tanju **za deverušu**.*

¹²⁹ Imenica, imenička sintagma, imenička zamenica

¹³⁰ Pridev, pridevska sintagma, pridevska zamenica ili redni broj

¹³¹ Prilog, priloška sintagma

Tabela 9 Tipovi predikata (tabela uz neznatne izmene preuzeta iz (Stanojčić & Popović, 1997, p. 250))

	glagolski predikat	kopulativni predikat	
		imenski	priloški
prosti predikat	<i>radim</i>	<i>bio sam odličan</i>	<i>bio sam tamo</i>
složeni predikat	<i>moram da radim/ moram raditi</i>	<i>moram da budem odličan/ moram biti odličan</i>	<i>moram da budem tamo/ moram biti tamo</i>

Pored ove podele, predikat se u datoj gramatici razvrstava prema složenosti na prost i složen (Stanojčić & Popović, 1997, p. 249), pri čemu je složen onaj koji uz sebe ima modalni (*morati, moći, hteti, smeti, trebati*)¹³² ili fazni glagol (*početi, počinjati, nastaviti, nastavljati, prestati, prestajati...*) (Tabela 9).

Uzimajući u obzir sve navedene podele, u sastav predikata u gramatici (Stanojčić & Popović, 1997) tako ulaze:

- modalni i fazni glagol i dopunski deo predikata, formirajući složeni predikat;
- kopula i kopulativni predikativ, formirajući imenski/priloški predikat;¹³³

Kako nas za potrebe formalne gramatike zanima i strukturni sastav samog glagola, pored navedenih delova predikata, možemo reći da, tehnički gledano, u sastav predikata ulaze i sledeći segmenti:

- pomoćni glagoli *hteti* i *jesam/biti* u složenim glagolskim oblicima;
- pomoćni glagol *jesam/biti* i glagol kod glagola u pasivu;
- rečca *se*;
- rečca *ne*.

¹³² Ove glagole autori navode kao najvažnije (Stanojčić & Popović, 1997, p. 247). U (Mrazović, 2009, p. 181,185) se u modalne glagole ubrajaju i *umeti, želeti, znati* i *imati* - *Ujutro ima rano da ustaneš... Miloš zna lepo da priča*. (Moskovljević Popović, 2007, p. 91) navodi i *valjati* i *vredeti*, kao bezlične glagole – *Valja ustati rano, Vredi raditi*. U prethodnom poglavlju smo već videli da status ovih glagola ipak nije isti – glagoli *želeti* i *hteti* spadaju u glagole kontrole, za razliku od ostalih modalnih glagola koji spadaju u glagole podizanja.

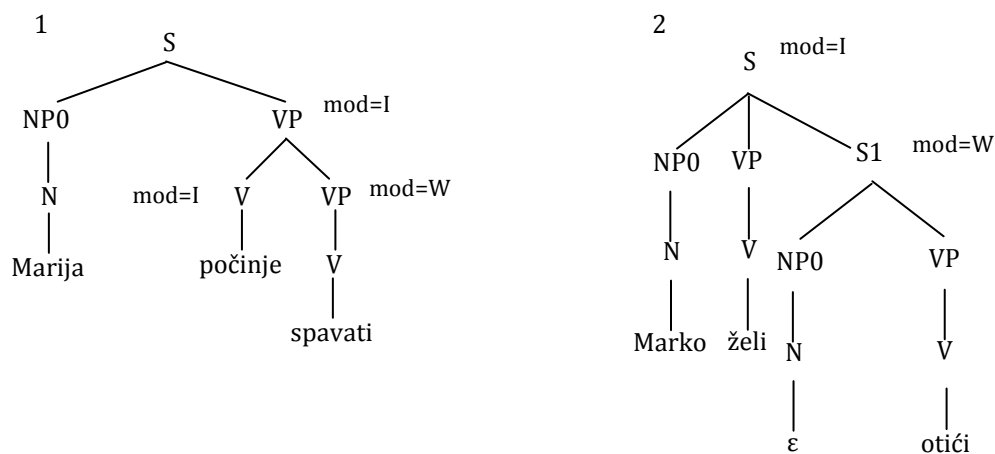
¹³³ U konstrukciji sa dopunskim predikativom, ova jedinica se smatra dopunom glagola, a ne njegovim sastavnim delom.

Stavke navedene u drugoj podeli se u našoj referentnoj gramatici ne vide kao deo predikata koliko kao karakteristike samog glagola, definisane kroz kategorije glagolskog roda u užem smislu¹³⁴, po kojoj se glagoli dele na aktivne, medijalne i pasivne, glagolskog roda u užem smislu, po kojoj se glagoli dele na prelazne, neprelazne ili povratne, glagolskog vremena, koje može biti prosto ili složeno i kategorije potvrdnosti/odričnosti.

U poređenju sa ovakvim pristupom, sastav glagolske fraze u TAG je nešto drugačiji. Pogledajmo kako izgleda svaka od konstrukcija navedenih u prethodne dve podele.

3.4.1 Modalni i fazni glagoli u TAG

Kao što smo mogli videti u prethodnom poglavlju, unutar glagolske fraze u TAG mogu se naći fazni i oni modalni glagoli koji imaju karakteristike glagole podizanja (Slika 44, stablo 1). U suprotnom, ukoliko modalni glagol pripada glagolima kontrole, biće predstavljen zasebnom glagolskom frazom, a njegov dopunski deo unutar stabla zavisne rečenice (Slika 44, stablo 2).



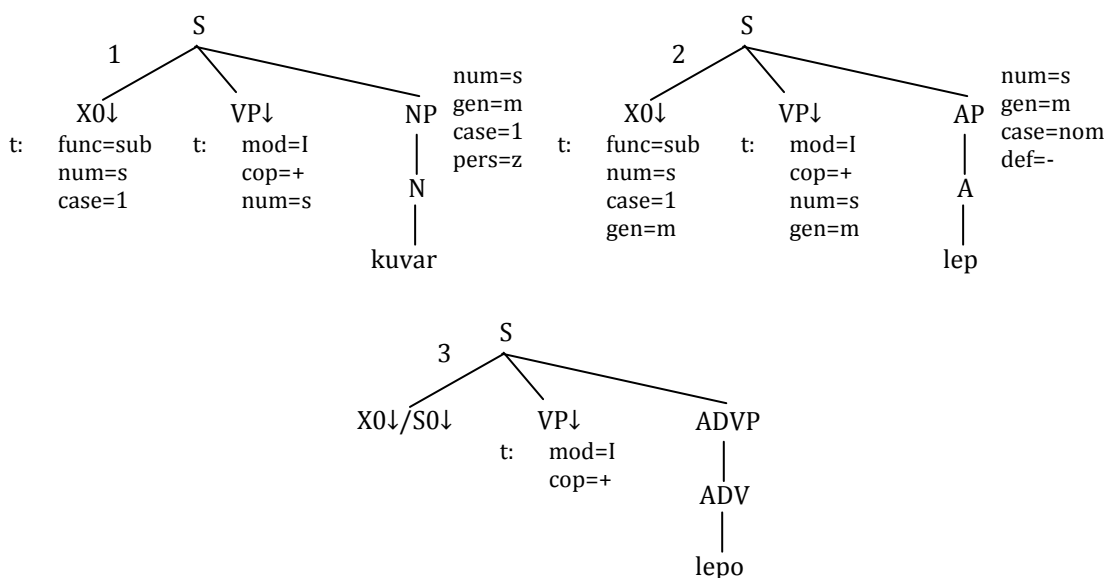
Slika 44 Stabla s modalnim glagolima

3.4.2 Kopulativni i semikopulativni glagoli u TAG

Kopulativne i semikopulativne konstrukcije se unutar TAG takođe definišu nešto drugačije nego u našoj referentnoj gramatici. U sastav glagolske fraze u TAG ulazi samo

¹³⁴ Dijateze ili glagolskog stanja

kopula. S druge strane, sidro kopulativne konstrukcije je samo predikativ. Kopula se u ove konstrukcije unosi zamenom (Slika 45).



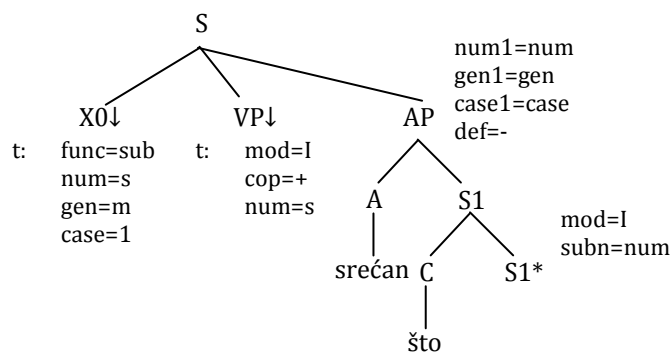
Slika 45 Stabla za kopulativni predikativ

Kopula se ne smatra nosiocem ove konstrukcije (sidrom ili makar drugim sidrom) iz nekoliko razloga:

- njena vrednost je pre svega funkcionalna, odnosno povezivačka, čak i kada je glagol punoznačan (semikopulativan). U TAG se u kopulativne glagole ubrajaju i neki glagoli koji se u našoj referentnoj gramatici definišu kao nepravi kopulativni ili semikopulativni glagoli: *postajati, činiti se, izgledati, dokazivati se...* (Abeillé, 2002, p. 149).
- predikativ kao sidro konstrukcije može direktno uticati na izbor subjekta – npr. na to da li se subjekat može javiti u formi infinitivnog rečeničnog stabla (Slika 45, stablo 3).
- u ovom tipu rečenica predikativi su ti koji primaju dopune – i to iz kojih je izmeštanje uvek moguće (Abeillé, 2002, p. 149), a ne sam kopulativni glagol. Da bi “izmeštanje” poput onog prikazanog u primeru 42 bilo moguće, takvi predikativi se predstavljaju kao sidro pomoćnog stabla¹³⁵ (Slika 46).

42. [Marko je srećan što] vidi Mariju. → Koga [je Marko srećan što] vidi?

¹³⁵ Sa eventualnim drugim sidrom, kao što je komplementizator *što* (Slika 46).



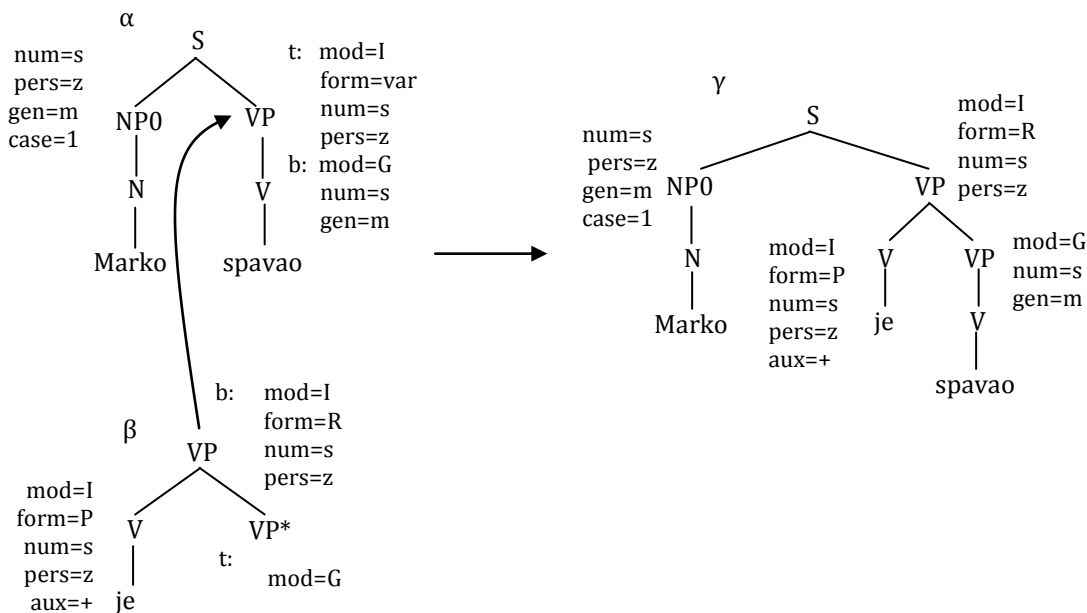
Slika 46 Pomoćno stablo za predikativ sa rečeničnom dopunom

3.4.3 Pomoćni glagoli u TAG

Srpski jezik ima dva pomoćna glagola: *jesam/biti* i *hteti*, koji se koriste za građenje složenih vremena (prošlog vremena, budućeg vremena, pluskvamperfekta) i načina (potencijala i futura 2). Tipično čine deo predikata, zajedno sa punoznačnim glagolom uz koji stoje. U TAG se pomoćni glagol takođe nalazi unutar glagolske fraze, ali kao dodatak, odnosno jedinica koja u glagolsku frazu ulazi pripajanjem. Za to se navode dva razloga (Abeillé, 2002, p. 111):

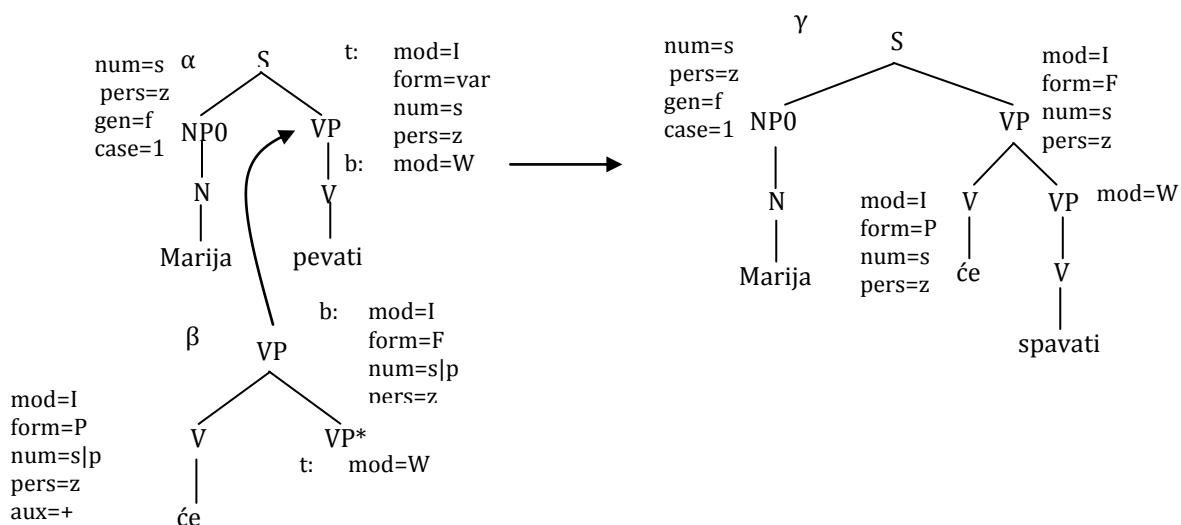
- punoznačni glagol je taj koji određuje broj i strukturu argumenata, ne pomoćni glagol.
- subjekat je i sa semantičke i sa sintaksičke strane subjekat glavnog glagola, a ne pomoćnog glagola.

U SrpTAG pomoćne glagole u neakcentovanom obliku predstavljamo kao sidra pomoćnih stabala čiji je koreni čvor VP, dok ih same obeležavamo kao glagole koji nose obeležje *aux=+*. Kao i kod modalnih glagola, ovakva analiza stoji za sve složene glagolske oblike sem onih koji se sastoje od konstrukcije *da+prezent* (fatur), što je, još jednom, konstrukcija koja ne može biti analizirana tako da se ispoštuju i jezički principi i principi same TAG. Kao i na stablima s glagolom u infinitivu iz primera s modalnim glagolima, i na stablima participa (Slika 47, stablo α), tj. infinitiva (Slika 48, stablo α) pripajanje odgovarajućeg pomoćnog glagola se zahteva zahvaljujući međusobno nekompatibilnim gornjim i donjim obeležjima na čvoru VP.



Slika 47 Izvođenje konstrukcije s glagolom u prošlom vremenu

Promenljiva var kao vrednost atributa form u gornjim obeležjima čvora VP u stablima α (Slika 47, Slika 48) označava bilo koju vrednost koja se propisuje za dati atribut. U ovom slučaju, ona će biti unifikovana sa vrednošću atributa form u donjim obeležjima čvora VP u stablu β , pa će tako dobiti vrednost R (prošlo vreme) (Slika 47), odnosno vrednost F (buduće vreme) (Slika 48).



Slika 48 Izvođenje konstrukcije u budućem vremenu

Ukoliko je sam pomoćni glagol složen, kao u slučaju davno prošlog vremena (*sam bio radio*), vrši se duplo pripajanje – jedno na samom pomoćnom stablu, kada se pomoćni glagol u obliku prezenta pripaja na drugi pomoćni glagol u obliku radnog glagolskog prideva, a zatim i pripajanje takvog izvedenog pomoćnog stabla na osnovno stablo.

3.4.4 Pasivni glagoli u TAG

Dok će o samoj pasivnoj konstrukciji biti reči u poglavlju 3.6.1.1. Ovde ćemo govoriti samo o strukturi pasivnog glagola. Osnovna razlika u tretmanu pasivnog glagola između naše referentne gramatike srpskog jezika i TAG jeste u tretmanu glagola *jesam/bit*. U gramatici (Stanojčić & Popović, 1997, p. 115) ovaj deo pasivnog glagola smatra se pomoćnim glagolom, koji zajedno sa glagolom u obliku glagolskog prideva trpnog čini pasivni glagol.

U TAG s druge strane, pasiv se smatra potpuno ekvivalentnim konstrukciji sa kopulativnim predikatom. Kao i u ranije predstavljenim kopulativnim konstrukcijama, za nosioca (sidro) pasivne konstrukcije uzima se glagol u obliku glagolskog prideva trpnog, dok se glagol *jesam/bit* ne smatra pomoćnim glagolom, već kopulom. Kao jedan od argumenata koji (Abeillé, 2002, p. 182-183) navodi u prilog ovakvoj analizi za francuski jeste razlika između oblika pomoćnog glagola kod aktivnih glagola (primeri 43 i 45) i glagola u istom glagolskom vremenu u pasivnoj konstrukciji (primeri 44 i 46).

43. *Jean est arrivé hier.* (prošlo vreme, aktiv)

Žan je stigao juče. (prošlo vreme, aktiv)

44. *Jean a été soigné par Marie.* (prošlo vreme, pasiv)

Žan je negovan od strane Mari. (prošlo vreme, pasiv)¹³⁶

45. *Quand Jean a été arrivé...* (davno prošlo vreme, aktiv)

Kad je Žan bio stigao... (davno prošlo vreme, aktiv)

46. *Quand Jean a eu été soigné...* (davno prošlo vreme, pasiv)

¹³⁶ (Stanojčić & Popović, 1997, pp. 115–116) oblik sa prezentom glagola *jesam/bit* (*Posao je urađen.*) smatraju isključivo prošlim vremenom. Isto i u (Piper et al., 2005). Ipak, (Mrazović, 2009, p. 109) ovakav oblik smatra jednim od oblika prezenta pasiva.

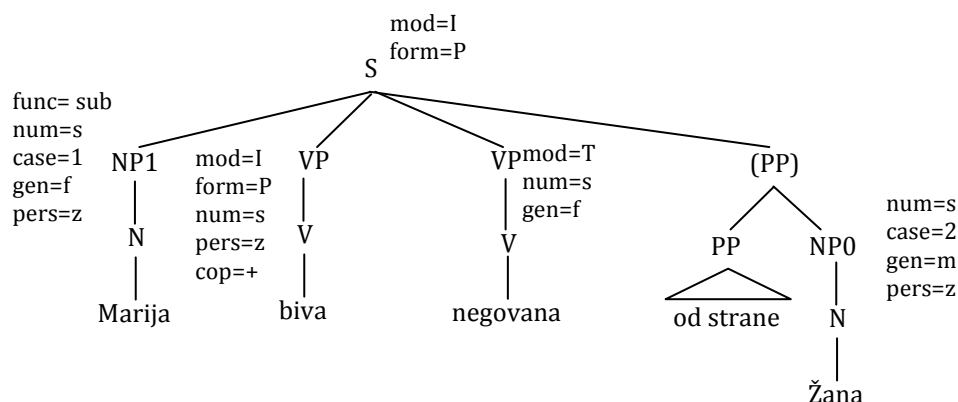
Kad je Žan bio negovan... (davno prošlo vreme, aktiv)

Ovaj argument međutim ne stoji za srpski jezik. Kao što se može videti iz prevoda primera za francuski jezik, oblik pomoćnog glagola u odgovarajućim složenim vremenima je isti u oba navedena vremena te se ovakva paralela sa francuskim ne može napraviti. Međutim, razlika se može jasno uočiti u prezentu, u kome je kod glagola u pasivnom obliku i dalje potreban glagol *jesam/biti*, i to u obliku netipičnom za pomoćne glagole.

47. *Marija stiže.* (prezent, aktiv)

48. *Marija biva negovana od strane Žana.* (prezent, pasiv)

Ovaj argument – postojanje glagola *jesam/biti* u pasivu u svakom od glagolskih vremena i načina, što ga čini obeležjem pasiva, je naš argument za njegovo razlikovanje od pomoćnih glagola, kao jedinica koje se javljaju samo u pojedinim vremenima i načinima. U nedostatku drugih argumenata za ili protiv TAG analize pasiva, prihvatili smo je kao takvu u ovoj verziji rada. Slika 49 daje prikaz stabla s pasivnom konstrukcijom.



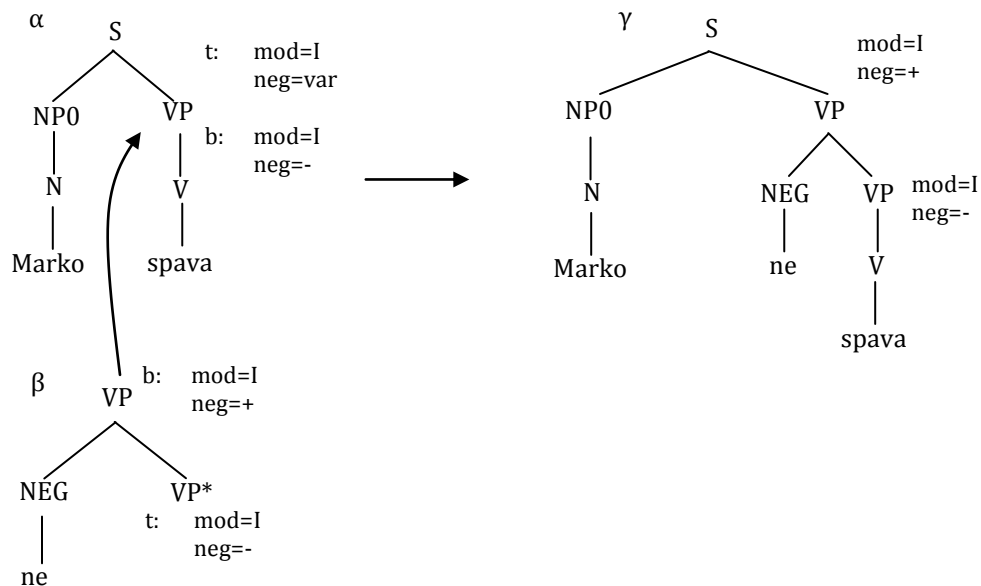
Slika 49 Stablo pasivnog glagola

3.4.5 Negacija

Rečca *ne*, još jedan element koji se u gramatici (Stanojčić & Popović, 1997) uvek nalazi u okviru predikata, to mesto ima i u TAG. Rečcu *ne* smatramo sidrom pomoćnog

glagolskog stabla, koje se pripaja na glagol u obliku indikativa. Pripajanjem čvora NEG na VP frazu, vrednost njenog obeležja *neg*, koje je dotad imalo neodređenu vrednost, dobija pozitivnu vrednost. Većina leksema u leksikonu ima negativnu vrednost ovog obeležja, sem negativnih oblika glagola *hteti* i *jesam/bit*, kao i oblika glagola *nemati*.

Kako je svrha pomoćnih stabala u TAG generalno da omoguće rekurziju, ni ovde, kao ni u prethodnim slučajevima u kojima smo koristili pomoćna stabla (modalni i pomoćni glagoli), ne postoji način da onemogućimo više pripajanja istog pomoćnog stabla na isto mesto u polaznom elementarnom stablu. Tako je na donji VP čvor u stablu γ (Slika 50) i dalje moguće pripajanje istog stabla za negaciju, koje bi u tom slučaju dalo rečenicu *Marko ne ne spava*. Međutim, ovakva dupla ili višestruka pripajanja je moguće sprečiti u metagramatici, time što će gornji čvor VP dobiti obeležje NP – zabranjeno pripajanje (videti poglavlje 4.5).



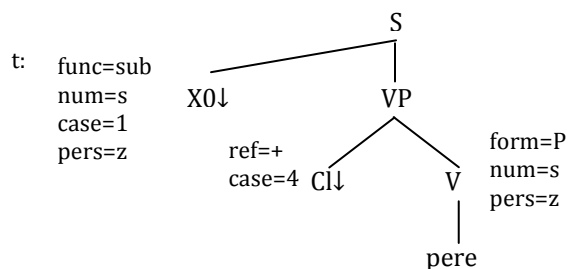
Slika 50 Izvođenje stabla sa negacijom

3.4.6 Refleksivni glagoli

Refleksivna rečca *se*, prema klasifikaciji datoj u (Stanojčić & Popović, 1997, p. 100), stoji uz tri vrste glagola – prave povratne, u kojima se tumači kao akuzativ zamenice *sebe* (*češljati se = češljati sebe, umivati se = umivati sebe*), uzajamno povratne, kod kojih više vršilaca vrše radnju jedni na drugima (*tući se, sprijateljiti se*) i nepravne povratne, za koje se

kaže da rečca *se* ne može biti tumačena kao akuzativ zamenice *sebe* (*ljutiti se, bojati se*). U gramatici nismo pronašli primer analize prva dva tipa glagola, te nismo sigurni kako se u rečenici analizira rečca *se* u odnosu na glagol¹³⁷. U slučaju treće grupe pak, rečca *se* čini sastavni deo glagola.

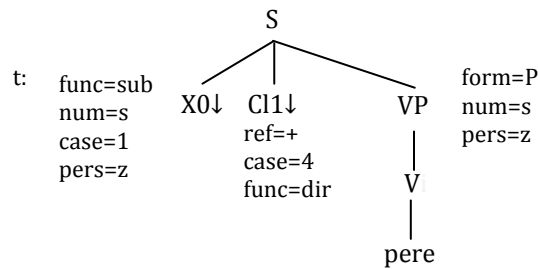
U francuskoj TAG, prva dva tipa refleksivnih konstrukcija nastaju kao proizvod leksičkog pravila refleksivizacije. Refleksivizacija kao pravilo redistribucije u francuskoj TAG vrši detranzitivizaciju prelaznog glagola (npr. *laver – prati (nešto)*), čime on postaje neprelazan (*se laver – prati se, kupati se*). Rečca *se* ostaje argument glagola samo semantički; u strukturi ona se nalazi u okviru VP fraze, u koju se unosi zamenom (Abeillé, 2002, p. 190). Ovo praktično znači da za ovakve glagole u leksikonu postoji samo jedan unosak – za njih kao prelazne, nerefleksivne glagole. Slika 51 predstavlja adaptiranu verziju stabla glagola *prati* nastalog refleksivizacijom.



Slika 51 Stablo glagola *prati* dobijeno procesom refleksivizacije

Druga mogućnost za tumačenje pravih i uzajamno povratnih glagola za srpski jezik jeste da su oni proizvod pravila kliticizacije. Pravilo kliticizacije je pravilo realizacije, tako da ne potiskuje sintaksičku funkciju direktnog objekta, već samo menja njegov pojavni oblik. U ovom slučaju, ovi povratni glagoli bi se i dalje tumačili kao prelazni, a rečca *se* bi bila u istom rangu kao i enklitički oblik ličnih zamenica u funkciji pravog objekta, i samim tim bi i dalje nosila funkciju pravog objekta (Slika 52).

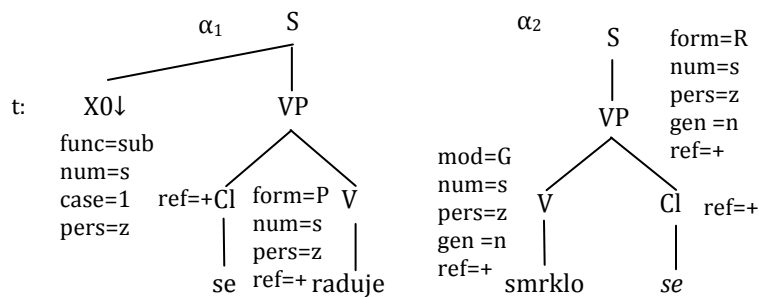
¹³⁷ Kao pravi objekat ili kao deo samog glagola.



Slika 52 Stablo glagola *prati* nastalo procesom kliticizacije

I jedna i druga analiza imaju svoje prednosti. Analiza u kojoj je rečca *se* unutar glagolske fraze po tome je identična analizi nepravih povratnih glagola¹³⁸, čime bi se postigla konzistentnost u predstavljanju refleksivnih glagola uopšte. S druge strane, predstavljanje rečce *se* kao čvora koji je direktno podređen čvoru S i paralelan drugim čvorovima dopuna, omogućio bi potpuno slobodnu primenu mehanizma raspoređivanja enklitika, bez pravljenja dodatnih stabala za rečcu *se* u zavisnosti od njene pozicije unutar VP, kao što bi se to moralo činiti kod nepravih nepovratnih glagola. U ovoj fazi rada na SrpTAG ipak smo odlučili da to bude prva analiza, sa rečcom *se* unutar fraze VP.

Za treću grupu glagola, koji su inherentno refleksivni, *se* čini deo leksičkog unoska samog glagola i posledično predstavlja drugo sidro inicijalnih rečenica u kojima je glagol prvo sidro. Slika 53 tako daje izgled stabala za glagole *radovati se* ili *smrći se*.



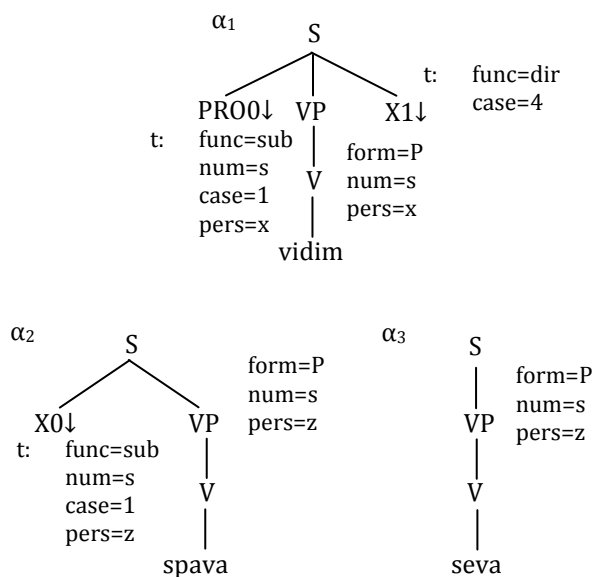
Slika 53 Stabla inherentno refleksivnih (nepravih povratnih) glagola

¹³⁸ Videti tekst u nastavku.

Kao što se može primetiti, refleksivna rečca se može javiti sa leve ili sa desne strane glagola. U određivanju toga koji ćemo od ova dva redosleda uzeti za primarni možemo se voditi bilo načinom na koji se refleksivni glagoli obično navode, sa rečcom *se* iza glagola (*radovati se*) ili onako kako se najčešće raspoređuju u rečeničnim konstrukcijama. Budući da smo već utvrdili da je u našim rečenicama podrazumevamo prisustvo subjekta kao prvog rečeničnog konstituenta za sve subjekatsko-predikatske rečenice, za osnovni redosled kod ovog tipa rečenica uzećemo onaj gde se rečca *se* nalazi ispred glagola. U stablima u kojima je subjekat izostavljen, alternativno stablo je sa osnovnim povezano leksičkim pravilom realizacije. Kod glagola koji u svojoj strukturi nemaju subjekat, osnovni redosled biće onaj u kome je rečca *se* iza glagola.

Činjenica da glagol i njegov dodatak, ovde rečca *se*, predstavljaju jedan leksički unosak, kao paralelna sidra, ne podrazumeva istovremeno i to da se ove jedinice moraju u stablu nalaziti jedna pored druge. One mogu biti međusobno udaljene – između njih je moguće pripajanje, kao što je moguće i da se izmešteni argumenti nađu između njih.

Za sam kraj navodimo neke minimalne strukture u kojima se mogu naći prelazni i neprelazni glagoli (Slika 54). Stablo α_1 je primer stabla dvovalentnih prelaznih glagola, s pravim objektom u obliku akuzativa bez predloga. Stablo α_2 je primer stabla jednovalentnih glagola, dok je stablo α_3 primer stabla glagola nulte valence, ili bezličnih glagola koji u svojoj strukturi nemaju subjekat.



Slika 54 Minimalna elementarna stabla glagola *videti*, *spavati*, *sevati*

3.5 Glagolski argumenti

3.5.1 Realizacija argumenata glagola

Glagolski argumenti se mogu realizovati na tri načina:

1. Kao pojedinačne kategorije
2. Kao fraze
3. Kao zavisne rečenice

1) Glagolski argumenti se realizuju kao dve pojedinačne kategorije – kao imenička zamenica (PRO)¹³⁹ i kao enklitički oblik imeničke zamenice¹⁴⁰ (Cl). Iako se u retkim situacijama može vršiti pripajanje čak i na zamenice¹⁴¹, odlučili smo da za njih ne uvodimo frazalnu kategoriju (PROP), budući da mahom nisu modifikovane. Pridevske zamenice se ne javljaju kao samostalni čvorovi pod čvorom S već isključivo kao modifikatori u okviru imeničkih fraza.

Slika 55 prikazuje tipična stabla imeničkih zamenica. Stablo 1 prikazuje čvor imeničkih zamenica koji se kao takav javlja na odgovarajućem mestu u glavnom rečeničnom stablu. Vrednosti obeležene promenljivama var_i ovde koristimo da označimo bilo koju vrednost unutar datog atributa. Na slikama α_{1-3} data su redom leksikalizovana inicijalna stabla lične, upitne i negativne zamenice, koje se međusobno razlikuju po vrednostima atributa *neg* (negacija) i *wh* (pitanje).

1	t:	func=var	α_1	PRO	func=var	α_2	PRO	func=var	α_3	PRO	func=var
		num=var0			num=s			num=s			num=s
		pers=var1			pers=z			pers=z			pers=z
	PRO↓	gen=var2			gen=m			gen=m			gen=m
		case=var3		on	case=1		ko	case=1		niko	case=1
		neg=+ -			neg=-			neg=-			neg=+
		wh=+ -			wh=-			wh=+			wh=-

Slika 55 Imeničke zamenice u SrpTAG

¹³⁹ Zamenica se može javiti i kao realizacija predikativa u kopulativnim i semikopulativnom konstrukcijama, ali kako predikativ ne smatramo argumentom već predikatom, ovde ga nećemo zasebno predstavljati. Za analizu kopulativne konstrukcije upućujemo nazad na poglavlje 3.4.2.

¹⁴⁰ Ili zamenice za svako lice *sebe, se* u slučaju refleksivnih glagola.

¹⁴¹ Kod odnosnih rečenica – ONAJ [koji je došao juče].

Slika 56 prikazuje izgled čvora za zameničke enklitike sa odgovarajućim atributima. Zameničke enklitike, kao i imeničke zamenice, u stablo se unose zamenom. Leksikalizovan primer takvog stabla je stablo α_1 na istoj slici.

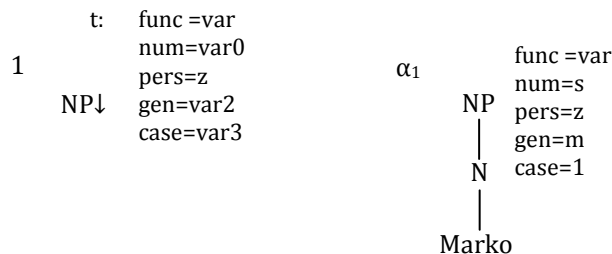


Slika 56 Zameničke enklitike u SrpTAG

2) Većina argumenata realizuje se kao neka od fraza – imenička, pridevska¹⁴², priloška, predložka ili brojna. Pojam fraze je u neposrednoj vezi sa pojmom sintagme, koju (Stanojčić & Popović, 1997, p. 251) definišu kao jedinicu koja se sastoji od glavne reči (centra sintagme) tj. reči koja predstavlja leksičko jezgro sintagme i jednog ili više zavisnih članova ili konstituenata. Prema centru sintagme određuje se i tip sintagme. Tako će u slučaju kada je glavna reč imenica, u pitanju biti imenička sintagma, kada je glavna reč pridev – pridevska, kada je to prilog – priloška sintagma, a kada su u pitanju brojevi ili brojne reči – partitivna ili paukalna sintagma. Pod predložkom frazom (PP) ovde podrazumevamo jedinicu koja se sastoji od predloga i imeničke jedinice (imeničke sintagme ili zamenice). U datoj gramatici srpskog jezika, ovakve jedinice se nazivaju predložko-padežnim konstrukcijama i ne smatraju se sintagmama.

Od svih pomenutih fraza, argumentsku vrednost imaju imenička, priloška, predložka i brojna, te ćemo njih predstaviti u nastavku teksta. Međutim, naglašavamo da u radu zapravo koristimo samo minimalne sintagme ili fraze, odnosno one koje se sastoje samo od centra sintagme. Izuzetak su partitivne i paukalne sintagme kod kojih ćemo uvek predstavljati i odgovarajuću imeničku dopunu, kao i predložke fraze ili predložko-padežne konstrukcije, gde ćemo uvek predstavljati predlog i odgovarajuću imeničku dopunu.

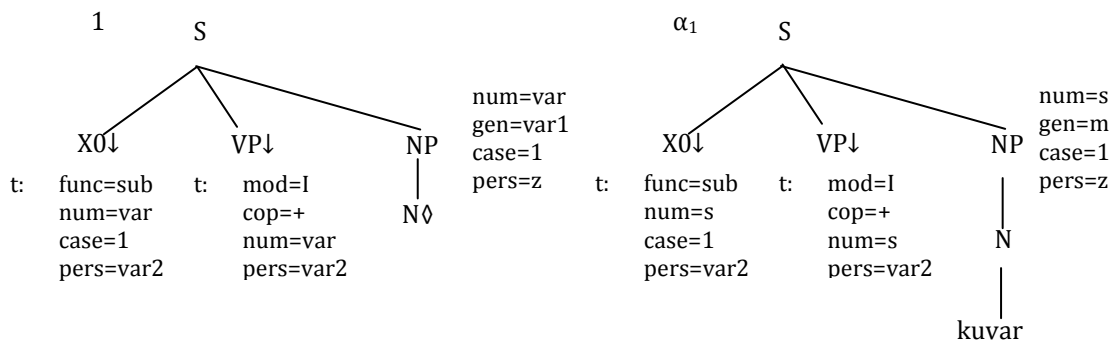
¹⁴² Pridevska fraza zapravo nema argumentsku vrednost, videti u nastavku teksta.



Slika 57 Minimalna imenička fraza

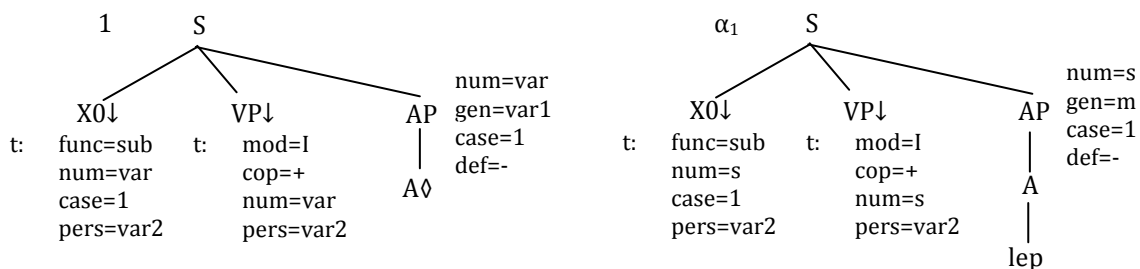
Slika 57 pod brojem 1 prikazuje izgled čvora unutar rečeničnog stabla na kome se unosi odgovarajuće imeničko stablo, poput onog prikazanog pod oznakom α_1 na istoj slici. U stablima fraza (Slika 56) i dalje u stablima u radu nećemo beležiti obeležja na međučvorovima (N), osim kada se ona razlikuju od onih na frazalnom čvoru. U ovakvim situacijama dakle podrazumevamo da su gornja i donja obeležja ista, kao i da su data obeležja terminalne i frazalne kategorije ista.

Imenička fraza se može javiti i kao sidro u rečenici – u kopulativnim konstrukcijama (Slika 58).



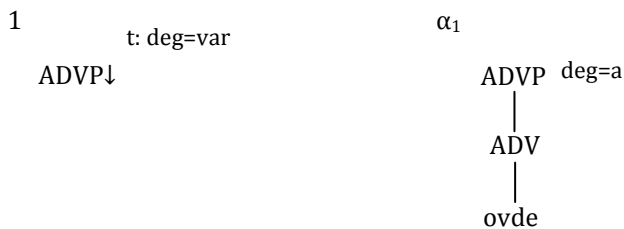
Slika 58 Imenička fraza kao predikativ u TAG

Pridevska fraza se ne javlja kao rečenični argument, odnosno nema svoju funkciju unutar SrpTAG. Ona može biti predikativ (Slika 59), koji, kao i upravo pomenutu imeničku frazu u istoj funkciji, ne smatramo argumentom glagola *jesam/biti*.



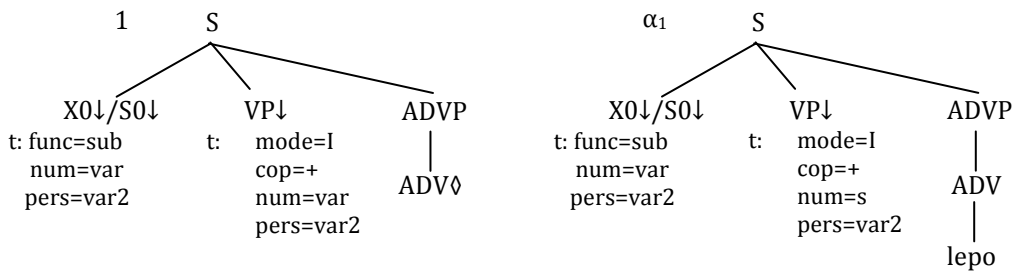
Slika 59 Pridevska fraza kao predikativ u SrpTAG

Prilog se kao glagolski argument u rečenično stablo takođe unosi zamenom. Čvor na kome se vrši zamena prikazan je pod brojem 1 (Slika 60). Stablo α₁ na istoj slici je tipično minimalno stablo priloga kao glagolskog argumenta.



Slika 60 Priloška fraza u SrpTAG

Kao i imeničke i zameničke fraze, i priloška fraza se može javiti kao predikativ (Slika 61). Kao i u prethodno navedenim primerima, ni ona u datom slučaju nema argumentsku vrednost.

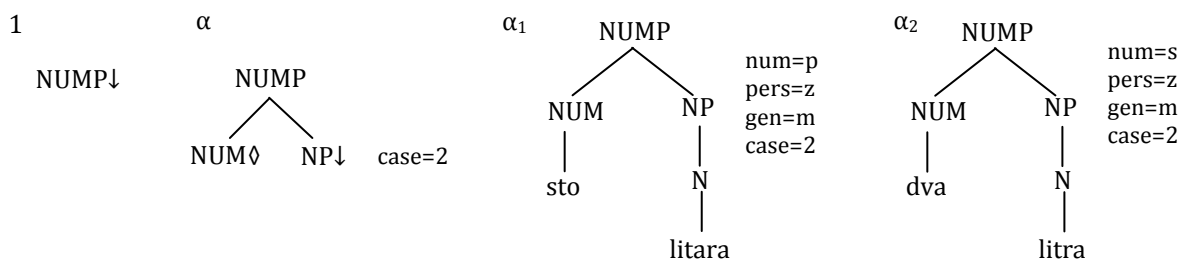


Slika 61 Priloška fraza kao predikativ u SrpTAG

(Stanojčić & Popović, 1997, pp. 260–261) sintagme koje sadrže broj ili brojnu i količinsku reč klasifikuju u partitivne i paukalne sintagme. Partitivne sintagme se sastoje od partitivne reči, koja predstavlja centar ovih sintagmi, i partitivne dopune, realizovane kao imenička jedinica u obliku partitivnog genitiva. Sama partitivna reč može biti imenica koja označava deo ili skup (*komad, veza...*), kardinalni broj, počevši od broja pet, redni broj ili brojna imenica (*pet, petoro, petorica*) i prilog za količinu (*koliko, mnogo...*). U elektronskom rečniku srpskog jezika postoje oznake za neke reči iz prve grupe, koje su označene kao MesApp (aproksimativna mera), zatim za reči iz druge grupe, označene sa NUM kao vrstom reči za prave brojeve (*dva*) ili Num kao dodatnom oznaka za imenice (*dvojica*), kao i za neke od reči iz treće grupe – one nose oznake Quant (kvantitet) i Amm (količina). Postojanje ovih oznaka će značajno olakšati prepoznavanje ovog tipa sintagmi u budućnosti.

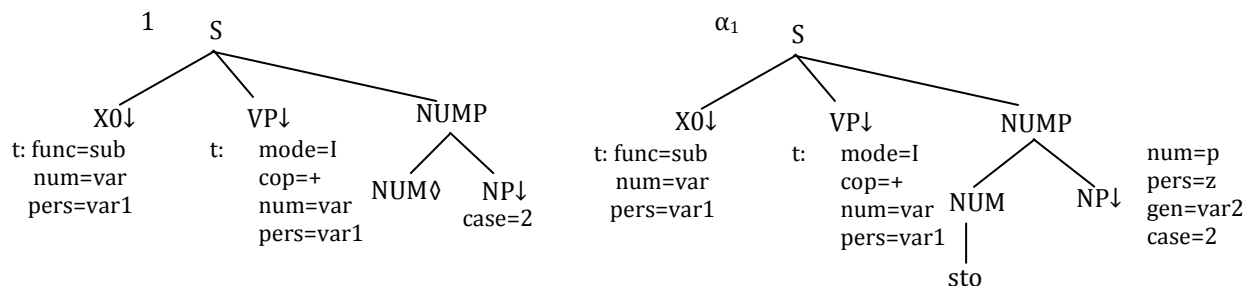
Paukalne sintagme su, prema autorima, drugačijeg tipa. Paukalne sintagme se sastoje od brojeva *dva, tri, četiri, oba* i *obadva* modifikatora imenice, koja je centar ovakve sintagme, a koja se nalazi u obliku paukala.

Za SrpTAG potrebno je da definišemo brojnu frazu koja se javlja kao dopuna glagolima kakvi su *koštati, iznositi* ili *težiti*, koja se suštinski sastoji samo od osnovnog broja i imenice u odgovarajućem obliku (genitiv singulara/plurala ili paukal). Definisaćemo tako samo dve vrste stabala, jednu za brojeve kakvi ulaze u partitivne sintagme (Slika 62, stablo α_1), a drugu za brojeve koji ulaze u paukalne sintagme (Slika 62, stablo α_2). U oba slučaja ćemo broj smatrati sidrom stabala, što se može videti na opštem stablu (Slika 62, stablo α). Pod brojem 1 prikazan je izgled rečeničnog čvora na koji se zamenom unose stabla poput α_1 i α_2 .



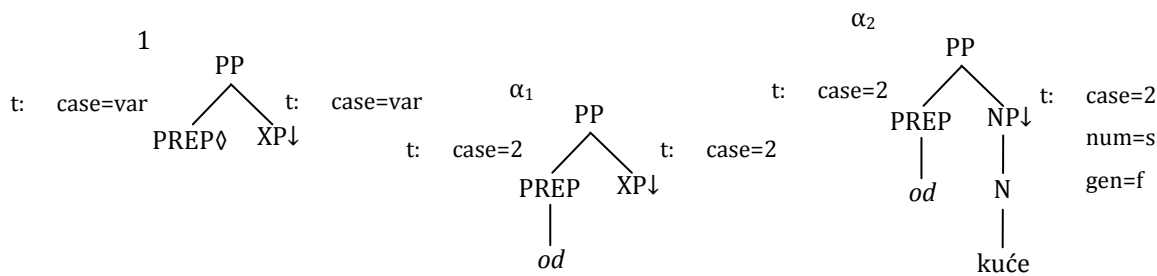
Slika 62 Brojne fraze u SrpTAG

Brojne fraze se mogu javiti i kao predikativi, pri čemu broj predstavlja sidro čitavog stabla (Slika 63). Stablo α_1 na istoj slici je stablo partitivne sintagme u funkciji predikativa.



Slika 63 Brojna fraza kao predikativ u SrpTAG

Što se predložkih fraza u SrpTAG tiče, u njihovoj internoj strukturi predlog predstavlja sidro. Slika 64 prikazuje izgled interne strukture predložke fraze – kao sheme (stablo 1), delimično leksikalizovane (stablo α_1) i potpuno leksikalizovane (stablo α_2).

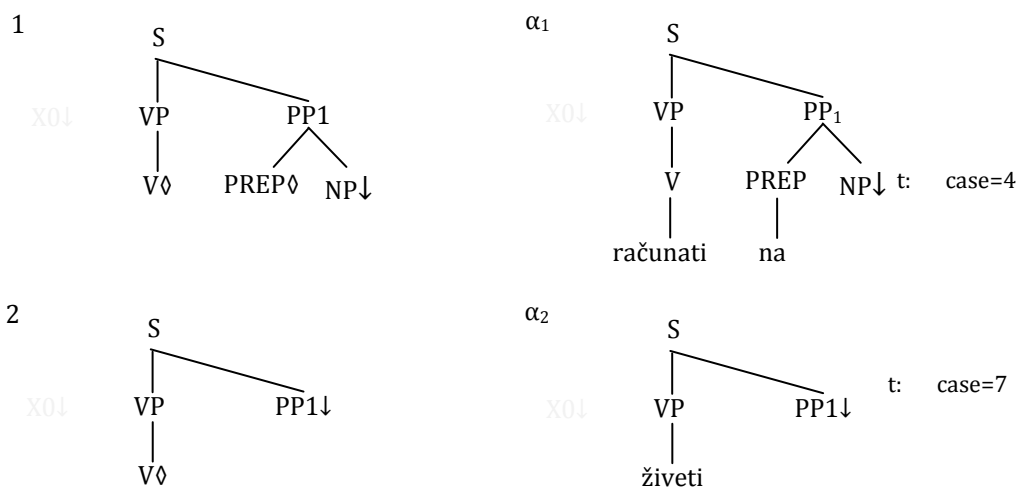


Slika 64 Predložka fraza u SrpTAG

Unutar rečeničnog stabla, predložke fraze se javljaju se u dva oblika. Predlog može predstavljati sidro rečenice, paralelno sa glagolom koji ga regira¹⁴³ (Slika 65, stabla 1 i α_1). Kod takvih predložko-padežnih konstrukcija u rečenično stablo se zamenom unosi samo imenička fraza koja se nalazi unutar predložko-padežne konstrukcije. S druge strane, cela predložko-padežna konstrukcija, tj. predložka fraza se može unositi u stablo

¹⁴³ Regirati = tražiti određenu jedinicu u konkretnom gramatičkom obliku kao dopunu. Ova pojava naziva se rekcija.

zamenom (Slika 65, stabla 2 i α_2). Na ovaj način tretiramo predložke fraze u kojima predlog nije regiran.



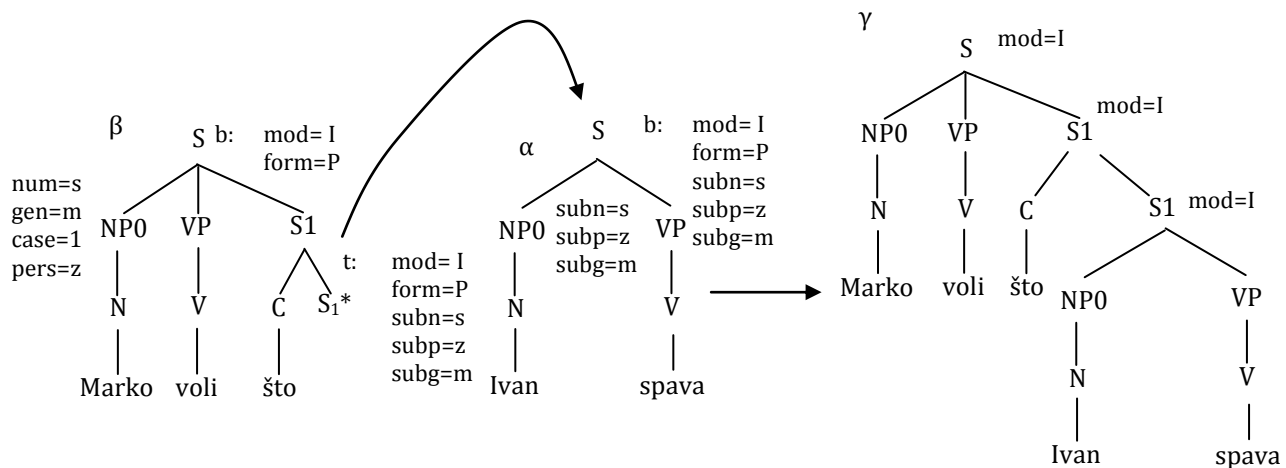
Slika 65 Odnos glagolske i predložke fraze u SrpTAG

Predložka fraza se može realizovati i kao priloški predikativ, međutim nije sasvim jasno šta ovde treba uzeti kao sidro konstrukcije. Iz definicije u poglavlju 3.3 zaključuje se da predlog sam po sebi ne može biti sidro čitave rečenične konstrukcije, a kako se dati predlog može javiti u kombinaciji sa bilo kojom imenicom, nema smisla beležiti ih zajedno u leksikonu. Ovu konstrukciju ćemo zato zasad ostaviti kao nerešenu.

3) Glagolski argumenti se mogu realizovati i kao neki od tipova zavisnih rečenica ili kao rečenica sa glagolom u infinitivu. (Stanojčić & Popović, 1997) navode devet tipova zavisnih rečenica, koje se, zajedno sa svojim funkcionalnim vrednostima mogu videti u tabeli u dodacima (Tabela 15). Sve zavisne rečenice imaju jednu od tri konstituentne vrednosti – imeničku (izrične i odnosne rečenice, rečenice s infinitivom), pridevsku (odnosne rečenice) ili prilošku (mesne, vremenske, uzročne, uslovne, dopusne, namerne, poredbene i posledične).

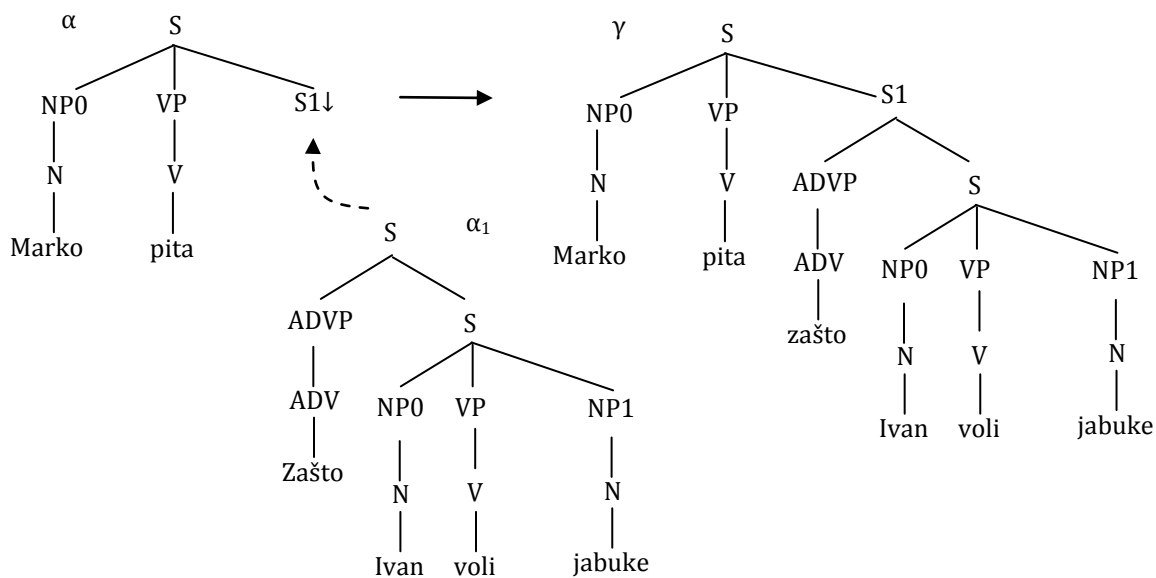
Kao što je već pomenuto u delu o elementarnoj rečenici u TAG, stabla zavisnih rečenica ćemo razlikovati prema tome da li je iz njih moguće vršiti izmeštanje ili ne. Svaka zavisna rečenica iz koje je izmeštanje moguće biće predstavljena kao inicijalno stablo, da bi se na nju dalje moglo pripojiti stablo glavne rečenice. U ovom slučaju,

subordinator čini deo glavne rečenice i unosi se u zavisnu rečenicu tek pripajanjem. Ovakav tretman imaju izrične deklarativne rečenice (Slika 66).



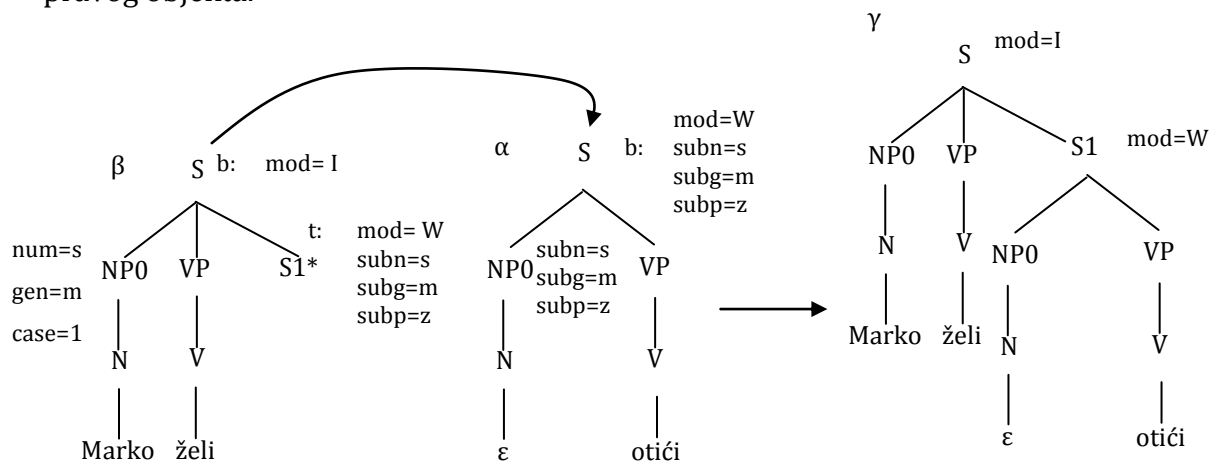
Slika 66 Izvođenje složene rečenice sa deklarativnom izričnom rečenicom

Upitne deklarativne rečenice, kako iz njih nije moguće izmeštanje, unose se zamenom u osnovnu rečenicu, pri čemu subordinator čini njihov sastavni deo (Slika 67).



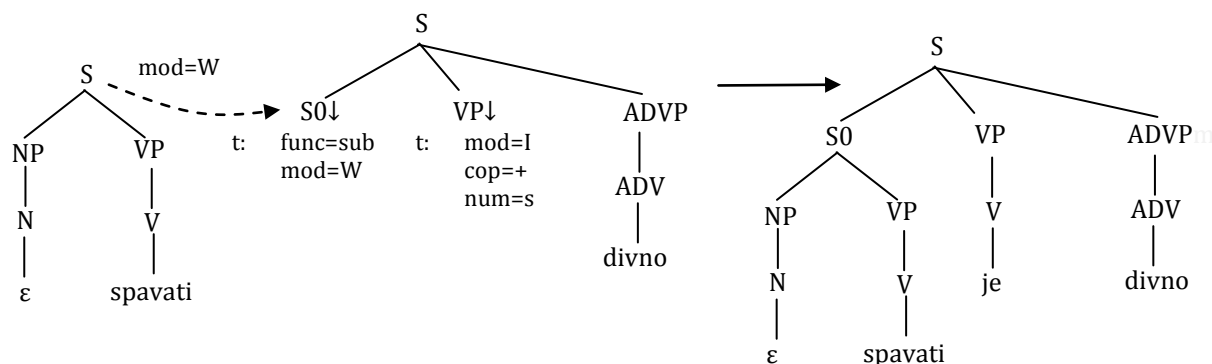
Slika 67 Izvođenje složene rečenice sa upitnom izričnom rečenicom

Rečenice s glagolom u infinitivu u funkciji dopune tretiramo kao inicijalne rečenice na koje se pripaja glavna rečenica. Slika 68 daje primer ovakve rečenice u funkciji pravog objekta.



Slika 68 Izvođenje složene rečenice s rečenicom u infinitivu u funkciji dopune

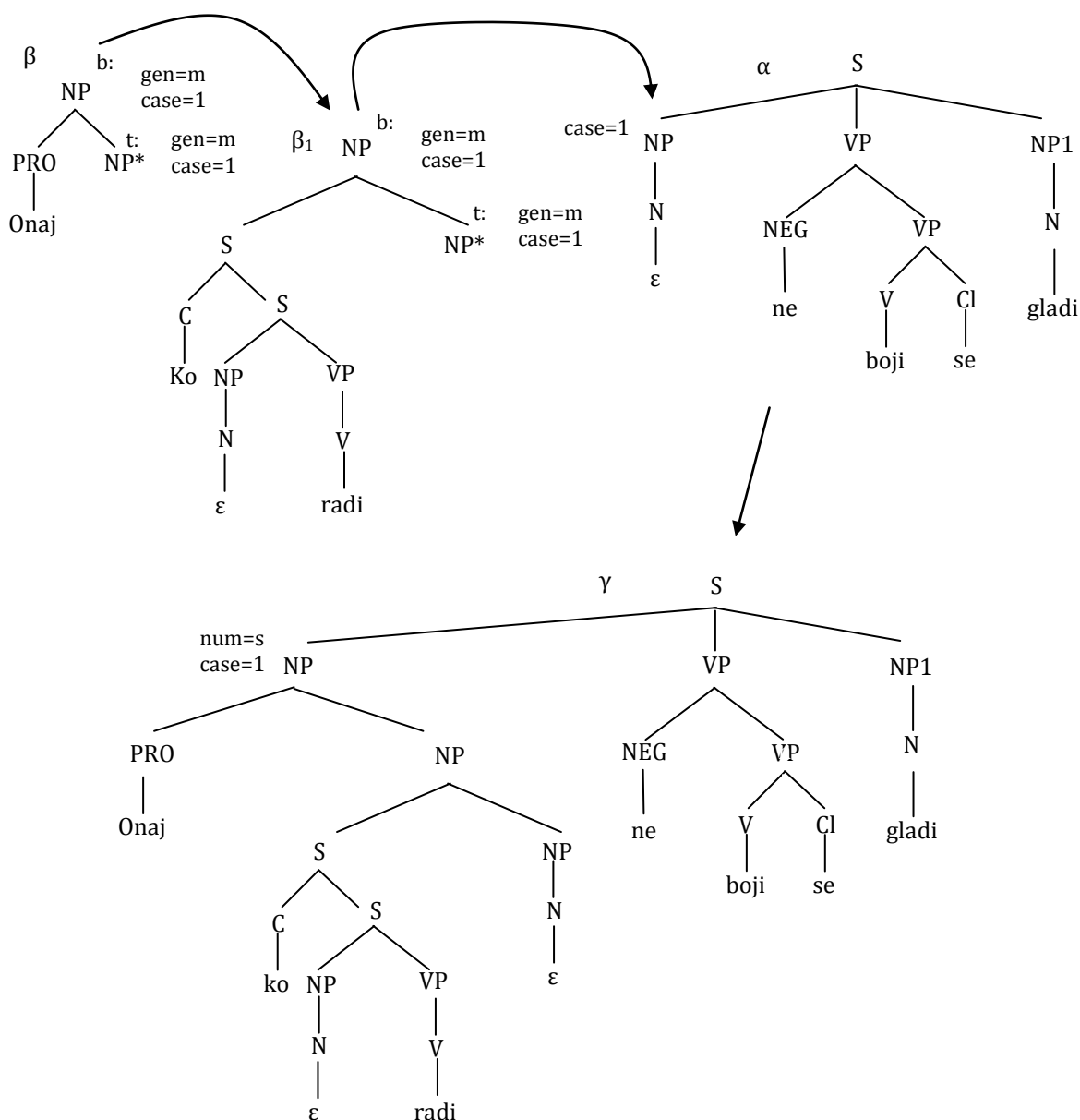
Rečenice s glagolom u infinitivu u funkciji subjekta unose se u glavno rečenično stablo procesom zamene (Slika 69).



Slika 69 Izvođenje rečenice sa subjektom u obliku infinitiva

(Stanojčić & Popović, 1997, pp. 305–307) navode da odnosne rečenice, imajući i imeničku konstituentnu vrednost, mogu da imaju funkcije glagolskih argumenata – subjekta i objekta. Ipak, u SrpTAG odnosne rečenice smatramo samo modifikatorima argumenta s određenom funkcijom. Slika 70 daje analizu rečenice *Onaj ko radi, ne boji se gladi*. Prema našoj analizi, ova rečenica je proizvod pripajanja pomoćnog stabla odnosne

rečenice *Ko radi* (stablo β_1) na nerealizovan subjekat u stablu glavne rečenice *Ne boji se gladi* (α). Zamenica *onaj* je kao koren pomoćnog stabla β najpre pripojena na stablo odnosne rečenice β_1 , čime je dobijen segment – *Onaj ko radi*, a zatim i na stablo α , dajući kao rezultat stablo izvedene rečenice γ . Ovako posmatrane, odnosne rečenice imaju isključivo modifikatorsku ulogu i kao takve mogu modifikovati bilo koji od imeničkih argumenata glagola. Nećemo se zato dalje baviti njima u ovom radu.



Slika 70 Modifikovanje subjekta odnosnom rečenicom

Od ostalih zavisnih rečenica, jedino namerne rečenice mogu imati argumentsku funkciju¹⁴⁴, međutim njih nećemo uključiti u ovaj rad.

3.5.2 Funkcije glagolskih argumenata

3.5.2.1 Gramatički subjekat

Gramatički subjekat se prema (Stanojčić & Popović, 1997, pp. 212–213) može realizovati na različite načine, ali njegova osnovna konstituentna vrednost jeste imenička jedinica u nominativu (49). On se dalje može realizovati i kao zavisna – izrična rečenica (50) ili nelična glagolska jedinica u obliku infinitiva (51) (Stanojčić & Popović, 1997, pp. 296, 306, 327)¹⁴⁵. Može biti i nerealizovan u površinskoj strukturi rečenice (52), ili potpuno nepostojeći u strukturi, kao u bezličnim rečenicama (53).

49. *Mačka spava. / Ovo je moja sestra.*

50. *Svima odgovara **da krenemo kući.***

51. ***Spavati** je divno.*

52. *Dolazim! / Idi!*

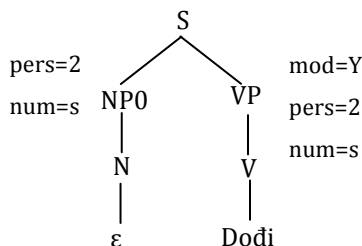
53. *Sevalo je.*

Takođe se navodi da glagol u predikatu kongruira (slaže se) sa subjektom u kategoriji broja, lica, i roda ukoliko glagolski oblik razlikuje rod. Treba napomenuti da se ova konstatacija odnosi samo na subjekat koji nije u obliku zavisne rečenice, realizovan kao glagolska jedinica u infinitivu, jer se u tom slučaju glagol nalazi u nekongruentnom obliku – 3. licu jednine.

¹⁴⁴ (Stanojčić & Popović, 1997) navode da i posledične i poredbene rečenice mogu imati funkciju dopuna. Međutim, pri pogledu na njihov oblik (Tabela 15) može se videti da ni jedan ni drugi tip rečenica ne predstavljaju argumente glagola, već dopune unutar odgovarajućih neglagoskih fraza ili sintagmi.

¹⁴⁵ (Stanojčić & Popović, 1997) navode da i odnosne rečenice mogu imati funkciju subjekta. Ipak, u prethodnom poglavlju smo mogli videti da smo prihvatili analizu po kojoj su odnosne rečenice modifikatori, te ih nećemo dalje navoditi pri definisanju jedinica koje mogu imati neku od imeničkih argumentskih funkcija. Za primere funkcija odnosnih rečenica svakako pogledati dodatke (Tabela 15).

Kako bezlični glagoli nemaju subjekat u svojoj argumentskoj strukturi, on se ne nalazi u bezličnim rečenicama. Kod ličnih glagola, s druge strane, čak i kada je izostavljen, subjekat je prisutan u argumentskoj strukturi glagola, u šta se možemo uveriti zahvaljujući kongruentnom obliku glagola. Sve dok subjekat postoji u argumentskoj strukturi glagola, on mora biti prisutan i u njegovom stablu, čak i kada je nerealizovan. Nerealizovan subjekat, kao i bilo koji drugi nerealizovani argument, obeležava se simbolom ϵ (prazna reč). Kako je već napomenuto u poglavlju o leksičkim pravilima (2.4.4), struktura s nerealizovanim argumentom smatra se alternativnom u odnosu na strukturu u kojoj je on realizovan. Prateći princip 2 dobre formiranosti elementarnih stabala¹⁴⁶, smatramo da i rečenice s glagolom u imperativu imaju subjekat (Slika 71).



Slika 71 Subjekt u stablima s imperativom

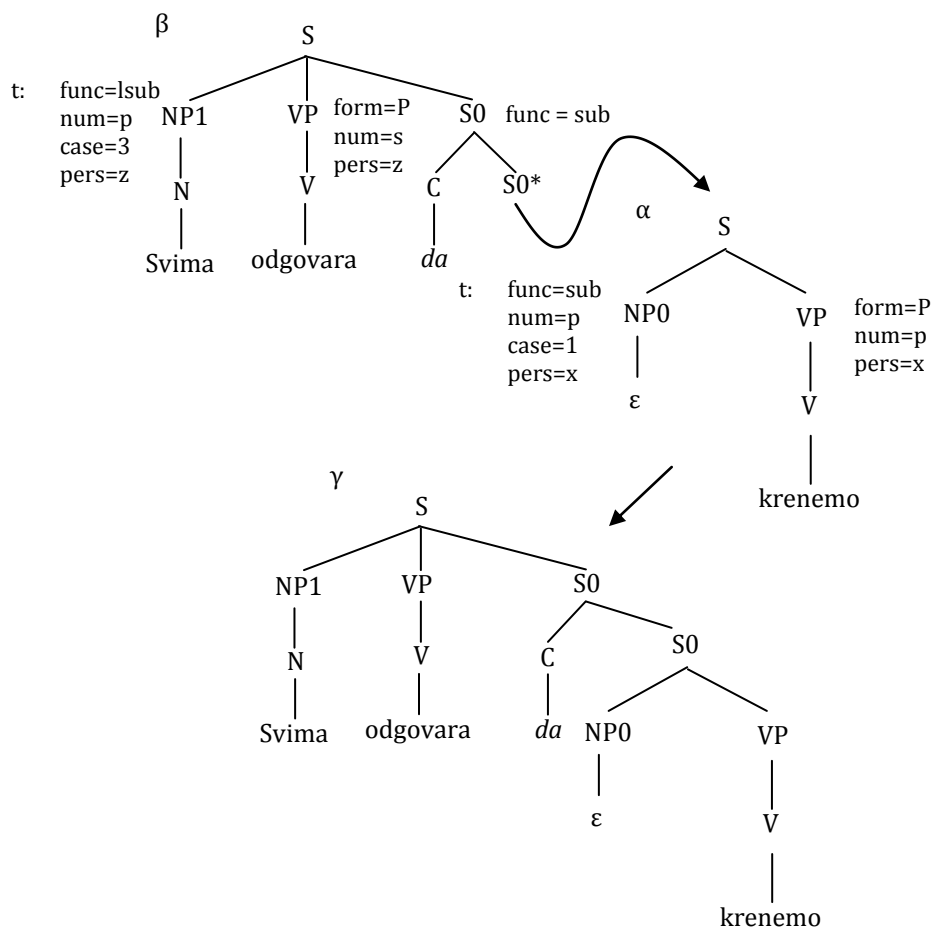
Kao što smo već imali prilike da vidimo u poglavlju 3.5.1 subjekat u formi glagola u infinitivu ima strukturu inicijalne rečenice, koja se u glavnu rečenicu unosi zamenom (Slika 69). Subjekt u formi izrične rečenice se, kako je moguće vršiti izmeštanja iz njega (primeri 54 i 55), predstavlja u formi inicijalnog stabla na koje se pripaja ostatak rečenice (Slika 72).

54. *Svima odgovara da [krenemo kući].*

55. *[Gde] svima odgovara da [krenemo]?*

Subjekt u stablima SrpTAG tipično nosi indeks 0.

¹⁴⁶ Upućujemo nazad na poglavlje 3.3



Slika 72 Izvođenje stabla sa izričnim subjektom (stablo β)

3.5.2.2 Logički subjekat

(Stanojčić & Popović, 1997, pp. 242) navode da je logički subjekat imenička jedinica u akuzativu (56), dativu (57) ili genitivu (58)¹⁴⁷. Logički subjekat u akuzativu i dativu javlja se u subjekatsko-predikatskim rečenicama, paralelno sa gramatičkim subjektom. U ovom tipu rečenica glagol uvek kongruira sa gramatičkim subjektom. Logički subjekat u genitivu javlja se u bezličnim rečenicama, te će glagol u njima stajati u bezličnom obliku, odnosno 3. licu jednine (srednjeg roda).

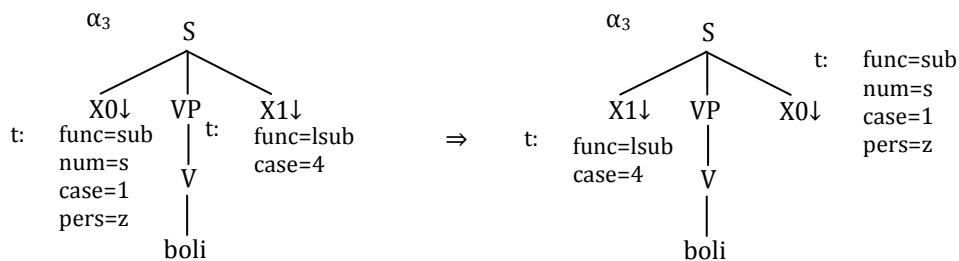
56. *Marka boli ruka.*

¹⁴⁷ Prilikom definisanja konstituentskih vrednosti zavisnih rečenica, (Stanojčić & Popović, 1997, p. 306) navode i da odnosne rečenice mogu vršiti funkciju logičkog subjekta, ali videti fusnotu 145.

57. *Marku se sviđa Marija.*

58. *Sutra nema časova.*

Uzimajući u obzir da je kanonski red reči u SrpTAG – SVO, u ovde navedenim stablima za logički subjekat (Slika 73), gramatički subjekat se ne nalazi na poziciji koju očekujemo od njega (vrednost indeksa – 0). Na ovom mestu možemo pretpostaviti da ovakve strukture predstavljaju alternative osnovnih struktura – u kojima se gramatički subjekat nalazi sa leve strane glagola, a logički sa desne strane, o čemu će svedočiti i indeksi oba argumenta.



Slika 73 Leksičko pravilo za promenu reda reči - logički i gramatički subjekat

3.5.2.3 Dopune

(Stanojčić & Popović, 1997) razlikuju nekoliko tipova glagolskih dopuna. Autori prave razliku između takozvanih rekcijskih dopuna (pravi i nepravi objekat), dopunskog predikativa i priloških dopuna. Kao i subjekat, i dopune se mogu izostaviti iz rečeničnog izraza, iako će uvek biti prisutne u argumentskoj strukturi glagola, pa samim tim i u njegovom elementarnom stablu. Pogledajmo svaku od ovih grupa detaljnije.

a) Rekcijske dopune (Stanojčić & Popović, 1997, p. 223) definišu kao dopune uslovljene rekcijom glagola, deleći ih pritom na pravi i nepravi objekat.

- **Pravi objekat** se može realizovati kao imenička jedinica u akuzativu bez predloga (59), kao imenica ili imenička sintagma u genitivu bez predloga (partitivni genitiv – 60, slovenski genitiv – 61), kao izrična rečenica (62) ili kao glagolska jedinica u obliku infinitiva (63) (Stanojčić & Popović, 1997, pp. 295, 306, 327). Može biti i nerealizovan u rečeničnoj strukturi (64).

59. *Ivan čita **novine**.*
 60. *Donesi **mleka**.*
 61. *Nije rekao ni **reči**.*
 62. *Ivan je čuo **da je Marko otišao na utakmicu**.*
 63. *Marko želi **krenuti kući**.*
 64. *Marko čita.*

Pravi objekat je tipična dopuna prelaznih glagola, o kojima je bilo reči u delu o predikatu (poglavlje 3.4).

- **Nepрави objekat** može biti realizovan kao imenička jedinica u dativu (65), genitivu (66) ili instrumentalu bez predloga (67), kao predložko-padežna konstrukcija (68) i izrična rečenica (69) (Stanojčić & Popović, 1997, pp. 298, 306).

65. *Marko se raduje **poklonu**.*
 66. *Marko se boji **mraka**.*
 67. *Bavim se **sportom**.*
 68. *Marko liči **na oca**.*
 69. *Ivan je razmišljao (**o tome**) **da se upiše na prava**.*

b) Priloške dopune su jedinice s nekim od priloških značenja (značenje mesta, vremena, načina, mere, namene, itd.) kojima se dopunjavaju glagoli odgovarajućeg značenja (Stanojčić & Popović, 1997, p. 230). Realizuju se kao prilog ili priloška sintagma (70), imenička jedinica u zavisnom padežu s priloškim značenjem (71), brojna (partitivna ili paukalna) sintagma¹⁴⁸ (72), kao predložko-padežna konstrukcija s priloškim značenjem (73), kao zavisna (namerna) rečenica s glagolom u obliku infinitiva (74), odnosno glagolom u ličnom glagolskom obliku (75).

70. *Nikola se ponašao **čudno**. [dopuna za način]*

¹⁴⁸ U SrpTAG i paukalnu i partitivnu sintagmu tretiramo istovetno, pogledati nazad poglavlje 3.5.1

71. *Jasna ide **kući**.* [dopuna za pravac kretanja]
 72. *Hleb košta **50 dinara**.* [partitivna dopuna]
 73. *Knjiga se nalazi **na stolu**.* [dopuna za mesto]
 74. *Otišao je **večerati**.* [dopuna za cilj]
 75. *Marko je **da večera**.* [dopuna za cilj]

c) Dopunski predikativ (Stanojčić & Popović, 1997, p. 227-229) definišu kao imeničku ili pridevsku jedinicu koja može biti dopuna prelaznog ili neprelaznog glagola koga, zbog svoje sličnosti sa kopulativnim glagolom, nazivaju nepravilni kopulativni (semikopulativni) glagol. Svoj sadržaj preko semikopulativnog glagola pripisuje subjektu (76, 77, 78) u slučaju kada je glagol neprelazan, odnosno objektu (79, 80, 81) kada je glagol prelazan. Realizuje se kao imenička jedinica u nominativu (76) ili instrumentalu bez predloga (78, 79) ili kao pridevska jedinica ili redni broj u nominativu (77) ili instrumentalu bez predloga (80). Može se realizovati i kao predložko-padežna konstrukcija *za*+akuzativ (78, 81). Autori još naglašavaju da ukoliko pridev razlikuje vid, on u nominativu i akuzativu jednakom nominativu stoji u neodređenom vidu.

76. *Marko je postao **predsednik**.*
 77. *Vreme je postalo **hladno**.*
 78. *Marko se proglasio **pobednikom / za pobednika**.*
 79. *Marka su proglasili **pobednikom**.*
 80. *Ljudi ga smatraju **dobrim**.*
 81. *Marija je izrabrala Tanju **za deverušu**.*

Zbog nejasnoće oko toga kako tretirati sam glagol u konstrukcijama s dopunskim predikativom, koji je istovremeno i kopulativan i glagol koji traži dopunu, kao i dopunskog dela predikata, koji ima karakteristike i dopune i predikativa, odlučili smo da u ovoj fazi izrade rada isključimo ovaj tip glagola i argumenata iz analize.

3.6 Rečenični modeli

Sada kada smo napravili pregled glagola i svih osnovnih glagolskih argumenata možemo da definišemo i rečenične modele, odnosno porodice stabala za SrpTAG. Kao polaznišnu tačku uzimamo rečenične modele definisane u (Stanojčić & Popović, 1997). Tabela 10 daje prikaz svih subjekatsko-predikatskih rečeničnih modela, sem modela sa semikopulativnim glagolima¹⁴⁹, dok Tabela 11 daje prikaz svih bezličnih (besubjekatskih) rečeničnih modela. U terminima TAG gramatike, o ovim modelima ćemo govoriti kao o porodicama stabala, pod kojima su objedinjene sve alternative datog rečeničnog modela, izvedene pomoću leksičkih pravila¹⁵⁰.

Svi primeri navedeni u tabelama preuzeti su u neizmenjenom obliku iz (Stanojčić & Popović, 1997, pp. 206–329). U našem teorijskom modelu, a u skladu sa tipologijom rečenica u datoj gramatici bavićemo se samo predikatskim (subjekatsko-predikatskim i bezličnim) rečenicama, odnosno rečenicama koje u svojoj strukturi imaju predikat. Specijalne rečenice, odnosno rečenice koje ne sadrže predikat već kao svoje leksičko jezgro imaju neku neglagolsku jedinicu (*Auto!*, *Evo autobusa!*) ostaju van okvira ovog rada.

Tabela 10 Rečenično modeli subjekatsko-predikatskih rečenica

Redni broj	Model	Glagoli
1	Rečenica _{glagol bez dopune} = SUBJEKAT PREDIKAT	Zoran spava. Dolazi autobus! Neko je došao. Ti se šališ.
2	Rečenica _{fakultativno prelazni glagol} = SUBJEKAT PREDIKAT (PRAVI OBJEKAT)	Jasna čita (knjigu). Zoran piše (pismo).
3	Rečenica _{prelazni glagol} = SUBJEKAT PREDIKAT PRAVI OBJEKAT	Jasna je kupila knjigu.
4	Rečenica _{neprelazni rekcijski glagol} = SUBJEKAT PREDIKAT NEPRAVI OBJEKAT	Otac pomaže sinu. Nikola je klimnuo glavom. Tanja se obradovala poklonu. Ivan se stidi brata. Nikola se zaljubio u Tanju. U koga se zaljubio Nikola?

¹⁴⁹ (Stanojčić & Popović, 1997) navode dva takva modela: Rečenica_{neprelazni nepravni kopulativni glagol} = SUBJEKAT PREDIKAT DOPUNSKI PREDIKATIV i Rečenica_{prelazni nepravni kopulativni glagol} = SUBJEKAT PREDIKAT (PRAVI) OBJEKAT DOPUNSKI PREDIKATIV.

¹⁵⁰ Rečenični modeli koji će biti navedeni u nastavku predstavljaju najviši nivo apstrakcije u metagramatici. O tome kako se ovi modeli (porodice) formiraju od alternativa povezanih leksičkim pravilima biće više reči u narednom delu (poglavlje 4.4).

		Nikola se interesuje za slikarstvo. Tanja se navikla na novu sredinu.
5	Rečenica _{neprelazni glagol sa priloškom dopunom} = SUBJEKAT PREDIKAT PRILOŠKA DOPUNA	Ja stanujem ovde. Knjiga se nalazi na stolu. Nikola se ponašao čudno. Račun iznosi 10.000 dinara. Ova sprava služi za košenje trave. Ovaj džak teži 10 kilograma. Torba leži na klupi. Koliko teži ovaj džak? Ivan živi u Beogradu. Nikola se drži pogrbljeno.
6	Rečenica _{neprelazni glagol sa logičkim subjektom} = LOGIČKI SUBJEKAT PREDIKAT (GRAMATIČKI) SUBJEKAT	Marku se igrala košarka. Marku je zadrhtala ruka. Marku sevaju oči. Jasnu je boleo zub. Marka svrbi nos. Zastao mu je dah.
7	Rečenica _{glagol davanja ili saopštavanja} = SUBJEKAT PREDIKAT NEPRAVI OBJEKAT PRAVI OBJEKAT	Tanja je vratila Zoranu knjigu.
8	Rečenica _{prelazni glagol sa priloškom dopunom} = SUBJEKAT PREDIKAT PRAVI OBJEKAT PRILOŠKA DOPUNA	Ivan je ubacio pismo u sanduče. Marko je položio dokumenta na pisači sto. Domaćin je uveo goste u salon. Marko je smestio stvari u torbu. Ko je ostavio knjige na klupi? Jasna je stavila knjigu na policu. Zoran je ubacio loptu u koš.
9	Rečenica _{imenski predikat¹⁵¹} = SUBJEKAT KOPULA IMENSKI PREDIKATIV	Jasna je bila bolesna. Jasna je pametna. Beograd je glavni grad Jugoslavije. Marko je ljut na Tanju. Ovo je moja sestra. Mi smo veoma zauzeti. Zima je bila blaga. Ovaj restoran je vrlo skup. Tanja je najbolja učenica.
10	Rečenica _{priloški predikat} = SUBJEKAT KOPULA PRILOŠKI PREDIKATIV	Stanica je daleko. Ivan je u školi. Ivanov brat je bio na zimovanju.

¹⁵¹ U dijagramskoj predstavi rečenica sa npr. imenskim predikativnom „bolestan“, ovaj model se naziva: **Rečenica**_{imenski predikat „bolestan“}. Isto važi i za priloške predikative: njihovim centrom se smatra priloška jedinica pa će rečenica s priloškim predikativom „daleko“ biti imenovana kao **Rečenica**_{priloški predikat „daleko“}.

		Predavanje je sutra. Gde je biblioteka? Biblioteka je vrlo blizu. Jasna je bila u šetnji. Kada je koncert? Mi smo bili u bioskopu. Oni su u teškoj situaciji. Kako je Nikola? Tanja je kod kuće Nikola je dobro. Mi smo bili na predavanju.
--	--	---

Tabela 11 Rečenični modeli bezličnih rečenica

Redni broj	Model	Primeri
1	Bezlična rečenica _{bezlični glagol} = PREDIKAT (bezlični glagol u 3. licu jednine (srednjeg roda) ličnog (finitnog) oblika)	(U daljini) je sevalo. (Juče) je zahladnelo. Smrklo se.
2	Bezlična rečenica _{priloški predikat} = GLAGOLSKI DEO PREDIKATA (bezlični glagol u 3. licu jednine (srednjeg roda) ličnog (finitnog) oblika) PRILOŠKI DEO PREDIKATA	(Danas) je toplo. (Napolju) je hladno. Sparno je.
3	Bezlična rečenica _{bezlični glagol} = LOGIČKI SUBJEKAT PREDIKAT (bezlični glagol u 3. licu jednine (srednjeg roda) ličnog (finitnog) oblika)	Marku se spavalo. Marku je pozlilo. (Danas) ima/nema nastave. (Juče) (ni)je bilo nastave. Sutra (ne)će biti nastave. (Još) ima vremena. (Više) nema vremena.
4	Bezlična rečenica _{priloški predikat} = LOGIČKI SUBJEKAT GLAGOLSKI DEO PREDIKATA (bezlični glagol u 3. licu jednine (srednjeg roda) ličnog (finitnog) oblika) PRILOŠKI DEO PREDIKATA	Marku je bolje. Marku je dosadno. (Sad) mi je lakše. (Ovde) nam je prijatno.

Navedeni rečenični modeli definisani su u datoj gramatici kao sintaksički sistem koji reguliše sve tipove rečeničnih konstrukcija. Svaki takav model sastoji se od informacije o tipu glagola (prelazni, neprelazni, kopulativni), odnosno predikata (imenski, priloški) i konstituentima tj. argumentima, kao jedinicama uslovljenim valencom glagola. Jedinice koje nisu uslovljene valencom glagola, ili odredbe, ne javljaju se u ovim modelima, te su oni u tom smislu minimalni.

Raspored glagola i njegovih argumenata unutar modela određen je njihovom ulogom u formiranju rečenične perspektive (Stanojčić & Popović, 1997, p. 352), te su oni

raspoređeni tako da se informacija koja se datom rečenicom prenosi prima postepeno, ali i tako da konstituenti koji su blisko vezani stoje zajedno. Tako će tipičan raspored biti subjekat, zatim predikat i onda dopune, od kojih se ona tešnje povezana s glagolom javlja direktno uz njega, a sledeća se naslanja na datu dopunu.

Ono što se takođe može primetiti kod navedenih modela jeste da su definisani preko funkcija rečeničnih argumenata. Nasuprot ovakvom definisanju modela stoji njihovo definisanje preko kategorija, karakteristično za transformaciono-generativnu gramatiku. I u TAG tradiciji rečenični modeli se definišu preko kategorija, dok se funkcije navode unutar obeležja. Razlog za to leži pre svega u nameni ovog opisa – dati modeli se koriste kao ulaz za parsere i druge vrste programa za analizu teksta te je potrebno da detaljno opisuju konkretne realizacije rečeničnih segmenata.

Pri definisanju rečeničnih modela, tj. porodica stabala za SrpTAG pratićemo konvenciju postavljenu u (Abeillé, 2002, pp. 170–175), uz određene izmene. Prema datoj konvenciji, univerzalni obrazac za beleženje rečeničnih modela jeste arg_0Varg_1 za modele sa glagolom kao svojim sidrom. Kao što se kategorije u samim stablima obeležavaju rednim brojem, tako se obeležavaju i u modelima, označavajući kanonsku poziciju tog argumenta u stablu tj. rečeničnoj strukturi. Budući da se na ovaj način za srpski jezik ne mogu razlikovati pojedine dopune koje se mogu naći na istoj poziciji (npr. pravi objekat prelaznih glagola i nepravi objekat neprelaznih), odlučili smo da uz kategorije iza indeksa argumenta beležimo i funkcije – arg_0Varg_{1dir} . Oznake funkcija koje koristimo su *dir* (pravi objekat), *ind* (nepravi objekat), *pril* (priloška dopuna), *lsub* (logički subjekat).

U sastav modela ulaze i konkretne lekseme, onda kada ih glagoli strogo regiraju. Zapravo, svaki put kada u stablu imamo sidro koje je paralelno s glagolom, leksema koja je nosilac tog sidra direktno će ulaziti u rečenični model.

arg_0Varg_1 je obrazac modela koje ćemo mahom videti za srpski jezik, ali nije jedini. Naime, u modelima u kojima rečenično sidro nije glagol već predikativ, velikim slovom u modelu biće obeleženo primarno sidro tog modela – imenička (arg_0vN)¹⁵², pridevska (arg_0vA) ili priloška jedinica (arg_0vAd).

¹⁵² Za oznake kategorija koje koristimo u modelima videti tabelu u nastavku teksta (Tabela 12).

Kao što u modele u našoj referentnoj gramatici srpskog jezika ne ulaze odredbe, ali ni jedinice kao što su pomoćni, modalni glagoli ili negacija, tako oni neće ulaziti ni u modele, tj. porodice SrpTAG gramatike. Subjekt se u oba modela beleži kao uvek prisutan, i pored činjenice da može biti izostavljen.

Kategorije u upotrebi u rečeničnim modelima, odnosno nazivima porodica stabala, nešto su drugačije od onih koje koristimo u stablima. Tabela 12 prikazuje sve oznake koje se javljaju u modelima. Imenice i zamenice grupišemo u jednu kategoriju, obeleženu kao *n* ili *N*, kako njihovo razdvajanje unutar modela koji su minimalni (ne sadrže modifikatore) nema značaja. Predložka fraza, odnosno predložko-padežna konstrukcija predstavlja se isključivo kao predlog i imenica (pn_i), da bi bilo moguće oznaku za predlog zameniti konkretnim regiranim predlogom (npr. **od** n_i kod glagola *zavisiti*).

Tabela 12 Simboli u upotrebi u nazivima porodica stabala u SrpTAG

SIMBOL	OBJAŠNENJE
n	imenička jedinica kao argument
N	imenička jedinica kao sidro
v	glagol koji se unosi zamenom
V	glagol kao sidro
A	pridevska jedinica kao sidro
ad	priloška jedinica kao argument
Ad	priloška jedinica kao sidro
num	brojna reč kao argument
NUM	brojna reč kao sidro
s	izrična deklarativna rečenica
sup	izrična upitna rečenica
sinf	rečenični konsituent u infinitivu
sad	namerna rečenica s komplementizatorom
c	komplementizator
da, što	konkretan komplementizator kao sidro
Cl	refleksivna rečca <i>se</i> kao drugo sidro
p	predlog
u, za, iz...	konkretan predlog kao sidro
dir, ind, lsub, pril	rečenična funkcija

Pre nego što zasebno pogledamo svaki od modela navedenih u gramatici (Stanojčić & Popović, 1997) i damo njegov ekvivalent definisan preko kategorija za potrebe SrpTAG, predstavimo detaljno leksička pravila na primeru srpskog jezika.

3.6.1 Leksička pravila

Kao što je već naglašeno u poglavlju 2.4.4, TAG ne propisuje operaciju transformacije. Ipak, potrebno je na neki način međusobno povezati stabla koja predstavljaju alternacije iste argumentske strukture glagola, odnosno rečeničnog modela. Ovo se u TAG čini pomoću leksičkih pravila. Sva stabla koja su međusobno povezana leksičkim pravilima čine jednu porodicu stabala. U tom smislu, rečenični modeli koji su definisani u prethodnom poglavlju predstavljaju formula kanonskog predstavnika porodice. U ovom delu, definisaćemo pravila koja povezuju te kanonske predstavnike sa njihovim strukturnim alternativama.

Obradićemo sva pravila redistribucije, ili pravila koja menjaju strukturu stabala, sem pravila za relativizaciju¹⁵³. Tako će u sledećem poglavlju biti reči o leksičkim pravilima za pasiv – participski i reflektivni, obezličavanje i reflektivizaciju. Zasebno ćemo obraditi i pitanja, kao primere pravila izmeštanja. Od pravila realizacije, ili pravila koja menjaju mesto i realizaciju argumenata, obradićemo pravilo za izostavljanje argumenata, kliticizaciju i ukratko ćemo prokomentarisati pravila koja menjaju red reči.

3.6.1.1 Pravila redistribucije

Pasiv

Pod pasivom se u srpskom jeziku podrazumevaju dve vrste konstrukcija – konstrukcija pod nazivom participski pasiv, i konstrukcija pod nazivom reflektivni pasiv. Prema (Stanojčić & Popović, 1997, p. 115) u konstrukciju s participskim pasivom ulaze samo prelazni glagoli. Ipak postoji niz prelaznih glagola koji ne mogu ući u ovu konstrukciju.

¹⁵³ Kako se relativizacijom ne dobijaju glagolski argumenti već modifikatori, ovo leksičko pravilo neće biti od značaja za nas u ovoj fazi oblikovanja porodica stabala. Leksičko pravilo relativizacije se ipak može se videti u poglavlju 3.3 (Slika 38), gde se može videti i jedan primer ovako izvedene konstrukcije (Slika 39). U poglavlju 3.5.1 može se videti primer izvođenja stabla s relativnom konstrukcijom (Slika 70).

(Moskvljević Popović, 2007, p. 59) kao takve navodi glagole *hteti, jesti, mrzeti, običi, piti, rasplamtati, razgoreti, razumeti, sresti, znati...* (Mrazović, 2009, p. 110) ovoj grupi dodaje i glagole posedovanja: *imati, nemati, posedovati*, kao i one koje naziva glagolima duhovnog posedovanja: *saznati, poznavati*. S druge strane, postoji i određena grupa neprelaznih glagola koja može ući u konstrukciju s participskim pasivom. (Moskvljević Popović, 2007, p. 61) tako navodi glagole *atakovati, motriti, napadati, nasrtati, navaljivati*. Iz ovog razloga je među obeležja glagola u leksikonu uvršteno obeležje pasiv s bulovskom vrednošću, kojim se označava da li glagol može ući u ovakvu konstrukciju.

Drugi tip pasivne konstrukcije, takozvani reflektivni pasiv, nije restriktivan i primenjuje se na sve prelazne glagole. Iz tog razloga nećemo uvoditi posebnu oznaku da označimo da je moguće primeniti ovo leksičko pravilo.

Pogledajmo sada obe konstrukcije s malo više detalja, kao i njihova leksička pravila.

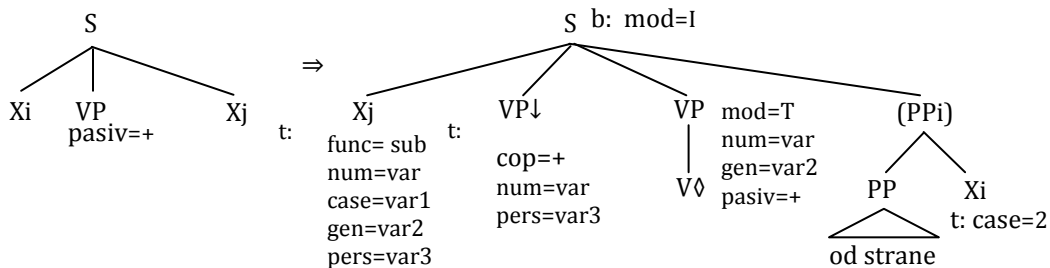
1. Participski pasiv

(Stanojčić & Popović, 1997, p. 245) pasivne rečenice, ili rečenice s pasivnom konstrukcijom svrstavaju u jedan od posebnih tipova predikatskih rečenica. To su rečenice sa ličnom pasivnom konstrukcijom, za razliku od lične ali aktivne konstrukcije subjekatsko-predikatskih rečenica. Razlika se ogleda u obliku glagola, koji je u ovom posebnom tipu rečenica u pasivu, kao i u rasporedu argumenata – subjektom je umesto agensa iskazan pacijens, dok se agens može potpuno izostaviti iz konstrukcije, ali ako je prisutan, javlja se kao agentivna odredba u obliku *od (strane)+genitiv*. U TAG aktivna i pasivna konstrukcija povezane su leksičkim pravilom, koje je prikazano na sledećoj slici (Slika 74).

Kao što je obrazloženo u poglavlju 3.4.4, pasivni glagol iz naše referentne gramatike u TAG razlaže se na dva odvojena čvora. Glagol *jesam/bit* se u pasivnoj konstrukciji ne tretira kao pomoćni glagol, već kao kopula, te nosi obežje *cop=+*. Isto tako, u strukturu se unosi zamenom, tj. ne predstavlja sidro pasivne konstrukcije. Samo po sebi, pasivno stablo je inicijalno i njegovo sidro je glagol u obliku trpnog glagolskog prideva. Kongruencija se uspostavlja između pacijensa u funkciji subjekta i kopule u kategorijama broja i lica, kao i glagola u obliku trpnog glagolskog prideva u kategorijama roda i broja.

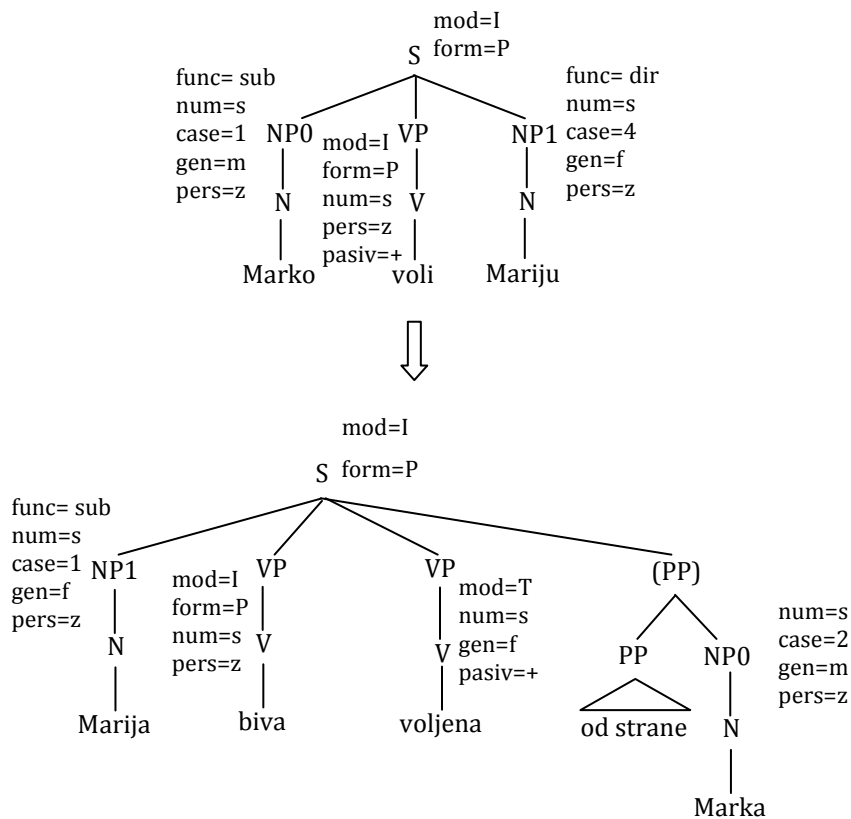
Ukoliko glagol u svojoj strukturi ima još argumenata, njihova struktura ostaje neizmenjena.

Supskripte *i* i *j* koristimo da označimo vezu između argumenata s leve i desne strane pravila.



Slika 74 Leksičko pravilo za participijski pasiv

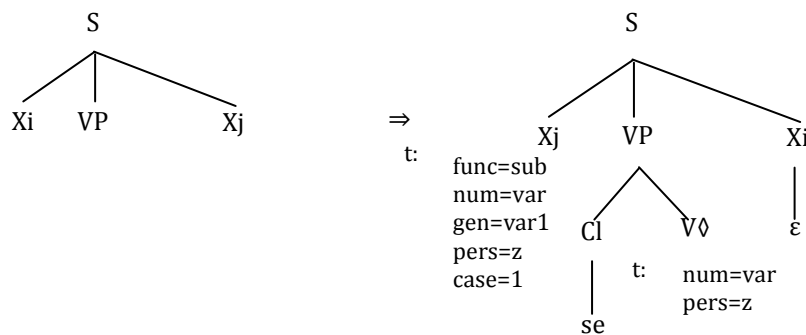
Ovim leksičkim pravilom tako su povezana dva stabla na sledećoj slici (Slika 75).



Slika 75 Primer leksičkog pravila za pasiv na glagolu *voleti*

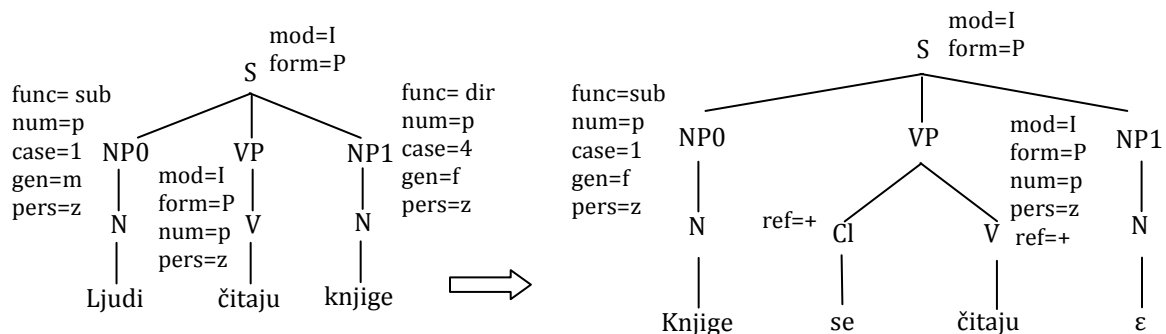
2. Refleksivni pasiv

Kao i u prethodnom slučaju, u našoj referentnoj gramatici srpskog jezika, konstrukcija s refleksivnim pasivom smatra se posebnim tipom subjekatsko-predikatske konstrukcije. Ona se koristi za izražavanje situacije s uopštenim ili kolektivnim agensom (Stanojčić & Popović, 1997, p. 246). Karakteristika ove konstrukcije, a u poređenju s odgovarajućom aktivnom, je to što se glagol nalazi u refleksivnom obliku. Subjekt se, iako podrazumevan, u ovim konstrukcijama ne iskazuje, dajući takozvanu deagentizovanu konstrukciju. Kako je agens svejedno implicitno prisutan u argumentskoj strukturi glagola, i dalje će za njega biti rezervisan prazan čvor u stablu. Rečca *se* postaje drugo sidro ovakve konstrukcije. Glagol kongruira sa pacijensom u funkciji subjekta u odgovarajućim kategorijama. Samo stablo s refleksivnim pasivom je inicijalno. Slika 76 prikazuje leksičko pravilo za ovakve konstrukcije.



Slika 76 Leksičko pravilo za refleksivni pasiv

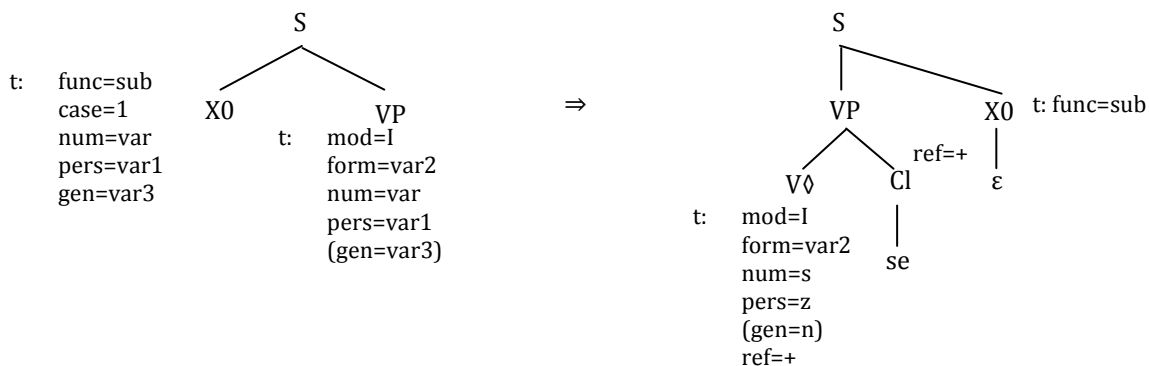
Slika 77 prikazuje primer rečenica povezanih ovim pravilom.



Slika 77 Primer leksičkog pravila za refleksivni pasiv na glagolu *čitati*

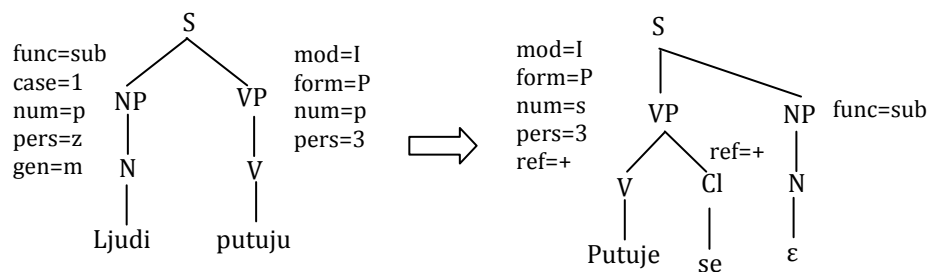
Obezličavanje

Obezličene konstrukcije po (Stanojčić & Popović, 1997, p. 243) predstavljaju vrstu bezlične konstrukcije. Njihova sličnost ogleda se u obliku glagola – i u bezličnim i u obezličanim konstrukcijama glagol se nalazi u neutralnom ili bezličnom obliku – 3. licu jednine (srednjeg roda). Obezličena se od bezličnih konstrukcija ipak razlikuje po tome što se u obezličanim konstrukcijama i dalje podrazumeva postojanje subjekta, iako je on ovde neiskazan. Struktura obezličanih rečenica je reflektivna, te je tako slična onoj kod reflektivnog pasiva, gde je rečca *se* takođe sidro. Obezličavanje se ipak javlja samo kod neprelaznih glagola. Budući da je subjekat semantički prisutan, kao i u prethodno pomenutoj konstrukciji, i ovde će za agensa će biti rezervisan prazan čvor. I ova struktura je predstavljena inicijalnim stablom. Slika 78 predstavlja dato leksičko pravilo.



Slika 78 Leksičko pravilo za obezličavanje

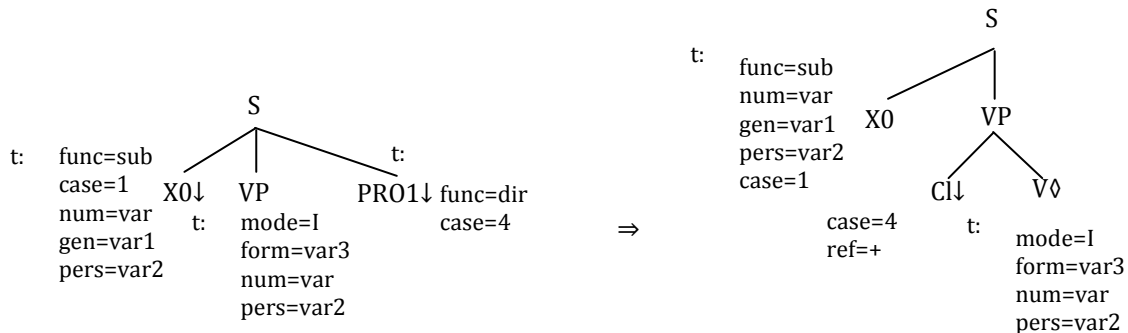
Na ovaj način su povezana dva stabla na sledećoj slici (Slika 79).



Slika 79 Primer leksičkog pravila za obezličavanje na glagolu *putovati*

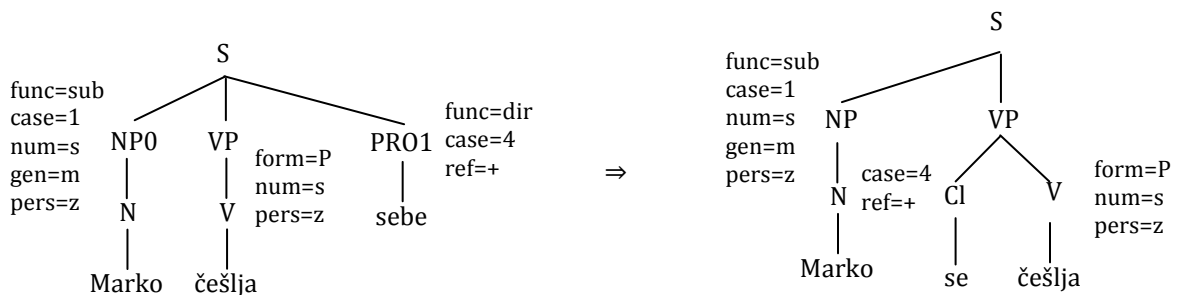
Refleksivizacija

Refleksivizacijom smatramo proces u kome se prelazni glagol transformiše u povratni neprelazni glagol, kao u primeru: *Marko češlja sebe.* → *Marko se češlja.*, pri čemu se dopuna u obliku pravog objekta zamenjuje rečom *se*. Refleksivizovanima smatramo sve glagole koji se ubrajaju u grupu pravih i uzajamo povratnih glagola prema klasifikaciji datoju u (Stanojčić & Popović, 1997, p. 100). Slika 80 prikazuje dato leksičko pravilo.



Slika 80 Leksičko pravilo refleksivizacije

Slika 81 predstavlja dva stabla povezana ovim leksičkim pravilom.



Slika 81 Primer leksičkog pravila refleksivizacije

3.6.1.2 Pravila izmeštanja

Pitanje

Za SrpTAG u ovom trenutku uzimamo u obzir tri vrste pitanja:

- pitanje koje podrazumeva izmeštanje dopune glagola, u ovom slučaju objekatsko pitanje:

82. *Marko jede jabuke.*

83. Šta Marko jede?

- subjekatsko pitanje

84. Marko jede jabuke.

85. Ko jede jabuke?

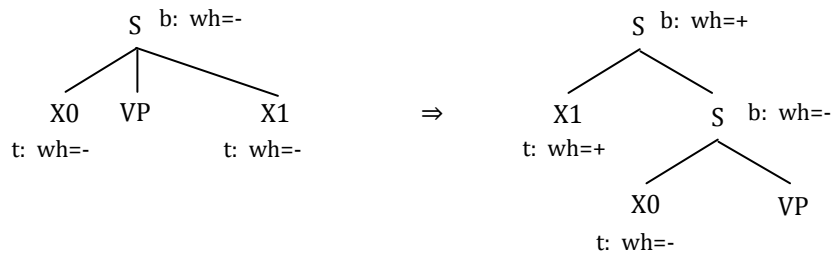
- pitanje s upitnim rečima koje ne predstavljaju varijante glagolskih argumenata:

86. Marko jede jabuke.

87. Zašto/Da li Marko jede jabuke?

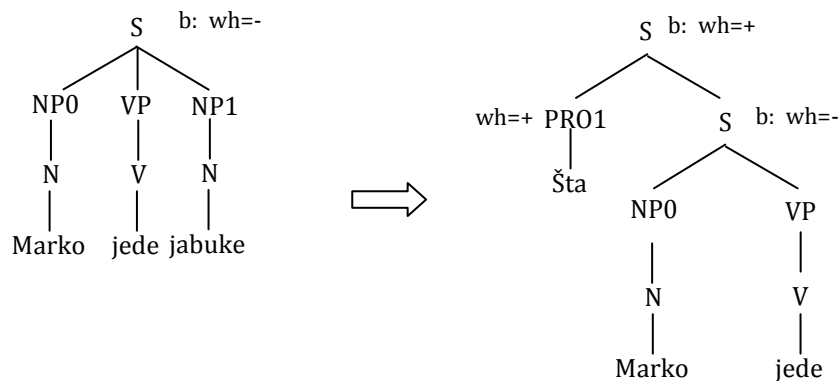
Od pomenutih pitanja, samo prvi i drugi tip pitanja se povezuju sa polaznom rečenicom pomoću leksičkih pravila. Pogledajmo kako izgleda svako od njih.

Leksičko pravilo za prvi tip pitanja dato je na narednoj slici (Slika 82). Kada se pogleda desna strana pravila, može se primetiti da je pored toga što je promenjeno mesto dopune, promenjena i vrednost obeležja upitnosti u pozitivnu (wh=+) na glavnom rečeničnom čvoru S i čvoru koji predstavlja izmeštenu dopunu. Iza izmeštenog čvora ne ostaje trag ni prazan čvor.



Slika 82 Izvođenje objekatskog pitanja - izmeštanje pravog objekta

Slika 83 prikazuje primer datog pravila.



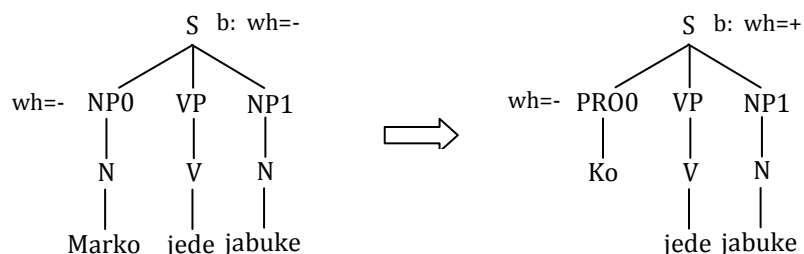
Slika 83 Leksičko pravilo izmeštanja za glagol *jesti*

Subjektska pitanja zapravo ne podrazumevaju izmeštanje već samo eventualnu promenu kategorije – ukoliko je polazna jedinica bila imenica, njena kategorija će postati zamenica – kao i promenu vrednosti obeležja upitnosti. Slika 84 prikazuje ovo leksičko pravilo.



Slika 84 Leksičko pravilo za subjektsko pitanje

Primer pravila za subjektsko pitanje dat je na narednoj slici (Slika 85).

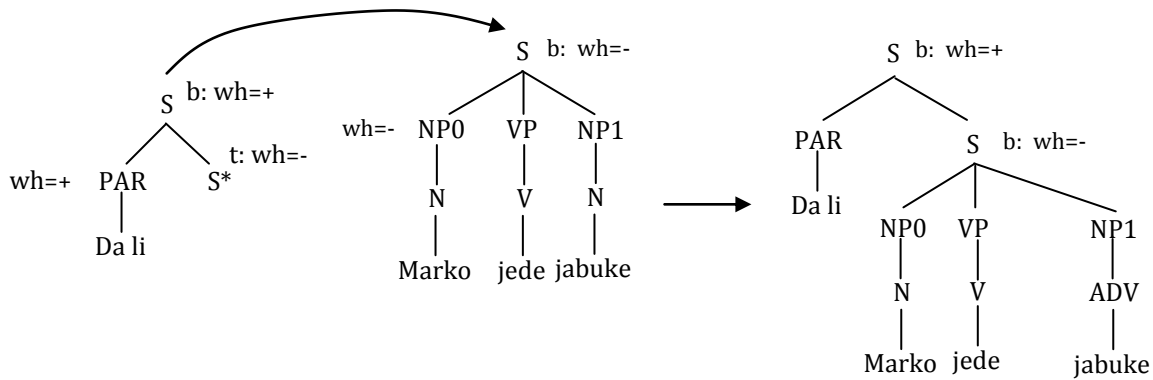


Slika 85 Primer leksičkog pravila za subjektsko pitanje *Ko jede jabuke?*

Treći tip pitanja nije povezan leksičkim pravilom sa svojom osnovnom verzijom. U ovom tipu pitanja dolazi samo do pripajanja upitne reči u obliku pomoćnog stabla s korenim čvorom S na koreni čvor S inicijalne rečenice.

Pitanja uvedena upitnim rečima koje ne predstavljaju alternative glagolskih argumenata ne podrazumevaju izmeštanje. Upitna reč se ovde predstavlja pomoćnim stablom s korenim čvorom S. To stablo se pripaja na bilo koje rečenično stablo koje već nije upitno. Slika 86 prikazuje izvođenje stabla s upitnom rečju *da li*. Ne ulazimo ovde u

strukturu jedinice *da li*, ali je inače tretiramo kao složenu jedinicu i svrstavamo je u kategoriju reči (PAR)¹⁵⁴.

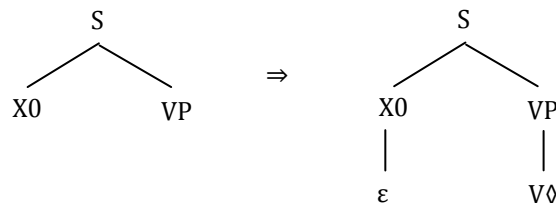


Slika 86 Primer izvođenja stabla *Da li Marko jede jabuke?*

3.6.1.3 Pravila realizacije

Nerealizovani argumenti

I pored toga što argumenti mogu da budu neiskazani u rečenici, sve dok su prisutni u argumentskoj strukturi glagola, potrebno je da budu predstavljeni i u strukturi odgovarajućeg stabla. Sami nerealizovani argumenti se predstavljaju kao inicijalno stablo čije je sidro prazno, a čiji je koren odgovarajuća kategorija. Na ovaj način se obežavaju subjekat ili dopuna izostavljena u rečenici¹⁵⁵. Primer leksičkog pravila za nerealizovan subjekat dato je na sledećoj slici (Slika 87).



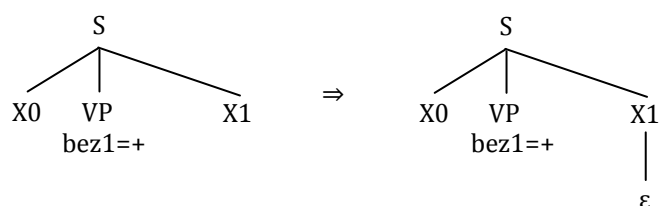
Slika 87 Leksičko pravilo za nerealizovan subjekat.

¹⁵⁴ Jedinicu *da li* posmatramo zajedno iz čisto praktičnih razloga, kako ovaj tip pitanja ne predstavlja pravu alternativu osnovnoj argumentskoj strukturi rečenica i kao takav nije od ključnog značaja u ovoj fazi izrade formalne gramatike. U budućnosti će svakako biti potrebe da se rečca *li* izdvoji kao upitna enklitika.

¹⁵⁵ ali isto tako i subjekat koji se nikada ne javlja u rečenici, i pored toga što se nalazi u argumentskoj strukturi glagola. U poglavlju o gramatičkom subjektu (3.5.2.1) mogli smo videti da takvu situaciju imamo kod rečenica u kojima je glagol u infinitivu i imperativu. Ovakve realizacije subjekta ipak nisu proizvod leksičkog pravila.

U leksičkom unosku svakog glagola nalaze se oznake bez1 i bez2 sa bulovskim vrednostima, koje koristimo da označimo da li dati glagol može imati izostavljene do dve dopune. Trovalentni glagol *dobiti* će tako imati sledeće vrednosti: bez1=- čime je označeno da ne može biti izostavljena prva dopuna (**Dobio sam od Marka¹⁵⁶.*), i bez2=+ koja označava da druga može biti izostavljena (*Dobio sam poklon.*). Pretpostavljamo da svi glagoli mogu imati izostavljen subjekat te za tu pojavu ne uvodimo posebno obeležje.

Slika 88 prikazuje opšte leksičko pravilo za nerealizovanu dopunu.



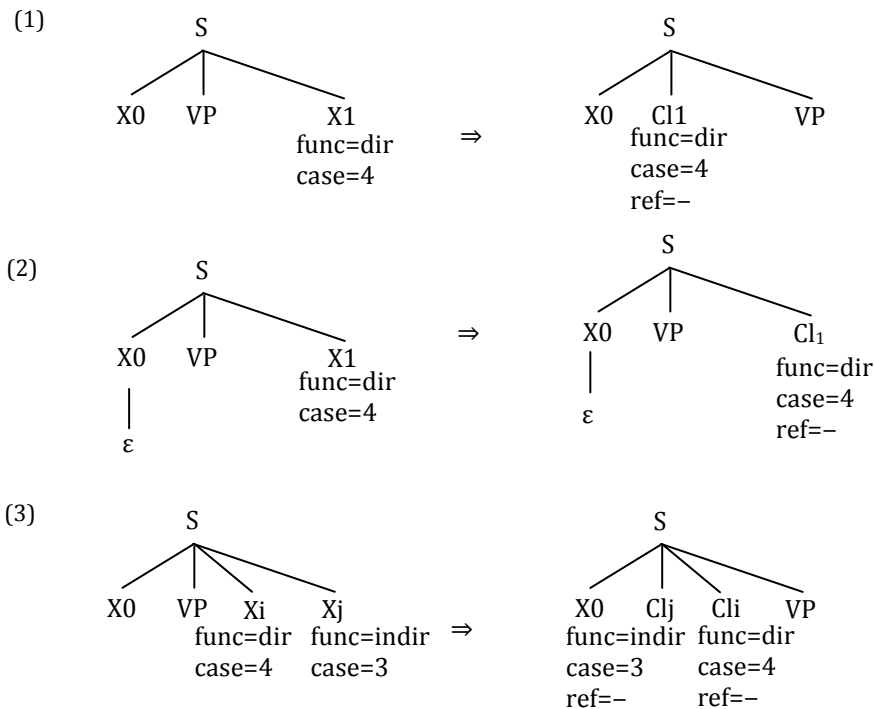
Slika 88 Leksičko pravilo za nerealizovanu dopunu

Kliticizacija

Pod pojmom kliticizacije podrazumevamo vezu koja se uspostavlja između jedinica u funkciji pravog objekta, nepravog objekta i logičkog subjekta i enklitika s odgovarajućom funkcijom. Budući da je podela zamenica na enklitike (Cl) i zamenice (PRO) tehnička, promenu kategorije koja se ovom prilikom dešava (PRO → Cl) ne smatramo dovoljnom da ovo pravilo proglasimo pravilom redistribucije. Još bitnije, funkcija ovih jedinica se ovom prilikom ne menja. Pored naziva kategorije, ono što se menja jeste njihov pojavni oblik, kao i pozicija u rečenici.

Slika 89 daje leksičko pravilo kliticizacije za pravi objekat. Pravilo kliticizacije je usko vezano za redosled argumenata u stablu. Ako posmatramo stabla s kanonskim redosledom argumenata, ukoliko je subjekatski argument realizovan, enklitika će se naći između subjekta i glagola (leksičko pravilo 1). Ukoliko subjekat nije realizovan, enklitika će se nalaziti iza glagola (leksičko pravilo 2). Leksičko pravilo 3 povezuje stabla u kojima su kliticizirana dva argumenta – u ovom slučaju pravi objekat (Xi) i nepravi objekat (Xj) kod dvorekcijskih glagola. Enklitike se raspoređuju po unapred određenom pravilu po kome se enklitika u dativu uvek nalazi ispred enklitike u akuzativu.

¹⁵⁶ Naravno, u odgovarajućem kontekstu i ova rečenica može biti gramatična.



Slika 89 Leksičko pravilo za kliticizaciju pravog objekta u slučaju realizovanog subjekta (1) i nerealizovanog subjekta (2), kliticizacije sa nepravim objektom (3)

Raspored argumenata

Opisivanje drugačijeg rasporeda argumenata u odnosu na kanonski podrazumeva definisanje po jednog stabla za svaki od tih redosleda. Ukoliko bismo to radili za, uzmimo, trovalentne glagole imali bismo 24 stabla koja propisuju različit raspored glagola i njegovih argumenata, i to u slučaju da nijedan od argumenata nije kliticiziran ili izostavljen. Jedno od rešenja za ovaj problem izrazite redundantnosti jeste definisanje stabala tako da budu neodređena po redosledu. I pored toga što smo uz čvorove stabala u našoj gramatici beležili indekse koji govore o linearnom rasporedu argumenata, mogli smo ih izostaviti i pretpostaviti da se argumenti mogu javiti u bilo kom redosledu. S druge strane, previše slobode u rasporedu argumenata predstavlja problem kod enklitika.

Odluka koju smo doneli za SrpTAG jeste da za argumente stabala zadržimo oznake koje govore o njihovom linearnom rasporedu. Pitanjem reda reči se tako nećemo dalje baviti unutar same SrpTAG. Međutim, moguće je red reči učiniti manje ili više slobodnim

unutar metagramatike. U ovom trenutku smo, zarad veće preciznosti, zadržali kanonski redosled argumenata i u metagramatici, uz neznatne izmene u slučaju izostavljenih argumenata. Eksperimentalno smo za stabla jedne rečenične porodice (porodica koja odgovara rečeničnom modelu broj šest) definisali slobodan red reči u odnosu na glagol, o čemu će biti više reči u poglavlju 4.3.2.

3.6.2 Subjekatsko-predikatski modeli

U ovom i narednom poglavlju dajemo pregled subjekatsko-predikatskih, a zatim i bezličnih modela, zajedno sa nazivima odgovarajućih porodica unutar SrpTAG. Uz modele navodimo i neke od glagola koji se javljaju u njima¹⁵⁷, kao i leksička pravila koja se javljaju za svaku od porodica. Budući da pravila realizacije ili izmeštanja nisu strukturnog tipa – ne menjaju broj, kategoriju ni funkciju rečeničnih argumenata, i kao takva se mogu primeniti na većinu rečeničnih modela¹⁵⁸, nećemo ih zasebno napominjati uz svaku porodicu, ali ćemo napominjati da li se na nju može primeniti neko od leksičkih pravila redistribucije.

1	Rečenica _{glagol bez dopune} = SUBJEKAT PREDIKAT	1) Zoran spava. Dolazi autobus! Neko je došao. 2) Ti se šališ.
---	---	---

Prvi, prost subjekatsko-predikatski model biće razdvojen na dva modela u SrpTAG, u skladu sa analizom refleksivnih glagola kao glagola sa dva sidra – glagolom i rečcom *se*. Odgovarajući modeli će tako biti **n₀V** za nerefleksivne i **n₀CIV** za refleksivne glagole.

Neki od glagola koji ulaze u prvi model su: *spavati, dolaziti (doći), sedeti, stajati, trčati, pasti, ustati, (za)plakati, spavati, plivati, leteti, rasti, skitati* a u drugi: *šaliti se, (na)smejati se, šetati se*.

Leksičko pravilo koje je primenjivo na ovaj model je obezličavanje.

2	Rečenica _{fakultativno prelazni glagol} = SUBJEKAT PREDIKAT (PRAVI OBJEKAT)	Jasna čita (knjigu). Zoran piše (pismo).
---	--	---

¹⁵⁷ Primere glagola mahom navodimo prema (Stanojčić & Popović, 1997).

¹⁵⁸ U slučaju kliticizacije – svuda gde postoji imenička dopuna ili zamenička dopuna u akcentovanom obliku.

Drugi rečenični model ćemo predstaviti jednom SrpTAG porodicom u kojoj dozvoljavamo mogućnost da drugi argument glagola, pravi objekat, bude izostavljen. Glagol koji ulazi u ovaj model će tako u leksikonu nositi obeležje bez1 s pozitivnom vrednošću. Model, sam po sebi, izgledaće ovako: **n₀Vn_{1dir}**.

U ovaj model ulaze glagoli: *čitati, pisati, pevati, jesti, piti*.

Leksička pravila primenljiva na ovaj model su pasiv i refleksivni pasiv.

3	Rečenica _{prelazni glagol} = SUBJEKAT PREDIKAT PRAVI OBJEKAT	1) Jasna je kupila knjigu . 2) Marko je pokušao da nađe karte za utakmicu . 3) Marko želi otići . 4) Marko zna da li ima hleba .
---	---	---

Pored modela identičnog kao u prethodnom primeru: **n₀Vn_{1dir}**, (primer 1) ovaj model može imati još svojih alternativa. Za glagole sa deklarativnom rečeničnom dopunom, u kom slučaju komplementizator predstavlja dodatno sidro u rečenici, definisali smo model: **n₀Vdas₁** (primer 2), a za glagole s dopunom u obliku infinitiva **n₀Vsinf_{1dir}** (primer 3), i za glagole sa upitnom rečeničnom dopunom, **n₀Vsup_{1dir}** (primer 4).

Neki od glagola koji ulaze u ovaj model su: *kupiti, pokušati, želeti, znati, razumeti, uzeti, doneti, češljati, prati, umivati*.

Pored leksičkih pravila pasivizacije (participske i refleksivne), na pojedine glagole iz ovog **n₀Vn_{1dir}** modela primenljiva je i refleksivizacija (*češljati, prati, umivati*).

4	Rečenica _{neprelazni rekijski glagol} = SUBJEKAT PREDIKAT NEPRAVI OBJEKAT	1) Otac pomaže sinu . / Nikola je klimnuo glavom . 2) Ivan se stidi brata . / Marko se napio vode . / Tanja se obradovala poklonu . / Nikola se služi voćem . 3) Marko je odustao od svoje ideje . 4) Nikola se zaljubio u Tanju . / Nikola se interesuje za slikarstvo . / Tanja se navikla na novu sredinu . 5) Ivan je razmišljao da se upiše na prava . / Ivan je pristao da on kupi karte za bioskop . 6) Ivan se pomirio da neće ići na skijanje .
---	--	--

Za ovaj rečenični model postoji niz ekvivalenata u SrpTAG. Ako pravimo jedan zajednički model za sve imeničke jedinice bez predloga, on bi izgledao ovako: **n₀Vn_{1ind}** (primer 1) ili ovako, ukoliko je glagol refleksivan: **n₀ClVn_{1ind}** (primer 2).

Po padežima nepravog objekta, glagoli koji ulaze u ovaj model su:

- genitiv: **n₀ClVn_{1ind}** – *najesti se, napiti se, prihvatiti se, latiti se, domoći se, zaželeći se*
- dativ: **n₀Vn_{1ind}** – *služiti, povlašćivati, verovati*, **n₀ClVn_{1ind}** – *obratiti se, zahvaliti (se), diviti se, radovati se*
- instrumental: **n₀Vn_{1ind}** – *upravljati, rukovoditi, vladati, rukovati, slegnuti (ramenima), klimnuti (glavom), mahnuti (rukom)* i **n₀ClVn_{1ind}** – *služiti se, koristiti se, baviti se, zanimati se, oduševiti se.*

n₀Vpn_{1ind} (primer 3) i **n₀ClVpn_{1ind}** (primer 4) su generalni obrasci za stabala sa predlogom kao drugim sidrom i imeničkom jedinicom u zavisnom padežu. Kako su u pitanju regirani predlozi, svi će ulaziti direktno u sastav dosad navedenih modela. Konkretni modeli su tako:

- **n₀Vodn_{1ind}**: (genitiv) *zavisiti, odustati, odudarati* i **n₀ClVodn_{1ind}**: (genitiv) *odučiti se, uzdržati se*
- **n₀Vpreman_{1ind}**: (dativ) *ohladneti*
- **n₀Vnan_{1ind}**: (akuzativ) *ličiti, misliti, nasrnuti*; (lokativ) *insistirati* i **n₀ClVnan_{1ind}**: (akuzativ) *navići se, odvažiti se*; (lokativ) *iskaliti se, zasnivati se*
- **n₀Vun_{1ind}**: (akuzativ) *sumnjati*; (lokativ) *učestvovati, uživati, uspeti* i **n₀ClVun_{1ind}**: (akuzativ) *zaljubiti se, uveriti se*
- **n₀Vzan_{1ind}**: (instrumental) *čeznuti, žudeti, žaliti, tugovati* i **n₀ClVzan_{1ind}**: (akuzativ) *zalagati se, opredeliti se, zanimati se, interesovati se*
- **n₀Vsan_{1ind}**: (instrumental) *početi, nastaviti, prestati, odugovlačiti, saosećati*
- **n₀Von_{1ind}**: (lokativ) *govoriti, pričati, pripovedati, raspravljati, diskutovati, razmišljati, misliti, maštati, sanjati, sanjariti*

Među primerima sa rečeničnim argumentima, tu su **n₀Vdas_{1ind}** (primer 5) i **n₀ClVdas_{1ind}** (primer 6).

Leksička pravila primenljiva na ovaj model su: participijski pasiv (za pojedine glagole – npr. *pomagati*) i obezličavanje.

5	Rečenica _{neprelazni glagol sa priloškom dopunom} = SUBJEKAT PREDIKAT PRILOŠKA DOPUNA	1) Ja stanujem ovde . 2) Nikola se ponašao čudno . / Nikola se drži pogrbljeno . 3) Torba leži na klupi . / Ivan živi u Beogradu . / Ova sprava služi za košenje trave . 4) Knjiga se nalazi na stolu . 5) Račun iznosi 10.000 dinara . / Ovaj džak teži 10 kilograma . 6) Marko je otišao večerati . 7) Marko je otišao da večera .
---	---	---

Tipični predstavnik ovog modela u SrpTAG je onaj u kome se priloška dopuna realizuje kao priloška fraza. Primer 1 je tako predstava sledećeg modela: **n₀Vad₁**. Refleksivni glagol s istim tipom dopune predstavljamo kao **n₀ClVad₁** (primer 2).

Neki glagoli koji ulaze u ove tipove modela su: **n₀Vad₁** – *stanovati, živeti* i **n₀ClVad₁** – *nalaziti se, ponašati se, držati se*.

Priloška dopuna realizovana kao predložka fraza ili predložko-padežna konstrukcija javlja se u odgovarajuća dva modela: **n₀Vpn_{1pril}** (primer 3) i sa refleksivnim glagolima: **n₀ClVpn_{1pril}** (primer 4).

U ove modele ulaze sledeći glagoli: **n₀Vpn_{1pril}** – *visiti, staviti, obesiti, stajati, stanovati, živeti, obitavati* i **n₀ClVpn_{1pril}** – *nalaziti se, nadneti se*.

Model sa priloškom dopunom u obliku brojne fraze definišemo kao **n₀Vnumn₁** (primer 5).

Glagoli koji ulaze u ovaj model su *težiti, koštati* i *iznositi*.

Priloška dopuna se može realizovati i kao namerna rečenica s glagolom u obliku infinitiva: **n₀Vsinf_{1pril}** (primer 6) ili glagolom u ličnom glagolskom obliku: **n₀Vsad₁** (primer 7).

Leksičko pravilo koje se može primeniti na ovaj model je obezličavanje.

6	Rečenica _{neprelazni glagol sa logičkim subjektom} = LOGIČKI SUBJEKAT PREDIKAT (GRAMATIČKI) SUBJEKAT	1) Marku je zadrhtala <u>ruka</u> . 2) Jasnu je boleo <u>zub</u> . 3) Marku smeta <u>što puše</u> . / Marku prija <u>da pliva</u> . 4) Marku se sviđa Marija. 5) Marku se sviđa da pušta muziku. / Marku se sviđa što Marija ne puši.
---	--	---

Rečnični modeli sa logičkim subjektom u svom sastavu uvek imaju i gramatički subjekat, kao jedinicu sa kojom kongruira glagol. Osnovni model za ovaj tip rečenica je $n_{1sub}Vn_0$, gde se n_{1sub} realizuje kao imenička jedinica u dativu (primer 1) ili akuzativu (primer 2), a n_0 kao gramatički subjekat. Gramatički subjekat se može realizovati i kao deklarativna izrična rečenica (primeri 3), čineći model $n_{1sub}Vcs_0$.

Glagoli koji ulaze u ovaj model su: (dativ) *zadrhtati, prijati, smetati, odgovarati, trebati, sevati*¹⁵⁹, (akuzativ) *boleti, svrbeti, interesovati, zanimati, radovati, nervirati*.

U ekvivalentne modele ulaze i povratni glagoli, te tako imamo model $n_{1sub}ClVn_0$ (primer 4) i sa gramatičkim subjektom u obliku izrične rečenice $n_{1sub}ClVcs_0$ (primeri 5).

Glagoli koji ulaze u ove konstrukcije su: *sviđati se, dopadati se*.

Na ovaj model se ne primenjuje nijedno od leksičkih pravila redistribucije.

7	Rečenica _{glagol davanja ili saopštavanja} = SUBJEKAT PREDIKAT NEPRAVI OBJEKAT PRAVI OBJEKAT	1) Tanja je vratila knjigu Zoranu . 2) Marko je poručio Ivanu da mu vrati ploče .
---	--	--

Glagoli koji ulaze u ove konstrukcije su takozvani dvorekcijski glagoli ili glagoli koji traže uz sebe dve rekcijske dopune: jednu u obliku pravog objekta, koji se može realizovati kao imenička jedinica u akuzativu bez predloga (primer 1) ili izrična rečenica (primer 2), a drugu u obliku nepravog objekta, koja se realizuje kao imenička jedinica u dativu bez predloga (primeri 1 i 2). Tipičan model u koji ulaze ovi glagoli je tako $n_0Vn_{1dir}n_{2ind}$ (primer 1). Ovi glagoli u leksikonu imaju pozitivnu vrednost obeležja bez2, koje označava da njihov nepravni objekat može biti izostavljen iz rečenice.

¹⁵⁹ Kao u primeru *Marku sevaju oči*.

Pravi objekat ovih glagola može se javiti i u obliku deklarativne izrične rečenice, čineći model: **n₀Vn_{2ind}das_{1dir}** (primer 2).

Glagoli koji ulaze u navedene modele su: *reći, kazati, saopštiti, informisati, javiti, poručiti, dati, pokloniti, preneti*.

Leksička pravila koja se mogu primeniti na ovaj model su: participski pasiv, reflektivni pasiv i reflektivizacija.

8	Rečenica _{prelazni glagol sa priloškom dopunom} = SUBJEKAT PREDIKAT PRAVI OBJEKAT PRILOŠKA DOPUNA	1) Marko je stavio kameru blizu . 2) Ivan je ubacio pismo u sanduče . 3) Marko je platio knjigu 100 dinara .
---	---	---

Tipičan model glagola koji traže pravi objekat i prilošku dopunu kao svoje argumente jeste **n₀Vn_{1dir}ad₂** (primer 1). Za prilošku dopunu realizovanu kao predložka fraza (primer 2), odgovarajući model je **n₀Vn_{1dir}pn_{2pril}**. Priloška fraza se realizuje kao brojna sintagma u primeru 3, ulazeći u model **n₀Vn_{1dir}numn₂**.

Glagoli koji ulaze u ovaj model su: *ubaciti, položiti, uvesti, smestiti, ostaviti, staviti, rasporediti, izvaditi, izbaciti, skloniti, pomeriti, platiti*.

Leksička pravila koja se primenjuju na ovaj model su: participski pasiv, reflektivni pasiv i reflektivizacija.

9	Rečenica _{imenski predikat} = SUBJEKAT KOPULA IMENSKI PREDIKATIV	1) Ovo je Marija . 2) Jasna je pametna .
---	--	---

Rečenice sa imenskim kopulativnim predikatom ulaze u dva osnovna modela. Oba modela su neglagolska, tj. glagol ne predstavlja njihovo sidro. Shodno tome, beležimo ga malim slovom, a nosioce modela, imenski predikativ, beležimo velikim. Primer 1 predstavlja model **n₀vN**, u kome se imenica *Marija* javlja kao sidro stabla. Primer 2 je predstavnik modela **n₀vA**, gde je pridev *pametna* odgovarajuće sidro.

Na ovaj model se ne primenjuje nijedno od leksičkih pravila redistribucije.

10	Rečenica _{priloški predikat} = SUBJEKAT KOPULA PRILOŠKI PREDIKATIV	1) Stanica je daleko . 2) Hleb je 50 dinara .
----	--	--

Tipičan model za ovaj tip rečenica u SrpTAG jeste onaj u kome se priloški predikativ realizuje kao priloška fraza: **noVAd** (primer 1). Kao i u modelu 9, i ovde se glagol beleži malim slovom, a predikativ velikim, budući da predstavlja sidro odgovarajućeg stabla.

U obliku brojne fraze (primer 2), priloški predikativ ulazi u model **noVNUMn**. U ovim konstrukcijama za sidro uzimamo samo brojnu reč, ne i imenicu koja ulazi u datu brojnu sintagmu.

Kako je već napomenuto u poglavlju 3.5.1, u ovom trenutku nećemo obrađivati realizaciju priloškog predikativa kao predložko-padežne konstrukcije, zbog čega ga i ne navodimo među primerima.

Kao i u prethodnom modelu, ni na ovaj model se ne primenjuje nijedno od leksičkih pravila redistribucije.

3.6.3 Bezlični modeli

Kako nemaju subjekat u svojoj strukturi, bezlične rečenice se karakterišu glagolom u 3. licu jednine (srednjeg roda). Autori navode da se ovaj tip rečenica koristi da se označe “atmosferske pojave, psihološka i fiziološka stanja i procesi, postojanje ili nepostojanje nekog pojma” (Stanojčić & Popović, 1997, p. 239).

Pogledajmo i ove modele zasebno, navodeći uz svaki i odgovarajuće porodice u SrpTAG.

1	Bezlična rečenica _{bezlični glagol} = PREDIKAT (bezlični glagol u 3. licu jednine (srednjeg roda) ličnog (finitnog) oblika)	1) Seva. 2) Smrkava se.
---	--	----------------------------

Ovaj model je ujedno i najjminimalniji model predikatskih rečenica – sastoji se samo od glagola: **V** (primer 1).

Ukoliko je glagol refleksivan, model će sadržati i rečcu *se* kao drugo sidro: **VCl** (primer 2).

U ove modele ulaze i sledeći glagoli: **V** – *svanjivati, svanuti, sevati, grmeti, duvati, pljuštati*; **VC1** – *naoblačiti se, razvedriti se, otopliti, zahladneti, smračiti se, smrći se, smrkavati se*.

Na ovaj, kao ni na ostale modele bezličnih rečenica, ne primenjuju se leksička pravila redistribucije.

2	Bezlična rečenica _{priloški predikat} = GLAGOLSKI DEO PREDIKATA (bezlični glagol u 3. licu jednine (srednjeg roda) ličnog (finitnog) oblika) PRILOŠKI DEO PREDIKATA	(Danas) je toplo . / Sparno je.
---	--	---

Ni ovaj model ne poseduje subjekatsku poziciju. Umesto subjekta, na toj poziciji se ponekad može javiti prilog, doduše samo kao odredba, te kao takav neće ulaziti u modele. Odgovarajući model u SrpTAG je **vAd**.

U ovakav model mogu ući prilozii: *hladno, toplo, vruće, zagušljivo, sparno*.

3	Bezlična rečenica _{bezlični glagol} = LOGIČKI SUBJEKAT PREDIKAT (bezlični glagol u 3. licu jednine (srednjeg roda) ličnog (finitnog) oblika)	1) Marku je pozlilo. 2) Marku se slošilo. 3) (Danas) ima/nema nastave .
---	---	--

U ovaj model ulaze glagoli koji kao svoju dopunu traže logički subjekat, i to u obliku dativa bez predloga (primeri 1 i 2) i genitiva bez predloga (primer 3). Primer 1 i primer 3 su tako primeri modela **n01subV**, odnosno **Vn01sub**.

U ovaj model ulaze glagoli: *pozliti, presemi, dosaditi, dojaditi, svanuti*.

Primer 2, kako sadrži refleksivni glagol, je primer modela **n01subCIV**.

U ovaj model ulaze glagoli: *složiti se, smučiti se*.

4	Bezlična rečenica _{priloški predikat} = LOGIČKI SUBJEKAT GLAGOLSKI DEO PREDIKATA (bezlični glagol u 3. licu jednine (srednjeg roda) ličnog (finitnog) oblika) PRILOŠKI DEO PREDIKATA	Marku je dosadno .
---	---	----------------------------------

Nosilac ovog modela u SrpTAG je, kao i u subjekatskim rečenicama, priloški deo predikata, te će model kojim predstavljamo ove rečenice biti **n01subvAd**.

3.7 Neobrađeni slučajevi

U ovom radu nismo obradili sledeće segmente srpske gramatike:

- Aktuelni kvalifikativ – (Stanojčić & Popović, 1997) uvode aktuelni kvalifikativ kao jedinicu koja može imati funkciju dopune uz pojedine glagole (*ostaviti* ili *zateći*) i funkciju odredbe, što je češće. Kao dopuna, ova jedinica definiše da je određena karakteristika subjekta (subjekatski aktuelni kvalifikativ – primer 88) ili objekta (objekatski aktuelni kvalifikativ, primer 89) aktuelna u vreme realizovanja situacije. Budući da pripisuje sadržaj subjektu ili objektu, srodan je predikativima. Može se realizovati kao pridevska jedinica u nominativu (u slučaju subjekatskog aktuelnog kvalifikativa), odnosno akuzativu (u slučaju objekatskog aktuelnog kvalifikativa). Budući da se ova jedinica mahom javlja kao odredba i ne javlja ni u jednom od rečeničnih modela u pomenutoj gramatici, nismo je ni mi uključili u obradu.

88. *Nikolu smo ostavili loše rasploženi.*

89. *Ivan je zatekao brata bolesnog.*

- Dopunski predikativ i modele sa dopunskim predikativom – takozvani *semikopulativni* glagoli, koji kao svoju dopunu traže dopunski predikativ, imaju istovremeno i karakteristike kopulativnih i punoznačnih glagola. Budući da ove dve vrste glagola potpuno drugačije tretiramo u SrpTAG, kao i njihove dopune, odnosno predikative (za detalje tretmana kopulativnih konstrukcija videti poglavlje 3.4.2) nije bilo potpuno jasno na koji način im treba pristupiti ovoj konstrukciji. Iz tog razloga smo ovu konstrukciju i rečenične modele sa njom za sada ostavili neobrađene.
- Zavisne rečenice, sem izričnih zavisnih rečenica – u ovom trenutku unutar formalne gramatike imamo datu detaljnu analizu izričnih rečenica. Iako ne dajemo detaljnu analizu namernih rečenica, one su uključene u modele porodica stabala. U radu dajemo i ilustrativnu analizu odnosnih rečenica (poglavlje 3.3 i 3.5.1), ali kako one u SrpTAG imaju samo modifikatorsku ulogu, i kao takve ne ulaze u sastav rečeničnih modela,

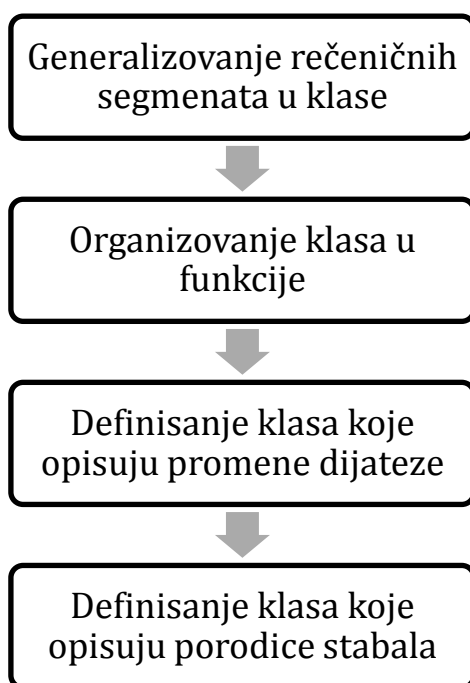
nismo ih uključili u sledeću fazu rada – izradu metagramatike za SrpTAG. Kako se generalno nismo bavili zavisnim rečenicama s modifikatorskom funkcijom, nismo obrađivali nijednu od preostalih vrsta zavisnih rečenica s ovom funkcijom.

- Odredbe, uključujući agentivnu odredbu, aktuelni kvalifikativ kao odredbu, apozitiv i apoziciju – kao što u obradu nismo uključili rečenične segmente s modifikatorskom ulogom, tako nismo uključili ni fraze ili pojedinačne kategorije s odredbenom ili modifikatorskom ulogom, iz istih razloga.
- Zavisne članove sintagmi – imeničkih sintagmi (*zanimljiva knjiga, džemper sive boje, ptica selica, gospodin Nikola...*), pridevskih i priloških sintagmi (*veoma brz(o), brz(o) poput munje, brži (brže) nego munja, dostojan (dostojno) nečega, svestan da je pogrešio...*). Ne postoji posebna tehnička poteškoća sa uključivanjem zavisnih članova sintagmi u rad. Jedini razlog što nisu u ovom trenutku uključeni jeste što postoji veliki broj različitih oblika zavisnih članova sintagmi, koji upravo zbog te raznovrsnosti i vremena koje zahtevaju za detaljnu izradu po nama zahtevaju posebnu pažnju i rad za sebe.
- Realizacije pojedinih čvorova – kao što je predložka fraza kao predikativ. Kako predlog, kao funkcionalna reč, ne može biti sidro rečenične strukture, a javljanje predložke fraze unutar kopulativne konstrukcije to zahteva, nije sasvim jasno kako analizirati ovu frazu u funkciji predikativa.
- Konstrukciju *da+prezent* kod faznih glagola – kako ne postoji način da se unutar TAG gramatike na adekvatan način tretira ova konstrukcija. Za detalje analize fazne konstrukcije s *da+prezent* i infinitivom, pogledati poglavlje 3.3.
- Red reči – u SrpTAG se ne bavimo eksplicitno redom reči, pre svega redosledom enklitika. Kako se neke od enklitika nalaze direktno pod čvorom S (enklitički oblici zamenica u funkciji pravog, nepravog objekta ili logičkog subjekta), a neke pod čvorom VP (rečca *se*, enklitički oblici pomoćnih glagola), nije u potpunosti jasno kako će njihov međusobni redosled moći da bude ostvaren. Više reči o tome kako se red reči može regulisati biće u delu o metagramatici.
- Neke slučajeve uslovljenosti, kao kod upotrebe negativnih zamenica, za čije je unošenje u rečenično stablo potrebno da ono nosi oznaku negativnosti (*wh=+*).

4 METAGRAMATIKA ZA SRPTAG

4.1 Metodologija

Prilikom izrade metagramatike za formalnu gramatiku srpskog jezika SrpTAG koristićemo metodologiju izloženu u (Crabbé, 2005). Prema ovoj metodologiji, metagramatika gramatike stabala izrađuje se u četiri faze, odnosno kroz četiri nivoa generalizovanja (Slika 90).

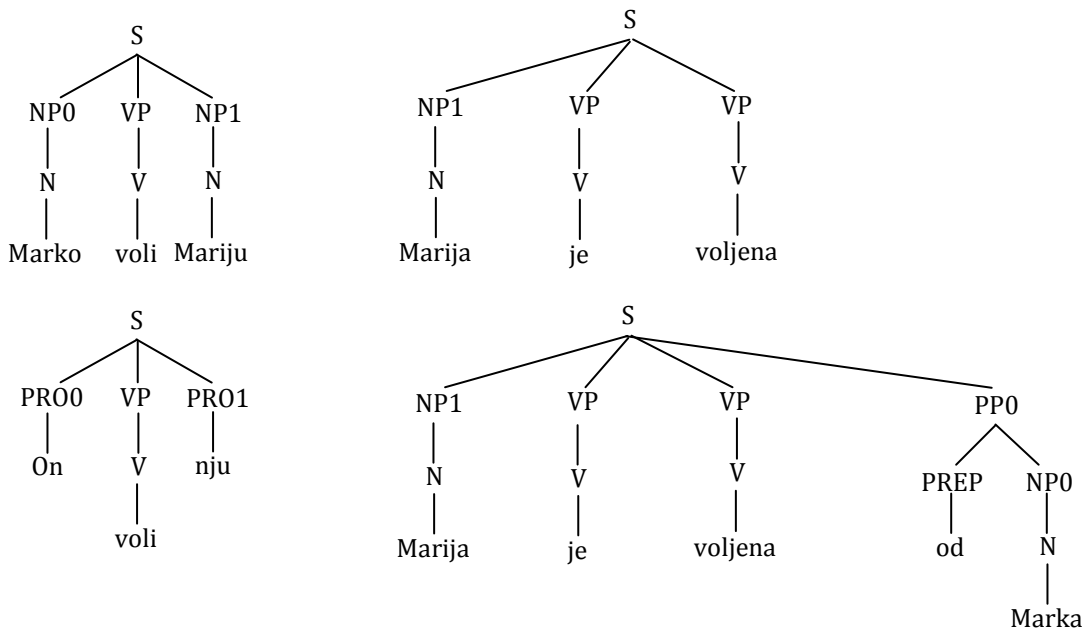


Slika 90 Faze izrade metagramatike

U poglavlju u kome smo napravili uvod u XMG (poglavlje 2.5.1) pomenuli smo da se XMG bazira na pojmu klase, i kao što iz navedenog dijagrama možemo videti, svaka od faza u izradi metagramatike podrazumeva izvesnu organizaciju u klase. Prva faza generalizacije sadrži klase koje čine osnovu cele metagramatike. Sve naredne faze u izradi metagramatike podrazumevaju manipulaciju ovim osnovnim klasama – udruženim u natklase, koje se dalje međusobno povezuju operacijom disjunkcije (*ili*) ili konjunkcije (*i*). Pogledajmo kako ovaj proces izgleda na jednom ilustrativnom primeru.

Recimo da želimo da izradimo gramatiku koja bi mogla da opiše podskup jezika koji se sastoji od sledećih rečenica: *Marko voli Mariju., On voli nju., Marko voli nju., Ona voli Marka., Marija je voljena od Marka., Ona je voljena od Marka., Marija je voljena od njega., Ona*

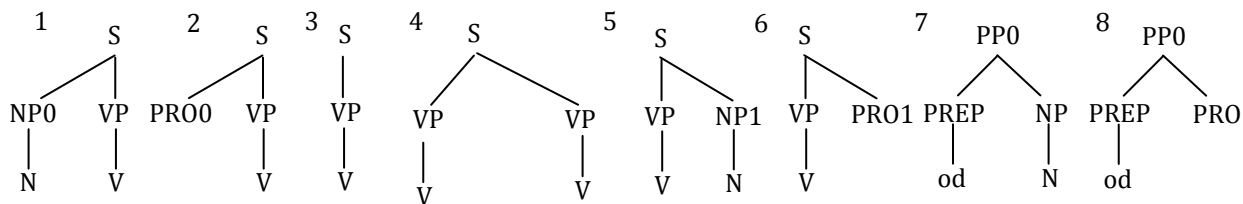
je voljena od njega., Marija je voljena., Ona je voljena. Gramatika koju želimo da koristimo za opisivanje ovih rečenica je, podrazumevano, gramatika stabala. Kako se svaka od pomenutih rečenica razlikuje, u gramatici stabala je potrebno definisati po jedno stablo za svaku od njih. Slika 91 prikazuje stabla za neke od ovih rečenica.



Slika 91 Stabla ilustrativne gramatike

Iz ovog primera možemo videti da čak i jedan sveden jezik sadrži značajan broj stabala. Da ne bismo svako od ovih stabala generisali ručno, za njihovu izradu koristimo mehanizam za opisivanje i generisanje stabala, odnosno metagramatiku.

Prvo što činimo kada pravimo metagramatiku jeste da utvrdimo da li postoje delovi stabala koji se ponavljaju, kao i koja je njihova funkcija. Tako možemo videti da se segmenti stabala (Slika 92) ponavljaju u većem broju stabala.



Slika 92 Segmenti stabala ilustrativne gramatike

U skladu sa najčešćim primerima upotrebe, nazvaćemo ove segmente redom, intuitivnim nazivima:

ImeničkiSubjekat (1), ZameničkiSubjekat (2), PredikatAktiv (3), PredikatPasiv (4), ImeničkiObjekat (5), ZameničkiObjekat (6), ImeničkaAgentivnaOdredba (7), ZameničkaAgentivnaOdredba (8).

Urativši ovo, definisali smo osnovne klase naše metagramatike i samim tim završili prvu fazu u procesu izrade metagramatike. Od ove faze više ne operišemo konkretnim segmentima stabala, već njihovim nazivima.

Kao što se može primetiti, među imenima klasa koja smo upravo definisali ima izvesnog poklapanja, koje pokazuje da se određene klase javljaju u istoj funkciji. U sledećoj fazi izrade metagramatike, sve takve klase udružujemo pod jednim zajedničkim imenom. U te svrhe koristimo operaciju disjunkcije (\vee), jednu od dve operacije koje propisuje metagramatika. Na taj način ćemo dobiti sledeće natklase:

Subjekat \rightarrow ImeničkiSubjekat \vee ZameničkiSubjekat

Predikat \rightarrow PredikatAktiv \vee PredikatPasiv

Objekat \rightarrow ImeničkiObjekat \vee ZameničkiObjekat

AgentivnaOdredba \rightarrow ImeničkaAgentivnaOdredba \vee ZameničkaAgentivnaOdredba

Razliku među dvema klasama za predikat ćemo iskoristiti za definisanje sledeće grupe klasa. Ovakvim grupisanjem klasa ujedno smo završili i drugu fazu građenja metagramatike.

U sledećoj fazi u izradi metagramatike želimo da grupišemo natklase koje smo napravili u prethodnoj fazi, i to tako da nove klase oslikavaju promene koje su se desile u predikatskoj strukturi naših rečenica. Ako pogledamo sve rečenice naše gramatike, videćemo da je jedan broj njih aktivan, odnosno da se njihov predikat sastoji od glagola u aktivnom obliku (aktivna dijateza). Drugi deo rečenica je pasivan, i njihov predikat se sastoji

od glagola u pasivnom obliku (pasivna dijateza). Ovo zapažanje pretočićemo u tri nove klase:

$$\textit{AktivnaDijateza} \rightarrow \textit{Subjekat} \wedge \textit{PredikatAktiv} \wedge \textit{Objekat}$$
$$\textit{PasivnaDijatezaBezAgensa} \rightarrow \textit{Subjekat} \wedge \textit{PredikatPasiv}$$
$$\textit{PasivnaDijatezaSaAgensom} \rightarrow \textit{Subjekat} \wedge \textit{PredikatPasiv} \wedge \textit{AgentivnaOdredba}$$

Dve klase pasivnih rečenica možemo udružiti u jednu operacijom disjunkcije:

$$\textit{PasivnaDijateza} \rightarrow \textit{PasivnaDijatezaBezAgensa} \vee \textit{PasivnaDijatezaSaAgensom}$$

Ovime smo okončali i treću fazu u definisanju metagramatike. Poslednja, četvrta faza podrazumeva udruživanje klasa napravljenih u prethodnom koraku u jednu, centralnu. Ovu klasu nazvaćemo *Prelazan*:

$$\textit{Prelazan} \rightarrow \textit{AktivnaDijateza} \vee \textit{PasivnaDijateza}$$

Na ovaj način smo sva stabla našeg jezika definisali preko jedne klase, koja će predstavljati generičko ime za porodicu stabala nastalih od prelaznih glagola. Operišući apstraktnim klasama kakva je klasa *Prelazan*, metagramatika, odnosno kompilator metagramatike XMG, može da kreira sheme stabala za sve gore navedene rečenice¹⁶⁰.

Treba napomenuti da se alternative u realizaciji rečeničnih predikata ne svode na aktivnu-pasivnu dijatezu. Na gore opisan način mogu se povezati sve alternative predikatskih stabala koje opisuje neko leksičko pravilo redistribucije, kakvo je pravilo za reflektivni pasiv ili obezličavanje. Jednu porodicu stabala (rečenični model) činiće skup svih alternativa mogućih za tip glagola koji čini osnovu te porodice.

Pri predstavljanju klasa metagramatike za SrpTAG (poglavlje 4.3) odlučili smo da ih predstavljamo u paru – prva dva koraka zajedno, a zatim i druga dva koraka zajedno. Iz našeg iskustva se pokazalo da su ove faze vrlo blisko povezane i njihovo eksplicitno

¹⁶⁰ XMG generiše isključivo sheme stabala, ne i leksikalizovana stabla.

izdvajanje u odvojena poglavlja bi bilo ne samo nepraktično, već bi i narušilo konzistentnost izlaganja sadržaja metagmatike.

4.2 Jezik za opis metagramatike

Pre nego što krenemo u opis metagramatike srpskog jezika, pogledajmo bliže kako izgleda konkretan programski jezik kojim ćemo se služiti pri opisu metagramatike. Za opis metagramatike koristi se programski jezik objektno-orijentisanog tipa¹⁶¹. Čitavu metagramatiku smo organizovali u tri glavna dokumenta ili modula:

1. Zaglavlja, u kome definišemo tipove, obeležja i svojstva, i gde se pozivaju moduli koji se primenjuju u gramatici. Ovaj modul smo nazvali *heder.mg*¹⁶²;
2. Definicije klasa, u kojima se definišu konkretne klase metagramatike. Podelili smo ih u četiri dokumenta – *argumenti.mg* u kome smo definisali klase glagolskih argumenata, *glagoli.mg* u kome smo definisali klase glagola i predikata, *ostalo.mg* gde smo definisali klase pojedinačnih čvorova i čvorova pojedinih pomoćnih stabala i *modeli.mg* gde smo definisali modele za porodice rečeničnih stabala;
3. Procedure za kompiliranje klasa. Ovaj modul smo nazvali *final.mg*.

1) U zaglavlju se definišu tipovi, obeležja i svojstva koja će biti korišćena u gramatici. Tipovi mogu biti ugrađeni (bulovski - *bool*, celobrojni – *int*, niske – *string*), može ih definisati sam korisnik, ili mogu biti nespecificovani – za situacije kada ne želimo da definišemo prihvatljivu vrednost. Korisnik definiše obeležja i svojstva, i njihove vrednosti. Po konvenciji, za tipove se koriste velika slova, a za svojstva i obeležja mala. U ovom delu se takođe pozivaju i moduli koji će biti korišćeni u gramatici, u našem slučaju modul koji reguliše identifikaciju čvorova putem sistema boja (modul *color*) i modul za regulisanje reda

¹⁶¹ Ukratko, objektno-orijentisan jezik je programski jezik koji se bazira na konceptu objekata (modula, klasa) koji sadrže određen skup informacija i njima dodeljenih procedura (metoda, akcija), koje mogu da modifikuju ciljane objekte. Objekti međusobno mogu interreagovati pozivanjem drugih objekata unutar sebe, čime im se omogućava i korišćenje njihovih informacija i procedura. Za više detalja o logici objektno-orijentisanih jezika pogledati (Kunz, 1991).

¹⁶² .mg je ekstenzija koju nose svi dokumenti ove metagramatike.

reči među enklitikama (modul *rank*). Slika 93 daje primer zaglavlja koje smo definisali za srpski jezik. Dodatne opise, obeležene znakom „%”, dajemo unutar same slike.

```
% pozivanje modula - boje, rang i redosled enklitika
use color with () dims (syn)
use rank with () dims (syn)
use unicity with (rank=1) dims (syn)
use unicity with (rank=2) dims (syn)
use unicity with (rank=3) dims (syn)
use unicity with (rank=4) dims (syn)
use unicity with (rank=5) dims (syn)

% definisanje tipova:
type CAT={S, VP, V, N, NP, P, PP, ADV, ADVP, Cl, PRO, C, AUX, NUM, NUMP, A, AP, PAR,
NEG, da} % 1) nabranjanjem vrednosti
type BOOL={+,-}
type MARK={flex,subst,foot,nadj,anchor,coanchor}
type CASE={1,2,3,5,6,7}
type RANK=[1 .. 5] % 2) davanjem intervala celobrojnih vrednosti
type PERSON={x,y,z}
type GENDER={m,f,n,MG,NG,FG}
type NUMBER={s,p,PL}
type FORM={P,R,F,A,I,C,L,D}
type COLOR={red,black,white}
type FUNC={sub,dir,indir,lsub,pdop}
type ASPECT={perf,imperf}
type MOD={W,I,Y,G,T,S,X}
type NAME={se}
type ATOMIC=[ % 3) davanjem strukturisane definicije
    form : FORM,
    num : NUMBER,
    gen : GENDER,
    pers : PERSON,
    ref : BOOL,
    cop : BOOL,
    func : FUNC,
    case : CASE,
    wh : BOOL,
    neg : BOOL,
    aux : BOOL,
```

```

    pasiv : BOOL,
    mod : MOD,
    subg : GENDER,
    subp : PERSON,
    subn : NUMBER,
    def : BOOL,
    asp : ASPECT
]

% definicije svojstava
property mark : MARK
property rank : RANK {
    i_ = 1,
    ii_ = 2,
    iii_ = 3,
    iv_ = 4,
    v_ = 5}
property color : COLOR

% definicije obeležja
feature top : ATOMIC
feature bot : ATOMIC

```

Slika 93 Zaglavlje metagramatike srpskog jezika

2) Unutar fajla sa konkretnim klasama svaka klasa se definiše ključnom rečju *class* i jedinstvenim imenom klase. Informacije koje se nalaze u telu klase čine *Sadržaj klase*¹⁶³. Nasleđivanje i razmena informacija među klasama omogućena je kroz stavke *import* – koja uključuje drugu klasu ili klase, i čiji se eksportovani čvorovi mogu nesmetano koristiti u ovoj klasi. Pod stavkom *export* navode se čvorovi koji će biti vidljivi kada se u nekoj drugoj klasi pozove ova klasa. Pod stavkom *declare* navode se svi čvorovi koji se po prvi put pominju u datoj klasi. Slika 94 daje izgled nekoliko klasa koje smo definisali u metagramatici srpskog jezika, sa pojašnjenjem simbola koji se u tom opisu koriste. Objašnjenja su i ovog puta data iza znaka „%”.

¹⁶³ Definicija sadržaja klase data je u poglavlju 2.5.1.2

```

class GlagolskiArgument % ime klase
export
    xS xVP fW % promenljive koje će biti vidljive pri uvozu ove klase. x
    stoji ispred čvorova, a f ispred obeležja.
declare
    ?xS ?xVP ?fW % znak „?“ uvodi promenljive
{
    <syn> % opis se odnosi na sintaksičku dimenziju
    {
        node xS (color = white) [cat = S, bot = [wh = ?fW]]{ % svojstva
        se navode u oblim zagrada, a obeležja u uglastim.
            node xVP (color = white) [cat = vp]
        }
    }
}

class PraviObjekatRealizovanN
import
    NeSubjekat[] % uvezena klasa
{
    <syn>{
        node xX1 (mark=subst) [cat=n, top=[func=dir, case=@{2,4}, wh = -]]
        % @ označava disjunkciju vrednosti.
    }
}

class SlaganjeSubjekatPredikat % klasa koja uređuje kongruenciju
export
    xSubKon      xVPKon
declare
    ?xSubKon ?xVPKon ?fN ?fP ?fG
{
    <syn>{
        node xSubKon [top=[num=?fN, pers=?fP, gen = fG]];
        node xVPKon [top=[num=?fN, pers=?fP, gen = fG]]
    }
}

```

```

class SubjekatRealizovan
import
    GlagolskiArgument[]
    SlaganjeSubjekatPredikat[]
export
    xSub
declare
    ?xSub
{
    <syn>{
        node xS {
            node xSub (color=red, mark=subst) [top=[func=sub, case=1,
wh = fW]]
            node xVP
            };
        xSub = xSubKon; % mapiranje čvorova
        xVP = xVPKon
    }
}

```

Slika 94 Definicije nekoliko klasa iz metagramatike za srpski jezik

Zagrade oko čvorova, kao i određen broj drugih simbola, određuju njihov hijerarhijski i linearni odnos. Tabela 13 daje pregled tih simbola i zagrada.

Tabela 13 Simboli za definisanje odnosa među čvorovima unutar XMG koda

Značenje	Simbol	Alternativni prikaz
nespecificovan, nabranje	;	;
neposredna nadređenost	->	čvor { čvor }
nadređenost	->+	čvor { ...+čvor }
nadređenost ili jednakost	->*	čvor { ...čvor }
neposredno prethođenje	>>	čvor čvor
prethođenje	>>+	čvor „,+čvor
prethođenje ili jednakost	>>*	čvor „,čvor
jednakost	=	=

	<pre>class Primer1 declare ?xS ?xN ?xV { <syn>{ node xS; node xN; node xV; xS -> xN; xS -> xV; xN >> xV } }</pre>	<pre>class Primer2 declare ?xS ?xN ?xV { <syn>{ node xS { node xN node xV } } }</pre>
--	---	---

- 1) Navođenjem ključne reči *value* (vrednost) i imena određene klase, tražimo valuaciju¹⁶⁴ date klase. Ukoliko je klasa neispravno formirana, kompilator će objaviti poruku o grešci, ali ukoliko je klasa ispravno formirana, XMG kompilator će je obraditi i kao izlaz proizvesti metagramatiku u formi .xml dokumenta koji sadrži svaki klasu čija je valuacija tražena. Slika 95 predstavlja primer poziva valuacije za dve porodice stabala – obe sa neprelaznim glagolom, ali u prvom slučaju sa glagolom koji je nerefleksivan (pomoću koje se prepoznaje rečenica npr. *Marija spava.*), a u drugom sa glagolom koji je refleksivan (*Marija se smeje.*).

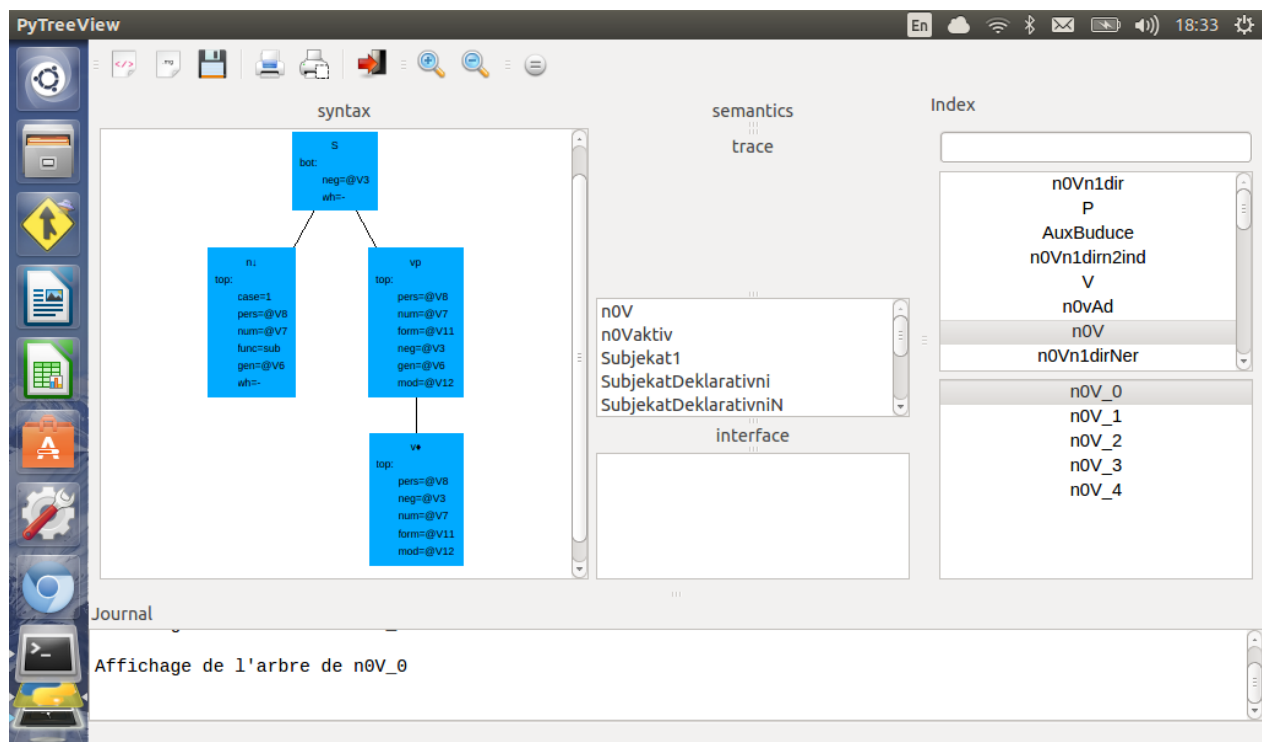
```
value n0V
value n0ClV
```

Slika 95 Primer poziva valuacija u XMG kodu

Ovaj dokument je ulaz za XMG grafički interfejs u kome se može videti izgled svih stabala koja ulaze u sastav date klase (Slika 96)¹⁶⁵. Stabla koja generiše XMG nisu leksikalizovana, već predstavljaju samo sheme stabala, kako se može i videti na datoj slici.

¹⁶⁴ engl. *valuation*, određivanje vrednosti

¹⁶⁵ Za detalje rada kompilatora upućujemo na (Petitjean, 2013)



Slika 96 Grafički korisnički interfejs za XMG

4.3 Generalizovanje rečeničnih segmenata u klase i organizovanje klasa u funkcije

Da bismo olakšali definisanje osnovnih klasa koje opisuju rečenične segmente, formirali smo za njih određene obrasce. Ovi obrasci se nazivaju apstraktnim klasama, i predstavljaju najviši nivo generalizacije rečeničnih struktura, tj. struktura stabala. Jedna apstraktna klasa propisuje sve čvorove i sva obeležja koje sve klase koje je nasleđuju moraju imati te zato retko sadrži detaljnije opise čvorova. U našoj metagramatici postoje dve takve opšte klase: jedna za glagole, pod nazivom *GlagolskiPredikat*, a druga za sve argumente glagola, pod nazivom *GlagolskiArgument*. Ove dve klase će neposredno ili posredno nasleđivati sve ostale klase u našoj metagramatici, te ćemo ih predstaviti pre svih ostalih klasa.

Prilikom predstavljanja svih klasa, čvorove i obeležja definišemo prateći konvenciju postavljenu u metagramatici za francuski, izrađenoj po principima izloženim u (Crabbé, 2005)¹⁶⁶. Po toj konvenciji, ispred imena čvorova stabala stavljamo oznaku "x", a ispred vrednosti obeležja oznaku "f". Tako će čvor pod imenom S (rečenica), u metagramatici biti označen kao xS, dok ćemo promenljivu kojom označavamo vrednosti obeležja, recimo num (broj), predstavljati kao fN¹⁶⁷. Ova konvencija je uvedena kako bi i za imena klasa i za imena obeležja mogle da se koriste iste oznake, a koje bi se onda razlikovale po pomenutom prefiksu.

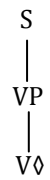
Uz svaku definiciju klase navodimo i neleksikalizovano stablo SrpTAG koje data klasa generiše. Poređenjem opisa klase i stabla, može se primetiti da nazivi čvorova nisu isti. Naime, nazivi čvorova u metagramatici su lokalni i definišu se nezavisno od imena čvorova u stablima. Nazivi čvorova u stablima definišu se kao vrednost kategorije cat unutar metagramatičkog opisa svakog čvora. Uz čvorove shema stabala nećemo navoditi obeležja, zbog nedostatka prostora, a i zato što se mogu pročitati iz samog metagramatičkog opisa.

¹⁶⁶ Ta metagramatika se može naći na sledećoj adresi:
<http://subversion.renater.fr/xmg/trunk/METAGRAMMARS/FrenchTAG/>.

¹⁶⁷ Promenljiva var koju smo koristili u prethodnim poglavljima je ekvivalent ovakvim vrednostima koje počinju slovom „f“.

4.3.1 Glagoli

Kao polaznu tačku u definisanju osnovnih klasa za glagole navodimo njihovu apstraktnu klasu – klasu `GlagolskiPredikat` (Slika 97). Ova klasa definiše osnovu svake glagolske klase, što su tri čvora: čvor samog glagola, definisan kao sidro, glagolska fraza, i rečenični čvor kao koreni čvor ove klase (*anchor*). Ako se pogleda reprezentacija ove klase u obliku stabla na istoj slici, biće jasno zašto se može reći da ova klasa definiše kičmu stabala – ovakvu spinalnu strukturu ćemo videti u istom obliku u svakom stablu unutar `SrpTAG`.

<pre>class GlagolskiPredikat export xS xVP xV declare ?xS ?xVP ?xV ?fNeg { <syn>{ node xS (color = black) [cat = S] [bot = [neg = fNeg]] { node xVP (color = black) [cat = VP] [top = [neg = fNeg]]{ node xV (mark = anchor, color = black) [cat = V] [top = [neg = fNeg]] } } } }</pre>	
--	---

Slika 97 Apstraktna klasa za glagole

Sva tri čvora ove klase definisana su kao crna¹⁶⁸, što znači da se mogu sjediniti sa drugim (crnim ili belim) čvorovima, ali mogu ostati i takvi kakvi jesu, što će biti slučaj kod, na primer, bezličnih neprelaznih glagola.

Kako je u pitanju visoko apstraktna klasa, za svaki od ovih čvorova smo definisali svega jedno obeležje – obeležje `neg`, kojim ćemo označiti da li je u pitanju pozitivan ili negativan glagol, odnosno čitava struktura. Ista vrednost ovog obeležja na svim čvorovima, definisana kao `fNeg`, omogućava prenošenje iste vrednosti sa sidra na ostale čvorove – ukoliko je glagol u leksikonu definisan kao negativan (npr. glagol *nemati*). U slučaju kada se negacija unosi u strukturu pripajanjem, obeležje negativnosti se prenosi sa korenog čvora `S` na niže čvorove (`VP` i `V`)¹⁶⁹. Nijedno od preostalih obeležja nije opšte za sve glagolske klase, te ćemo ih zasebno definisati u klasama koje nasleđuju ovu klasu.

¹⁶⁸ Za opis sistema boja upućujemo na poglavlje 2.5.1.4.

¹⁶⁹ Pri pripajanju negacije čvor `V` zapravo treba i dalje da nosi negativnu vrednost obeležja negativnosti, dok čvorovi `S` i `VP` nose pozitivnu. Iz tog razloga tehnički ovakva jednaka distribucija vrednosti ovog obeležja nije

Ovu klasu nasleđuju tri ključne klase glagolskih predikata: AktivniGlagol, PasivniGlagol i Refleksivnost.

Klasa AktivniGlagol je polazna klasa za glagole aktivne dijateze. U definiciji klase (Slika 98) vidi se da nasleđuje klasu GlagolskiPredikat preko metode *import*. Kada se na ovaj način nasledi klasa, u sadržaju same klase naslednika dovoljno je pomenuti samo one čvorove iz originalne klase čija se vrednosti bliže definiše. U ovom slučaju, to su čvorovi VP i V. Za ove čvorove se definišu i obeležja osnovnih kongruentnih kategorija – broja, roda i lica, kao i obeležja form, u kome se navodi glagolsko vreme datog glagola, i mod, u kome se navodi oblik glagola.

<pre>class AktivniGlagol import GlagolskiPredikat[] declare ?fM ?fF ?fN ?fP ?fG { <syn>{ node xVP [top = [mod = fM, form = fF, num = fN, pers = fP, gen = fG]] { node xV [top = [mod = fM, form = fF, num = fN, pers = fP, gen = fG]] } } }</pre>	<pre>S VP V∅</pre>
---	------------------------

Slika 98 Klasa za glagole aktivne dijateze

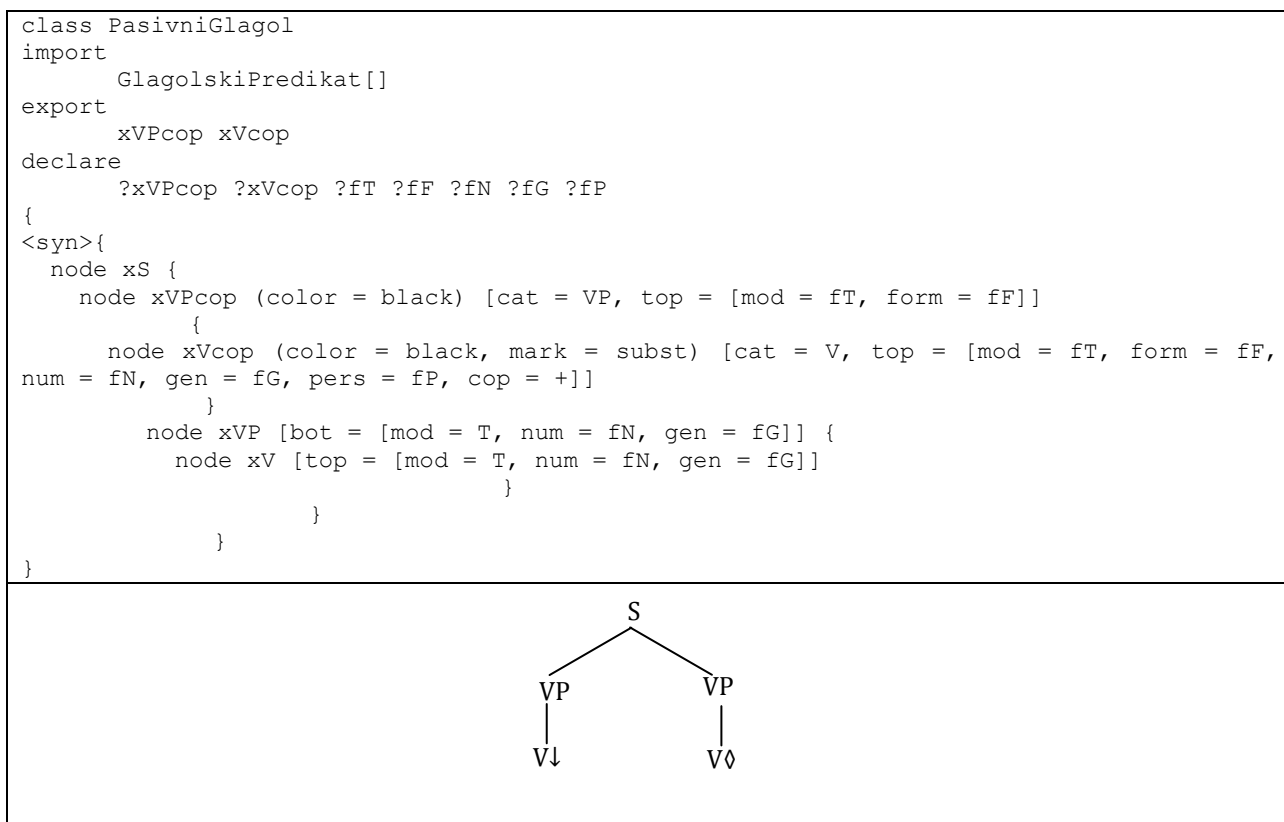
Klasu BezlicniAktivniGlagol (Slika 99) definišemo za bezlične rečenične porodice. Od polazne klase AktivniGlagol razlikuje se po dodeljenim vrednostima kategorija roda, broja i lica.

<pre>class BezlicniAktivniGlagol import AktivniGlagol[] { <syn>{ node xVP [top = [num = s, pers = z, gen = n]] { node xV [top = [num = s, pers = z, gen = n]] } } }</pre>	<pre>S VP V∅</pre>
---	------------------------

Slika 99 Klasa za bezlične glagole aktivne dijateze

tačna za slučaj pripajanja negacije, ali zbog prirode procesa pripajanja prilikom parsiranja, ona ipak funkcioniše. Za pripajanje prilikom parsiranja pogledati poglavlje 4.5.

Klasa PasivniGlagol koristi se za sve glagole u pasivnom obliku (Slika 100). I ova klasa nasleđuje klasu GlagolskiPredikat, pritom je proširujući za dva nova čvora – čvorove glagola *jesam/bit*i koji čine sastavni deo pasivne konstrukcije. Novi čvorovi se unose u već postojeću glagolsku strukturu definisanjem njihovog odnosa prema njima nadređenom čvoru (S), sestrijskom čvoru (VP), kao i definisanjem međusobnog odnosa samih novih čvorova (VPCop i Vcop).



Slika 100 Klasa za glagole pasivne dijateze

Klasu Refleksivnost (Slika 101) koristimo kao apstraktnu klasu za sve glagole s refleksivnom strukturom (inherentno refleksivne, nastale refleksivizacijom, u refleksivnom pasivu i obezličene). I ova klasa nasleđuje klasu GlagolskiPredikat, u čiju strukturu unosi novi čvor, ovog puta čvor refleksivne rečice *se*, koji nosi pozitivnu vrednost obeležja *ref*. U ovoj klasi nije definisana priroda tog čvora – da li se unosi zamenom ili je sidro, i to će biti definisano u konkretnim klasama koje je nasleđuju. Obeležje *rank* ukazuje na rang ili redosled koji bi automatski zauzela ova eklitika ukoliko bi se pored nje našle druge

enklitike (zameničke i glagolske). Kao vrednost obeležja name daje se interni naziv čvora, koji se koristi da bi se na dati čvor referisalo unutar leksikona. Ovo ime predstavlja id čvora i neophodno je da bi se dati čvor definisao kao dodatno sidro (Slika 183 daje takav primer).

```

class Refleksivnost
import
  GlagolskiPredikat[]
export
  ?xCl
declare
  ?xCl ?fN ?fP ?fF ?fG
{
  <syn>{
    node xVP [bot = [num = fN, pers = fP, gen = fG]];
    node xCl (color = red, rank = 5, name = se) [cat = Cl, top = [ref = +]];

    node xV [top = [num = fN, pers = fP, gen = fG]];

    xVP -> xCl;
    xVP -> xV;
  }
}

```

Slika 101 Klasa za refleksivnu strukturu

Za razliku od prethodno navedenih klasa, kod kojih smo pomoću zagrada definisali odnos među čvorovima, u ovoj klasi (Slika 101) najpre samo navodimo njene čvorove, ne definišući odnos među njima, a zatim navodimo samo da se i čvor Cl i čvor V nalaze direktno pod čvorom VP, ne definišući odnos između Cl i V. Na ovaj način ostavljamo prostor za definisanje tog odnosa u klasama koje će naslediti ovu. Slika 102 prikazuje jednu od tih klasa, u kojoj je poslednjim redom opisa definisano da čvor V prethodi čvoru Cl. Ovakvo stablo smo definisali za primere refleksivnih konstrukcija u kojima je subjekat nerealizovan. Ovo je istovremeno i klasa koju koristimo za inherentno refleksivne glagole, što se može videti iz definicije čvora CL kao dodatnog sidra (*coanchor*). Za stabla u kojima se čvor Cl nalazi sa leve strane glagola definisali smo klasu RefleksivniGlagolLevi, koja se od navedene

klase razlikuje samo po redosledu čvorova u poslednjem redu opisa ($xCl \gg+ xV$), te je iz tog razloga nećemo posebno navoditi.

<pre>class RefleksivniGlagolDesni import Refleksivnost[] { <syn>{ node xVP; node xCl (mark = coanchor); node xV; xVP -> xCl; xVP -> xV; xV >>+ xCl } }</pre>	
---	---

Slika 102 Klasa desno reflektivne strukture

Ove dve klase se pomoću disjunkcije (ovde označene kao „|”) udružuju u klasu Refleksivni (Slika 103).

```
class Refleksivni
{ RefleksivniGlagolLevi[] | RefleksivniGlagolDesni[] }
```

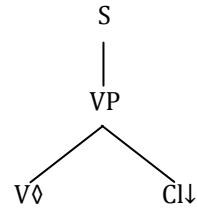
Slika 103 Klasa za reflektivne glagole gde je rečca se dodatno sidro

Za potrebe rečenica s obezličenom i bezličnom konstrukcijom, u kojima nema subjekta pa tako ni kongruencije s njim, definišemo klasu BezlicniDesni (Slika 104), koja uvozi klasu RefleksivniGlagolDesni, a zatim dodatno definiše kategorije roda, broja i lica na glagolu. Samo stablo je isto kao u pomenutoj klasi, pa ga ovde nećemo ponavljati. Za potrebe obezličениh i bezličnih konstrukcija sa logičkim subjektom (*Marku se pomaže.*) definisali smo klasu BezlicniLevi, koja se od klase BezlicniDesni razlikuje po tome što uvozi klasu RefleksivniGlagolLevi. Ovu klasu nećemo posebno navoditi ovde.

```
class BezlicniDesni
import
  RefleksivniGlagolDesni[]
{
  <syn>{
    node xCl (name = se);
    node xV [top = [pers = z, num = s, gen = n]];
  }
}
```

Slika 104 Klasa za obezličenu glagolsku konstrukciju

Postojala je potreba da definišemo zasebne refleksivne klase za konstrukcije sa refleksivnim glagolom u koje se rečca se unosi zamenom (stabla dobijena refleksivizacijom). Slika 105 prikazuje takvo stablo, u kome se rečca raspoređuje desno od glagola. I ovde smo definisali klasu u kojoj se rečca raspoređuje s leve strane, ali kako nema posebnih razlika u odnosu na ovu klasu, ne navodimo je posebno.

<pre>class RefleksivniGlagolDesniSubst import Refleksivnost[] { <syn>{ node xVP; node xCl (mark = subst); node xV; xVP -> xCl; xVP -> xV; xV >>+ xCl } }</pre>	 <pre> graph TD S --> VP VP --> V[V] VP --> Cl[Cl] style V stroke-dasharray: 5 5 style Cl stroke-dasharray: 5 5 </pre>
---	--

Slika 105 Klasa desno refleksivne strukture za čvorom za zamenom

Ove dve klase se udružuju u klasu RefleksivniSubst (Slika 106).

<pre>class RefleksivniSubst { RefleksivniGlagolLeviSubst[] RefleksivniGlagolDesniSubst[] }</pre>
--

Slika 106 Klasa za refleksivne glagole gde se rečca se unosi zamenom

Kopulativni predikat

Segment stabla kopulativnog predikata ima istu strukturu kao i segment pasivnog glagola (Slika 100), te smo mogli da za ove dve klase definišemo odgovarajuću apstraktnu klasu koju bi klase PasivniGlagol i KopulativniPredikat zatim nasleđivale. Ipak, za potrebe metagramatike u ovom trenutku rešili smo da definišemo samo terminalne neglagolske čvorove, kako trenutno ne obrađujemo modifikatore sintagmi. Zato smo samo za kopulativni predikat definisali apstraktnu klasu koja ima sidro – predikativ – direktno ispod čvora S (Slika 107). Ovu klasu dalje nasleđuju sve klase kopulativnih glagola – imenički predikat (Slika 108), pridevski predikat (Slika 110) i priloški predikat (Slika 111).

Može se primetiti da smo za kopulu definisali i obeležje roda, koje ona nema u slučaju prostih glagolskih oblika. Kako subjekat ove konstrukcije definišemo odvojeno, bilo je potrebno na neki način usloviti to da npr. pridevski predikativ bude istog roda kao i subjekat, kako bismo izbegli da se kao tačne prepoznaju rečenice tipa: *Marko je pametna*. Ovo obeležje na kopuli ima dakle samo povezuvačku vrednost, odnosno služi za uspostavljanje kongruencije između subjekta i predikativa.

```

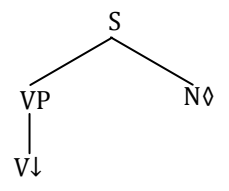
class KopulativniPredikat
export
    xS xPred xVP xV
declare
    ?xS ?xPred ?xVP ?xV ?fN ?fP ?fG ?fW

{
<syn>{
    node xS (color = black) [cat = S, bot = [mod = I, wh = fW]]{
        node xVP (color = black) [cat = VP, top = [cop = +, num = fN, pers = fP, gen = fG]] {
            node xV (color = black, mark = subst) [cat = V, top = [cop = +, num = fN, pers =
fP, gen = fG]]
            node xPred (color = black, mark = anchor) [top = [num = fN, pers = fP, gen =
fG]]
        }
    }
}
}

```

Slika 107 Apstraktna klasa za kopulativni predikat

Slika 106 prikazuje jednu od klasa imeničkog predikata – onu gde je imenički predikativ definisan kao imenica. U metagramatici imamo definisanu i klasu za imenički predikativ kao zamenicu – ImenickiPredikatPro. Osim što je morfološka kategorija ovog predikativa drugačija, od klase s imenicom kao predikativom razlikuje se po otvorenoj vrednosti obeležja pers, koje nije nužno u 3. licu (pers = fP). Klasu ImenickiPredikat tako definišemo kao disjunkciju ove dve klase (Slika 109).

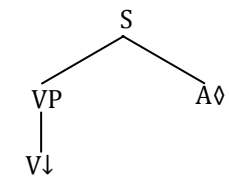
<pre>class ImenickiPredikatN import KopulativniPredikat[] { <syn>{ node xPred [cat = N, top = [case = 1, pers = z]] } }</pre>	
---	---

Slika 108 Klasa za imenički predikat u obliku imenice

<pre>class ImenickiPredikat { ImenickiPredikatN[] ImenickiPredikatPro[] }</pre>

Slika 109 Klasa za imenički predikat

Jedino karakteristično obeležje pridevskog predikativa (Slika 110) jeste obeležje određenosti (def), kako je pridevski predikativ uvek određenog vida.

<pre>class PridevskiPredikat import KopulativniPredikat[] { <syn>{ node xPred [cat = A, top = [case = 1, pers = z, def = +]] } }</pre>	
--	--

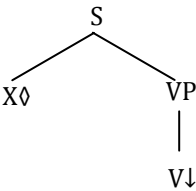
Slika 110 Klasa za pridevski predikat

Za priloški predikat subjekatsko-predikatskih rečenica trenutno imamo definisanu samo klasu koja na mestu predikativa prepoznaje prilog (Slika 111). Odluka da u ovom trenutku ne obrađujemo predikative kao fraze uticala je i na prepoznavanje priloških predikativa u obliku predložko-padežne konstrukcije i u obliku brojne sintagme. Da bismo mogli da prepoznamo ovakve fraze, i predikative kao fraze generalno, bilo bi potrebno da u klasi KopulativniPredikat definišemo predikativ kao frazu s nespecificovanim sidrom, i da zatim u nove dve klase – PriloskiPredikatPP i PriloskiPredikatNum – definišemo dodatni čvor za imenicu kao sestrinski čvor terminalnog čvora pod tom frazom (PP ili NUMP). U slučaju obrade predložkih fraza generalno, kao što smo već pomenuli u poglavlju 3.5.1, postoji dodatna poteškoća jer je potrebno odrediti da li će i koji još čvor pored predložkog čvora biti posmatran kao sidro.

<pre> class PriloskiPredikatAd import KopulativniPredikat[] { <syn>{ node xPred [cat = ADV] } } } </pre>	 <pre> graph TD S --> VP S --> ADV["ADVØ"] VP --> V["V↓"] </pre>
--	--

Slika 111 Klasa za priloški predikat

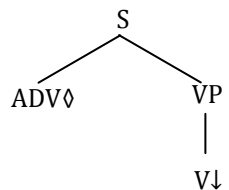
Za potrebe bezličnih rečenica s priloškim predikatom¹⁷⁰, bilo je potrebno da definišemo obrnut raspored kopule i predikativa, te smo definisali apstraktnu klasu koja ima drugačiji redosled glagolske fraze i samog predikativa (Slika 112).

<pre> class KopulativniPredikatLevi export xS xPred xVP xV declare ?xS ?xPred ?xVP ?xV ?fN ?fP ?fG { <syn>{ node xS (color = black) [cat = S, bot = [mod = I, wh = fW]]{ node xPred (color = black, mark = anchor) [top = [wh = fW]] node xVP (color = black) [cat = VP, top = [cop = +, num = fN, pers = fP, gen = fG]] { node xV (color = black, mark = subst) [cat = V, top = [cop = +, num = fN, pers = fP, gen = fG]] } } } } } } </pre>	 <pre> graph TD S --> X["XØ"] S --> VP VP --> V["V↓"] </pre>
---	---

Slika 112 Apstraktna klasa za levi kopulativni predikat

Slika 113 tako predstavlja klasu priloškog predikata za bezlične rečenice. Za iste potrebe smo definisali i klasu PriloskiPredikatAdDesni, koja uvozi klasu KopulativniPredikatDesni, i koju zatim koristimo za klasu četvrtog bezličnog rečeničnog modela (Slika 178). Nijednu od ove dve pomenute klase nećemo zasebno definisati ovde.

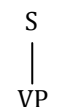
¹⁷⁰ Ali isto tako i za potrebe kopulativnih rečenica koje imaju izostavljen subjekat, iako nećemo ovde zasebno definisati takve slučajeve.

<pre> class PriloskiPredikatAdLevi import KopulativniPredikatLevi[] { <syn>{ node xPred [cat = ADV] node xVP [top = [num = s, pers = z, gen = n]] { node xV [top = [num = s, pers = z, gen = n]] } } } </pre>	 <pre> graph TD S --> ADV S --> VP VP --> V </pre>
---	--

Slika 113 Priloški predikat za bezlične rečenice

4.3.2 Glagolski argumenti

Apstraktna klasa koju nasleđuju svi glagolski argumenti jeste klasa GlagolskiArgument (Slika 114). Budući da je u pitanju najopštija klasa koja treba da predstavi bilo koji glagolski argument, ona definiše samo segment rečeničnog stabla koji služi da se argument spoji sa nekom od klasa za glagole – koreni čvor S i sestrički čvor datog argumenta, VP. Na to da ovakve klase ne mogu stajati same, odnosno da se moraju sjediniti sa nekim od naslednika klase GlagolskiPredikat govori njihova boja – bela. Jedina informacija koja se uz čvor S ovde još javlja jeste obeležje upitnosti (wh) na čvoru S, s opštom, nedefinisanom vrednošću.

<pre> class GlagolskiArgument export xS xVP fW declare ?xS ?xVP ?fW { <syn>{ node xS (color = white) [cat = S, bot = [wh = fW]]{ node xVP (color = white) [cat = VP] } } } </pre>	 <pre> graph TD S --> VP </pre>
---	---

Slika 114 Apstraktna klasa za sve glagolske argumente

4.3.2.1 Subjekat

Kongruencija

Pre nego što krenemo u definisanje subjekatskih klasa, potrebno je da definišemo još jednu apstraktnu klasu kojom ćemo obezbediti slaganje između subjekta i predikata u odgovarajućim kategorijama (rod, broj i padež). Slika 115 prikazuje tu klasu. Čvorovi

SubKon i VPKon označavaju čvor Subjekta i čvor VP, za koje se definišu obeležja sa istim vrednostima. Da bi se ova klasa stavila u upotrebu, uvozimo je pri definisanju subjekatskih klasa i tom prilikom vršimo izjednačavanje čvorova (Slika 117).

```
class SlaganjeSubjekatPredikat
export
    xSubKon xVPKon
declare
    ?xSubKon ?xVPKon ?fN ?fP ?fG
{
    <syn>{
        node xSubKon [top = [num = fN, pers = fP, gen = fG]];
        node xVPKon [top = [num = fN, pers = fP, gen = fG]]
    }
}
```

Slika 115 Klasa za slaganje subjekta i predikata

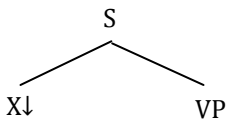
Dve apstraktne subjekatske klase su klase za nerealizovan subjekat – SubjekatNerealizovan (Slika 116, primer 1) i za realizovan subjekat – SubjekatRealizovan (Slika 117). Klasa SubjekatNerealizovan zapravo neće sadržati nijedan čvor osim čvorova klase GlagolskiArgument. Iako se čvor nerealizovanog subjekta javlja u stablima u SrpTAG, u stablima koja trenutno generiše metagramatika za njih nećemo definisati poseban čvor. Razlog za to je činjenica da ova metagramatika generiše strukture koje se koriste kao ulaz za parser, i da ispravno parsiranje ne bi bilo moguće ukoliko bi se u stablu nalazio i prazan čvor bilo kog tipa. Ovaj opis ipak u svakom trenutku možemo promeniti tako da generiše stabla sa praznim čvorom, ukoliko želimo da metagramatika zaista generiše samo originalna FBLTAG stabla.

Klasu SubjekatNerealizovan nasleđuje klasa SubjekatImperativ u kojoj se glagol definiše kao glagol u imperativu (Slika 116, primer 2). Klasa na slici koristi se za obradu imperativa u drugom licu jednine i množine.

<p>1.</p> <pre>class SubjekatNerealizovan import GlagolskiArgument[]</pre>	<p>2.</p> <pre>class SubjekatImperativ import SubjekatNerealizovan[] { <syn>{ node xVP [top = [mod = Y, pers = 2, num = @{s,p}]] } }</pre>	<p style="text-align: center;">S VP</p>
--	--	---

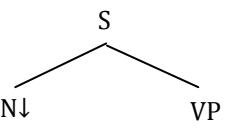
Slika 116 Apstraktna klasa za nerealizovan subjekat (primer 1) i njena implementacija u obliku subjekta predikata u imperativu

U osnovnu klasu za realizovan subjekat (Slika 117) uvozimo i ranije navedenu klasu za kongruenciju. U poslednja dva reda klase mogu se videti jednačine u kojima se navodi da je čvor za subjekat definisan u ovoj klasi (xSub) identičan subjekatskom čvoru definisanom u klasi SlaganjeSubjekatPredikat – xSubKon. Isto to definišemo i za čvorove VP unutar klase za realizovan subjekat i čvora xVPKon iz klase za kongruenciju između subjekta i predikata. Na ovaj način smo za sve klase koje nasleđuju datu klasu za realizovan subjekat obezbedili da se sa predikatom slaže u licu, broju i rodu.

<pre> class SubjekatRealizovan import GlagolskiArgument[] SlaganjeSubjekatPredikat[] export xSub declare ?xSub { <syn>{ node xS { node xSub (color = red, mark = subst) [top = [func = sub, case=1, wh = fW]] node xVP }; xSub = xSubKon; xVP = xVPKon } } </pre>	 <pre> graph TD S --- X["X↓"] S --- VP </pre>
---	--

Slika 117 Apstraktna klasa za realizovan subjekat

Dve konkretne klase koje nasleđuju ovu klasu su: klasa za imenički subjekat, koju smo nazvali SubjekatDeklarativniN (Slika 118) i zamenički subjekat – SubjekatDeklarativniPro, koju nećemo ovde navoditi zasebno kako se od imeničke klase razlikuje samo po vrednosti obeležja cat. Ove dve klase disjunkcijom spajamo u klasu SubjekatDeklarativni (Slika 119, primer 1). Ova klasa stoji nasuprot klasi SubjekatUpitni (ista slika, primer 2), koju smo definisali za subjekat u obliku upitne zamenice.

<pre> class SubjekatDeklarativniN import SubjekatRealizovan[] { <syn>{ node xSub [cat = N, top = [wh = -]] } } </pre>	 <pre> graph TD S --- N["N↓"] S --- VP </pre>
---	--

Slika 118 Klasa za imenički subjekat

Ako se pogledaju unazad klase za RealizovanSubjekat (Slika 117) i klasa za GlagolskiArgument (Slika 114), može se videti da je vrednost obeležja upitnosti za čvorove xS i xSub postavljena tako da ima istu vrednost. Zahvaljujući tome će se pozitivna vrednost obeležja upitnosti sa čvora xSub u klasi SubjekatUpitni preneti i na čvor S, te će cela rečenica biti obeležna kao upitna. Isto tako, ukoliko je subjekat deklarativan, čvor S će imati negativnu vrednost obeležja upitnosti, prenetu sa čvora xSub iz klase unutar klase SubjekatDeklarativni.

1.	2.
<pre>class SubjekatDeklarativni { SubjekatDeklarativniN[] SubjekatDeklarativniPro[] }</pre>	<pre>class SubjekatUpitni import SubjekatRealizovan[] { <syn>{ node xSub [cat = PRO, top = [wh = +]] } }</pre>

Slika 119 Klase za deklarativni subjekat (1) i upitni subjekat (2)

Spajanjem dve navedene klase (Slika 119) ostvarujemo drugi nivo apstrakcije u metagramatici – udružujemo realizacije određene funkcije pod zajedničkim nazivom – Subjekat (Slika 120). Ovo je univerzalna klasa koju koristimo za refleksivne strukture u kojima subjekat mora biti prisutan da bi se generisao pravilan redosled reči. Za subjekat nerefleksivnih konstrukcija definisali smo i klasu Subjekat1, koju ne navodimo posebno ovde, a koja pored date dve klase sadrži i klasu SubjekatNerealizovan.

<pre>class Subjekat { SubjekatDeklarativni[] SubjekatUpitni[] }</pre>

Slika 120 Klasa za subjekat

Za potrebe modela s logičkim subjektom, definisali smo i klasu za deklarativni subjekat koji se slobodno raspoređuje u odnosu na glagol. Takav subjekat smo nazvali SubjekatDeklarativniSlobodan (Slika 121) i probno ga definisali samo za imenice. Takva klasa, naravno, može da postoji i za zamenice.

<pre> class SubjekatDeklarativniSlobodan import GlagolskiArgument[] SlaganjeSubjekatPredikat[] export xSub declare ?xSub { <syn>{ node xS; node xSub (color = red, mark = subst) [cat = N, top = [func = sub, case = 1, wh = fW]]; node xVP; xS -> xSub; xS -> xVP; xSub = xSubKon; xVP = xVPKon } } </pre>	
--	--

Slika 121 Klasa za nefiksirani imenički deklarativni subjekat

Ova klasa, zajedno sa klasama SubjekatUpitni i SubjekatImperativ čini klasu SubjekatSlobodan (Slika 122), koju ćemo dalje koristiti u modelima s logičkim subjektom (Slika 170).

<pre> class SubjekatSlobodan { SubjekatDeklarativniSlobodan[] SubjekatUpitni[] SubjekatImperativ[] } </pre>

Slika 122 Klasa za nefiksirani subjekat

U metagramatici trenutno imamo dva tipa rečeničnih subjekata – izrični i infinitivni. Slika 123 daje opis i strukturu izričnog subjekta. Čvor komplementizatora xC nosi oznaku flex, što znači da je u stablo direktno uneta leksema – u ovom slučaju komplementizator *da*¹⁷¹. Vrednost obeležja mark na čvoru xS00 je foot, čime se označava da je u pitanju priključni čvor.

¹⁷¹ Identično izgleda i stablo s komplementizatorom *što*, u kome je vrednost obeležja cat na čvoru xC sto.

<pre> class IzricniSubjekat import GlagolskiArgument[] declare ?xS0 ?xC ?xS00 { <syn>{ node xS{ node xS0 (color = red) [cat = S, top = [func = sub]] { node xC (color = red, mark = flex) [cat = da] node xS00 (color = red, mark = foot) [cat = S, top = [func = sub, mod = I, wh = -]] } node xVP } } } </pre>	<pre> graph TD S --> S0 S --> VP S0 --> C S0 --> S0_star[S0*] C --> da </pre>
--	--

Slika 123 Klasa za izrični subjekat

Slika 124 prikazuje klasu za infinitivni subjekat.

<pre> class SubjekatInfinitiv import SubjekatNerealizovan[] declare ?xS0 { <syn>{ node xS{ node xS0 (color = red, mark = subst) [cat = S, top = [func = sub, mod = W, wh = -]] node xVP } } } </pre>	<pre> graph TD S --> S_down[S↓] S --> VP </pre>
--	---

Slika 124 Klasa za infinitivni subjekat

Ove dve rečenične klase, povezane disjunkcijom, daju natklasu RecenicniSubjekat (Slika 125).

<pre> class RecenicniSubjekat { SubjekatInfinitiv[] IzricniSubjekat[] } </pre>
--

Slika 125 Klasa za rečenični subjekat

Klase Subjekat i RecenicniSubjekat su klase koje ćemo nadalje koristiti kao predstavnike ove funkcije u složenijim klasama.

4.3.2.2 *Nesubjekatski argumenti*

Za nesubjekatske rečenične argumente definisali smo jednu centralnu apstraktnu klasu pod imenom NeSubjekat (Slika 126). Za razliku od klase Subjekat (Slika 120), koja propisuje javljanje subjekatskog argumenta s leve strane glagola, ova klasa propisuje poziciono javljanje nesubjekatskih argumenata s desne strane glagola. U ovoj klasi imamo definisana dva obeležja – obeležje funkcije, koje ostaje otvoreno za definisanje u klasama koje je nasleđuju, i obeležje upitnosti. Čvor nesubjekatskog argumenta smo ovog puta odredili kao čvor koji se nalazi desno od čvora VP, ali ne neposredno do njega, kako bismo omogućili prepoznavanje većeg broja redosleda među nesubjekatskim argumentima.

<pre>class NeSubjekat import GlagolskiArgument[] export ?xX1 declare ?xX1 ?fF ?fW { <syn>{ node xS{ node xVP ,,,+node xX1 (color=red) [top = [wh = fW, func = fF]] } } }</pre>	<pre>graph TD S --- VP S --- X</pre>
--	--

Slika 126 Apstraktna klasa za nesubjekatske argumente

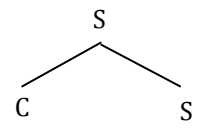
Tri apstrakne klase, za svaki od već pomenuta tri tipa nesubjekatskih argumenata, definisaćemo u narednim tabelama.

Za argumente u obliku predložko-padežne konstrukcije definisali smo klasu PP (Slika 127).

<pre>class PP import NeSubjekat[] export xPREP xX declare ?xPREP ?xX ?fK { <syn>{ node xX1 [cat = PP, top = [wh = -]]{ node xPREP (color = red) [top = [case = fK]] node xX (color = red, mark = subst) [top = [case = fK, gen = fG, num = fN, pers = fP]] } } }</pre>	<pre>graph TD PP --- PREP PP --- Xd[X↓]</pre>
--	---

Slika 127 Apstraktna klasa za predložke fraze

Za rečenične konstituente, konkretnije, konstituente u obliku izrične rečenice, definisali smo klasu RecenicniNeSubjekatKomp (Slika 128), koja definiše opšte karakteristike nesubjekatskog rečeničnog argumenta – koji sadrži komplementizator.

<pre>class RecenicniNeSubjekatKomp import NeSubjekat[] export ?xC ?xXX1 declare ?xC ?xXX1 ?fX { <syn>{ node xX1 [cat = S, top = [mod = I]] { node xC (color = red, mark = flex) [top = [wh = fW]] node xXX1 (color = red) [cat = S, top = [mod = I, wh = fX]] } } }</pre>	 <pre> graph TD S --> C S --> S2[S] </pre>
---	---

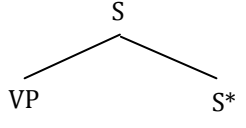
Slika 128 Apstraktna klasa za izrične rečenice u funkciji dopune

Rečenični konstituenti koji ne sadrže komplementizator su oni u obliku infinitiva, i za njih smo definisali zasebnu klasu. Za potrebe definisanja ove klase bilo je, ipak, potrebno definisati tri odvojene klase. Dve od tih klasa koristimo da definišemo subjekatsku kontrolu – slaganje između subjekta glavne rečenice i glagola zavisne rečenice u infinitivu (Slika 129, primeri 1 i 2).

<p>1.</p> <pre>class Infinitiv export xKontrolisani declare ?xKontrolisani { <syn>{ node xKontrolisani (color = red) [top = [mod = W, wh = -]] } }</pre>	<p>2.</p> <pre>class SubjekatskaKontrola import Infinitiv[] export xVPKontroler declare ?xVPKontroler ?fN ?fG ?fP { <syn>{ node xKontrolisani [top = [subn = fN, subg = fG, subp = fP]]; node xVPKontroler (color = white) [top = [subn = fN, subg = fG, subp = fP]] } }</pre>
--	--

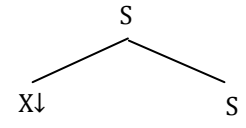
Slika 129 Klase za definisanje subjekatske kontrole

Apstraktna klasa koju će, pored klase SubjekatskaKontrola, nasledivati argumenti u obliku infinitiva jeste klasa ZavisneReceniceBezKomp (Slika 130).

<pre>class ZavisneReceniceBezKomp import NeSubjekat[] { <syn>{ node xX1 (color = red, mark = foot) [cat = S, top = [mod = W, wh = -, func = fF]] } }</pre>	
--	---

Slika 130 Klasa za zavisne rečenice bez komplementizatora

Još jedna apstraktna klasa koju definišemo jeste klasa Izmestanje, za potrebe izmeštenih argumenata u pitanjima. Ova klasa direktno nasleđuje klasu GlagolskiArgument. Čvorovi xSizm i xIzm su čvorovi koji se unose na već postojeći čvor S klase GlagolskiArgument. Čvor izmeštenog argumenta, xIzm, se, u zavisnosti od tipa dopune, može realizovati kao zamenica (u slučaju imeničke ili zameničke dopune), predložko-padežna konstrukcija (u slučaju nepravog objekta ili priloške dopune u ovom obliku) ili priloga (u slučaju priloške dopune), te će u samom stablu biti obeležen kao X (Slika 131).

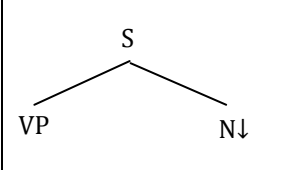
<pre>class Izmestanje import GlagolskiArgument[] export xSizm xIzm declare ?xSizm ?xIzm ?fF { <syn>{ node xSizm (color = red) [cat = S, bot = [wh = +]] { node xIzm (color = red, mark = subst) [top = [func = ?fF, wh = +]] node xS [top = [wh = -]] } } }</pre>	
---	---

Slika 131 Apstraktna klasa za izmeštene dopunske argumente

Pogledajmo sada kako se ovde definisane apstraktnne klase primenjuju na konkretne rečenične argumente.

Pravi objekat

Pravi objekat definišemo kao klasu koja direktno nasleđuje klasu NeSubjekat. Kao i kod subjekta, za pravi objekat razlikujemo dve osnovne klase – onu za realizovan i za nerealizovan pravi objekat. Slika 132 prikazuje klasu za realizovan pravi objekat u obliku imenice. Klasa u kojoj se pravi objekat realizuje kao zamenica nosi naziv `ObjekatRealizovanPro` i ne navodimo je ovde posebno kako se od klase na pomenutoj slici razlikuje samo po kategoriji. Ove dve klase spajamo u natklasu `ObjekatRealizovan` (Slika 133, primer 1), a klasu za nerealizovan subjekat možemo videti na istoj slici, u primeru 2. Klasu za nerealizovan objekat definišemo identično kao i klasu za nerealizovan subjekat (Slika 116, primer 1) – preko klase `GlagolskiArgument`, budući da ni ovde nećemo imati zaseban prazan čvor za nerealizovan pravi objekat.

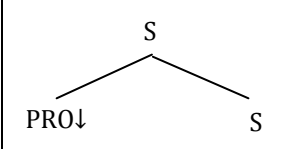
<pre>class PraviObjekatRealizovanN import NeSubjekat[] { <syn>{ node xX1 (mark = subst) [cat = N, top = [func = dir, case = @{2,4}, wh = -]] } }</pre>	
---	--

Slika 132 Klasa za imenički realizovan objekat

1.	2.
<pre>class PraviObjekatRealizovan { PraviObjekatRealizovanN[] PraviObjekatRealizovanPro[] } </pre>	<pre>class PraviObjekatNerealizovan import GlagolskiArgument[] </pre>

Slika 133 Klasa za realizovan pravi objekat (1) i nerealizovan pravi objekat (2)

Za tretman izmeštenog pravog objekta u pitanjima koristimo već definisanu apstraktnu klasu `Izmeštanje` (Slika 131). Klasu `Izmestanje` zatim nasleđuje klasa `IzmestenPraviObjekat` (Slika 134). Izmešteni čvor je u klasi `Izmestanje` već definisan kao upitan, tako da ćemo mu u ovoj klasi dodeliti samo kategoriju i funkciju.

<pre>class IzmestenPraviObjekat import Izmestanje[] { <syn>{ node xIzm [cat = PRO, top = [func = dir]] } }</pre>	
---	---

Slika 134 Klasa za izmešten pravi objekat

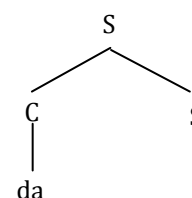
Realizovane tipove pravog objekta organizovaćemo u klasu PraviObjekat (Slika 135). Za potrebe stabala koja mogu ali ne moraju imati realizovan objekat definisaćemo još jednu klasu identičnog sadržaja, kojoj ćemo dodati i klasu PraviObjekatNerealizovan. Ovu klasu, koju nećemo zasebno predstavljati ovde, nazvali smo ObjekatPlusNerealizovan.

```
class PraviObjekat
{ PraviObjekatRealizovan[] | IzmestenPraviObjekat[] }
```

Slika 135 Klasa za pravi objekat

Za rečenični pravi objekat definišemo ukupno tri klase. Prva od njih je apstraktna i ima istu strukturu kao i apstraktna klasa koju nasleđuje, RecenicniNeSubjekatKomp, s tim što se za koreni čvor S (xX1) definiše njegova funkcija kao pravog objekta (Slika 136).

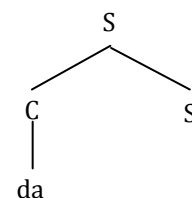
```
class ObjekatRecenicniKomp
import
  RecenicniNeSubjekatKomp[]
{
  <syn>{
    node xX1 [top = [func = dir]] {
      }
  }
}
```



Slika 136 Klasa za rečenični objekat s komplementizatorom

Sledeća klasa predstavlja deklarativnu izričnu rečenicu, te će S (xXX1) čvor unutar ove zavisne rečenice biti obeležen kao priključni (Slika 137).

```
class ObjekatRecenicniKompDeklarativni
import
  ObjekatRecenicniKomp[]
{
  <syn>{
    node xX1 [top = [wh = -]] {
      node xC [cat = da]
      node xXX1 (mark = foot) [top = [wh = -]]
    }
  }
}
```



Slika 137 Klasa za deklarativni rečenični objekat

Treća klasa predstavlja upitnu izričnu rečenicu (Slika 138). Tip komplementizatora ovde nije definisan, ali se njegova pozitivna vrednost obeležja upitnosti automatski prenosi na čvor xX1.

<pre>class ObjekatRecenicniKompUpitni import ObjekatRecenicniKomp[] { <syn>{ node xX1 { node xC [cat = C, top = [wh = +]] node xXX1 (mark = subst) [top = [wh = -]] } } }</pre>	<pre> graph TD S --> C S --> Sd[S↓] </pre>
---	--

Slika 138 Klasa za upitni rečenični objekat

Infinitivni pravi objekat nasleđuje generalnu klasu za zavisne rečenice bez komplementizatora (Slika 130), kao i klasu koja definiše subjekatsku kontrolu (Slika 129, primer 2). Ovom klasom se definiše slaganje u obeležjima između umetnutog čvora S zavisne rečenice i glagola unutar nje (Slika 139).

<pre>class ObjekatInfinitiv import ZavisneReceniceBezKomp[] SubjekatskaKontrola[] { <syn>{ node xX1 [top = [func = dir]]; xKontrolisani = xX1; xVPKontroler = xVP } }</pre>

Slika 139 Klasa za rečenični objekat u infinitivu

Sve klase koje opisuju rečenični objekat smo spojili u klasu RecenicniObjekat (Slika 140).

<pre>class RecenicniObjekat { ObjekatRecenicniKompDeklarativni[] ObjekatRecenicniKompUpitni[] ObjekatInfinitiv[] }</pre>
--

Slika 140 Klasa za rečenični pravi objekat

Nepрави objekat

Klasa koju smo definisali za imenički nepravi objekat (Slika 141) identična je onoj za imenički pravi objekat, s razlikom u funkciji i padežu čvora nepravog objekta, koji je kod

nepravog objekta definisan kao genitiv, dativ ili instrumental. Kao i kod pravog objekta, nećemo ovde posebno davati klasu za zamenički nepravi objekat.

<pre>class NepraviObjekatN import NeSubjekat[] { <syn>{ node xX1 (mark = subst) [cat = N, top = [func = indir, wh = -, case = @{2,3,6}]] } }</pre>	<pre> graph TD S --> VP S --> N["N↓"] </pre>
---	--

Slika 141 Klasa za imenički nepravi objekat

Imenički i zamenički nepravi objekat spajamo u klasu NepraviObjekatRealizovan (Slika 142, primer 1). Klasa NerealizovanNepraviObjekat (Slika 142, primer 2) izgleda potpuno isto kao odgovarajuća klasa za pravi objekat (Slika 133, primer 2).

<p>1.</p> <pre>class NepraviObjekatRealizovan { PraviObjekatRealizovanN[] PraviObjekatRealizovanPro[] } </pre>	<p>2.</p> <pre>class NepraviObjekatNerealizovan import GlagolskiArgument[] </pre>
--	---

Slika 142 Klasa za realizovan nepravi objekat (1) i nerealizovan nepravi objekat (2)

Klasa za izmešten nepravi objekat u pitanjima (Slika 143) se od one za izmešten pravi objekat (Slika 134) razlikuje samo po funkciji.

<pre>class IzmeštenNepraviObjekat import Izmeštanje[] { <syn>{ node xIzm [cat = PRO, top = [func = indir]] } }</pre>	<pre> graph TD S --> PRO["PRO↓"] S --> S2["S"] </pre>
---	---

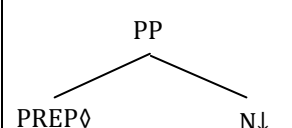
Slika 143 Klasa za izmešten nepravi objekat

Ovakve realizacije nepravog objekta, nasuprot kojima stoje rečnične i predložko-padežne realizacije, spojićemo u klasu NepraviObjekatIm (Slika 144).

<pre>class NepraviObjekatIm { NepraviObjekatRealizovan[] NepraviObjekatNerealizovan[] IzmeštenNepraviObjekat[] }</pre>
--

Slika 144 Klasa za realizovan, nerealizovan i izmešten nepravi objekat

Za nepravi objekat u obliku predložko-padežne konstrukcije razlikujemo dve varijante, u zavisnosti od toga da li se dopuna predloga realizuje kao imenica ili kao zamenica. Slika 145 prikazuje imeničku varijantu predložko-padežne konstrukcije, i ponovo, zameničku varijantu – NepraviObjekatPPPro nećemo ovde zasebno prenositi.

<pre>class NepraviObjekatPPN import PP[] { <syn>{ node xX1 [top = [func = indir]]{ node xPREP (mark = anchor) [cat = P] node xX [cat = N] } } }</pre>	
---	---

Slika 145 Klasa za imenički predložki nepravi objekat

Klasa NepraviObjekatPP predstavlja disjunkciju ove dve klase (Slika 146).

<pre>class NepraviObjekatPP { NepraviObjekatPPN[] NepraviObjekatPPPro[] }</pre>

Slika 146 Klasa za predložki nepravi objekat

Trenutno nemamo definisanu klasu za izmeštene predložki nepravi objekat, i za izmeštene predložke argumente uopšte, kako je za nju bilo potrebno napraviti malu izmenu u klasi Izmestanje. Naime, da bi klasa Izmestanje mogla da služi za analizu predložke fraze, potrebno je definisati frazalni nivo i terminalni nivo na izmeštenom čvoru, kako bi PP čvor mogao da se sjedini sa čvorom koji je trenutno definisan kao xIzm, a čvor predloga sa terminalnim čvorom koji bi mogao da bude definisan ispod čvora xIzm, nazovimo ga xIzm2.

Na samom kraju, natklasu NepraviObjekat (Slika 147) definišemo kao disjunkciju dosad definisanih klasa za nepravi objekat.

<pre>class NepraviObjekat { NepraviObjekatIm[] NepraviObjekatPP[] }</pre>

Slika 147 Klasa za nepravi objekat

Za definisanje rečeničnog nepravog objekta koristimo klasu RecenicniNepraviObjekatKomp, u kojoj smo najpre definisali funkciju ove jedinice (Slika 148).

```
class RecenicniNepraviObjekatKomp
import
    RecenicniNeSubjekatKomp[]
{
    <syn>{
        node xX1 [top = [func = indir]]
    }
}
```

Slika 148 Apstraktna klasa za rečenični nepravi objekat

Ovu klasu nasleđuju dve klase rečeničnog nepravog objekta – deklarativni nepravi objekat (Slika 149) i upitni nepravi objekat (Slika 150). U slučaju deklarativnog nepravog objekta, ovde je dat primer sa komplementizatorom *da*.

<pre>class RecenicniNepraviObjekatKompDeklarativni import RecenicniNepraviObjekatKomp[] { <syn>{ node xX1 [top = [wh = -]] { node xC [cat = da, top = [wh = -]] node xXX1 (mark = foot) [top = [wh = -]] } } }</pre>	<pre> graph TD S --> C S --> S_star[S*] C --> da[da] </pre>
--	--

Slika 149 Klasa za deklarativni rečenični nepravi objekat

<pre>class RecenicniNepraviObjekatKompUpitni import RecenicniNepraviObjekatKomp[] { <syn>{ node xX1 [top = [wh = +]] { node xC [cat = C, top = [wh = +]] node xXX1 (mark = subst) [top = [wh = -]] } } }</pre>	<pre> graph TD S --> C S --> S_down[S↓] C --> C[C] </pre>
---	--

Slika 150 Klasa za upitni rečenični nepravi objekat

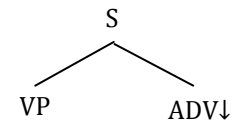
Date dve klase udružujemo u klasu RecenicniObjekatNepravi (Slika 151).

```
class RecenicniObjekatNepravi
{ RecenicniNepraviObjekatKompDeklarativni[] | RecenicniNepraviObjekatKompUpitni[] }
```

Slika 151 Klasa za rečenični nepravi objekat

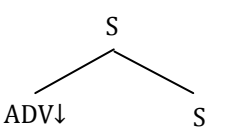
Priloška dopuna

Za osnovnu realizaciju priloške dopune kao priloga definisali smo klasu Prilog (Slika 152).

<pre>class Prilog import NeSubjekat[] { <syn>{ node xX1 (mark = subst) [cat = ADV, top = [func = pdop, wh = -]] } }</pre>	 <pre>graph TD S --> VP S --> ADV[ADV↓]</pre>
---	--

Slika 152 Klasa za prilog

Slika 153 prikazuje klasu izmeštenog, upitnog priloga.

<pre>class IzmeštenPrilog import Izmestanje[] { <syn>{ node xIzm [cat = ADV, top = [func = pdop]] } }</pre>	 <pre>graph TD S --> ADV[ADV↓] S --> S2[S]</pre>
---	--

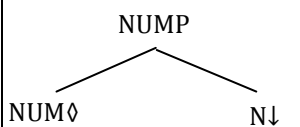
Slika 153 Klasa za izmešten prilog

Ove dve dopune udružujemo u osnovnu natklasu za priloge – PriloskaDopunaOsnovna (Slika 154).

<pre>class PriloskaDopunaOsnovna { Prilog[] IzmeštenPrilog[] }</pre>
--

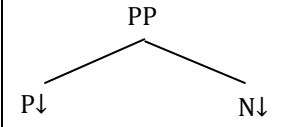
Slika 154 Klasa za osnovnu prilošku dopunu

Slika 155 prikazuje klasu koju smo definisali za brojne dopune.

<pre> class BrojnaDopuna import NeSubjekat[] export xNUM xNP declare ?xNUM ?xNP { <syn>{ node xX1 [cat = NUMP, top = [func = pdop, wh = -]] { node xNUM (color = red, mark = anchor) [cat = NUM] node xNP (color = red, mark = subst) [cat = N] } } } </pre>	 <pre> graph TD NUMP --- NUM◇ NUMP --- N↓ </pre>
--	---

Slika 155 Klasa za brojnu dopunu

Za prilošku dopunu u obliku predložke fraze prikazujemo samo varijantu sa imenicom kao čvorom za zamenu unutar fraze (Slika 156). Kao i u slučaju nepravog objekta, ni ovde ne obrađujemo izmeštenu predložku frazu. Iz istog razloga ne obrađujemo ni izmeštenu brojnu frazu – za uključivanje obe fraze je potrebna izvesna modifikacija u klasi *Izmestanje*.

<pre> class PriloskaDopunaPPN import PP[] { <syn>{ node xX1 [top = [func = pdop]]{ node xPREP (mark = subst) [cat = P, top = [case = ?fC]] node xX [cat = N] } } } </pre>	 <pre> graph TD PP --- P↓ PP --- N↓ </pre>
---	---

Slika 156 Klasa za prilošku dopunu u obliku predložke fraze s imenicom

Imenička i zamenička verzija predložke fraze udružuju se u klasu *PriloskaDopunaPP* (Slika 157).

<pre> class PriloskaDopunaPP { PriloskaDopunaPPN[] PriloskaDopunaPPPro[] } </pre>

Slika 157 Klasa za prilošku dopunu u obliku predložke fraze

Klasa PriloskaDopuna predstavlja konačnu klasu koja objedinjuje nerečenične priloške dopune, i predstavlja disjunkciju prethodno definisanih priloških klasa (Slika 158).

```
class PriloskaDopuna
{ PriloskaDopunaOsnovna[] | BrojnaDopuna[] | PriloskaDopunaPP[] }
```

Slika 158 Klasa za prilošku dopunu

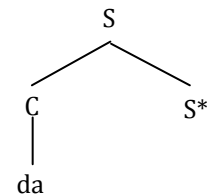
Slika 159 prikazuje klasu za infinitivnu jedinicu s priloškom funkcijom.

```
class PrilogInfinitiv
import
    ZavisneReceniceBezKomp[]
    SubjekatskaKontrola[]
{
    <syn>{
        node xX1 [top = [mod = W, func = pdop]];
        xKontrolisani = xX1;
        xVPKontroler = xVP
    }
}
```

Slika 159 Klasa za rečenicu u infinitivu s priloškom funkcijom

Za namerne rečenice s priloškom funkcijom ćemo pretpostaviti da imaju istu strukturu kao ranije navedene izrične rečenice, te će nasleđivati istu polaznu apstraktnu klasu. Od njih će se razlikovati samo po obeležju funkcije.

```
class PrilogRecenicni
import
    RecenicniNeSubjekatKomp[]
{
    <syn>{
        node xX1 [top = [wh = -, func = pdop]] {
            node xC [cat = da]
            node xXX1 (mark = foot) [top = [wh = -]]
        }
    }
}
```



Slika 160 Klasa za namerne rečenice u priloškoj funkciji

Klasu PriloskaDopunaRecenicna će tako činiti disjunkcija ove dve rečenične klase (Slika 161).

```
class PriloskaDopunaRecenicna
{ PrilogInfinitiv[] | PrilogRecenicni[] }
```

Slika 161 Klasa za rečeničnu prilošku dopunu

Logički subjekat

Za logički subjekat razlikujemo dve osnovne realizacije – imeničku (Slika 162) i zameničku. Kao i u ranijim primerima ovakve dvostruke realizacije, nećemo navoditi zameničku varijantu. Za razliku od većine nesubjektaskih argumenata, logički subjekat implementira klasu `GlagolskiArgument`. Redosled između čvora logičkog subjekta i predikata smo u ovoj (i zameničkoj) klasi definisali slobodno kako bismo kao samo jedan od redosleda definisali onaj u kome se logički subjekat nalazi na prvom mestu u rečenici. Ova odluka nije ni bolja ni gora od one da fiksiramo mesto logičkog subjekta za prvu poziciju, ali kako ćemo u jednom trenutku svakako definisati ovakav slobodan redosled za većinu glagolskih argumenata, mislili smo da bi ovo mogao da bude početak. U kombinaciji sa ovakvom klasom ide i klasa `SubjekatSlobodan` (Slika 122).

<pre>class LogickiSubjekatN import GlagolskiArgument[] export xLog declare ?xLog { <syn>{ node xS; node xLog (color = red, mark = subst) [cat = N, top = [func = lsub, wh = -, case = @{2,3,4}]]; node xVP; xS -> xLog; xS -> xVP } }</pre>	<pre>graph TD S1[S] --- N1[N↓] S1 --- VP1[VP] S2[S] --- VP2[VP] S2 --- N2[N↓]</pre>
---	---

Slika 162 Klasa za imenički logički subjekat

Klasa `LogickiSubjekatOpsti` (Slika 163) definisana je da obuhvati dve pomenute realizacije logičkog subjekta.

<pre>class LogickiSubjekatOpsti { LogickiSubjekatN[] LogickiSubjekatPro[] }</pre>

Slika 163 Klasa za imenički i zamenički logički subjekat

Slika 164 predstavlja klasu izmeštenog logičkog subjekta.

<pre>class IzmestenLogickiSubjekat import Izmestanje[] { <syn>{ node xIzm [cat = PRO, top = [func = lsub]] } }</pre>	
--	--

Slika 164 Klasa za izmešten logički subjekat

Klasa LogickiSubjekat objedinjuje sve pomenute varijante ove funkcije u rečeničnim stablima (Slika 165).

<pre>class LogickiSubjekat { LogickiSubjekatOpsti[] IzmestenLogickiSubjekat[] }</pre>

Slika 165 Klasa za logički subjekat

4.4 Definisane klase koje opisuju promenu dijateze i porodice stabala

U ovoj fazi definišemo porodice stabala ili rečenične modele kroz njihove alternative. Ova faza podrazumeva najpre definisanje klase za određenu dijatezu, odnosno neki drugi tip strukturne promene koji se vezuje za datu porodicu stabala, a zatim udruživanje svih takvih alternativnih struktura u jednu porodicu stabala. Prođimo sada još jednom kroz rečenične modele kakvi su dati u poglavlju 3.6, najpre subjekatsko-predikatske, a zatim i bezlične. Za svaki od njih daćemo opis strukture koja se određenim leksičkim pravilom vezuje za datu porodicu, a koji je označen komentaram *% alternative*, kao i definiciju same porodice, označenu komentaram *% porodice*. Uz svaku od klasa u modelima naveden je ilustrativni primer rečenice koji data klasa prepoznaje.

4.4.1 Subjekatsko-predikatski modeli

4.4.1.1 Klase prvog rečeničnog modela

Za prvi rečenični model (Slika 166) definisane su dve alternativne klase. Za klasu s aktivnim glagolom definisali smo klasu n0Vaktiv kao minimalnu subjekatsko-predikatsku klasu. Dok su natklase u primerima definisanja funkcija podrazumevale spajanje potklasa disjunkcijom, ovde se klasa Subjekat i AktivniGlagol, koje čine ovu klasu, spajaju

konjunkcijom, označenom kao „;”. n0Vobezl na istoj slici predstavlja obezličenu verziju ovog modela – sa nerealizovanim subjektom i refleksivnim glagolom. Reprezentativnu klasu ovog modela – klasu n0V činiće tako disjunkcija ove dve klase. Drugu klasu ovog modela definisali smo za refleksivne glagole – za one sa realizovanim subjektom (n0ClVSub) i sa nerealizovanim subjektom (n0ClVNesub), koje spajamo u kompaktnu klasu n0ClV. Refleksivne klase koje koristimo za njih su one kod kojih je rečca se definisana kao sidro, budući da su u ovom modelu refleksivni glagoli uvek nepravni povratni.

```

% alternative
class n0Vaktiv % Zoran spava.
{ Subjekat1[]; AktivniGlagol[] }

class n0Vobezl % Spava se.
{ SubjekatNerealizovan[]; BezlicniDesni[] }

% porodice
class n0V
{n0Vaktiv[] | n0Vobezl[] }

class n0ClVSub % Ti se salis.
{ Subjekat[]; RefleksivniGlagolLevi[] }

class n0ClVNesub % Salis se.
{ SubjekatNerealizovan[]; RefleksivniGlagolDesni[] }

class n0ClV
{ n0ClVSub[] | n0ClVNesub[] }

```

Slika 166 Klase prvog rečeničnog modela

4.4.1.2 Klase drugog i trećeg rečeničnog modela

Za drugu i treću klasu (Slika 167) definisali smo iste alternative. U pitanju su klasa s aktivnim glagolom (n0Vn1diraktiv), klasa s pasivnim glagolom (n0Vn1dirpasiv), klasa s glagolom u obliku refleksivnog pasiva (n0Vn1dirrefpas) i klasa koja predstavlja refleksivnu verziju prelaznog glagola (n0Vn1dirref). Pri definisanju aktivne klase može se videti da smo kao gradivni blok iskoristili aktivnu klasu iz porodice 1, na koju zatim samo dodajemo pravi objekat.

```

% alternative
class n0Vn1diraktiv % Jasna kupuje knjigu.
{ n0Vaktiv[]; PraviObjekat[] }

class n0Vn1dirpasiv % pasivizacija: Knjiga je kupljena.
{Subjekat[]; PasivniGlagol[] }

```

```

class n0Vn1dirrefpas % reflektivni pasiv: Knjige se kupuju.
{ Subjekat[]; ReflektivniGlagolLevi[] }

class n0Vn1dirref % reflektivizacija: Marija se ceslja.
{ Subjekat[]; ReflektivniGlagolLeviSubst[] }

% porodice
class n0Vn1dir
{n0Vn1diraktiv[] | n0Vn1dirpasiv[] | n0Vn1dirrefpas[] | n0Vn1dirref[] }

class n0Vn1dirNer % Jasna cita.
{n0Vaktiv[]; ObjekatPlusNerealizovan[] }

class n0Vsinfl1dir % Jasna zeli citati.
{ n0Vaktiv[]; ObjekatInfinitiv[] }

class n0Vdas1dir % Marko pokusava da nadje karte za utakmicu.
{ n0Vaktiv[]; ObjekatRecenicniKompDeklarativni[] }

class n0Vsup1dir % Marko zna da li ima hleba.
{ n0Vaktiv[]; ObjekatRecenicniKompUpitni[] }

```

Slika 167 Klase drugog i trećeg rečeničnog modela

4.4.1.3 Klase četvrtog rečeničnog modela

Za četvrtu porodicu (Slika 168) smo definisali tri klase koje predstavljaju alternative osnovnog modela s nerefleksivnim glagolom. Prva klasa među alternativama je ona sa aktivnim glagolom. Definisali smo je pomoću klase koju smo predstavili u aktivnoj alternativni prvog rečeničnog modela i dodatka (imeničkog ili zameničkog) nepravog objekta, uključujući i njihove izmeštene varijante. Za klasu pasivne dijateze, subjekat ovog modela smo definisali kao logički subjekat, budući da se po padežima koji se mogu javiti na ovoj jedinici (padeži nepravog objekta u aktivnoj strukturi) poklapaju sa padežima logičkog subjekta. Kao poslednju alternativu definisali smo klasu za obezličenu varijantu ovog modela.

Osnovna klasa za ovaj rečenični model, tj. porodicu stabala je tako klasa koju smo definisali kao disjunkciju svih prethodno navedenih alternativa. Za nerefleksivni glagol sa nepravim objektom u obliku predložko-padežne konstrukcije i deklarativne izrične rečenice smo u ovom trenutku definisali po jednu klasu, iako je i za njih moguće definisati obezličenu alternativu (*Odustalo se od ideje* i *Razmišljalo se da se upiše na prava.*). Sa reflektivnim glagolom su definisane ekvivalentne klase, gde svaku definišemo preko iste polazne klase za reflektivne glagole iz prvog rečeničnog modela, osim u slučaju poslednjeg modela, u kome

glagol *pomiriti se* posmatramo kao rezultat refleksivizacije, te koristimo ekvivalentnu klasu iz drugog i trećeg modela.

Za svaki od regiranih predloga bi prema modelima koje smo videli u poglavlju 3.6 trebalo definisati poseban model, gde bi predlog direktno ulazio u naziv modela, i u rečenično stablo, međutim ovde navodimo samo jedan opšti koji smo zasad definisali u metagramatici (klase *n0Vpn1ind* i *n0ClVpn1ind*).

```
% alternative
class n0Vn1indaktiv % Otac pomaze sinu.
{ n0Vaktiv[]; NepraviObjekatIm[] }

class n0Vn1indpasiv % pasivizacija: Sinu je pomagano.
{ LogickiSubjekat[]; PasivniGlagol[] }

class n0Vn1indobezl % obezlicena struktura: Sinu se pomaze.
{ LogickiSubjekat[]; SubjekatNerealizovan[]; BezlicniLevi[] }

% porodice
class n0Vn1ind
{ n0Vn1indaktiv[] | n0Vn1indpasiv[] | n0Vn1indobezl[] }

class n0Vpn1ind % Marko je odustao od svoje ideje.
{ n0Vaktiv[]; NepraviObjekatPP[] }

class n0Vdas1ind % Ivan je razmisljao da se upise na prava.
{ n0Vaktiv[]; RecenicniNepraviObjekatKompDeklarativni[] }

class n0ClVn1ind % Ivan se stidi brata.
{ n0ClV[]; NepraviObjekatIm[] }

class n0ClVpn1ind % Marko se zaljubio u Tanju.
{ n0ClV[]; NepraviObjekatPP[] }

class n0ClVdas1ind % Ivan se pomirio da nece ici na skijanje.
{ n0Vn1dirref[]; RecenicniNepraviObjekatKompDeklarativni[] }
```

Slika 168 Klase četvrtog rečeničnog modela

4.4.1.4 Klase petog rečeničnog modela

Za peti model (Slika 169) već nismo direktno morali da definišemo alternative. Skoro sve klase sa nerefleksivnim glagolom definisali smo preko osnovne klase iz prvog rečeničnog modela – *n0V*, koja već predstavlja alternativu aktivnog i obezličenog modela, kakav je moguć i u ovom slučaju (npr. *Stanuje se ovde*. kao obezličena alternativa za prvu klasu). Na takvu alternativu smo zatim samo dodali odgovarajuću prilošku dopunu. Za klasu sa brojnou dopunou iskoristili smo samo klasu definisanu za aktivne neprelazne glagole.

Klase sa refleksivnim glagolom se takođe pozivaju na refleksivnu klasu iz prvog modela i odgovarajuće dopune.

```
% porodice
class n0Vad1 % Ja stanujem ovde.
{ n0V[]; PriloskaDopunaOsnovna[] }

class n0Vpnlpril % Torba lezi na klupi.
{ n0V[]; PriloskaDopunaPP[] }

class n0Vnumn1 % Racun iznosi 10.000 dinara.
{ n0Vaktiv[]; BrojnaDopuna[] }

class n0Vsinflpril % Marko odlazi vecerati.
{ n0V[]; PrilogInfinitiv[] }

class n0Vsad1 % Marko odlazi da vecera.
{ n0V[]; PrilogRecenicni[] }

class n0ClVad1 % Nikola se ponasa cudno.
{ n0ClV[]; PriloskaDopunaOsnovna[] }

class n0ClVpnl % Knjiga se nalazi na stolu.
{ n0ClV[]; PriloskaDopunaPP[] }
```

Slika 169 Klase petog rečeničnog modela

4.4.1.5 Klase šestog rečeničnog modela

Za model šest (Slika 170) nemamo alternative koje mogu biti povezane nekim od leksičkih pravila redistribucije, te smo za njih definisali četiri odvojena modela. Za potrebe ovog modela smo napravili novu subjekatsku klasu, kako bismo omogućili slobodno raspoređivanje subjekta u odnosu na glagol. U primeru modela sa izričnim subjektom trenutno imamo definisan samo dati redosled argumenata.

```
% porodice
class n1lsubVn0 % Jasnu boli zub.
{ LogickiSubjekat[]; AktivniGlagol[]; SubjekatSlobodan[] }

class n1lsubVs0 % Marku odgovara da dodju.
{ LogickiSubjekat[]; AktivniGlagol[]; IzricniSubjekat[] }

class n1lsubClVn0 % Marku se svidja Marija.
{ LogickiSubjekat[]; Refleksivni[]; SubjekatSlobodan[] }

class n1lsubClVs0 % Marku se svidja da pusta muziku.
{ LogickiSubjekat[]; Refleksivni[]; IzricniSubjekat[] }
```

Slika 170 Klase šestog rečeničnog modela

4.4.1.6 Klase sedmog rečeničnog modela

Za ovaj model (Slika 171) smo alternative ponovo definisali preko jedne od prethodno formiranih disjunkcija alternativa – klase iz drugog, trećeg i četvrtog modela. Na njih smo zatim samo dodali odgovarajuće klase za nepravi, odnosno pravi objekat.

```
% porodice
class n0Vn1dirn2ind % Tanja je vratila knjigu Zoranu.
{ n0Vn1dir[]; NepraviObjekatIm[] }

class n0Vn2inddas1dir % Marko je porucio Ivanu da mu vrati ploce.
{ n0Vn1ind[]; ObjekatRecenicniKompDeklarativni[] }
```

Slika 171 Klase sedmog rečeničnog modela

4.4.1.7 Klase osmog rečeničnog modela

I za osmi model (Slika 172) alternative definišemo na isti način kao u klasama sedmog rečeničnog modela, a kako je ponovo u pitanju prelazni glagol, koristimo istu polaznu klasu koju smo koristili u prvoj porodici u sedmom modelu n0Vn1dir. Na ovu klasu zatim samo dodajemo klase za različite realizacije priloške dopune.

```
% porodice
class n0Vn1dirad2 % Marko stavlja kameru blizu.
{ n0Vn1dir[]; PriloskaDopunaOsnovna[] }

class n0Vn1dirpn2 % Ivan ubacuje pismo u sanduce.
{ n0Vn1dir[]; PriloskaDopunaPP[] }

class n0Vn1dirnumn2 % Marko je platio knjigu 100 dinara.
{ n0Vn1dir[]; BrojnaDopuna[] }
```

Slika 172 Klase osmog rečeničnog modela

4.4.1.8 Klase devetog rečeničnog modela

Kako za ove klase (Slika 173) nema alternativa definisanih preko leksičkog pravila redistribucije, definisali smo samo osnovne klase – jednu za imenički predikativ, a drugu za pridevski. U ovom trenutku je redosled reči u ovim klasama fiksiran, te podrazumevamo da one uvek sadrže subjekat i da se kopula nalazi između subjekta i predikativa.

```

% porodice
class n0vN % Ovo je Marija.
{Subjekat[]; ImenickiPredikat[] }

class n0vA % Jasna je pametna.
{Subjekat[]; PridevskiPredikat[] }

```

Slika 173 Klase devetog rečeničnog modela

4.4.1.9 Klase desetog rečeničnog modela

Kao i za prethodnu klasu, ni ovde (Slika 174) ne postoje alternative u smislu leksičkih pravila redistribucije, te definišemo samo osnovnu klasu, u kojoj se priloški predikativ u ovom trenutku realizuje samo kao prilog. Takođe, ni za ovu klasu nije trenutno definisan slobodan red reči.

```

% porodice
class n0vAd % Stanica je daleko.
{ Subjekat[]; PriloskiPredikatAd[] }

```

Slika 174 Klase desetog rečeničnog modela

4.4.2 Bezlični modeli

4.4.2.1 Klase prvog bezličnog modela

Ovaj model (Slika 175), kao ni ostali bezlični modeli, nemaju alternative povezane leksičkim pravilom redistribucije, pa za svaki od njih definišemo samo osnovne klase. Za prepoznanje prve rečenične porodice koristimo klasu koju smo definisali za bezlične aktivne glagole.

Za prepoznavanje klase sa refleksivnim glagolom koristimo klasu koju smo definisali za refleksivne bezlične i obezličene glagole.

```

% porodice
class V % Seva.
{ BezlicniAktivniGlagol[] }

class VC1 % Smrkava se.
{ BezlicniDesni[] }

```

Slika 175 Klase prvog bezličnog modela

4.4.2.2 Klase drugog bezličnog modela

Drugi model (Slika 176) definišemo preko kopulativne klase definisane posebno za ove potrebe.

```
% porodice
class Adv % Sparno je.
{ PriloskiPredikatAdLevi[] }
```

Slika 176 Klasa drugog bezličnog modela

4.4.2.3 Klase trećeg bezličnog modela

I treći model definisan je na intuitivni način, kombinovanjem pojedinačnih klasa za argumente i odgovarajući tip predikata (Slika 177).

```
% porodice
class n0lsubV % Marku je pozlilo.
{ LogickiSubjekat[];BezlicniAktivniGlagol[] }

class n0lsubClV % Marku se slosilo.
{ LogickiSubjekat[]; BezlicniLevi[] }
```

Slika 177 Klase trećeg bezličnog modela

4.4.2.4 Klase četvrtog bezličnog modela

Za četvrti model (Slika 178) smo iskoristili posebnu kopulativnu klasu kao i u drugom bezličnom modelu 2.

```
% porodice

class n0lsubvAd % Marku je dobro.

{ LogickiSubjekat[]; PriloskiPredikatAdDesni[] }
```

Slika 178 Klasa četvrtog bezličnog modela

4.5 Dodatne klase

Postoji još nekoliko klasa koje je bilo potrebe da definišemo, a koje su definisane kao takve za potrebe parsiranja. U jednu takvu grupu spadaju pojedinačne klase za svaki od čvorova koji je obeležen za zamenu. Da bi leksikalizacija gramatike bila moguća, i da bi bilo

moгуće parsirati rečenicu, potrebno je da postoje posebni čvorovi koji direktno vezuju gramatiku za leksikon. Sve takve čvorove koje trenutno imamo definisane u metagramatici dajemo niže (Slika 179). Nećemo za njih posebno navoditi stabla, kako bi se ona sastojala samo od jednog čvora uz koji bi stajala obeležja navedena u klasama.

```

class N % imenice
declare
    ?xN ?fG ?fN ?fC
{
    <syn>{
        node xN (color = red, mark = anchor) [cat = N, bot = [pers = z, wh = -,
gen = fG, num = fN, case = fC]]
    }
}

class PRO % zamenice
declare
    ?xPRO ?fP ?fW ?fG ?fN ?fC
{
    <syn>{
        node xPRO (color = red, mark = anchor) [cat = PRO, bot = [pers = fP, wh =
fW, gen = fG, num = fN, case = fC]]
    }
}

class Se % reflektivna rečca koja se unosi zamenom
declare
    ?xC1
{
    <syn>{
        node xC1 (color = red, mark = anchor, rank = 5) [cat = C1, bot = [ref =
+]]
    }
}

class Cop % kopula
declare
    ?xVcop ?fM ?fF ?fP ?fN
{
    <syn>{
        node xVcop (color = red, mark = anchor) [cat = V, bot = [mod = fM, form =
fF, pers = fP, num = fN, cop = +]]
    }
}

class P % predlog
declare
    ?xP ?fC
{
    <syn>{
        node xP (color = red, mark = anchor) [cat = P, bot = [case = fC]]
    }
}

class ADV % prilog
declare

```

```

?xADV
{
  <syn>{
    node xADV (color = red, mark = anchor) [cat = ADV]
  }
}

class NUM % broj
declare
  ?xNUM
{
  <syn>{
    node xNUM (color = red, mark = anchor) [cat = NUM]
  }
}

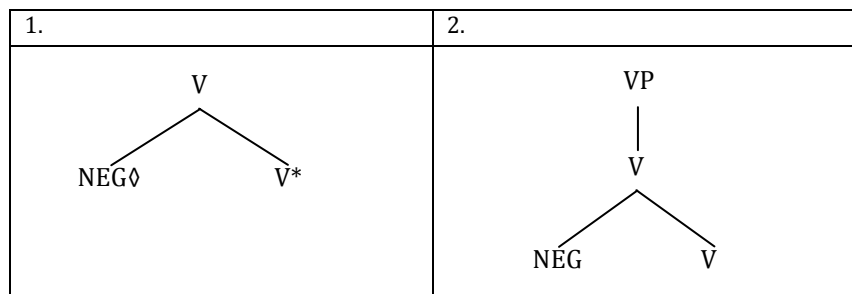
class C % komplementizator
declare
  ?xC
{
  <syn>{
    node xC (color = red, mark = anchor) [cat = C]
  }
}

```

Slika 179 Klase čvorova obeleženih za zamenu

U drugu grupu klasa koje je bilo potrebno definisati odvojeno za potrebe parsiranja spadaju klase pomoćnih, neargumentskih rečeničnih delova, kao što su negacija i pomoćni glagol. U ovu posebnu grupu će u budućnosti ući i klase modifikatora sintagmi, kao i klasa za neargumentska pitanja, npr. pitanje sa upitnom rečcom *da li*.

Sve klase koje smo definisali na slici niže (Slika 181) su klase struktura koje se pripajaju na glagolsku frazu. Međutim, da bi ovakvo stablo, odnosno njegov opis u vidu klase, mogao da se identifikuje sa opisom glagola, bilo je potrebno da ova stabla imaju strukturu kakva je prikazana niže (Slika 180, primer 1). U tom slučaju, identifikacija ovakve klase i klase glagola daje strukturni opis kakav se može videti na slici pored (Slika 180, primer 2).



Slika 180 Glagolska fraza na koju je pripojena negacija u SrpTAG (1) i kao izlaz metagramatike (2)

Za negaciju smo definisali prvu klasu (Neg) na slici niže (Slika 181). Za pomoćne glagole smo definisali dve odvojene klase, jednu za prošlo vreme (AuxProslo) i jednu za buduće vreme (AuxBuduce). Klase za ostala složena vremena i načine će u budućnosti biti definisala po istom principu. Definisali smo i odvojenu klasu za fazne glagole (Fazni) u prezentu. Modalne glagole, kako je njihova struktura drugačija, trenutno ne analiziramo. U svim primerima čvor xTop se odnosi na gornji čvor V koji se može videti u primeru 2 (Slika 180), dok se čvor xR odnosi na donji čvor V. Vrednost obeležja mark na čvoru xTop – nadj, zabranjuje da na datom čvoru dođe do daljeg pripajanja.

```

class Neg % negacija
export
  xTop xNeg xR
declare
  ?xTop ?xNeg ?xR
{
<syn>{
  node xTop (color = red, mark = nadj) [cat = V, bot = [neg = +]] {
    node xNeg (mark = anchor, color = red) [cat = NEG, top = [neg = +]]
    node xR (mark = foot, color = red) [cat = V, top = [mod = I, neg = -]]
  }
}

class AuxProslo % pomoćni glagol za građenje prošlog vremena
export
  xTop xL xR
declare
  ?xTop ?xL ?xR ?fP ?fN ?fG
{
<syn>{
  node xTop (color = red, mark = nadj) [cat = V, bot = [mod = I, form = R, num = fN,
pers = fP, gen = fG]] {
    node xL (mark = anchor, color = red) [cat = V, top = [mod = I, form = P, num = fN,
pers = fP, aux = +]]
    node xR (mark = foot, color = red) [cat = V, top = [mod = G, num = fN, gen = fG]]
  }
}

```

```

}

class AuxBduce % pomoćni glagol za građenje budućeg vremena
export
    xTop xL xR
declare
    ?xTop ?xL ?xR ?fP ?fN ?fG
{
<syn>{
    node xTop (color = red, mark = nadj) [cat = V, bot = [mod = I, form = F, num = fN,
pers = fP]] {
        node xL (mark = anchor, color = red) [cat = V, top = [mod = I, form = P, num = fN,
pers = fP, aux = +]]
        node xR (mark = foot, color = red) [cat = V, top = [mod = W]]
            }
    }
}

class Fazni % fazni glagoli
export
    xTop xL xR
declare
    ?xTop ?xL ?xR ?fF ?fP ?fN ?fG
{
<syn>{
    node xTop (color = red, mark = nadj) [cat = V, bot = [mod = I, form = fF, num = fN,
pers = fP, gen = fG]] {
        node xL (mark = anchor, color = red) [cat = V, top = [mod = I, form = P, num = fN,
pers = fP, fazni = +]]
        node xR (mark = foot, color = red) [cat = V, top = [mod = W, asp = imprf]]
            }
    }
}
}

```

Slika 181 Pomoćna glagolska stabla

5 PRIMENA METAGRAMATIKE

5.1 Leksikoni

Klase metagramatike koje smo definisali u poglavlju 4 predstavljaju obrazac po kome možemo vršiti parsiranje rečeničnih struktura. Međutim, da bi bilo moguće parsiranje konkretnog teksta, potrebno je da pored metagramatike postoje i odgovarajući leksikoni u kojima je svaka od reči povezana sa odgovarajućom klasom u metagramatici. Povezivanje rečeničnih struktura sa leksikonima ne vrši se pomoću same metagramatike, već pomoću parsera TuLiPA.

Parser TuLiPA kao svoj ulaz zahteva dva leksikona – leksikon lema i morfološki leksikon. Oba leksikona se nalaze u .xml formatu i za njih je propisan odgovarajući DTD¹⁷², koji se može se videti u dodacima (poglavlje 8.1).

U leksikonima za predstavljanje srpskih slova koja sadrže dijakritike ili su digrafi (č, ć, đ, dž, š, ž, nj i lj) koristimo aurora kodni raspored, koji se koristi i u elektronskim rečnicima i korpusu srpskog jezika SrpKOR. Tabela u kojoj je prikazan odnos pomenutih srpskih slova i aurora karaktera data je niže¹⁷³ (Tabela 14).

Tabela 14 AURORA kod za ne-ASCII srpska latinična slova

Veliko slovo	AURORA kod	Malo slovo	AURORA kod	Primer
Č	CY, Cy	č	cy	Čačak = Cyacyak
Ć	CX, Cx	ć	cx	Ćićevac = Cxixevac
Đ	DX, Dx	đ	dx	Đorđe = Dxordxe
Š	SX, Sx	š	sx	Šuškatı = Sxusxkatı
Ž	ZX, Zx	ž	zx	Žižak = Zxizxak
NJ, Nj	NX, Nx	nj	nx	Njegoš = Nxegosx
LJ, Lj	LX, Lx	lj	lx	Ljubljana = Lxublxana
DŽ, Dž	DY, Dy	dž	dy	Džordž = Dyordy

Leksikon lema se sastoji od osnovnih oblika reči, uz koje se, ukoliko su u pitanju reči koje se u rečeničnim modelima mogu javiti kao sidra, navode i rečenični modeli u kojima

¹⁷² engl. Document Type Definition, obrazac koji propisuje obaveznu strukturu XML dokumenta.

¹⁷³ Tabela je uz izmene preuzeta sa sajta: <http://korpus.matf.bg.ac.rs/prezentacija/uputstvo.html>

imaju ulogu sidra. Svaki glagol je tako u ovom leksikonu povezan sa odgovarajućim rečeničnim modelom ili modelima u kojima se može naći (Slika 182).

```
<lemma name="spavati" cat="V" >
  <anchor tree_id="family[@name=n0V]" >
  </anchor>
</lemma>
```

Slika 182 Unosak glagola *spavati* u leksikonu lema

Jedinice koje kao svoje dodatno sidro imaju još neku leksemu – u našem slučaju su to refleksivni glagoli – u okviru svog unoska sadrže i opis dodatnog sidra, u ovom slučaju rečce *se* (Slika 183).

```
<lemma name="smrkavati" cat="V" >
  <anchor tree_id="family[@name=VC1]" >
    <coanchor node_id="se" cat="Cl">
      <lex>se</lex>
    </coanchor>
  </anchor>
</lemma>
```

Slika 183 Unosak glagola *smrkavati se* u leksikonu lema

Svaka imenica, zamenica i pridev potencijalno mogu biti sidra u rečenicama s kopulativnim glagolima, te se i oni povezuju sa odgovarajućim rečeničnim modelima (Slika 184, drugi primer). Budući da se ove jedinice u rečenična stabla češće unose zamenom, za njih se definišu dupli unosci, pri čemu se u slučaju unosa zamenom povezuju sa klasom koja opisuje ponašanje njihove kategorije (Slika 184, prvi primer). U slučaju imenica, to će biti kategorija N, koju smo opisali u dodatnim klasama metagramatike (Slika 179).

```
<lemma name="Sonxa" cat="N" >
  <anchor tree_id="family[@name=N]" >
  </anchor>
</lemma>
<lemma name="Sonxa" cat="N" >
  <anchor tree_id="family[@name=n0vN]" >
  </anchor>
</lemma>
```

Slika 184 Unosci imenice *Sonja* u leksikonu lema

U leksikon lema su uneta i leksička ograničenja na formiranje participskog pasiva (kroz obeležje pasiv) i izostavljanje argumenata (obeležja bez1 i bez2).

Morfološki leksikon se sastoji od morfoloških oblika leme i vrednosti njenih morfoloških kategorija (Slika 185). U ovom leksikonu svaka reč definiše se zasebno, bez obzira na to što neke od njih čine jednu jedinicu u leksikonu lema (kao glagol *smrkavati se*). Za svaku od lema ovde postoji onoliko unosaka koliko ima njenih morfoloških oblika.

```
<morph lex="spava" >
  <lemmaref cat="V" name="spavati" >
    <fs>
      <f name="pers">
        <sym value="z"/>
      </f>
      <f name="num">
        <sym value="s"/>
      </f>
      <f name="mod">
        <sym value="I"/>
      </f>
      <f name="form">
        <sym value="P"/>
      </f>
      <f name="neg">
        <sym value="-"/>
      </f>
    </fs>
  </lemmaref>
</morph>
```

Slika 185 Jedan od unosaka glagola *spavati* u morfološkom leksikonu

Trenutan broj jedinica u oba leksikona je mali – uneti su predstavnici svake od klasa i svega još po nekoliko primera kako bi bilo moguće testiranje sistema (leksikoni su u celosti dati u dodacima, poglavlja 8.2 i 8.3). Da bi se metagramatika mogla primeniti na rečenice sa raznovrsnijim izborom reči, bilo bi potrebno informacije koje trenutno postoje u DELAS i DELAF rečnicima automatski transformisati u format leksikona lema i morfološkog leksikona, uz ručno dodavanje određenih informacija. Pogledajmo najpre kako bi izgledala transformacija iz rečnika DELAS.

Za jedinice koje se u stabla unose zamenom, transformacija bi bila manje-više direktna. Kako smo zadržali nazive za vrste reči, i kako smo klase koje opisuju ponašanje date vrste reči nazvali na isti način kao u elektronskim rečnicima za srpski jezik (N, PRO, ADV...), potrebno je samo drugačije formatirati unosak (Slika 186). Pri transformaciji bi došlo do određenog gubitka informacija – u ovom slučaju semantičkih informacija (Hum) i lingvističkih informacija (Ek), ali se i one mogu uneti u nadi da će u jednom trenutku i ove informacije biti dodate u analizi.

```

devojka.N+Hum+Ek      →      <lemma name="devojka"  cat="N" >
                                <anchor tree_id="family[@name=N]" >
                                </anchor>
                                </lemma>

```

Slika 186 Transformacija unoska leme *devojka* iz rečnika DELAS u leksikon lema

Za glagole i jedinice koje mogu biti sidra u rečenici, ova transformacija mora biti praćena ručnim dodavanjem porodice stabala kojoj data jedinica pripada (Slika 187).

```

spavati.V              →      <lemma name="spavati"  cat="V" >
                                <anchor tree_id="family[@name=n0V]" >
                                </anchor>
                                </lemma>

```

Slika 187 Transformacija unoska leme *spavati* iz rečnika DELAS u leksikon lema

Za glagole koji su označeni kao refleksivni, transformacija bi se vršila u unosak koji sadrži rečcu *se* kao dodatno sidro (Slika 188).

```

bojati.V+Imperf+It+Ref  →      <lemma name="bojati"  cat="V" >
                                <anchor tree_id="family[@name=n0ClVn1ind]" >
                                <coanchor node_id="se"  cat="Cl" >
                                    <lex>se</lex>
                                </coanchor>
                                </anchor>
                                </lemma>

```

Slika 188 Transformacija unoska leme *bojati (se)* iz rečnika DELAS u leksikon lema

Pri transformaciji unosaka iz rečnika DELAF u morfološki leksikon, deo informacija bi se potpuno identično preneo. Kao ilustraciju ove transformacije, dajemo primere niže (Slika 189). Postoji nekoliko kategorija koje u morfološkom leksikonu koristimo, a koje ne postoje u rečniku DELAF. Dve od njih, koje su obeležene u pomenutom primeru, su kategorija lica za imenice – koja je podrazumevano treće lice, te ne predstavlja veći problem, i kategorija upitnosti. Vrednost kategorije upitnosti je za imenice generalno negativna, sem ukoliko se ne navede suprotno oznakom Wh, tako da ni ova kategorija ne predstavlja problem.

1.

```
<morph lex="devojci" >
  <lemmaref cat="N" name="devojka" >
    <fs>
      <f name="case">
        <sym value="3"/>
      </f>
      <f name="pers">
        <sym value="z"/>
      </f>
      <f name="gen">
        <sym value="f"/>
      </f>
      <f name="num">
        <sym value="s"/>
      </f>
      <f name="wh">
        <sym value="-"/>
      </f>
    </fs>
  </lemmaref>
</morph>
```

devojci, devojka.N+Hum+Ek:fs3v:fs7v →

2.

```
<morph lex="devojci" >
  <lemmaref cat="N" name="devojka" >
    <fs>
      <f name="case">
        <sym value="7"/>
      </f>
```

```

    <f name="pers">
      <sym value="z"/>
    </f>
    <f name="gen">
      <sym value="f"/>
    </f>
    <f name="num">
      <sym value="s"/>
    </f>
    <f name="wh">
      <sym value="-"/>
    </f>
  </fs>
</lemmaref>
</morph>

```

Slika 189 Transformacija unoska za reč *devojka* iz rečnika DELAF u morfološki leksikon

Pogledajmo još kako bi ova transformacija izgledala kod glagola (Slika 190).

```

    <morph lex="stavlx" >
      <lemmaref cat="V" name="stavljati" >
        <fs>
          <f name="pers">
            <sym value="z"/>
          </f>
          <f name="num">
            <sym value="s"/>
          </f>
          <f name="mod">
            <sym value="I"/>
          </f>
          <f name="form">
            <sym value="P"/>
          </f>
          <f name="asp">
            <sym value="imperf"/>
          </f>
        </fs>
      </lemmaref>
    </morph>

```

stavlx,stavlxati.V+Imperf+Tr+Iref:Pzs →

Slika 190 Transformacija unoska za glagol *stavljati* iz rečnika DELAF u morfološki leksikon

Sve morfološke informacije iz DELAF klase mogu se direktno transformisati u vrednosti odgovarajućih obeležja u leksikonu. Informacija o glagolskom vidu takođe ulazi u leksikon. Dve informacije koje ne ulaze direktno u leksikon – informacija o tome da je glagol prelazan (Tr) i nerefleksivan (Iref) mogu se grubo iskoristiti za određivanje klase kojoj pripada ovaj glagol (modeli 2, 3, 7 ili 8).

5.2 Evaluacija

U ovom poglavlju ćemo dati opšti pregled konstrukcija i rečenica koje je trenutno moguće automatski prepoznati koristeći navedenu metagramatiku. U primerima će biti korišćen ograničen vokabular, iz razloga koje smo naveli u poglavlju o leksikonima (5.1), ali možemo smatrati da bi bilo koja varijacija vokabulara iste klase bila moguća.

Kako bi navođenje rečenica koje sistem prepoznaje bio izuzetno redundatan posao, umesto po modelima, navodićemo slučajeve koje sistem prepoznaje po rečeničnim funkcijama, dodajući gde je potrebno informacije o pojedinačnim modelima. Pre toga je potrebno dodati još par napomena:

- Parser TuLiPA pravi razliku između malih i velikih slova. Samim time, moguće je prepoznavati reči samo u onom obliku u kome se nalaze u leksikonu.
- Interpunkcija nije uneta u klase, te se trenutno ne prepoznaje ni u kakvom obliku.

Pogledajmo sada kakav je stepen prepoznavanja svakog od glagolskih argumenata i samih glagola.

Subjekt

Sistem u ovom trenutku prepoznaje sve imeničke vrste subjekta (Marko/Sonja), zameničke vrste subjekta – kao lične zamenice i odgovarajuće upitne zamenice za formiranje subjekatskog pitanja (*ko, šta, kao u Ko spava?*). Kongruencija subjekta sa pomoćnim glagolom, kopulom i participom funkcioniše ispravno kroz sve klase. Rečenični subjekat ne prepoznajemo u ovom trenutku.

Jedina pozicija subjekta koju trenutno prepoznajemo jeste ona na prvom mestu u rečenici. Izuzetak su rečenice sa logičkim subjektom, za koje smo formulisali eksperimentalnu klasu koja ima nezavisan redosled reči te subjekat u ovom slučaju prepoznajemo i iza glagola. Nerealizovan subjekat se ispravno prepoznaje kroz modele s glagolskim predikatom, uključujući i refleksivne glagole, međutim u modelima s kopulativnim glagolima ne uzimamo u obzir to da subjekat može biti izostavljen.

Predikat

Sistem ispravno prepoznaje glagolske predikate u pozitivnom i negativnom obliku u prezentu, uključujući i slučajeve sa refleksivnim glagolima (npr. *Marko se ne šali.*). Oblici refleksivnog pasiva i obezličene varijante glagola se uspešno prepoznaju u oba slučaja. Trenutno ne prepoznajemo negaciju na pomoćnim glagolima i kopuli, pa tako ni u participskom pasivu. Prošlo vreme i buduće vreme prepoznajemo na glagolima koji nisu refleksivni i nisu kopule. Pomoćne glagole trenutno prepoznajemo samo kada se nalaze neposredno ispred glagola.

Ne prepoznajemo futur u obliku da+prezent i ne prepoznajemo modalne glagole – ni jedna ni druga konstrukcija trenutno nisu unete u sistem.

Pravi objekat

Trenutno prepoznajemo imenički i zamenički pravi objekat, kao i varijantu sa upitnim pravim objektom (*Koga Sonja voli?*). Klasa za rečenični pravi objekat još uvek ne funkcioniše, dok pravi objekat u obliku enklitike dosad nismo obrađivali.

Pravi objekat trenutno prepoznajemo samo na poziciji iza glagola.

Nepravi objekat

Uspešno prepoznajemo imeničke i zameničke objekte, uključujući i upitnu varijantu nepravog objekta (*Kome daje lutku?*). Prepoznajemo nepravne objekte u obliku predložko-padežne konstrukcije, iako trenutno sve predloge tretiramo kao jedinice koje se unose zamenom, a ne kao dodatna sidra uz glagol. Trenutno ne prepoznajemo predložko-padežne konstrukcije u upitnom obliku (*U čemu spava Marko?*), enklitičke oblike nepravog objekta, kao ni rečenični nepravni objekat. Primer rezultata parsiranja za rečenicu *Kome Marko daje lutku?* – izvedeno stablo (Slika 191) i stablo izvođenja (Slika 192) dati su u nastavku.

Nepravi objekat prepoznajemo samo na poziciji iza glagola. U rečenicama u kojima postoje i pravi i nepravni objekat, prepoznajemo oba njihova međusobna redosleda.

Priloška dopuna

Od priloških dopuna trenutno prepoznamo priloge, predložko-padežne konstrukcije i brojne dopune. Prepoznamo pitanja u kojima su prilog, predložko-padežna konstrukcija ili brojna dopuna zamenjeni upitnom priloškom rečju (npr. *Gde Marko živi?*), ali ne i upitne forme u obliku predložko-padežne konstrukcije ili brojne dopune.

Priloške dopune trenutno prepoznamo samo na poziciji iza glagola.

Logički subjekat

Logički subjekat prepoznamo u imeničkom, zameničkom obliku, i kao upitnu zamenicu (*Koga boli glava?*). Ne prepoznamo enkliticke oblike logičkog subjekta.

Iako je za logički subjekat definisana glavna pozicija ispred glagola, prepoznamo ga i iza glagola.

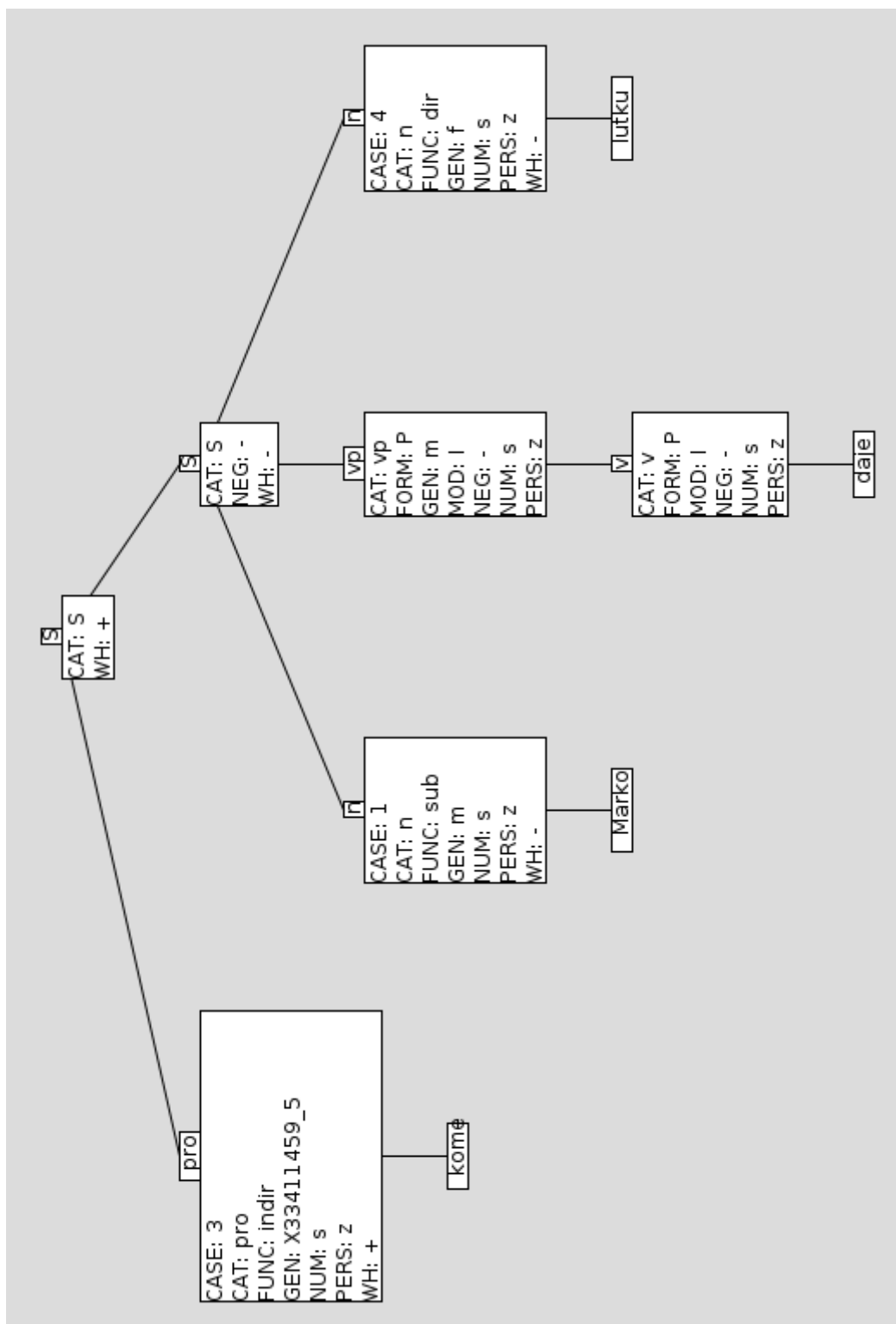
Predikativi

Imenski predikativ prepoznamo kao imenicu ili zamenicu. Kod pridevskih predikativa ispravno funkcioniše kongruencija sa glagolom. Od priloških predikativa trenutno prepoznamo samo priloge. Ni za jedan od predikativa trenutno nemamo definisan upitni oblik. Za predikative trenutno nemamo definisanu klasu koja omogućava prepoznavanje rečenica kao što je *Pametan je*, ali možemo da prepoznamo bezlične modele ovakve strukture, sa prilogom kao glagolskim delom predikata i subjektom koji nije prisutan u strukturi – *Dobro je*.

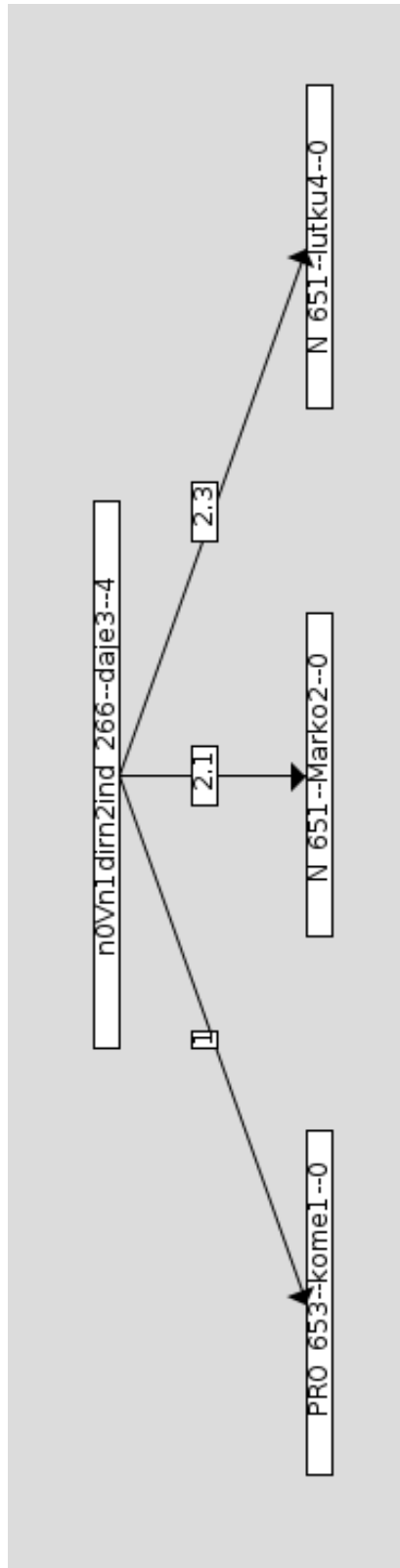
Primeri koje trenutno sistem ne prepoznaje mogu se podeliti na dve grupe. U jednu grupu spadaju oni za koje ne postoji tehnička ni jezička poteškoća i koji samo još uvek nisu obrađeni u sistemu. U ovu grupu spadaju negacija i prošlo i buduće vreme za kopulativne glagole, prepoznavanje primera bez subjekta za kopulativne glagole, prepoznavanje brojnih fraza, modalnih glagola, opštih pitanja i sl. U drugu grupu spadaju pojave koje izazivaju neku vrstu tehničke ili jezičke poteškoće, a za koje će biti potrebno značajnije modifikovati sistem i klase koje ih obrađuju. U ovu grupu spada obrada predložko-padežnih konstrukcija kao

upitnih fraza i kao predikativa, obrada konstrukcije *da*+prezent, tretman rečeničnih argumenata, tretman enklitika i tretman reda reči.

Kompiliranje gramatike ove veličine i sastava traje par sekundi preko jednog minuta (minut i tri sekunde). Dvosmislenost prilikom parsiranja je trenutno minimalna, i slobodno se može reći nepostojeća. Računamo na to da se dvosmislenost u određenoj meri povećati kada se u leksikone unese veći broj jedinica i kada se u metagramatici poveća broj klasa.



Slika 191 Izlaz parsera TuLiPA za rečenicu *Kome Marko daje lutku?* – izvedeno stablo



Slika 192 Izlaz parsera TuLiPA za rečenicu Kome Marko daje lutku? – stablo izvođenja

6 ZAKLJUČAK I DALJI RAD

6.1 Zaključak

U ovom radu smo prikazali proces izrade i rezultate izrade formalne gramatike srpskog jezika SrpTAG i njegove metagramatike.

U prvom delu rada smo dali uvod u obradu prirodnih jezika kao disciplinu, fokusirajući se na obradu srpskog jezika. Prikazali smo metode, alate i resurse koji se trenutno koriste za srpski jezik i time napravili uvod za gramatiku za automatsku obradu teksta koja je dalje izrađivana u radu.

U drugom poglavlju smo dali prikaz pojma formalne gramatike, kao i prikaz konkretne formalne gramatike koju smo nadalje koristili u radu – FBLTAG. U drugom delu poglavlja uvodimo pojam metagramatike, kao i konkretne metagramatike koju smo koristili u daljem radu – XMG.

U trećem poglavlju smo prikazali proces izrade formalne gramatike srpskog jezika – SrpTAG, počevši od morfoloških odluka, preko odluka o gramatici prirodnog jezika koja nam je služila kao uzor, pa sve do odluka o izboru struktura koje bi najbolje predstavile fenomene u srpskom jeziku. SrpTAG smo predstavili detaljno, preko glagola i njegovih argumenata, pa sve do rečeničnih modela.

U četvrtom poglavlju smo na osnovu rezultata dobijenih u trećem poglavlju izradili metagramatiku za srpski jezik. Kao i SrpTAG, i metagramatiku smo prikazali detaljno, po argumentima i rečeničnim modelima.

Peto poglavlje smo posvetili primeni metagramatike koju smo definisali u poglavlju četiri. Predstavili smo leksikone koji su neophodni za njenu primenu i dali evaluaciju trenutnog stanja metagramatike. Opisali smo koliko uspešno prepoznavamo svaki od segmenata definisanih u metagramatici, dali vreme potrebno za njeno kompiliranje i rezultate koji se tiču dvosmislenosti gramatike.

Možemo zaključiti da je cilj koji je bio postavljen – da se obrade osnove gramatike srpskog jezika – ispunjen. Gramatika, odnosno metagramatika, koju smo napravili u ovom radu mogu se uspešno koristiti za prepoznavanje osnovnih (deklarativna rečenica u svakom od rečeničnih modela) i izvedenih struktura srpskog jezika (pitanje, negacija, pasiv, reflektivni pasiv, obezličavanje). Data gramatika isto tako ima i generativni kvalitet. U stanju u kome je trenutno, ova gramatika ne samo da prepoznaje ispravne strukture

srpskog jezika, već i odbacuje one koje nisu gramatične, odnosno – ne generiše i ne prepoznaje negramatične strukture srpskog jezika.

6.2 Dalji rad

Rad na formalnoj gramatici bilo kog jezika nije nimalno lak zadatak, ali moramo reći da je on dodatno otežan kada se radi sa morfološki bogatim jezikom sa slobodnim redom reči kakav je srpski. Naš cilj u ovom radu bio je da damo osnove ove formalne gramatike, i smernice za to u kom pravcu je dalje razvijati. Samim tim, ostalo je još puno materijala za dalji rad. Navešćemo ovde samo neke od daljih pravaca:

- osposobljavanje svih realizacija pojedinih rečeničnih funkcija (npr. realizacija predloške fraze kao predikativa, osposobljavanje klasa za rečenične argumente, unošenje enklitika)
- dodavanje i ostalih složenih glagolskih vremena i načina (potencijal, futur 2, futur u obliku *da+present*)
- bavljenje redom reči, pre svega redom reči enklitika
- unošenje modifikatora fraza (prideva i priloga) i rečeničnih modifikatora (priloga, predloško-padežnih konstrukcija)
- proširivanje domena pitanja na opšte (*da li, je li*) pitanje i ostale tipove pitanja koji ne predstavljaju transformaciju osnovne rečenične strukture.
- proširivanje leksikona poluautomatskom transformacijom elektronskog rečnika srpskog jezika.

Dalji razvoj ovog resursa omogućio bi potpunu automatsku sintaksičku analizu teksta, i kao takav, znatno bi doprineo razvoju mašinskog prevođenja sa srpskog jezika i na srpski jezik. Ovaj resurs bi mogao da bude od velikog značaja i studentima srpskog jezika, kako onima kojima je srpski maternji jezik, tako i onima koji ga uče kao strani. Nadamo se zato da će rezultati ovog rada poslužiti naučnicima kao koristan alat, kao osnova i

inspiracija za njihov dalji rad, kao i da će biti naučnika koji će želiti da se uključe u profinjavanje dosad napravljene SrpTAG.

7 LITERATURA

7.1 Korišćena literatura

- Abeillé, A. (1990). Lexical and Syntactic Rules in a Tree Adjoining Grammar. In *Proceedings of the 28th annual meeting on Association for Computational Linguistics*. Pittsburgh: Association for Computational Linguistics.
- Abeillé, A. (2002). *Une grammaire électronique du français*. Paris: CNRS.
- Abeillé, A. (2007). *Les grammaires d'unification*. Paris: Lavoisier, Hermes Science.
- Aho, A. V. (1968). Indexed Grammars—An Extension of Context-Free Grammars. *Journal of the ACM (JACM)*, 15(4), 647–671.
- Balvet, A., Stošić, D., & Miletić, A. (2014). TALC-Sef: a Manually-Revised POS-Tagged Literary Corpus in Serbian, English and French. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)* (pp. 4105–4110). Reykjavik: European Language Resources Association (ELRA).
- Berwick, R. C. (1984). Strong Generative Capacity, Weak Generative Capacity, and Modern Linguistic Theories. *Computational Linguistics*, 10(3–4), 189–202.
- Candito, M.-H. (1996). A Principle-Based Hierarchical Representation of LTAGs. In *Proceedings of the 16th International Conference on Computational linguistics COLING96* (pp. 194–199). Copenhagen: Association for Computational Linguistics.
- Candito, M.-H. (1999). *Organisation modulaire et paramétrable de grammaires électroniques lexicalisées: Application au français et à l'italien*. Ph.D. thesis. Paris: Université Paris 7.
- Chomsky, N. (1956). Three Models for the Description of Language. In *I. R. E. Transactions on Information Theory* (Vol. IT-2, pp. 113–123).
- Chomsky, N. (1963). Formal Properties of Grammars. In *Handbook of Mathematical Psychology*. New York: Wiley.
- Chomsky, N. (1965). *Aspects of the Theory of Syntax*. Cambridge, Massachusetts: MIT press.
- Chomsky, N. (1981). *Lectures on Government and Binding*. Dordrecht: Foris Publications.
- Constant, M., Krstev, C., & Vitas, D. (2015). Hybrid Lexical Tagging in Serbian. In Z. Vetulani & J. Mariani (Eds.), *Proceedings of 7th Language & Technology Conference* (pp. 461–465). Poznań: Fundacja Uniwersytetu im. A. Mickiewicz.
- Crabbé, B. (2005). *Représentation informatique de grammaires fortement lexicalisées: Application à la grammaire d'arbres adjoints*. Ph.D. thesis. Paris: Université Nancy 7.

- Crabbé, B., & Duchier, D. (2005). Metagrammar Redux. *Lecture Notes in Computer Science*, 3438, 32–47.
- Crabbé, B., Duchier, D., Gardent, C., Le Roux, J., & Parmentier, Y. (2013). XMG: eXtensible MetaGrammar. *Computational Linguistics*, 39(3), 591–629.
- Dalrymple, M. (2001). *Lexical Functional Grammar*. New York: Academic Press.
- de La Clergerie, É. V. (2005). DyALog: A Tabular Logic Programming Based Environment for NLP. In *Proceedings of International Workshop on Constraint Solving and Language Processing (CSLP04)* (pp. 18–33). Barcelona.
- Dobrić, N. (2012). Savremeni jezički korpusi na Zapadnom Balkanu - istorijat, trenutno stanje i budućnost. *Slavistična Revija*, 60(4), 677–692.
- Doran, C., Egedi, D., Hockey, B. A., Srinivas, B., & Zaidel, M. (1994). XTAG system - A Wide Coverage Grammar for English. In *Proceedings of the 15th conference on Computational linguistics* (Vol. 2, pp. 922–928). Kyoto: Association for Computational Linguistics.
- Đorđević, B. (2012). Automatic Recognition of Composite Verb Forms in Serbian. Retrieved March 26, 2017, from <http://ceur-ws.org/Vol-920/p89-djordjevic.pdf>
- Đorđević, B. (2014). Initial Steps in Building Serbian Treebank: Morphological Annotation. In G. Pavlović-Lažetić, C. Krstev, I. Obradović, & D. Vitas (Eds.), *Natural Language Processing for Serbian: Resources and Applications* (pp. 41–53). Belgrade: University of Belgrade, Faculty of Mathematics.
- Erbach, G., & Uszkoreit, H. (1990). Grammar Engineering: Problems and Prospects. In *In Proceedings of ANLP-92* (pp. 1–20). Trento: Association for Computational Linguistics.
- Gazdar, G. (1988). Applicability of Indexed Grammars to Natural Languages. In *Natural Language Parsing and Linguistic Theories* (pp. 69–94). Dordrecht: Springer Netherlands.
- Gazdar, G., Klein, E., Pullum, G. K., & Sag, I. A. (1985). *Generalized phrase structure grammar*. Massachusetts: Blackwell Publishing and Harvard University Press.
- Gerdes, K. (2002). DTAG? In *Proceedings of the Sixth International Workshop on Tree Adjoining Grammar and Related Frameworks (TAG+6)* (pp. 242–251). Università di Venezia.
- Gesmulto, A., & Samardžić, T. (2012). Lemmatizing Serbian as Category Tagging with

- Bidirectional Sequence Classification. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)* (pp. 2103–2106).
- Gross, M. (1975). *Méthodes en syntaxe*. Paris: Hermann.
- Gross, M. (1979). On the Failure of Generative Grammar. *Language*, 55(4), 859–885.
- Gucul-Milojević, S. (2010). Vlastita imena u ekstrakciji informacija. *INFOteka*, 11(1), 47–58.
- Han, C., Yoon, J., Kim, N., & Palmer, M. (2000). *A Feature-Based Lexicalized Tree Adjoining Grammar for Korean*. Philadelphia: University of Pennsylvania, Institute for Research in Cognitive Science.
- Hutchins, J. (2005). The First Public Demonstration of Machine Translation: the Georgetown-IBM System , 7th January 1954. Retrieved October 26, 2016, from www.hutchinsweb.me.uk/GU-IBM-%0A2005.pdf
- Jaćimović, J. (2016). *Automatsko prepoznavanje i normalizacija vremenskih izraza u nestrukturiranim novinskim i medicinskim tekstovima na srpskom jeziku*. Ph.D. thesis. Beograd: Filološki fakultet, Univerzitet u Beogradu.
- Joshi, A. K., Levy, L. S., & Takahashi, M. (1975). Tree Adjunct Grammars. *Journal of Computer and System Sciences*, 10(1), 136–163.
- Jurafsky, D., & Martin, J. H. (2000). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Upper Saddle River, New Jersey: Prentice-Hall, Inc., Pearson Education International.
- Kallmeyer, L., Lichte, T., Maier, W., Parmentier, Y., & Dellert, J. (2008). Developing a TT-MCTAG for German with an RCG-based Parser. In *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)* (pp. 782–789). Marrakech: LREC.
- Kallmeyer, L., Lichte, T., Maier, W., Parmentier, Y., Dellert, J., & Evang, K. (2008). TuLiPA : Towards a Multi-Formalism Parsing Environment for Grammar Engineering. In S. Clark & T. H. King (Eds.), *Proceedings of the Workshop on Grammar Engineering Across Frameworks (Coling 2008)* (pp. 1–8). Manchester: Association for Computational Linguistics.
- Kallmeyer, L., & Yoon, S. (2004). Tree-Local MCTAG with Shared Nodes : Word Order Variation in German and Korean. *Traitement Automatique Des Langues TAL*, 45(3), 40–47.

- Kaplan, R., & Bresnan, J. (1982). *Lexical-Functional Grammar: A Formal System for Grammatical Representation*. In *The mental representation of grammatical relations*. Cambridge, Massachusetts: MIT Press.
- Kay, M. (1984). *Functional Unification Grammar: A Formalism for Machine Translation*. In *Proceedings of the 10th International Conference on Computational Linguistics and 22nd annual meeting on Association for Computational Linguistics*. Stanford: Association for Computational Linguistics.
- Kostić, A. (2003). Elektronski korpus srpskog jezika Đorđa Kostića. *Zbornik Matice Srpske Za Slavistiku*, 64, 260–264.
- Kostić, Đ. (2001). *Kvantitativni opis strukture srpskog jezika - Korpus srpskog jezika*. Beograd: Institut za eksperimentalnu fonetiku i patologiju govora i Laboratorija za eksperimentalnu psihologiju Filozofskog fakulteta u Beogradu.
- Kracht, M. (2003). *The Mathematics of Language*. Berlin, New York: Mouton de Gruyter.
- Kroch, A., & Joshi, A. K. (1985). *The Linguistic Relevance of Tree Adjoining Grammar. Technical Reports (CIS)*. Philadelphia: Department of Computer & Information Science, University of Pennsylvania.
- Krstev, C. (2008). *Processing of Serbian – Automata, Texts and Electronic Dictionaries*. Belgrade: Faculty of Philology, University of Belgrade.
- Krstev, C., Stanković, R., Obradović, I., & Lazić, B. (2015). Terminology Acquisition and Description Using Lexical Resources and Local Grammars. In T. Poibeau & P. Faber (Eds.), *Proceedings of the 11th Conference on Terminology and Artificial Intelligence* (pp. 81–89). Granada, Spain: LexiCon (Universidad de Granada).
- Krstev, C., Utvić, M., & Jaćimović, J. (2015). Ako koza laže, rog ne laže - gde su i ko su žene u srpskoj dnevnoj štampi. *Knjiženstvo - Časopis Za Studije Književnosti, Roda I Kulture*, 5.
- Krstev, C., & Vitas, D. (2005). Corpus and Lexicon - Mutual Incompleteness. In P. Danielsson & M. Wagenmakers (Eds.), *Proceedings of the Corpus Linguistics Conference*. Birmingham.
- Krstev, C., & Vitas, D. (2009). An Aligned English-Serbian Corpus. In N. Tomović & J. Vujić (Eds.), *ELLSIIR Proceedings (English Language and Literature Studies: Image, Identity, Reality)* (pp. 495–508). Belgrade: Faculty of Philology, University of Belgrade.
- Krstev, C., Vitas, D., & Erjavec, T. (2004). MULTEXT-East Resources for Serbian. In T. Erjavec

- & J. Z. Gros (Eds.), *Zbornik 7. mednarodne multikonference Informacijska družba IS 2004, Jezikovne tehnologije*. Ljubljana: Institut Jožef Stefan.
- Krstev C., Stanković R., Vitas D., O. I. (2006). WS4LR: A Workstation for Lexical Resources. In *Proc. of the 5th International Conference on Language Resources and Evaluation, LREC 2006* (pp. 1692–1697). Genoa, Italy.
- Kunz, P. (1991). *Object Oriented Programming*. Stanford: Stanford Linear Accelerator Center, Stanford University.
- Le Roux, Y. P. J. (2005). XMG: a Multi-Formalism Metagrammatical Framework. *Loria.Fr*, 1–12.
- Levine, R. D., & Meurers, W. D. (2006). Head-Driven Phrase Structure Grammar. In *Encyclopedia of language and linguistics* (pp. 1–24). Boston: Elsevier.
- Lopez, P. (2000). Extended Partial Parsing for Lexicalized Tree Grammars. In *Sixth International Workshop on Parsing Technologies IWPT 2000*. Trento.
- Miller, P. H. (1991). Scandinavian Extraction Phenomena Revisited: Weak and Strong Generative Capacity. *Linguistics and Philosophy*, 14(1), 101–113.
- Mitkov, R. (ed. . (2005). *The Oxford Handbook of Computational Linguistics*. Oxford: Oxford University Press.
- Moskvljević Popović, J. (2007). *Ogledi o glagolskoj potkategorizaciji*. Beograd: Čigoja štampa.
- Mrazović, P. (2009). *Gramatika srpskog jezika za strance*. Novi Sad: Izdavačka knjižarnica Zorana Stojanovića.
- Müller, S. (2016). *Grammatical Theory: From Transformational Grammar to Constraint-Based Approaches. Foundations of Cognitive Science*. Berlin: Language Science Press.
- Pelletier, F. J. (1980). The Generative Power of Rule Orderings in Formal Grammars. *Linguistics*, 1–2(18), 17–72.
- Pereira, F. C. N., & Warren, D. H. D. (1980). Definite Clause Grammars for Language Analysis—A Survey of the Formalism and a Comparison with Augmented Transition Networks. *Artificial Intelligence*, 13, 231–278.
- Peters, P. S., & Ritchie, R. W. (1973). On the Generative Power of Transformational Grammars. *Information Sciences*, 6(C), 49–83.

- Petitjean, S. (2013). XMG: A Modular MetaGrammar Compiler. In *Student Sessions at the European Summer School in Logic, Language and Information*. Berlin, Heidelberg: Springer.
- Piper, P., Antić, I., Ružić, V., Tanasić, S., Popović, L., & Tošović, B. (2005). *Sintaksa savremenog srpskog jezika-prosta rečenica*. Beograd: Institut za srpski jezik SANU.
- Pollard, C., & Sag, I. A. (1994). *Head-Driven Phrase Structure Grammar*. Chicago: University of Chicago Press.
- Popović, Z. (2010). Programi za etiketiranje teksta na srpskom jeziku. *INFOteka*, 11(2), 19–36.
- Pullum, G. K., & Gazdar, G. (1982). Natural Languages and Context-Free Languages. *Linguistics and Philosophy*, 4(4), 471–504.
- Rambow, O., & Lee, Y.-S. (1994). Word Order Variation and Tree Adjoining Grammar. *Computational Intelligence*, 10(4), 386–400.
- Resnik, P. (1992). Probabilistic Tree-Adjoining Grammar as a Framework for Statistical Natural Language Processing. In *Proceedings of the 14th conference on Computational linguistics COLING '92* (pp. 418–424). Nantes: Association for Computational Linguistics.
- Savitch, W. J. (1987). Context-Sensitive Grammar and Natural Language Syntax. In W. J. Savitch, E. Bach, W. E. Marsh, & G. Safran-Naveh (Eds.), *The Formal complexity of natural language* (pp. 358–368). Dordrecht: Springer Netherlands.
- Schabes, Y. (1990). *Mathematical and Computational Aspects of Lexicalized Grammars*. Ph.D. Dissertation. Philadelphia: University of Pennsylvania.
- Schabes, Y., Abeillé, A., & Joshi, A. K. (1988). Parsing Strategies with “Lexicalized” Grammars: Application to Tree Adjoining Grammars. In D. Vargha (Ed.), *Proceedings of the 12th Conference on Computational Linguistics* (pp. 578–583). Budapest: Association for Computational Linguistics.
- Schuler, W., Chiang, D., & Dras, M. (2000). Multi-Component TAG and Notions of Formal Power. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics* (p. 455). Hong Kong: Association for Computational Linguistics.
- Shieber, S. M. (1985). Evidence Against the Context-Freeness of Natural Language.

Linguistics and Philosophy, 8, 333–343.

Shieber, S. M. (2003). *An Introduction to Unification-Based Approaches to Grammar*. Brookline, Massachusetts: Microtome Publishing.

Shieber, S. M., & Schabes, Y. (1990). Synchronous Tree-Adjoining Grammars. In H. Karlgren (Ed.), *Proceedings of the 13th conference on Computational linguistics* (Vol. 3, pp. 253–258). Helsinki: Association for Computational Linguistics.

Shieber, S. M., Uszkoreit, H., Pereira, F., Robinson, J., & Tyson, M. (1983). The Formalism and Implementation of PATR-II. *Research on Interactive Acquisition and Use of Knowledge*, 39–79.

Stankovic, R., Krstev, C., Obradović, I., Lazić, B., & Trtovac, A. (2016). Rule-Based Automatic Multi-Word Term Extraction and Lemmatization. In *Proceedings of the 10th International Conference on Language Resources and Evaluation, LREC 2016* (pp. 507–514). Portorož, Slovenia: LREC.

Stanković, R., Obradović, I., & Utvić, M. (2014). Developing Termbases under the TBX Standard. In G. Pavlović-Lažetić, C. Krstev, I. Obradović, & D. Vitas (Eds.), *Natural Language Processing for Serbian: Resources and Applications* (pp. 12–26). Belgrade: University of Belgrade, Faculty of Mathematics.

Stanojčić, Ž., & Popović, L. (1997). *Gramatika srpskoga jezika: udžbenik za I, II, III i IV razred srednje škole*. Beograd: Zavod za udžbenike i nastavna sredstva.

Tesnière, L. (1959). *Éléments de syntaxe structurale*. Paris: Librairie C. Klincksieck.

Trtovac, A., & Andonovski, J. (2014). Enrichment of Morphological Dictionary of MWUs from Library and Information Science. In G. Pavlović-Lažetić, C. Krstev, I. Obradović, & D. Vitas (Eds.), *Natural Language Processing for Serbian: Resources and Applications* (pp. 27–40). Belgrade: University of Belgrade, Faculty of Mathematics.

Utvić, M. (2011). Annotating the Corpus of Contemporary Serbian. *INFOtheca*, 12(2), 36a–47a.

Utvić, M. (2013). *Izgradnja referentnog korpusa savremenog srpskog jezika*. Ph.D. thesis. Beograd: Filološki fakultet.

Vijay-Shanker, K., & Joshi, A. K. (1988). Feature Structures Based Tree Adjoining Grammar. In D. Vargha (Ed.), *Proceedings of COLING '88* (pp. 714–719). Budapest: Association for

Computational Linguistics.

Vijay-Shanker, K., & Joshi, A. K. (1991). *Unification-Based Tree Adjoining Grammars. Technical Reports (CIS)*. Philadelphia: Department of Computer & Information Science, University of Pennsylvania.

Vijay-Shanker, K., & Schabes, Y. (1992). Structure Sharing in Lexicalized Tree-Adjoining Grammars. In *Proceedings of the Fifteenth International Conference on Computational Linguistics (COLING'92)* (pp. 205–211). Kyoto: Association for Computational Linguistics.

Vitas, D. (2006). *Prevodioci i interpretatori*. Beograd: Matematički fakultet.

Vitas, D., Koeva, S., Krstev, C., & Obradović, I. (2008). Tour du monde through the dictionaries. In M. Constant, T. Nakamura, M. De Gioia, & S. Vecchiato (Eds.), *Actes du 27eme Colloque International sur le Lexique et la Grammaire* (pp. 249–256). L'Aquila: Universite Paris-Est, Institut Gaspard-Monge.

Vitas, D., & Krstev, C. (2003). Composite Tense Recognition and Tagging in Serbian. In T. Erjavec & D. Vitas (Eds.), *Proceedings of the Workshop on Morphological Processing of Slavic Languages: 10th Conference of the European Chapter, EACL 2003* (pp. 54–61). Budapest: Association for Computational Linguistics.

Vitas, D., & Krstev, C. (2006). Literature and Aligned Texts. *Readings in Multilinguality*, 148–155.

Vitas, D., Krstev, C., & Laporte, E. (2006). Preparation and Exploitation of Bilingual Texts. *Lux Coreana*, 1, 110–132.

Weir, D. J. (1988). *Characterizing Mildly Context-Sensitive Grammar Formalisms*. Ph.D. thesis. University of Pennsylvania.

XTAG Research Group. (2001). *A Lexicalized Tree Adjoining Grammar for English*. Philadelphia: IRCS, University of Pennsylvania.

7.2 Referentne veb lokacije

Aurora kodni raspored

<http://korpus.matf.bg.ac.rs/prezentacija/uputstvo.html>

Btagger

<http://clcl.unige.ch/SOFTWARE.html>

Francuska metagramatika FrenchTAG

<http://subversion.renater.fr/xmg/trunk/METAGRAMMARS/FrenchTAG/>

Korpus *Put oko sveta za 80 dana* Žila Verna izrađen u okviru projekta MULTEXT-East

<https://www.clarin.si/repository/xmlui/handle/11356/1043>

Morfosintaksički opis unutar MULTEXT-East projekta

<http://nl.ijs.si/ME/V4/msd/html/>

MXPOST

http://www.inf.ed.ac.uk/resources/nlp/local_doc/MXPOST.html

Nooj

<http://www.nooj-association.org>

Projekat MULTEXT-East

<http://nl.ijs.si/ME/>

Rule Based Tagger

<http://www.ling.gu.se/~lager/mogul/brill-tagger/>

SVMTool

<http://www.cs.upc.edu/~nlp/SVMTool/>

TnT

<http://www.coli.uni-saarland.de/~thorsten/tnt/>

TreeTagger

<http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/>

Unitex

<http://www-igm.univ-mlv.fr/~unitex/>

Uputstvo za instalaciju TuLiPA

<https://sourcesup.renater.fr/tulipa/overview.html>

Uputstvo za instalaciju XMG

http://dokufarm.phil.hhu.de/xmg/?animal=xmg#getting_started

8 PRILOZI

8.1 DTD za leksikon lema i morfološki leksikon

```
<!-- Kompilirana metagramatika -->
<!ELEMENT mcgrammar (morphs?|lemmas?|(entry* , mcset*))>
<!-- Potencijalni leksički unosci (lema ili morfološki oblik) -->
<!ELEMENT morphs (morph*)>
<!ELEMENT lemmas (lemma*)>
<!-- Morfološki unosci u leksikonu -->
<!ELEMENT morph (lemmaref+)>
<!ATTLIST morph lex CDATA #REQUIRED>
<!-- Lema kao unosak -->
<!ELEMENT lemma (anchor+)>
<!ATTLIST lemma name CDATA #REQUIRED
               cat CDATA #REQUIRED>
<!-- Reference na leme u morfološkim unoscima -->
<!ELEMENT lemmaref (fs?)>
<!ATTLIST lemmaref name CDATA #REQUIRED
                   cat CDATA #REQUIRED>
<!-- Ograničenja na usidrenje kod lema -->
<!ELEMENT anchor (filter?, coanchor*, equation*, sem?)>
<!ATTLIST anchor tree_id CDATA #REQUIRED>
<!-- Filteri za odabir stabala -->
<!ELEMENT filter (fs?)>
<!-- Interfejs između sintakse i semantike u leksikonu lema -->
<!ELEMENT sem (semclass*)>
<!-- Instancijacija semantičke klase -->
<!ELEMENT semclass name (args)>
<!ATTLIST semclass name CDATA #REQUIRED>
<!-- Lista argumenata za instancijaciju -->
<!ELEMENT args (f*)>
```

```
<!-- Leksička ograničenja kod sidara -->

<!ELEMENT coanchor (lex+)>
<!ATTLIST coanchor node_id CDATA #REQUIRED>

<!-- Leksička vrednost -->

<!ELEMENT lex (#PCDATA)>

<!-- Unifikaciona ograničenja kod sidara -->

<!ELEMENT equation (fs)>
<!ATTLIST equation type ( top | bot ) #REQUIRED
node_id CDATA #REQUIRED>

<!-- Interfejsi -->

<!ELEMENT interface (fs?)>
```

8.2 Leksikon lema

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<!DOCTYPE mcgrammar SYSTEM "xmg-mctag.dtd,xml">

<mcgrammar>
  <lemmas>

<!-- GLAGOLI -->

  <lemma name="biti" cat="V" >
    <anchor tree_id="family[@name=Cop]" >
      </anchor>
    </lemma>

  <lemma name="biti" cat="V" >
    <anchor tree_id="family[@name=AuxProslo]" >
      </anchor>
    </lemma>

  <lemma name="smrkavati" cat="V" >
    <anchor tree_id="family[@name=VC1]" >
      <coanchor node_id="se" cat="Cl">
        <lex>se</lex>
      </coanchor>
    </anchor>
    </lemma>

  <lemma name="smrkavati" cat="V" >
    <anchor tree_id="family[@name=VC1]" >
      <coanchor node_id="se" cat="Cl">
        <lex>se</lex>
      </coanchor>
    </anchor>
    </lemma>

  <lemma name="voleti" cat="V" >
    <anchor tree_id="family[@name=n0Vn1dir]" >
      <filter>
        <fs>
```



```

        <f name="passive" >
          <sym value="+" />
        </f>
        <f name="bez1" >
          <sym value="-" />
        </f>
      </fs>
    </filter>
  </anchor>
</lemma>

<lemma name="voleti"  cat="V" >
  <anchor tree_id="family[@name=n0Vn1dirNer]" >
    <filter>
      <fs>
        <f name="passive" >
          <sym value="+" />
        </f>
        <f name="bez1" >
          <sym value="-" />
        </f>
      </fs>
    </filter>
  </anchor>
</lemma>

<lemma name="spavati"  cat="V" >
  <anchor tree_id="family[@name=n0V]" >
    </anchor>
  </lemma>

<lemma name="nemati"  cat="V" >
  <anchor tree_id="family[@name=n0Vn1dir]" >
    <filter>
      <fs>
        <f name="passive" >
          <sym value="-" />
        </f>
        <f name="bez1" >
          <sym value="-" />

```

```

        </f>
      </fs>
    </filter>
  </anchor>
</lemma>

<lemma name="zxiveti"  cat="V" >
  <anchor tree_id="family[@name=n0Vpn1ind]" >
    </anchor>
  </lemma>

<lemma name="zxiveti"  cat="V" >
  <anchor tree_id="family[@name=n0Vad1]" >
    </anchor>
  </lemma>

<lemma name="stideti"  cat="V" >
  <anchor tree_id="family[@name=n0ClVn1ind]" >
    <coanchor node_id="se"  cat"C1">
      <lex>se</lex>
    </coanchor>
  </anchor>
</lemma>

<lemma name="svitati"  cat="V" >
  <anchor tree_id="family[@name=V]" >
    </anchor>
  </lemma>

<lemma name="igrati"  cat="V" >
  <anchor tree_id="family[@name=n0Vn1]" >
    <filter>
      <fs>
        <f name="passive" >
          <sym value="+" />
        </f>
        <f name="bez1" >
          <sym value="-" />
        </f>
      </fs>
    </filter>
  </lemma>

```

```

    </anchor>
  </lemma>

<lemma name="sxaliti"  cat="V" >
  <anchor tree_id="family[@name=n0ClV]" >
    <coanchor node_id="se"  cat"C1">
      <lex>se</lex>
    </coanchor>
  </anchor>
</lemma>

<lemma name="hteti"  cat="V" >
  <anchor tree_id="family[@name=AuxBduce]" >
  </anchor>
</lemma>

<lemma name="boleti"  cat="V" >
  <anchor tree_id="family[@name=n11subVn0]" >
  </anchor>
</lemma>

<lemma name="pomagati"  cat="V" >
  <anchor tree_id="family[@name=n0Vn1ind]" >
    <filter>
      <fs>
        <f name="passive" >
          <sym value="+" />
        </f>
        <f name="bez1" >
          <sym value="-" />
        </f>
      </fs>
    </filter>
  </anchor>
</lemma>

<lemma name="smetati"  cat="V" >
  <anchor tree_id="family[@name=n11subVs0]" >
  </anchor>
</lemma>

```

```

<lemma name="davati" cat="V" >
  <anchor tree_id="family[@name=n0Vn1dirn2ind]" >
    <filter>
      <fs>
        <f name="passive" >
          <sym value="+" />
        </f>
        <f name="bez1" >
          <sym value="-" />
        </f>
        <f name="bez2" >
          <sym value="+" />
        </f>
      </fs>
    </filter>
  </anchor>
</lemma>

```

```

<lemma name="stavlxati" cat="V" >
  <anchor tree_id="family[@name=n0Vn1dirad2]" >
    <filter>
      <fs>
        <f name="passive" >
          <sym value="+" />
        </f>
        <f name="bez1" >
          <sym value="-" />
        </f>
        <f name="bez2" >
          <sym value="+" />
        </f>
      </fs>
    </filter>
  </anchor>
</lemma>

```

```

<lemma name="cyesxlxati" cat="V" >
  <anchor tree_id="family[@name=n0Vn1dir]" >
    <filter>
      <fs>
        <f name="passive" >

```

```

        <sym value="+" />
    </f>
    <f name="bez1" >
        <sym value="-" />
    </f>
</fs>
</filter>
</anchor>
</lemma>

```

```

<!-- IMENICE -->

```

```

<lemma name="Sonxa" cat="N" >
    <anchor tree_id="family[@name=N]" >
        </anchor>
</lemma>

<lemma name="Sonxa" cat="N" >
    <anchor tree_id="family[@name=n0vN]" >
        </anchor>
</lemma>

<lemma name="Marko" cat="N" >
    <anchor tree_id="family[@name=N]" >
        </anchor>
</lemma>

<lemma name="Marko" cat="N" >
    <anchor tree_id="family[@name=n0vN]" >
        </anchor>
</lemma>

<lemma name="Dyon" cat="N" >
    <anchor tree_id="family[@name=N]" >
        </anchor>
</lemma>

<lemma name="Dyon" cat="N" >
    <anchor tree_id="family[@name=n0vN]" >
        </anchor>

```

```

</lemma>

<lemma name="lutka" cat="N" >
  <anchor tree_id="family[@name=N]" >
    </anchor>
</lemma>

<lemma name="krevet" cat="N" >
  <anchor tree_id="family[@name=N]" >
    </anchor>
</lemma>

<lemma name="krevet" cat="N" >
  <anchor tree_id="family[@name=n0vN]" >
    </anchor>
</lemma>

<lemma name="glava" cat="N" >
  <anchor tree_id="family[@name=N]" >
    </anchor>
</lemma>

<lemma name="glava" cat="N" >
  <anchor tree_id="family[@name=n0vN]" >
    </anchor>
</lemma>

<!-- ZAMENICE -->

<lemma name="ona" cat="PRO" >
  <anchor tree_id="family[@name=PRO]" >
    </anchor>
</lemma>

<lemma name="ona" cat="PRO" >
  <anchor tree_id="family[@name=n0vN]" >
    </anchor>
</lemma>

<lemma name="on" cat="PRO" >
  <anchor tree_id="family[@name=PRO]" >

```

```

    </anchor>
</lemma>

<lemma name="on"  cat="PRO" >
    <anchor tree_id="family[@name=n0vN]" >
        </anchor>
    </lemma>

<lemma name="ovo"  cat="PRO" >
    <anchor tree_id="family[@name=PRO]" >
        </anchor>
    </lemma>

<lemma name="ovo"  cat="PRO" >
    <anchor tree_id="family[@name=n0vN]" >
        </anchor>
    </lemma>

<lemma name="ko"  cat="PRO" >
    <anchor tree_id="family[@name=PRO]" >
        </anchor>
    </lemma>

<lemma name="ko"  cat="PRO" >
    <anchor tree_id="family[@name=n0vN]" >
        </anchor>
    </lemma>

<lemma name="sxta"  cat="PRO" >
    <anchor tree_id="family[@name=PRO]" >
        </anchor>
    </lemma>

<lemma name="sebe"  cat="PRO" >
    <anchor tree_id="family[@name=PRO]" >
        </anchor>
    </lemma>

<!-- PRIDEVI -->

<lemma name="pametan"  cat="A" >

```

```

    <anchor tree_id="family[@name=n0vA]" >
      </anchor>
</lemma>

<lemma name="kakav" cat="A" >
  <anchor tree_id="family[@name=n0vA]" >
    <filter>
      <fs/>
    </filter>
  </anchor>
</lemma>

<!-- ENKLITIKE -->

<lemma name="se" cat="Cl" >
  <anchor tree_id="family[@name=Se]" >
    <filter>
      <fs/>
    </filter>
  </anchor>
</lemma>

<!-- PREDLOZI -->

<lemma name="u" cat="p" >
  <anchor tree_id="family[@name=P]" >
    <filter>
      <fs/>
    </filter>
  </anchor>
</lemma>

<!-- PRILOZI -->

<lemma name="ovde" cat="ADV" >
  <anchor tree_id="family[@name=ADV]" >
    <filter>
      <fs/>
    </filter>
  </anchor>

```



```

</lemma>

<lemma name="ovde" cat="ADV" >
  <anchor tree_id="family[@name=n0vAd]" >
    <filter>
      <fs/>
    </filter>
  </anchor>
</lemma>

<lemma name="ovde" cat="ADV" >
  <anchor tree_id="family[@name=Adv]" >
    <filter>
      <fs/>
    </filter>
  </anchor>
</lemma>

<lemma name="gde" cat="ADV" >
  <anchor tree_id="family[@name=ADV]" >
    <filter>
      <fs/>
    </filter>
  </anchor>
</lemma>

<lemma name="gde" cat="ADV" >
  <anchor tree_id="family[@name=Adv]" >
    <filter>
      <fs/>
    </filter>
  </anchor>
</lemma>

<lemma name="gde" cat="ADV" >
  <anchor tree_id="family[@name=n0lsubvAd]" >
    <filter>
      <fs/>
    </filter>
  </anchor>
</lemma>

```

```

<lemma name="dobro" cat="ADV" >
  <anchor tree_id="family[@name=ADV]" >
    <filter>
      <fs/>
    </filter>
  </anchor>
</lemma>

<lemma name="dobro" cat="ADV" >
  <anchor tree_id="family[@name=Adv]" >
    <filter>
      <fs/>
    </filter>
  </anchor>
</lemma>

<lemma name="dobro" cat="ADV" >
  <anchor tree_id="family[@name=n01subvAd]" >
    <filter>
      <fs/>
    </filter>
  </anchor>
</lemma>

<!-- KOMPLEMENTIZATORI -->

<lemma name="da" cat="C" >
  <anchor tree_id="family[@name=C]" >
    <filter>
      <fs/>
    </filter>
  </anchor>
</lemma>

<!-- NEG -->

<lemma name="ne" cat="neg" >
  <anchor tree_id="family[@name=Neg]" >
    <filter>
      <fs/>

```

```
        </filter>
      </anchor>
    </lemma>

  </lemmas>
</mcgrammar>
```

8.3 Morfološki leksikon

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<!DOCTYPE mcgrammar SYSTEM "xmg-mctag.dtd,xml">

<mcgrammar>
  <morphs>

<!-- GLAGOLI -->

  <morph lex="je" >
    <lemmaref cat="V" name="biti" >
      <fs>
        <f name="form">
          <sym value="P"/>
        </f>
        <f name="mod">
          <sym value="I"/>
        </f>
        <f name="pers">
          <sym value="z"/>
        </f>
        <f name="num">
          <sym value="s"/>
        </f>
        <f name="cop">
          <sym value="+"/>
        </f>
        <f name="aux">
          <sym value="-"/>
        </f>
        <f name="neg">
          <sym value="-"/>
        </f>
      </fs>
    </lemmaref>
  </morph>

  <morph lex="je" >
    <lemmaref cat="V" name="biti" >
      <fs>
```

```

    <f name="form">
      <sym value="P"/>
    </f>
  <f name="mod">
    <sym value="I"/>
  </f>
  <f name="pers">
    <sym value="z"/>
  </f>
  <f name="num">
    <sym value="s"/>
  </f>
  <f name="cop">
    <sym value="-"/>
  </f>
  <f name="aux">
    <sym value="+"/>
  </f>
  <f name="neg">
    <sym value="-"/>
  </f>
</fs>
</lemmaref>
</morph>

<morph lex="su" >
  <lemmaref cat="V" name="biti" >
    <fs>
      <f name="form">
        <sym value="P"/>
      </f>
      <f name="mod">
        <sym value="I"/>
      </f>
      <f name="pers">
        <sym value="z"/>
      </f>
      <f name="num">
        <sym value="p"/>
      </f>
      <f name="cop">

```

```

        <sym value="+"/>
    </f>
<f name="aux">
    <sym value="-"/>
</f>
<f name="neg">
    <sym value="-"/>
</f>
</fs>
</lemmeref>
</morph>

<morph lex="su" >
    <lemmeref cat="V" name="biti" >
        <fs>
            <f name="form">
                <sym value="P"/>
            </f>
            <f name="mod">
                <sym value="I"/>
            </f>
            <f name="pers">
                <sym value="z"/>
            </f>
            <f name="num">
                <sym value="p"/>
            </f>
            <f name="cop">
                <sym value="-"/>
            </f>
            <f name="aux">
                <sym value="+"/>
            </f>
            <f name="neg">
                <sym value="-"/>
            </f>
        </fs>
    </lemmeref>
</morph>

<morph lex="cxe" >

```

```

<lemmeref cat="V" name="hteti" >
  <fs>
    <f name="form">
      <sym value="P"/>
    </f>
    <f name="mod">
      <sym value="I"/>
    </f>
    <f name="pers">
      <sym value="z"/>
    </f>
    <f name="num">
      <sym value="s"/>
    </f>
    <f name="cop">
      <sym value="-"/>
    </f>
    <f name="aux">
      <sym value="+"/>
    </f>
    <f name="neg">
      <sym value="-"/>
    </f>
  </fs>
</lemmeref>
</morph>

<morph lex="voleo" >
  <lemmeref cat="V" name="voleti" >
    <fs>
      <f name="mod">
        <sym value="G"/>
      </f>
      <f name="num">
        <sym value="s"/>
      </f>
      <f name="gen">
        <sym value="m"/>
      </f>
      <f name="neg">
        <sym value="-"/>
      </f>
    </fs>
  </lemmeref>
</morph>

```

```

        </f>
    </fs>
</lemmeref>
</morph>

<morph lex="volela" >
    <lemmeref cat="V" name="voleti" >
        <fs>
            <f name="mod">
                <sym value="G"/>
            </f>
            <f name="num">
                <sym value="s"/>
            </f>
            <f name="gen">
                <sym value="f"/>
            </f>
            <f name="neg">
                <sym value="-"/>
            </f>
        </fs>
    </lemmeref>
</morph>

<morph lex="voleti" >
    <lemmeref cat="V" name="voleti" >
        <fs>
            <f name="mod">
                <sym value="W"/>
            </f>
            <f name="neg">
                <sym value="-"/>
            </f>
        </fs>
    </lemmeref>
</morph>

<morph lex="voli" >
    <lemmeref cat="V" name="voleti" >
        <fs>
            <f name="mod">

```



```

        <sym value="I"/>
    </f>
    <f name="form">
        <sym value="P"/>
    </f>
    <f name="pers">
        <sym value="z"/>
    </f>
    <f name="num">
        <sym value="s"/>
    </f>
    <f name="neg">
        <sym value="-"/>
    </f>
</fs>
</lemmeref>
</morph>

```

```

<morph lex="vole" >
    <lemmeref cat="V" name="voleti" >
        <fs>
            <f name="mod">
                <sym value="I"/>
            </f>
            <f name="form">
                <sym value="P"/>
            </f>
            <f name="pers">
                <sym value="z"/>
            </f>
            <f name="num">
                <sym value="p"/>
            </f>
            <f name="neg">
                <sym value="-"/>
            </f>
        </fs>
    </lemmeref>
</morph>

```

```

<morph lex="volxen" >

```

```
<lemmeref cat="V" name="voleti" >
  <fs>
    <f name="mod">
      <sym value="T"/>
    </f>
    <f name="num">
      <sym value="s"/>
    </f>
    <f name="gen">
      <sym value="m"/>
    </f>
    <f name="neg">
      <sym value="-"/>
    </f>
  </fs>
</lemmeref>
</morph>
```

```
<morph lex="volxena" >
  <lemmeref cat="V" name="voleti" >
    <fs>
      <f name="mod">
        <sym value="T"/>
      </f>
      <f name="num">
        <sym value="s"/>
      </f>
      <f name="gen">
        <sym value="f"/>
      </f>
      <f name="neg">
        <sym value="-"/>
      </f>
    </fs>
  </lemmeref>
</morph>
```

```
<morph lex="cyeslxa" >
  <lemmeref cat="V" name="cyeslxa" >
    <fs>
      <f name="mod">
```

```

        <sym value="I"/>
    </f>
    <f name="form">
        <sym value="P"/>
    </f>
    <f name="num">
        <sym value="s"/>
    </f>
    <f name="pers">
        <sym value="z"/>
    </f>
    <f name="neg">
        <sym value="-"/>
    </f>
</fs>
</lemmaref>
</morph>

<morph lex="sxaliti" >
    <lemmaref cat="V" name="sxaliti" >
        <fs>
            <f name="mod">
                <sym value="W"/>
            </f>
            <f name="neg">
                <sym value="-"/>
            </f>
        </fs>
    </lemmaref>
</morph>

<morph lex="sxali" >
    <lemmaref cat="V" name="sxaliti" >
        <fs>
            <f name="mod">
                <sym value="I"/>
            </f>
            <f name="form">
                <sym value="P"/>
            </f>
            <f name="num">

```

```

        <sym value="s"/>
    </f>
    <f name="pers">
        <sym value="z"/>
    </f>
    <f name="neg">
        <sym value="-"/>
    </f>
</fs>
</lemmaref>
</morph>

<morph lex="video" >
    <lemmaref cat="V" name="videti" >
        <fs>
            <f name="mod">
                <sym value="G"/>
            </f>
            <f name="form">
                <sym value="C"/>
            </f>
            <f name="num">
                <sym value="s"/>
            </f>
            <f name="gen">
                <sym value="m"/>
            </f>
            <f name="neg">
                <sym value="-"/>
            </f>
        </fs>
    </lemmaref>
</morph>

<morph lex="vidim" >
    <lemmaref cat="V" name="videti" >
        <fs>
            <f name="mod">
                <sym value="I"/>
            </f>
            <f name="form">

```

```

        <sym value="P"/>
    </f>
    <f name="pers">
        <sym value="x"/>
    </f>
    <f name="num">
        <sym value="s"/>
    </f>
    <f name="neg">
        <sym value="-"/>
    </f>
</fs>
</lemmaref>
</morph>

<morph lex="smrkava" >
    <lemmaref cat="V" name="smrkavati" >
        <fs>
            <f name="mod">
                <sym value="I"/>
            </f>
            <f name="form">
                <sym value="P"/>
            </f>
        </fs>
        <fs>
            <f name="pers">
                <sym value="z"/>
            </f>
            <f name="num">
                <sym value="s"/>
            </f>
            <f name="neg">
                <sym value="-"/>
            </f>
        </fs>
    </lemmaref>
</morph>

<morph lex="svicxe" >
    <lemmaref cat="V" name="svitati" >

```

```

    <fs>
    <f name="mod">
      <sym value="I"/>
    </f>
    <f name="form">
      <sym value="P"/>
    </f>
    <f name="pers">
      <sym value="z"/>
    </f>
    <f name="num">
      <sym value="s"/>
    </f>
    <f name="neg">
      <sym value="-"/>
    </f>
  </fs>
</lemmaref>
</morph>

<morph lex="boli" >
  <lemmaref cat="V" name="boleti" >
    <fs>
    <f name="mod">
      <sym value="I"/>
    </f>
    <f name="form">
      <sym value="P"/>
    </f>
    <f name="pers">
      <sym value="z"/>
    </f>
    <f name="num">
      <sym value="s"/>
    </f>
    <f name="neg">
      <sym value="-"/>
    </f>
  </fs>
</lemmaref>
</morph>

```

```

<morph lex="igra" >
  <lemmeref cat="V" name="igrati" >
    <fs>
      <f name="mod">
        <sym value="I"/>
      </f>
      <f name="form">
        <sym value="P"/>
      </f>
      <f name="pers">
        <sym value="z"/>
      </f>
      <f name="num">
        <sym value="s"/>
      </f>
      <f name="neg">
        <sym value="-"/>
      </f>
    </fs>
  </lemmeref>
</morph>

```

```

<morph lex="spava" >
  <lemmeref cat="V" name="spavati" >
    <fs>
      <f name="pers">
        <sym value="z"/>
      </f>
      <f name="num">
        <sym value="s"/>
      </f>
      <f name="mod">
        <sym value="I"/>
      </f>
      <f name="form">
        <sym value="P"/>
      </f>
      <f name="neg">
        <sym value="-"/>
      </f>
    </fs>
  </lemmeref>
</morph>

```

```

    </fs>
  </lemmeref>
</morph>

<morph lex="spavaju" >
  <lemmeref cat="V" name="spavati" >
    <fs>
      <f name="pers">
        <sym value="z"/>
      </f>
      <f name="num">
        <sym value="p"/>
      </f>
      <f name="mod">
        <sym value="I"/>
      </f>
      <f name="form">
        <sym value="P"/>
      </f>
      <f name="neg">
        <sym value="-"/>
      </f>
    </fs>
  </lemmeref>
</morph>

```

```

<morph lex="stidi" >
  <lemmeref cat="V" name="stideti" >
    <fs>
      <f name="pers">
        <sym value="z"/>
      </f>
      <f name="num">
        <sym value="s"/>
      </f>
      <f name="mod">
        <sym value="I"/>
      </f>
      <f name="form">
        <sym value="P"/>
      </f>
    </fs>
  </lemmeref>
</morph>

```



```

        <f name="neg">
            <sym value="-"/>
        </f>
    </fs>
</lemmaref>
</morph>

<morph lex="pomazxe" >
    <lemmaref cat="V" name="pomagati" >
        <fs>
            <f name="pers">
                <sym value="z"/>
            </f>
            <f name="num">
                <sym value="s"/>
            </f>
            <f name="mod">
                <sym value="I"/>
            </f>
            <f name="form">
                <sym value="P"/>
            </f>
            <f name="neg">
                <sym value="-"/>
            </f>
        </fs>
    </lemmaref>
</morph>

<morph lex="pomagano" >
    <lemmaref cat="V" name="pomagati" >
        <fs>
            <f name="mod">
                <sym value="T"/>
            </f>
            <f name="num">
                <sym value="s"/>
            </f>
            <f name="gen">
                <sym value="N"/>
            </f>
        </fs>
    </lemmaref>
</morph>

```

```

        <f name="neg">
            <sym value="-"/>
        </f>
    </fs>
</lemmaref>
</morph>

<morph lex="zxivi" >
    <lemmaref cat="V" name="zxiveti" >
        <fs>
            <f name="pers">
                <sym value="z"/>
            </f>
            <f name="num">
                <sym value="s"/>
            </f>
            <f name="mod">
                <sym value="I"/>
            </f>
            <f name="form">
                <sym value="P"/>
            </f>
            <f name="neg">
                <sym value="-"/>
            </f>
        </fs>
    </lemmaref>
</morph>

<morph lex="smeta" >
    <lemmaref cat="V" name="smetati" >
        <fs>
            <f name="pers">
                <sym value="z"/>
            </f>
            <f name="num">
                <sym value="s"/>
            </f>
            <f name="mod">
                <sym value="I"/>
            </f>
        </fs>
    </lemmaref>
</morph>

```

```

        <f name="form">
        <sym value="P"/>
    </f>
    <f name="neg">
        <sym value="-"/>
    </f>
</fs>
</lemmaref>
</morph>

<morph lex="stavlxa" >
    <lemmaref cat="V" name="stavlxati" >
        <fs>
            <f name="pers">
                <sym value="z"/>
            </f>
            <f name="num">
                <sym value="s"/>
            </f>
            <f name="mod">
                <sym value="I"/>
            </f>
            <f name="form">
                <sym value="P"/>
            </f>
            <f name="neg">
                <sym value="-"/>
            </f>
        </fs>
    </lemmaref>
</morph>

<morph lex="nema" >
    <lemmaref cat="V" name="nemati" >
        <fs>
            <f name="mod">
                <sym value="I"/>
            </f>
            <f name="form">
                <sym value="P"/>
            </f>
        </fs>
    </lemmaref>
</morph>

```

```
<f name="pers">
  <sym value="z"/>
</f>
<f name="num">
  <sym value="s"/>
</f>
<f name="neg">
  <sym value="+"/>
</f>
</fs>
</lemmaref>
</morph>
```

```
<morph lex="daje" >
  <lemmaref cat="V" name="davati" >
    <fs>
      <f name="pers">
        <sym value="z"/>
      </f>
      <f name="num">
        <sym value="s"/>
      </f>
      <f name="mod">
        <sym value="I"/>
      </f>
      <f name="form">
        <sym value="P"/>
      </f>
      <f name="neg">
        <sym value="-"/>
      </f>
    </fs>
  </lemmaref>
</morph>
```

```
<!-- IMENICE -->
```

```
<morph lex="Sonxa" >
  <lemmaref cat="N" name="Sonxa" >
    <fs>
      <f name="case">
```

```

        <sym value="1"/>
    </f>
    <f name="wh">
        <sym value="-"/>
    </f>
    <f name="gen">
        <sym value="f"/>
    </f>
    <f name="pers">
        <sym value="z"/>
    </f>
    <f name="num">
        <sym value="s"/>
    </f>
</fs>
</lemmaref>
</morph>

```

```

<morph lex="Sonxu" >
    <lemmaref cat="N" name="Sonxa" >
        <fs>
            <f name="case">
                <sym value="4"/>
            </f>
            <f name="wh">
                <sym value="-"/>
            </f>
            <f name="gen">
                <sym value="f"/>
            </f>
            <f name="pers">
                <sym value="z"/>
            </f>
            <f name="num">
                <sym value="s"/>
            </f>
        </fs>
    </lemmaref>
</morph>

```

```

<morph lex="Dyon" >

```

```

<lemmeref cat="N" name="Dyon" >
  <fs>
    <f name="case">
      <sym value="1"/>
    </f>
    <f name="pers">
      <sym value="z"/>
    </f>
    <f name="gen">
      <sym value="m"/>
    </f>
    <f name="num">
      <sym value="s"/>
    </f>
    <f name="wh">
      <sym value="-"/>
    </f>
  </fs>
</lemmeref>
</morph>

```

```

<morph lex="Dyona" >
  <lemmeref cat="N" name="Dyon" >
    <fs>
      <f name="case">
        <sym value="2"/>
      </f>
      <f name="wh">
        <sym value="-"/>
      </f>
      <f name="gen">
        <sym value="m"/>
      </f>
      <f name="pers">
        <sym value="z"/>
      </f>
      <f name="num">
        <sym value="s"/>
      </f>
    </fs>
  </lemmeref>

```

</morph>

```
<morph lex="Dyona" >
  <lemmaref cat="N" name="Dyon" >
    <fs>
      <f name="case">
        <sym value="4"/>
      </f>
      <f name="wh">
        <sym value="-"/>
      </f>
      <f name="gen">
        <sym value="m"/>
      </f>
      <f name="pers">
        <sym value="z"/>
      </f>
      <f name="num">
        <sym value="s"/>
      </f>
    </fs>
  </lemmaref>
</morph>
```

```
<morph lex="park" >
  <lemmaref cat="N" name="park" >
    <fs>
      <f name="case">
        <sym value="1"/>
      </f>
      <f name="wh">
        <sym value="-"/>
      </f>
      <f name="gen">
        <sym value="m"/>
      </f>
      <f name="pers">
        <sym value="z"/>
      </f>
      <f name="num">
        <sym value="s"/>
      </f>
    </fs>
  </lemmaref>
</morph>
```

```

        </f>
    </fs>
</lemmeref>
</morph>

<morph lex="park" >
    <lemmeref cat="N" name="park" >
        <fs>
            <f name="case">
                <sym value="4"/>
            </f>
            <f name="wh">
                <sym value="-"/>
            </f>
            <f name="gen">
                <sym value="m"/>
            </f>
            <f name="pers">
                <sym value="z"/>
            </f>
            <f name="num">
                <sym value="s"/>
            </f>
        </fs>
    </lemmeref>
</morph>

<morph lex="parku" >
    <lemmeref cat="N" name="park" >
        <fs>
            <f name="case">
                <sym value="7"/>
            </f>
            <f name="wh">
                <sym value="-"/>
            </f>
            <f name="gen">
                <sym value="m"/>
            </f>
            <f name="pers">
                <sym value="z"/>
            </f>
        </fs>
    </lemmeref>
</morph>

```



```
    </f>
  <f name="num">
    <sym value="s"/>
  </f>
</fs>
</lemmaref>
</morph>
```

```
<morph lex="dinara" >
  <lemmaref cat="N" name="dinar" >
    <fs>
      <f name="case">
        <sym value="2"/>
      </f>
      <f name="wh">
        <sym value="-"/>
      </f>
      <f name="gen">
        <sym value="m"/>
      </f>
      <f name="pers">
        <sym value="z"/>
      </f>
      <f name="num">
        <sym value="s"/>
      </f>
    </fs>
  </lemmaref>
</morph>
```

```
<morph lex="lutku" >
  <lemmaref cat="N" name="lutka" >
    <fs>
      <f name="case">
        <sym value="4"/>
      </f>
      <f name="wh">
        <sym value="-"/>
      </f>
      <f name="gen">
```

```

        <sym value="f"/>
    </f>
<f name="pers">
    <sym value="z"/>
</f>
<f name="num">
    <sym value="s"/>
</f>
</fs>
</lemmeref>
</morph>

<morph lex="Marko" >
    <lemmeref cat="N" name="Marko" >
        <fs>
            <f name="case">
                <sym value="1"/>
            </f>
            <f name="wh">
                <sym value="-"/>
            </f>
            <f name="gen">
                <sym value="m"/>
            </f>
            <f name="pers">
                <sym value="z"/>
            </f>
            <f name="num">
                <sym value="s"/>
            </f>
        </fs>
    </lemmeref>
</morph>

<morph lex="Marka" >
    <lemmeref cat="N" name="Marko" >
        <fs>
            <f name="case">
                <sym value="4"/>
            </f>
            <f name="wh">

```

```

        <sym value="-"/>
    </f>
<f name="gen">
    <sym value="m"/>
</f>
<f name="pers">
    <sym value="z"/>
</f>
<f name="num">
    <sym value="s"/>
</f>
</fs>
</lemmaref>
</morph>

```

```

<morph lex="Marku" >
  <lemmaref cat="N" name="Marko" >
    <fs>
      <f name="case">
        <sym value="3"/>
      </f>
      <f name="wh">
        <sym value="-"/>
      </f>
      <f name="gen">
        <sym value="m"/>
      </f>
      <f name="pers">
        <sym value="z"/>
      </f>
      <f name="num">
        <sym value="s"/>
      </f>
    </fs>
  </lemmaref>
</morph>

```

```

<morph lex="krevet" >
  <lemmaref cat="N" name="krevet" >
    <fs>
      <f name="case">

```

```

        <sym value="1"/>
    </f>
    <f name="wh">
        <sym value="-"/>
    </f>
    <f name="gen">
        <sym value="m"/>
    </f>
    <f name="pers">
        <sym value="z"/>
    </f>
    <f name="num">
        <sym value="s"/>
    </f>
</fs>
</lemmaref>
</morph>

```

```

<morph lex="krevet" >
    <lemmaref cat="N" name="krevet" >
        <fs>
            <f name="case">
                <sym value="4"/>
            </f>
            <f name="wh">
                <sym value="-"/>
            </f>
            <f name="gen">
                <sym value="m"/>
            </f>
            <f name="pers">
                <sym value="z"/>
            </f>
            <f name="num">
                <sym value="s"/>
            </f>
        </fs>
    </lemmaref>
</morph>

```

```

<morph lex="krevetu" >

```

```
<lemmeref cat="N" name="krevet" >
  <fs>
    <f name="case">
      <sym value="7"/>
    </f>
    <f name="wh">
      <sym value="-"/>
    </f>
    <f name="gen">
      <sym value="m"/>
    </f>
    <f name="pers">
      <sym value="z"/>
    </f>
    <f name="num">
      <sym value="s"/>
    </f>
  </fs>
</lemmeref>
</morph>
```

```
<morph lex="glava" >
  <lemmeref cat="N" name="glava" >
    <fs>
      <f name="case">
        <sym value="1"/>
      </f>
      <f name="wh">
        <sym value="-"/>
      </f>
      <f name="gen">
        <sym value="f"/>
      </f>
      <f name="pers">
        <sym value="z"/>
      </f>
      <f name="num">
        <sym value="s"/>
      </f>
    </fs>
  </lemmeref>
```

</morph>

```
<morph lex="glavu" >
  <lemmaref cat="N" name="glava" >
    <fs>
      <f name="case">
        <sym value="4"/>
      </f>
      <f name="wh">
        <sym value="-"/>
      </f>
      <f name="gen">
        <sym value="f"/>
      </f>
      <f name="pers">
        <sym value="z"/>
      </f>
      <f name="num">
        <sym value="s"/>
      </f>
    </fs>
  </lemmaref>
</morph>
```

<!-- ZAMENICE -->

```
<morph lex="on" >
  <lemmaref cat="PRO" name="on" >
    <fs>
      <f name="case">
        <sym value="1"/>
      </f>
      <f name="wh">
        <sym value="-"/>
      </f>
      <f name="gen">
        <sym value="m"/>
      </f>
      <f name="pers">
        <sym value="z"/>
      </f>
    </fs>
  </lemmaref>
</morph>
```

```

    <f name="num">
      <sym value="s"/>
    </f>
  </fs>
</lemmaref>
</morph>

<morph lex="nxega" >
  <lemmaref cat="PRO" name="on" >
    <fs>
      <f name="case">
        <sym value="4"/>
      </f>
      <f name="wh">
        <sym value="-"/>
      </f>
      <f name="gen">
        <sym value="m"/>
      </f>
      <f name="pers">
        <sym value="z"/>
      </f>
      <f name="num">
        <sym value="s"/>
      </f>
    </fs>
  </lemmaref>
</morph>

<morph lex="mu" >
  <lemmaref cat="Cl" name="on" >
    <fs>
      <f name="case">
        <sym value="3"/>
      </f>
      <f name="wh">
        <sym value="-"/>
      </f>
      <f name="gen">
        <sym value="m"/>
      </f>

```

```

    <f name="pers">
      <sym value="z"/>
    </f>
    <f name="num">
      <sym value="s"/>
    </f>
  </fs>
</lemmaref>
</morph>

<morph lex="ona" >
  <lemmaref cat="PRO" name="ona" >
    <fs>
      <f name="case">
        <sym value="1"/>
      </f>
      <f name="wh">
        <sym value="-"/>
      </f>
      <f name="gen">
        <sym value="f"/>
      </f>
      <f name="pers">
        <sym value="z"/>
      </f>
      <f name="num">
        <sym value="s"/>
      </f>
    </fs>
  </lemmaref>
</morph>

<morph lex="nxu" >
  <lemmaref cat="PRO" name="ona" >
    <fs>
      <f name="case">
        <sym value="4"/>
      </f>
      <f name="wh">
        <sym value="-"/>
      </f>

```



```

    <f name="gen">
      <sym value="f"/>
    </f>
    <f name="pers">
      <sym value="z"/>
    </f>
    <f name="num">
      <sym value="s"/>
    </f>
  </fs>
</lemmaref>
</morph>

<morph lex="nxoj" >
  <lemmaref cat="PRO" name="ona" >
    <fs>
      <f name="case">
        <sym value="3"/>
      </f>
      <f name="wh">
        <sym value="-"/>
      </f>
      <f name="pers">
        <sym value="z"/>
      </f>
      <f name="gen">
        <sym value="f"/>
      </f>
      <f name="num">
        <sym value="s"/>
      </f>
    </fs>
  </lemmaref>
</morph>

<morph lex="sebe" >
  <lemmaref cat="PRO" name="sebe" >
    <fs>
      <f name="case">
        <sym value="4"/>
      </f>
    </fs>
  </lemmaref>
</morph>

```

```

    <f name="wh">
      <sym value="-"/>
    </f>
    <f name="pers">
      <sym value="z"/>
    </f>
    <f name="num">
      <sym value="s"/>
    </f>
  </fs>
</lemmaref>
</morph>

<morph lex="koji" >
  <lemmaref cat="PRO" name="koji" >
    <fs>
      <f name="case">
        <sym value="1"/>
      </f>
      <f name="wh">
        <sym value="-"/>
      </f>
      <f name="gen">
        <sym value="m"/>
      </f>
      <f name="num">
        <sym value="s"/>
      </f>
    </fs>
  </lemmaref>
</morph>

<morph lex="ko" >
  <lemmaref cat="PRO" name="ko" >
    <fs>
      <f name="case">
        <sym value="1"/>
      </f>
      <f name="wh">
        <sym value="+"/>
      </f>
    </fs>
  </lemmaref>
</morph>

```

```

    <f name="pers">
      <sym value="z"/>
    </f>
    <f name="num">
      <sym value="s"/>
    </f>
  </fs>
</lemmaref>
</morph>

<morph lex="sxta" >
  <lemmaref cat="PRO" name="sxta" >
    <fs>
      <f name="case">
        <sym value="1"/>
      </f>
      <f name="wh">
        <sym value="+"/>
      </f>
      <f name="pers">
        <sym value="z"/>
      </f>
      <f name="num">
        <sym value="s"/>
      </f>
    </fs>
  </lemmaref>
</morph>

<morph lex="sxta" >
  <lemmaref cat="PRO" name="sxta" >
    <fs>
      <f name="case">
        <sym value="4"/>
      </f>
      <f name="wh">
        <sym value="+"/>
      </f>
      <f name="pers">
        <sym value="z"/>
      </f>
    </fs>
  </lemmaref>
</morph>

```

```

    <f name="num">
      <sym value="s"/>
    </f>
  </fs>
</lemmaref>
</morph>

<morph lex="cyemu" >
  <lemmaref cat="PRO" name="sxta" >
    <fs>
      <f name="case">
        <sym value="7"/>
      </f>
      <f name="wh">
        <sym value="+"/>
      </f>
      <f name="pers">
        <sym value="z"/>
      </f>
      <f name="num">
        <sym value="s"/>
      </f>
    </fs>
  </lemmaref>
</morph>

<morph lex="kome" >
  <lemmaref cat="PRO" name="ko" >
    <fs>
      <f name="case">
        <sym value="3"/>
      </f>
      <f name="wh">
        <sym value="+"/>
      </f>
      <f name="pers">
        <sym value="z"/>
      </f>
      <f name="num">
        <sym value="s"/>
      </f>
    </fs>
  </lemmaref>
</morph>

```

```

    </fs>
  </lemmaref>
</morph>

<morph lex="koga" >
  <lemmaref cat="PRO" name="ko" >
    <fs>
      <f name="case">
        <sym value="4"/>
      </f>
      <f name="wh">
        <sym value="+"/>
      </f>
      <f name="pers">
        <sym value="z"/>
      </f>
      <f name="num">
        <sym value="s"/>
      </f>
    </fs>
  </lemmaref>
</morph>

<morph lex="ovo" >
  <lemmaref cat="PRO" name="ovo" >
    <fs>
      <f name="case">
        <sym value="1"/>
      </f>
      <f name="wh">
        <sym value="-"/>
      </f>
      <f name="pers">
        <sym value="z"/>
      </f>
      <f name="num">
        <sym value="s"/>
      </f>
      <f name="gen">
        <sym value="n"/>
      </f>
    </fs>
  </lemmaref>
</morph>

```

```
</fs>
</lemmaref>
</morph>
```

```
<!-- PRIDEVI-->
```

```
<morph lex="pametana" >
  <lemmaref cat="A" name="pametana" >
    <fs>
      <f name="case">
        <sym value="1"/>
      </f>
      <f name="wh">
        <sym value="-"/>
      </f>
      <f name="pers">
        <sym value="z"/>
      </f>
      <f name="num">
        <sym value="s"/>
      </f>
      <f name="gen">
        <sym value="m"/>
      </f>
    </fs>
  </lemmaref>
</morph>
```

```
<morph lex="pametna" >
  <lemmaref cat="A" name="pametana" >
    <fs>
      <f name="case">
        <sym value="1"/>
      </f>
      <f name="wh">
        <sym value="-"/>
      </f>
      <f name="pers">
        <sym value="z"/>
      </f>
      <f name="num">
```

```

        <sym value="s"/>
    </f>
    <f name="gen">
        <sym value="f"/>
    </f>
    </fs>
</lemmeref>
</morph>

<morph lex="kakav" >
    <lemmeref cat="A" name="kakav" >
        <fs>
            <f name="case">
                <sym value="1"/>
            </f>
            <f name="wh">
                <sym value="+"/>
            </f>
            <f name="pers">
                <sym value="z"/>
            </f>
            <f name="num">
                <sym value="s"/>
            </f>
            <f name="gen">
                <sym value="m"/>
            </f>
        </fs>
    </lemmeref>
</morph>

<morph lex="kakva" >
    <lemmeref cat="A" name="kakav" >
        <fs>
            <f name="case">
                <sym value="1"/>
            </f>
            <f name="wh">
                <sym value="+"/>
            </f>
            <f name="pers">

```

```
        <sym value="z"/>
    </f>
<f name="num">
    <sym value="s"/>
</f>
<f name="gen">
    <sym value="f"/>
</f>
</fs>
</lemmaref>
</morph>
```

<!-- PRILOZI-->

```
<morph lex="danas" >
  <lemmaref cat="ADV" name="danas" >
    <fs>
      <f name="wh">
        <sym value="-"/>
      </f>
    </fs>
  </lemmaref>
</morph>
```

```
<morph lex="prlxavo" >
  <lemmaref cat="ADV" name="prlxavo" >
    <fs>
      <f name="wh">
        <sym value="-"/>
      </f>
    </fs>
  </lemmaref>
</morph>
```

```
<morph lex="ovde" >
  <lemmaref cat="ADV" name="ovde" >
    <fs>
      <f name="wh">
        <sym value="-"/>
      </f>
    </fs>
```



```

        </lemmeref>
    </morph>

<morph lex="gde" >
    <lemmeref cat="ADV" name="gde" >
        <fs>
            <f name="wh">
                <sym value="+"/>
            </f>
        </fs>
    </lemmeref>
</morph>

<morph lex="dobro" >
    <lemmeref cat="ADV" name="dobro" >
        <fs>
            <f name="wh">
                <sym value="-"/>
            </f>
        </fs>
    </lemmeref>
</morph>

<!-- ENKLITIKE -->

<morph lex="se" >
    <lemmeref cat="Cl" name="se" >
        <fs>
            <f name="ref">
                <sym value="+"/>
            </f>
            <f name="rank">
                <sym value="5"/>
            </f>
        </fs>
    </lemmeref>
</morph>

<!-- PREDLOZI -->

```

```
<morph lex="u" >
  <lemmeref cat="P" name="u" >
    <fs>
      <f name="case">
        <sym value="4"/>
      </f>
    </fs>
  </lemmeref>
</morph>
```

```
<morph lex="u" >
  <lemmeref cat="P" name="u" >
    <fs>
      <f name="case">
        <sym value="7"/>
      </f>
    </fs>
  </lemmeref>
</morph>
```

```
<!-- KOMPLEMENTIZATORI -->
```

```
<morph lex="da" >
  <lemmeref cat="C" name="da" >
    <fs>
      <f name="wh">
        <sym value="-"/>
      </f>
    </fs>
  </lemmeref>
</morph>
```

```
<!-- NEG -->
```

```
<morph lex="ne" >
  <lemmeref cat="NEG" name="ne" >
    <fs>
      <f name="neg">
        <sym value="+"/>
      </f>
    </fs>
  </lemmeref>
</morph>
```

```
</lemmaref>
</morph>
```

```
<!-- BROJEVI -->
```

```
<morph lex="dva" >
  <lemmaref cat="NUM" name="dva" >
    <fs>
      <f name="case">
        <sym value="2"/>
      </f>
      <f name="wh">
        <sym value="-"/>
      </f>
      <f name="pers">
        <sym value="z"/>
      </f>
      <f name="num">
        <sym value="s"/>
      </f>
      <f name="gen">
        <sym value="m"/>
      </f>
    </fs>
  </lemmaref>
</morph>
```

```
</morphs>
</mcgrammar>
```

8.4 Pregled tipova zavisnih rečenica i njihovih konstituentskih vrednosti

Tabela 15 Pregled tipova zavisnih rečenica i njihovih konstituentskih vrednosti

Redni broj	Tip rečenice	Subordinatori	Funkcija	Primeri
1	Izrična	<i>da, kako, da li, li, ko, šta, koji, gde, kad, kako</i>	<ul style="list-style-type: none"> • subjekat • pravi objekat • nepravi objekat • dopunski atribut (imenička sintagma) • dopunski zavisni član 	<p><i>Jasno je da se priredba neće održati.</i> (subjekat)</p> <p><i>Marko je pokušao da nađe karte za utakmicu.</i> (pravi objekat)</p> <p><i>Ivan je pristao (na to) da on kupi karte za bioskop.</i> (nepravi objekat)</p> <p><i>zahtev da se sve knjige vrate u biblioteku</i> (dopunski atribut)</p> <p><i>svestan da je pogrešio</i> (dopunski zavisni član)</p>
2	Odnosna	<i>koji, čiji, kakav, koliko, ko, šta, što, gde, kuda, kad, kako, koliko.</i>	<ul style="list-style-type: none"> • subjekat • logički subjekat • pravi objekat • nepravi objekat • atribut • apozitiv 	<p><i>Ko rano rani, dve sreće grabi.</i> (subjekat)</p> <p><i>Što možeš da uradiš danas, (to) ne ostavljaj za sutra.</i> (pravi objekat)</p> <p><i>Pričao je o onome što ga je mučilo.</i> (nepravi objekat)</p> <p><i>Ko rano rani, njemu se ceo dan spava.</i> (logički subjekat)</p> <p><i>Podigao je (onu) ruku u kojoj je držao telegram.</i> (atribut)</p> <p><i>Podigao je desnu ruku, u kojoj je držao telegram.</i> (apozitiv)</p>
3	Mesna	<i>gde, kuda, kamo, odakle</i>	<ul style="list-style-type: none"> • odredba za mesto 	<p><i>Gde on udari, tu trava ne raste.</i></p> <p><i>Prešao je tamo odakle je imao bolji pogled na igralište.</i></p>
4	Vremenska	<i>kad, dok, pošto, čim, nakon što, pre nego što</i>	<ul style="list-style-type: none"> • odredba vremena • odredba mere 	<p><i>Kad sam se vraćao kući, kupio sam novine.</i></p> <p><i>Čim sam završio posao, otišao sam u šetnju.</i></p>

Redn i broj	Tip rečenice	Subordinatori	Funkcija	Primeri
5	Uzročna	<i>jer, zato što, stoga što, pošto, kako, budući da</i>	<ul style="list-style-type: none"> odredba uzroka 	<i>Tanja nije došla u školu jer se razbolela od gripa. Svi smo prihvatili Jasnin predlog budući da je to bilo jedino rešenje.</i>
5	Uslovna	<i>ako, ukoliko, li, kad, da</i>	<ul style="list-style-type: none"> odredba uslova 	<i>Ako dobijem/budem dobio premiju, kupiću auto. Kad bih dobio premiju/Da dobijem premiju, kupio bih auto. Da sam dobio premiju, kupio bih auto.</i>
6	Dopusna	<i>iako, mada, premda</i>	<ul style="list-style-type: none"> dopusna odredba 	<i>Iako je požurio, ipak nije stigao na vreme. Mada je bio bolestan, otišao je u školu.</i>
7	Namerna	<i>da, kako</i>	<ul style="list-style-type: none"> odredba cilja dopuna za cilj 	<i>Uzeo je taksi da stigne/da bi stigao/kako bi stigao na vreme. (odredba cilja) Marko je otišao da spava. (dopuna za cilj)</i>
8	Poredbena	<i>kao što, kao da, kako, koliko, no/nego što, no da/nego</i>	<ul style="list-style-type: none"> poredbena odredba poredbena dopuna 	<i>Gledao nas je kao da nas prvi put vidi. (poredbena odredba) Taj film je bolji nego/no što misliš. (poredbena dopuna)</i>
9	Posledična ¹⁷⁵	<i>da</i> ¹⁷⁴	<ul style="list-style-type: none"> posledična dopuna 	<i>Zoran tako vozi auto da se svi dive. Jasna toliko radi da se svi dive.</i>

¹⁷⁴ U okviru glavne rečenice uvek se javljaju prilozi i zamenice koji ukazuju na prisustvo ove dopune. To su *tako, toliko, takav* i *toliki*

¹⁷⁵ Ove zavisne rečenice se uvek javljaju iza glavne rečenice.

9 BIOGRAFIJA AUTORA

Bojana Đorđević je rođena 6. jula 1983. godine u Kruševcu, gde je završila osnovnu školu i gimnaziju. Diplomirala je u oktobru 2006. godine na Filološkom fakultetu u Beogradu, na smeru Opšta lingvistika, sa prosečnom ocenom 9,61 (devet i 61/100).

Nakon studija počinje da radi kao profesor engleskog jezika i srpskog jezika za strance u Centru za srpski jezik i kulturu Azbukum, gde i sada radi kao profesor i direktor nastave. Paralelno sa time, kao glavni stručni saradnik radi na izradi *Rečnika sinonima* Pavla Ćosića i saradnika, koji izlazi u štampu 2008. godine. Bavi se lekturom i prevođenjem (sa engleskog jezika i na engleski jezik), za različite poslodavce i biltene. Uporedo sa komercijalnim radom, počevši od završetka studija radila je na projektima iz računarske lingvistike, kakav je dopunjavanje srpskog WordNeta i izrada paralelnog francusko-srpsko-engleskog korpusa ParCoLab. Istovremeno je radila na obučavanju studenata i istraživača za izradu paralelnih korpusa.

Doktorske studije upisuje 2009. godine. U periodu od 2009. do 2011. godine radi kao saradnik u nastavi na predmetima *Metode obrade prirodnih jezika* i *Digitalni dokument*.

Član je Grupe za jezičke tehnologije (JERTEH) Univerziteta u Beogradu. Učestvovala je u međunarodnom projektu obrade višečlanih reči PARSEME. Predstavljala je svoje radove na konferencijama i radionicama u Srbiji i inostranstvu.

Njena uža oblast interesovanja je računarska lingvistika, posebno automatska sintaksička analiza teksta (parsiranje).

Govori engleski jezik i služi se francuskim jezikom.

10 IZJAVE O DISERTACIJI

Прилог 1.

Изјава о ауторству

Потписана Бојана П. Ђорђевић

број индекса 08089 Д

Изјављујем

да је докторска дисертација под насловом

**ИЗРАДА ОСНОВА ФОРМАЛНЕ ГРАМАТИКЕ СРПСКОГ ЈЕЗИКА УПОТРЕБОМ
МЕТАГРАМАТИКЕ**

- резултат сопственог истраживачког рада,
- да предложена дисертација у целини ни у деловима није била предложена за добијање било које дипломе према студијским програмима других високошколских установа,
- да су резултати коректно наведени и
- да нисам кршио/ла ауторска права и користио интелектуалну својину других лица.

Потпис докторанда

У Београду, 30.03.2017.

Ђорђевић

Прилог 2.

Изјава о истоветности штампане и електронске верзије докторског рада

Име и презиме аутора _____ Бојана П. Ђорђевић _____

Број индекса _____ 08089 Д _____

Студијски програм _____ Језик – Општа лингвистика _____

Наслов рада _____ Израда основа формалне граматике српског језика
употребом метаграматике _____

Ментор _____ проф. др Цветана Крстев _____

Потписана _____ Бојана П. Ђорђевић _____

Изјављујем да је штампана верзија мог докторског рада истоветна електронској верзији коју сам предао/ла за објављивање на порталу **Дигиталног репозиторијума Универзитета у Београду**.

Дозвољавам да се објаве моји лични подаци везани за добијање академског звања доктора наука, као што су име и презиме, година и место рођења и датум одбране рада.

Ови лични подаци могу се објавити на мрежним страницама дигиталне библиотеке, у електронском каталогу и у публикацијама Универзитета у Београду.

Потпис докторанда

У Београду, 30.03.2017.

Ђорђевић

Прилог 3.

Изјава о коришћењу

Овлашћујем Универзитетску библиотеку „Светозар Марковић“ да у Дигитални репозиторијум Универзитета у Београду унесе моју докторску дисертацију под насловом:

ИЗРАДА ОСНОВА ФОРМАЛНЕ ГРАМАТИКЕ СРПСКОГ ЈЕЗИКА УПОТРЕБОМ МЕТАГРАМАТИКЕ

која је моје ауторско дело.

Дисертацију са свим прилозима предао/ла сам у електронском формату погодном за трајно архивирање.

Моју докторску дисертацију похрањену у Дигитални репозиторијум Универзитета у Београду могу да користе сви који поштују одредбе садржане у одабраном типу лиценце Креативне заједнице (Creative Commons) за коју сам се одлучио/ла.

1. Ауторство
2. Ауторство - некомерцијално
3. Ауторство – некомерцијално – без прераде
4. Ауторство – некомерцијално – делити под истим условима
5. Ауторство – без прераде
6. Ауторство – делити под истим условима

(Молимо да заокружите само једну од шест понуђених лиценци, кратак опис лиценци дат је на полеђини листа).

Потпис докторанда

У Београду, 30.03.2017.

Borotaw

1. Ауторство - Дозвољавање умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце, чак и у комерцијалне сврхе. Ово је најслободнија од свих лиценци.

2. Ауторство – некомерцијално. Дозвољавање умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце. Ова лиценца не дозвољава комерцијалну употребу дела.

3. Ауторство - некомерцијално – без прераде. Дозвољавање умножавање, дистрибуцију и јавно саопштавање дела, без промена, преобликовања или употребе дела у свом делу, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце. Ова лиценца не дозвољава комерцијалну употребу дела. У односу на све остале лиценце, овом лиценцом се ограничава највећи обим права коришћења дела.

4. Ауторство - некомерцијално – делити под истим условима. Дозвољавање умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце и ако се прерада дистрибуира под истом или сличном лиценцом. Ова лиценца не дозвољава комерцијалну употребу дела и прерада.

5. Ауторство – без прераде. Дозвољавање умножавање, дистрибуцију и јавно саопштавање дела, без промена, преобликовања или употребе дела у свом делу, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце. Ова лиценца дозвољава комерцијалну употребу дела.

6. Ауторство - делити под истим условима. Дозвољавање умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце и ако се прерада дистрибуира под истом или сличном лиценцом. Ова лиценца дозвољава комерцијалну употребу дела и прерада. Слична је софтверским лиценцама, односно лиценцама отвореног кода.