Morfološka normalizacija tekstova na hrvatskome jeziku za dubinsku analizu i pretraživanje informacija

Šnajder, Jan

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Jan Šnajder

MORFOLOŠKA NORMALIZACIJA TEKSTOVA NA HRVATSKOME JEZIKU ZA DUBINSKU ANALIZU I PRETRAŽIVANJE INFORMACIJA

DOKTORSKA DISERTACIJA

Zagreb, 2010.

Doktorska disertacija izrađena je na

Zavodu za elektroniku, mikroelektroniku, računalne i inteligentne sustave

Fakulteta elektrotehnike i računarstva

Mentori:

prof. dr. sc. Bojana Dalbelo Bašić

prof. dr. sc. Marko Tadić,
Filozofski Fakultet Sveučilišta u Zagrebu, Ivana Lučića 3, Zagreb

Doktorska disertacija ima: 184 stranice.

Disertacija br.:

Povjerenstvo za ocjenu doktorske disertacije:

1. Akademik dr. sc. Leo Budin, profesor emeritus (u miru)
   Sveučilište u Zagrebu Fakultet elektrotehnike i računarstva

2. Dr. sc. Bojana Dalbelo Bašić, redovita profesorica
   Sveučilište u Zagrebu Fakultet elektrotehnike i računarstva

3. Dr. sc. Marko Tadić, redoviti profesor
   Sveučilište u Zagrebu Filozofski fakultet

4. Dr. sc. Nikola Bogunović, redoviti profesor
   Sveučilište u Zagrebu Fakultet elektrotehnike i računarstva

5. Dr. sc. Strahil Ristov, docent
   Institut Ruđer Bošković

Povjerenstvo za obranu doktorske disertacije:

1. Akademik dr. sc. Leo Budin, profesor emeritus (u miru)
   Sveučilište u Zagrebu Fakultet elektrotehnike i računarstva

2. Dr. sc. Bojana Dalbelo Bašić, redovita profesorica
   Sveučilište u Zagrebu Fakultet elektrotehnike i računarstva

3. Dr. sc. Marko Tadić, redoviti profesor
   Sveučilište u Zagrebu Filozofski fakultet

4. Dr. sc. Nikola Bogunović, redoviti profesor
   Sveučilište u Zagrebu Fakultet elektrotehnike i računarstva

5. Dr. sc. Strahil Ristov, docent
   Institut Ruđer Bošković

Datum obrane disertacije: 1. lipnja 2010. godine.

Sadržaj

Popis slika

Popis tablica

Poglavlje 1

Uvod

Većina ljudskog znanja zapisana je i pohranjena u tekstovnom obliku. Suvremeno informacijsko društvo temelji se na mogućnosti pristupa tom znanju, njegovom iskorištavanju i upravljanju njime. Sustavi za pretraživanje informacija čine okosnicu moderne informacijske infrastrukture, a potreba za učinkovitim i kvalitetnim pretraživanjem tekstnih informacija u stalnom je porastu. Usporedo s time sve je izraženija i potreba za boljim iskorištavanjem tih informacija, opseg kojih često nadilazi ograničene obradbene mogućnosti čovjeka. Ovi su problemi u domeni područja pretraživanja informacija (engl. *information retrieval*, IR) i dubinske analize teksta (engl. *text mining*, TM). Područje pretraživanja informacija bavi se razvojem sustava za pretraživanje dokumenata s ciljem pronalaženja onih dokumenata koji zadovoljavaju korisnikovu informacijsku potrebu (Manning *et al.*, 2008). Dubinska analiza teksta bavi se postupcima automatskog izvođenja informacije iz teksta, uključivo kategorizacijom, grupiranjem i sažimanjem dokumenata te ekstrakcijom informacija iz teksta (Feldman & Sanger, 2006). Budući da su informacije u tekstu prikazane prirodnim jezikom, oba se područja oslanjaju na spoznaje iz područje obrade prirodnoga jezika (engl. *natural language processing*, NLP), odnosno računalne lingvistike. Učinkovita rješenja iziskuju dakle interdisciplinarni pristup koji objedinjuje računarsku znanost, informacijske znanosti i lingvistiku.

1.1 Opis problema

Moderne metode pretraživanja informacija i dubinske analize teksta temelje se na modelu vektorskoga prostora (Salton *et al.*, 1975) i na statističkome jezičnom modelu (engl. *probability language model*) (Ponte & Croft, 1998). Kod oba ova modela tekstovni dokumenti prikazani su zbirkom pozicijski nezavisnih riječi, odnosno tzv. *vrećom riječi* (engl. *bag-of-words*). Premda se ovakav način prikaza u praksi pokazao iznimno

učinkovitim i isplativim – ponajviše zato što zaobilazi potrebu za složenim postupcima dublje strojne obrade i strojnog razumijevanja teksta – uz njega su ipak vezani i određeni problemi, od kojih je najizraženiji problem pojavljivanja u tekstu različitih varijanti istoga pojma. Problem se s jedne strane očituje u tome da se jedan te isti pojam u tekstu može pojavljivati u različitim morfološkim, leksičko-semantičkim ili ortografskim varijantama, a s druge u tome da se jedna te ista varijanta može odnositi na više različitih pojmova (problem višeznačnosti).

Od navedenih problema, osobito kod morfološki složenih jezika, najnotorniji je problem morfološke varijacije: umjesto da bude predstavljen jednim jedinim oblikom, pojam se raspršuje na više različitih morfoloških varijanti. U pretraživanju informacija, pojam iz korisničkog upita može se u dokumentima pojavljivati u drugačijim morfološkim varijantama, pa se može dogoditi da relevantni dokument u rezultatima bude suviše nisko rangiran, ili da uopće ne bude dohvaćen (Kraaij & Pohlmann, 1996). U dubinskoj analizi teksta, morfološke varijacije smanjuju pouzdanost statističkih procjena i stoga mogu narušiti ukupnu performansu sustava. Pored toga, morfološka varijacija povećava dimenzionalnost prostora značajki koja je za većinu postupaka dubinske analize teksta već ionako dovoljno problematična (Yang & Pedersen, 1997).

Morfološka varijacija predmet je proučavanja lingvistike, odnosno morfologije kao jedne od njezinih temeljnih grana. Morfologija proučava strukturu riječi sazdanu od morfema, najmanjih jezičnih jedinica koje imaju kakav sadržaj (Barić *et al.*, 2005). Tradicionalne gramatike tipično razlikuju dvije sastavnice morfologije: *flektivnu* i *derivacijsku*. Prva se bavi fleksijom, odnosno tvorbom oblika riječi (engl. *word-forms*); npr. *kuća – kućom* i *bogat – najbogatijih*. Oblici riječi izražavaju gramatičke značajke riječi, primjerice broj i padež kod imenica; broj, padež, rod i stupanj kod pridjeva i sl. Fleksija se uobičajeno ostvaruje dodavanjem *afikasa* (gramatičkih morfema) osnovi riječi (dijelu riječi koji nosi leksičko značenje). Većinom su to sufiksi (afiksi koji se nadovezuju na osnovu riječi), a rjeđe prefiksi (afiksi koji prethode osnovi). S druge strane, predmet proučavanja tvorbene ili derivacijske morfologije jest tvorba novih, značenjski povezanih riječi; npr. *kuća* → *kućanski* i *bogat* → *bogatstvo*. Derivacija se ostvaruje derivacijskim afiksima (sufikasima i prefiksima) ili slaganjem dviju ili više riječi.

Kako bi se uklonili negativni utjecaji morfološke varijacije, u sustavima za pretraživanje informacija i dubinsku analizu teksta uobičajeno se primjenjuje neki oblik *morfološke normalizacije*. Morfološka normalizacija podrazumijeva sažimanje morfoloških varijanti jedne ili više riječi u jedan jedinstveni oblik, morfološku *normu*. Taj postupak može se smatrati posebnim slučajem sažimanja pojmova (engl. *term confla-*

tion), odnosno grupiranja pojmova (engl. *term clustering*), koji međutim cilja ukloniti isključivo morfološku varijaciju, ali ne i druge vrste varijacije (Galvez *et al.*, 2005). U području pretraživanja informacija morfološka normalizacija prvenstveno se smatra tehnikom povećanja odziva, ali i preciznosti pri niskim razinama odziva, budući da relevantnim dokumentima može omogućiti probijanje do višega ranga (Kraaij & Pohlmann, 1996), što je kod sustava rangiranog pretraživanja informacija jednako važno.

Problem morfološke varijacije može se u sustavima za pretraživanje informacija riješiti i tehnikom (morfološkog) proširivanja upita (engl. *query expansion*), na način da se pojmovi iz upita prije pretraživanja prošire svim mogućim morfološkim varijantama (Tadić & Bekavac, 2006; Moreau *et al.*, 2007; Stanković *et al.*, 2009). Takav se pristup može koristiti ukoliko se ne želi ili se ne može mijenjati postojeći način funkcioniranja sustava. Proširenje upita međutim ne dovodi do smanjenja indeksa te usporava postupak pretraživanja, a k tome nije primjenjivo u sustavima za dubinsku analizu teksta. Zbog toga se ta tehnika u praksi primarno koristi za semantičko proširivanje upita, dok se morfološka varijacija uobičajeno rješava morfološkom normalizacijom.

Premda rana istraživanja za engleski jezik nisu bila indikativna (Harman, 1991), danas se općenito smatra da morfološka normalizacija poboljšava pretraživanje informacija (Hull, 1996). Poboljšanja uslijed primjene morfološke normalizacije u načelu su izraženija kod morfološki složenih jezika. Pokazano je da morfološka normalizacija pospješuje pretraživanje informacija u slučaju europskih jezika iz romanske, germanske i ugro-finske porodice (Tomlinson, 2003; Savoy, 2006). Za slavenske jezike to je utvrđeno za slovenski (Popovic & Willett, 1992) i češki (Majumder *et al.*, 2007a). Preliminarna istraživanja provedena za hrvatski jezik (Lauc *et al.*, 1998; Krstinić & Slapničar, 2004; Tadić & Bekavac, 2006) upućuju na slične zaključke. Kod postupaka dubinske analize teksta, utjecaj morfološke normalizacije također ovisi o jeziku. Dok za engleski (Stein & Potthast, 2007), njemački (Leopold & Kindermann, 2002; Stein & Potthast, 2007) i nizozemski (Gaustad & Bouma, 2002) nisu utvrđena značajna poboljšanja, kod morfološki složenijeg hrvatskog jezika poboljšanja su nešto izraženija (Dalbelo Bašić *et al.*, 2005; Šilić *et al.*, 2007; Malenica *et al.*, 2008). Istraživanja također sugeriraju da je utjecaj morfološke normalizacije izraženiji pri manjem broju značajki (Liao *et al.*, 2003; Malenica *et al.*, 2008). No, čak i kada poboljšanja nisu značajna, normalizacija se redovito ipak provodi kako bi se smanjila dimenzionalnost prostora značajki (Sebastiani, 2002). Tako je u (Malenica *et al.*, 2008) pokazano da je kod hrvatskog jezika uporabom morfološke normalizacije u sprezi s odgovarajućom metodom izbora značajki dimenzionalnost prostora moguće smanjiti za više od polovice, i to bez ikakvog

utjecaja na performansu sustava. Uzmemo li u obzir da prostor značajki u ovakvim slučajevima redovito poprima razmjere reda veličine 10^5, postaje jasan praktični značaj koji morfološka normalizacija ima kod postupaka za dubinsku analizu teksta.

Kod morfološke normalizacije važnu ulogu igra morfološka složenost jezika. Ta je složenost određena učestalošću afiksacije u jeziku te složenošću postupka segmentacije afikasa u pojedinim oblicima riječi (Pirkola, 2001). Hrvatski jezik, kao i većina drugih slavenskih jezika, morfološki je izuzetno složen. Primjerice, većina pridjeva u hrvatskome jeziku može poprimiti i više od 40 različitih oblika. Pored toga, hrvatski jezik je fuzijski, što znači da jedan te isti gramatički morfem može sadržavati više stopljenih gramatičkih kategorija. Zbog toga postoji problem velikog broja višeznačnih flektivnih nastavaka, tj. nastavaka koji, ovisno o tome s kojim ih se osnovama kombinira, označavaju različite morfosintaktičke značajke. Primjerice, flektivni nastavak *-e* izražava značajke genitiva jednine većine imenica ženskoga roda (npr. *vode*, *ruke*), međutim isti nastavak kod imenica muškog roda može izražavati značajku akuzativa množine (npr. *vojnike*, *zakone*). Zbog ove višeznačnosti hrvatski jezik obiluje morfološkim sinkretizmima i homografima, tj. oblicima koji se jednako pišu, ali su različitog značenja. Primjerice, nastavak *-e* u obliku *vode* može iskazivati genitiv jednine, ali također i nominativ, akuzativ ili vokativ množine (sinkretizam padeža), a isti oblik može ujedno biti oblik imenice *voda*, imenice *vod* ili glagola *voditi* (homografija). Segmentaciju afikasa u oblicima riječi hrvatskog jezika dodatno otežavaju alternacije na morfemskim granicama. Primjerice, dok nastavak *-e* u obliku *vojnike* označava akuzativ množine, isti nastavak u *vojniče* izražava vokativ jednine. Zbog ovako visokog stupnja morfološke složenosti razvoj učinkovitog postupka morfološke normalizacije velik je izazov.

1.2 Cilj, svrha i metoda istraživanja

S obzirom na visoku morfološku složenost hrvatskog jezika, a uzevši u obzir rezultate preliminarnih istraživanja, opravdano je očekivati da morfološka normalizacija tekstova na hrvatskom jeziku može dovesti do znatnih poboljšanja pri pretraživanju informacija i dubinskoj analizi teksta. Unatoč tome, postupci morfološke normalizacije hrvatskog jezika nisu još dovoljno istraženi, nisu odgovarajuće vrednovani, niti je razvijen postupak flektivno-derivacijske normalizacije koji bi bio primjenjiv za pretraživanje informacija ili dubinsku analizu teksta. Postojeća jezično nezavisna rješenja u načelu daju lošije rezultate od onih jezično specifičnih, stoga naročito u slučaju morfološki složenog hrvatskog jezika ta rješenja za većinu primjena ne mogu biti zadovoljavajuća.

Cilj istraživanja opisanog u ovom radu jest razvoj postupka za morfološku normalizaciju tekstova na hrvatskome jeziku koji bi bio pogodan za uporabu u sustavima za pretraživanje informacija i dubinsku analizu teksta. Istraživanje se temelji na dvije polazne pretpostavke. Prva je ta da je pozitivni učinak morfološke normalizacije na performanse sustava za pretraživanje informacija i dubinsku analizu teksta to veći što je veća kakvoća normalizacije. Druga je pretpostavka da je visoka kakvoća normalizacije tekstova na hrvatskome jeziku ostvariva jedino ako postupak uzima u obzir specifičnosti hrvatskoga jezika. Zbog toga je osnovni cilj ovoga rada razvoj visoko kvalitetnog postupka morfološke normalizacije prilagođenog upravo hrvatskome jeziku.

Kod morfološki složenih jezika kao što je hrvatski, visoka kakvoća normalizacije u načelu iziskuje uporabu neke vrste leksičke baze, odnosno morfološkog leksikona. Izgradnja morfološkog leksikona iziskuje golem napor i veliko stručno znanje, što ga čini iznimno skupim jezičnim resursom. U nastojanju da se taj problem zaobiđe, u okviru ovog rada razvijen je postupak za automatsko pribavljanje flektivnoga morfološkog leksikona iz neoznačenog korpusa tekstova. Time su drastično smanjeni i vrijeme i cijena izgradnje takvog resursa koji se onda, osim za morfološku normalizaciju, može koristiti i u druge svrhe. Automatska akvizicija leksikona ujedno rješava problem ograničenosti opsega leksikona, što je glavni problem pristupa temeljenih na leksikonu.

Postupak pribavljanja leksikona temelji se na računalnolingvističkom modelu morfologije hrvatskoga jezika. U okviru ovog rada osmišljen je općenit, jezično nezavisni formalni model flektivne i derivacijske morfologije temeljen na konceptima funkcijskog programiranja (Hudak, 1989). Nastojala se ostvariti visoka izražajnost modela kako bi taj čim više nalikovao opisima morfologije kakve nalazimo u tradicionalnim gramatikama, uz premisu da je takav model onda lako primjenjiv i na druge jezike. Razvijen morfološki model je dvosmjerni, omogućava dakle generiranje oblika kao i njihovu redukciju na osnovni oblik, što je kod morfološke normalizacije veoma bitno. U okviru istraživanja razrađen je cjelovit model fleksije i sufiksalne tvorbe hrvatskoga jezika, koji primjenu može naći i u drugim zadacima obrade prirodnog jezika.

Normalizacijski postupak opisan u ovome radu omogućava normalizaciju na flektivnoj, ali i na flektivno-derivacijskoj razini. Dok je korisnost flektivne normalizacije neupitna, kod derivacijske normalizacije ona ovisi o stupnju normalizacije i o konretnoj primjeni. Iz tog razloga postupak je osmišljen tako da strogo razdvaja flektivnu i derivacijsku razinu normalizacije, a stupanj derivacijske normalizacije prilagodiv je potrebama specifičnog zadatka.

Posebna pažnja posvećena je intrinzičnome vrednovanju razvijenog normalizacij-

skog postupka te usporedbi s drugim dostupnim rješenjima, uključivo Hrvatskim morfološkim leksikonom (Tadić, 2002) kao referentnim lingvističkim resursom, ali i nekim drugim, jezično nezavisnim pristupima. U radu je predložen nov način vrednovanja kakvoće normalizacijskog postupka koji omogućava da se vrednovanje provede zasebno na flektivnoj i na flektivno-derivacijskoj razini normalizacije. Time se ostvaruje dublji uvid u način funkcioniranja normalizacijskog postupka te je moguće bolje procijeniti koliko je koji postupak prikladan za specifične zadatke pretraživanja informacija i dubinske analize teksta. Vrednovanje normalizacijskog postupka na specifičnim zadacima nadilazi granice ovoga rada i ostavljeno je za buduće istraživanje.

1.3 Struktura rada

Rad je organiziran u osam poglavlja. U drugome poglavlju dan je pregled srodnih radova značajnih za istraživanje provedeno u ovome radu. Opisani su tipični pristupi morfološkoj normalizaciji te su razmotrena postojeća rješenja razvijena za hrvatski jezik. U trećem poglavlju opisan je formalni model flektivne i derivacijske morfologije kao temelj normalizacijskog postupka razvijenog u ovome radu. U četvrtom je poglavlju formalni model primijenjen na morfologiju hrvatskoga jezika. Posebno su opisana flektivna i derivacijska sastavnica modela te modeliranje fonoloških i morfonoloških alternacija. Peto poglavlje opisuje postupak akvizicije flektivnoga morfološkog leksikona temeljen na morfološkom modelu. Detaljno su opisani koraci akvizicijskog algoritma te je provedeno zasebno vrednovanje akvizicijskog postupka. U šestom poglavlju opisan je postupak flektivne i flektivno-derivacijske morfološke normalizacije temeljen na pribavljenim leksikonima, a razmotren je i problem značenjske povezanosti tvorbeno povezanih riječi. Eksperimentalno vrednovanje normalizacijskog postupka opisano je u sedmom poglavlju. Zaključak rada i smjernice budućeg istraživanja iznesene su u osmom poglavlju.

U rad su uključena i dva dodatka. U dodatku A opisana je programska izvedba morfološkog modela, akvizicijskog postupka i morfološkog leksikona u funkcijskome programskom jeziku Haskell (Jones, 2003). U dodatku B izložena je cjelovita programska izvedba modela fleksije i derivacije hrvatskoga jezika.

Poglavlje 2

Srodni radovi

Postupci morfološke normalizacije već su desetljećima predmetom istraživanja u području pretraživanja informacija, a istraživanja su danas dodatno intenzivirana sve većom potrebom za učinkovitim višejezičnim pretraživanjem informacija. U ovom je poglavlju dan pregled postupaka morfološke normalizacije, uz osvrt na srodna istraživanja vezana za hrvatski jezik te usporedbu s pristupom korištenim u ovom radu.

2.1 Taksonomija postupaka morfološke normalizacije

U literaturi su predložene različite taksonomije pristupa morfološkoj normalizaciji (Gelbukh *et al.*, 2004; Galvez & Moya-Anegón, 2006; Galvez *et al.*, 2005; Goldsmith, 2000; Baeza-Yates, 1992; Stein & Potthast, 2007). Jedna općenita i ortogonalna podjela može se, sa stajališta oblikovanja postupka normalizacije, načiniti prema:

- Vrsti obuhvaćene morfološke varijacije – na *flektivne* i *derivacijske* postupke. Derivacijski postupci u načelu obuhvaćaju i flektivnu varijaciju, pa otud možemo govoriti o flektivno-derivacijskim postupcima;
- Vrsti uporabljenog znanja – na *lingvističke* i *statističke* postupke. Lingvistički postupci oslanjaju se na morfološke leksikone, strojno čitljive rječnike, popise lema, popise sufikasa, morfološke uzorke i pravila i sl. Statistički postupci temelje se na podacima sakupljenim iz korpusa;
- Načinu izgradnje – na *ručne* i *automatske* postupke. Potonji uključuju automatsku akviziciju iz jezičnih resursa te metode (ne)nadziranog strojnog učenja (engl. *(un)supervised machine learning*);

- Načinu na koji se normalizacija provodi – na *rječničke* (engl. *dictionary-based*) postupke i one temeljene na *pravilima* (engl. *rule-based*). Pristupi temeljeni na pravilima normalizaciju provode korištenjem neke vrste pravila koja su ili eksplicitirana popisima ili su u implicitnom algoritamskom obliku. Rječnički pristupi oslanjaju se na fiksne i konačne popise parova oblika i pripadnih morfoloških normi. Mogući su i hibridni pristupi koji pravila kombiniraju s rječnikom.

Gornju je podjelu moguće dovesti u vezu sa sljedećim svojstvima:

- Kakvoća normalizacije – pristupi koji uključuju lingvističko znanje u načelu su točniji od statističkih pristupa;
- Jezična ovisnost – postupci koji uključuju lingvističko znanje u načelu su jezično zavisni i nisu prilagodivi drugim jezicima;
- Stupanj normalizacije – pristupi koji obuhvaćaju derivacijsku morfološku varijaciju po definiciji ostvaruju veći stupanj normalizacije od onih koji obuhvaćaju samo fleksiju;
- Opseg – rječnički postupci ograničenog su opsega jer oni, za razliku od pristupa temeljenih na pravilima, nemaju mogućnost generalizacije.

Sa stajališta ekonomičnosti, poželjno je da postupak morfološke normalizacije bude automatski izgrađen, jezično nezavisan i po mogućnosti neograničena opsega.

2.2 Korjenovanje i lematizacija

Dva osnovna pristupa morfološkoj normalizaciji jesu korjenovanje (engl. *stemming*) i lematizacija (engl. *lemmatisation*). U užem smislu, korjenovanje podrazumijeva uklanjanje afikasa iz oblika riječi kako bi se dobio korijen zajednički svim oblicima (Lovins, 1968). Tako dobiven korijen ne mora odgovarati pravom korijenu riječi u lingvističkome smislu, pa neki autori, kako bi naglasili ovu razliku, koriste pojam *pseudokorijen*. Također valja primijetiti da je pseudokorijen u praksi sličniji morfološkoj osnovi riječi (engl. *stem*) nego njezinom korijenu (engl. *root*) odnosno korijenskom (leksičkom) morfemu.[1] U širem smislu, korjenovanje je svaki onaj postupak koji rezultira razredima ekvivalencije morfološki povezanih riječi, tzv. korijenskim razredima (engl. *stem classes*) ili sažimajućim skupovima (engl. *conflation sets*). Korjenovanje može obuhvatiti

[1] U tom smislu bilo bi možda ispravnije *stemming* prevoditi kao "svođenje na (obličnu) osnovu". Međutim, "korjenovanje" se u hrvatskome već ustalilo (usp. Slavić (1998); Lauc (2001); Ljubešić (2009)), pa je taj izraz korišten i u ovom radu.

i flektivnu i derivacijsku morfološku varijaciju te može rezultirati različitim stupnjevima normalizacije, od slabog ili konzervativnog do jakog ili agresivnog korjenovanja.

Za razliku od korjenovanja, lematizacija je postupak kojim se pronalazi lingvistički ispravan osnovni (kanonski, natuknički) oblik neke riječi, odnosno njezina *lema*. Lematizacija također može uključivati razrješavanje homografije te morfosintaktičko označavanje, što je čini šire primjenjivom od korjenovanja. Lematizacija u načelu iziskuje više lingvističkog znanja od korjenovanja, i kod morfološki složenih jezika to se znanje izravno ili neizravno temelji na nekom obliku morfološkog leksikona. Za morfološki složene jezike lematizacija je u stvari bliska problemu morfološke analize, odnosno zadatku raščlanjivanja riječi na morfeme i njihovo označavanje.

U kontekstu morfološke normalizacije, dvije su osnovne razlike između lematizacije i korjenovanja. Prvo, za razliku od korjenovanja, lematizacija rezultira normom koja je ujedno i lingvistički ispravan oblik riječi. Drugo, lematizacija obuhvaća samo flektivnu, dok korjenovanje može obuhvatiti i derivacijsku morfološku varijaciju.

U sustavima za pretraživanje informacija i dubinsku analizu teksta morfološka normalizacija najčešće se provodi korjenovanjem: kod morfološki jednostavnih jezika, kao što je engleski, razlika između lematizacije i korjenovanja ionako je u praksi zanemariva, dok je kod morfološki složenih jezika korjenovanje u načelu ekonomski isplativije. Naime, za većinu primjena morfološke normalizacije uopće nije potrebno da morfološka norma bude lingvistički ispravan oblik; dovoljno je samo da se morfološke varijante jedne te iste riječi svedu na identičnu i jedincatu normu, kako god ona izgledala. Izbor između lematizacije i korjenovanja ovisi također i o tome kani li se normalizacijom obuhvatiti samo flektivna ili također i derivacijska morfološka varijacija; u drugom slučaju lematizacija nije dovoljna. Na primjer, lematizacijom se oblik *bogatstava* svodi na normu *bogatstvo*, dok se primjenom flektivno-derivacijskog korjenovanja isti oblik može svesti na normu *bogat*, čime se ostvaruje veći stupanj sažimanja. Čak i ako se normalizacijom cilja ukloniti samo flektivna morfološka varijacija, u mnogim slučajevima postupak flektivnog korjenovanja opet može biti isplativije rješenje od lematizacije. S druge strane, u primjenama gdje je važno da morfološka norma bude lingvistički ispravan oblik (primjerice, ako se norma prikazuje korisnicima sustava), lematizacija je preferirani način flektivne normalizacije.

2.3 Tipični pristupi morfološkoj normalizaciji

U nastavku su detaljnije razmotreni neki tipični pristupi morfološkoj normalizaciji.

2.3.1 Rječnička lematizacija

Rječnička lematizacija (engl. *dictionary-based lemmatisation*) podrazumijeva uporabu ručno sastavljenih morfoloških leksikona koji oblike neke riječi, eventualno označene dodatnom morfosintaktičkom informacijom, povezuju s pripadnom lemom (odnosno, u slučaju homografije, s više mogućih lema).[2] Rječnička normalizacija teoretski nudi apsolutnu lingvističku točnost, no izgradnja morfološkog leksikona iziskuje veliko lingvističko znanje i ogroman ljudski napor. Pored toga, rječnička normalizacija nužno je ograničenog opsega i jezično ovisna. Primjer rječničke lematizacije jest lematizacijski poslužitelj (Tadić, 2005; Tadić & Bekavac, 2006), izgrađen nad Hrvatskim morfološkim leksikonom (Tadić & Fulgosi, 2003).

Problem ograničenosti opsega morfološkog leksikona i ljudskog napora potrebnog za njegovu izgradnju u velikoj mjeri rješavaju postupci (polu)automatske akvizicije leksikona iz korpusa. Takvi postupci koriste jezično znanje u obliku računalnolingvističkog modela morfologije u sprezi sa statističkom informacijom iz korpusa kako bi iz korpusa izlučili leme i povezali ih s odgovarajućim morfološkim paradigmama. Ljudski napor potreban je u tom slučaju (samo) za razradu i razvoj računalnog modela morfologije te eventualno za povremene ručne intervencije kako bi se razriješile nezaobilazne jezične višeznačnosti. Budući da je postupak akvizicije leksikona moguće ponavljati nad različitim korpusima, problem ograničenosti opsega manje je izražen. Ovakav je pristup korišten za akviziciju morfološkog leksikona za francuski (Clement *et al.*, 2004), slovački (Sagot, 2005), češki (Hana, 2008) i ruski jezik (Oliver, 2003) te za proširivanje postojećeg Hrvatskog morfološkog leksikona (Oliver & Tadić, 2004), a sličan pristup predložen je i u okviru formalizma funkcijske morfologije (Forsberg *et al.*, 2006) (v. odjeljak 2.5). Istraživanje opisano u ovome radu, dio kojega je opisano u (Šnajder *et al.*, 2008), također pripada ovoj kategoriji.

Problematičnom se na prvi pogled kod rječničkih postupaka može činiti brzina provođenja normalizacije. Međutim, taj se problem u praksi vrlo učinkovito rješava primjenom konačnih automata, odnosno preobličivača (engl. *transducers*) (Karttunen, 2001). Preobličivači omogućavaju da se ulazni oblik riječi preoblikuje u normalizirani oblik u vremenu proporcionalnom s brojem slova ulaznog oblika, a pored toga omogućavaju znatne prostorne uštede.

[2] Takav se specijalizirani leksikon također naziva *lemarij* (Tadić, 2003).

2.3.2 Korjenovanje temeljeno na pravilima

Najrašireniji pristup korjenovanju čine algoritmi temeljeni na pravilima (engl. *rule-based stemming*), također poznati kao algoritmi uklanjanja afikasa (engl. *affix removal algorithms*). Ovi algoritmi pronalaze korijen riječi primjenom niza ručno kodiranih pravila odsijecanja (odnosno zamjene) sufikasa i prefikasa. Uobičajeno je riječ o heurističkim pravilima temeljenima na ograničenom lingvističkom znanju. Premda su, s računarskog stajališta, algoritmi korjenovanja brzo i elegantno rješenje, njihovo oblikovanje može biti poprilično izazovno. To je osobito slučaj kod morfološki složenih jezika koji intenzivno koriste afiksaciju i koji obiluju glasovnim promjenama, pa stoga iziskuju velik broj pravila.

Prvi algoritmi odsijecanja sufikasa razvijeni su za engleski jezik: *Lovinsin algoritam* (Lovins, 1968), *Porterov algoritam* (Porter, 1980) i *Paice-Huskov algoritam* (Paice, 1990). Lovinsin algoritam uklanja najdulji sufiks koji se podudara s jednim od sufikasa iz pripremljenog popisa. Kod Porterovog algoritma pravila odsijecanja sufiksa kaskadno su posložena u pet razina. Prvom razinom ostvaruje se flektivna, a ostalim četirima derivacijska normalizacija. Primjerice, pravilo

$$(m > 1) \quad ence \rightarrow \varepsilon$$

opisuje odsijecanje sufiksa -*ence* (npr. *inference*→*infer*), uz uvjet da je heuristička mjera korijena m veća od 1. Mjera m otprilike odgovara broju slogova u korijenu, a definirana je kao $[C](VC)^m[V]$, gdje V označava niz samoglasnika, C niz suglasnika, $[\cdot]$ opcionalno pojavljivanje te m broj ponavljanja niza VC. Uvjet $m > 1$ u gornjem pravilu osigurava da ono nije primjenjivo na, primjerice, oblik *defence* (mjera m korijena *def* je 1). Na taj se način mogu spriječiti mnoge pogreške prekorjenovanja (engl. *over-stemming*), odnosno pogreške pretjeranog sažimanja morfološki ili značenjski potpuno nepovezanih oblika riječi (npr. korjenovanje oblika *defence* i *define* na korijen *def*). Paice-Huskov algoritam koristi slična pravila, no primjenjuje ih iterativno sve dok ne dosegne zaustavno pravilo ili dok niti jedno pravilo više nije omogućeno.

Po uzoru na algoritme za engleski jezik, slični su algoritmi razvijeni za druge, morfološki složenije jezike, uključivo amharski (Alemayehu & Willett, 2002), arapski (Larkey *et al.*, 2002), nizozemski (Kraaij & Pohlmann, 1994; Kraaij & Pohlman, 1995), francuski (Savoy, 1999), grčki (Kalamboukis, 1995), mađarski i portugalski (Savoy, 2006), latinski (Schinke *et al.*, 1996), malezijski (Ahmad *et al.*, 1996), ruski i ukrajinski (Kovalenko, 2002), slovenski (Popovic & Willett, 1992), poljski (Weiss, 2005),

španjolski (Figuerola *et al.*, 2000) i švedski (Carlberger *et al.*, 2001). U okviru projekta *Snowball* razvijeni su i algoritmi korjenovanja za neke druge jezike.[3]

2.3.3 Hibridno korjenovanje

Unatoč tome što postavljaju uvjete za primjenu pojedinih pravila, ipak su kod algoritama odsijecanja sufiksa česte pogreške prekorjenovanja. Na primjer, Porterov algoritam (Porter, 1980) pogrešno normalizira oblike *policy* i *police* na zajednički korijen *polic*. Ovaj problem donekle se može riješiti hibridnim postupcima koji uz pravila odsijecanja sufikasa dodatno koriste rječnik. Primjer takvog postupka jest Krovetzov algoritam korjenovanja KSTEM (Krovetz, 1993) koji pravilo odsijecanja primjenjuje samo na one oblike koji nisu sadržani u rječniku. Dodatno, kako bi se spriječila normalizacija semantički nepovezanih oblika, KSTEM ograničava primjenu pravila odsijecanja derivacijskih sufikasa temeljem podudaranja u rječničkim definicijama (engl. *gloss overlap*). Ovaj pristup, inače prvi puta upotrijebljen za razrješavanje leksičke višeznačnosti (Lesk, 1986), temelji se na pretpostavci da će semantički srodne riječi biti u rječniku definirane uporabom istih riječi. Eksperimenti u kontekstu engleskog jezika pokazali su da algoritam KSTEM poboljšava preciznost sustava za pretraživanje informacija više nego Porterov algoritam, no da su razlike ipak male (manje od 5% promjene preciznosti).

Drugi pristup problemu prekorjenovanja semantički nepovezanih riječi predložen je u (Xu & Croft, 1998). Algoritam provodi naknadno particioniranje korijenskih razreda dobivenih Porterovim algoritmom na temelju statističke informacije o supojavljivanju riječi u korpusu. Particioniranje se temelji na pretpostavci da će se semantički srodni oblici blisko supojavljivati u tekstu te da samo takve oblike treba svesti na identičnu normu. Primjerice, u tekstu u domeni kulture uputno je na istu normu normalizirati oblike *izdati* i *izdavač*, no to ne vrijedi univerzalno te bi u nekoj drugoj domeni moglo biti štetno. Mjera supojavljivanja riječi a i b definirana je na sljedeći način:

$$em(a,b) = \max\Big(\frac{n_{ab} - En(a,b)}{n_a + n_b}, 0\Big)\,,$$

gdje su n_a i n_b broj pojavljivanja riječi a odnosno b u korpusu, n_{ab} je broj njihova supojavljivanja u prozoru neke određene veličine, a $En(a,b)$ je očekivani broj supojavljivanja. Očekivani broj supojavljivanja je, uz pretpostavku da su a i b statistički neovisni, definiran kao $En(a,b) = kn_an_b$, gdje je k konstanta ovisna o veličini korpusa i prozora. Očekivani broj supojavljivanja sprječava sažimanje riječi koje se slučajno supojavljuju. Primjerice, oblici *general* i *generally* gotovo se nikada neće supojavljivati

[3] `http://snowball.tartarus.org`

i stoga niti neće biti svedeni na istu normu, ali bi kod Porterovog algoritma to bio slučaj. Particioniranje korijenskih razreda provodi se tako da se oni najprije predoče grafom, a zatim se uklanjaju svi oni bridovi za koje je vrijednost mjere supojavljivanja ispod nekog predefiniranog praga. Ovakvim statističkim pristupom dobiva se konzervativnija te korpusu i domeni prilagođenija normalizacija. Eksperimenti u kontekstu engleskog jezika pokazali su međutim razmjerno mala poboljšanja u preciznosti sustava za pretraživanje informacija (manje od 5% promjene preciznosti).

2.3.4 Lematizacija temeljena na metodama nadziranog učenja

Učestali pristup morfološkoj normalizaciji, odnosno lematizaciji, predstavlja uporaba metoda nadziranog strojnog učenja (engl. *supervised machine learning*) za automatsku indukciju lematizacijskih pravila iz postojećih morfoloških leksikona ili morfološki označenih korpusa. Ovaj pristup osobito je prikladan za morfološki složene jezike koji iziskuju velik broj lematizacijskih pravila. Očit nedostatak predstavlja potreba za morfološkim leksikonom, vrlo skupim jezičnim resursom.

Tipičan primjer ovog pristupa jest sustav ATRIS za učenje lematizacijskih pravila slovenskog jezika temeljem morfološkog leksikona (Mladenić, 2002). Lematizacijska pravila definirana su kao zamjena sufiksa oblika $X \rightarrow Y$, gdje je X sufiks oblika riječi a Y je sufiks leme. Sufiksi dakako ne moraju odgovarati pravim obličnim nastavcima. Algoritam učenja koristi paradigmu slijednog pokrivanja (engl. *sequential covering*) kako bi na temelju primjera za učenje, odnosno parova $(oblik, lema)$, kombinatoričkom optimizacijom inducirao pravila zamjene sufiksa. Primjer takvog pravila jest:

$$\textbf{if } (\mathit{3LastCh} = \mathit{hom} \vee \mathit{nom} \vee \mathit{dom}) \textbf{ then } \mathit{om} \rightarrow \varepsilon \ .$$

Kombinatorička optimizacija temelji se na *2-Opt* "pohlepnom" pretraživanju (engl. *greedy search*), odnosno mijenjaju po dva literala u *ako*-dijelu pravila dok se ne dosegne maksimalna kvaliteta pravila mjerena na primjerima za učenje. Ovim je pristupom ostvarena točnost lematizacije od oko 60%, odnosno oko 75% kada se primijeni dekompozicija problema metodom slijednog modeliranja (engl. *sequential modeling*).

U (Plisson *et al.*, 2008) predložen je sličan pristup temeljen na pravilima *ripple-down* (engl. *ripple-down rules*, RDR), tradicionalno korištenima za modeliranje znanja u kontekstu ekspertnih sustava. Osnovna zamisao jest pravila strukturirati prema specifičnosti, pri čemu su specifična pravila definirana kao iznimke općenitijih opravila. Primjer takvog pravila je:

if *ed* **then** *ed*→ ε **because of** *walked*

except if *ied* **then** *ied*→*y* **because of** *classified*

Pravila se grade automatski, na način da se općenitija pravila proširuju iznimkama kad god je njihovom primjenom za neki oblik izvedena pogrešna lema (tj. lema drugačija od one definirane morfološkim leksikonom). Ostvarena točnost lematizacije nepoznatih riječi iznosi gotovo 88%, odnosno gotovo 93% kada se koristi i označivač vrste riječi.

U (Džeroski & Erjavec, 2000; Erjavec & Džeroski, 2004) opisan je pristup indukcije lematizacijskih pravila temeljem induktivnog logičkog programiranja (engl. *inductive logic programming*, ILP). Pravila su inducirana iz morfološki označenog korpusa i morfološkog leksikona, a točnost lematizacije nepoznatih riječi iznosi 92%, odnosno više od 98% ako se koristi i označivač vrste riječi.

2.3.5 Nenadzirani pristupi korjenovanju

Korjenovanje temeljeno na pravilima problematično je kod jezika visoke morfološke složenosti, dok je strojno učena lematizacija problematična kod jezika za koje ne postoje ili nisu dostupni morfološki leksikoni. Alternativu predstavljaju metode nenadziranog strojnog učenja (engl. *unsupervised machine learning*), kojima se rječnici ili pravila korjenovanja induciraju automatski iz neoznačenih korpusa. Prednost ovakvih postupaka jest ta što su oni u načelu jezično neovisni. Pristupi se razlikuju po tome izvode li neko novo znanje iz postojećeg lingvističkog znanja (npr. Xu & Croft (1998); Gaussier (1999)) ili indukciju obavljaju bez ikakvog apriornog znanja (engl. *knowledge-free*), oslanjajući se isključivo na statističke informacije iz korpusa (npr. Goldsmith (2001); Schone & Jurafsky (2001); Melucci & Orio (2003); Gelbukh *et al.* (2004)). Pristupi se također razlikuju prema tome ciljaju li grupirati morfološki srodne riječi (problem grupiranja), raščlaniti riječ na morfeme (problem segmentacije) ili identificirati pojedinačne morfeme (problem morfološke analize).[4] Ovi pristupi, za koje se u literaturi često koristi naziv "indukcija morfologije" (engl. *morphology induction*), u posljednih se pet godina intenzivno istražuju u okviru radionica pod nazivom *Morpho challenge* (Kurimo *et al.*, 2009).[5]

Jedan od najranijih nenadziranih pristupa korjenovanju temelji se na tzv. raznolikosti sljedbenika (engl. *successor variety*) (Hafer & Weiss, 1974). Raznolikost sljedbenika jest broj različitih slova koja se mogu pojaviti nakon nekog prefiksa. Metoda

[4] Za morfološku normalizaciju od interesa je samo problem grupiranja morfološki srodnih riječi.

[5] `http://www.cis.hut.fi/morphochallenge2010/`

predložena u (Hafer & Weiss, 1974) pronalazi korijen riječi temeljem pretpostavke da se duljinom prefiksa raznolikost sljedbenika smanjuje, da bi na granici korijena naglo porasla.[6] Granicom korijena može se proglasiti slovo čija raznolikost sljedbenika prelazi neki predefinirani prag (metoda odsijecanja), slovo za koje je raznolikost sljedbenika veća od prethodnog slova i idućeg slova (metoda vrha i platoa) i sl. Pristup temeljen na raznolikosti sljedbenika, u kojemu se koristilo heurističko poboljšanje metode vrha i platoa, isproban je u (Stein & Potthast, 2007) na engleskom i njemačkom jeziku. Na zadatku grupiranja dokumenata postupak se pokazao tek neznatno lošijim od korjenovanja temeljenog na pravilima.

Tipičan primjer nenadziranog pristupa korjenovanju jest grupiranje morfološki povezanih oblika riječi u klase ekvivalencije (korijenske klase) temeljem neke mjere sličnosti (odnosno mjere udaljenosti) znakovnih nizova. Iz dobivenih grupa riječi moguće je zatim konstruirati normalizacijske rječnike ili inducirati pravila korjenovanja. U (Adamson & Boreham, 1974) mjera sličnosti SC (engl. *similarity coefficient*) dviju riječi temelji se na broju podudarajućih podnizova fiksne duljine, odnosno *n-grama*. Ta je mjera definirana Diceovim koeficijentom (Dice, 1945) na sljedeći način:

$$\mathrm{SC} = \frac{2 \cdot (\textit{broj zajedničkih unikatnih n-grama})}{\textit{zbroj unikatnih n-grama u svakom nizu}} \,.$$

Korištenjem ove mjere provodi se zatim grupiranje temeljeno na minimalnoj udaljenosti (engl. *single linkage clustering*) u ovisnosti o heuristički postavljenom pragu. Ovaj je pristup uspješno primijenjen na mnoge jezike, uključivo engleski (Freund & Willett, 1982), turski (Ekmekcioglu *et al.*, 1996) i hrvatski (Šnajder & Dalbelo Bašić, 2009). U (Roeck & Al-Fares, 2000) opisana je prilagodba pristupa za arapski jezik u kojoj se, zbog učestale infiksacije u jeziku i manje prosječne duljine riječi, umjesto Diceovog koeficijenta koristi Jaccardov koeficijent. Pored toga, grupiranje se provodi tek nakon primjene konzervativnog korjenovanja temeljenog na pravilima.

Sličan pristup, primijenjen na bengalski jezik, opisan je u (Majumder *et al.*, 2007b). Ondje je za mjeru sličnosti (odnosno mjeru udaljenosti) predložena, pored ostalih, sljedeća mjera:

$$D_4 = \frac{n - m + 1}{n + 1} \sum_{i=m}^{n} \frac{1}{2^{i-m}} \,, \tag{2.1}$$

gdje je m mjesto prvog nepodudaranja znakovnih nizova (brojano slijeva nadesno),

[6] Pristup je inspiriran strukturalnom lingvistikom, kod koje se granice morfema nastoje utvrditi temeljem distribucije fonema (Baeza-Yates, 1992).

a $n+1$ je duljina nizova (kraći od dva niza nadopunjuje se zdesna). Intuitivno, mjera nagrađuje dug zajednički prefiks te kažnjava svako daljnje nepodudaranje znakova. Primjerice, za riječi *astronomer* i *astronomically* vrijedi $m = 8$, $n = 13$, pa $D_4 = \frac{6}{13}(\frac{1}{2^0} + \cdots + \frac{1}{2^{13-8}}) = 0,84$. S ciljem dobivanja što kompaktnijih grupa, za grupiranje je korištena maksimalna, a ne minimalna udaljenost (engl. *complete linkage clustering*). Kako bi se pronašla optimalna vrijednost praga, napravljena je analiza broja grupa u ovisnosti o vrijednosti praga te je kao optimalna uzeta ona vrijednost oko koje se broj grupa stabilizira. Eksperimenti u kontekstu engleskog jezika pokazali su da je korjenovanje provedeno na ovaj način usporedivo s Porterovim algoritmom (Porter, 1980) u smislu učinaka na performanse sustava za pretraživanje. Pristup je uspješno primijenjen i na mađarskom, češkom (Majumder *et al.*, 2007a) i hrvatskom jeziku (Šnajder & Dalbelo Bašić, 2009).

U (Gaussier, 1999) opisan je pristup za indukciju pravila derivacijske morfologije temeljem flektivnoga morfološkog leksikona. Postupak najprije grupira morfološki srodne leme iz leksikona, a zatim iz dobivenih grupa izlučuje pravila sufiksacije za tvorbu jedne leme iz druge, gradeći na taj način za svaku grupu pripadna derivacijska stabla. Takva se pravila mogu koristiti za derivacijsko korjenovanje, ali i kao pomoć leksikografima pri izradi jezičnih resursa.

Mnogi pristupi indukciji morfologije iskušavaju i kombiniraju različite metode nenadziranog strojnog učenja. U (Goldsmith, 2000; Creutz & Lagus, 2002) opisan je postupak za indukciju pravila odsijecanja sufikasa temeljen na algoritmu najmanje duljine opisa (engl. *minimum description length*, MDL). U (Melucci & Orio, 2003) opisan je postupak koji koristi skriveni Markovljev model (engl. *hidden Markov model*, HMM) za pronalaženje najvjerojatnijeg korijena riječi. U (Gelbukh *et al.*, 2004) opisan je postupak izlučivanja skupa korijena i sufikasa temeljen na minimizaciji broja elemenata u tim skupovima uporabom genetičkog algoritma. U (Schone & Jurafsky, 2001) opisan je postupak koji kombinira više vrsta lingvističkoga znanja – ortografiju, semantiku i sintaksu – te koristi analizu raznolikosti sljedbenika i latentnu semantičku analizu (engl. *latent semantic analysis*, LSA) kako bi iz korpusa izlučio klase ekvivalencija ne samo morfološki već i semantički srodnih riječi. Sličan je postupak opisan u (Baroni *et al.*, 2002), koji riječi grupira temeljem ortografske i semantičke sličnosti, modelirane na temelju supojavljivanja riječi u dokumentima. Zanimljiv je i postupak opisan u (Bernhard, 2010), kojim se oblici riječi najprije grupiraju temeljem sličnosti znakovnih nizova, a zatim se na tako dobivene grupe primjenjuju metode analize mreža s ciljem da ih se razdijeli u korijenske razrede optimalne veličine.

2.4 Srodni radovi za hrvatski jezik

Računalna obrada morfologije hrvatskoga jezika bila je i još je uvijek predmetom raznih istraživanja.[7] Ta se istraživanja ugrubo mogu razdijeliti u dva pravca. Prvi čine lingvistički ili leksikografski motivirana istraživanja, koja ciljaju razviti općenite jezične resurse i alate za obradu morfologije hrvatskoga jezika. Drugi pravac predstavljaju ciljana nastojanja rješavanja problema morfološke varijacije u sustavima za pretraživanje informacija, bez ambicija za ostvarenjem lingvistički vjerodostojnih modela.

2.4.1 Računalnolingvistički resursi i alati

Jedan od najranijih jezičnih resursa za hrvatski jezik jest Rječnička baza hrvatskoga književnog jezika (Kržak & Boras, 1985), zasnovana na generativnom modelu kojim je oblike riječi bilo moguće izvoditi iz nekih ishodišnih oblika. Kao ishodišni oblik uzet je početak riječi, na koji su onda vezivani nastavci ovisno o pridruženoj paradigmi. Svaka je natuknica u bazi mogla imati više različitih početaka te jedan skup nastavaka. Kasnije je organizacija baze promijenjena tako da je svaka riječ imala samo jednu osnovu (Kržak, 1988), koja je onda bila jednaka nepromjenjivom dijelu riječi, dok su promjenjivi dijelovi tretirani kao nastavci. Velik su izazov u to vrijeme predstavljali ograničeni kapaciteti računala, pa je ovakav način sažimanja popisa oblika predstavljao značajnu uštedu. S druge strane, ovakav pristup, budući da ne poštuje jezične zakonitosti o segmentaciji riječi na morfeme (npr. *vuk-a*, ali *vu-če*), nije lingvistički opravdan i može se smatrati "informatičkim pristupom" obradi morfologije (Tadić, 1992).[8]

U tom smislu posve drugačiji pristup jest dvorazinski model morfonoloških alternacija razrađen u (Lopina, 1992). Dvorazinski model, izvorno predložen u (Koskenniemi, 1983), razlikuje tzv. dubinsku i površinsku razinu prikaza riječi. Dubinska (rječnička) razina prikazuje temeljni oblik riječi, iz kojega se zatim morfonološkim preoblikama mogu na površinskoj razini izvesti različiti oblici. Tako bi, primjerice, dubinski prikaz oblik *vuče* bio morf *vuk*, dok bi površinski prikaz sačinjavali alomorf *vuč* i oblični nastavak *-e*. Morfonološke preoblike povezuju dubinski i površinski prikaz te mogu biti uvjetovane okolinom fonema ili grafema s jedne i s druge razine. Takve preoblike mogu

[7] Zapravo, računalnoj obradi morfologije posvećeno je mnogo više (dužne) pažnje nego ostalim razinama obrade hrvatskog jezika. To je i razumljivo s obzirom da je morfološka razina polazna razina razvoja jezičnih alata svakog jezika, a posebice onih morfološki složenih. Pregled razvitka jezičnih tehnologija s osvrtom na (tadašnje) stanje razvoja jezičnih tehnologija za hrvatski jezik može se naći u (Tadić, 2003).

[8] Time se dakako ne želi reći da je pristup neprikladan; prikladnost pristupa ovisi o svrsi koja moža biti više lingvistički ili više informatički orijentirana. Tako se u (Tadić, 1994) naznačuje razlika između "strogoinformatički" i "lingvistički" orijentiranih pristupa; s računarskog/informatičkog stajališta, možda je pak prikladnije govoriti o "informatičkom" i "strogolingvističkom" pristupu.

se u računalu učinkovito prikazati pretvaračem (engl. *transducer*) s konačnim brojem stanja (Karttunen, 2001).

Dvorazinski model tipičan je primjer tzv. morfemno-leksemnog pristupa opisu morfologije, poznatog i pod nazivom *item-and-process*, koji morfološke procese opisuje na razini pojedinačnih leksema (Aronoff & Fudeman, 2005). Takav pristup nije prikladan za opis morfologije fuzijskih jezika (Spencer, 1991), kakav je i hrvatski jezik.[9] Za opis morfologije fuzijskih jezika prikladniji je model koji se, prema (Hockett, 1954), naziva *morfologija s rječnom osnovom* (engl. *word-based morphology*),[10] odnosno model "riječ i paradigma" (engl. *word-and-paradigm*). Kod ovog pristupa, koji se čini najbližim tradicionalnim gramatičkim pristupima, preoblike su definirane u ovisnosti o gramatičkim kategorijama (koje onda mogu biti stopljene) te o paradigmi kojoj riječ pripada.

Na takvom se tradicionalnom pristupu temelji generativni model flektivne morfologije hrvatskog jezika GENOBLIK, razrađen i ostvaren u (Tadić, 1994). Model se sastoji od triju popisa: popisa lema s osnovama (lemarija), popisa nastavaka i popisa preoblika. U popisu lema svakoj su lemi pridruženi uzorci promjene kojima je definirano generiranje svih njezinih oblika. Ukupno je za opis hrvatske fleksije u modelu GENOBLIK definirano 613 uzoraka promjene, od kojih 404 imeničkih, 64 pridjevskih (posebno za deklinaciju i za komparaciju) te 155 glagolskih. Popis nastavaka služi kao poveznica između uzoraka i morfosintaktičkih kategorija s jedne strane i odgovarajuće preoblike s druge strane. Popis preoblika sastoji se od 35 preoblika koje u potpunosti obuhvaćaju izvođenje svih alomorfnih varijacija svih hrvatskih flektivnih osnova (Tadić, 2003). Na ovom se generativnom modelu temelji Hrvatski morfološki leksikon (Tadić & Fulgosi, 2003), referentni računalnolingvistički resurs koji u ovome trenutku sadržava više od 100.000 lema i gotovo 4 milijuna različitih oblika.[11]

Recentni doprinos računalnoj obradi morfologije hrvatskoga jezika predstavlja alat za morfološku analizu i lematizaciju CroMo (Ćavar *et al.*, 2008). CroMo je ostvaren kao preobličivač s konačnim brojem stanja (FST) izgrađen nad leksičkom bazom od 120.00 morfema i alomorfa. Za razliku od gore opisanih pristupa, posebna pažnja posvećena je računalnoj optimizaciji modela.

Bitno je napomenuti da su svi gore navedeni pristupi primarno lingvistički motivirani, u smislu da ciljaju na lingvističku ispravnost i stoga imaju visoku cijenu razvoja.

[9] Fuzijski jezici od tzv. aglutinativnih razlikuju se po tome što jednim morfemom mogu izraziti više gramatičkih kategorija, zbog čega su onda u takvim jezicima česti i morfološki sinkretizmi. Morfemno-leksemni pristupi pretpostavljaju, međutim, aglutinativnu strukturu riječi (i to ako već ne na površinskoj, onda svakako na dubinskoj razini prikaza).

[10] Rasprava o problemu prijevoda ovih pojmova na hrvatski jezik može se naći u (Birtić, 2006).

[11] `http://hml.ffzg.hr`

2.4.2 Postupci morfološke normalizacije

Do sada razvijeni postupci za morfološku normalizaciju tekstova na hrvatskom jeziku fokusirani su uglavnom na problem flektivne morfološke varijacije u sustavima za pretraživanje informacija. Prvi takav sustav, namijenjen doduše morfološkom proširivanju upita, a ne normalizaciji, jest robusni model za prepoznavanje i izvođenja oblika riječi ROMO (Lauc *et al.*, 1998; Lauc, 2001). Model se temelji na pravilima za izvođenje flektivnih oblika imenica i pridjeva. Postupak proširivanja provodi se u dva koraka: u prvome se riječi iz upita primjenom pravila najprije svode na sve moguće osnovne oblike, a zatim se u drugom koraku iz svake tako dobivene osnove izvode svi mogući oblici. Pritom se nakon odbacivanja nastavka u prvom koraku, a prije dodavanja nastavka u drugom koraku, na međuoblik primjenjuju pravila morfonološke alternacije, kojih je u modelu definirano ukupno dvadesetak. Ta su pravila definirana po uzoru na ona iz dvorazinskog modela (Lopina, 1992), ali su pojednostavljena tako da se primjenjuju na površinskoj razini. Za izvođenje oblika ne koristi se dakle niti rječnik niti flektivni uzorci, što rezultira velikim brojem jezično neispravnih oblika. Zbog toga je potrebno provesti dodatno filtriranje kroz popis riječi iz korpusa tekstova koji se pretražuje.

U (Krstinić & Slapničar, 2004) opisan je pristup morfološkoj normalizaciji temeljen na pravilima ugrađenima u pravopisni provjernik *Ispell*. Tim je pravilima, čiji je ukupan broj ograničen na 26, opisan način na koji se iz leme izvode svi njezini oblici (pravila dakle odgovaraju flektivnim uzorcima). Ručno je sastavljen popis od 20.000 lema (imenica i glagola) i njima pridruženih pravila. Temeljem tog popisa moguće je iz lema generirati pripadne oblike riječi, čime se dobiva rječnik koji se može koristiti za normalizaciju.

U (Ljubešić *et al.*, 2007) opisano je korjenovanje temeljeno na manjem broju ručno sastavljenih pravila odsijecanja sufiksa, čiji se optimalni podskup utvrđuje optimizacijskom metodom. Postupak je namijenjen uporabi u sustavima za pretraživanje informacija te je razvijen uz pretpostavku da se riječi u upitu pojavljuju u osnovnom obliku.[12] To međutim u bitnome ograničava primjenu postupka, a također ga čini neuporabivim u sustavima za dubinsku analizu teksta.

Znatno više lingvistički orijentiran jest već spomenuti postupak rječničke lematizacije temeljen na Hrvatskom morfološkom leksikonu (Tadić, 2005; Tadić & Bekavac, 2006). U (Agić *et al.*, 2009) opisana je nadogradnja tog pristupa koja lematizaciju

[12]Ta pretpostavka nije eksperimentalno provjerena, a u pitanje je dovodi činjenica da su korisnički upiti nerijetko imeničke skupine čije se pojedinačne riječi mogu javljati u različitim padežima (npr. u genitivu, budući da je imenički atribut često izražen upravo genitivom). Više o zastupljenosti padeža u hrvatskome jeziku može se naći u (Kolaković, 2007).

kombinira s postupkom označivanja vrste riječi. Na taj je način moguće razriješiti višeznačnost uslijed homografije, tj. oblik je moguće svesti na ispravnu lemu u zavisnosti od rečeničnog konteksta u kojemu se on pojavljuje. Točnost lematizacije koja se pri tome ostvaruje iznosi gotovo 97%, što ukazuje na značajno poboljšanje u odnosu na slučaj kada se označivač riječi ne koristi. Razrješavanje homografije pri morfološkoj normalizaciji moglo bi pridonijeti njezinoj djelotvornosti, što međutim tek valja eksperimentalno utvrditi. Da problem možda nije tako ključan, ukazuje preliminarna analiza udjela homografije u tekstovima na hrvatskom jeziku provedena u (Tadić, 2003), gdje je utvrđen razmjerno nizak udio homografa od svega 1,3%.[13]

Prethodno navedeni pristupi morfološkoj normalizaciji jezično su specifični i namijenjeni su upravo hrvatskom jeziku. Pored toga, svi navedeni postupci ograničeni su na flektivnu razinu normalizacije. Utoliko drugačiji jest pristup opisan u (Šnajder & Dalbelo Bašić, 2009), gdje je iskušan nenadzirani, jezično nezavisni pristup korjenovanju temeljen na sličnosti znakovnih nizova. Slično kao i u (Majumder *et al.*, 2007b), za grupiranje oblika upotrijebljen je hijerarhijski aglomerativni algoritam, ali je umjesto metode maksimalne udaljenosti korištena metoda prosječne udaljenosti (engl. *average linkage*). Pritom se s računalnog stajališta problematičnim pokazao velik broj oblika (njih više od 560.000), pa je prije grupiranja oblike bilo potrebno razdijeliti u manje grupe temeljem usporedbe prefiksa. Vrednovanje, provedeno zasebno na flektivnoj i na derivacijskoj razini, pokazalo je da kakvoća normalizacije doseže 85% (v. odjeljak 7.2.5), što je dobar rezultat ako se u obzir uzme jednostavnost i jezična nezavisnost postupka. Između više ispitanih mjera udaljenosti znakovnih nizova najboljom se pokazala mjera dana izrazom (2.1).

2.5 Model funkcijske morfologije

Pristupu opisanom u ovom radu donekle je srodan pristup temeljen na modelu funkcijske morfologije (Forsberg & Ranta, 2003).[14] Funkcijska morfologija programsko je okruženje za modeliranje flektivne morfologije u funkcijskome programskom jeziku Haskell (Jones, 2003). Osnovna je zamisao iskoristiti visoku ekspresivnost funkcijskog programskog jezika kako bi se morfologija definirala na visokoj razini apstrakcije, odnosno na razini višoj od razine regularnih izraza, koji se u tu svrhu tipično koriste.

[13] Ovaj se udio odnosi na tzv. "vanjske homografe", odnosno slučajeve u kojima jedna pojavnica ima više mogućih lema. "Unutarnja homografija" odnosi se na slučajeve morfološkog sinkretizma kada jedna pojavnica predstavlja različite oblike iste leme (Tadić, 2003). U kontekstu morfološke normalizacije problematičnom bi se mogla pokazati samo vanjska homografija.

[14] `http://www.cs.chalmers.se/ markus/FM/`

Funkcijska morfologija flektivne uzorke apstrahira funkcijama, osnovnim gradivnim elementima funkcijske programske paradigme. Takve su funkcije definirane nad konačnim skupom algebarskih tipova[15] koji odgovaraju gramatičkim kategorijama (padež, rod, broj i sl.) Primjenom takve funkcije na zadanu lemu i skup gramatičkih značajki iz leme je moguće generirati odgovarajući oblik, dok je iteriranjem po svim dopuštenim vrijednostima algebarskog tipa neke paradigme moguće izvesti sve oblike neke leme. Zahvaljujući vrlo visokoj izražajnosti programskog jezika Haskell, model je moguće definirati vrlo sažeto i pregledno, a proširivanje modela novim uzorcima razmjerno je jednostavno. Funkcijski modeli morfologije razvijeni su za mnoge jezike, uključivo latinski, švedski, talijanski, španjolski i ruski.

Funkcijska morfologija zapravo je generativni model morfologije. To znači da taj model nije moguće koristiti za lematizaciju, odnosno općenito morfološku normalizaciju. Zbog toga se u (Forsberg *et al.*, 2006) predlaže postupak za ekstrakciju leksikona iz korpusa temeljem modela funkcijske morfologije. U tu svrhu model se proširuje dodatnim pravilima, tzv. ekstrakcijskim pravilima, koja definiraju uzorke za izvođenje oblika iz oblične osnove. Na primjer, uzorkom `x+"a"` određeno je izvođenje oblika dodavanjem nastavka *-a* obličnoj osnovi `x`. Više ovakvih uzoraka mogu se kombinirati logičkim operatorima kako bi se definirao uvjet koji paradigma (odnosno flektivni uzorak definiran funkcijom modela) mora zadovoljavati. Na ovaj je način moguće vrlo precizno specificirati koji oblici moraju biti ovjereni u korpusu (uporabom operatora logičkog "i"), koji su oblici opcionalni (uporabom operatora logičkog "ili") te koji se oblici ne bi smjeli pojavljivati u korpusu (uporabom logičke negacije). Dodatna ograničenja moguće je definirati regularnim izrazima.

2.6 Pristup korišten u ovome radu

Istraživanje provedeno u okviru ovoga rada oslanja se na ranija istraživanja opisana u (Šnajder *et al.*, 2008), (Šnajder & Dalbelo Bašić, 2008) i (Šnajder & Dalbelo Bašić, 2009). Razvijeni postupak morfološke normalizacije temelji se na morfološkom leksikonu, modelu flektivne i derivacijske morfologije te postupku akvizicije flektivnoga leksikona iz korpusa.

Flektivna sastavnica morfološkog modela temelji se, baš kao i modeli opisani u (Tadić, 1994; Forsberg & Ranta, 2003), na modelu "riječ i paradigma". Premda je u

[15] U funkcijskoj programskoj paradigmi algebarski tip odnosi se na tip podataka čije su vrijednosti preuzete od drugog tipa podataka, ali se dodatno obilježene tzv. konstruktorom (usp. Hudak (1989)).

konačnici namijenjen morfološkoj normalizaciji, model je lingvistički[16] orijentiran te u velikoj mjeri poštuje jezične zakonitosti; nije dakle primjerom "informatičkog pristupa" kao što su to, primjerice, modeli (Kržak, 1988; Krstinić & Slapničar, 2004). Slično kao i u (Lopina, 1992; Lauc *et al.*, 1998), posebna je pažnja posvećena modeliranju fonoloških alternacija. Nadalje, za razliku od modela opisanih u (Tadić, 1994; Forsberg & Ranta, 2003), model razvijen u okviru ovog rada jest generativno-redukcijski, omogućava dakle kako generiranje oblika tako i svođenje oblika na osnovni oblik. Model je, slično kao i onaj opisan u (Forsberg & Ranta, 2003), inspiriran funkcijskom programskom paradigmom, ali je međutim postavljen na višu razinu apstrakcije i nije vezan za konkretan programski jezik. Visoka razina apstrakcije i koncepti funkcijskog programiranja model čine sažetijim i preglednijim od onog opisanog u (Tadić, 1994). Konačno, model se od prethodno razvijenih modela morfologije hrvatskog jezika razlikuje po tome što, pored fleksije, obuhvaća i derivaciju.[17]

Razvijeni morfološki model ne koristi se izravno za morfološku normalizaciju, kao što je to slučaj kod (Lauc *et al.*, 1998). Umjesto toga, a slično kao i kod (Clement *et al.*, 2004; Oliver & Tadić, 2004; Forsberg *et al.*, 2006; Hana, 2008), model se koristi za akviziciju flektivnoga leksikona. Razlika je međutim u tome što se ne cilja na lingvističku besprijekornost pribavljenog leksikona, budući da to za morfološku normalizaciju nije potrebno, pa je akviziciju moguće posve automatizirati. Nadalje, budući da je model generativno-redukcijski, nema potrebe za definiranjem dodatnih pravila za ekstrakciju, kao što je to slučaj kod (Forsberg *et al.*, 2006). Postupak akvizicije rješava također problem ograničenog opsega leksikona, što je osnovni problem normalizacijskih postupaka temeljenih na leksikonu, poput onog opisanog u (Tadić, 2005).

Slično kao i kod (Gaussier, 1999), iz flektivnog se leksikona grupiranjem derivacijski povezanih unosaka izgrađuje flektivno-derivacijski leksikon. To u konačnici omogućava provođenje normalizacije i na derivacijskoj razini, po čemu se ovaj postupak također razlikuje od drugih do sada razvijenih postupaka za hrvatski jezik.

Sukladno taksonomiji izloženoj u 2.1, postupak razvijen u okviru ovog rada mogli bismo dakle klasificirati kao:

- flektivni i derivacijski,
- lingvistički (koristi morfološki model) i statistički (akvizicija leksikona temelji se na statističkoj informaciji iz korpusa),

[16] No ne i "strogolingvistički".

[17] Modelom je obuhvaćena samo sufiksalna tvorba hrvatskoga jezika, budući da u kontekstu morfološke normalizacije prefiksalna tvorba nije od interesa. Ne postoji međutim prepreka da se modelom u budućnosti obuhvati i prefiksalna tvorba.

- poluautomatski te
- rječnički (normalizacija se u konačnici provodi pomoću leksikona).

Prednosti ovakvog pristupa mogu se sažeti u sljedećem:

1. Kakvoća normalizacije – Budući da se oslanja na leksikon, normalizacija je mnogo preciznija od one koja bi se mogla ostvariti lematizacijskim pravilima, pravilima odsijecanja sufiksa ili izravnom uporabom morfološkog modela;
2. Promjenjiv stupanj normalizacije – Postupak strogo odjeljuje flektivnu razinu normalizaciju od derivacijske. Flektivnim leksikonom normalizacija se provodi točno na flektivnoj razini, dok se flektivno-derivacijski leksikonima mogu ostvariti veći stupanjevi normalizacije;
3. Odabir normi – Uporaba leksikona omogućava da se kao morfološke norme koriste leme riječi (ili barem lingvistički ispravni oblici), što je bitno u slučajevima kada se norme prikazuju korisnicima. U tom smislu postupak predstavlja kombinaciju lematizacije i korjenovanja;
4. Razgraničavanje homografa – Normalizacija temeljem leksikona omogućava razgraničavanje (no ne i razrješavanje) homografije. Za razliku od korjenovanja, kojim se homografni oblici pogrešno svode na identični korijen, ovdje opisani postupak homografne oblike svodi na više odgovarajućih normi. Na taj se način smanjuje gubitak informacije, a otvara se mogućnost primjene postupka za razrješavanje homografije.

Istraživanje provedeno u okviru ovog rada od prethodnih se istraživanja razlikuje i po tome što je provedeno iscrpno intrinzično vrednovanje kakvoće normalizacijskog postupka.

Poglavlje 3

Formalni model flektivne i derivacijske morfologije

U ovome poglavlju opisan je jezično neovisni računalnolingvistički model flektivne i derivacijske morfologije. Model je opisan putem triju razina apstrakcije. Na najvišoj razini definiran je apstraktni funkcijski model kao okvir za daljnju razradu modela. U drugom odjeljku definiran je model temeljen na funkcijama višeg reda kao specijalizacija apstraktnog funkcijskog modela. Daljnja nadogradnja modela opisana je u trećem odjeljku.

3.1 Apstraktni funkcijski model

Apstraktni funkcijski model opisuje fleksiju i tvorbu riječi na najvišoj razini apstrakcije. Opis se temelji na modelu "riječ i paradigma" (engl. *word-and-paradigm*), za koji je već napomenuto da je prikladan način opisa flektivne morfologije fuzijskih jezika. Model sačinjavaju dvije sastavnice: flektivna i derivacijska. Flektivna se sastavnica sastoji od generativnog i redukcijskog dijela: generativnim dijelom opisano je generiranje oblika riječi na temelju ishodišnog oblika, a redukcijskim dijelom obrnut postupak svođenje oblika riječi na ishodišni oblik. Derivacijskom sastavnicom opisano je izvođenje riječi iz ishodišnog oblika.

3.1.1 Flektivna sastavnica modela

Tradicionalne gramatike fleksiju (odnosno tvorbu oblika) opisuju pomoću flektivnih uzoraka. Flektivni uzorci opisuju način na koji se iz oblične osnove tvore oblici koji izražavaju određene morfološke kategorije. Flektivni su uzorci u apstraktnome funkcij-

skom modelu apstrahirani skupom $\mathcal{F}$, a oblici riječi i njihove oblične osnove predstavljeni su skupom $\mathcal{S}$.

Generativni dio

Generativni dio flektivne sastavnice modela sačinjavaju tri funkcije:

$$sWfs \quad : \quad \mathcal{S} \times \mathcal{F} \to \wp(\mathcal{S}) \,, \tag{3.1}$$

$$sWfsMsd \quad : \quad \mathcal{S} \times \mathcal{F} \to \wp(\mathcal{S} \times \mathcal{X}) \,, \tag{3.2}$$

$$sLemma \quad : \quad \mathcal{S} \times \mathcal{F} \to \mathcal{S} \,. \tag{3.3}$$

Funkcija *sWfs* opisuje izvođenje svih oblika riječi iz oblične osnove temeljem zadanog flektivnog uzorka iz skupa $\mathcal{F}$. Isto vrijedi i za funkciju *sWfsMsd*, s tom razlikom da ta funkcija svakom izvedenom obliku dodatno pridjeljuje morfosintaktički opis iz skupa $\mathcal{X}$.[1] Funkcija *sWfs* može se definirati pomoću funkcije *sWfsMsd* tako da se morfosintaktički opisi jednostavno odbace:

$$lWfs(l, f) = \left\{ w \ : \ (w, x) \in lWfsMsd(l, f) \right\}. \tag{3.4}$$

Vezu između oblične osnove i leme uspostavlja funkcija *sLemma*, koja obličnu osnovu temeljem zadanog flektivnog uzorka $f \in \mathcal{F}$ preslikava u lemu riječi. Primjerice, $sLemma(slik, f) = slika$.

Redukcijski dio

Redukcijski dio flektivne sastavnice modela definiran je također trima funkcijama:

$$lStem \quad : \quad \mathcal{S} \times \mathcal{F} \to \mathcal{S} \,, \tag{3.5}$$

$$wStem \quad : \quad \mathcal{S} \times \mathcal{F} \to \mathcal{S} \,, \tag{3.6}$$

$$wStemMsd \quad : \quad \mathcal{S} \times \mathcal{F} \to \wp(\mathcal{S} \times \mathcal{X}) \,. \tag{3.7}$$

Funkcija *lStem* preslikava lemu u obličnu osnovu temeljem zadanog flektivnog uzorka. Na primjer, $lStem(slika, f) = slik$. Funkcija je dakle suprotna funkciji generativnog

[1] Morfosintaktički opisi nisu potrebni za postupak normalizacije opisan u ovome radu, no ipak su radi općenitosti ugrađeni u model.

modela *sLemma*. Slično, funkcija *wStem* temeljem flektivnog uzorka obliku riječi pridružuje odgovarajuću obličnu osnovu; npr. $wStem(slici) = slik$. Funkcija *wStemMsd* proširenje je ove funkcije koje obliku, osim oblične osnove, pridjeljuje i morfosintaktičku oznaku. Ovdje u obzir treba uzeti moguću višeznačnost uslijed (vanjske) homografije, što je u modelu riješeno preslikavanjem u skup mogućih rezultata. Funkcija *wStem* može se definirati temeljem funkcije *wStemMsd* odbacivanjem morfosintaktičkih opisa.

Vezu između generativne i redukcijske sastavnice modela izražavaju sljedeće dvije ekvivalencije:

$$lStem(sLemma(f,s),f) = s\,,$$
$$w \in sWfs(s,f) \iff wStem(w,f) = s\,.$$

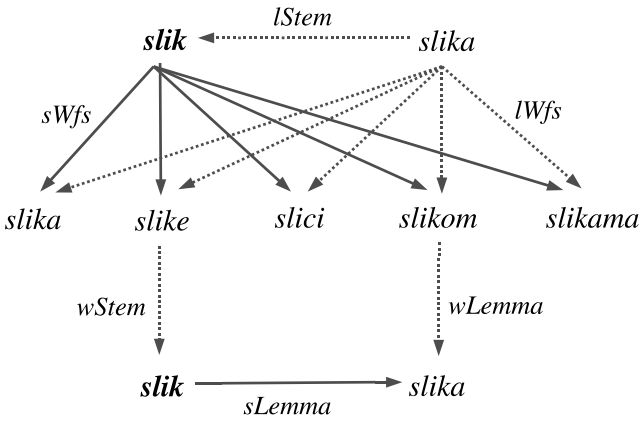
U funkcijskom smislu najopćenitji – a u praktičnom smislu najiskoristiviji – jest model koji podržava i generativni i redukcijski smjer izvođenja. Takav generativno-redukcijski model može se koristiti kako za generiranje oblika tako i za svođenje bilo kojeg oblika na obličnu osnovu. Također, budući da je kod takvog modela iz leme moguće izvesti osnovu (primjenom funkcije *lStem*), to se u ishodište fleksije umjesto oblične osnove može smjestiti lema, što je u praksi mnogo prikladnije (v. odjeljak 3.2.5). Ako se u ishodište fleksije želi smjestiti lema, model treba proširiti sljedećim funkcijama:

$$lWfs : \mathcal{S} \times \mathcal{F} \to \wp(\mathcal{S})\,, \tag{3.8}$$
$$lWfsMsd : \mathcal{S} \times \mathcal{F} \to \wp(\mathcal{S} \times \mathcal{X})\,, \tag{3.9}$$
$$wLemma : \mathcal{S} \times \mathcal{F} \to \mathcal{S}\,, \tag{3.10}$$
$$wLemmaMsd : \mathcal{S} \times \mathcal{F} \to \wp(\mathcal{S} \times \mathcal{X})\,. \tag{3.11}$$

definiranima na sljedeći način:

$$lWfsMsd(l,f) = sWfsMsd(lStem(l,f),f)\,, \tag{3.12}$$
$$wLemmaMsd = sStemMsd(wStem(l,f),f)\,. \tag{3.13}$$

Funkcija *lWfsMsd* analogna je funkciji *sWfsMsd*, s tom razlikom da za ishodište fleksije uzima lemu. Slično, funkcija *wLemmaMsd* analogna je funkciji *wStemMsd*, ali oblik umjesto na osnovu svodi na lemu. Funkcije *lWfs* i *wLemma* definiraju se temeljem gornjih dviju funkcija odbacivanjem morfosintaktičkih opisa, po uzoru na (3.4). Na



Slika 3.1: Primjer uporabe funkcija morfološkog modela za generiranje i redukciju oblika riječi *slika*. Punom crtom prikazano je izvođenje pomoću generativnog dijela modela, a isprekidanom crtom izvođenje pomoću redukcijskog dijela modela. Izvođenje oblika s lemom kao ishodišnim oblikom iziskuje (polu)redukcijski model, odnosno primjenu funkcije *lStem*.

slici 3.1 skicirani su odnosi između funkcija apstraktnoga funkcijskog modela.

Napomene

Kako model u konačnici treba biti računalno ostvarljiv, generativno-redukcijski model zaslužuje nekoliko dodatnih napomena. U računalnom smislu, funkcijama (3.1)–(3.3) definirano je izračunavanje od oblične osnove k oblicima, dok je funkcijama (3.5)–(3.7) definiran obrnut smjer izračunavanja. Pretpostavimo li generativni smjer izračunavanja kao podrazumijevani, generativno-redukcijski računalni model mora također podržati inverzni smjer izračunavanja. U praksi to znači da flektivni uzorak treba biti iskoristiv u oba smjera, iz čega onda proizlaze i određena ograničenja za način računalnog prikaza flektivnog uzorka. Primjerice, uzorak ne može biti izveden izravno kao funkcija ili procedura računalnog programa, budući da u tom slučaju ne bi bio reverzibilan. Primjer za to je model funkcijske morfologije (Forsberg & Ranta, 2003), koji je generativan, ali nije redukcijski. Računalni model koji podržava izračunavanje funkcije *lStem* (a onda i funkcija *lWfsMsd* i *lWfs*), ali ne i izračunavanje ostalih funkcija redukcijskog smjera, možemo u tom smislu smatrati *poluredukcijskim*. Takav se model može koristiti za generiranje oblika leme, ali ne i za obrnuti smjer svođenja oblika na lemu.

Druga važna napomena tiče se višeznačnosti gramatike. Pravilima gramatike, odnosno flektivnim i tvorbenim uzorcima, inherentan je određeni stupanj generalizacije. To znači da u praksi postoje višeznačnosti koje se očituju kao izvođenje nepravilnih

(u jeziku nepotvrđenih) oblika. Višeznačnosti su u načelu moguće i u generativnom i u redukcijskom smjeru, premda su u redukcijskome smjeru više izražene.[2] Postojanje višeznačnosti modelom se može obuhvatiti na način da se kodomena funkcija definiranih izrazima (3.3), (3.5), (3.6) i (3.10) sa skupova proširi na odgovarajuće partitivne skupove.

3.1.2 Derivacijska sastavnica modela

Derivacijskom odnosno tvorbenom morfologijom određeno je kako se iz jedne riječi, tzv. polazne ili *osnovne riječi*, tvori druga riječ, tzv. *tvorenica*. Ako je tvorenica izvedena iz jedne riječi, nazivamo je *izvedenica*, a takav naziv tvorbe *izvođenje* (Barić *et al.*, 2005). Tradicionalne gramatike tvorbeni sustav jezika opisuju pomoću *tvorbenih uzoraka*. Tvorbeni uzorak povezuje osnovnu riječ i tvorenicu te određuju značenjsku kategoriju tvorenice.

U apstraktnome funkcijskom modelu tvorbeni uzorci apstrahirani su skupom $\mathcal{D}$, a značenjske kategorije skupom $\mathcal{Y}$. Izvođenje riječi opisano je funkcijom:

$$lDerive \quad : \quad \mathcal{S} \times \mathcal{F} \times \mathcal{D} \rightarrow \wp(\mathcal{S} \times \mathcal{Y})\,. \tag{3.14}$$

Funkcija *lDerive* temeljem tvorbenog uzorka osnovnoj riječi pridružuje izvedenicu (ili više njih) i njoj pripadajuću značenjsku kategoriju. Ovdje je za ishodišni oblik izvođenja uzeta lema, ali to bi jednako tako mogla biti i oblična osnova. Izvođenje se provodi u ovisnosti o morfološkoj vrsti osnovne riječi koja je određena povezanošću te riječi s flektivnim uzorkom iz $\mathcal{F}$.

3.1.3 Lematizacija

Osnovna funkcionalnost redukcijskog flektivnog modela – a pogotovo u kontekstu morfološke normalizacije – jest lematizacija, odnosno svođenje proizvoljnog oblika neke riječi na njezin kanonski oblik. Taj je postupak donekle obuhvaćen funkcijom *wLemma*, no tamo je također potrebno navesti flektivni uzorak kojoj ta riječ pripada, što u praksi nije moguće.[3] Lematizacijski postupak možemo opisati funkcijom $lm_{\mathcal{F}} : \mathcal{S} \rightarrow \wp(\mathcal{S} \times \mathcal{F})$ definiranom na temelju skupa flektivnih uzoraka $\mathcal{F}$ kako slijedi:

[2] Razlog tome mogao bi biti taj što su opisi kakve nalazimo u tradicionalnim gramatikama uglavnom usredotočeni na generativni smjer.

[3] U ovome radu lematizacijom nazivamo isključivo postupak svođenja oblika riječi na lemu kada flektivni uzorak nije predodređen.

$$lm_{\mathcal{F}}(w) = \left\{(l, f) \ : \ f \in \mathcal{F},\, l \in wLemma(w, f)\right\}. \tag{3.15}$$

Postupak lematizacije općenito je višeznačan, pa funkcija *lm* u općenitom slučaju rezultira skupom lema. Dva su moguća uzroka toj višeznačnosti: prvi je homografnost oblika koji se lematizira, a drugi je u višeznačnosti pravila gramatike.

3.2 Model temeljen na funkcijama višega reda

Model morfologije temeljen na funkcijama višeg reda (engl. *Higher Order Functional Morphology*, HOFM) razrada je gore opisanog apstraktnog modela. Model je inspiriran osnovnim konceptima funkcijske programske paradigme (Hudak, 1989). Tako osnovni gradivni element modela predstavlja funkcija kao apstrakcija flektivnog i tvorbenog pravila, odnosno preoblike iz jednog oblika u drugi. Te se preoblike u modelu opisuju pomoću funkcija višeg reda, a za njihovo kombiniranje koristi se funkcijska kompozicija. Osnovna ideja takvog pristupa jest morfološki opis učiniti što bliskiji onome kakvog nalazimo u tradicionalnim gramatikama, te na taj način olakšati i ubrzati modeliranje. S druge strane, bliskost modela s funkcijskom programskom paradigmom znatno pojednostavljuje računalnu izvedbu modela.

Model HOFM je generativno-redukcijski, omogućava, dakle, kako generiranje oblika riječi tako i svođenje tih oblika na ishodišni oblik, što je u kontekstu morfološke normalizacije osobito bitno. Nadalje, model je jezično neovisan i zasigurno je primjenjiv na jezike koji su po morfološkoj tipologiji slični hrvatskome. Primjeri navedeni u ovome poglavlju odnose se na morfologiju hrvatskog jezika, no konkretna se izvedba modela morfologije za hrvatski jezik razmatra tek u idućem poglavlju.

Modelom HOFM obuhvaćena je flektivna i derivacijska morfologija koja je formalizirana skupom flektivnih i derivacijskih uzoraka. U nastavku je najprije opisan način prikaza flektivnih i tvorbenih pravila pomoću funkcije preoblike znakovnoga niza, a zatim je opisano kako se te preoblike mogu definirati posredno putem različitih funkcija višeg reda.

3.2.1 Preoblike

Neka je $\mathcal{S}$ skup svih znakovnih nizova koji se mogu generirati abecedom Σ, tj. $\mathcal{S} = \Sigma^*$. Neka ε označava prazan znakovni niz, $\varepsilon \in \mathcal{S}$. Općenita transformacija jednog oblika riječi u drugi oblik predstavljena je u formalizmu HOFM *funkcijom preoblike znakovnoga*

niza (kraće: *preoblikom*):[4]

$$t : \mathcal{S} \to \mathcal{S}\,,$$

pri čemu:

(i) $t(\varepsilon) = \varepsilon\,,$

(ii) $\forall s_1, s_2 \in \mathcal{S}.\ (t(s_1) = t(s_2) \neq \varepsilon) \ \Rightarrow\ (s_1 = s_2)\,.$

To jest, funkcija t je injektivno preslikavanje sa skupa znakovnih nizova na skup znakovnih nizova, izuzev za one znakovne nizove koji se preslikavaju u prazan znakovni niz ε, uključivo i sâm niz ε. Prazan znakovni niz ε koristi se kao indikacija da preoblika nije primjenjiva na zadani znakovni niz, i u tom slučaju kažemo da je preoblika t na znakovnim nizu s *zakazala*. Na primjer, preoblika će zakazati ako iz znakovnoga niza pokuša odbaciti nastavak koji u tom nizu ne postoji. Svrha uvjeta (ii) jest osigurati da sve one preoblike koje ne zakazuju imaju svoj inverz. Neka $\mathcal{T}$ označava skup svih preoblika znakovnih nizova. Uvodimo i dvije posebne preoblike: preobliku $nul \in \mathcal{T}$ definiranu kao

$$\forall s \in \mathcal{S}.\ nul(s) = s\,,$$

koju nazivamo *nul-preoblikom*, te preobliku $fail \in \mathcal{T}$ definiranu kao

$$\forall s \in \mathcal{S}.\ fail(s) = \varepsilon\,.$$

U modelu HOFM preoblika znakovnoga niza apstrahirana je funkcijom t koja može biti bilo kakva funkcija koja zadovoljava gornja ograničenja. Na ovoj razini modela ne postoje nikakva ograničenja za ekspresivnost te funkcije. U praksi je takva općenitost nepotrebna i nepraktična. Naime, u većini slučajeva preoblike je moguće izraziti regulanim relacijama između znakovnih nizova. Zbog toga će ovdje skicirani dijelovi modela u odjeljku 3.3 biti dodatno razrađeni.

3.2.2 Funkcije višeg reda

U praksi je broj različitih preoblika razmjerno ograničen. Za konkatenativne morfologije, kakva je morfologija hrvatskog jezika, tipično je riječ o preoblikama kojima se

[4] Izraz "preoblika" u radu se koristi u smislu *transformacije* jednog znakovnoga niza u drugi, a ne u smislu pravila kojima se provodi semantička analiza tvorbe, za što se ovaj izraz također koristi (usp. Barić *et al.* (2005)).

Tablica 3.1: Primjer fleksije imenice *vojnik*. Lema (oblik u nominativu jednine) je podcrtana, dok su fonološke alternacije oblične osnove prikazane masnim slovima

Padež	Jednina	Množina
N.	*vojnik-*$\emptyset$	*vojni***c***-i*
G.	*vojnik-a*	*vojnik-a*
D.	*vojnik-u*	*vojni***c***-ima*
A.	*vojnik-a*	*vojnik-e*
V.	*vojni***č***-e*	*vojni***c***-i*
L.	*vojnik-u*	*vojni***c***-ima*
I.	*vojnik-om*	*vojni***c***-ima*

ostvaruje sufiksacija (dodavanje obličnih nastavaka ili sufikasa), prefiksacija (dodavanje prefiksa), različite fonološke alternacije ili pak različite kombinacije ovih preoblika. Preoblike je prema tome moguće razvrstati u određene tipove, koje je u okviru funkcijskog modela zatim moguće definirati posredno pomoću *funkcija višeg reda*. Pojam funkcije višeg reda (engl. *higher-order function*) koristi se u funkcijskome programiranju za funkciju u čijoj su domeni ili kodomeni funkcije, tj. funkciju koja preslikava sa skupa ili na skup funkcija (Hudak, 1989). Primjerice, ako je $f : X \to G$ funkcija višeg reda koja elemente skupa X preslikava na funkcije iz skupa G, tada je $f(x)(y)$ rezultat primjene (aplikacije) funkcije $f(x) = g \in G$ na element y, tj., $f(x)(y) = g(y)$. U okviru modela HOFM, te će funkcije općenito biti oblika $f : X \to \mathcal{T}$, odnosno one će elementima proizvoljne domene X, koji ovdje imaju ulogu parametara preoblike, pridjeljivati konkretne preoblike iz $\mathcal{T}$.

Tipičan primjer uporabe funkcije višeg reda za definiranje preoblike jest funkcija $sfx : \mathcal{S} \to \mathcal{T}$, definirana na način da $sfx(s)$ daje znakovnu preobliku sufiksacije znakovnoga niza s nekom drugom znakovnom nizu. Odnosno:[5]

$$sfx(s_1) = \lambda s.(ss_1)\,.$$

Pomoću ove funkcije preoblika kojom se, primjerice, obličnoj osnovi dodaje nastavak *-a* može se jednostavno definirati kao $sfx(a)$, pri čemu vrijednost a djeluje kao parametar preoblike.[6]

Uporabu preoblika znakovnih nizova ilustrirat ćemo na primjeru imenice *vojnik*. Oblici ove imenice tvore se prema flektivnom uzorku prikazanom slikom 3.1. U ovom

[5] Ovdje, kao i u nastavku gdje god je to prikladno, korišteni su notacijski elementi lambda-računa.

[6] U gramatici hrvatskoga jezika sufiksima se nazivaju isključivo derivacijski odnosno tvorbeni nastavci, dok se flektivni nastavci nazivaju obličnim nastavcima (Barić *et al.*, 2005). S funkcijskog stajališta, u oba je slučaja riječ o preoblici sufiksacije, stoga ova dva slučaja u modelu nećemo razlikovati.

je slučaju lema (oblik nominativa jednine) identična obličnoj osnovi riječi, dok se drugi oblici dobivaju dodavanjem obličnih nastavaka. Pored toga, u vokativu i svim padežima množine osim genitiva i akuzativa provodi se fonološka alternacija sibilarizacije (općenito: zamjena velara k, g, i h sa c, z, odnosno s), dok se u vokativu jednine provodi palatalizacija (općenito: zamjena velara k, g i h sa $č$, $ž$ odnosno $š$). Sufiksaciju, kao što je već objašnjeno, u modelu HOFM možemo definirati funkcijom višeg reda $sfx : \mathcal{S} \to \mathcal{T}$. Tako se primjerice genitivni oblik imenice *vojnik* može se dobiti kao $sfx(a)(vojnik) = vojnika$. Za preoblike kojima se ostvaruju fonološke alternacije na kraju obliče osnove definirat ćemo funciju višeg reda $asfx : \wp(\mathcal{S} \times \mathcal{S}) \to \mathcal{T}$. Fonološka alternacija definirana je kao skup parova znakovnih nizova, $\{(s_i, s_j)\}$, svaki od kojih definira završetak obliče osnove i njegovu zamjenu.[7] Na taj način, preobliku sibilarizacije definiramo kao

$$sbl = asfx\Big(\{(k, c), (g, z), (h, s)\}\Big)\,,$$

dok preobliku palatalizacije definiramo kao

$$plt = asfx\Big(\{(k, č), (g, ž), (h, š)\}\Big)\,. \tag{3.16}$$

Sibilarizacija odnosno palatalizacija obliče osnove ostvaruje se zatim primjenom odgovarajuće funkcije preoblike, primjerice $sbl(vojnik) = vojnic$ odnosno $plt(vojnik) = vojnič$. Naposlijetku, kombiniranje fonološke alternacije s dodavanjem nastavka ostvaruje se *funkcijskom kompozicijom*. Primjer je preoblika kojom se iz obliče osnove *vojnik* dobiva vokativ jednine kao kompozicija dodavanja nastavka i palatalizacije:

$$(sfx(e) \circ plt)(vojnik) = sfx(e)(vojnič) = vojniče\,.$$

3.2.3 Uvjet primjenjivosti

Pored flektivnih uzoraka, tradicionalne gramatike uobičajeno propisuju uvjete primjenjivosti uzoraka. Tako se, na primjer, uvjetima može ograničiti primjena uzorka na samo one obliče osnove koje završavaju određenim nastavcima. Takovi uvjeti primjenjivosti u modelu HOFM prikazuju se *uvjetnim funkcijama*, koje su oblika

$$c : \mathcal{S} \to \{\top, \bot\}\,,$$

[7] Ovdje podrazumijevamo da su zamjene međusobno isključive, pa da su odgovarajuće preoblike funkcijske i injektivne. O ovome će više govora biti u odjeljku 3.3.2.

i koje znakovne nizove preslikavaju u elemente dvočlanog skupa $\{\top, \bot\}$, gdje '$\top$' označava istinu a '$\bot$' označava laž. Ako je uzorak bezuvjetno primjenjiv, koristimo tautološku funkciju uvjetovanja *always* definiranu tako da vrijedi $\forall s \in \mathcal{S}.\, always(s) = \top$. Sa $\mathcal{C}$ ćemo označavati skup svih funkcija uvjetovanja.

Kao i kod funkcija preoblike, kod uvjetnih funkcija koristit ćemo funkcije višeg reda kako bismo što jednostavnije izrazili različite uvjete primjenjivosti. Ideju možemo ponovno ilustrirati na primjeru flektivnog uzorka imenice *vojnik* sa slike 3.1. Gramatika propisuje da je ovaj uzorak primjenjiv na oblične osnove koje završavaju velarima (*k*, *g* i *h*), osim ukoliko oni oblikuju suglasnički skup od dva slijedna suglasnika. Kako bismo provjerili završava li oblična osnova određenim nastavkom, definirat ćemo funkciju višeg reda $ends : \wp(\mathcal{S}) \to \mathcal{C}$. Ova funkcija skup nastavaka preslikava u funkciju uvjetovanja koja ispituje završava li oblična osnova bilo kojim od navedenih nastavaka. Tako, primjerice, $ends(\{k, g, h\})(vojnik) = \top$, dok $ends(\{k, g, h\})(vojniče) = \bot$. Pomoću funkcije *ends* može se zatim definirati funkcija uvjetovanja *cgr*, koja ispituje završava li oblična osnova suglasničkim skupom:

$$cgr = ends(\{x_1x_2 \mid (x_1, x_2) \in cons \times cons\}),$$

gdje *cons* označava skup suglasnika. Konačno, ove dvije funkcije možemo udružiti u jednu funkciju na sljedeći način:

$$c = ends(\{k, g, h\}) \wedge \neg cgr\,,$$

pri čemu logičke operatore '$\wedge$', '$\vee$' i '$\neg$' prirodno proširujemo na funkcije, kako slijedi:

$$c_1 \wedge c_2 = \lambda s.(c_1(s) \wedge c_2(s)), \quad c_1 \vee c_2 = \lambda s.(c_1(s) \vee c_2(s)), \quad \neg c = \lambda s.(\neg c(s)). \quad (3.17)$$

Za funkciju *c* vrijedi slična napomena kao i za funkciju preoblike: u modelu HOFM uvjet primjenjivosti uzorka apstrahiran je funkcijom *c* na koju nisu postavljena nikakva ograničenja. U praksi je takva razina općenitosti nepotrebna budući da su uvjeti u većini slučajeva izrazivi regularnim jezicima.

3.2.4 Flektivni uzorak

U modelu HOFM, jednako kao i tradicionalnim gramatikama, sustav fleksije opisan je flektivnim uzorcima. Flektivnim su uzorkom različiti oblici riječi, uključivo osnovni

oblik odnosno lema, definirani kao preoblike obliične osnove. Pored toga, flektivni uzorak definira uvjet koji oblična osnova mora zadovoljavati kako bi uzorak na nju bio primjenjiv. Formalno, flektivni uzorak f definiramo kao uređenu dvojku:

$$f = (c, \{t_0, \ldots, t_n\}) \ \in \mathcal{C} \times \wp(\mathcal{T})\,. \tag{3.18}$$

Uvjetna funkcija c određuje uvjet koji oblična osnova s mora zadovoljiti kako bi uzorak f bio primjenjiv, dok skup $\{t_0, \ldots, t_n\}$ definira preoblike znakovnoga niza kojima se iz oblične osnove s izvodi ukupno n različitih oblika, $t_0(s), t_1(s), \ldots, t_n(s)$. Dogovorno, funkcija t_0 preobličava obličnu osnovu s u lemu l, odnosno, $l = t_0(s)$. Neka $\mathcal{F}$ označava skup svih flektivnih uzoraka definiranih modelom.

U izrazu (3.18) funkcija t_0 odgovara funkciji *sLemma* apstraktnog funkcijskog modela, pa vrijedi:

$$\begin{aligned} sLemma\Big(s, \big(c, \{t_0, \ldots, t_n\}\big)\Big) &= t_0(s)\,, \\ lStem\Big(l, \big(c, \{t_0, \ldots, t_n\}\big)\Big) &= t_0^{-1}(l)\,. \end{aligned}$$

Primjer 1 – imenička paradigma

Kao primjer razmotrimo ponovo sklanjanje imenice *vojnik* sa slike 3.1. Odgovarajući flektivni uzorak f_{N1} definiramo na sljedeći način:[8]

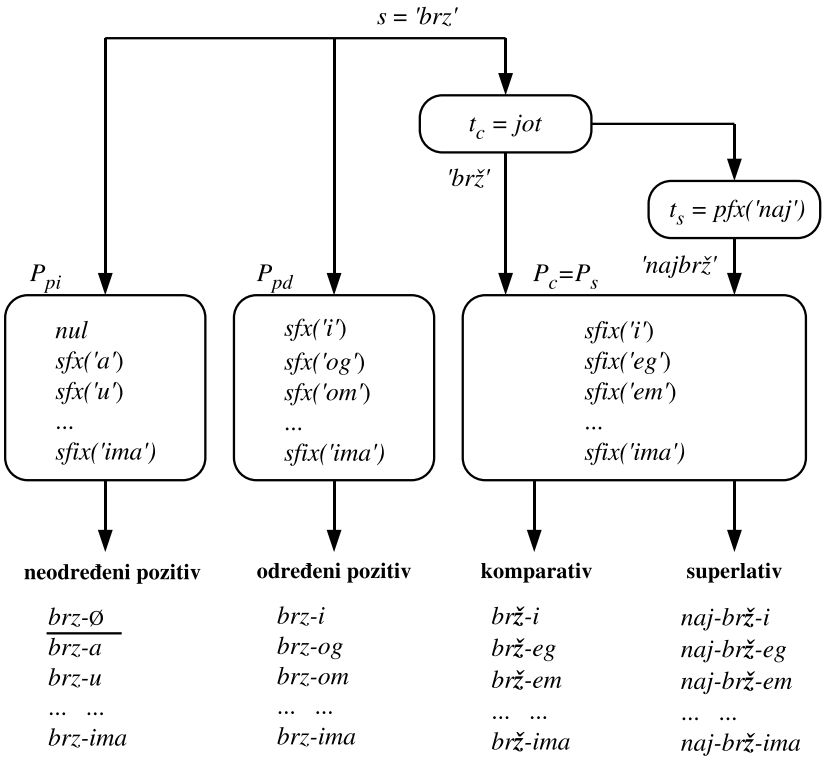
$$\begin{aligned} f_{N1} \ = \ \Big(\lambda s.\big(ends(\{k, g, h\}) \wedge \neg cgr\big), \Big\{ & nul, sfx(a), sfx(u), sfx(om), \\ & sfx(e) \circ plt, sfx(i) \circ sbl, sfx(ima) \circ sbl, sfx(e) \Big\}\Big)\,. \end{aligned} \tag{3.19}$$

Budući da je u ovome slučaju lema identična obličnoj osnovi, odnosno $l = s$, to definiramo $t_0 = nul$.

Primjer 2 – pridjevska paradigma

Složeniji primjer flektivnog uzorka ilustrirat ćemo na primjeru pridjevske paradigme. Pridjevska je paradigma u hrvatskom jeziku složenija od imeničke ili glagolske budući da se, pored gramatičkih kategorija padeža i broja, u njoj iskazuju kategorije roda i stupnja. Pritom svaki od tri stupnja koristi izmijenjenu osnovu: drugi stupanj odnosno komparativ tipično koristi sufiksiranu ili fonološki alterniranu osnovu prvog stupnja

[8] U nastavku rada nazive imeničkih flektivnih uzoraka potpisivat ćemo sa $N1$, $N2$, itd., nazive pridjevskih uzoraka sa $A1$, $A2$, itd., a nazive glagolskih uzoraka sa $V1$, $V2$, itd.



Slika 3.2: Flektivni uzorak za pridjev *brz*, razložen na tri različita poduzorka. Poduzorak za drugi (komparativ) i treći (superlativ) stupanj su identični, ali se primjenjuju na drugačiju preobliku oblične osnove. Lema (neodređeni oblik jednine muškog roda) je podcrtana; fonološke alternacije u obličnoj osnovi prikazane su masnim slovima.

odnosno pozitiva, dok treći stupanj odnosno superlativ na tako izmijenjenu osnovu prefiksira *naj-*. Primjerice, kod fleksije pridjeva *brz* u prvome se stupnju koristi osnova *brz*, u drugome se stupnju koristi jotirana osnova *brž*, dok se u trećemu stupnju koristi osnova *najbrž*. Pored ovoga, pridjev *brz*, kao i većina drugih pridjeva, ima zasebne uzorke za određeni i neodređeni vid.

U modeliranju pridjevske paradigme, u većini je slučajeva uputno flektivni uzorak definirati pomoću četiri zasebna poduzoraka, svakom od kojih odgovara zaseban skup preoblika znakovnoga niza. Ta četiri poduzorka – svaki zasebno – definiraju: neodređene oblike prvoga stupnja, određene oblike prvoga stupnja, oblike drugoga stupnja te oblike trećega stupnja. Pritom su preoblike kod prva dva poduzorka definirane u odnosu na obličnu osnovu s, dok su preoblike kod druga dva poduzorka (drugi i treći stupanj) definirane u odnosu na preobličenu obličnu osnovu.

Ilustrirajmo navedeno upravo na primjeru fleksije pridjeva *brz*. Struktura flektivnog uzorka, razloženog na četiri (odnosno tri različita) poduzorka, prikazana je slikom 3.2.

Flektivni je uzorak definiran na sljedeći način:

$$f_{A1} = \Big(c,\, P_{pi} \cup P_{pd} \cup \{t_i \circ t_c :\ t_i \in P_c\} \cup \{t_i \circ t_c \circ t_s :\ t_i \in P_s\}\Big)\,. \tag{3.20}$$

Funkcija uvjetovanja c definirana je kao:

$$c_{A1} = ends(nonpalatal \setminus \{r\}) \wedge \Big(\neg cgr \vee ends(\{st, št\})\Big)\,.$$

Prema tome, sukladno pravilima gramatike, ovaj će uzorak biti primjenjiv na oblične osnove koje završavaju nepalatalnim suglasnicima, izuzev sonanta r, pod uvjetom da suglasnici ne oblikuju suglasnički skup, izuzev skupova *st* i *št*. Skup preoblika P_{pi} odgovara poduzorku za neodređene oblike pozitiva, skup P_{pd} odgovara poduzorku za određene oblike istoga stupnja, dok skupovi P_c, i P_s odgovaraju poduzorcima za komparativ odnosno superlativ. Ovi su poduzorci definirani na sljedeći način (ovdje skraćeno):

$$\begin{aligned} P_{pi} &= \{nul, sfx(a), sfx(u), \ldots, sfx(ima)\}\,, \\ P_{pd} &= \{sfx(i), sfx(og), sfx(om), \ldots, sfx(ima)\}\,, \\ P_c = P_s &= \{sfx(i), sfx(eg), sfx(em), \ldots, sfx(ima)\}\,. \end{aligned} \tag{3.21}$$

Budući da je i ovdje, kao i kod imenice *vojnik*, lema identična oblič̃noj osnovi, to ponovno vrijedi $t_0 = nul$. Funkcijama t_c i t_s definirane su preoblike oblične osnove u osnovu za drugi odnosno treći stupanj. Preoblike su definirane kako slijedi:

$$\begin{aligned} t_c &= asfx(jot)\,, \\ t_s &= pfix(naj)\,, \end{aligned} \tag{3.22}$$

pri čemu funkcija preoblike *jot* definira fonološku alternaciju jotacije. Konačno, uvrštavanjem desnih strana izraza (3.21) i (3.22) u izraz (3.20) dobivamo:

$$\begin{aligned} f_{A1} = (c_{A1},\ & \{nul, sfx(a), sfx(u), \ldots, sfx(ima)\} \cup \\ & \{sfx(i), sfx(og), sfx(om), \ldots, sfx(ima)\} \cup \\ & \{sfx(i) \circ asfx(jot), sfx(eg) \circ asfx(jot), \ldots, sfx(ima) \circ asfx(jot)\} \cup \\ & \{sfx(i) \circ asfx(jot) \circ pfix(naj), \ldots, sfx(ima) \circ asfx(jot) \circ pfix(naj)\}\,, \end{aligned} \tag{3.23}$$

tj. oblik flektivnog uzorka u kojem su pojedini poduzorci stopljeni u jedan skup preoblika.

3.2.5 Primjenjivost flektivnog uzorka

Flektivni uzorak u vezu dovodi obličnu osnovu riječi sa svim njezinim oblicima. Svi oblici neke riječi u potpunosti su određeni obličnom osnovom riječi i flektivnim uzorkom, pod uvjetom da je uzorak na tu obličnu osnovu primjenjiv. Ako bismo međutim, umjesto oblične osnove, u ishodište fleksije smjestili lemu, mogli bismo reći da flektivni uzorak povezuje lemu sa svim ostalim oblicima riječi. Ova su dva pogleda u načelu ravnopravna, no budući da je lema lingvistički valjan oblik riječi, dok oblična osnova riječi to najčešće nije, zbog čitljivosti jednostavnije je u ishodište fleksije smjestiti upravo lemu. To je opravdano posebice u kontekstu morfološke normalizacije, gdje je za morfološku normu bolje odabrati lemu nego obličnu osnovu, budući da je kod potonje višeznačnost izraženija u smislu da različite leme mogu imati identičnu obličnu osnovu (npr. imenice *rob* i *roba*). Iz ovih razloga u modelu HOFM u ishodište fleksije (a kasnije i tvorbe) smještena je lema a ne oblična osnova, pa tako u nastavku govorimo o vezi između leme i (ostalih) flektivnih oblika.

U tom smislu, u nastavku definiramo primjenjivost flektivnog uzorka na lemu. Intuitivno, flektivni će uzorak biti primjenjiv na lemu ako je uvjetna funkcija uzorka zadovoljena na odgovarajućoj obličnoj osnovi (dobivenoj iz leme) te ako niti jedna od preoblika znakovnoga niza definirana flektivnim uzorkom na dotičnoj obličnoj osnovi ne zakazuje. Primjenjivost flektivnog uzorka f na lemu l označavat ćemo sa '$f \vDash l$'. Formalno, *primjenjivost flektivnog uzorka* definirana je kao:

$$(c, \{t_0, \ldots, t_i, \ldots, t_n\}) \vDash l \quad \Longleftrightarrow \quad c(s) \wedge \left(\forall t_i.\ (t_i \circ t_0^{-1})(l) \neq \varepsilon\right). \tag{3.24}$$

Oblična osnova se iz leme izvodi primjenom inverza preoblike t_0. Pritom, budući da vrijedi $t(\varepsilon) = \varepsilon$, to kompozicija preoblika $t_0 \circ t_0^{-1}$ zakazuje čim zakaže jedna od ovih dviju preoblika.

Uvjet da niti jedna od preoblika definiranih flektivnim uzorkom ne smije zakazati razumno je postroženje uvjeta definiranog uvjetnom funkcijom c. Flektivni uzorak naime definira sve oblike neke leme, pa ako ijedna preoblika znakovnoga niza zakaže, onda taj uzorak zasigurno nije primjenjiv na dotičnu lemu. To u praksi znači da funkciju uvjetovanja nije potrebno definirati u onim slučajevima kada je njome iskazan uvjet već obuhvaćen implicitnim uvjetom nezakazivanja neke od preoblika. Primjer je flektivni uzorak f_{N1} definiran izrazom (3.19). Budući da ovaj uzorak koristi preobliku *sbl* za sibilarizaciju oblične osnove, i da će ta preoblika zakazati ukoliko oblična osnova ne

završava suglasnicima *k*, *g* ili *h*, nije bilo potrebno taj uvjet izričito ponavljati u uvjetu uzorka. Naravno, to ne znači da je svaki uvjet moguće iskazati implicitno, niti da je to potrebno.

3.2.6 Generiranje oblika riječi

Nakon što smo definirali flektivni uzorak i primjenjivost uzorka, možemo napokon opisati na koji se način uzorak može upotrijebiti za izvođenje svih oblika neke riječi, krenuvši od leme kao ishodišnog oblika. Generiranje oblika u apstraktnome funkcijskom modelu opisano je funkcijom $lWfs : \mathcal{S} \times \mathcal{F} \to \wp(\mathcal{S})$. Konkretna definicija ove funkcije u modelu HOFM je sljedeća:

$$lWfs\Big(l, \underbrace{(c, \{t_0, \ldots, t_i, \ldots, t_n\})}_{f}\Big) = \begin{cases} \bigcup_i \{(t_i \circ t_0^{-1})(l)\} & \text{ako } f \vDash l, \\ \emptyset & \text{inače.} \end{cases} \tag{3.25}$$

gdje je '$\vDash$' relacija primjenjivosti flektivnog uzorka definirana izrazom (3.24). U proceduralnom smislu svaki se oblik izvodi tako da se na lemu l najprije primijeni inverz funkcije preoblike t_0 kako bi se dobila oblična osnova, a zatim se primjenjuje svaka od preoblika t_i kako bi se iz oblične osnove izveli različiti oblici riječi. Ako uzorak f na lemu l nije primjenjiv, $lWfs(f, l)$ rezultira praznim skupom.

3.2.7 Svođenje oblika na lemu

Obrnuti postupak, kojim se na temelju flektivnog uzorka izvodi lema (odnosno skup mogućih lema) danog oblika riječi, u apstraktnom je modelu opisan funkcijom $wLemma : \mathcal{S} \times \mathcal{F} \to \wp(\mathcal{S})$.[9] U modelu HOFM ta je funkcija definirana na sljedeći način:

$$wLemma\Big(w, \underbrace{(c, \{t_0, \ldots, t_i, \ldots, t_n\})}_{f}\Big) = \Big\{(t_0 \circ t_i^{-1})(w) \;:\; f \vDash (t_0 \circ t_i^{-1})(w)\Big\}. \tag{3.26}$$

Funkcijom se oblik najprije pokušava preobličiti u obličnu osnovu, i to primjenom svih preoblika definiranih flektivnim uzorkom, a zatim se svaka tako dobivena oblična osnova pokušava preobličiti u lemu. Od dobivenih se lema zadržavaju samo one na koje je flektivni uzorak primjenjiv sukladno (3.24).

[9] U skladu s napomenom iz odjeljka 3.1.1, ovaj postupak nazivamo "svođenje na lemu", dok lematizacijom nazivamo postupak opisan funkcijom (3.15).

Ovdje je važno napomenuti da se u izrazu (3.26) pojavljuje inverz funkcije preoblike t_i^{-1}. To u bitnome određuje način programske izvedbe funkcije preoblike znakovnog niza, koja dakle mora biti prikazana tako da je izračunljiv njezin inverz.

Flektivni uzorci općenito mogu biti višeznačni, pa funkcija *wLemma* općenito rezultira skupom mogućih lema. Ilustrirajmo to na primjeru oblika *bržeg* (genitiv i akuzativ jednine muškog roda u komparativu te genitiv jedine srednjeg roda u komparativu pridjeva *brz*) i flektivnog uzorka f_{A1} danog izrazom (3.23). Vrijedi:

$$wLemma(bržeg, f_{A1}) = \{brz, bržeg\}\,. \tag{3.27}$$

U ovom slučaju, pored ispravne leme *brz*, dobivamo i neispravnu lemu *bržeg*. Problem je u tome što se, sukladno uzorku f_{A1}, oblik *bržeg* može smatrati kako oblikom tako i samom lemom. Naime, budući da vrijedi $t_0 = nul$, to vrijedi

$$(t_1 \circ t_0^{-1})(bržeg) = (nul \circ nul^{-1})(bržeg) = nul(bržeg) = bržeg\,,$$

dok ujedno vrijedi i $f_{A1} \vDash bržeg$. Navedeni primjer ukazuje na to da je lematizacija višeznačna čak i onda kada je poznato koji flektivni uzorak treba primijeniti na zadani oblik.

3.2.8 Homografija

Homografi ili istopisnice jesu oblici istoga izraza, ali različitog značenja. Odnosno, to su oblici koji se identično pišu, no pripadaju (barem) dvama različitim leksemima. U okviru modela HOFM, homograf će, posredstvom različitih flektivnih uzoraka, biti povezan s dvije ili više lema. Formalno, oblik $w \in \mathcal{S}$ je homograf ako vrijedi:

$$\exists l_1, l_2 \in \mathcal{S},\, \exists f_1, f_2 \in \mathcal{F}.\quad (f_1 \neq f_2) \wedge (w \in lWfs(l_1, f_1)) \wedge (w \in lWfs(l_2, f_2))\,.$$

Valja primijetiti da obrat tvrdnje ne vrijedi: budući da su flektivni uzorci općenito višeznačni, oblik w može biti flektivno povezan s dvije leme, no ipak ne biti pravi homograf.

Relaciju homografije preciznije ćemo definirati nad parovima dvojki sastavljenih od leme i flektivnog uzorka. Par takvih dvojki smatrat ćemo homografnima (odnosno homografnim parom), ako dijele barem jedan identičan oblik. Formalno:

$$hPair\Big((l_1, f_1), (l_2, f_2)\Big) \quad \Longleftrightarrow \quad lWfs(l_1, f_1) \cap lWfs(l_2, f_2) \neq \emptyset\,.$$

3.2.9 Derivacijski uzorci

U uvodnome dijelu ovoga poglavlja rečeno je da tradicionalne gramatike tvorbeni sustav jezika opisuju pomoću tvorbenih uzoraka. Tvorbenim uzorkom određen je postupak tvorbe, vrsta osnovne riječi te značenjska kategorija tvorenice. Značenjska kategorija ovisi o vrsti tvorenice (npr. značenjske kategorije imenice mogu biti vršitelj radnje, etnik, hipokoristik i sl.). Postupak tvorbe iskazan je u odnosnu na tvorbenu osnovu kao ishodišni oblik i lemu izvedenice kao ciljni oblik. U hrvatskome jeziku tvorbena osnova izvedenice identična je obličnoj osnovi polazne riječi. Na primjer, pri tvorbi *banka* → *bankar*, iz tvorbene odnosno oblične osnove *bank* izvodi se sufiksacijom *-ar* lema izvedenice *bankar*. Tvorbena osnova kod glagola može, u ovisnosti o morfološkoj vrsti glagola, biti infinitivna osnova (npr. *ploviti*→*plovidba*) ili prezentska osnova (npr. *kupiti*→*kupac*).[10]

Modelom HOFM obuhvaćena je sufiksalna, prefiksalna i sufiksalno-prefiksalna tvorba, odnosno izvođenje. Jednako kao i u tradicionalnim gramatikama, tvorba je u modelu HOFM opisana skupom derivacijskih uzoraka. Derivacijski uzorak definira preobliku osnovne riječi u izvedenicu, kategoriju osnovne riječi te kategoriju izvedenice, pri čemu kategorije odgovaraju skupu flektivnih uzoraka. Kategorije dakle određuju vrstu riječi te eventualno neke dodatne morfološke kategorije koje proizlaze iz povezanosti riječi s određenim flektivnim uzorkom. Kategorije nisu značenjske budući da značenje riječi nije razlučivo samo temeljem njezine morfološke pripadnosti.

Formalno, *derivacijski uzorak* d je uređena trojka:

$$d = (t, \mathcal{F}_1, \mathcal{F}_2) \;\in \mathcal{T} \times \wp(\mathcal{F}) \times \wp(\mathcal{F})\,, \tag{3.28}$$

gdje funkcija t definira preobliku oblične osnove u lemu izvedenice, dok skupovi flektivnih uzoraka $\mathcal{F}_1$ i $\mathcal{F}_2$, pri čemu $\mathcal{F}_1 \subseteq \mathcal{F}$ i $\mathcal{F}_2 \subseteq \mathcal{F}$, definiraju kategorije osnovne riječi odnosno izvedenice. Sa $\mathcal{D}$ ćemo označiti skup svih derivacijskih uzoraka definiranih modelom.

Razmotrimo kao primjer uzorak koji opisuje tvorbu apstraktnih imenica ženskoga roda motiviranih pridjevom dodavanjem sufiksa *-ina*, kao npr. *brz* → *brzina*. Ako pretpostavimo da skupovi flektivnih uzoraka $\{f_{A1}, f_{A2}\}$ i $\{f_{N2}, f_{N3}, f_{N4}\}$ odgovaraju pri-

[10] Kod većine glagola infinitivna se osnova dobiva odbacivanjem infinitivnog nastavka *-ti*, a prezentska odbacivanjem nastavka u 3. licu množine prezenta (Barić *et al.*, 2005).

djevima odnosno imenicama ženskoga roda, odgovarajući derivacijski uzorak možemo definirati kao:

$$d_1 = (sfx(ina), \{f_{A1}, f_{A2}\}, \{f_{N2}, f_{N3}, f_{N4}\})\,.$$

Pritom valja napomenuti da smo apstraktnost kao značajku izvedenice ovdje zanemarili, budući da je riječ o značenjskoj kategoriji koja nije razlučiva na morfološkoj razini.

Derivacijski uzorak modela HOFM razlikuje se od uzorka iz tradicionalne gramatike i po tome što je u modelu HOFM preoblika t definirana u odnosu na obličnu osnovu kao ishodišni oblik, dok je u gramatici ishodište preoblike tvorbena osnova. U slučajevima u kojima tvorbena i oblična osnova nisu identične (npr. kod nekih tvorbenih uzoraka motiviranih glagolima), preobliku je potrebno definirati kao kompoziciju dviju preoblika: jedne kojom se ostvaruje preoblika iz oblične osnove u tvorbenu te druge kojom se tako dobivena tvorbena osnova preobličuje u lemu izvedenice.

3.2.10 Izvođenje riječi

Derivacijski uzorak koristimo za izvođenje izvedenice zadane riječi. Postupak izvođenja riječi u apstraktnome je modelu formaliziran funkcijom *lDerive* prema (3.14). U modelu HOFM kategorija izvedenice predstavljena je skupom flektivnih uzoraka, to jest $\mathcal{Y} = \wp(\mathcal{F})$. Prema tome, funkcija za izvođenje izvedenice oblika je $lDerive : \mathcal{S} \times \mathcal{F} \times \mathcal{D} \to \wp(\mathcal{S} \times \wp(\mathcal{F}))$, a definirana je na sljedeći način:

$$lDerive\Big(l_1, f_1, (t, \mathcal{F}_1, \mathcal{F}_2)\Big) = \begin{cases} \Big\{(\overbrace{(t \circ t_0^{-1})(l_1)}^{l_2}, \mathcal{F}_2')\Big\} & \text{ako } f_1 \in \mathcal{F}_1 \wedge \mathcal{F}_2' \neq \emptyset\,, \\ \emptyset & \text{inače}\,, \end{cases} \tag{3.29}$$

gdje

$$\mathcal{F}_2' = \{f_2 \in \mathcal{F}_2 \ : \ f_2 \vDash l_2\}\,,$$

pri čemu je t_0 preoblika oblične osnove u lemu definirana flektivnim uzorkom f_1. U proceduralnom smislu izvođenje se provodi tako da se iz leme osnovne riječi l_1 najprije izvodi oblična osnova uporabom inverza preoblike t_0 definirane flektivnim uzorkom f_1, a zatim se tako dobivena tvorbena osnova preoblikom t preobličuje u lemu izvedenice l_2. Lemi izvedenice pridružuju se oni flektivni uzorci iz skupa $\mathcal{F}_2$ koji su na dotičnu lemu primjenjivi. Ako osnovna riječ ne pripada kategoriji određenoj skupom flektivnih uzoraka $\mathcal{F}_1$, ili pak ako niti jedan od uzoraka iz skupa $\mathcal{F}_2$ na izvedenicu nije primjenjiv,

tada derivacijski uzorak nije primjenjiv na osnovnu riječ i funkcija *lDerive* rezultira praznim skupom.

3.2.11 Relacija tvorbene veze

Derivacijskim se uzorkom između osnovne riječi i izvedenice implicitno uspostavlja tvorbena veza. Tu ćemo vezu definirati formalno na sljedeći način. Relacija *(izravne) tvorbene veze* temeljem uzorka d, označena kao '$\rightarrow_d$', jest binarna relacija na skupu $S \times \mathcal{F}$ definirana kao:

$$(l_1, f_1) \rightarrow_d (l_2, f_2) \iff (l_2, f_2) \in lDerive(d, l_1, f_1)\,. \tag{3.30}$$

Relacija se uspostavlja između parova dvojki sastavljenih od leme i flektivnog uzorka; flektivnim uzorkom određena je pripadnost leme određenoj kategoriji. Dvojka (l_1, f_1) tvorbeno je, temeljem derivacijskog uzorka d, povezana s dvojkom (l_2, f_2), odnosno $(l_1, f_1) \rightarrow_d (l_2, f_2)$, ako i samo ako je uporabom uzorka d lemu l_2 (i pripadnu joj kategoriju) moguće izvesti iz leme l_1 (i njoj pripadne kategorije).

Važno je napomenuti da relacija tvorbene veze $\rightarrow_d$ ne implicira stvarnu tvorbenu povezanost. Relacija samo upućuje na to da je tvorba gramatički moguća, odnosno da između osnovne riječi i izvedenice postoji izrazna veza. Za tvorbenu je povezanost potrebno međutim postojanje kako izrazne tako i značenjske veze između tih riječi. Kao primjer može poslužiti derivacijski uzorak $d = (sfx(ica), \mathcal{F}_1, \mathcal{F}_2)$, kojim je opisana tvorba imenica ženskog roda dodavanjem sufiksa *-ica* imenicama muškog roda (npr. *bankar* → *bankarica*). Ovim će uzorkom u vezu također biti dovedene i riječi koje nisu u tvorbenoj vezi, poput *šal* i *šalica*, između njih ne postoji sadržajna veza. Ovaj se problem u kontekstu morfološke normalizacije detaljnije razmatra u odjeljku 6.2.1.

3.3 Nadogradnja modela

U ovom odjeljku opisano je nekoliko nadogradnji osnovnog modela HOFM. Nadogradnjama se konkretiziraju neki aspekti osnovnog modela, s ciljem povećanja izražajnosti i pojednostavljenja modeliranja određenih morfoloških procesa. Najbitniju nadogradnju predstavlja prikaz preoblika kao nefunkcijskih relacija (višestruke preoblike) te modeliranje različitih vrsta opcionalnosti, čime se povećava izražajnost modela. S druge strane, uvođenjem osnovnih vrsta preoblika ograničava se izražajnost modela, ali se pojednostavljuje modeliranje i programska izvedba. U nastavku je najprije opisana nadogradnja flektivnog uzorka tako da on uključuje dodatne morfosintaktičke opise.

3.3.1 Morfosintaktički opisi

Flektivni uzorak definiran izrazom (3.18) može se proširiti tako da se svakoj preoblici pridruži morfosintaktički opis oblika izvedivog dotičnom preoblikom:

$$f = \Big(c, \{(t_0, x_0), \ldots, (t_n, x_n)\}\Big) \ \in \mathcal{C} \times \wp(\mathcal{T} \times \mathcal{X})\,, \tag{3.31}$$

gdje je $\mathcal{X}$ skup morfosintaktičkih opisa, a $x_0, \ldots, x_n$ su morfosintaktički opisi pridruženi pojedinim preoblikama znakovnoga niza. Morfosintaktički opisi, osim vrste riječi, mogu sadržavati i opise gramatičkih kategorija poput roda, broja, padeža te drugih kategorija ovisno o vrsti riječi. Za hrvatski jezik u tu svrhu mogu se koristiti morfosintaktički opisi prema normi MULTEXT-East (Erjavec *et al.*, 2003), kod koje su vrijednosti pojedinih morfosintaktičkih kategorija sažeto kodirani u jedan znakovni niz (v. odjeljak 4.2.4).

3.3.2 Višestruke preoblike

U odjeljku 3.2.1 preoblika znakovnoga niza definirana je kao funkcija oblika $t : \mathcal{S} \to \mathcal{S}$ koja ima dobro definirani inverz za sve znakovne nizove iz skupa $\mathcal{S} \setminus \{\varepsilon\}$. U praksi se međutim, barem u slučaju hrvatskoga jezika, takav zahtjev pokazuje suviše ograničavajućim. Problem naime predstavljaju fonološke alternacije koje u mnogim slučajevima nisu injektivne, a u nekima čak nisu niti funkcijske.

Tipičan je primjer fonološka alternacija palatalizacije. Palatalizaciju smo u dijelu 3.2.2 prikazali preoblikom *plt* i definirali izrazom (3.16). No puni oblik te preoblike trebao bi, sukladno (Barić *et al.*, 2005), biti sljedeći:

$$plt = asfx(\{(k, č), (g, ž), (h, š), (c, č), (z, ž)\})\,. \tag{3.32}$$

Problem predstavljaju parovi alternacija *k/č* (npr. *vojnik→vojniče*) i *c/č* (npr. *stric→striče*) te *g/ž* (npr. *drug→druže*) i *z/ž* (npr. *vitez→viteže*), zbog kojih ova preoblika nije injektivna funkcija, pa njezin inverz plt^{-1} nije definiran.

Ovaj se problem u načelu može riješiti razdvajanjem (faktorizacijom) preoblike na skup injektivnih preoblika. To je rješenje nepraktično jer bi, čak i onda kada bi se automatiziralo, rezultiralo umjetnim odnosno tehnički motiviranim povećanjem broja flektivnih uzoraka. Bolje rješenje, koje otvara i neke druge mogućnosti, jest dopustiti da preoblika bude neinjektivna, ali onda u obzir uzimati sve njezine moguće inverze. Tako poopćena funkcija preoblike znakovni niz preslikava u skup mogućih znakovnih nizova, pa ćemo je nazivati *višestrukom preoblikom*.

Formalno, višestruka preoblika je funkcija

$$t : \mathcal{S} \rightarrow \wp(\mathcal{S})\,.$$

Pritom $t(s) = \emptyset$ indicira da je preoblika t na znakovnom nizu s zakazala. Preobliku t za koju vrijedi $\exists s \in \mathcal{S}.\ |t(s)| > 1$ zvat ćemo *višeznačnom preoblikom*. Nul-preobliku definiramo kao $nul(s) = \{s\}$, dok $fail(s) = \emptyset$. Neka je $\mathcal{T}$ skup svih višestrukih preoblika.[11]

Funkcijska kompozicija višestrukih preoblika, $\circ : \mathcal{T} \times \mathcal{T} \rightarrow \mathcal{T}$, definirana je kao:

$$(t_2 \circ t_1)(s_1) = \{t_2(s_2) \ : \ s_2 \in t_1(s_1)\}\,. \tag{3.33}$$

Kompozicija dviju višestrukih preoblika jest višestruka preoblika koja izvodi sve moguće kombinacije znakovnoga niza. Iz definicije slijedi da, ako $t_1(s_1) = \emptyset$, onda $(t_2 \circ t_1)(s_1) = \emptyset$, tj. ako ijedna od pojedinačnih preoblika zakazuje, također zakazuje i njihova kompozicija. Nul-preoblika *nul* je neutralni element kompozicije, $nul \circ t = t \circ nul = t$.

Inverz višestruke preoblike, $^{-1} : \mathcal{T} \rightarrow \mathcal{T}$, definiran je kao

$$t^{-1}(s_1) = \{s_2 \ : \ s_1 \in t(s_2)\}\,. \tag{3.34}$$

Formalno gledano, poopćenje modela preoblike s funkcije oblika $\mathcal{S} \rightarrow \mathcal{S}$ na funkciju oblika $t : \mathcal{S} \rightarrow \wp(\mathcal{S})$ odgovara poopćenju na relaciju T, $T \subset (\mathcal{S} \times \mathcal{S})$, pri čemu $s_2 \in t(s_1) \Leftrightarrow (s_1, s_2) \in T$. Višestruka preoblika u tom smislu odgovara nefunkcijskoj relaciji.[12]

Ilustrirajmo uporabu višestrukih preoblika na primjeru preoblike *vojnik→vojniče*, definirane kao $sfx(a) \circ plt$, gdje je preoblika *plt* dana izrazom (3.32). Primjenom preoblike t na obličnu osnovu *vojnik* dobivamo:

$$(sfx(a) \circ plt)(vojnik) = sfx(a)(vojnič) = \{vojniče\}\,,$$

dok primjenom inverza na dobiveni oblik dobivamo:

$$\begin{aligned}(sfx(a) \circ plt)^{-1}(vojniče) &= (plt^{-1} \circ sfx(a)^{-1})(vojniče)\\ &= plt^{-1}(vojnič) = \{vojnik, vojnic\}\,,\end{aligned}$$

tj. dobivamo dvije moguće oblične osnove budući da preoblika *plt* izvorno nije injektivna.

[11] Ovdje koristimo istu oznaku kao i za jednostruke preoblike, no budući da se u nastavku koriste isključivo višestruke preoblike, mogućnost zabune ne postoji.

[12] Budući da je model temeljen na funkcijama kao osnovnim gradivnim blokovima, u nastavku je ipak zadržan "funkcijski pogled" na ovakvu vrstu preoblika.

3.3.3 Operatori odabira

Osim što njima rješavamo problem inverza neinjektivnih preoblika, višestrukim preoblikama možemo u modelu eksplicitno obuhvatiti koncept opcionalnosti. Taj je mehanizam koristan u slučajevima kada jednom preoblikom želimo eksplicitno modelirati izvođenje više mogućih oblika. To može biti zato što je na tome mjestu moguće postojanje više ravnopravnih oblika (tzv. dvostrukosti), ili zato što je preoblika nekonzistentna u smislu da se na neke oblike primjenjuje, a na neke ne, pa modeliranje opcionalnosti predstavlja svojevrsno sredstvo apstrakcije.

Opcionalnost preoblika obuhvatit ćemo uvođenjem dvaju operatora. Prvi je operator *ravnopravnog odabira* između dviju preoblika, $| : \mathcal{T} \times \mathcal{T} \to \mathcal{T}$, definiran kao:

$$(t_1 \,|\, t_2)(s) = t_1(s) \cup t_2(s)\,. \tag{3.35}$$

Rezultat primjene operatora jest preoblika koja kao rezultat daje uniju rezultata pojedinačnih preoblika. Ako preoblike t_1 i t_2 nisu međusobno isključive, preoblika $t_1 \,|\, t_2$ je višeznačna preoblika. Na nekim znakovnim nizovima neka od dviju preoblika može zakazati, pa će onda i njezin rezultat izostati.

Drugi je operator *pristranog odabir* između dviju preoblika, $|| : \mathcal{T} \times \mathcal{T} \to \mathcal{T}$, definiran na sljedeći način:

$$(t_1 \,||\, t_2)(s) = \begin{cases} t_1(s) & \text{ako } t_1(s) \neq \emptyset\,, \\ t_2(s) & \text{inače}\,. \end{cases} \tag{3.36}$$

U ovome slučaju druga se preoblika primjenjuje samo onda kada prva preoblika zakaže. Ako preoblike t_1 i t_2 nisu višeznačne, onda to neće biti niti preoblika $t_1 \,||\, t_2$. Operator pristranog odabira koristan je u slučajevima kada je prva preoblika specifičnija od druge te zbog toga nad drugom predoblikom ima prednost. Pritom preobliku t_1 smatramo *specifičnijom* od preoblike t_2, ako i samo ako $\{s \in \mathcal{S} :\ t_1(s) \neq \emptyset\} \subset \{s \in \mathcal{S} :\ t_2(s) \neq \emptyset\}$ tj. ako je preoblika t_1 primjenjiva (ne zakazuje) samo na nekim od znakovnih nizova na kojima je primjenjiva (ne zakazuje) preoblika t_2.

Operatori '$|$' i '$||$' distributivni su prema operaciji kompozicije preoblika:

$$t_1 \circ (t_2 \,|\, t_3) = (t_1 \circ t_2) \,|\, (t_1 \circ t_3)\,, \tag{3.37}$$

$$(t_1 \,|\, t_2) \circ t_3 = (t_1 \circ t_3) \,|\, (t_2 \circ t_3)\,, \tag{3.38}$$

$$t_1 \circ (t_2 \,||\, t_3) = (t_1 \circ t_2) \,||\, (t_1 \circ t_3)\,, \tag{3.39}$$

$$(t_1 \,||\, t_2) \circ t_3 = (t_1 \circ t_3) \,||\, (t_2 \circ t_3)\,. \tag{3.40}$$

Za inverze preoblika dobivenih pomoću ovih operatora vrijedi, sukladno (3.34):

$$(t_1 \,|\, t_2)^{-1}(s) = t_1^{-1}(s) \cup t_2^{-1}(s)\,, \tag{3.41}$$

$$(t_1 \,||\, t_2)^{-1}(s) = t_1^{-1}(s) \cup \{r \in t_2^{-1}(s) \;:\; t_1(r) = \emptyset\}\,. \tag{3.42}$$

Računanje inverza preoblike $t1 \,||\, t_2$ iziskuje dodatno pojašnjenje, budući da je ono bitno za programsku izvedbu. Sukladno definiciji (3.36), kod operatora pristranog odabira preoblika t_1 ima prednost nad preoblikom t_2, pa $(t_1 \,||\, t_2)^{-1}(s)$ u svakom slučaju uključuje skup $t_1^{-1}(s)$. No, znakovni niz s možda se također može dobiti preoblikom t_2 iz nekog znakovnoga niza r, tj. $r = t_2^{-1}(s)$. Ako na znakovni niz r preoblika t_1 nije primjenjiva, tj. ako $t_1(r) = \emptyset$, onda je, sukladno (3.36), na nizu r bila primijenjena preoblika t_2. Zbog toga sve takve znakovne nizove također treba uključiti u rezultat.

3.3.4 Modeliranje opcionalnosti

U nastavku su razmotrena četiri tipična slučaja uporabe operatora za modeliranje opcionalnosti preoblika.

1. Prvi slučaj odnosi se na situaciju kada postoji više preoblika, ali su one međusobno isključive. Kod fleksije, primjerice, to može biti slučaj kada odabir ispravne preoblike zavisi o završetku oblične osnove. Primjer su imenice *krv* i *mast*, koje se obje dekliniraju prema istome flektivnom uzorku, no kod prve se u instrumentalu jednine provodi jotacija oblične osnove *v/vlj*, dok se kod druge provodi alternacija suglasničkog skupa *st/šć*. Ovu vrstu opcionalnosti možemo formalizirati preoblikom definiranom na sljedeći način:

$$t = sfx(u) \circ (rsfx(v, vlj) \,|\, rsfx(st, šć))\,,$$

   gdje je *rsfx* funkcija višeg reda koja definira preobliku zamjene završetka osnove (v. odjeljak 3.3.6). Primjenom preoblike t na oblične osnove *krv* i *mast* dobivamo $t(krv) = \{krvlju\}$ odnosno $t(mast) = \{mašću\}$. Valja primijetiti je da u ovom slučaju, budući da su preoblike međusobno isključive, svejedno upotrebljava li se operator ‘ | ’ ili ‘ || ’.

2. Drugi slučaj odnosi se na situacije u kojima jedna preoblika treba rezultirati s više mogućih oblika. U tom slučaju međusobno neisključive preoblike potrebno je udružiti operatorom ‘ | ’ u jednu višeznačnu preobliku. Tipičan primjer kod fleksije

predstavljaju slučajevi tzv. dvostrukosti kod kojih za jednu te istu gramatičku kategoriju postoji više mogućih oblika. Tako se, primjerice, preoblika za genitiv pridjeva muškog roda, koji u hrvatskom jeziku može imati kraći nastavak '*-og*' ili duži nastavak '*-oga*', može definirati preoblikom:

$$t = sfx(og) \,|\, sfx(oga)\,,$$

pa onda $t(brz) = \{brzog, brzoga\}$. Moguća je i kombinacija ovog i prvoga navedenog slučaja, tj. situacija u kojoj su samo neke od preoblika međusobno isključive. Npr. imenice *krv* i *mast* u instrumentalu jednine, pored gore spomenutih oblika, imaju i dodatne oblike *krvi* odnosno *masti*. Preobliku koja obuhvaća oba slučaja možemo definirati kao

$$t = \Big(sfx(u) \circ (rsfx(v, vlj) \,|\, rsfx(st, šć)\Big) \,|\, sfx(i)\,,$$

pa onda $t(krv) = \{krvlju, krvi\}$ i $t(mast) = \{mašću, masti\}$.

3. Treći slučaj jesu preoblike koje su opcionalne u smislu da se mogu, ali i ne moraju provesti. Tu vrstu opcionalnosti formalizirat ćemo funkcijom $opt : \mathcal{T} \to \mathcal{T}$ definiranom kao:

$$opt(t) = t \,|\, nul\,. \tag{3.43}$$

Primjer je imenica *tvrtka* koja u dativu i lokativu jednine može imati dva oblika: *tvrtci* i *tvrtki*, tj. sibilarizacija oblične osnove *tvrtk* u ovom je slučaju opcionalna. Odgovarajuću preobliku možemo (temeljem jednakosti (3.37) i (3.43)) definirati kao

$$t = (sfx(i) \circ sbl) \,|\, sfx(i) = sfx(i) \circ opt(sbl)\,,$$

Tada vrijedi $t(tvrtk) = \{tvrtci, tvrtki\}$, dok, primjerice, $t(žaba) = \{žabi\}$, budući da preoblika *sbl* na osnovi *žab* zakazuje.

4. Četvrti slučaj jesu preoblike koje ne moraju nužno uspjeti i koje, ako zakažu, znakovni niz trebaju ostaviti nepromijenjenim. Kako je potreba za takvim preoblikama česta, uvodimo funkciju $try : \mathcal{T} \to \mathcal{T}$, definiranu kao:

$$try(t) = t \,||\, nul\,. \tag{3.44}$$

Tipičan primjer uporabe funkcije *try* jest opisivanje fonološki uvjetovanih alternacija. Na primjer, kod imenice *vrabac* neproširena osnova *vrabc* javlja se obezvučena, tj. ispred završnog suglasnika *c* provodi se alternacija *b/p*. Odgovarajuću funkciju preoblike za, primjerice, genitiv jednine, možemo definirati na sljedeći način:

$$t = sfx(a) \circ try\Big(rsfx(bc, pc)\Big)\,.$$

Primjenom ove preoblike na obličnu osnovu dobivamo $t(vrabc) = \{vrapca\}$. Na osnovama kod kojih obezvučenja nema, kao npr. kod imenice *vranac*, preoblika *rsfx* zakazuje, no ne zakazuje i preoblika t, već vrijedi $t(vranc) = \{vranca\}$.

Druga učestala primjena funkcije *try* jest opisivanje alternacija refleksa jata, i to onda kada je ta alternacija, ukoliko je ostvariva, ujedno i obavezna. Primjer je alternacija *ije/je* koja se javlja u dugoj množini nekih imenica, npr. imenice *cvijet*. Odgovarajuću preobliku za, primjerice, nominativ množine definiramo na sljedeći način:

$$sfx(ovi) \circ try\Big(rifx(ije, je)\Big)\,,$$

gdje je *rifx* funkcija višeg reda koja definira zamjenu infikasa (v. odjeljak 3.3.6).

Za razliku od operatora '|', operator '||' nije komutativan, pa treba paziti da preoblika na lijevoj strani nije općenitija od preoblike s desne strane jer bi ta time bila zasjenjena. Na primjer, preoblika $nul \,||\, t$ identična je preoblici *nul* budući da preoblika *nul* ni na kojem znakovnom nizu ne zakazuje.

Kombiniranjem pojedinačnih preoblika lako se može dobiti neinjektivna preoblika, pa makar sve pojedinačne preoblike bile injektivne. Tipičan je primjer preoblika $try(t)$ koja je, uz (razumnu) pretpostavku $t \neq nul$ i $t \neq fail$, također neinjektivna. Sukladno (3.44) i (3.42), za njezin inverz vrijedi:

$$(try(t))^{-1}(s) = \begin{cases} t^{-1}(s) \cup \{s\} & \text{ako } t(s) = \emptyset\,, \\ t^{-1}(s) & \text{inače}\,. \end{cases} \tag{3.45}$$

Prema tome, ako preoblika t nije primjenjiva na znakovni niz s, $s \in t(s)$ – što će u praksi biti čest slučaj – onda je $try^{-1}(t)$ višeznačna preoblika. Tako na primjer $try(rsfx(bc, pc))(vrabc) = \{vrapc\}$, ali $(try(rsfx(bc, pc)))^{-1}(vrapc) = \{vrabc, vrapc\}$.

3.3.5 Generiranje oblika, lematizacija i izvođenje riječi

Izrazom (3.25) definirana je funkcija *wfs* za izvođenje svih oblika riječi predstavljene lemom kao osnovnim oblikom. Proširena inačica te funkcije, primjenjiva na flektivne uzorke s morfosintaktičkim opisima prema (3.31) te višestrukim preoblikama opisanima u odjeljku 3.3.2, jest funkcija $lWfsMsd : \mathcal{S} \times \mathcal{F} \to \wp(\mathcal{S} \times \mathcal{X})$ definirana na sljedeći način:

$$lWfsMsd\Big(l, \underbrace{(c, \{(t_0, x_0), \ldots, (t_n, x_n)\})}_{f}\Big) = \begin{cases} \bigcup_i \Big\{(w, x_i) \ : \ w \in (t_i \circ t_0^{-1})(l)\Big\} & \text{ako } f \vDash l, \\ \emptyset & \text{inače.} \end{cases} \tag{3.46}$$

Pojednostavljena inačica kod koje se morfosintaktički opisi jednostavno zanemaruju dana je s:

$$lWfs(f, l) = \Big\{w \ : \ (w, x) \in lWfsMsd(l, f)\Big\}. \tag{3.47}$$

Funkcija svođenja na lemu $wLemmaMsd : \mathcal{S} \times \mathcal{F} \to \wp(\mathcal{S} \times \mathcal{X})$, primjenjiva na flektivne uzorke s morfosintaktičkim opisima i višestrukim preoblikama, definirana je kao:

$$wLemmaMsd\Big(w, \overbrace{(c, T)}^{f}\Big) = \Big\{\Big(\underbrace{(t_0 \circ t_i^{-1})(w)}_{l}, x_i\Big) \ : \ (t_i, x_i) \in T, f \vDash l\Big\}, \tag{3.48}$$

dok je pojednostavljena inačica definirana kao:

$$wLemma(w, f) = \{l \ : \ (l, x) \in wLemmaMsd(l, f)\}. \tag{3.49}$$

Kao što je već bilo naglašeno u odjeljku 3.2.7, postupak svođenja oblika na lemu općenito je višeznačan. Prema tome, funkcije *wLemmaMsd* i *wLemmaMsd* u općenitom slučaju rezultiraju skupom mogućih lema. Kada se koriste višestruke preoblika, ta je višeznačnosti još izraženija. Tri su moguća izvora višeznačnosti pri svođenju oblika na lemu:

1. Ako neka od preoblika t_i nije injektivna, onda je t_i^{-1} višeznačna preoblika, pa $(t_0 \circ t_i^{-1})(w)$ može (ako t_0 ne zakaže) dati više lema. Ako je flektivni uzorak primjenjiv na više njih, postupak rezultira s više lema.

2. Ako je na oblik w primjenjivo više različitih preoblika $t_0 \circ t_i$, i ako je flektivni

uzorak primjenjiv na više od jedne tako dobivene leme, onda postupak također rezultira s više lema. (Ovaj slučaj ilustriran je u dijelu 3.2.7.)

3. Ako je preoblika t_0 višeznačna, onda $(t_0 \circ t_i^1)(w)$ također može dati više lema. Ako je flektivni uzorak primjenjiv na više od jedne takve leme, postupak opet rezultira s više lema.

Dodatni i u praksi najizraženiji izvor višeznačnosti proizlazi međutim iz toga što pri lematizaciji nije unaprijed poznato koji flektivni uzorak treba primijeniti na dani oblik riječi, već treba primijeniti sve uzorke, prema (3.15).

Izrazom (3.29) definirano je izvođenje izvedenice temeljem derivacijskog uzorka modela. Ako se za izvođenje koriste višestruke preoblike, funkciju je potrebno redefinirati kako slijedi:

$$lDerive\Big(l_1, f_1, (t, \mathcal{F}_1, \mathcal{F}_2)\Big) = \begin{cases} \Big\{(l_2, \mathcal{F}_2') \ : \ l_2 \in (t \circ t_0^{-1})(l_1)\Big\} & \text{ako } f_1 \in \mathcal{F}_1 \ \wedge \ \mathcal{F}_2' \neq \emptyset\,, \\ \emptyset & \text{inače}\,, \end{cases} \tag{3.50}$$

gdje

$$\mathcal{F}_2' = \{f_2 \in \mathcal{F}_2 \ : \ f_2 \vDash l_2\}\,.$$

Ako je preoblika t višeznačna, funkcija će rezultirati skupom izvedenica; ako derivacijski uzorak na par (l_1, f_1) nije primjenjiv, funkcija će rezultirati praznim skupom.

3.3.6 Osnovne vrste preoblika

U odjeljku 3.2.2 rečeno je da se u modelu HOFM preoblike znakovnoga niza tipično definiraju funkcijama višega reda, te da je za različite vrste preoblika moguće definirati različite funkcije višeg reda. Te funkcije u načelu mogu biti proizvoljne, međutim radi jednostavnosti modela poželjno je da se definira jedan manji polazni skup osnovnih funkcija pomoću kojih se onda mogu definirati i dodatne funkcije. Takav pristup također bitno pojednostavljuje programsku izvedbu modela.

Kod konkatenativnih morfologija, kakva je morfologija hrvatskoga jezika, preoblike su izrazive regularnim relacijama.[13] Nadalje, regularne relacije iskazive su operacijama zamjene (znakovnih) podnizova (Karttunen, 1995), što omogućava opis na višoj razini

[13] U okviru teorije automata, regularne relacije predstavljaju proširenje koncepta regularnih jezika na parove nizova. Regularni jezici mogu biti modelirani automatom s konačnim brojem stanja, dok regularne relacije mogu biti modelirane pretvaračem (preobličivačem) s konačnim brojem stanja (engl. *finite state transducer*, FST) (Karttunen, 2001).

apstrakcije. U nastavku ćemo definirati tri osnovne funkcije višega reda koje odgovaraju trima operacijama zamjene znakovnih podnizova:[14]

- zamjena sufiksa znakovnoga niza,[15]
- zamjena prefiksa znakovnoga niza i
- zamjena infiksa znakovnoga niza.

U nastavku 'α' i 'β' označavaju proizvoljne (moguće prazne) podnizove znakovnoga niza, $\alpha, \beta \in \mathcal{S}$. Zamjena sufiksa znakovnoga niza definirana je funkcijom $rsfx : \mathcal{S} \times \mathcal{S} \to \mathcal{T}$, kako slijedi:

$$rsfx(s_1, s_2) = \lambda w. \begin{cases} \{\alpha s_2\} & \text{ako } w = \alpha s_1\,, \\ \emptyset & \text{inače}\,. \end{cases} \tag{3.51}$$

Funkcija rezultira preoblikom koja, primijenjena na znakovni niz w, nastavak s_1 tog znakovnoga niza zamjenjuje nastavkom s_2. Ako w ne završava znakovnim nizom s_1, preoblika će zakazati.

Slično, zamjena prefiksa znakovnoga niza definirana je funkcijom $rpfx : \mathcal{S} \times \mathcal{S} \to \mathcal{T}$ na sljedeći način:

$$rpfx(s_1, s_2) = \lambda w. \begin{cases} \{s_2\alpha\} & \text{ako } w = s_1\alpha\,, \\ \emptyset & \text{inače}\,. \end{cases} \tag{3.52}$$

Obje su ove funkcije injektivne te vrijedi:

$$rsfx^{-1}(s1, s2) = rsfx(s2, s1)\,, \tag{3.53}$$

$$rpfx^{-1}(s1, s2) = rsfx(s2, s1)\,. \tag{3.54}$$

Funkcije za dodavanje odnosno uklanjanje sufiksa i prefiksa znakovnoga niza sada se mogu definirati na sljedeći način:

[14] U smislu izražajnosti (izražajne snage) ove su operacije redundantne. Uz uvođenje posebnih simbola za početak i kraj znakovnoga niza, zamjena sufiksa i zamjena prefiksa također se mogu opisati operacijom zamjene infiksa. Ipak, radi preglednosti, to ovdje nije učinjeno.

[15] Pojmovi "sufiks", "prefiks" i "infiks" u lingvistici i računarstvu imaju različita značenja. U računarstvu, riječ je o podnizu znakovnoga niza koji se nalazi na kraju, početku, odnosno unutar znakovnoga niza. U lingvistici su sufiks, prefiks i infiks funkcionalni morfemi, dakle isključivo oni odsječci riječi koji imaju gramatički sadržaj. U nastavku, gdje god postoji mogućnost zabune, za računarski smisao ovih riječi korišten je pojam "sufiks/prefiks/infiks znakovnoga niza".

$$\begin{aligned} sfx(s) &= rsfx(\varepsilon, s)\,, && (3.55)\\ pfx(s) &= rpfx(\varepsilon, s)\,, && (3.56)\\ dsfx &= sfx^{-1}\,, && (3.57)\\ dpfx &= pfx^{-1}\,. && (3.58) \end{aligned}$$

Treća osnovna vrsta preoblike jest zamjena infiksa znakovnoga niza definirana funkcijom $rifx : \mathcal{S} \times \mathcal{S} \to \mathcal{T}$ kao:

$$rifx(s_1, s_2) = \lambda w.\, \{\alpha s_2 \beta \;:\; \alpha, \beta \in \mathcal{S}, w = \alpha s_1 \beta\}\,. \quad (3.59)$$

Za razliku od prethodno definiranih funkcija, funkcija *rifx* rezultira višeznačnom preoblikom budući da se u općenitom slučaju infiks s_1 može u znakovnom nizu w pojavljivati na više mjesta. Za inverz, koji dakle također može biti višeznačna preoblika, vrijedi:

$$rifx^{-1}(s1, s2) = rifx(s2, s1)\,. \quad (3.60)$$

Kao što je ranije opisano, fonološke se alternacije mogu opisati funkcijom višega reda $asfx : \wp(\mathcal{S} \times \mathcal{S}) \to \mathcal{T}$. Funkcija rezultira preoblikom koja provodi zamjenu prema zadanom skupu parova završetaka osnove. Ovu funkciju možemo definirati na sljedeći način:

$$\begin{aligned} asfx(\emptyset) &= fail\,, && (3.61)\\ asfx(\{(s_1, r_1), \ldots, (s_n, r_n)\}) &= rsfx(s_1, r_1) \mid \ldots \mid rsfx(s_n, r_n)\,. \end{aligned}$$

Rezultirajuća preoblika bit će višeznačna samo onda kada pojedinačne zamjene nisu međusobno isključive. To će biti slučaj ako neki od znakovnih nizova $s_1, s_2, \ldots, s_n$ imaju identične sufikse. S druge strane, ako neki od znakovnih nizova $r_1, r_2, \ldots, r_n$ imaju identične sufikse, rezultirajuća preoblika neće biti injektivna, odnosno njezin će inverz biti višeznačna preoblika.

Korisnom će se također pokazati i funkcija $aifx : \wp(\mathcal{S} \times \mathcal{S}) \to \mathcal{T}$ kojom se dobiva (višeznačna) preoblika zamjene infikasa znakovnoga niza, definirana kako slijedi:

$$\begin{aligned} \mathit{aifx}(\emptyset) &= \mathit{fail}\,, \\ \mathit{aifx}(\{(i_1, r_1), \ldots, (i_n, r_n)\}) &= \mathit{rifx}(i_1, r_1) \,|\, \ldots \,|\, \mathit{rifx}(i_n, r_n)\,. \end{aligned} \tag{3.62}$$

U nastavku ćemo, preglednosti radi, umjesto $\mathit{asfx}(\{(s_1, r_1), (s_2, r_2), \ldots, (s_n, r_n)\})$ pisati $\mathit{asfx}\,\{s_1/r_1, s_2/r_2, \ldots, s_n/r_n\}$, te slično za funkciju *aifx*.

Opisane funkcije, uz ranije uvedene funkcije *nul*, *try* i *opt*, predstavljaju osnovni skup funkcija višeg reda pomoću kojih su u idućem poglavlju definirane i neke jezično-specifične preoblike za hrvatski jezik.

Poglavlje 4

Model morfologije hrvatskoga jezika

U ovome poglavlju opisana je primjena model HOFM, i to njegove nadograđene inačice, na morfologiju hrvatskoga standardnog jezika. Model se temelji na opisu morfologije prema (Barić *et al.*, 2005), a obuhvaća fleksiju i sufiksalnu tvorbu imenica, glagola i pridjeva. Ostale promjenjive vrste riječi modelom nisu obuhvaćene: u kontekstu morfološke normalizacije prilozi se mogu tretirati kao pridjevi, dok zamjenice i brojevi nisu od interesa budući da se u većini postupaka pretraživanja informacija i dubinske analize teksta te riječi ionako zanemaruju. Prefiksalna i prefiksalno-sufiksalna tvorba nisu obuhvaćene modelom iz razloga što je kod tih načina tvorbe značenjska povezanost polazne riječi i izvedenice nedovoljno jaka, pa bi u tom slučaju upitna bila korisnost morfološke normalizacije.

U nastavku je najprije opisan način modeliranja preoblika, uključivo jezično-specifičnih preoblika za modeliranje fonoloških alternacija. U drugom i trećem dijelu opisano je modeliranje flektivnih odnosno derivacijskih uzoraka.

4.1 Preoblike

Model flektivne i derivacijske morfologije hrvatskoga jezika svodi se na modeliranje sljedeće tri vrste morfoloških procesa (preoblika):

- dodavanje obličnih nastavaka (kod fleksije) odnosno sufikasa (kod tvorbe),
- dodavanje prefiksa kod fleksije (isključivo za superlative pridjeva),
- fonološke alternacije.

Tablica 4.1: Preoblike korištene za modeliranje morfologije hrvatskoga jezika.

Funkcija	Opis	Primjer primjene
Dodavanje/zamjena nastavka, sufiksa i prefiksa:		
$rsfx$	Zamjena završetka osnove	$rsfx(ij, ik)(akademija) = \{akademik\}$
sfx	Dodavanje nastavka ili sufiksa	$sfx(a) = (vojnik) = \{vojnika\}$
pfx	Dodavanje prefiksa	$(pfx(naj) \circ sfx(a) \circ jot)(brz) = \{najbrži\}$
Fonološki uvjetovane alternacije:		
pca_1	Alternacija osnove uvjetovana fonemom unutar osnove	$(sfx(a) \circ pca_1)(vrabc) = \{vrapca\}$
pca_2	Alternacija uvjetovana fonemom u sufiksu	$(pca_2 \circ sfx(ba))(svat) = \{svadba\}$
Morfološki uvjetovane alternacije:		
sbl	Sibilarizacija	$(sfx(i) \circ sbl)(vojnik) = \{vojnici\}$
plt	Palatalizacija	$(sfx(e) \circ plt)(vojnik) = \{vojniče\}$
jot	Jotacija	$(jot(u) \circ jot)(krv) = \{krvlju\}$
acg	Alternacije sugl. skupova	$(sfx(u) \circ acg)(mast) = \{mašću\}$
exa	Proširenje samoglasnikom *a*	$(sfx(a) \circ exa)(vrabc) = \{vrabaca\}$
exe	Proširenje samoglasnikom *e*	$exa(Čakovc) = \{Čakovec\}$
jat_1	Alternacija refleksa jata *ije/je*	$(sfx(ovi) \circ jat_1)(snijeg) = \{snjegovi\}$
jat_2	Alternacija refleksa jata *ije/e*	$(sfx(ovi) \circ jat_2)(brijeg) = \{bregovi\}$
jat_3	Alternacija refleksa jata *ije/i*	$(sfx(vati) \circ {jat_3}^{-1})(izli) = \{izlijevati\}$

Navedeni morfološki procesi modelirani su preoblikama definiranima funkcijama višeg reda opisanima u odjeljku 3.3.6. Dodavanje nastavka i sufiksa modelirano je funkcijom dodavanja sufiksa znakovnoga niza *sfx*, a dodavanje prefiksa funkcijom dodavanja prefiksa znakovnoga niza *pfx*. Fonološke alternacije modelirane su funkcijama zamjene sufiksa znakovnoga niza (*rsfx* odnosno *asfx*) i funkcijama zamjene infiksa znakovnoga niza (*rifx* odnosno *aifx*). Model se temelji na višestrukim preoblikama opisanima u odjeljku 3.3.2, budući da to model čini znatno sažetijim i preglednijim.

U tablici 4.1 sažeto su prikazane sve preoblike korištene za modeliranje morfologije hrvatskog jezika. U nastavku su razmotrene neke jezično-specifične preoblike kojima su modelirane fonološke alternacije.

4.1.1 Fonološke alternacije

Kao posljedica međusobnih veza između morfema u riječi, odnosno fonema kao njihovih sastavnih dijelova, u morfemima može doći do promjene u njihovom fonemskom sastavu. Zbog toga se jedno te isti morfem može u različitim oblicima iste riječi ili u različitim tvorbeno povezanim riječima pojaviti u različitim izrazima odnosno *alomorfima* (Barić *et al.*, 2005). Promjene u fonemskom sastavu morfema očituju se kao alternacije pojedinačnih fonema i fonemskih skupova (odnosno njima odgovarajućih grafema i grafemskih skupova). Alternacije mogu mijenjati osnovu riječi, oblični nastavak, sufiks ili prefiks. Pritom razlikujemo:

- *Fonološki uvjetovane alternacije*: alternacije uvjetovane sastavom fonemskog skupa na morfemskoj granici (npr. jednačenje po zvučnosti odnosno obezvučivanje *b/p* u *vrabac→vrapca*);[1]
- *Morfološki uvjetovane alternacije*: alternacije uvjetovane morfološkim kategorijama (npr. palatalizacija *k/č* u *vojnik→vojniče*).

Osnovna je dakle razlika između ovih dviju vrsta alternacija u tome što se morfološki uvjetovane alternacije provode samo u određenim morfološkim kategorijama (npr. samo u određenim padežima nekih flektivnih uzoraka), dok se fonološki uvjetovane alternacije u načelu odnose na sav jezični materijal.[2] Zbog ove je razlike fonološki uvjetovane alternacije potrebno modelirati drugačije od onih morfološki uvjetovanih. U nastavku se najprije razmatra modeliranje morfološki uvjetovanih alternacija.

4.1.2 Modeliranje morfološki uvjetovanih alternacija

Morfološki uvjetovane alternacije jesu: nepostojani samoglasnici, prijeglas, proširivanje osnove, vokalizacija, palatalizacija, sibilarizacija, jotacija, alternacije suglasničkih skupova i alternacija refleksa jata.

Prijeglas (npr. *selom – polj̲em*) i vokalizacija (npr. *anđeo – anđe̲la*) nisu eksplicitno modelirane kao alternacije već su implicitno ugrađene u preoblike pojedinih flektivnih uzoraka. Ostale alternacije modelirane su kako slijedi.

Sibilarizacija. Zamjenjivanje velaranih suglasnika *k*, *g* i *h* sibilantima *c*, *z*:

[1] U nastavku ćemo alternacije bilježiti kao parove grafema, umjesto kao parove transkribiranih fonema, kao što je to u lingvistici uobičajeno.

[2] Premda i ovdje postoje neka odstupanja, i to uglavnom na granicama riječi kod složenica, kod posuđenica (npr. *plebs*) te kod stranih vlastitih imena (npr. *Liszt*).

$$sbl \;=\; asfx\,\{(k,c),(g,z),(h,s)\}\,. \tag{4.1}$$

Palatalizacija. Zamjenjivanje nepalatalnih suglasnika palatalnim suglasnicima:

$$plt = asfx\,\{(k,\check{c}),(g,\check{z}),(h,\check{s}),(c,\check{c}),(z,\check{z})\}\,. \tag{4.2}$$

Jotacija. Zamjenjivanje nepalatalnih suglasnika palatalnim suglasnikom, a suglasnika *p*, *b*, *m*, *v* i *f* suglasničkim skupovima *plj*, *blj*, *mlj*, *vlj* odnosno *flj*:

$$\begin{aligned} jot \;=\; & plt \mid asfx\,\{(s,\check{s}),(t,\acute{c}),(d,\đ),(l,lj),(n,nj),(p,plj),(b,blj),\\ & (m,mlj),(v,vlj),(f,flj)\}\,. \end{aligned}$$

Jotacija je modelirana pomoću palatalizacije, budući da su su svi alternirajući parovi palatalizacije ujedno i alternirajući parovi jotacije.

Alternacija suglasničkog skupa. Na granici osnove i nastavka odnosno na granici osnove i sufiksa zamjenjuje se skup od dva suglasnika:

$$\begin{aligned} acg \;=\; & asfx\,\{(ht,\check{s}\acute{c}),(sk,\check{s}\check{c}),(sl,\check{s}lj),(sn,\check{s}nj),(st,\check{s}\acute{c}),(st,\check{s}t),\\ & (zd,\check{z}\đ),(zn,\check{z}nj)\}\,. \end{aligned}$$

Proširivanje osnove. Proširivanje osnove samoglasnikom *a* (tzv. nepostojano *a*) odnosno samoglasnikom *e* (tipično za kajkavsko narječje), kojom se ti samoglasnici umeću između dva suglasnika pri završetku osnove, modelirano je također funkcijom zamjene sufiksa *asfx*, na sljedeći način:

$$exa \;=\; asfx\,\{(a\alpha,\alpha)\;:\;\alpha \in cons\}\,, \tag{4.3}$$

$$exe \;=\; asfx\,\{(e\alpha,\alpha)\;:\;\alpha \in cons\}\,. \tag{4.4}$$

gdje je *cons* skup suglasnika. Na primjer, $exa(vrabc) = \{vrabac\}$ i $exe^{-1}(\check{C}akovec) = \{\check{C}akovc\}$. Proširivanje osnove različitim umetcima ispred nastavaka u pojedinim padežima (npr. *kralj – kralj-ev-i* nije modelirano kao alternacija, već je impli-

citno ugrađeno u flektivne uzorke (npr. kao preoblika $sfx(evi)$). Isto vrijedi i za proširivanje kraja oblične osnove tzv. navescima (npr. *brzog – brzoga*; v. str. 47), te proširenje početka osnova nekih glagola u infinitivu.

Alternacije refleksa jata. Alternacije tzv. refleksa jata *ije/je*, *ije/e*, *ije/i* modelirane su preoblikama:

$$jat_1 = rifx(ije, je)\,, \tag{4.5}$$

$$jat_2 = rifx(ije, e)\,, \tag{4.6}$$

$$jat_3 = rifx(ije, i)\,, \tag{4.7}$$

odnosno njihovim inverzima. Ako je alternacija jata uvjetovana određenom fonološkom okolinom, onda je taj uvjet eksplicitno ugrađen u podnizove koji se zamjenjuju. Primjerice, ako se alternacija *ije/e* provodi samo iza suglasnika *r*, modelirana je kao $rifx(rije, re)$. Alternacija *ije/e*, koja se provodi iza alternacije *l/lj* odnosno *n/nj* (kao npr. u *snijeg→snjegovi*), modelirana je izravno kao alternacija jat_1.

Preoblike za morfološki uvjetovane alternacije kod flektivnih se uzoraka u načelu primjenjuju bezuvjetno, dok se kod derivacijskih uzoraka primjenjuju uvjetno u smislu operatora *try*. U slučaju dvostrukosti, preoblike se modeliraju kao opcionalne u smislu operatora *opt*.

4.1.3 Modeliranje fonološki uvjetovanih alternacija

Fonološki uvjetovane alternacije jesu jednačenje po zvučnosti (obezvučivanje i ozvučivanje, npr. *vrabac→vrapca* i *svat→svadba*), jednačenje po mjestu tvorbe (npr. *stan→stambeni*) i ispadanje suglasnika (npr. *zadatak→zadaci*).

Budući da prefiksalna tvorba modelom nije obuhvaćena, a da kod flektivnog prefiksa *naj-* nema alternacija, to niti alternacija prefiksa nije obuhvaćena modelom. Uz izuzeće dakle alternacija koje mijenjaju prefiks ili početak osnove, većina fonološki uvjetovanih alternacija mijenja kraj osnove. Takve alternacije možemo prikazati kao skup trojki (s, r, C), gdje su $(s, r) \in \mathcal{S} \times \mathcal{S}$ alternirajući par grafema, a $C \subset \mathcal{S}$ je skup grafema koji uvjetuju alternaciju (odnosno desna fonološka okolina). Da bi se provela alternacija s/r, grafem s mora se u znakovnome nizu naći neposredno ispred nekog od grafema iz skupa

C. Obezvučivanje, ozvučivanje, jednačenje po mjestu tvorbe i ispadanja suglasnika mogu se dakle, sukladno (Barić *et al.*, 2005), definirati na sljedeći način:

$$\begin{aligned} obzv &= \Big\{(p,b,\{dž\}),(t,d,\{b\}),(s,z,\{b,d,g\}),(š,ž,\{b\}), \\ &\quad (č,dž,\{b\}),(k,g,\{dž\}),(s,ž,\{dž\}),(z,ž,\{dž\})\Big\}, \end{aligned} \tag{4.8}$$

$$\begin{aligned} ozv &= \Big\{(b,p,\{c,č,ć,f,h,k,s,š,t\}),(d,t,\{f,h,k,p\}),(g,k,\{c,č\}), \\ &\quad (đ,ć,\{k\}),(z,s,\{c,f,h,k,p,t\}),(ž,š,\{c,k\})\Big\}, \end{aligned} \tag{4.9}$$

$$\begin{aligned} jmt &= \Big\{(s,š,\{č,ć,dž,đ,lj,nj\}),(z,š,\{č,ć,dž,đ,lj,nj\}), \\ &\quad (z,ž,\{č,ć,dž,đ,lj,nj\}),(h,š,\{č,ć,dž,đ,lj,nj\}), \\ &\quad (n,m,\{b,p\})\Big\}, \end{aligned} \tag{4.10}$$

$$\begin{aligned} isg &= \Big\{(t,\varepsilon,\{c,č,ština\},(d,\varepsilon,\{c,č,ština\},(\alpha t,t,\{\beta\}),(\alpha d,d,\{\beta\})\Big\} \\ &\quad \alpha=\{s,z,š,ž\},\ \beta=cons\setminus\{r,v\}, \end{aligned} \tag{4.11}$$

gdje je *cons* skup suglasnika hrvatskog jezika.

Pri modeliranju fonološki uvjetovanih alternacija razlikujemo dva slučaja:

1. Alternacija nastupa u obličnoj osnovi ispred fonema koji je dio osnove. Primjer je obezvučivanje osnove *ž/š* ispred suglasnika *c* ili *k* (npr. u *lažac – laš̲ca*). Ovaj slučaj nastupa kod riječi koje završavaju suglasničkom skupinom, a koje u oblicima s nultim sufiksom imaju proširenu osnovu.

2. Alternacija nastupa u obličnoj osnovi ispred fonema koji je dio obličnog nastavka ili sufiksa. Primjer je ozvučivanje osnove *t/d* ispred suglasnika *b* (npr. u *svatba → svad̲ba*). Ovaj slučaj odnosi se gotovo isključivo na tvorbu riječi.

Prvi slučaj modelirat ćemo preoblikom definiranom na sljedeći način:

$$pca_1 = asfx\left\{(s\alpha, r\alpha)\ :\ (s,r,C)\in obzv\cup ozv\cup jmt\cup isg,\ \alpha\in C\right\}. \tag{4.12}$$

Ova preoblika alternira osnovu sukladno definicijama (4.8)–(4.11). Ako niti jedna od alternacija na osnovu nije primjenjiva, preoblika će zakazati. Primjerice $pca_1(vrabc) = \{vrapc\}$, no $pca_1(vranc) = \emptyset$.

U drugome slučaju alternacija je uvjetovana fonemom koji je dio nastavka ili sufiksa. Nakon što se takav nastavak veže na osnovu, alternaciju je potrebno provesti unutar

znakovnog niza. To se u modelu može ostvariti zamjenom infiksa znakovnoga niza, na sljedeći način:

$$pca_2 = aifx\left\{(s\alpha, r\alpha) \ : \ (s, r, C) \in obzv \cup ozv \cup jmt \cup isg,\ \alpha \in C\right\}. \tag{4.13}$$

Primjerice, preoblika *svat→svadba* može se definirati kao $pca_2 \circ sfx(ba)$.

Valja primijetiti da je preobliku pca_2 potrebno primijeniti nakon, a preobliku pca_1 prije sufiksacije. Preoblika pca_1 također bi se mogla definirati kao zamjena infiksa, ali to nije preporučljivo budući da se time može povećati višeznačnost preoblike (infiks koji se zamjenjuje ili kojim se zamjenjuje može se u znakovnom nizu pojavljivati na više mjesta).

Preoblike fonološki uvjetovane alternacije mogu zakazati ako se ne ostvari određena fonološka okolina. Zbog toga će svaka takva preoblika biti definirana kao opcionalna pomoću operatora *try*. S druge strane, dvostrukosti, odnosno slučajevi kod kojih je provedba fonološki uvjetovane alternacije opcionalna, modelirani su operatorom *opt*.

Opisane se preoblike u modelu kod flektivnih uzoraka upotrebljavaju na jedan, a kod derivacijskih uzoraka na drugi način. Kod flektivnih uzoraka preoblika pca_1 korištena je samo ondje gdje je ta alternacija očekivana, a to je kod preobličavanja neproširene osnove. Kod derivacijskih uzoraka fonološka alternacija implicitno je ugrađena u model, na način da je preoblika sufiksacije kombinirana s preoblikom za fonološku alternaciju:

$$sfx'(s) \quad = \quad try(pca_2) \circ sfx(s) \circ try(pca_1)\,. \tag{4.14}$$

Ovime se osigurava provođenje fonološke alternacije kraja osnove (prije sufiksacije) te ujedno provođenje fonološke alternacije na granici osnove i sufiksa (nakon sufiksacije).

4.2 Flektivni uzorci

Flektivnu sastavnicu modela sačinjavaju flektivni uzorci (skup $\mathcal{F}$) koja su, uz neka odstupanja opisana u nastavku, modelirana prema (Barić *et al.*, 2005). Od 93 flektivna uzorka, koliko ih je u modelu ukupno definirano, 48 ih je imeničkih, 32 glagolskih i 13 pridjevskih. Ovi uzorci pokrivaju gotovo sav flektivni sustav morfologije hrvatskog jezika; modelom nisu obuhvaćene jedino rijetke iznimke i nepravilnosti. U nastavku je dan sažet opis modeliranja imeničkih, glagolskih i pridjevskih flektivnih uzoraka, dok

je konkretna programska izvedba tih uzoraka izložena u dodatku 3.2.4.

4.2.1 Imenički uzorci

Imeničkim uzorcima obuhvaćena je sklonidba imenica nastavaka vrste *a* (imenice muškog i srednjeg roda), vrste *e* (uglavnom imenice ženskog roda) i vrste *i* (imenice ženskog roda).[3] Sklonidba imenica definirana je u (Barić *et al.*, 2005) pomoću ukupno 31 flektivnih uzoraka. U većini slučajeva jedan flektivni uzorak gramatike odgovara jednome flektivnom uzorku modela, no ponekad je jedan uzorak gramatike predočen dvama ili više uzorcima modela.

Imenički flektivni uzorci modelirana su u skladu sa sljedećim načelima:

- Sve preoblike iskazane u odnosu na (neproširenu i nealterniranu) obličnu osnovu kao ishodišni oblik.
- Dvama flektivnim uzorcima modelirani su uzorci gramatike kod kojih postoje alternante nastavaka (npr. uzorak koji u jednoj od dvije varijante u množini ima umetak *-ev-* modeliran je dvama flektivnim uzorcima).
- Dva uzorka gramatike koji se međusobno razlikuju samo po tome što se u jednome provodi morfološki uvjetovana alternacija osnove, modelirana su također dvama zasebnim flektivnim uzorcima. Zbog toga su preoblike kojima se ostvaruju morfološki uvjetovane alternacije (sibilarizacija, palatalizacija, jotacija i proširenje osnove) modelirane kao obvezatne (dakle u tim slučajevima nije korišten operator *try*).
- Iznimku od gornjeg pravila predstavlja alternacija refleksa jata *ije/je* i *ije/e*, koja je modelirana kao uvjetna u smislu operatora *try*.
- Alternacija refleksa jata iza suglasnika *r*, kod koje postoji dvostrukost (npr. *brijeg→bregovi/brjegovi*), modelirana je preoblikom $try(jat_1 \mid jat_2)$.
- Umetci između osnove i nastavka (umetci za dugu množinu *-ev-* i *-ov-* te umetak *-in-* za jedninu kod imenica sa skraćenom množinom) modelirani su funkcijom *sfx*, dakle kao sufiksi koji se osnovi dodaju prije obličnoga nastavka.[4]
- Preoblike za duge množine (umetci *-ov-* i *-ev-*) definirane su u odgovarajućim flektivnim uzorcima kao opcionalne operatorom *opt*.

[3] Vrsta nastavka određena je nastavkom u genitivu jednine.

[4] Na sličan su način, a temeljem zaključka da pokazuju visok stupanj pravilnosti, množinski umetci tretirani i u (Tadić, 1994).

- Fonološki uvjetovana alternacija modelirana je samo kod onih uzoraka kod kojih postoji proširenje osnove, i to pomoću $try(pca_1)$, osim ako je njeno provođenje opcionalno (npr. *zadatak – zadatci/zadaci*), kada je modelirana pomoću $opt(pca_1)$.

- Ako je u flektivnom uzorku moguća dvostrukost uslijed opcionalnosti alternacija, bilo morfološki uvjetovanih (npr. *guska – gusci/guski*) ili fonološki uvjetovanih (npr. *sudac – sudče/suče*), onda je taj uzorak u modelu predstavljem dvama flektivnim uzorcima: jednim kod kojeg nema dvostrukosti i drugim kod kojeg je dvostrukost modelirana operatorom *opt*.

- Sklonidba imenica koje imaju samo množinu (*pluralia tantum*) modelirana je zasebnim flektivnim uzorcima koje definiraju preoblike samo za množinu. To nije učinjeno za zbirne imenice, koje imaju samo jedninu (*singularia tantum*), pa model za takve imenice generira neispravne oblike množine (u kontekstu morfološke normalizacije to međutim nije problematično).

4.2.2 Glagolski uzorci

Glagolskim flektivnim uzorcima obuhvaćeno je sprezanje glagola po gramatičkim kategorijama lica, načina, vremena, broja i roda. Ostale glagolske gramatičke kategorije (uz iznimku kategorije vida koja je obuhvaćena tvorbenom sastavnicom modela) modelom nisu obuhvaćene. Motivirana činjenicom da će model biti korišten za morfološku normalizaciju, načinjena su još i sljedeća pojednostavljenja. Unutar kategorija vremena i načina izostavljeni su složeni glagolski oblici. Ti se oblici sastoje od jednostavnih glagolskih oblika i pomoćnog glagola ili zamjenice, koji se pri pretraživanju informacija i dubinskoj analizi teksta ionako zanemaruju. Nadalje, izostavljeni su oblici za aorist i imperfekt. Ti su oblici razmjerno rijetki za vrstu teksta kojoj je sustav morfološke normalizacije namijenjen. Također su izostavljeni oblici glagolskog pridjeva trpnog budući da su oni u potpunosti obuhvativi pridjevskom paradigmom.

Flektivnim je uzorcima modelirano sprezanje svih šest vrsta glagola, uključivo svih razreda prve, treće i pete vrste. Nepravilni glagoli nisu modelirani. Sve su preoblike iskazane u odnosu na prezentsku osnovu (osnova koja se dobiva odbacivanjem nastavka u 3. licu jednine).

4.2.3 Pridjevski uzorci

Pridjevskim flektivnim uzorcima obuhvaćena je sklonidba pridjeva određenog i neodređenog vida te poredba (komparacija). Kako bi se što je više moguće ograničio broj

uzoraka, modeliranje pridjeva temeljeno je na konceptu poduzoraka izloženom u dijelu 3.2.4. Za sklonidbu su korištena četiri poduzorka: dva za neodređene oblike i dva za određene oblike. Od toga se dva poduzorka koriste na obličnoj osnovi koja završava na nepčani suglasnik, a dva na osnovi koja završava na nenepčani suglasnik. Za duže nastavke, koji postoje samo u nekim padežima, definirane su u poduzorcima opcionalne preoblike pomoću operatora *opt*.

Na temelju ovih poduzoraka definirano je 13 flektivnih uzoraka, koji se međusobno razlikuju već prema tome koje poduzorke koriste i na koji način preobličuju obličnu osnovu za tvorbu kompartiva i superlativa. Budući da kod posvojnih pridjeva u pravilu nema poredbe, to su ti pridjevi modelirani flektivnim uzorcima kod kojih su te preoblike izostavljene.[5] Također su posebno definirani flektivni uzorci za pridjeve koji imaju oblike samo određenog ili samo neodređenog vida. Gdje god je to moguće, primjenjivost takvih uzoraka ograničena je uvjetom na završetak obličnе osnove. Na primjer, pridjevi na *-ći*, odnosno pridjevi čija osnova završava na *-ć*, imaju oblike samo određenog vida, pa je taj uvjet ugrađen u uzorke koji izvode samo određene oblike. S druge strane, odgovarajući uvjeti kojima bi se ograničila primjenjivost onih uzoraka koji izvode i određene i neodređene oblike u modelu nisu definirani.[6]

4.2.4 Morfosintaktički opisi

Flektivni su uzorci, sukladno (3.18), prošireni morfosintaktičkim opisima prema normi MULTEX-East (Erjavec *et al.*, 2003). Normom MULTEXT-East propisano je kako se vrijednosti različitih morfosintaktičkih kategorija kodiraju u jedan znakovni niz. Vrijednost svake kategorije prikazana je jednim znakom na točno određenom položaju u znakovnome nizu. Kod imenica, na primjer, prvi znak je *N*, dok su drugim mjestima redom kodirane sljedeće gramatičke kategorije: značenje (opće ili vlastito), rod, broj i padež. Na primjer, morfosintaktički opis oblika *vojniče* je znakovni niz *Ncmsv*. Iz tog znakovnoga niza možemo zaključiti da je riječ o obliku opće imenice muškog roda (*Ncm*) u vokativu (*v*) jednine (*s*).

Normom MULTEXT-East propisano je da se vrijednosti onih kategorija koje za određeni oblik (ili određeni jezik) nisu primijenjive zamijenjuju crticom. U modelu se crtica koristi i na mjestima vrijednosti onih kategorija koje nisu razlučive na isključivo

[5] Od triju vrsta pridjeva – opisnih, gradivnih i posvojnih – kompariraju se u pravilu samo opisni pridjevi. No budući da je ovdje riječ o podjeli prema značenju, gradivne pridjeve u modelu nije moguće razlikovati od opisnih.

[6] U načelu su uvjeti primjenjivosti flektivnih uzoraka definirani prema eksplicitno navedenim uvjetima u (Barić *et al.*, 2005). U modelu nisu definirani implicitni uvjeti, tj. uvjeti koji (eventualno uz neke dodatne pretpostavke) logički slijede iz eksplicitnih uvjeta.

morfološkoj razini. Tako je, primjerice, morfosintaktički opis za oblik *vojnik* u modelu *N-msv*, budući da kategorija značenja ne proizlazi iz pripadnosti imenice određenom flektivnom uzorku.

4.3 Derivacijski uzorci

Tvorbenu sastavnicu modela (skup $\mathcal{D}$) sačinjava skup od 244 derivacijskih uzoraka. Uzorcima je opisana sufiksalna tvorba imenica, glagola i pridjeva, i to tvorba između svih devet parova ovih vrsta riječi. Kao što je ranije napomenuto, prefiksalna je tvorba izostavljena budući da uglavnom nije primjenjiva u morfološkoj normalizaciji. Uzorci slijede opise iz (Barić *et al.*, 2005), s time da je definirano i nekoliko dodatnih uzoraka koji ondje nisu bili navedeni, što je utvrđeno naknadno analizom učinkovitosti morfološke normalizacije. Programska izvedba derivacijskih uzoraka dana je u dodatku 3.2.9.

Sukladno (3.28), derivacijski uzorak definiran je u odnosu na kategoriju polazne riječi i izvedenice, a te su kategorije definirane skupovima flektivnih uzoraka. Model razlikuje sljedećih sedam kategorija: (1) imenice muškog roda, (2) imenice ženskog roda, (3) imenice srednjeg roda, (4) posvojni pridjevi, (5) opisni ili gradivni pridjevi, (6) glagoli na *-ći* i (7) glagoli na *-ti* (v. dodatak B.2.4).

Derivacijski uzorci grupirani su, uz neka pojednostavljenja, prema tvorbenom značenju izvedenice u značenjske kategorije sukladno (Barić *et al.*, 2005). Pregled derivacijskih uzoraka prema značenjskim kategorijama dan je u tablici 4.2. Kao što je iz tablice vidljivo, većina uzoraka odnosi se na izvođenje imenica i pridjeva. Zbog tvorbene višeznačnosti, odnosno činjenice da jedan te isti sufiks može izražavati različite značenjske kategorije ovisno o tvorbenoj osnovi, neka derivacijski uzorci nalaze se u više kategorija.

Kako je opisano u odjeljku 3.2.10, tvorbena osnova ovisi o morfološkoj vrsti riječi. Kod imenica i pridjeva tvorbena je osnova identična obličnoj, dok kod glagola tvorbena osnova može biti prezentska ili infinitivna osnova. Budući da su preoblike kod derivacijskih uzoraka iskazane u odnosu na obličnu osnovu, to je kod modeliranja tvorbe motivirane glagolima u nekim slučajevima za jedan tvorbeni uzorak bilo potrebno koristiti više različitih preoblika. Takvi su uzorci, radi preglednosti, modelirani pomoću više derivacijskih uzoraka.

Morfološki uvjetovane alternacije modelirane su kod derivacijskih uzoraka kao uvjetne u smislu operatora *try*. To je zato što jedan te isti uzorak mora biti primjenjiv i na

one tvorbene osnove na koje sama alternacija nije primjenjiva. Na primjer, tvorba *kazniti* → *kažnjavati* provodi se sufiksacijom *-avati* i alternacijom suglasničkog skupa *zn/žnj* u tvorbenoj osnovi. Međutim, kod tvorbe *odobriti* → *odobravati*, alternacija suglasničkog skupa nije primjenjiva. Stoga je preoblika definirana kao $sfx(avati) \circ try(acg)$.

Kod derivacijskih se uzoraka pojavljuju još dva slučaja modeliranja opcionalnosti alternacija. Prvi je slučaj modeliranje dvostrukosti, i to uglavnom kod alternacije refleksa jata. Na primjer, dvostrukost se pojavljuje u tvorbi *brijeg* → *bregovit/bregovit*, pa je odgovarajući derivacijski uzorak definiran s preoblikom $sfx(ovi) \circ try(jat_1 \mid jat_2)$. Drugi slučaj je modeliranje nekonzistentnosti primjene pojedinih fonoloških alternacija. Na primjer, pri tvorbi imenica sa sufiksom *-ar* koje su motivirane imenicama, osim alternacije *ije/je* (npr. *cvijeće* → *cvjećar*), tvorbena osnova nekad je jotirana (npr. *tvornica* → *tvorničar*), no nekad nije (npr. *biblioteka*→*bibliotekar*). Preoblika je u tom slučaju definirana kao $sfx(ar) \circ try(jat_1) \circ opt(jot)$.

Tablica 4.2: Modelirani derivacijski uzorci sufiksalne tvorbe hrvatskoga jezika.

Skupina	Značenjska kategorija izvedenice	Broj uzoraka	Primjer
N-1	Imenica muškog roda za vršitelja radnje	21	*banka* → *bankar*
N-2	Imenica muškog roda kao nositelj osobine	5	*sretan* → *sretnik*
N-3	Imenica muškog roda kao sljedbenik	3	*Franjo* → *franjevac*
N-4	Imenica za žensku osobu	11	*rukometaš* → *rukometašica*
N-5	Imenica za mušku i žensku osobu	5	*izdati* → *izdajica*
N-6	Etici	11	*Varaždin* → *Varaždinac*
N-7	Imenice za životinje i bilje	6	*otrovan* → *otrovnica*
N-8	Imenice za stvari	11	*mijenjati* → *mjenjač*
N-9	Mjesne imenice	9	*cigla* → *ciglana*
N-10	Apstraktne imenice	18	*prijatelj* → *prijateljstvo*
N-11	Radne imenice (uključivo glagolske imenice)	24	*čuvati* → *čuvanje*
N-12	Umanjenice i uvećanice	19	*orah* → *oraščić*
N-13	Zbirne imenice	7	*radnik* → *radništvo*
N-14	Ostale značenjske skupine imenica	6	*brod* → *brodarina*
A-1	Opisni pridjevi	35	*mrak* → *mračan*
A-2	Posvojni pridjevi	19	*djed* → *djedov*
A-3	Glagolski pridjevi trpni	9	*spasiti* → *spašen*
V-1	Glagoli s promjenom u vidu	12	*baciti* → *bacati*
V-2	Deminutivni i pejorativni glagoli	6	*govoriti* → *govorkati*
V-3	Glagoli iz imenica	5	*večera* → *večerati*
V-4	Glagoli iz pridjeva	2	*sitan* → *sitniti*

Poglavlje 5

Akvizicija flektivnoga morfološkog leksikona

U ovome poglavlju opisan je postupak akvizicije flektivnoga morfološkog leksikona iz neoznačenog korpusa. Postupak se temelji na ranije opisanom generativno-redukcijskom flektivnome morfološkom modelu. Zbog višeznačnosti gramatike, morfološka normalizacija koja bi se temeljila na izravnoj uporabi tog modela ne bi postizala zadovoljavajuću razinu preciznosti. Glavni je problem što se na jedan te isti oblik riječi može primijeniti veći broj flektivnih uzoraka. Akvizicijski postupak, s druge strane, oslanja se na dodatnu informaciju o učestalosti oblika riječi u korpusu kako bi iz tog istog korpusa pribavio flektivni leksikon kvalitete dovoljno dobre za morfološku normalizaciju. Normalizacija pomoću tako pribavljenog leksikona znatno je preciznija od one koja bi se ostvarila izravnom uporabom redukcijskog morfološkog modela.

Poglavlje je organizirano na sljedeći način. U prvome odjeljku ukratko su razmotreni mogući pristupi akviziciji leksikona temeljeni na uporabi flektivnog morfološkog modela. U drugome odjeljku opisan je postupak akvizicije flektivnoga leksikona iz neoznačena korpusa. Postupak se temelji na mjerama za ocjenu vjerodostojnosti pribavljenih lema i njima pridruženih flektivnih uzoraka, i te su mjere posebno razmatrane u trećemu odjeljku. U posljednjem odjeljku opisano je eksperimentalno vrednovanje akvizicijskog postupka.

5.1 Načini akvizicije leksikona

U ovome radu flektivni morfološki leksikon odnosi se na zbirku sastavljenu od riječi i njima pridruženih oblika. Temeljem morfološkog modela opisanog u prethodnom

poglavlju, oblike neke riječi moguće je sažeto prikazati dvojkom sastavljenom od leme i odgovarajućeg flektivnog uzorka, $(l, f) \in \mathcal{S} \times \mathcal{T}$.[1] Oblici pridruženi lemi l mogu se tada, temeljem uzorka f, izvesti pomoću funkcije *lWfs* definirane izrazom (3.47). Dvojku (l, f), sastavljenu od leme i flektivnog uzorka, nazivamo u nastavku *LU-parom*. LU-parovi predstavljaju osnovne gradivne elemente flektivnog morfološkog leksikona korištenog u ovome radu.

Postupak akvizicije flektivnog leksikona svodi se na pribavljanje LU-parova temeljem nekog raspoloživog jezičnog resursa. Ako su leme unaprijed poznate, postupak se svodi na određivanje ispravnoga flektivnog uzorka za svaku pojedinu lemu. Ako leme nisu unaprijed poznate, potrebno je također odrediti koji oblik predstavlja ispravnu lemu riječi. Tako možemo razlikovati sljedeće slučajeve:

(a) Raspoloživ je popis lema te potpun ili djelomičan popis njima pridruženih oblika;

(b) Raspoloživ je potpun ili djelomičan popis oblika za svaku lemu, ali same leme nisu poznate;

(c) Raspoloživ je popis lema te eventualno neke dodatne informacije o svakoj lemi (poput morfološke vrste, morfološke kategorije, značenjske skupine ili sl.);

(d) Raspoloživi su samo oblici riječi.

Slučaj (a) je najjednostavniji, no u praksi nije od interesa. Naime, popis lema i njima pridruženih oblika zapravo već sačinjava morfološki leksikon, pa se kao takav može izravno koristiti za morfološku normalizaciju. Akvizicija u tom slučaju ima smisla jedino kao način konverzije između dvaju leksikona temeljenih na različitim morfološkim modelima. Slučaj (b) sličan je slučaju (a), s tom razlikom da leme nisu posebno istaknute, pa je svaki oblik potrebno razmotriti kao moguću lemu. U oba slučaja međutim oblici svake riječi grupirani su zajedno, a to znatno olakšava određivanje ispravnog flektivnog uzorka.

Zanimljiviji su u praksi slučajevi (c) i (d). Slučaj (c) može biti situacija u kojoj je raspoloživ strojno čitljiv rječnik (engl. *machine readable dictionary*, MRD) koji, osim popisa lema, može sadržavati i dodatne informacije kao što su vrsta riječi, rod imenice, genitivni oblik ili sl. Akviziciju u tom slučaju olakšava činjenica da je lema unaprijed poznata, a dodatne informacije mogu se iskoristiti kako bi se skup mogućih flektivnih uzoraka dodatno ograničio.

[1] U ishodište fleksije umjesto leme može se dakako smjestiti i oblična osnova. U nastavku međutim, sukladno razmatranjima iznesenima u odjeljku 3.2.5, pretpostavljamo da je u ishodištu fleksije lema, te shodno tome da su unosci morfološkog leksikona sastavljeni od lema.

Slučaj (d) od navedenih je slučajeva najsloženiji utoliko što postupak raspolaže s najmanje informacija: oblici riječi nisu grupirani zajedno, niti je poznato koji oblici mogu biti leme riječi. Oblici riječi mogu se onda crpsti iz korpusa, a statistička informacija o pojavljivanju pojedinih oblika riječi može se iskoristiti kao dodatna informacija pri odabiru ispravne leme i ispravnoga uzorka. Algoritam opisan u nastavku namijenjen je upravo ovom najopćenitijem slučaju akvizicije leksikona.

5.2 Akvizicijski algoritam

Akvizicijski algoritam temelji se na dvosmjernoj primjeni generativno-redukcijskoga morfološkog modela. Primjenom redukcijskoga smjera na oblik iz korpusa dobivaju se svi mogući LU-parovi koji su pridruženi tom obliku, dok se primjenom generativnog smjera na svaki tako dobiveni LU-par izvode svi pripadni oblici. Temeljem informacije o učestalosti pojavljivanja tih oblika u korpusu moguće je procijeniti lingvističku vjerodostojnost LU-parova i u leksikon uključiti samo one parove koji su dovoljno vjerodostojni, odnosno idealno samo one koje su lingvistički ispravni.

5.2.1 Korpus i leksikon

U opisu koji slijedi korpus je prikazan kao zbirka (multiskup) W definiran nad skupom znakovnih nizova $\mathcal{S}$, tj. $W \in \mathcal{S}^{\infty}$. Neka '$\#(w, W)$' označava broj pojavljivanja riječi w u korpusu W. Radi jednostavnosti, 'W' ćemo koristiti kao oznaku za zbirku i za odgovarajući skup; značenje će biti razaberivo iz konteksta.

Flektivni leksikon $\mathcal{L}$ prikazan je kao skup unosaka, $\mathcal{L} = \{E_i\}$. Svaki se unosak E_i u idealnom slučaju sastoji od samo jednoga LU-para. Međutim, treba predvidjeti i slučajeve kod kojih zbog višeznačnosti gramatike neće biti moguće izdvojiti jedan ispravan LU-par. Zbog toga je unosak leksikona E_i definiran kao skup LU-parova, $E_i = \{(l, r)_j\} \subset \mathcal{S} \times \mathcal{F}$. Unosak leksikona koji sadržava više od jednog LU-para nazivat ćemo *nerazriješenim*.

5.2.2 Opis algoritma

Postupak akvizicije leksikona formaliziran je funkcijom $acquireLexicon : \mathcal{S}^{\infty} \rightarrow \mathcal{L}$ definiranom tablicom 5.1. Argument funkcije je korpus W_0, a povratna vrijednost je skup unosaka koji sačinjavaju flektivni leksikon. Akvizicija se provodi iterativno: u svakome koraku leksikonu se pridodaje jedan unosak pomoću funkcije *acquire*. Radi preglednosti, funkcija *acquire* definirana je kao slijed naredbi pridruživanja, a varijable w, LF,

Tablica 5.1: Algoritam za akviziciju flektivnoga leksikona iz neoznačenog korpusa.

$acquireLexicon(W_0) = acquire(W_0)$

gdje

1 : **funkcija** $acquire(W)$ **vraća** skup unosaka leksikona $\{E_i\}$

2 : **ako** $W \neq \emptyset$ **onda**

3 : $w = (any \circ wfpref)(W)$

4 : $LF = \left\{(l, f) \in lm_{\mathcal{F}}(w) \ : \ accept(W, l, f)\right\}$

5 : **ako** $LF \neq \emptyset$ **onda**

6 : $blf = \underset{(l,f)\in LF}{\operatorname{argmax}}\, score(W_0, l, f)$

7 : $E = \{lf \in LF : tie(W_0, blf, lf)\}$

8 : $WFS = \bigcup_{(l,f)\in E} lWfs(l, f)$

9 : **vrati** $\{E\} \cup acquire(W \setminus WFS)$

10 : **inače**

11 : **vrati** $acquire(W \setminus \{w\})$

12 : **inače**

13 : **vrati** $\emptyset$

---

blf, *BLF*, *WFS* i *E* korištene su za pohranu međurezultata. Iteracija je izvedena kao rekurzija nad multiskupom W. Budući da se u svakom rekurzivnom pozivu (retci 9 i 11) iz korpusa W uklanja barem jedan oblik, korpus se u svakoj iteraciji zapravo smanjuje i akvizicijska funkcija u konačnici terminira (redak 14). U nastavku koristimo 'W_0' kako bismo označili potpuni korpus (tj. početni multiskup na koji je funkcija *acquire* početno primijenjena), a 'W' za korpus koji se u svakoj iteraciji smanjuje.

Akvizicija leksikona može se razdijeliti u tri slijedna koraka: pribavljanje LU-parova (retci 3 i 4), ocjenjivanje LU-parova (redak 6) te oblikovanje unoska leksikona (retci 7 i 8). Ovi koraci detaljno su opisani u nastavku.

Korak 1: Pribavljanje LU-parova

Pribavljanje LU-parova započinje odabirom nekog oblika w iz korpusa W (redak 3 u tablici 5.1). Budući da pribavljanje jednog LU-para može spriječiti kasnije pribavljanje nekog drugog LU-para, redoslijed odabira oblika može imati utjecaja na konačnu kvalitetu leksikona. Zbog toga je definirana funkcija $wfpref : \wp(\mathcal{S}) \to \wp(\mathcal{S})$ kojom se skup nepribavljenih oblika iz korpusa W najprije sužava na one oblike koje preferiramo, da bi se zatim iz tog skupa proizvoljno odabrao jedan oblik. Ovdje i u nastavku rada odabir proizvoljnog elementa iz nepraznog skupa X formaliziran je funkcijom odabira $any : \wp(X) \to X$. Optimalna definicija funkcije *wfpref* u smislu kvalitete leksikona razmotrena je u dijelu 5.4.2.

Na odabrani oblik w primjenjuje se redukcijski smjer morfološkog modela, odnosno oblik se lematizira uporabom funkcije *lm* definirane izrazom (3.15). Tako se dobivaju svi LU-parovi kojima oblik w može pripadati (skup LF u retku 4). U idealnom slučaju – i to pod uvjetom da oblik w nije homograf – funkcija *lm* rezultira samo jednim LU-parom. Međutim, zbog višeznačnosti gramatike, takvi će slučajevi u praksi biti iznimno rijetki. Na primjer, čak i ako se skup flektivnih uzoraka sastoji od samo dva uzorka, $\mathcal{F} = \{f_{N1}, f_{A1}\}$, gdje su f_{N1} i f_{A1} definirani s (3.19) odnosno (3.23), vrijedit će $lm_{\mathcal{F}}(brže g) = \{(brz, f_{A1}), (brže g, f_{A1}), (brz, f_{N1})\}$. LU-parovi (l, f) kod kojih je uzorak f primjenjiv na lemu l, ali je par međutim lingvistički neispravan – bilo zato što l nije ispravna lema ili zato što f nije ispravan flektivni uzorak za lemu l – nazivat ćemo *lažnim LU-parom*. U ovome primjeru, lažni LU-parovi su $(brže g, f_{A1})$ i (brz, f_{N1}), dok je LU-par (brz, f_{A1}) lingvistički ispravan.

Problem pri akviziciji leksikona predstavlja činjenica da svaki korpus iz ovog ili onog razloga neminovno sadržava i određeni broj jezično nepravilnih oblika. Takvi oblici rezultiraju lažnim LU-parovima koji, ako ih se pribavi, u konačnici mogu narušiti

kvalitetu leksikona. Kako bi se to spriječilo, moguće je postaviti minimalni uvjet koji svaki LU-par mora zadovoljiti, a da bi taj uopće bio prihvatljiv. Taj je uvjet formaliziran funkcijom (predikatom) $accept : \mathcal{S}^{\infty} \times \mathcal{S} \times \mathcal{F} \rightarrow \{\top, \bot\}$ (redak 4), definiranom na sljedeći način:

$$accept(W, l, f) = |lWfs(f, l) \cap W| \geq \alpha \ \wedge \ length(l) \geq 3\,. \tag{5.1}$$

Kako bi LU-par (l, r) bio prihvatljiv, on mora generirati najmanje α oblika koji su ovjereni u korpusu. Tu se međutim u obzir uzimaju samo oni oblici koji još nisu pribavljeni (multiskup W u retku 4 jest korpus koji sadržava samo još nepribavljene oblike). Drugim riječima, barem α oblika mora pripadati isključivo paru (l, r), a da bi taj bio pribavljen. Na taj način pribavljanje ispravnog LU-para sprječava kasnije pribavljanje lažnog LU-para koji s njime dijeli iste oblike. Vrijednosti parametra α ne smije biti prevelika, kako se ne bi spriječilo pribavljanje LU-parova s manjim brojem različitih oblika u korpusu, a niti LU-parova s homografnim oblicima, koji su po definiciji dijeljeni između više LU-parova. Također se iziskuje da lema pribavljenog LU-para bude dulja od tri znaka; leme kraće od toga su, uz svega nekoliko iznimaka, sigurno neispravne.

Korak 2: Ocjenjivanje LU-parova

U drugome koraku od LU-parova koji nisu odbačeni odabiru se oni lingvistički najvjerodostojniji. Ako međutim vrijedi $LF = \emptyset$, onda to znači da za oblik w niti jedan LU-par nije prihvatljiv i akvizicija se nastavlja s oblikom w izuzetim iz korpusa (redak 11 u tablici 5.1). U suprotnome, lingvistička vjerodostojnost LU-para ocjenjuje se temeljem informacije o učestalosti pojave oblika tog LU-para u korpusu. Pritom pretpostavljamo da će oblici lingvistički ispravnih LU-parova u korpusu biti znatno učestaliji nego oblici lažnih LU-parova.

Lingvistička ispravnost LU-para (l, f) procjenuje se pomoću *mjere ocjene* formalizirane funkcijom $score : \mathcal{S}^{\infty} \times \mathcal{S} \times \mathcal{F} \times \rightarrow \mathbb{R}$. Najjednostavnija mjere ocjene jest ona koja prebrojava koliko je oblika dotičnog LU-para doista ovjereno u korpusu:

$$score_0(W, l, f) = |lWfs(f, l) \cap W|\,, \tag{5.2}$$

dok su nešto složenije mjere ocjene opisane u odjeljku 5.3. Mjera ocjene primjenjuje se na svaki LU-par iz skupa LF dobivenog u prethodnome koraku kako bi se odredio LU-par s najvišom ocjenom (vrijednost *blf* u retku 6 tablice 5.1). Valja primijetiti da, ovisno

o razlučivosti funkcije ocjene, takvih LU-parova može biti više; ovdje pretpostavljamo da funkcija argmax vraća bilo koji najbolje ocijenjeni LU-par.

Korak 3: Dodavanje unoska leksikonu

U trećemu koraku leksikon se proširuje novim unoskom E (redak 7 tablice 5.1). Skup E sastoji se od LU-parova koji su u prethodnome koraku najbolje ocijenjeni. U općenitome slučaju takvih parova može biti više, pa je među njima potrebno probrati one koji će sačinjavati jedan nerazriješeni unosak leksikona. To je u algoritmu ostvareno pomoću funkcije (predikata) $tie : \mathcal{S}^\infty \times (\mathcal{S} \times \mathcal{F}) \times (\mathcal{S} \times \mathcal{F}) \to \{\top, \bot\}$ koja uspoređuje dva LU-para i vrednuje se istinito ako oba LU-para treba smjestiti u isti unosak leksikona. Funkcija je definirana na sljedeći način:

$$\begin{aligned} tie(W, (l_1, f_1), (l_2, f_2)) \quad &\iff \quad score(W, l_1, f_1) = score(W, l_2, f_2) \\ &\wedge \quad lWfs(f_1, l_1) \cap W = lWfs(f_2, l_2) \cap W\,. \end{aligned} \tag{5.3}$$

tj. LU-parovi će sačinjavati isti unosak leksikona samo ako su potpuno identično ocijenjeni i ako im je identičan skup oblika ovjerenih u korpusu. Tada je naime najizglednije da se svi ti LU-parovi odnose na isti leksem, ali da su svi osim jednoga lažni. Takve LU-parove, budući da su identično ocijenjeni, moramo tretirati kao jednako vjerodostojne i zbog toga oni u leksikonu trebaju formirati jedan nerazriješeni unosak. S druge strane, LU-parovi koji ispunjavaju prvi, ali ne i drugi uvjet vjerojatno se odnose na različite lekseme i bit će pribavljeni u nekoj od idućih iteracija kada bude odabran neki drugi, njima pridruženi oblik, pod uvjetom da i tada zadovolje predikat *accept*. Skup E pridodaje se leksikonu kao novi unosak, a akvizicija se nastavlja nad korpusom W iz kojega su prethodno izuzeti oblici iz skupa *WFS* (redak 9 u tablici 5.1).

Očito je da će nerazriješeni unosci smanjivati ukupnu lingvističku kvalitetu leksikona. Postavlja se međutim pitanje hoće li zbog toga leksikon biti manje pogodan za morfološku normalizaciju, odnosno hoće li nerazriješeni unosci narušavati učinkovitost morfološke normalizacije. Pokazuje se da to ovisi o tome što se pri morfološkoj normalizaciji uzima kao norma. Ako norma odgovara unoscima leksikona, a ne pojedinačnim LU-parovima od kojih su unosci sastavljeni, potpuno je nebitno jesu li ti unosci razriješeni. Za morfološku je normalizaciju ključno samo to da svi oblici riječi budu povezani s jednim unoskom, kako god taj unosak izgledao. Budući da je to upravo način na koji će leksikon biti korišten, unoske leksikona u načelu nije potrebno razrješavati.

5.3 Mjere ocjene LU-parova

Kvaliteta pribavljenog leksikona uvelike ovisi o odabiru mjere ocjene LU-parova. Mjera (5.2), koja LU-par ocjenjuje temeljem broja ovjerenih oblika u korpusu, pokazat će se u nekim slučajevima nedovoljno preciznom. Stoga su u nastavku predložene i neke druge mjere temeljene na različitim intuitivnim pretpostavkama.

5.3.1 Osnovne mjere

Sljedeće mjere temelje se na prebrojavanju ovjerenih oblika u korpusu. Mjera $score_1$ izračunava ukupnu učestalost oblika para (l, r) u korpusu W:

$$score_1(W, l, f) = \sum_{w \in lWfs(f,l)} \#(w, W_0)\,. \tag{5.4}$$

Intuitivno, mjera $score_1$ robusnija je od mjere $score_0$ budući da se temelji na učestalosti oblika, a ne broju različitih oblika. Međutim, nedostatak obaju mjera jest taj što u obzir ne uzimaju ukupan mogući broj oblika nekog LU-para, pa tako lažni LU-par može biti visoko ocijenjen temeljem malog broja oblika ovjerenih u korpusu. Opravdano je očekivati da je za neki LU-par broj ovjerenih oblika u korpusu donekle razmjeran broju oblika koji su izvedivi iz dotičnog LU-para. Tako, primjerice, očekujemo da će LU-par (brz, f_{A1}) imati u korpusu više ovjerenih oblika nego LU-par $(vojnik, f_{N1})$, budući da različitih oblika pridjeva ima više nego različitih oblika imenice. Ovu ideju obuhvaća mjera $score_2$, koja izračunava udio oblika ovjerenih u korpusu W u ukupnom broju oblika izvedivih iz LU-para (l, r), na sljedeći način:

$$score_2(W, l, f) = \frac{|lWfs(f, l) \cap W|}{|lWfs(f, l)|}\,. \tag{5.5}$$

Ova mjera može donekle uspješno riješiti problem nerazlikovanja imenica od pridjeva. Taj je problem tipičan za hrvatski jezik, budući da mnogi imenički flektivni uzorci predstavljaju u stvari podskup pridjevskih flektivnih uzoraka, ili se pak u velikoj mjeri preklapaju. Na primjer, većina (šest od osam) oblika ispavnog LU-para $(vojnik, f_{N1})$ podskup su oblika lažnog LU-para $(vojnik, f_{A1})$. Ako su u korpusu W sadržani samo oni oblici koji su zajednički obama LU-parovima, onda vrijedi $score_0(W, vojnik, f_{N1}) = score_0(W, vojnik, f_{A1})$, a također i $score_1(W, vojnik, f_{N1}) = score_1(W, vojnik, f_{A1})$, pa lažni LU-par nije moguće razlikovati od ispravnoga. No, vrijedi $score_2(W, vojnik, f_{A1}) \leq score_2(W, vojnik, f_{N1})$, budući da je udio u korpusu ovjerenih oblika uzorka f_{A1} manji od udjela u korpusu ovjerenih oblika uzorka f_{N1}.

5.3.2 Heuristička mjera

Specifičnosti hrvatskog jezika mogu se pri ocjenjivanju LU-parova pokušati uvažiti na način da se definira jezično-specifična heuristička mjera. Pored spomenute nemogućnosti razlikovanja između oblika imenica i pridjeva, problem predstavlja i pridjevski vid, odnosno određivanje ima li pridjev neodređen vid ili nema. Ova razlika utječe na sam odabir leme: ako pridjev nema neodređenog vida, lema je u određenom obliku, inače je u neodređenom obliku. Heuristička mjera ocjene koja uvažava ove specifičnosti, a koja se u ispitivanjima provedenima u (Šnajder *et al.*, 2008) pokazala uspješnom, može se definirati na sljedeći način:

$$score_H(W, l, f) = 10 \cdot score_1(W, l, f) + \beta_1 + \beta_2\,, \tag{5.6}$$

gdje je

$$\beta_1 = \begin{cases} 4 & \text{ako } f \in \mathcal{F}_N\,, \\ 0 & \text{ako } f \in \mathcal{F}_{Aqd}\,, \\ 2 & \text{inače}\,, \end{cases} \qquad \beta_2 = \begin{cases} 1 & \text{ako } l \in W_0\,, \\ 0 & \text{inače}\,. \end{cases} \tag{5.7}$$

Mjera se temelji na mjeri $score_1$ koja izračunava učestalost oblika ovjerenih u korpusu. Ako je ta vrijednost za dva LU-para identična, odlučujuću ulogu imaju vrijednosti β_1 i β_2. Prvom je vrijednošću određen prioritet LU-para u ovisnosti o uzorku. Imeničkim uzorcima (uzorci iz skupa $\mathcal{F}_N$) pridijeljen je najviši prioritet. Ako je dakle LU-par s imeničkim uzorkom ocijenjen jednako kao i LU-par s pridjevskim uzorkom, što znači da im je učestalost oblika ovjerenih u korpusu identična, onda se prednost daje LU-paru s imeničkim uzorkom. Najniži prioritet dan je flektivnim uzorcima koji se odnose na opisne pridjeve bez neodređenog vida (uzorke iz skupa $\mathcal{F}_{Aqd}$). Naime, takvih je uzoraka razmjerno malo, pa se preciznost leksikona može poboljšati tako da se preferiraju ostali pridjevski uzorci. Vrijednost β_2 koristi se kako bi se dodatno povisio prioritet onih LU-parova čije su leme ovjerene u korpusu.

5.3.3 Vjerojatnosne mjere

Do sada opisane mjere ne uzimaju u obzir činjenicu da i između oblika iste riječi postoji razlika u učestalosti. Intuitivno je jasno da će učestalost oblika neke riječi u korpusu zavisiti o morfološkoj kategoriji koju taj oblik (odnosno njegov gramatički morfem) izražava. Tako, primjerice, možemo očekivati da će nominativni oblik neke imenice biti

učestaliji nego vokativni oblik, kao i to da će superlativni oblici nekog pridjeva biti manje učestali nego oblici pozitiva. Zbog toga je opravdano očekivati da će mjere koje na neki način uvažavaju vjerojatnosnu razdiobu oblika uspješnije diskriminirati između ispravnih i lažnih LU-parova. To vrijedi i u onim slučajevima kada se dva LU-para odnose na istu vrstu riječi i izvode identičan skup oblika ovjerenih u korpusu, budući da očekivane razdiobe oblika za takva dva LU-para općenito neće biti identične.

Očekivanu razdiobu oblika flektivnoga uzorka f izrazit ćemo uvjetnom vjerojatnošću $P(x|f)$. Vrijednost $P(x|f)$ je vjerojatnost da oblik izveden pomoću flektivnog uzorka f ima morfosintaktički opis x. Ovdje dakle pretpostavljamo da riječi istog flektivnog uzorka imaju sličnu razdiobu oblika, odnosno pretpostavljamo da na razdiobu oblika riječi više utječe izbor flektivnoga uzorka nego li sama riječ. Vjerojatnost $P(x|f)$ može se procijeniti temeljem vjerojatnosti oblika u nekom reprezentativnom uzorka. Takav se uzorak može izgraditi tako da se najprije napravi uzorak ispravnih LU-parova, a zatim se iz korpusa sakupe podaci o učestalosti oblika svakog pojedinog LU-para. Temeljem korpusa W, najvjerodostojniju procjenu vjerojatnosti oblika w kao oblika LU-para (l, f) možemo izraziti kao:

$$P(w|\, l, f) = \frac{\#(w, W)}{\sum_{w \in lWfs(f,l)} \#(w, W)}\,. \tag{5.8}$$

Označimo uzorak LU-parova sa $\mathcal{U}$, a sa $\mathcal{U}_f$ poduzorak LU-parova čiji je flektivni uzorak jednak f. Vjerojatnost $P(x|f)$ sada možemo izračunati na sljedeći način:

$$P(x|f) = \frac{1}{|\mathcal{U}_f|} \sum_{(l,f) \in \mathcal{U}_f} \sum_{(w,X) \in lWfsMsds(f,l,x)} \frac{P(w|\, l, f)}{|X|}, \tag{5.9}$$

gdje je

$$\begin{aligned} lWfsMsds(f, l, x) \;=\; \Big\{ & (w, X) \;:\; (w, x) \in lWfsMsd(f, l), \\ & X = \{x' \;:\; (w, x') \in lWfsMsd(f, l)\} \Big\}, \end{aligned} \tag{5.10}$$

funkcija koja za zadani LU-par (l, f) i morfosintaktički opis x izvodi skup oblika s tim morfosintaktičkim opisom. Pritom, ako oblik ima više mogućih morfosintaktičkih opisa (tj. ako je oblik unutarnji homograf), onda su svi ti opisi sažeti u jedan skup X. Vjerojatnost $P(x|f)$ računa se dakle pomoću vjerojatnosti oblika $P(w|\, l, f)$ čija je morfosintaktička oznaka x, pri čemu se, u slučaju višeznačnosti, ta vjerojatnost razdjeljuje na sve moguće morfosintaktičke opise. Ukupna vjerojatnost dobiva se uprosječivanjem

nad svim LU-parovima iz poduzorka $\mathcal{U}_f$.

Pouzdanost procjene vjerojatnosti $P(x|f)$ ovisi dakako o veličini korpusa i veličini uzorka $\mathcal{U}$. Realno je međutim očekivati će uzorak $\mathcal{U}$ biti ograničene veličine, budući da ga je potrebno sastavljati ručno. Kako bi se i u takvim slučevima dobile pouzdane vjerojatnosne procjene, uputno je napraviti nešto grublju podjelu uzorka $\mathcal{U}$, primjerice na poduzorke prema morfološkim vrstama. Takva podjela implicira pretpostavku da riječi iste morfološke vrste imaju sličnu razdiobu oblika. Također je potrebno napomenuti da bi se pozudanije vjerojatnosne procjene mogle dobiti prethodnim razrješavanjem unutarnje homografije, no taj pristup ovdje nije razmatran budući da je akvizicijski postupak namijenjen akviziciji iz neoznačenog korpusa.

Temeljem opisanih vjerojatnosti $P(x|f)$ i $P(w|\,l,f)$, u nastavku su definirane tri mjere ocjene LU-parova. Mjera $score_{P1}$ proširuje mjeru $score_1$ na način da učestalost oblika množi s vjerojatnošću njima odgovarajućeg morfosintaktičkog opisa:

$$score_{P1}(W,l,f) = \sum_{(w,x)\in lWfsMsd(f,l)} P(x|f)\cdot \#(w,W_0)\,. \tag{5.11}$$

Intuitivno, mjera daje veći značaj učestalosti onog oblika koji je za dani uzorak vjerojatniji.

Mjera $score_2$ izračunava ukupnu vjerojatnosnu masu koju su, svojim pojavljivanjem u korpusu, ostvarili oblici nekog LU-para:

$$score_{P2}(W,l,f) = \sum_{(w,x)\in lWfsMsd(f,l)} P(x|f)\cdot \max(1,\#(w,W_0))\,, \tag{5.12}$$

Ova mjera predstavlja svojevrsno poopćenje mjera $score_0$ i $score_2$.

Mjera $score_{P3}$ zasniva se na usporedbi očekivane razdiobe oblika (za dani flektivni uzorak) i stvarne razdiobe oblika (za dani LU-par):

$$score_{P3}(W,l,f) = \frac{1}{N}\sum_{(w,x)\in lWfsMsd(f,l)} \min\left(1,\frac{P(w|\,l,f)}{P(x|f)}\right), \tag{5.13}$$

gdje $N = |lWfsMsd(f,l)|$. Intuitivno, ocjena LU-para je to veća što je stvarna razdioba oblika sličnija očekivanoj, odnosno što je omjer $P(w|\,l,f)/P(x|f)$ bliži jedinici. Omjer se ograničava na jedinicu, kako bi mjera bila normirana na jedinični interval, ali i kako bi se zanemarili slučajevi kod kojih $P(w|\,l,f) > P(x|f)$. Takvi slučajevi naime očekivani su za homografne oblike, koji, budući da pripadaju različitim riječima, mogu imati veću učestalost od očekivane.

5.3.4 Kombinacija mjera

Budući da se pojedine mjere temelje na različitim intuitivnim pretpostavkama, neke od njih ima smisla kombinirati i tako možda ostvariti još uspješniju procjenu ispravnosti LU-para. U nastavku su navedene samo neke od mogućih kombinacija, i to one koje su se u preliminarnim eksperimentima pokazale najboljima.

Nedostatak mjera $score_0$ i $score_1$, definiranih izrazima (5.2) i (5.4), jest njihova niska razlučivost. U velikom broju slučajeva, uključivo onim gore spomenutima, događa se da nije moguće razlučiti između ispravnog i lažnog LU-para samo na temelju broja ili učestalosti oblika ovjerenih u korpusu. U takvim slučajevima opravdano je očekivati da će ispravni LU-par imati veći udio oblika ovjerenih u korpusu, što je iskazano mjerom $score_2$ definiranom izrazom (5.5). Uputno je stoga definirati sljedeće kombinacije mjera:

$$score_{02}(W,l,f) = score_0(W,l,f) + score_2(W,l,f)\,, \tag{5.14}$$

$$score_{12}(W,l,f) = score_0(W,l,f) + score_2(W,l,f)\,. \tag{5.15}$$

Mjera $score_2$ je normirana na jedinični interval, pa će dakle do izražaja doći jedino pri usporedbi LU-parova s identičnim brojem odnosno identičnom učestalošću oblika ovjerenih u korpusu.

Na sličan je način moguće kombinirati osnovne mjere s vjerojatnosnim mjerama $score_{P2}$ i $score_{P3}$, koje su obje normirane na jedinični interval. Dvije takve kombinacije koje su se pokazale vrlo uspješnima jesu:

$$score_{1P2}(W,l,f) = score_1(W,l,f) + score_{P2}(W,l,f)\,, \tag{5.16}$$

$$score_{1P3}(W,l,f) = score_1(W,l,f) + score_{P3}(W,l,f)\,. \tag{5.17}$$

5.4 Vrednovanje akvizicijskoga postupka

Kvaliteta morfološkog leksikona dobivenog opisanim akvizicijskim postupkom zavisit će o kvaliteti korpusa nad kojim je provedena akvizicija, potpunosti i ispravnosti morfološkog modela te o parametrima akvizicijskog algoritma. U ovome je dijelu razmotren utjecaj parametara akvizicijskog postupka na kvalitetu pribavljenog leksikona. Opisana su dva eksperimenta. U prvome se eksperimentu razmaraju optimalni parametri algoritma, dok se u drugom ti parametri koriste za akviziciju leksikona iz korpusa.

5.4.1 Korpus

Korpus koji se koristi u većini eksperimenata što slijede dio je novinskog potkorpusa Hrvatskoga nacionalnog korpusa[2] (Tadić, 2002) i sačinjavaju ga tekstovi 94.465 članaka Vjesnikove digitalne arhive[3] objavljenih u razdoblju od 2000. do 2003. godine. Nakon opojavničenja, iz korpusa su izbačene zaustavne riječi, riječi kraće od tri slova, riječi koje sadržavaju znamenke, riječi u verzalu te sve riječi čija je učestalost pojavljivanja manja od dva (tzv. *hapax/dis legomenon*). U svim oblicima sva su slova pretvorena u mala slova. Time je dobiven korpus od ukupno 22.307.260 pojavnica odnosno 253.566 različnica.

5.4.2 Akvizicijski parametri

Parametri akvizicijskog algoritma izravno utječu na kvalitetu pribavljenoga leksikona. Ti su parametri sljedeći:

- redoslijed pribavljanja oblika iz korpusa (funkcija *wfpref* u tablici 5.1),
- minimalan broj oblika po LU-paru (parametar α u izrazu (5.1)) te
- funkcija ocjene *score*.

Kod vjerojatnosnih mjera, kvaliteta pribavljenog leksikona ovisit će također o uzorku na kojemu su izračunate vjerojatnosne razdiobe oblika.

S ciljem određivanja optimalnih vrijednosti ovih parametara, provedeno je izravno vrednovanje ispravnosti pribavljenih LU-parova na ručno sastavljenome uzorku. Postupak je vrednovan kao klasifikacijski problem: LU-parovi iz uzorka predstavljaju ciljni razred, dok iz njih izvedivi oblici predstavljaju podatke koje treba klasificirati. Za svaki je LU-par iz uzorka učinjeno sljedeće: (1) uporabom generativnog smjera morfološkog modela izvedeni su svi oblici tog LU-para, (2) jedan je od tih oblika izabran i lematiziran uporabom redukcijskog smjera modela, (3) svi LU-parovi dobiveni lematizacijom ocijenjeni su mjerom ocjene te su (4) odabrani oni najbolje ocijenjeni LU-parovi. Vrednovanje je zatim napravljeno u smislu uobičajenih mjera preciznosti P, odziva R i njihove harmonijske sredine iskazane mjerom F_1 (van Rijsbergen, 1979). Preciznost je dakle to veća što je pribavljeno manje lažnih LU-parova, dok je odziv to veći što je pribavljeno više ispravnih LU-parova.

[2] `http://www.hnk.ffzg.hr`
[3] `http://www.vjesnik.hr/html/`

Tablica 5.2: Uzorci LU-parova.

		Broj ovjerenih oblika			Učestalost ovjerenih oblika		
Uzorak	LU-parova	Prosječni	Najmanji	Najveći	Prosječna	Najmanja	Najveća
A	250	6,1	1	23	475,87	10	22,643
B	256	8,2	1	39	1.520,24	101	90,387

Uzorci

U eksperimentu su korištena dva ručno sastavljena uzorka: uzorak A od 250 LU-parova i uzorak B od 256 LU-parova (tablica 5.2). Uzorak A korišten je za vrednovanje postupka, a uzorak B korišten za izračunavanje vjerojatnosti oblika (za vjerojatnosne mjere ocjene). Uzorci su izgrađeni tako da je najprije načinjena lematizacija nasumično odabranih oblika iz korpusa, a zatim je za svaki oblik u rezultatima lematizacije ručno identificiran ispravan LU-par. Iz uzorka A izbačeni su LU-parovi kod kojih je ukupna učestalost oblika ovjerenih u korpusu manja od deset, dok su iz uzorka B izbačeni LU-parovi kod kojih je ukupna učestalost oblika manja od 100, kako bi se dobile pouzdanije vjerojatnosne procjene. Iz oba su uzorka izbačeni LU-parovi kojima je neki od oblika homografan, čime je zajamčeno da svaki oblik ima samo jedan ispravan LU-par.

Vjerojatnosna razdioba oblika flektivnih uzoraka izračunata je na uzorku B tako da je uzorak razdijeljen na pet poduzoraka prema sljedećim morfološkim vrstama: (1) imenice, (2) glagoli, (3) posvojni pridjevi, (4) opisni pridjevi bez određenog vida, (5) opisni pridjevi s određenim vidom. Pridjevi su razdijeljeni u tri kategorije jer je očekivano da se razdiobe oblika kod ovih triju kategorija podosta razlikuju. Valja napomenuti da je raspodjela na poduzorke načinjena nakon uzorkovanja, pa veličine poduzoraka nisu ujednačene.

Rezultati vrednovanja na uzorku A prikazani su tablicom 5.3. Rezultati su grupirani prema vrsti mjere ocjene, a unutar svake grupe masnim su slovima označeni maksimumi vrijednosti preciznosti, odziva i mjere F_1.

Kako bi se utvrdio utjecaj redoslijeda odabira oblika iz korpusa, za svaku su mjeru ocjene napravljena po tri izračuna. U prvome je za svaki LU-par odabiran i potom lematiziran najdulji oblik (oblik s najviše slova), u drugom je oblik proizvoljno odabiran, a u trećem je odabiran najkraći oblik svakog LU-para. Kao referentni navedeni su u prva dva retka tablice rezultati koji se dobivaju pribavljanjem svih LU-parova za odabrani oblik odnosno pribavljanjem jednog prozvoljno odabranog LU-para.

Tablica 5.3: Ispravnost pribavljenih LU-parova mjerena na uzorku A.

	Najdulji oblik			Proizvoljni oblik			Najkraći oblik		
Mjera	P	R	F_1	P	R	F_1	P	R	F_1
Referentne vrijednosti:									
Svi	4,33	**100**	8,29	6,19	**100**	11,65	6,55	**100**	12,30
Proizvoljno	0,40	0,40	0,40	12,35	12,40	12,38	**15,14**	15,20	**15,17**
Osnovne:									
$score_0$	36,66	98,40	53,42	26,29	95,60	41,24	25,49	94,40	40,14
$score_1$	36,98	**98,80**	53,81	26,19	94,40	41,01	25,49	94,00	40,10
$score_2$	**70,29**	77,60	**73,76**	34,73	46,40	39,73	33,55	42,00	37,30
Heuristička:									
$score_H$	**70,78**	**94,00**	**80,76**	48,88	87,20	62,64	48,53	86,00	62,05
Vjerojatnosne:									
$score_{P1}$	**74,28**	**82,00**	**77,95**	37,50	50,40	43,00	35,85	45,60	40,14
$score_{P2}$	71,27	76,40	73,75	42,73	56,40	48,62	44,15	52,80	48,09
$score_{P3}$	65,25	67,60	66,40	27,04	33,20	29,80	27,18	31,20	29,05
Kombinacije:									
$score_{02}$	79,43	89,60	84,21	59,42	74,40	66,07	58,71	72,80	65,00
$score_{12}$	79,79	90,00	84,59	58,41	73,60	65,13	58,58	72,40	64,76
$score_{1P2}$	80,00	**91,20**	85,23	61,89	76,00	68,22	27,18	31,20	29,05
$score_{1P3}$	**87,65**	88,00	**87,82**	65,09	71,60	68,19	64,26	71,20	67,55

Referentne vrijednosti potvrđuju visoku višeznačnost lematizacije: pribavljanjem svih LU-parova za odabrani oblik, preciznost doseže najviše 6,55%, što znači da je prosječan broj LU-parova po obliku veći od 15. Tako visok stupanj višeznačnosti lematizacije bitno ograničava preciznost morfološke normalizacije ostvarive izravnom primjenom redukcijskog morfološkog modela te opravdava pristup temeljen na akviziciji morfološkog leksikona.

Redoslijed odabira oblika

Što se redoslijeda odabira oblika tiče, razvidno je da on ima velik utjecaj na uspješnost pribavljanja ispravnih LU-parova. Uvjerljivo najbolji rezultati (izuzev za referentne vrijednosti) ostvaruju se onda kada se prvo odabiru najdulji oblici riječi, tj. oblici riječi s najviše slova, dok su rezultati najlošiji kada se odabiru najkraći oblici. Zanimljiva je međutim i na prvi pogled posve proturječna činjenica da je višeznačnost lematizacije kod duljih riječi zapravo veća nego kod kraćih riječi. Tako na uzorku A prosječan broj LU-parova za najdulje oblike riječi iznosi $23,32$, dok za najkraće oblik riječi iznosi $15,63$. To je zapravo i očekivano budući da su dulji oblici neke riječi upravo oni koji imaju dulje oblične nastavke, pa su znakovni sufiksi takvih nastavaka često ujedno i legitimni oblični nastavci. Na primjer, u obliku *brzoga*, oblični nastavak *-oga* u sebi sadržava sufiks *-a*, koji je također oblični nastavak, pa će lematizacija oblika *brzoga* dati više LU-parova nego li, primjerice, lematizacija oblika *brza*. Razlog zašto je pribavljanje ipak uspješnije kada se lematiziraju dulji oblici riječi leži u tome što je lažne LU-parove lakše identificirati kao takve ako su oni izvedeni iz duljih lema. Naime, dulje leme izvode i dulje oblike, a vjerojatnost da se neispravni oblik pojavi u korpusu, i tima poveća vjerodostojnost lažnog LU-para, to je manja što je taj oblik dulji. Zbog toga su dulji oblici riječi općenito indikativniji za određivanje ispravnog LU-para nego što su to kraći oblici.

Mjera ocjene

Najbolji se rezultati ostvaruju kombiniranim mjerama, i to ponajviše onima koje kombiniraju osnovne mjere i vjerojatnosne mjere. Tako mjera $score_{1P3}$ ostvaruje maksimum vrijednosti F_1 te je ujedno i najpreciznija mjera, dok najbolji odziv ostvaruje osnovna mjera $score_1$. Heuristička mjera $score_H$ udvostručuje preciznost osnovnih mjera $score_0$ i $score_1$, zadržavajući pritom razmjerno visoku razinu odziva. U smislu vrijednosti F_1, heuristička je mjera bolja i od vjerojatnosnih mjera, ali je lošija od kombiniranih mjera. Vjerojatnosne mjere, osobito mjera $score_{P1}$, znatno povećavaju preciznost os-

novnih mjera, ali također znatno narušavaju odziv. Ovdje međutim valja naglasiti da vjerojatnosne mjere bitno ovise o kvaliteti uzorka korištenog za procjene vjerojatnosti, pa je možda manje uputno uspoređivati ih s drugim mjerama.

Ovjerenost leme

Valja naglasiti da od 250 LU-parova uzorka A njih 28 u korpusu nema ovjerenu lemu. To znači da otprilike 10% LU-parova[4] ne bi bilo moguće pribaviti generativnim niti poluredukcijskim morfološkim modelom (v. odjeljak 3.1.1). Kani li se dakle pribaviti leksikon velikoga opsega, nužno je koristiti redukcijski morfološki model.

Minimalan broj oblika po LU-paru

Minimalan broj (još nepribavljenih) oblika po LU-paru određuje u izrazu (5.1) parametar α. Očito je da će preciznost leksikona biti to veća, a opseg leksikona to manji, što je veća vrijednost parametra α. Na uzorku A izmjereno je da $98,4\%$ LU-parova ima više od dva oblika ovjerenih u korpusu, dok ih samo $88,4\%$ ima više od tri ovjerena oblika. Vrijednost $\alpha = 2$ čini se dakle dobrim kompromisom između preciznosti i odziva.

5.4.3 Akvizicija leksikona

U drugome eksperimentu provedeno je pribavljanje flektivnog morfološkog leksikona iz korpusa opisanog u dijelu 5.4.1. Postupak je ponovljen pet puta s različitim mjerama ocjene LU-parova: osnovnom mjerom $score_1$, heurističkom mjerom $score_H$, vjerojatnosnom mjerom $score_{P1}$ te kombiniranim mjerama $score_{12}$ i $score_{1P3}$. Algoritam prvo odabire najdulje oblike, a između oblika identične duljine odabire one s većom učestalošću. Minimalan broj oblika po LU-paru je $\alpha = 2$.

Karakteristike pet pribavljenih leksikona sažeto su prikazane u tablici 5.4. Za svaki je leksikon prikazan ukupan broj unosaka, broj i udio nerazriješenih unosaka, broj oblika koji se mogu izvesti iz unosaka leksikona te broj i udio oblika koji su ovjereni u korpusu. Razvidno je da se pribavljeni leksikoni poprilično razlikuju po opsegu i stupnju nerazriješenosti. Najviše nerazriješenih unosaka ima leksikon L1 pribavljen mjerom ocjene $score_1$, što je i očekivano s obzirom na nisku razlučivost te mjere. Najmanje nerazriješenih unosaka imaju leksikoni pribavljeni mjerama $score_P$ i $score_{1P3}$, što je pak objašnjivo visokom razlučivošću tih mjera. Broj oblika proporcionalan je stupnju

[4] Udio bi vjerojatno bio još i veći kada bi se u obzir uzeli i LU-parovi s manje od deset ovjerenih oblika u korpusu.

Tablica 5.4: Karakteristike flektivnih leksikona pribavljenih iz korpusa.

		Unosci		Izvedivi oblici	
Leksikon	Mjera	Ukupno	Nerazriješeni (%)	Ukupno	Ovjereni (%)
L1	$score_1$	47.416	34.793 (73,38%)	2.093.703	217.411 (10,38%)
L2	$score_H$	47.276	27.417 (57,99%)	915.688	217.381 (23,74%)
L3	$score_{P1}$	50.871	6.378 (12,54%)	698.275	212.093 (30,37%)
L4	$score_{12}$	47.416	12.309 (25,96%)	632.511	217.410 (34.37%)
L5	$score_{1P3}$	47.415	4.841 (10,21%)	639.839	217.409 (33.98%)

nerazriješenosti leksikona, dok je broj ovjerenih oblika za sve leksikone otprilike jednak i kreće se u rasponu od 212 do 217 tisuća oblika, što čini oko 85% ukupnog broja različnica u korpusu. Leksikoni L1–L5 koristit će se u nastavku za morfološku normalizaciju temeljenu na leksikonu, te će se zasebno vrednovati njihova lematizacijska i normalizacijska točnost.

Poglavlje 6

Morfološka normalizacija temeljem leksikona

Morfološka normalizacija jest postupak kojim se flektivni, a eventualno i tvorbeni (derivacijski) oblici riječi zamjenjuju jednim reprezentativnim oblikom. U ovome poglavlju opisan je postupak flektivno-derivacijske morfološke normalizacije temeljen na flektivnom leksikonu pribavljenome iz korpusa te generativno-redukcijskome morfološkom modelu. Flektivni leksikon koristi se za flektivnu normalizaciju, a derivacijski uzorci za dodatnu derivacijsku normalizaciju. Budući da se oslanja na leksikon, postupak omogućuje visoku preciznost normalizacije, čak i onda kada su unosci leksikona nerazriješeni. Stupanj odnosno snagu normalizacije moguće je podesiti izborom odgovarajućih derivacijskih uzoraka. Također, prikladnim odabirom derivacijskih uzoraka moguće je u određenoj mjeri utjecati na stupanj značenjske povezanosti normaliziranih riječi.

U nastavku je najprije opisan normalizacijski postupak na flektivnoj i derivacijskoj razini, dok je problem značenjske povezanosti derivacijski normaliziranih riječi razmatran u drugome odjeljku.

6.1 Normalizacijski postupak

Normalizacija temeljena na leksikonu zasniva se na upitima nad pribavljenim flektivnim leksikonom $\mathcal{L}$. Osnovna zamisao jest upotrijebiti flektivni leksikon za što precizniju flektivnu normalizaciju, a zatim upotrijebiti skup derivacijskih uzoraka $\mathcal{D}$ za dodatnu derivacijsku normalizaciju. Na primjer, flektivnom normalizacijom oblika *brzinama* dobiva se norma *brzina* koju se, ukoliko je to potrebno i poželjno, daljnjom derivacijskom

normalizacijom može svesti na normu *brz*, čime se ostvaruje jači stupanj normalizacije. Osim za normalizaciju, leksikon se također može koristiti za lematizaciju i određivanje vrste riječi, uz ograničenje, dakako, da samo uporabom leksikona nije moguće razriješiti višeznačnost homografnih oblika.

6.1.1 Lematizacija

Temeljem leksikona $\mathcal{L}$ i skupa flektivnih uzoraka $\mathcal{F}$, lematizaciju ostvaruje funkcija $lemma_{\mathcal{L}} : \mathcal{S} \to \wp(\mathcal{S})$ definirana na sljedeći način:

$$lemma_{\mathcal{L},\mathcal{F}}(w) = \left\{ l \ : \ E \in \mathcal{L},\, (l, f) \in lm_{\mathcal{F}}(w) \cap E \right\}, \tag{6.1}$$

gdje je $lm_{\mathcal{F}}$ lematizacijska funkcija morfološkoga modela definirana izrazom (3.15). Budući da leksikon sadržava LU-parove, ali ne i njima pridružene oblike, upit se računalno izvodi u dva koraka. U prvome koraku upotrebljava se funkcija $lm_{\mathcal{F}}$ kako bi se izveli svi LU-parovi pridruženi obliku w, a zatim se od tih parova odabiru oni za koje postoji unosak u leksikonu, čime se zapravo povećava preciznost lematizacije funkcije $lm_{\mathcal{F}}$.

Ako je neki od ovih unosaka nerazriješen, funkcija će rezultirati s više LU-parova, a to će biti slučaj i ako je oblik w homografan, budući da je tada povezan s više unosaka leksikona. Na primjer, ako flektivni leksikon $\mathcal{L}$ sačinjavaju sljedeća dva unoska:

$$E_1 = \{(brz, f_{A1}), (br\check{z}eg, f_{A1})\},$$

$$E_2 = \{(brzati, f_{V1}), (brzati, f_{V2})\},$$

onda je, zbog toga što je unosak E_1 nerazriješen, lematizacija oblika *bržeg* višeznačna i rezultira s $lemma_{\mathcal{L},\mathcal{F}}(br\check{z}eg) = \{brz, br\check{z}eg\}$. Ako se dakle leksikon želi koristiti za lematizaciju, poželjno je da sadrži što manje nerazriješenih unosaka.

Budući da svaki flektivni uzorak odgovara jednoj vrsti riječi, vrstu riječi moguće je odrediti na osnovi povezanosti leme s flektivnim uzorkom. Neka funkcija $pos : \mathcal{F} \to POS$ flektivnim uzorcima pridružuje odgovarajuće oznake vrste riječi. Lematizaciju uz određivanje vrste riječi ostvaruje funkcija $lemmaPos_{\mathcal{L},\mathcal{F}} : \mathcal{S} \to \wp(\mathcal{S} \times POS)$ definirana na sljedeći način:

$$lemmaPos_{\mathcal{L},\mathcal{F}}(w) = \left\{ (l, pos(f)) \ : \ E \in \mathcal{L},\, (l, f) \in lm_{\mathcal{F}}(w) \cap E \right\}.$$

Ponekad je, osim leme, potrebno odrediti i morfosintaktički opis oblika, odnosno

skup mogućih višeznačnih morfosintaktičkih opisa. To je moguće ostvariti funkcijom $lemmaMsd_{\mathcal{L},\mathcal{F}} : \mathcal{S} \to \wp(\mathcal{S} \times \mathcal{X})$, definiranom kako slijedi:

$$lemmaMsd_{\mathcal{L},\mathcal{F}}(w) = \left\{(l, x) \ : \ E \in \mathcal{L},\ (l, f) \in lm_{\mathcal{F}}(w) \cap E,\ (w, x) \in lWfsMsd(l, f)\right\}. \tag{6.2}$$

Kao i gornje funkcije, funkcija *lemmaMsd* rezultirat će s više parova ako je oblik w homograf. Pored toga, ako je oblik w unutarnji homograf, funkcija će rezultirati s po jednim dodatnim parom za svaki morfosintaktički opis oblika w.

6.1.2 Flektivna normalizacija

Postupak morfološke normalizacije može se opisati funkcijom koja oblik riječi preslikava u morfološku normu. Međutim, kako normalizacija općenito može biti višeznačna – bilo zato što je oblik homograf ili zato što je leksikon nedovoljno precizan – jedan oblik općenito može imati više normi. Normalizaciju temeljenu na leksikonu $\mathcal{L}$ i skupu flektivnih uzoraka $\mathcal{F}$ opisuje funkcija $norm_{\mathcal{L},\mathcal{F}} : \mathcal{S} \to \wp(\mathcal{S})$, definirana na sljedeći način:

$$norm_{\mathcal{L},\mathcal{F}}(w) = \left\{encode(E) \ : \ E \in \mathcal{L},\ lm_{\mathcal{F}}(w) \cap E \neq \emptyset\right\}, \tag{6.3}$$

gdje je $encode : \wp(\mathcal{S} \times \mathcal{F}) \to \mathcal{S}$ funkcija koja unoske leksikona preslikava u norme, a $lm_{\mathcal{F}}$ je funkcija lematizacije definirana izrazom (3.15). Premda norma ne mora nužno biti znakovni niz, ne smanjujući općenitost u nastavku pretpostavljamo da to jest slučaj. S računalnoga stajališta, normalizacija oblika w provodi se u dva koraka. U prvome koraku, primjenom redukcijskog smjera morfološkog modela, odnosno funkcije $lm_{\mathcal{F}}$, oblik w se lematizira i iz leksikona $\mathcal{L}$ dohvaćaju se unosci povezani s oblikom w. Pritom smatramo da je unosak E povezan s oblikom w ako neki od LU-parova iz unoska E izvodi taj oblik. Ako je oblik w u leksikonu homografan, bit će dohvaćeno više unosaka. U drugome koraku svaki se dohvaćeni unosak pomoću funkcije *encode* preslikava u odgovarajuću normu. Definicija funkcije *encode* u načelu je posve proizvoljna, uz uvjet da je funkcija injektivna kako bi svaki unosak imao jedinstvenu normu. U praksi je ipak uputno za normu odabrati neku od lema iz LU-parova koji sačinjavaju unosak E, budući da je takve norme onda lakše interpretirati. Pritom treba osigurati da je odabrana lema jedincata, odnosno da neće biti upotrijebljena kao norma nekih drugih unosaka.

Zbog toga što se odabir norme temelji na unosku leksikona, a ne na pojedinačnim LU-parovima od kojih je taj unosak sačinjen, nerazriješenost unoska nema utjecaja na

preciznost normalizacije. To je u suprotnosti s gore opisanim postupkom lematizacije, kod kojega nerazriješeni unosci mogu uzrokovati smanjenje preciznosti. Razmotrimo kao primjer unosak E_1 sa slike 6.1. Budući da je taj unosak nerazriješen, lematizacija oblika *bržeg* je višeznačna i rezultira s $lemma_{\mathcal{L},\mathcal{F}}(bržeg) = \{brz, bržeg\}$. S druge strane, normalizacija tog istog oblika je jednoznačna te – uz pretpostavku da funkcija *encode* odabire najkraću lemu unoska – rezultira s $norm_{\mathcal{L},\mathcal{F}}(bržeg) = \{brz\}$.

6.1.3 Derivacijska normalizacija

Svrha derivacijske normalizacije jest zamijeniti jedinstvenom normom sve tvorbeno povezane riječi, odnosno sve riječi iste tvorbene porodice. U praksi se, naravno, riječi u tekstu pojavljuju u različitim flektivnim oblicima, pa je u tom smislu točnije govoriti o flektivno-derivacijskoj normalizaciji, tj. postupku kojim se jednom normom zamjenjuju svi *oblici* tvorbeno povezanih riječi.

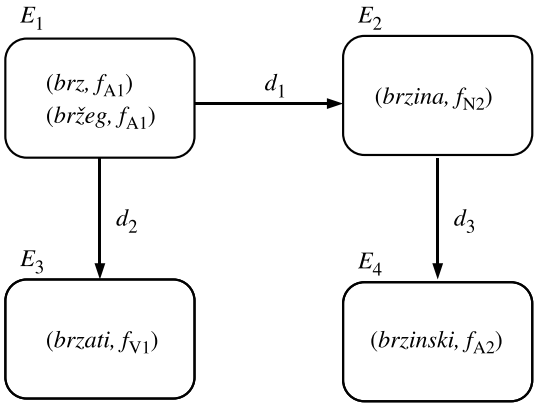
U ranije opisanom morfološkome modelu, flektivna sastavnica jasno je odvojena od derivacijske, pa je normalizaciju na tim dvjema morfološkim razinama moguće provesti odvojeno. Flektivna normalizacija ostvaruje se temeljem leksikona na gore opisani način, dok se derivacijska normalizacija može ostvariti prethodnim grupiranjem i stapanjem odgovarajućih unosaka leksikona. Ako su unosci leksikona stopljeni na način da tvorbeno povezane riječi (odnosno tvorbeno povezani LU-parovi) formiraju jedan unosak, onda se normalizacijom pomoću takvog leksikona ostvaruje flektivno-derivacijska normalizacija.

Grupiranje i stapanje leksikona definirat ćemo sasvim općenito na sljedeći način. Neka je R relacija ekvivalencije nad unoscima leksikona $\mathcal{L} = \{E_i\}$, te neka razred ekvivalencije R odgovara grupi unosaka koje treba stopiti. Neka je $\mathcal{L}/R$ leksikon koji se dobiva iz leksikona $\mathcal{L}$ grupiranjem i stapanjem unosaka prema relaciji R, kako slijedi:

$$\mathcal{L}/R = \left\{ E_2 \ : \ E_1 \in \mathcal{L},\, E_2 \in [E_1] \right\}, \tag{6.4}$$

gdje je $[E_1] = \{E_2 \in \mathcal{L} \ : \ (E_1, E_2) \in R\}$ razred ekvivalencije unoska E_1.

Kod derivacijske normalizacije, relaciju R treba definirati tako da razredi ekvivalencije odgovaraju unoscima čiji su LU-parovi tvorbeno povezani. U okviru morfološkog modela HOFM, tvorbena povezanost LU-parova modelirana je relacijom izravne tvorbene veze $\rightarrow_d$ prema (3.30). Temeljem ove relacije može se definirati njoj analogna relacija (izravne) tvorbene veze između parova unosaka leksikona na sljedeći način:



Slika 6.1: Grupa unosaka leksikona tvorbeno povezanih derivacijskim uzorcima d_1, d_2, i d_3. Unosak E_1 je nerazriješen i sadržava lažni LU-par $(brže g, f_{A1})$.

$$E_1 \rightarrow_{\mathcal{D}} E_2 \iff \exists d \in \mathcal{D}.\, \exists (l_1, r_1) \in E_1.\, \exists (l_2, r_2) \in E_2.\, (l_1, r_1) \rightarrow_d (l_2, r_2)\,. \tag{6.5}$$

Relacija $E_1 \rightarrow_{\mathcal{D}} E_2$ vrijedi čim je jedan od LU-parova iz unoska E_1 tvorbeno povezan s bilo kojim od LU-parova iz unoska E_2, i to temeljem bilo kojeg derivacijskog uzorka iz skupa $\mathcal{D}$. Nadalje, neka je $\overset{*}{\rightarrow}_{\mathcal{D}}$ tranzitivno zatvorenje relacije $\rightarrow_{\mathcal{D}}$ definirano nad leksikonom $\mathcal{L}$ kako slijedi:

$$E_1 \overset{*}{\rightarrow}_{\mathcal{D}} E_2 \iff (E_1 \rightarrow_{\mathcal{D}} E_2) \vee \left(\exists E \in \mathcal{L}.\, (E_1 \rightarrow_{\mathcal{D}} E) \wedge (E \overset{*}{\rightarrow}_{\mathcal{D}} E_2)\right). \tag{6.6}$$

Konačno, neka je $=_{\mathcal{D}}$ refleksivno i simetrično zatvorenje relacije $\overset{*}{\rightarrow}_{\mathcal{D}}$. Grupe tvorbeno povezanih unosaka leksikona odgovaraju dakle razredima ekvivalencije relacije $=_{\mathcal{D}}$, a $\mathcal{L}/=_{\mathcal{D}}$ je leksikon kod kojega su tvorbeno povezanih unosci stopljeni. U nastavku ćemo, radi preglednosti, umjesto $\mathcal{L}/=_{\mathcal{D}}$ pisati jednostavnije $\mathcal{L}/\mathcal{D}$.

Način rada flektivno-derivacijske normalizacije može se ilustrirati na sljedećem primjeru. Pretpostavimo da su morfološkim modelom, pored ranije definiranoga imeničkog uzorka f_{N1} i pridjevskog uzorka f_{A1}, također definirani flektivni uzorci f_{N2}, f_{A2} i f_{V1} za imenice, pridjeve odnosno glagole. Pretpostavimo nadalje da je flektivni leksikon $\mathcal{L}$ sačinjen od sljedeća četiri unoska (slika. 6.1):

$$E_1 = \{(brz, f_{A1}), (bržeg, f_{A1})\},$$

$$E_2 = \{(brzina, f_{N2})\},$$

$$E_3 = \{(brzati, f_{V1})\},$$

$$E_4 = \{(brzinski, f_{A2})\}.$$

U ovome je primjeru unosak leksikona E_1 nerazriješen budući da, pored lingvistički ispravnog LU-para (brz, f_{A1}), sadržava i lažni LU-par $(bržeg, f_{A1})$. Navedene riječi tvorbeno su povezane: pridjev *brz* motivira izvođenje glagola *brzati* i imenice *brzina*, dok potonja motivira izvođenje posvojnog pridjeva *brzinski*. Pretpostavimo da su ove tvorbene veze modelirane skupom derivacijskih uzoraka $\mathcal{D} = \{d_1, d_2, d_3\}$, tako da vrijedi $(brz, f_{A1}) \rightarrow_{d_1} (brzina, f_{N2})$, $(brz, f_{A1}) \rightarrow_{d_2} (brzati, f_{V1})$, i $(brzina, f_{N2}) \rightarrow_{d_3} (brzinski, f_{A2})$. Tada za relaciju $=_{\mathcal{D}}$ vrijedi $[E_1] = [E_2] = [E_3] = [E_4] = \{E_1, E_2, E_3, E_4\}$, pa je $\mathcal{L}/\mathcal{D} = \{E_1 \cup E_2 \cup E_3 \cup E_4\}$. Flektivno-derivacijski leksikon $\mathcal{L}/\mathcal{D}$ sadržava dakle samo jedan unosak, pa će normalizacijom pomoću tog leksikona svi oblici navedenih pet riječi biti svedeni na zajedničku normu. Na primjer:

$$norm_{\mathcal{L}/\mathcal{D},\mathcal{F}}(brzinama) = norm_{\mathcal{L}/\mathcal{D},\mathcal{F}}(brza) = \{brz\},$$

uz pretpostavku da funkcija *encode* za normu odabire najkraću lemu unoska. Valja primijetiti da je oblik *brza* homograf (može se raditi o nominativu jednine ženskoga roda pridjeva *brz* ili trećem licu jednine prezenta glagola *brzati*), no budući da su obje njegove leme tvorbeno povezane, normalizacija ipak rezultira samo jednom normom. U ovome primjeru za grupiranje unosaka leksikona upotrijebljena su sva tri derivacijska uzorka; uporabom manjeg broja uzoraka ostvario bi se slabiji stupanj derivacijske normalizacije.

Važno je naglasiti da je se, sukladno izrazu (6.6), tranzitivno zatvorenje relacije tvorbene veze provodi nad postojećim unoscima leksikona, a ne nad svim mogućim LU-parovima. Prema tome, ako neki unosci leksikona nedostaju (zato jer nisu bili pribavljeni), grupa tvorbeno povezanih unosaka bit će razdijeljena u više razreda ekvivalencije, što može narušiti učinkovitost derivacijske normalizacije. Na primjer, ako nije pribavljen unosak E_2 sa slike 6.1, preostala tri unoska leksikona sačinjavaju dva odvojena razreda ekvivalencije, pa oblike riječi *brzinski* ne bi bilo moguće normalizirati na zajedničku normu *brz*.

6.2 Problem značenjske povezanosti

Opisani postupak derivacijske normalizacije tvorbene porodice izjednačava s razredima ekvivalencije relacije inducirane skupom derivacijskih uzoraka. No time je posve zanemaren leksičko-semantički aspekt derivacijske morfologije. Za razliku od flektivne morfologije – kod koje je značenjska veza između oblika jedno te iste riječi neupitna i kod koje je zbog toga moguće u potpunosti zanemariti leksičko-semantički aspekt morfološke normalizacije – na derivacijskoj je razini taj aspekt po definiciji prisutan te se u kontekstu morfološke normalizacije očituje u vidu problema uvjetovanosti značenjske povezanosti osnovne riječi i izvedenice. Zbog toga je kod derivacijske normalizacije nužno u obzir uzeti sljedeće:

1. Izrazno povezane riječi nisu uvijek i sadržajno odnosno tvorbeno povezane (problem sadržajne nepovezanosti);

2. Značenja dviju tvorbeno povezanih riječi neminovno se u određenoj mjeri razlikuju, a razlika može biti uvjetovana i kontekstom (problem stupnja i konteksta značenjske veze).

Ovi su problemi detaljnije razmotreni u nastavku.

6.2.1 Problem sadržajne nepovezanosti

Riječi smatramo tvorbeno povezanima ako su povezane na planu izraza i na planu sadržaja. Međutim, kako je već istaknuto u odjeljku 3.2.11, relacija tvorbene veze $\rightarrow_d$, definirana derivacijskim uzorkom d, ne implicira stvarnu tvorbenu povezanost. Relacija $\rightarrow_d$ opisuje zapravo samo izraznu povezanost između dviju riječi, odnosno njima odgovarajućih LU-parova, dok sadržajna povezanost može, ali i ne mora postojati. Ako su relacijom $\rightarrow_d$ u vezu dovedene sadržajno nepovezane riječi, one će postupkom flektivno-derivacijske normalizacije biti svedene na identičnu normu, a to onda neminovno rezultira gubitkom informacije. Primjer su riječi *šal* i *šalica* koje su izrazno, ali ne i sadržajno povezane, pa bi tako flektivno-derivacijska normalizacija oblika, primjerice, *šalovima* i *šalicu* na zajedničku normu *šal* očito predstavljala neželjen gubitak informacije. Gubitak informacije predstavlja također i normalizacija etimološki povezanih riječi koje više nisu tvorbeno povezane. Primjerice, riječi *nov* i *novac* su etimološki i izrazno povezane, ali ne i sadržajno, pa njihova normalizacija na zajedničku normu vjerojatno nije poželjna.

Kako bi se stekao bolji uvid u razmjere sadržajne nepovezanosti uslijed primjene derivacijskih uzoraka morfološkoga modela, načinjen je pokus opisan u nastavku.

Uzorak LU-parova

Pokusom se želi utvrditi u kojoj mjeri pojedini derivacijski uzorci dovode u vezu sadržajno nepovezane riječi. Pokus je načinjen na uzorku LU-parova grupiranih u tvorbene porodice, izgrađenom na sljedeći način. Najprije je izgrađen uzorak od 10.000 oblika ručno grupiranih u flektivne i derivacijske grupe; ovaj uzorak i način njegove izrade detaljnije su opisani u odjeljku 7.2.2. Grupe flektivno povezanih oblika lematizirane su zatim pomoću leksikona L5 (v. tablicu 5.4) na način opisan u odjeljku 6.1.1, i to tako da je za svaku grupu odabran onaj LU-par koji izvodi sve oblike grupe, što je u većini slučajeva omogućilo razrješavanje višeznačnosti uslijed nerazriješenih unosaka ili homografije. Grupe kod kojih višeznačnost nije bilo moguće razriješiti na opisani način, ili kod kojih lematizacija nije uspjela, izuzete su iz uzorka. Tako je dobiven uzorak od 4.819 LU-parova grupiranih u 3.307 tvorbenih porodica. Prosječan broj LU-parova po tvorbenoj porodici iznosi 1,46, a najveći broj LU-parova po tvorbenoj porodici iznosi 11 (tvorbena porodica motivirana riječju *stvar*).

Analiza

Na opisanom je uzorku LU-parova provedena analiza svih 244 derivacijskih uzoraka definiranih morfološkim modelom (v. odjeljak 4.3). Za svaki je uzorak izračunat broj sadržajno podvezanih i nepovezanih LU-parova koje taj uzorak izvodi. Izračun je proveden tako da se svaki derivacijski uzorak pokušao primijeniti na svaki LU-par iz uzorka uporabom funkcije *derive* (v. odjeljak 3.2.10). Ako su na taj način izvedeni LU-parovi koji se u uzorku nalaze u istoj tvorbenoj porodici, oni su se brojali kao sadržajno povezani za taj uzorak, inače su se brojali kao sadržajno nepovezani.

U tablici 6.1 prikazani su rezultati za deset najčešće primjenjivanih derivacijskih uzoraka, dok su u tablici 6.2 prikazani rezultati za sve uzorke, grupirani u skupine prema značenjskoj kategoriji izvedenice (sukladno tablici 4.2). Od ukupno 244 derivacijskih uzoraka, na uzorku ih je bilo primjenjivo 151 (50,7%). Vidljivo je da postoje znatne razlike u zastupljenosti pojedinih uzoraka, što je i očekivano s obzirom da se tvorbeni uzorci razlikuju u plodnosti. Najčešće primjenjivani uzorci su oni koja izvode opisne pridjeve (skupine A-1, A-2 i A-3) te apstraktne i radne imenice (skupine N-10 i N-11), a od glagolskih izvedenica najčešći su vidski parnjaci (skupina V-1).[1] Od ukupno

[1] Ovi rezultati ukazuju na plodnost tvorbenih uzoraka u hrvatskome jeziku, ali ih ipak treba uzeti

Tablica 6.1: Analiza deset najčešće primijenjenih derivacijskih uzoraka.

			Broj izvedenica		
Skupina	Sufiks	Primjer	Povezanih	Nepovezanih	(%)
A-1	*-an*	*beskraj→beskrajan*	108	9	7,7
A-3	*-en*	*dogovoriti→dogovoren*	79	0	0
N-11	*-nje*	*čitati→čitanje*	64	0	0
A-2	*-ski*	*autor→autorski*	43	1	2,3
N-10	*-ost*	*aktivan→aktivnost*	40	2	4,8
A-3	*-an*	*čitati→čitan*	37	3	7,5
N-10	*-ost*	*duhovit→duhovitost*	28	0	0
N-11	*-enje*	*donositi→donošenje*	23	0	0
N-11	*-a*	*nagraditi→nagrada*	21	1	4,6
V-1	*-ati*	*javiti→javljati*	18	2	10,0

1.131 izvedenica, sadržajno povezanih izvedenica je 1.043, a sadržajno nepovezanih 88. Prosječni udio sadržajno nepovezanih izvedenica iznosi dakle razmjerno niskih 7,8%, ali i tu postoje zamjetne razlike između pojedinih derivacijskih uzoraka i skupina uzoraka.

Derivacijska normalizacija sadržajno nepovezanih riječi dovodi doduše do gubitka informacija, ali takvi su slučajevi, kao što je ovim pokusom pokazano, ipak razmjerno rijetki. U praksi mnogo veći problem predstavljaju značenjske razlike koje postoje između tvorbeno povezanih riječi. Taj se problem razmatra u nastavku.

6.2.2 Problem stupnja i konteksta značenjske veze

Tvorbena veza između dviju riječi pretpostavlja postojanje značenjske veze, ali također i značenjske različitosti. Značenje tvorenice, tzv. *tvorbeno značenje*, preinačeno je leksičko značenje osnovne riječi (Barić *et al.*, 2005). Preinakom je zadržana značenjska povezanost između osnovne riječi i tvorenice, no ta je povezanost podložna sljedećim leksičko-semantički čimbenicima:

1. Značenjska kategorija – značenjski otklon tvorenice može biti veći ili manji, što načelno ovisi o njenoj značenjskoj kategoriji (npr. riječi *baciti* i *bacati* doimaju se jače značenjski povezane nego riječi *baciti* i *bacač*);

s ogradom, i to iz barem tri razloga: za tu vrstu analize uzorak je vjerojatno premalen, nije reprezentativan, a nije provedeno niti razrješavanje tvorbene višeznačnosti koja se javlja kod pojedinih sufikasa (npr. sufiks *-aš* u *košarka→košarkaš* i *školjka→školjaš*).

Tablica 6.2: Analiza primjene derivacijskih uzoraka (uzorci su grupirani u skupine prema tablici 4.2).

		Broj izvedenica		
Skupina	Udio (%)	Povezanih	Nepovezanih	(%)
A-1	16,4	167	18	9,7
N-11	15,7	170	7	4,0
A-3	14,1	129	3	2,3
A-2	11,2	144	11	7,1
N-10	10,5	114	5	4,2
V-1	6,1	67	2	2,9
V-3	5,8	58	8	12,1
V-4	4,8	49	6	10,9
N-1	4,2	44	3	6,4
N-6	1,8	17	3	15,0
N-8	1,5	17	1	5,6
N-12	1,5	15	2	11,8
N-13	1,6	10	7	41,2
N-2	1,2	12	2	14,3
N-4	1,2	10	3	23,1
N-9	1,1	9	4	30,8
N-7	1,0	9	2	18,2
N-3	0,2	1	1	50,0
V-2	0,1	1	0	0
N-5	0	0	0	–
N-14	0	0	0	–
Ukupno:	100	1.131	88	7,8

2. Stilska obilježenost – tvorenica izrazom može pripadati jednoj značenjskoj skupini, no biti stilski obilježena kao da pripada drugoj značenjskoj skupini (npr. pogrdnica *činovničić*);

3. Daljina tvorbene veze – između riječi jedne tvorbene porodice postoje bliže i dalje tvorbene veze (npr. riječi *sluga* i *služiti* su u bliskoj, izravnoj tvorbenoj vezi, dok su riječi *sluga* i *služben* u daljoj, posrednoj tvorbenoj vezi);

4. Leksička višeznačnost – riječ može imati više različitih značenja, pa značenjska veza može biti uvjetovana smislom u kojem je ta riječ upotrijebljena u tekstu (npr. između riječi *kritičan* i *kritika* postoji značenjska veza ako je pridjev *kritičan* upotrijebljen u smislu onoga koji iskazuje kritiku, ali ne ako je upotrijebljen u smislu onoga što se odnosi na krizu);

5. Semantička netransparentnost – riječ može biti dijelom okamenjenog izraza u kojemu posve ili djelomično gubi svoje uobičajeno, samostalno značenje (npr. riječ *pas* u frazemu *morski pas* ili riječ *mačak* u frazemu *mačak u vreći*).

Ovi čimbenici različito djeluju na značenjsku povezanost: prva tri čimbenika utječu na snagu značenjske povezanosti, a posljedna dva uvjetuju njezino postojanje. Prema tome, značenjska se povezanost očituje u određenoj mjeri, a dodatno može biti kontekstno uvjetovana.

Derivacijskom normalizacijom tvorbeno povezane riječi svode se na identičnu normu, čime između njih iščezava svaka značenjska razlika. S obzirom na karakter značenjske povezanosti, postavlja se pitanje u kojoj je mjeri derivacijska normalizacija uopće poželjna. Treba razlikovati dva slučaja:

1. Ako se smislovi riječi ne podudaraju (čimbenik 4), ili ako je jedna od njih upotrijebljena u semantički netransparentnom kontekstu (čimbenik 5), njihova derivacijska normalizacija općenito nije poželjna jer rezultira gubitkom informacije;

2. Ako između riječi postoji značenjska veza, onda je derivacijska normalizacija poželjna utoliko što se njome mogu ukloniti nebitne značenjske razlike između tih riječi. No to istovremeno rezultira gubitkom informacije koji je to veći što je veća značenjska razlika između riječi (čimbenici 1–3). U konačnici, derivacijska normalizacija bit će korisna ako dobitak informacije ostvaren sažimanjem značenjski povezanih riječi nadmašuje gubitak informacije koji nastaje dokidanjem značenjskih razlika između riječi. Riječ se dakle o odabiru optimalnog stupnja derivacijske normalizacije.

Kako bi se spriječio neželjeni gubitak informacije, očito je da je u nekim slučajevima potrebno ograničiti derivacijsku normalizaciju tvorbeno povezanih riječi. Ograničavanje je, sasvim općenito, moguće provesti na:

1. leksičkoj razini (razini parova riječi),
2. razini tvorbenih uzoraka i
3. razini značenjskih kategorija.

Sprječavanje gubitka informacije uslijed višeznačnosti i semantičke netransparentnosti moguće je riješiti isključivo ograničavanjem normalizacije na leksičkoj razini, i to idealno na razini konteksta pojavnice.[2] Na stupanj derivacijske normalizacije može se utjecati ograničavanjem na svim trima razinama.

6.2.3 Ograničavanje odabirom derivacijskih uzoraka

Kod postupka derivacijske normalizacije opisanog u odjeljku 6.1.3, stupanj normalizacije može se ograničiti prikladnim odabirom derivacijskih uzoraka koji se koriste za grupiranje unosaka leksikona. Što je više uzoraka uključeno, to će razredi ekvivalencije biti veći, a također će biti veće i značenjske razlike između riječi svedenih na istu normu. Odabirom manjeg broja derivacijskih uzoraka tvorbene se porodice zapravo usitnjavaju u manje razrede ekvivalencije, čime se onda ukupno smanjuje stupanj derivacijske normalizacije. Na taj način doduše nije moguće riješiti problem višeznačnosti i semantičke netransparentnosti, no može se pretpostaviti da je taj problem ionako manje izražen, pogotovo ako je stupanj derivacijske normalizacije ograničen.[3]

Postavlja se pitanje koje derivacijske uzorke koristiti za normalizaciju. Odabir se može načiniti na razini pojedinačnih uzoraka ili na razini značenjskih kategorija. U nedostatku daljnjih spoznaja,[4] odabir na razini pojedinačnih uzoraka bilo bi teško opravdati. S druge strane, uvidom u primjere uz značenjske kategorije dane tablicom 4.2 stječe se dojam da je kod nekih kategorija razlika između osnovne riječi i izvedenice

[2] Kvalitetno rješenje ovog problema iziskivalo bi primjenu postupaka za razrješavanje ili razgraničenje višeznačnosti (engl. *word sense disambiguation/discrimination*, WSD) (Agirre & Edmonds, 2007), odnosno identificiranje semantički netransparentnih višerječnih jedinica, primjerice uporabom statističkih mjera za procjenu leksičke povezanosti riječi (Pecina, 2010).

[3] Jednako kao i za homografiju, bit će potrebno najprije utvrditi razmjere leksičke višeznačnosti u tekstovima na hrvatskom jeziku kako bi se utvrdilo u kojoj mjeri ona može utjecati na morfološku normalizaciju. U tom smislu indikativan može biti podatak da u elektroničkome izdanju Anićevog Velikog rječnika hrvatskoga jezika (Anić, 2003) na jednu natuknicu dolazi u prosjeku 1,43 definicija.

[4] Dobar uvid u semantičke odnose mogao bi se steći analizom na temelju modela računalne leksičke semantike (Saint-Dizier & Viegas, 1994), na primjer distribucijskih semantičkih modela (engl. *distributional semantic models*) (Lenci, 2008).

Tablica 6.3: Skupovi derivacijskih uzoraka grupirani prema procijenjenoj snazi značenjske povezanosti osnovne riječi i izvedenice.

Skup	Obuhvaćene značenjske kategorije	Broj uzoraka	% Nepovezanosti
$\mathcal{D}_1$	A-2, V-1	31	2,5
$\mathcal{D}_2$	A-1, A-2, A-3, V-1, V-3, V-4, N-4, N-11	117	6,2
$\mathcal{D}_3$	Sve	244	7,8

manja, a kod drugih veća. Na temelju takve subjektivne procjene značenjske skupine moguće je razdijeliti u sljedeće tri razine:

1. razina (jaka povezanost): A-2, V-1;
2. razina (umjerena povezanost): A-1, A-3, V-3, V-4, N-4, N-11;
3. razina (slaba povezanost): sve ostale kategorije.

Značenjski otklon doima se najmanjim kod izvedenica iz kategorija A-2 (posvojni pridjevi) i V-1 (vidski parnjaci), pa te značenjske kategorije sačinjavaju prvu razinu. Drugu razinu sačinjavaju kategorije A-1 i A-3 (opisni pridjevi i glagolski pridjevi), V-3 i V-4 (glagoli iz pridjeva i imenica) te N-4 i N-10 (mocijski parnjaci i radne imenice). Kod preostalih 13 kategorija značenjski otklon izvedenice doima se najvećim, pa su te kategorije svrstane na treću razinu. Prema ovoj podjeli, derivacijski su uzorci grupirani u tri skupa, $\mathcal{D}_1$, $\mathcal{D}_2$ i $\mathcal{D}_3$, pri čemu $\mathcal{D}_1 \subset \mathcal{D}_2 \subset \mathcal{D}_3$. U tablici 6.3 naveden je broj uzoraka u svakom skupu te udio sadržajno nepovezanih izvedenica mjeren na način opisan u odjeljku 6.2.1.

Uporabom opisanih skupova derivacijskih uzoraka, provedeno je grupiranje i stapanje unosaka flektivnih leksikona L1, L3 i L5 (v. tablicu 5.4). Karakteristike tako dobivenih flektivno-derivacijskih leksikona prikazane su u tablici 6.4. Iz tablice se može razabrati da su razredi ekvivalencije derivacijski povezanih unosaka u prosjeku veći kod leksikona L1 nego kod druga dva leksikona. Uzrok tome jest razlika u broju nerazriješenih unosaka leksikona: leksikon L1 sadržava mnogo više nerazriješenih unosaka, a takve je unoske lakše dovesti u vezu temeljem relacije $\rightarrow_{\mathcal{D}}$. Kod leksikona L2 i L3 nerazriješenost unosaka mnogo je manje izražena, odnosno ti su leksikoni precizniji, što općenito ograničava mogućnost njihova grupiranja.

Rezultati također ukazuju na problem koji nastaje zbog toga što se grupiranje provodi temeljem razreda ekvivalencije. Naime, kada se za grupiranje koriste svi derivacij-

Tablica 6.4: Karakteristike pribavljenih flektivno-derivacijskih leksikona.

	Broj unosaka			Veličina razreda ekv.	
Leksikon	Flektivnih	Stopljenih	Redukcija (%)	Prosječna	Najveća
L1-D1	47.416	43.327	8,62	1,09	8
L1-D2	”	35.403	25,55	1,34	27
L1-D3	”	30.708	35,23	1,54	71
L2-D1	47.276	43.775	7,41	1,10	5
L2-D2	”	37.089	21,55	1,28	19
L2-D3	”	32.674	30,89	1,45	61
L5-D1	47.415	44.477	6,20	1,07	5
L5-D2	”	38.158	19,52	1,24	19
L5-D3	”	34.310	27,64	1,38	53

ski uzorci modela, veća je izglednost povezivanja dvaju sadržajno nepovezanih unosaka (v. tablicu 6.3). U tom slučaju, budući da se unosci povezuju tranzitivno, lako nastaju veći razredi koji su sačinjeni od tvorbeno nepovezanih unosaka. Na primjer, leksikon L1-D3 sadržava razred ekvivalencije sačinjen od čak 71 flektivnog unoska. Ti unosci većinom nisu tvorbeno povezani, ali su se u konačnici ipak našli u istome razredu zbog ponekih sadržajno povezivanja sadržajno nepovezanih parova unosaka. Ovaj problem mogao bi se u perspektivi riješiti primjenom metoda grupiranja uz uporabu prikladno definirane mjere koja bi iskazivala stupanj i pouzdanost derivacijske povezanosti između unosaka.

Poglavlje 7

Eksperimentalno vrednovanje

U ovome poglavlju opisano je eksperimentalno vrednovanje kakvoće postupka morfološke normalizacije temeljene na leksikonu. Takvo se vrednovanje može napraviti na ekstrinzičan ili intrinzičan način. Ekstrinzičnim vrednovanjem kakvoća se ispituje posredno, mjerenjem utjecaja morfološke normalizacije na uspješnost izvođenja nekog specifičnog zadatka. Ekstrinzično vrednovanje je važno, no ono ne pruža baš nikakav uvid u funkcioniranje postupka normalizacije. Točnije, ekstrinzično vrednovanje ne omogućava nam razlučiti između slučajeva neispravne normalizacije i slučajeva u kojima je normalizacija ispravna, ali nije od koristi. Za to je potrebno provesti intrinzično vrednovanje, koje će dati ocjenu točnosti normalizacijskog postupka neovisno o nekom specifičnom zadatku. U ovome poglavlju opisano je iscrpno intrinzično vrednovanje normalizacijskog postupka temeljenog na leksikonu te usporedba tog postupka s nekim drugim pristupima morfološkoj normalizaciji, uključivo onim jezično nezavisnima.

7.1 Vrednovanje lematizacije

Kao što je ranije istaknuto, pribavljeni flektivni leksikoni mogu se, osim za normalizaciju, koristiti i za lematizaciju. Točnost lematizacije ovisit će o lingvističkoj kakvoći leksikona: ako leksikon sadržava lingvistički neispravne (lažne) LU-parove, kakvoća lematizacije bit će narušena. Također, kakvoća lematizacije bit će to manja što je manji opseg leksikona, tj. što je manje LU-parova u njega uključeno.

7.1.1 Način vrednovanja

Kako bi se izmjerila lematizacijska točnost pribavljenih flektivnih leksikona, potrebno je najprije utvrditi zlatni standard u odnosu na koji se mjerenje provodi. U tu svrhu

korišten je Hrvatski morfološki leksikon (HML) (Tadić & Fulgosi, 2003), kojemu je – budući da je sastavljan ručno i da je svaki njegov unosak pomno provjeravan – lingvistička kakvoća gotovo besprijekorna. Mjerenje točnosti lematizacije može se načiniti na oblicima iz pribavljenog leksikona, na oblicima iz HML-a ili na oblicima uzorkovanima iz korpusa. Vrednovanje na oblicima iz pribavljenog leksikona u suprotnosti je s idejom da se ti leksikoni koriste u redukcijskome, a ne u generativnome smjeru. Pribavljeni leksikoni mogu naime sadržavati brojne lažne LU-parove koji će generirati mnoge lingvistički neispravne oblike, a takvi se oblici neće pojavljivati u stvarnim tekstovima koji se normaliziraju. S druge strane, vrednovanje na oblicima iz HML-a nije prikladno jer moguće ne odražava stvarnu zastupljenost oblika u korpusu. Zbog toga se najprikladnijim čini lematizaciju vrednovati na oblicima iz uzorka.

U tu svrhu načinjen je uzorak oblika imenica, glagola i pridjeva slučajno uzrokovanih iz korpusa Vjesnika (v. odjeljak 5.4.1). Uzorak se sastoji od 100.000 pojavnica odnosno 28.715 različnica. Na tom je uzorku točnosti lematizacije izračunata u terminima mjera preciznosti (P) i odziva (R), definiranima na sljedeći način:

$$P = \frac{broj\ ispravnih\ lema}{broj\ dobivenih\ lema}, \qquad R = \frac{broj\ ispravnih\ lema}{broj\ mogućih\ lema}.$$

Broj ispravnih lema i broj mogućih lema utvrđuje se na temelju rezultata dobivenih HML-om. Pritom su iz uzorka izuzeti oblici koji ne postoje u jednom od dva leksikona, a također su iz rezultata dobivenih HML-om izuzete leme koje se dobivaju iz oblika glagolskog pridjeva trpnog (u modelu HOFM ti oblici obuhvaćeni su samo pridjevskim flektivnim uzorcima; v. odjeljak 4.2.2) te leme nepromjenjivih riječi. Ovakav način izračunavanja preciznosti i odziva uvažava činjenicu da je lematizacija višeznačna te da homografni oblici imaju više mogućih lema[1] (udio homografnih pojavnica na ovome konkretnom uzorku iznosi razmjerno visokih 13,64%).[2] Potpun odziv značio bi dakle da lematizacija svaki oblik svodi na sve njegove moguće leme, dok bi potpuna preciznost značila da lematizacijom niti jedan oblik nije sveden na neispravnu lemu. Kao kombinirana mjera preciznosti i odziva izračunata je mjera F_1 (van Rijsbergen, 1979), definirana kao $F_1 = 2PR/(P + R)$. Na istome uzorku izačunato je i pokrivanje (engl. *coverage*) leksikona, iskazano kao:

[1] Za razliku od, primjerice, sličnog pristupa vrednovanju korištenog u (Galvez & Moya-Anegón, 2006).

[2] Ovo je znatno veći udio od onog navedenog u (Tadić, 2003). To može biti i djelomično zbog toga što su sve pojavnice u uzorku zapisane malim slovima, pa homografnima postaju parovi oblika koji to inače nisu (npr. *jelena* – *Jelena*).

Tablica 7.1: Kakvoća lematizacije i pokrivanje mjereni na uzorku.

	Na pojavnicama				Na različnicama			
Leksikon	P	R	F_1	COV	P	R	F_1	COV
L1	62,41	**85,31**	72,08	**98,49**	60,71	**89,47**	72,34	**97,34**
L2	82,88	81,92	82,40	98,21	86,50	86,95	86,72	97,23
L3	78,38	78,33	78,36	97,17	80,29	80,83	80,56	95,65
L4	77,45	84,13	80,65	98,48	78,99	87,35	82,96	97,32
L5	**85,68**	82,46	**84,04**	98,48	**89,61**	87,52	**88,55**	97,32

$$COV = \frac{broj\ oblika\ iz\ uzorka\ koji\ postoje\ u\ lekiskonu}{broj\ oblika\ u\ uzorku}.$$

7.1.2 Rasprava rezultata

Rezultati vrednovanja prikazani su u tablici 7.1. Navedene su vrijednosti preciznosti, odziva, mjere F_1 i pokrivanja za lematizaciju pomoću leksikona L1–L5 (v. tablicu 5.4), izračunate zasebno na pojavnicama i na različnicama iz uzorka. HML na uzorku pokriva 95,27% pojavnica te 90,15% različnica. Kakvoća lematizacije, u smislu mjere F_1 i u smislu preciznosti, najbolja je za leksikon L5. Preciznost i odziv lematizacije (mjereni na različnicama) iznose 89,61% odnosno 86,38%, što znači da je u prosjeku devet od deset lema lingvistički ispravno i sadržano u leksikonu. Za pribavljanje leksikona L5 kao mjera ocjene LU-parova korištena je kombinirana mjera $score_{1P3}$, koja se već pokazala najboljom u smislu da najbolje diskriminira između ispravnih i lažnih LU-parova (v. tablicu 5.3). Malo je lošija lematizacija pomoću leksikona L2, za čije je pribavljanje korištena heuristička funkcija ocjene LU-parova $score_H$. Najveći odziv postiže se lematizacijom pomoću leksikona L1, kod kojega su međutim mnogi unosci nerazriješeni (v. tablicu 5.4), pa je preciznost lematizacije niska. Kod svih je leksikona kakvoća lematizacije bolja kada je mjerena na različnicama nego kada je mjerena na pojavnicama, što upućuje na to da je lematizacija nešto lošija kod čestih oblika. Pokrivanje je kod svih leksikona vrlo dobro te, uz iznimku leksikona L3, nadmašuje 97%.

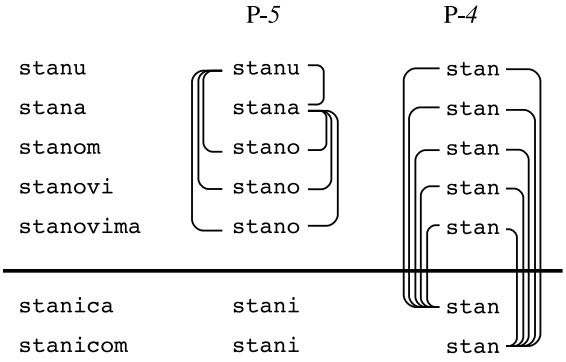
7.2 Intrinzično vrednovanje morfološke normalizacije

Kakvoća lematizacije može biti indikativna za kakvoću normalizacije, ali se one ne mogu poistovjetiti. Pri vrednovanju kakvoće normalizacije u obzir treba uzeti sljedeće. Prvo, nerazriješeni unosci leksikona narušavaju preciznost lematizacije, dok, kao što je opisano u odjeljku 6.1.2, nerazriješeni unosci ne utječu na normalizaciju. Drugo, za razliku od lema, morfološke norme ne moraju biti lingvistički ispravni oblici: kod morfološke je normalizacije bitno da morfološke varijante jedne te iste riječi budu svedeni na istu normu, neovisno o tome kako ta norma izgleda. Konačno, lematizacija je po definiciji ograničena na flektivnu razinu, pa kakvoća lematizacije ne govori ništa o kakvoći normalizaciji na derivacijskoj razini. Budući da je svrha morfološke normalizacije temeljene na leksikonu ista kao i ona korjenovanja, normalizaciju zapravo treba vrednovati na isti način kao i korjenovanje.

7.2.1 Vrednovanje metodom prebrojavanja pogrešaka

Postupci korjenovanja tradicionalno se vrednuju ekstrinzično, mjerenjem njihova utjecaja u specifičnim zadacima pretraživanja informacije ili dubinske analize teksta. Kako je istaknuto u (Paice, 1996), takav način vrednovanja ne pruža uvid u funkcioniranje postupka normalizacije, što je nužno kako bi se postupak mogao poboljšati. Bolji uvid u način funkcioniranja postupka normalizacije stječe se intrinzičnim vrednovanjem koje se provodi neovisno o specifičnome zadatku. U ovome je radu korištena metoda intrinzičnog vrednovanja predložena u (Paice, 1996). Slična je metoda predložena u (Kraaij & Pohlman, 1995).

Paiceova metoda temelji se na prebrojavanja pogrešaka *potkorjenovanja* i *prekorjenovanja* koje nastaju pri korjenovanju, odnosno općenito pri morfološkoj normalizaciji. Pogreške potkorjenovanja (engl. *understemming errors*) nastaju kada se oblici koji bi trebali biti svedeni na istu normu ne svedu na istu normu, dok pogreške prekorjenovanja (engl. *overstemming errors*) nastaju kada se na istu normu svedu oblici koji ne bi trebali biti svedeni na istu normu. Ove dvije vrste pogrešaka prikazane su na slici slici 7.1. Pogreške potkorjenovanja i prekorjenovanja prebrojavaju se na uzorku sastavljenom od oblika riječi grupiranih u grupe koje normalizacijskim postupkom treba svesti na identičnu normu. Kakvoću normalizacijskog postupka iskazuju *indeks potkorjenovanja* (UI) i *indeks prekorjenovanja* (OI) normalizacijskog postupa, izračunati na sljedeći način.



Slika 7.1: Primjer pogrešaka potkorjenovanja i prekorjenovanja prema (Paice, 1996): odsijecanje znakovnog niza nakon petog slova (P-5) uzrokuje sedam pogrešaka potkorjenovanja, dok odsijecanje nakon četvrtog slova (P-4) uzorkuje deset pogrešaka prekorjenovanja.

Indeks potkorjenovanja

Indeks potkorjenovanja *UI* odražava udio parova oblika iz uzorka koji nisu svedeni na istu normu, premda su u uzorku smješteni u istu grupu. Ovaj se indeks izračunava tako da se za svaku grupu g najprije izračuna broj parova koje treba svesti na istu normu, DMT_g (engl. *desired merge total*), kao:

$$DMT_g = \binom{N_g}{2} = \frac{1}{2} N_g(N_g - 1)\,, \tag{7.1}$$

gdje je N_g ukupan broj oblika u grupi g. Ukupan broj parova koje treba svesti na istu normu, $GDMT$ (engl. *global desired merge total*), dobiva se zbrajanjem vrijednosti DMT_g sviju grupa iz uzorka. Za svaku se grupu g zatim izračunava broj parova oblika koji nisu svedeni na istu normu, UMT_g (engl. *unachieved merge total*), kao:

$$UMT_g = \frac{1}{2} \sum_{i=1,\ldots,f_g} n_{gi}(N_g - n_{gi})\,, \tag{7.2}$$

gdje je f_g broj različitih normi u grupi g, a n_{gi} je broj pojavljivanja norme i u toj grupi. Ukupni broj parova koji nisu svedeni na istu normu, $GUMT$ (engl. *global unachieved merge total*), dobiva se zbrajanjem vrijednosti UMT_g sviju grupa iz uzorka. Indeks potkorjenovanja definiran je kao omjer ukupnog broja parova koji nisu svedeni na istu normu i ukupnog broja parova koji su trebali biti svedeni na istu normu, $UI = GUMT/GDMT$.

Indeks prekorjenovanja

Indeks prekorjenovanja *OI* odražava udio parova oblika koji u uzorku pripadaju različitim grupama, a svedeni su na istu normu. Izračunava se tako da se za svaku normu s najprije izračuna broj parova koji su svedeni na tu normu, AMT_s (engl. *actual merge total*), kao:

$$AMT_s = \binom{N_s}{2} = \frac{1}{2} N_s(N_s - 1)\,, \tag{7.3}$$

gdje je N_s ukupan broj pojavljivanja norme s u uzorku. Zbrajanjem po svim normama u uzorku dobiva se ukupan broj parova koji su svedeni na istu normu, $GAMT$ (engl. *global actual merged total*). Nadalje, za svaku normu s izračunava se broj parova oblika koji su pogrešno svedeni na tu normu, WMT_s (engl. *wrongly merged total*), kao:

$$WMT_s = \frac{1}{2} \sum_{i=1,\ldots,f_s} n_{si}(N_s - n_{si})\,, \tag{7.4}$$

gdje je f_s ukupan broj grupa kod kojih je neki od oblika sveden na normu s, a n_{si} je broj različitih normi u grupi i. Zbrajanjem po svim normama iz uzorka dobiva se ukupan broj parova koji su pogrešno svedeni na istu normu, $GWMT$ (engl. *global wrongly merged total*). Indeks prekorjenovanja definiran je kao omjer ukupnog broja parova koji su pogrešno svedeni na istu normu i ukupnog broja parova koji su svedeni na istu normu, $OI = GWMT/GAMT$. U (Paice, 1996), ovako definirani indeks prekorjenovanja naziva se *lokalnim*. Može se definirati i globalni indeks prekorjenovanja, kod kojega je udio pogrešno normaliziranih parova iskazan u odnosu na ukupan mogući broj parova u uzorku, ali je pokazano da je takva mjera suviše nestabilna te da uvelike zavisi o veličini uzorka.

Valja napomenuti da je ovdje opisani način vrednovanja zapravo istovjetan vrednovanju kakvo se uobičajeno provodi kod postupaka grupiranja. Grupiranje se naime može promatrati kao slijed binarnih odluka koje se donose između svih mogućih parova elemenata (Manning *et al.*, 2008). Ispravna pozitivna odluka jest ona kod koje su dva elementa ispravno svrstana u istu grupu, a ispravna negativna odluka ona kod koje su dva elementa ispravno svrstana u različite grupe. Slično, neispravna pozitivna odluka je ona kod koje su dva elementa pogrešno svrstana u istu grupu, a neispravna negativna ona kod koje dva elementa pogrešno nisu svrstana u istu grupu. Na temelju ovih vrijednosti moguće je zatim izračunati uobičajene mjere preciznosti (P) i odziva (R), i to uprosječivanjem vrijednosti svih grupa (tzv. makro-uprosječivanje) ili uprosječivanjem nad cijelim uzorkom (tzv. mikro-uprosječivanje). U kontekstu normalizacije, odluci da

se dva oblika svrstaju u istu grupu odgovara njihovo svođenje na identičnu normu. Ta je odluka ispravna ako se oblici u uzorku već nalaze u istoj grupi, a inače je neispravna. Uz takvu interpretaciju, između indeksa prekorjenovanja i potkorjenovanja te mikro-uprosječenih vrijednosti preciznosti i odziva vrijedi $OI = 1 - P$ i $UI = 1 - R$.

Snaga korjenovanja

Kao što je istaknuto u (Paice, 1996), mjere potkorjenovanja i prekorjenovanja u praksi su suprotstavljene u smislu da manje prekorjenovanja obično povlači više potkorjenovanja i obrnuto. Algoritam korjenovanja kod kojega je potkorjenovanje više izraženo nego prekorjenovanje naziva se "slabim" ili "konzervativnim", dok se algoritam kod kojega je više izraženo prekorjenovanje naziva "jakim" ili "agresivnim". Ovaj odnos iskazuje *snaga korjenovanja* (engl. *stemming weight*), definirana kao $SW = OI/UI$.

7.2.2 Uzorak oblika riječi

Paiceova metoda omogućava da se kakvoća normalizacijskog postupka kvantificira temeljem broja pogrešaka potkorjenovanja i prekorjenovanja. Izračun ovih pogrešaka provodi se na uzorku ručno grupiranih oblika, pa rezultati bitno ovise o načinu na koji je grupiranje provedeno. Grupe bi trebale biti sačinjene od morfološki i značenjski povezanih oblika, dakle oblika koje bi postupkom normalizacije trebalo svesti na istu normu. Problem predstavlja činjenica da je u mnogim slučajevima nemoguće utvrditi jesu li dvije riječi dovoljno značenjski povezane, a da bi ih se svelo na istu normu (taj je problem razmatran u odjeljku 6.2). U (Paice, 1996) predlaže se dvije riječi smjestiti u istu grupu ako se one "tipično odnose na istu temeljnu ideju". Pored toga, predloženo je da se grupiranje provodi u dvije razine: u tzv. "čvrste" i "labave" grupe, ovisno o procijenjenoj snazi značenjske veze. Indeksi prekorjenovanja i potkorjenovanja mogu se onda izračunati zasebno za čvrste i za labave grupe, čime se na neki način uvažava neizrazitost i uvjetovanost značenjskih veza. U praksi je, međutim, vrlo teško načiniti konzistentnu podjelu riječi u ovakve grupe, pa čak i onda kada postoje unaprijed dogovorena pravila o načinu grupiranja, kakva su djelomično opisana u (Paice, 1996). Odluke su naposlijetku većinom ipak subjektivne, a to onda znači da su granice između grupa u velikoj mjeri proizvoljne.

Grupiranje u flektivne i derivacijske grupe

U ovome radu problem nekonzistentnosti pri grupiranju nastojao se izbjeći na sljedeći način. Slično kao i kod (Paice, 1996), grupiranje je provedeno u dvije razine, ali su

(1)	{arheolog} {arheologija, arheologiju} {arheološka, arheološke, arheološki, arheoloških, arheološkog}
(2)	{arhitekt, arhitekta} {arhitekturi, arhitekture, arhitekturama} {arhitektonske, arhitektonskih, arhitektonskim, arhitektonskog}
(3)	{arhiva, arhivima, arhivu} {arhivske, arhivskim, arhivskoj} {arhivar}
(4)	{arija, arije, ariju}
(5)	{izdajicom} {izdatke} {izdat, izdati} {izdaje, izdavati} {izdavanje} {izdaje, izdajom} {izdanja, izdanje, izdanjem, izdanjima, izdanju} {izdavača, izdavače, izdavači} {izdavačka}
(6)	{grad, grada, grade, gradom, gradova} {gradska, gradske}
(7)	{grade, gradimo, graditi} {gradnja, gradnje}

Slika 7.2: Primjer flektivnih i derivacijskih grupa iz uzorka.

razine temeljene isključivo na morfološkim, a ne leksičko-semantičkim odnosima. Tako prvu razinu sačinjavaju grupe flektivnih oblika riječi (*flektivne grupe*), dok drugu razinu čine grupe oblika tvorbeno povezanih riječi, odnosno oblici onih riječi koje čine tvorbene porodice (*derivacijske grupe*). Primjer takvih grupa dan je slikom 7.2. Granice flektivnih grupa jasno su određene budući da su oblici riječi jednoznačno definirani flektivnim uzorcima gramatike. Derivacijske grupe dobivaju se stapanjem tvorbeno povezanih flektivnih grupa, pri čemu se dvije flektivne grupe smatraju tvorbeno povezanima ako između odgovarajućih riječi postoji tvorbena veza. Veće derivacijske grupe dobivaju se tranzitivnom primjenom ovoga pravila.

Uzorak korišten u ovome radu sastoji se od 10.000 različnica koje se dobivene slučajnim uzorkovanjem iz dijela korpusa Vjesnika opisanog u odjeljku 5.4.1. Kako bi se povećao broj tvorbeno povezanih oblika u uzorku, domena je ograničena na tekstove iz rubrike kulture. Rubrika kulture odabrana je zbog razmjerno visokog omjera broja različnica i pojavnica (engl. *type-token ratio*) u korpusu; u tekstovima iz rubrike kulture taj omjer iznosi 53%, dok, primjerice, u tekstovima iz rubrike sporta omjer iznosi svega 39%. Ručnim grupiranjem dobiveno je 5.510 flektivnih i 3.773 derivacijskih grupa. Flektivne grupe u prosjeku sadrže $1,82$ oblika, dok derivacijske grupe u prosjeku sadrže $1,46$ flektivnih grupa, odnosno $2,65$ oblika. Oblici glagolskoga pridjeva trpnog nisu tretirani kao oblici glagola, već kao zasebna flektivna grupa.

Problem tvorbene veze

Tvorbena veza, kao što je istaknuto u odjeljku 6.2, podrazumijeva postojanje izrazne i sadržajne veze, pri čemu problem može predstavljati izostanak ili kontekstna uvjetovanost sadržajne veze. Pri izgradnji uzorka, u slučajevima gdje postojanje sadržajne veza nije na prvi pogled očito, konzultirana je rječnička baza Hrvatskoga jezičnog portala,[3] koja je, pored ostaloga, temeljena na (Anić, 2003). Riječi koje su etimološki, ali ne više i sadržajno povezane (npr. *stol* i *stolica*) smatraju se tvorbeno nepovezanima. Ako su riječi višeznačne, smatraju se tvorbeno povezanima čim između nekih od njihovih smislova postoji značenjska veza (npr. *kritičan* i *kritika* smatraju se tvorbeno povezanima, a također i *izdati* i *izdavač*).

Budući da se veće derivacijske grupe dobivaju tranzitivnim grupiranjem, neće uvijek postojati izravna tvorbena veza između pojedinih parova riječi unutar derivacijske grupe. Posljedično, kod derivacijskih grupa postoji raznolikost u snazi i pouzdanosti značenjske veze. Primjer takvog slučaja jest derivacijska grupa (5) sa slike 7.2. Ta je grupa motivirana višeznačnim glagolom *izdati*, zbog kojeg su se u istoj grupi našle značenjski gotovo nepodudarne riječi *izdajica* i *izdavač*. Bitno je međutim naglasiti da ovakva raznolikost u snazi i kontekstnoj uvjetovanosti značenjske veze ne proizlazi ni iz kakvih subjektivnih i moguće nekonzistentnih procjena, kao što je to slučaj kod (Paice, 1996). Opisani način vrednovanja omogućava nam dakle da, unatoč zamršenim leksičko-semantičkim odnosima između riječi, pogreške prekorjenovanja i potkorjenovanja promatramo posve odvojeno od pogrešaka koje nastaju normalizacijom značenjski nepodudarnih riječi ili slabo povezanih oblika riječi.

Problem homografije

Drugi problem pri izgradnji uzorka predstavljaju (vanjski) homografi. Za razliku od korjenovanja, koje uvijek rezultira samo jednom normom, normalizacija temeljena na leksikonu homografne oblike svodi na više mogućih normi. Paiceva metoda vrednovanja, kako je izvorno predložena, nije prikladna za takve slučajeve. Moguće rješenje ovog problema, predloženo u (Kraaij & Pohlman, 1995), bilo bi da se iz uzorka izuzmu svi homografi. Time bi se međutim potpuno zanemario problem homografije, odnosno u obzir se ne bi uzele pogreške uslijed neispravne normalizacije homografa. Takav pristup nije zadovoljavajući, osobito ako se uzme u obzir da ukupan udio homografa u uzorku, izračunat na temelju rezultata lematizacije pomoću HML-a, iznosi razmjerno visokih 12,15%.

[3] `http://hjp.srce.hr`

Zbog toga je u ovome radu korišten drugačiji pristup. Svi homografi su zadržani u uzorku. Pored toga, homografi koji su se mogli pridijeliti u više flektivnih grupa kopirani su i pridijeljeni odgovarajućim flektivnim grupama, pod uvjetom da te grupe imaju barem jedan oblik koji nije homograf. Primjer je oblik *izdaje* koji je u derivacijskoj grupi (5) sa slike 7.2 upotrijebljen u dvije flektivne grupe, a također i oblik *grade* koji je u upotrijebljen u dvjema derivacijskim grupama, (6) i (7). Takvih oblika koji su u uzorku upotrijebljeni više puta ima svega 1% (udio je manji od ukupnog udjela homografa u uzorku budući da za većinu homografa u uzorku ne postoje dodatne flektivne grupe kojima bi se oni mogli pridijeliti). Prije izračunavanja vrijednosti indekasa *UI* i *OI*, provodi se dodatni korak razrješavanja u kojemu se za svaki homografni oblik odabire ona norma koja je za dotičnu grupu najčešća. Budući da svaka grupa ima barem jedan oblik koji nije homograf, razrješavanje se uvijek može provesti jednoznačno. U idealnom slučaju, ako normalizacijski postupak homografe ispravno normalizira (u smislu da svaki homografni oblik svodi na sve moguće norme), nakon razrješavanja svi će oblici iste grupe biti svedeni na identičnu normu, pa pogrešaka potkorjenovanja i prekorjenovanja neće biti.

7.2.3 Mjere kakvoće normalizacije

Indeksi prekorjenovanja i potkorjenovanja, kako su definirane u Paice (1996), pružaju uvid u kakvoću postupka normalizacije, ali ne daju ukupnu ocjenu kakvoće. Zbog toga se u ovom radu predlažu neke dodatne mjere za vrednovanje kakvoće normalizacije.

Kakvoća flektivne i derivacijske normalizacije

Idealni normalizacijski postupak jest onaj kod kojega nema niti pogrešaka potkorjenovanja niti pogrešaka prekorjenovanja. U praksi je ipak potrebno pronaći kompromis između ovih dviju vrsta pogrešaka. Taj je kompromis istovjetan onome koji kod pretraživanja informacija postoji između preciznosti i odziva, i zbog kojega se te mjere često objedinjuju mjerom F_1, odnosno općenito mjerom F_β (van Rijsbergen, 1979). Po uzoru na mjeru F_1, a uzimajući u obzir da vrijedi $P = 1 - OI$ i $R = 1 - UI$, mjeru *kakvoće flektivne normalizacije*, *iSQ* (engl. *inflectional stemming quality*), definiramo na sljedeći način:[4]

$$iSQ = \frac{2(1 - iUI)(1 - iOI)}{2 - iUI - iOI}\,, \tag{7.5}$$

[4]Kao i kod indekasa *UI* i *OI*, pojam *korjenovanja* (engl. *stemming*) ovdje se koristi u širem smislu.

gdje su *iUI* i *iOI* indeksi potkorjenovanja i prekorjenovanja izračunati na flektivnim grupama. Mjera *iSQ* poprima vrijednost jedinice ako na razini flektivnih grupa nema pogrešaka potkorjenovanja niti prekorjenovanja, što bi značilo da je flektivna normalizacija na uzorku savršeno točna.

Na sličan način definiramo mjeru *kakvoće derivacijske normalizacije*, *dSQ* (engl. *derivational stemming quality*), kao:

$$dSQ = \frac{2(1 - dUI)(1 - dOI)}{2 - dUI - dOI}, \tag{7.6}$$

gdje su *dUI* i *dOI* indeksi potkorjenovanja i prekorjenovanja mjereni na derivacijskim grupama. Mjera *dSQ* poprima vrijednost jedinice ako na razini derivacijskih grupa nema pogrešaka potkorjenovanja niti prekorjenovanja, što bi pak značilo da je flektivno-derivacijska normalizacija na uzorku savršeno točna.

Mjere *iSQ* i *dSQ* pretpostavljaju da pogreške potkorjenovanja i prekorjenovanja imaju jednaku težinu. To je u većini slučajeva razumna pretpostavka, premda su moguće i primjene kod kojih je jedna vrsta pogreške manje problematična od druge.[5]

Kakvoća flektivno-derivacijske normalizacije

Mjere *iSQ* i *dSQ* kakvoću morfološke normalizacije opisuju zasebno na flektivnoj i na derivacijskoj razini. Mjera *iSQ* dobar je pokazatelj kakvoće flektivne normalizacije i ona može biti korisna kada je normalizaciju potrebno provesti upravo na flektivnoj razini, odnosno kada normalizacija tvorbeno povezanih riječi nikako nije poželjna. Međutim, niti jedna od ovih dviju mjera nije dobar pokazatelj kakvoće flektivno-derivacijske normalizacije. Mjera *iSQ* očito za tu svrhu nije prikladna jer flektivno-derivacijska normalizacija normalizira i flektivno nepovezane oblike, što bi se na flektivnoj razini odrazilo kao prekorjenovanje. S druge strane, mjera *dSQ* nije prikladna zato što pretpostavlja svođenje baš svih tvorbeno povezanih oblika na jednu normu, što u praksi ipak nije poželjno. Naime, kao što je istaknuto u odjeljku 6.2.2, značenjske veze između tvorbeno povezanih riječi različite su jakosti i nerijetko su kontekstno uvjetovane. Kako bi se spriječio gubitak informacije, potrebno je, ovisno o specifičnome zadatku, odabrati prikladan stupanj derivacijske normalizacije, odnosno nekako ograničiti derivacijsku normalizaciju, a to bi se kod mjere *dSQ* neminovno odrazilo kao povećanje broja pogrešaka potkorjenovanja.

[5] Primjer takvog zadatka jest ekstrakcija kolokacija iz korpusa (Delač *et al.*, 2009), kod kojega su pogreške prekorjenovanja manje bitne od pogrešaka potkorjenovanja. Kolokacije se sastoje od dvije ili više riječi, pa to umanjuje vjerojatnost da se zbog prekorjenovanja dvije različite kolokacije svedu na isti normalizirani oblik.

Mjera kakvoće flektivno-derivacijske normalizacije treba dakle uvažiti činjenicu da stupanj derivacijske normalizacije u načelu može biti proizvoljan, odnosno da stupanj derivacijske normalizacije ne određuje kakvoću normalizacije u smislu pogrešaka potkorjenovanja i prekorjenovanja. Ovo je moguće ostvariti ako se u obzir uzme sljedeće. Kod flektivno-derivacijske normalizacije izvjesno je da svi flektivno povezani oblici trebaju biti svedeni na istu normu, kao što je izvjesno i to da tvorbeno nepovezani oblici ne smiju biti svedeni na istu normu. U kontekstu metode brojanja pogrešaka, to znači da se normalizacija oblika treba provesti barem unutar flektivnih grupa, ali nikako izvan derivacijskih grupa. Normalizacija oblika koji pripadaju istim derivacijskim, ali različitih flektivnim grupama može se i ne mora biti provesti. To znači da pri izračunu kakvoće flektivno-derivacijske normalizacije u obzir treba uzeti samo pogreške potkorjenovanja na flektivnoj razini te pogreške prekorjenovanja na derivacijskoj razini. U tom smislu, mjeru *kakvoće flektivno-derivacijske normalizacije* (*idSQ*) moguće je definirati na sljedeći način:

$$idSQ = \frac{2(1 - iUI)(1 - dOI)}{2 - iUI - dOI}\,, \tag{7.7}$$

gdje je *iUI* indeks potkorjenovanja na flektivnim grupama, a *dOI* indeks prekorjenovanja na derivacijskim grupama. Mjera *idSQ* neosjetljiva je na razlike u stupnju derivacijske normalizacije te u obzir uzima samo one pogreške potkorjenovanja i prekorjenovanja za koje je neupitno da su doista pogreške.

Stupanj derivacijske normalizacije

Mjera kakvoće flektivno-derivacijske normalizacije *idSQ* odražava kakvoću normalizacijskog postupka u smislu pogrešaka potkorjenavanja i prekorjenovanja, ali ne govori ništa o ostvarenom stupanju derivacijske normalizacije, odnosno o tome koliko je tvorbeno povezanih riječi postupkom normalizacije svedeno na zajedničku normu. U tu svrhu definiramo mjeru *stupnja derivacijske normalizacije*, *dSS* (engl. *derivational stemming strength*), na način opisan u nastavku.

Po uzoru na (Paice, 1996), najprije za svaku derivacijsku grupu g iz uzorka izračunavamo ukupan broj tvorbeno povezanih parova, $PDMT_g$ (engl. *possible derivational merge total*), kao:

$$PDMT_g = \sum_{i<j} n_{g,i}\, n_{g,j}\,. \tag{7.8}$$

Taj je broj jednak broju parova oblika (s_1, s_2) za koje se oba oblika nalaze u derivacijskoj

grupi g, ali nisu u istoj flektivnoj grupi. Valja ponoviti da, zbog leksičko-semantičke prirode tvorbene veze, oblici s_1 i s_2 ne moraju biti u izravnoj tvorbenoj vezi, te da u nekim kontekstima ta veza može izostati. Zbrajanjem po svim derivacijskim grupama iz uzorka dobiva se ukupan broj tvorbeno povezanih parova, *GPDMT* (engl. *global possible derivational merge total*).

Nadalje, za svaku derivacijsku grupu g izračunava se broj tvorbeno povezanih parova oblika koji su doista svedeni na istu normu, *ADMT* (engl. *actual derivational merge total*), kao:

$$ADMT_g = \sum_{i<j} \sum_{s_1 \in g_i, s_2 \in g_j} \mathbf{1}(s_1 = s_2)\,, \tag{7.9}$$

gdje je $\mathbf{1}(X)$ indikacijska funkcija koja je jednaka jedinici ako je uvjet X ispunjen, a inače je jednaka nuli. Zbrajanjem po svim derivacijskim grupama iz uzorka izračunava se ukupan broj tvorbeno povezanih parova oblika svedenih na istu normu, *GADMT* (engl. *global actual derivational merge total*). Stupanj derivacijske normalizacije zatim je definiran kao omjer ukupnog broja normaliziranih parova tvorbeno povezanih oblika i ukupnog broja tvorbeno povezanih oblika, $dSS = GADMT/GPDMT$. Što je vrijednost mjere *dSS* veća, to je veći stupanj derivacijske normalizacije, dok za čisto flektivnu normalizaciju vrijedi $dSS = 0$.

7.2.4 Analiza kakvoće normalizacije

Rezultati intrinzičnog vrednovanja normalizacije temeljene na leksikonu prikazani su tablicom 7.2. Tablica je organizirana u dva dijela: u gornjem dijelu tablice dani su rezultati normalizacije ostvareni pomoću flektivnih leksikona L1–L5. U donjem dijelu prikazani su rezultati normalizacije pomoću flektivno-derivacijskih leksikona dobivenih iz leksikona L1, L2 i L5 (v. tablicu 6.4), grupirani u tri grupe prema korištenim derivacijskim uzorcima. Prikazane su vrijednosti indeksa potkorjenovanja i prekorjenovanja te kakvoća i snaga normalizacije, zasebno na flektivnoj i na derivacijskoj razini, te kakvoća flektivno-derivacijske normalizacije i stupanj derivacijske normalizacije. Optimumi ovih vrijednosti unutar svake grupe rezultata istaknuti su masnim slovima. U posljednjem stupcu dane su vrijednosti *faktora sažimanja indeksa*, *ICF* (engl. *index compresion factor*), definiranog kao $ICF = (W - N)/W$, gdje je W broj različnica prije normalizacije, a N je broj različitih normi (Frakes & Fox, 2003). Oblici za koje normalizacije nije davala rezultata (tj. oblici koji ne postoje u leksikonu) ostavljeni su nepromijenjenima.

Tablica 7.2: Kakvoća normalizacije temeljene na leksikonu mjerena na flektivnoj i na derivacijskoj razini.

	Flektivna razina				Derivacijska razina						
Leksikon	% *iUI*	% *iOI*	% *iSQ*	*iSW*	% *dUI*	% *dOI*	% *dSQ*	*dSW*	% *idSQ*	% *dSS*	% *ICF*
Flektivni leksikoni:											
L1	5,77	5,16	**94,53**	0,89	63,47	2,95	53,08	0,05	**95,62**	1,34	44,73
L2	6,02	5,30	94,34	0,88	63,49	2,89	53,07	0,05	95,52	1,46	44,46
L3	11,11	**4,61**	92,03	0,42	65,57	**2,47**	50,90	0,04	93,01	1,22	42,75
L4	**5,76**	5,23	94,50	0,91	63,44	2,95	53,11	0,05	**95,62**	1,38	44,76
L5	5,79	5,56	94,32	0,96	**63,40**	3,15	**53,12**	0,05	95,51	1,47	**44,77**
Flektivno-derivacijski leksikoni:											
L1-D1	**5,59**	18,53	87,46	3,32	**57,47**	3,13	**59,11**	0,05	95,63	10,89	**47,15**
L2-D1	5,85	18,27	87,50	3,12	57,65	**2,95**	58,97	0,05	95,58	10,76	46,93
L5-D1	5,61	**18,25**	**87,62**	3,25	57,59	3,04	59,01	0,05	**95,66**	10,71	46,99
L1-D2	**5,00**	48,76	**66,58**	9,76	**37,39**	10,86	73,56	0,29	91,98	42,85	**55,33**
L2-D2	5,26	**47,02**	67,96	8,94	38,41	**9,09**	73,43	0,23	**92,78**	41,37	54,51
L5-D2	5,02	47,07	67,98	9,39	38,49	9,52	**73,23**	0,25	92,68	41,09	54,49
L1-D3	**4,83**	60,74	55,59	12,58	**24,71**	18,01	78,49	0,73	88,09	63,16	**59,19**
L2-D3	5,09	58,39	57,85	11,47	26,63	15,10	**78,71**	0,57	89,63	60,23	58,25
L5-D3	4,85	**57,66**	**58,60**	11,89	27,61	**14,97**	78,21	0,54	**89,81**	58,51	57,94

Flektivna normalizacija

Rezultati otkrivaju da je na flektivnoj razini kakvoća normalizacije za sve leksikone, osim za leksikon L3, podjednaka i iznosi gotovo 95%. Indeksi prekorjenovanja i potkorjenovanja ujednačeni su i iznose razmjerno niskih 5%, osim za leksikon L3 kod kojega je potkorjenovanje izraženije. Očekivano, kakvoća derivacijske normalizacije flektivnih leksikona je niska budući da na derivacijskoj razini rezultira velikim brojem pogrešaka potkorjenovanja. Flektivno-derivacijska kakvoća normalizacije za sve leksikone, izuzev za leksikon L3, iznosi više od 95%. Stupanj derivacijske normalizacije kreće se između 1% i 2%, što ukazuje na to da normalizacija u vrlo malom broju slučajeva prelazi flektivna granicu, odnosno da je ono gotovo potpuno flektivna. Faktor sažimanja indeksa iznosi oko 45%, što znači da je samo flektivnom normalizacijom veličinu indeksa moguće gotovo prepoloviti.

Flektivno-derivacijska normalizacija

Kod normalizacije pomoću flektivno-derivacijskih leksikona rezultati za flektivnu razinu nisu od interesa (normalizacija flektivno-derivacijskim leksikonima na flektivnoj razini uzrokuje pogreške prekorjenovanja). Na derivacijskoj razini udio pogrešaka potkorjenovanja i prekorjenovanja ovisi o stupnju derivacijske normalizacije. Očekivano, što je manje derivacijskih uzoraka korišteno za derivacijsku normalizaciju, to je veći indeks potkorjenovanja i to je manji indeks prekorjenovanja. Kada se za derivacijsku normalizaciju koristi skup derivacijskih uzoraka $\mathcal{D}_1$ (v. tablicu 6.3), stupanj derivacijske normalizacije iznosi oko 10%, a kakvoća flektivno-derivacijske normalizacije iznosi srazmjerno visokih 95%, pri čemu između leksikona L1-D1, L2-D1 i L5-D1 nema značajnijih razlika. Uključivanjem većeg broja derivacijskih uzoraka, stupanj derivacijske normalizacije raste na oko 42% za skup $\mathcal{D}_2$, odnosno oko 60% za skup $\mathcal{D}_3$. Međutim, uključivanjem većeg broja uzoraka ujedno raste i broj pogrešaka prekorjenovanja na derivacijskoj razini, što smanjuje ukupnu kvalitetu flektivno-derivacijske normalizacije s 95% na nešto manje od 90%. S porastom stupnja derivacijske normalizacije uočljiva postaje i razlika u kvaliteti između pojedinih leksikona: kod leksikona s većim brojem nerazriješenih unosaka (leksikon L1) kakvoća flektivno-derivacijske normalizacije je manja, budući da je u tom slučaju veća vjerojatnost grupiranja tvorbeno nepovezanih unosaka. Faktor sažimanja indeksa *ICF* kreće se u intervalu od oko 47% do najviše oko 60%, ovisno o ostvarenom stupnju derivacijske normalizacije.

Zanimljivo je da stupanj derivacijske normalizacije, čak i kada se upotrijebe svi derivacijski uzorci definirani modelom, ne prelazi 60%, što upućuje na to da mnogi

tvorbeno povezani unosci u leksikonu ostaju negrupirani. To može biti zato što neki derivacijski uzorci nedostaju ili – što je vjerojatno bitnije – zbog toga što u leksikonu nedostaju unosci preko kojih bi se ostvarilo tranzitivno zatvaranje tvorbeno povezanih unosaka.

Kakvoća normalizacije kreće se dakle u intervalu od 90 do 96%. Koliko je to dobro moguće je ocijeniti samo na temelju usporedbe s drugim postupcima normalizacije, pa je to tema idućeg odjeljka.

7.2.5 Poredbena analiza kakvoće normalizacije

Opisani način intrinzičnog vrednovanja moguće je primijeniti na bilo koji postupak morfološke normalizacije. To omogućuje da se kakvoća normalizacije temeljene na leksikonu izravno (dakle na istome uzorku) usporedi s kakvoćom drugih postupaka. U nastavku su uspoređeni sljedeći pristupi:

- L5(-D1/D2/D3) – Flektivni normalizacijski leksikon L5 te flektivno-derivacijski leksikoni L5-D1, L5-D2 i L5-D3. Budući da prethodnim eksperimentom nije utvrđeno postojanje značajnih razlika u kakvoći pribavljenih leksikona, odabran je leksikon L5 kao onaj leksikon koji postiže najbolju kakvoću lematizacije.

- P-n – Odsijecanje znakovnoga niza nakon n-tog slova, odnosno zadržavanje prefiksa znakovnoga niza duljine n. Ovo je najjednostavniji mogući pristup morfološkoj normalizaciji i stoga postavlja donju granicu (engl. *baseline*) kakvoće normalizacije;[6]

- S-1 – Korjenovanje odbacivanjem sufiksa znakovnoga niza do (uključivo) posljednjeg samoglasnika, pod uvjetom da je duljina uklonjenog sufiksa manja ili jednaka duljini ostatka (pseudokorijena). Ovaj jednostavni postupak, upotrijebljen u (Ljubešić, 2009), opravdan je činjenicom da većina obličnih nastavaka u hrvatskome jeziku započinje samoglasnikom;

- S-2 – Korjenovanje primjenom redukcijskog smjera modela HOFM. Postupak primjenjuje sve flektivne uzorke modela kako bi dobio obličnu osnovu riječi, a zatim od više mogućih osnova odabire onu koja je najkraća (izbor se u prosjeku čini između čak 6,85 različitih osnova, što opet odražava visok stupanj višeznačnosti

[6] Alternativu predstavlja odbacivanje završnih n slova znakovnoga niza, što međutim nije dobar normalizacijski postupak jer su oblični nastavci često različite su duljine, pa različiti oblici riječi ne bi bili svedeni na istu normu.

flektivnih uzoraka). Ovaj postupak odgovara najjednostavnijem pristupu korjenovanja temeljenom na odbacivanju sufiksa, kod kojega se u slučaju višeznačnosti dulji sufiks preferira nad onim kraćim, i u tom smislu postavlja donju granicu kakvoće za postupke korjenovanja temeljene na odbacivanju sufiksa;

- D-1/2/3 – Korjenovanje temeljeno na usporedbi sličnosti (odnosno udaljenosti) znakovnih nizova prema (Šnajder & Dalbelo Bašić, 2009). Za mjeru udaljenosti znakovnih nizova upotrijebljena je mjera D_4 dana izrazom (2.1), a grupiranje oblika provedeno je metodom prosječne udaljenosti (engl. *average linkage*), i to za tri odabrane vrijednosti praga: $t_1 = 0,54$ (D-1), $t = 0,86$ (D-2) i $t = 1,10$ (D-3). Prema ispitivanjima provedenima u (Šnajder & Dalbelo Bašić, 2009), s ovim se vrijednostima ostvaruje optimalna flektivna, derivacijska odnosno flektivno-derivacijska kakvoća normalizacije;

- HML – Lematizacija uporabom Hrvatskoga morfološkog leksikona (Tadić & Fulgosi, 2003). Budući da lematizacija pomoću HML-a, jednako kao i normalizacija pomoću leksikona L5 i L5-D1/D2/D3, homografe svodi na više mogućih lema, pri vrednovanju je proveden dodatni korak razrješavanja na način opisan u odjeljku 7.2.2.

Navedeni se postupci uvelike razlikuju prema količini upotrijebljenog lingvističkog znanja. Postupci korjenovanja P-n, S-1 i D-1/2/3 ne koriste nikakvo lingvističko znanje i zbog toga su potpuno jezično nezavisni.[7] Ostali postupci oslanjaju se na lingvističko znanje formalizirano morfološkim modelom odnosno morfološkim leksikonom hrvatskoga jezika. Postupci se također razlikuju i prema tome provode li normalizaciju na flektivnoj ili na derivacijskoj razini: postupci L5, S-1, S-2 i HML namijenjeni su normalizaciji na flektivnoj razini, postupci L5-D1, L5-D2 i L5-D3 namijenjeni su normalizaciji na flektivno-derivacijskoj razini, dok za postupake P-n i D-1/2/3 možemo reći da su flektivno-derivacijski zbog toga što niti ne razlučuju između ove dvije razine.

Rasprava rezultata

Rezultati intrinzičnog vrednovanja navedenih postupaka prikazani su u tablici 7.3. U tablici su prikazane samo najznačajnije mjere: za kakvoću flektivne normalizacije indikativna je mjera iSQ, a za kakvoću flektivno-derivacijske normalizacije mjere $idSQ$ i

[7] Time se ne želi reći da se ovi postupci mogu primijeniti na bilo koji jezik; sva tri postupka prikladna su naime samo za jezike konkatenativnih morfologija.

dSS. Rezultati su grupirani prema vrsti normalizacijskog postupka, a optimalne vrijednosti unutar svake grupe istaknuti su masnim slovima. Dodatno, na slici 7.3 prikazan je grafikon vrijednosti indekasa *UI*-*OI*, posebno na flektivnoj razini (indeksi *iUI* i *iOI*) i na flektivno-derivacijskoj razini (indeksi *iUI* i *dOI*). Crtkana linija opisuje vrijednosti koje ostvaruje najjednostavniji normalizacijski postupak odbacivanja slova P-n. Puna crta odgovara vrijednostima ostvarenima postupkom temeljenom na sličnosti znakova, izmjerenima za više vrijednosti praga udaljenosti, od kojih su tri optimalne vrijednosti (D-1, D-2 i D-3) posebno naznačene.

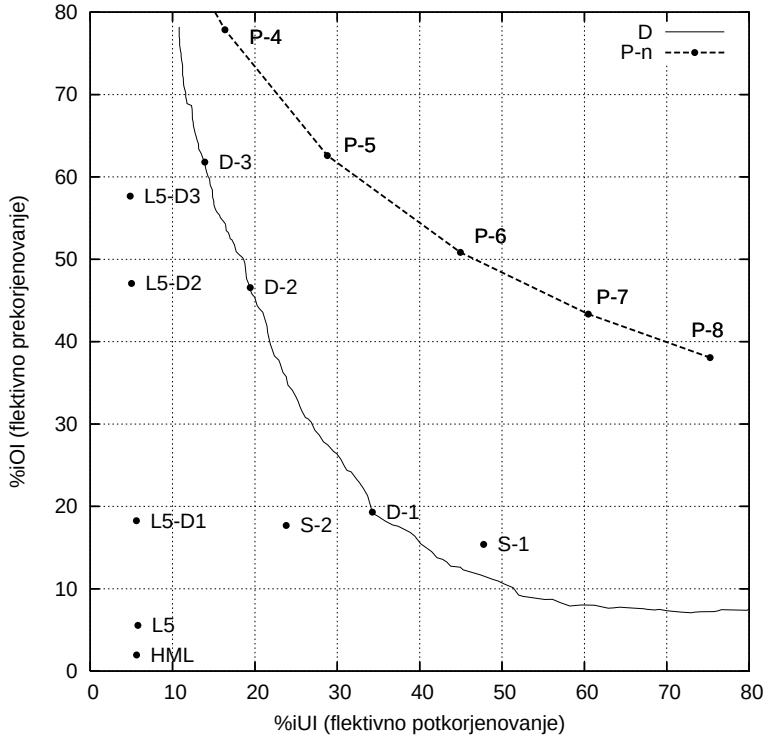
Općenito najbolja kakvoća normalizacije ostvaruje se lematizacijom pomoću HML, a zatim normalizacijom pomoću leksikona L5 (na flektivnoj razini) odnosno leksikona L5-D1 (na flektivno-derivacijskoj razini). Kakvoće normalizacije ovih triju postupaka približno su izjednačene, dok je kod svih ostalih postupaka kakvoća normalizacije znatno (barem 15%) lošija.

Na flektivnoj razini, kakvoća normalizacije HML-a iznosi oko 96%, dok kakvoća normalizacije pomoću leksikona L5 iznosi oko 94%. Pritom treba napomenuti da su pogreške prekorjenovanja kod leksikona L5 više izražene (više od 5%), dok su kod HML-a zanemarive (indeks prekorjenovanja iznosi 1,96%, no to je uglavnom zbog toga što su glagolski pridjevi trpni u uzorku smješteni u zasebne flektivne grupe, dok su kod HML-a oni svedeni pod oblike glagola). Od ostalih postupaka, na flektivnoj razini najbolju kakvoću normalizacije ostvaruje postupak odbacivanja sufiksa S-2 (više od 79%), a zatim jezično nezavisni postupak temeljen na sličnosti znakovnih nizova D-1 (više od 72%). Kakvoća normalizacije kod postupka odbacivanja sufiksa S-1 razmjerno je niska (oko 65%), i to ponajviše zbog velikog broja pogrešaka potkorjenovanja. Odsijecanje znakovnog niza nakon n-tog slova daje, očekivano, najlošije rezultate; najbolja kakvoća normalizacije ostvaruje se za $n = 6$ (oko 52%), ali sa stupnjem derivacijske normalizacije većim od 30%, što znači da normalizacija više nije čisto flektivna. Stupanj ostvarene derivacijske normalizacije ukazuje na to se kod flektivnih normalizacijskih postupaka HML, L5, S-1 i S-2 normalizacija uglavnom zadržava na flektivnoj razini.

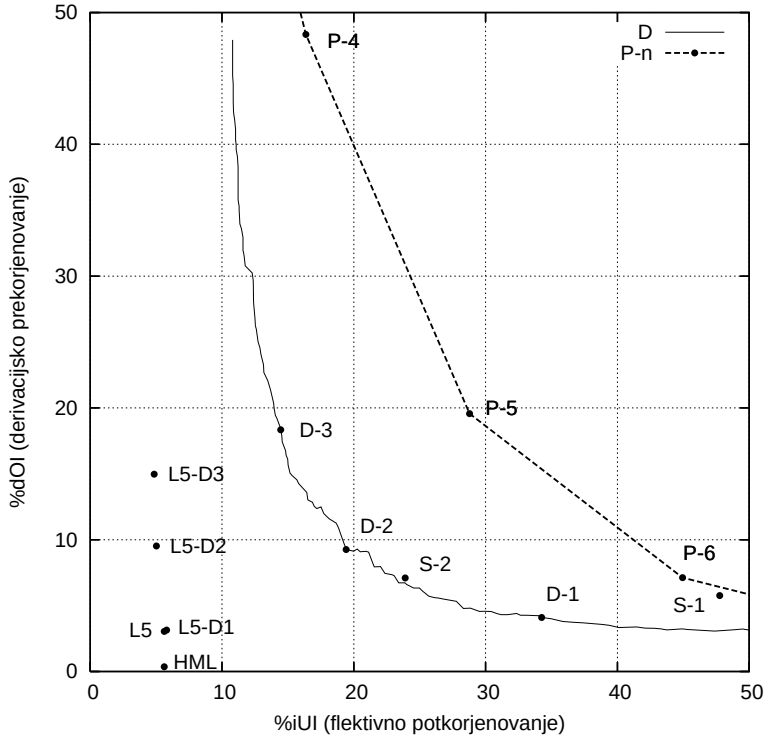
Na flektivno-derivacijskoj razini kakvoća normalizacije najbolja je kod HML-a, ali je – budući da je ipak riječ o flektivnome leksikonu – stupanj derivacijske normalizacije približno jednak nuli (stupanj derivacijske normalizacije nije jednak nuli zbog već spomenute razlike u tretiranju glagolskih pridjeva trpnih). Leksikonom L5-D1 ostvaruje se usporediva kakvoća flektivno-derivacijske normalizacije, međutim uz veći stupanj derivacijske normalizacije (više od 10%). Kod jezično nezavisnih postupaka, kakvoća flektivno-derivacijske normalizacije najbolja je za postupak D-2 (više od 85%), kojim se

Tablica 7.3: Kakvoća normalizacije temeljene na leksikonu u usporedbi s drugim postupcima normalizacije.

	Flektivna razina						
Postupak	% *iUI*	% *iOI*	% *iSQ*	% *dOI*	% *idSQ*	% *dSS*	% *ICF*
Normalizacija temeljem leksikona:							
L5	5,79	**5,56**	**94,32**	3,15	95,51	1,47	44,77
L5-D1	5,61	**18,25**	87,62	**3,04**	**95,66**	10,71	46,99
L5-D2	5,02	47,07	67,98	9,52	92,68	41,09	54,49
L5-D3	**4,85**	57,66	58,60	14,97	89,81	58,51	57,94
Prefiks znakovnoga niza:							
P-7	60,49	**43,40**	46,53	**2,53**	56,23	17,40	26,66
P-6	44,98	50,82	**51,94**	6,22	69,35	30,43	38,39
P-5	28,82	62,58	49,05	17,09	**76,60**	52,78	52,67
P-4	16,40	77,86	35,01	46,01	65,61	73,37	68,43
P-3	**7,45**	93,63	11,91	83,14	28,53	93,06	85,14
Odbacivanje sufiksa:							
S-1	47,76	15,39	64,60	5,76	67,22	3,63	30,60
S-2	**23,80**	**17,69**	**79,14**	6,84	**83,84**	6,13	40,78
Sličnost znakovnih nizova:							
D-1	34,25	**19,30**	**72,46**	**3,45**	78,23	7,88	40,01
D-2	19,42	46,55	64,27	8,22	**85,82**	35,26	52,95
D-3	**13,91**	61,79	52,93	18,34	83,82	59,70	61,93
Lematizacija:							
HML	**5,62**	**1,96**	**96,17**	**0,36**	**96,94**	0,94	41,93



(a)



(b)

Slika 7.3: Grafikon izmjerenih vrijednosti indeksa prekorjenovanja (UI) i potkorjenovanja (OI) za normalizaciju temeljenu na leksikonu i druge postupke normalizacije: (a) flektivna razina; (b) flektivno-derivacijska razina.

ostvaruje stupanj derivacijske normalizacije od oko 35%. Kakvoća flektivno-derivacijske normalizacije kod postupka odsijecanja najveća je za $n = 5$ te iznosi oko 77%, uz razmjerno visok stupanj derivacijske normalizacije od oko 53%.

7.2.6 Zaključci

Rezultati intrinzičnog vrednovanja opisani u prethodna dva odjeljka mogu se sažeti u sljedećim točkama:

1. Na flektivnoj razini kakvoća normalizacije pomoću pribavljenih flektivnih leksikona iznosi oko 95%, čime je gotovo izjednačena sa zlatnim standardom, odnosno lematizacijom pomoću HML-a;

2. Na flektivno-derivacijskoj razini najbolja kakvoća ostvaruje se flektivno-derivacijskim leksikonima kod kojih je grupiranje unosaka provedeno pomoću uzoraka iz skupa $\mathcal{D}_1$, uz stupanj derivacijske normalizacije od oko 10%;

3. Veći stupanj derivacijske normalizacije (do gotovo 60%) ostvaruje se flektivno-derivacijskim leksikonima kod kojih je za grupiranje unosaka korišteno više uzoraka, ali to povlači i blago smanjenje kakvoće normalizacije uslijed većeg broja pogrešaka prekorjenovanja;

4. Od jezično nezavisnih postupaka, najbolja se kakvoća normalizacije ostvaruje postupkom temeljenim na sličnosti znakovnih nizova, i ona na flektivno-derivacijskoj razini iznosi oko 86%;

5. Kakvoća normalizacije ostvariva odsijecanjem znakovnog niza iznosi na flektivno-derivacijskoj razini 77% i predstavlja donju granicu koju su nadmašili svi drugi razmatrani normalizacijski postupci.

Intrinzično vrednovanje pokazalo je da predloženi postupak normalizacije temeljen na morfološkom leksikonu pribavljenom iz korpusa dostiže kakvoću normalizacije ostvarivu pomoću ručno sastavljenog leksikona. Udio pogrešaka potkorjenovanja i prekorjenovanja iznosi razmjerno niskih 5%, što bi za većinu primjena u zadacima pretraživanja informacija i dubinske analize teksta trebalo biti i više nego dovoljno. Također je pokazano da je stupanj derivacijske normalizacije moguće povećati, a pritom ipak zadržati razmjerno visoku kakvoću normalizacije.

Poglavlje 8

Zaključak

Morfološka normalizacija postupak je strojne obrade teksta kojim se različite morfološke varijante jedne ili više riječi sažimlju na jedan jedinstveni oblik. Time se uklanjaju negativni učinci koje morfološka varijacija može imati na pretraživanje informacija i dubinsku analizu teksta, a koji se očituju u smanjenju preciznosti i odziva te povećanju dimenzija prostora značajki. Potreba za morfološkom normalizacijom naročito je izražena kod morfološki složenih jezika kao što je hrvatski. S druge strane, upravo zbog visoke morfološke složenosti, razvoj učinkovitog postupka morfološke normalizacije za hrvatski jezik zahtjevan je zadatak.

U okviru ovog rada razvijen je postupak za flektivnu i flektivno-derivacijsku normalizaciju tekstova na hrvatskome jeziku temeljen na morfološkom leksikonu. Polazna je pretpostavka bila da je za dosezanje visoke kakvoće normalizacije nužno razviti postupak koji se oslanja na jezično znanje. Zbog toga je najprije razvijen generativno-redukcijski model morfologije hrvatskoga jezika kojim je obuhvaćena fleksija i sufiksalna tvorba imenica, glagola i pridjeva. Model je inspiriran konceptima funkcijske programske paradigme, napose funkcijame višega reda kao načinu apstrakcije flektivnih i tvorbenih pravila jezika. Flektivna sastavnica modela upotrijebljena je zatim za akviziciju flektivnoga leksikona iz neoznačenog korpusa, čime je zaobiđen problem visoke cijene i dugog vremena razvoja tog jezičnog resursa. Budući da za morfološku normalizaciju, za razliku od lematizacije, nije presudno da leksikon bude lingvistički besprijekoran, postupak akvizicije moguće je provesti posve automatski. Derivacijska sastavnica modela korištena je za grupiranje tvorbeno povezanih unosaka flektivnoga leksikona. Na taj su način dobiveni flektivno-derivacijski leksikoni kojima se, ovisno o načinu grupiranja, ostvaruju različiti stupnjevi derivacijske normalizacije. Odabirom adekvatnog skupa derivacijskih uzoraka može se smanjiti broj pogrešaka normalizacije

značenjski nepovezanih riječi te stupanj normalizacije prilagoditi specifičnom zadatku.

U radu je predložen nov način vrednovanja intrinzične kakvoće normalizacijskog postupka koji omogućava da se kakvoća vrednuje zasebno na flektivnoj i na flektivno-derivacijskoj razini, te da se kvantificira ostvareni stupanj derivacijske normalizacije. Vrednovanje ukazuje na to da razvijeni normalizacijski postupak dostiže visoku intrinzičnu kakvoću. Na flektivnoj je razini kakvoća normalizacije usporediva s onom ručno sastavljenog leksikona. Na flektivno-derivacijskoj razini kakvoća normalizacije također je visoka, ali je primjetno da s porastom stupnja derivacijske normalizacije blago opada. Ispitani su i neki jezično nezavisni pristupi morfološkoj normalizaciji, za koje je utvrđeno da su lošije intrinzične kakvoće. To govori u prilog tezi da kvalitetna normalizacija morfološki složenog hrvatskoga jezika iziskuje neki oblik jezičnog znanja.

Ekstrinzično vrednovanje, koje je predmetom budućeg rada, trebalo bi pak odgovoriti na pitanja kolika je kakvoća normalizacije uopće potrebna te koji je stupanj derivacijske normalizacije optimalan za pojedine specifične zadatke. U tu će svrhu biti potrebno najprije izgraditi odgovarajuće ispitne zbirke, što je zahtjevan, ali nezaobilazan zadatak, koji je za mnoge druge jezike već odavno riješen.

Istraživanje opisano u ovome radu bilo je usredotočeno na hrvatski jezik. S druge strane, nastojao se ipak zadržati općenit, jezično nezavisan okvir kako bi se isti pristup u budućnosti mogao primijeniti i na druge, morfološki slične jezike.

Istraživanje je otvorilo i neke druge mogućnosti za daljnji rad. U kontekstu derivacijske morfološke normalizacije pojavljuje se niz interesantnih pitanja koja se dotiču leksičko-semantičke prirode tvorbenih veza. Buduće istraživanje trebalo bi detaljnije razmotriti odnos između leksičke semantike i derivacijske normalizacije, za što bi se mogao upotrijebiti neki od modela računalne leksičke semantike. Ti bi se modeli pri grupiranju unosaka flektivnoga leksikona mogli primijeniti na razini značenjskih skupina, derivacijskih uzoraka ili pojedinačnih tvorbeno povezanih riječi. Na taj bi se način mogla spriječiti normalizacija značenjski nepovezanih ili značenjski slabo povezanih riječi, što bi bez sumnje povećalo djelotvornost morfološke normalizacije.

Kod morfološki složenih jezika kao što je hrvatski na flektivnoj se razini normalizacije javlja problem homografije. Razmjeri homografije i njen utjecaj na djelotvornost morfološke normalizacije tekstova na hrvatskom jezika također bi mogli biti predmetom budućeg istraživanja. Takvo bi istraživanje trebalo usporediti postojeće postupke za razrješavanje homografije temeljene na kontekstno ovisnom označavanju vrste riječi s nekim jednostavnijim pristupima, koji bi se međutim u kontekstu morfološke normalizacije mogli pokazati zadovoljavajućima.

Nešto drugačiji pravac daljnjeg istraživanja tiče se primjene postupaka strojnog učenja kao alternative ručnoj izgradnji morfološkog modela. Ovdje se otvaraju barem tri mogućnosti. Prva se odnosi na primjenu metoda nadziranog učenja za indukciju flektivnih uzoraka temeljem morfološkog leksikona. Radovi iz tog područja doduše već postoje, ali bi se tom problemu u kontekstu morfološke normalizacije – za koju lingvistička ispravnost nije presudna – moglo pristupiti drugačije. Druga mogućnost tiče se primjene metoda nenadziranog učenja za indukciju derivacijskih uzoraka iz korpusa. Prethodna normalizacija pomoću flektivnog leksikona omogućila bi da se postupak usredotoči na isključivo derivacijsku razinu. Dobiveni derivacijski uzorci mogli bi se koristiti za nadopunu derivacijske sastavnice modela. Konačno, vrijedilo bi pri indukciji flektivnih i derivacijskih uzoraka pokušati inducirati i preoblike koje bi bile prikazane funkcijama višega reda. Na taj bi se način dobili morfološki modeli koji bi bili znatno sažetiji i razumljiviji nego da su pravila prikazana zamjenom sufikasa, kao što je to tipično slučaj. Navedeni pristupi omogućili bi da se dodatno pojednostavi razvoj postupka morfološke normalizacije za druge jezike.

Dodatak A

Programska izvedba

U ovom dodatku opisana je programska izvedba morfološkog modela, akvizicijskog algoritma i morfološkog leksikona u funkcijskome programskom jeziku Haskell. Funkcijski jezici pripadaju skupini deklarativnih programskih jezika temeljenih na lambda računu, formalnom modelu izračunljivosti funkcija.[1] Budući da su deklarativni, funkcijski jezici u načelu pružaju visoku razinu apstrakcije te omogućavaju pisanje vrlo sažetog programskog koda. Potencijal funkcijskih programskih jezika za rješavanje problema obrade prirodnog jezika otkriva se tek u novije vrijeme (Frost, 2006). Temeljni koncepti funkcijskih jezika jesu, pored ostalih, funkcija i funkcijska kompozicija, što te jezike čini idealnima za izvedbu morfološkog modela kakav je opisan u ovome radu.

U nastavku je dan vrlo kratak opis programskog jezika Haskell, u mjeri u kojoj je to potrebno da bi programski kôd izložen u nastavku bio razumljiv. Programska izvedba organizirana je hijerarhijski u module, koji su opisani u drugome dijelu ovog dodatka.

A.1 Programski jezik Haskell

Haskell[2] je moderan funkcijski jezik čiji je razvoj započeo krajem 80-ih godina u nastojanju da se objedine tada brojni divergentni pravci razvoja funkcijskih jezika. Zbog svoje elegantnosti, ali i činjenice da je jedan od rijetkih čisto funkcijskih jezika, Haskell je danas *de facto* standard funkcijskog programiranja, i to kako u u akademskim, tako i u industrijskim krugovima. Detaljna definicija programskog jezika Haskell može se naći u (Jones, 2003), dok dobar uvod u programiranje u Haskellu daju (Thompson, 1999) i (Hudak, 2000).[3]

[1] Izvrstan uvod u sustav lambda računa može se naći u (Hankin, 2004).
[2] Jezik je nazvan u čast logičaru Haskellu Curryju, poznatome po radu na kombinatoričkoj logici.
[3] Mnogi resursi povezani s Haskellom dostupni su na internetskoj stranici `http://www.haskell.org`.

Osnovne karakteristike programskog jezika Haskella su sljedeće:

- Haskell je čisti funkcijski jezik (engl. *purely functional*), što znači da pri izvođenju programa nije moguće nastupanje popratnih učinaka (engl. *side effects*). Zbog toga svi izrazi zadržavaju referencijalnu prozirnost,[4] a to onda smanjuje mogućnost pogrešaka, omogućava (formalno) rasuđivanje o programu te omogućava lakšu paralelizaciju programa.

- Haskell je također *lijeni* funkcijski jezik, odnosno jezik s lijenom (nestriktnom) evaluacijom (engl. *lazy evaluation*). Izračunavanje izraza odgađa se sve do trenutka dok vrijednost tog izraza nije doista potrebna, što u načelu omogućava učinkovitije izvođenje programa.

- Haskell je strogo i statički tipiziran (engl. *strongly and statically typed*). Sve vrijednosti moraju imati unaprijed određene tipove koji se moraju ispravno kombinirati. Tipski sustav oslanja se na mehanizam zaključivanja o tipovima (engl. *type inference*) koji se temelji na Hindley-Milnerovu algoritmu. Tipovi vrijednosti često se eksplicitno naznačuju. Npr. `s::String` naznačuje da je vrijednost `s` tipa `String` (znakovni niz), dok `f::String->(Int,Int)` naznačuje da je `f` funkcija koja znakovnom nizu pridružuje uređenu dvojku sastavljenu od dva cijela broja.

- Tipski sustav podržava polimorfizam. Polimorfizam obuhvaća slučajeve kada različite vrijednosti imaju sličnu podatkovnu strukturu. Polimorfni tipovi naznačeni su tzv. tipskim varijablama, koje se pišu malim slovima. Npr. `f::[a]->Int` naznačuje da je funkcija `f` polimorfnog tipa te da listi sačinjenoj od vrijednosti proizvoljnog tipa `a` pridružuje cijeli broj.

- Tipski sustav podržava i nadjačavanje (engl. *overloading*), kojim se obuhvaćaju slučajevi kada su nad različitim podatkovnim strukturama definirane slične operacije. Nadjačavanje se ostvaruje tzv. tipskim razredima (engl. *type classes*). Tipski razred predstavlja familiju tipova koji podržavaju neke definirane operacije. Npr. tipski razred `Eq` obuhvaća sve tipove koji podržavaju provjeru jednakosti, dok tipski razred `Ord` obuhvaća sve potpuno uređene tipove. Tipskim razredima ograničava se polimorfizam funkcija. Npr. `f::(Eq a)=>[a]->Int` naznačuje polimorfnu funkciju koja je primjenjiva samo na liste čiji elementi podržavaju provjeru jednakosti, odnosno samo na one elemente čiji je tip primjerak razreda `Eq`.

[4] Referencijalna prozirnost odnosi se na mogućnost zamjene istovrijednih izraza u svim kontekstima.

- Haskell podržava tzv. Curryjev oblik funkcije koji omogućava djelomičnu aplikaciju funkcija od više argumenata. Funkcija od više argumenata tipično se definira kao funkcija višeg reda koja se primjenjuje na jedan (prvi slijeva) argument, a rezultira novom funkcijom višeg reda koja se zatim primjenjuje na idući (drugi slijeva) argument, itd. Primjerice, umjesto funkcije `f::(a,b,c)->d`, koja preslikava s trojki tipa `(a,b,c)` na vrijednosti tipa `d`, u Haskellu se tipično definira funkcija `f::a->(b->(c->d))`, odnosno kraće `f::a->b->c->d`, budući da je operator `->` definiran kao desno asocijativan.

- Kada je to ipak potrebno, programiranje s popratnim učincima može se ostvariti bez narušavanja referencijalne prozirnosti primjenom tzv. monada. Tipičan primjer monade jest ulazno-izlazna monada `IO`, koja omogućava izvođenje ulazno-izlaznih operacija. Funkcije koje se izvode unutar monade imaju to eksplicitno naznačeno u svome tipu. Npr. `f::String->IO Int` naznačuje funkciju koja znakovnom nizu pridružuje cijeli broj, ali se izvodi unutar monade `IO`, što znači da dodatno obavlja neku ulazno-izlaznu operaciju. U tom slučaju kažemo da je vrijednost tipa `Int` *omotana* u monadu `IO`.

Popularnosti Haskella zasigurno je pridonio i izvrstan, javno dostupan programski prevodilac GHC (Glasgow Haskell Compiler).[5] GHC je optimizirajući prevodilac, ali se također može koristiti i kao interpreter u interaktivnom načinu rada, što bitno olakšava razvoj programa. Za razvoj i ispitivanje programskog kôda opisanog u nastavku korišten je upravo GHC.

A.2 Moduli

Programski kôd hijerarhijski je organiziran u module. Svaki modul definira sučelje u vidu funkcija, tipova i tipskih razreda koji su vidljivi izvan modula. Razvijena su tri osnovna modula: `MorphModel` (apstraktni morfološki model), `Hofm` (morfološki model temeljen na funkcijama višeg reda) te `Molex` (morfološki leksikon). U nastavku su opisana sučelja ovih modula i njihovih podmodula.

A.2.1 Modul `MorphModel`

Modul `MorphModel` definira apstraktni morfološki model opisan u odjeljku 3.1. Flektivna sastavnica modela definirana je tipskim razredima `IP` (generativni dio) i `IPR` (re-

[5] `http://www.haskell.org/ghc/`

dukcijski dio). Ovim tipskim razredima definiran je zapravo flektivni uzorak i funkcije koje se na njega primjenjuju.

Tipski razred `IP` definiran je na sljedeći način:

```
class (Eq ip, Eq x, Labeled ip) => IP ip x | ip -> x where
  sWfs    :: ip -> String -> [String]
  sWfsMsd :: ip -> String -> [(String,x)]
  sLemma  :: (MonadPlus m) => ip -> String -> m String
  msds    :: ip -> [x]
```

Ovime je definiran dvoparametarski razred `IP ip x`, gdje je `ip` tipska varijabla uzorka, a `x` tipska varijabla morfosintaktičkog opisa. Oba ova tipa moraju pripadati razredu `Eq`, dok `ip` ujedno mora pripadati i razredu `Labeled`, odnosno mora imati definiranu oznaku. Da bi dakle neki tip pripadao razredu `IP`, on mora implementirati navedene četiri funkcije. Posljednja funkcija, funkcija `msds`, uzorku pridružuje skup morfosintaktičkih opisa koji se u njemu koriste. Prve tri funkcije, `sWfs`, `sWfsMsd` i `sLemma`, odgovaraju onima definiranima izrazima (3.1)–(3.3), uz napomenu da je redoslijed argumenata zamijenjen kako bi se omogućilo jednostavnije definiranje kompozicije funkcija. Primjerci razreda `IP` moraju definirati samo funkcije `sLemma`, `sWfsMsd` i `msds`, dok funkciju `sWfs` nije potrebno eksplicitno definirati budući da za nju postoji podrazumijevana definicija prema (3.4).

Valja napomenuti da kodomenu funkcije `sLemma` čine znakovni nizovi omotani unutar monade `MonadPlus`. Monada, odnosno tipski razred `MonadPlus`, obuhvaća dva tipa: listu i algebarski tip `Maybe`. Listom je moguće prikazati skup mogućih rješenja, dok je tipom `Maybe` moguće prikazati jedno ili nijedno rješenje, ovisno o ishodu izračuna. Na taj je način osigurano da funkcija `wLemma`, u slučaju višeznačnosti, može rezultirati skupom (listom) rješenja, ukoliko je to potrebno.

Kod redukcijskog dijela modela flektivni uzorak `ip` mora biti primjenjiv i u obratnome smjeru. To je obuhvaćeno tipskim razredom `IPR`, definiranim na sljedeći način:

```
class (IP ip x) => IPR ip x where
  lStem     :: (MonadPlus m) => ip -> String -> m String
  wStem     :: (MonadPlus m) => ip -> String -> m String
  wStemMsd  :: (MonadPlus m) => ip -> String -> m (String,x)
```

Tipski razred `IPR` proširuje (nasljeđuje) razred `IP`, tj. tip `ip` može biti članom tipskog razreda `IPR` samo ako je član tipskog razreda `IP`. Razred `IPR` odgovara dakle generativno-redukcijskom modelu koji generativni model proširuje trima funkcijama (3.5)–(3.7). Za funkciju `wStem` nije potrebna eksplicitna definicija budući da postoji podrazumijevana definicija (temeljem funkcije `wStemMsd`). Sve tri funkcije mogu, po

potrebi, dati više rezultata, pa su njihove povratne vrijednosti omotane u monadu `MonadPlus`.

Za generativno-redukcijski model, odnosno za tipove iz razreda `IPR`, u modelu su definirane sljedeće funkcije:

```
lWfs      :: (IPR ip x) => ip -> String -> [String]
lWfsMsd   :: (IPR ip x) => ip -> String -> [(String,x)]
wLemma    :: (MonadPlus m, IPR ip x) => ip -> String -> m String
wLemmaMsd :: (MonadPlus m, IPR ip x) => ip -> String -> m (String,x)
```

Navedene funkcije omogućavaju generiranje oblika i lematizaciju, i to s lemom kao ishodišnim oblikom.

Lematizacija temeljem skupa (liste) flektivnih uzoraka `[ip]` ostvaruje se funkcijama:

```
lm    :: (IPR ip x, MonadPlus m) => [ip] -> String -> m (String,ip)
lmMsd :: (IPR ip x, MonadPlus m) => [ip] -> String -> m (String,x,ip)
```

Derivacijska sastavnica modela definirana je tipskim razredom `DP`, koji odgovara derivacijskom uzorku. Taj je tipski razred definiran kao:

```
class (Labeled dp, IPR ip x) => DP dp ip x y where
  lDerive :: (MonadPlus m) => dp -> (String,ip) -> m (String,y)
```

Razred `DP dp ip x y` je četveroparametarski razred: `dp` je tipska varijabla derivacijskog uzorka, `ip` je tipska varijabla flektivnog uzorka, `x` je tipska varijabla morfosintaktičkog opisa, dok je `y` tipska varijabla značenjske kategorije izvedenice. Tipski razred deklarira funkciju `lDerive`, sukladnu onoj definiranoj izrazom (3.29), koja na temelju LU-para izvodi listu parova izvedenice i pripadne značenjske kategorije (ili jedan takav par, ovisno o tome koji se primjerak razreda `MonadPlus` koristi).

Tri opisana tipska razreda, `IP`, `IPR` i `DP`, definiraju apstraktno sučelje koje je konkretizirano (instancirano) u modulima `Hofm.IPattern` odnosno `Hofm.DPattern`.

A.2.2 Modul `MorphModel.MsdProb`

Podmodul `MorphModel.MsdProb` definira funkcionalnost za izračun razdiobe oblika flektivnih uzoraka temeljem korpusa. Ta je razdioba potrebna za vjerojatnosne mjere ocjene LU-parova, kako je opisano u odjeljku 5.3.3.

Korpus je prikazan kao multiskup znakovnih nizova:

```
type Corpus = Data.Set.MultiSet String
```

Uvjetnoj vjerojatnosti $P(x|f)$ odgovara funkcija:

```
type MsdProb ip x = ip -> x -> Double
```

koja se, temeljem uzorka LU-parova, izračunava pomoću sljedeće funkcije višega reda:

```
msdProb ::(IPR ip x, Ord ip) =>
  (ip -> Int) -> Corpus -> (ip -> x -> x) -> [(String,ip)] -> MsdProb ip x
```

Prvi argument jest funkcija tipa `ip->Int` koja flektivne uzorke grupira prema morfološkoj vrsti. Drugi je argument korpus temeljem kojega se izračunavaju vjerojatnosti. Treći je argument funkcija tipa `ip->x->x`, kojom se morfološki opisi skraćuju, odnosno kojom se odbacuju vrijednosti gramatičkih kategorija koje se pri izračunu razdiobe trebaju zanemariti. Četvrti argument, lista LU-parova, jest uzorak za koji se razdioba izračunava.

A.2.3 Modul `Hofm`

Modul `Hofm` definira morfološki model temeljen na funkcijama višeg reda, kako je opisano u odjeljku 3.2. Modul sačinjavaju tri podmodula: `Hofm.Transf` (preoblike znakovnog niza), `Hofm.IPattern` (flektivni uzorak) te `Hofm.DPattern` (derivacijski uzorak).

A.2.4 Modul `Hofm.Transf`

Podmodul `Hofm.Transf` definira apstraktno sučelje preoblike znakovnog niza, i to putem tipskog razreda `Transf` definiranog kako slijedi:

```
class Transf t where
  ($$) :: (MonadPlus m) => t -> String -> m String
  (&)  :: t -> t -> t
  rsfx :: String -> String -> t
  rpfx :: String -> String -> t
  rifx :: String -> String -> t
  nul  :: t
  fail :: t
```

Razred deklarira funkcije koje svaka preoblika tipa `t` mora podržati. Funkcija (odnosno infiksni operator) `$$` jest operator primjene (aplikacije) preoblike na znakovni niz. Rezultat operatora omotan je u monadu `MonadPlus`, što znači da preoblika može biti višeznačna i rezultirati skupom mogućih znakovnih nizova. Operator `&` definira funkcijsku kompoziciju dviju preoblika, prema (3.33). Funkcije višeg reda `rsfx`, `rpfx` i `rifx` odgovaraju zamjeni sufiksa, prefiksa i infiksa znakovnog niza, prema (3.51), (3.52) odnosno (3.59).

Za tipove razreda `Transf` u modulu su definirane sljedeće funkcije višeg reda:

```
sfx  :: (Transf t) => String -> t
pfx  :: (Transf t) => String -> t
dsfx :: (Transf t) => String -> t
dpfx :: (Transf t) => String -> t
```

sukladno definicijama (3.55)–(3.58).

Preoblike koje podržavaju opcionalnost obuhvaćene su tipskim razredom `OptTransf`, koji proširuje razred `Transf`, kako slijedi:

```
class (Transf t) => OptTransf t where
  (.|.)  :: t -> t -> t
  (.||.) :: t -> t -> t
```

Funkcija `.|.` jest operator ravnopravnog odabira definiran izrazom (3.35), dok je funkcija `.||.` operator pristranog odabira definiran izrazom (3.36). Za preoblike koje podržavaju opcionalnost definirane su također, prema definicijama (3.61) i (3.62), sljedeće funkcije višeg reda:

```
asfx :: (OptTransf t) => [(String,String)] -> t
aifx :: (OptTransf t) => [(String,String)] -> t
```

Funkcije `opt` i `try` definirane su, sukladno (3.43) odnosno (3.44), na sljedeći način:

```
opt :: (OptTransf t) => t -> t
opt = (.|. nul)

try :: (OptTransf t) => t -> t
try = (.||. nul)
```

Konačno, tip preoblika koji podržava izračunavanje inverza preoblike obuhvaćen je tipskom klasom `InvTransf`, definiranom kao:

```
class (Transf t) => InvTransf t where
  inv :: t -> t
```

Razredom je deklarirana samo jedna funkcija, `inv`, koja izračunava inverz preoblike `t`.

A.2.5 Modul `Hofm.Transf.TTransf`

Podmodul `Hofm.Transf.TTransf` definira konkretnu implementaciju preoblike znakovnog niza koja zadovoljava gore opisana apstraktna sučelja. To je ostvareno na način da je najprije definiran novi tip `TTransf`, a zatim su nad njime definirane funkcije tipskih razreda `Transf`, `OptTransf` i `InvTransf`, čime taj tip postaje primjerkom tih razreda.

Budući da je preoblika znakovnog niza u stvari funkcija, bilo bi najjednostavnije implementirati je izravno kao funkciju programskoga jezika Haskell. Takvo rješenje međutim nije zadovoljavajuće jer ne omogućava izračunavanje inverza preoblike. Drugim riječima, za tako definiranu preobliku ne bi bilo moguće definirati funkciju `inv`, pa takva preoblika ne bi mogla biti primjerkom razreda `InvTransf`. To bi onda, u skladu s napomenama iz odjeljka 3.1.1, značilo da morfološki model ne može biti redukcijski.

Ovaj je problem ovdje riješen tako da je funkcija preoblike prikazana posredno, pomoću odgovarajuće podatkovne strukture. Konkretno, preoblika tipa `TTransf` (od engl. *tree-transform*) prikazana je podatkovnom strukturom binarnoga stabla. Stablo odgovara strukturi izraza kojim je preoblika definirana: čvorovi stabla odgovaraju binarnim operatorima '|', '||', '∘', dok listovi stabla odgovaraju osnovnim operacijama nad znakovnim nizovima. Algebarski tip `TTransf` definiran je rekurzivno na sljedeći način:

```
data TTransf = Else TTransf TTransf
             | Ilse TTransf TTransf
             | Or   TTransf TTransf
             | Comp TTransf TTransf
             | Tip  Op
             | Fail
             deriving (Eq,Show)
```

gdje su `Else`, `Or` i `Comp` tzv. tipski konstruktori koji naznačuju vrstu čvora. Čvor `Ilse` je pomoćni čvor koji se koristi za prikaz invertiranog čvora `Else`. List stabla može biti ili temeljna operacija nad znakovnim nizom (tipski konstruktor `Tip`) ili čvor `Fail`, koji označava preobliku koja uvijek zakazuje.

Osnovne operacije nad znakovnim nizom, kako je pojašnjeno u odjeljku 3.3.6, jesu zamjena prefiksa, sufiksa i infiksa. Ove su operacije obuhvaćene algebarskim tipom `Op`, definiranim na sljedeći način (tipski konstruktor `NO` koristi se za prikaz nul-preoblike):

```
data Op = RP  String String
        | RS  String String
        | RI  String String
        | NO
        deriving (Eq,Show)
```

Primjerice, preoblika $sfx(a)$ prikazana je kao:

```
Tip (RS "" "a")
```

Primjer složenije preoblike jest preoblika $sfx(ača) \circ try(jat_1)$, koja je prikazana kao:

```
Comp (Tip (RS "" "ača"))
     (Else (Tip (RI "ije" "je"))
           (Tip NO))
```

ili preoblika $sfx(i) \circ sbl$, koja je prikazana kao:

```
Comp (Tip (RS "" "i"))
     (Or (Or (Tip (RS "k" "c"))
             (Tip (RS "h" "s")))
         (Tip (RS "g" "z")))
```

Nad ovako definiranom podatkovnom strukturom funkcija `inv` definirana je na sljedeći način. Za unutarnje čvorove funkcija je definirana rekurzivno: za čvor tipa `Or` prema izrazu (3.41), za čvor tipa `Else` prema izrazu (3.42), a za čvor tipa `Comp` sukladno izrazu $(t_1 \circ t_2)^{-1} = t_2^{-1} \circ t_1^{-1}$. Pri invertiranju se čvor tipa `Else` zamjenjuje čvorom tipa `Ilse`, i obrnuto. To je potrebno kako bi se osigurala ispravna primjena invertirane preoblike, sukladno (3.42). Za čvorove listove fukcija `inv` definirana je sukladno izrazima (3.53), (3.54) i (3.60), ovisno o operaciji. Na primjer (primjer interakcije u interpreteru GHC-a):

```
> inv(sfx "a") :: TTransf
Tip (RS "a" "")
> inv(sfx "ača" & try(jat1)) :: TTransf
Comp (Ilse (Tip (RI "je" "ije")) (Tip NO)) (Tip (RS "ača" ""))
```

Operator primjene preoblike na znakovni niz, `$$`, definiran je također rekurzivno nad strukturom binarnoga stabla, i to za unutarnje čvorove sukladno (3.33), (3.35), (3.36) i (3.42), a za listove stabla sukladno (3.51), (3.52) i (3.59). Primjenu preoblika ilustriraju sljedeći primjeri:

```
> (sfx "i" & sbl :: TTransf) $$ "slik" :: [] String
["slici"]
> (sfx "ača" & try(jat1) :: TTransf) $$ "cvijet" :: Maybe String
Just "cvjetača"
```

Budući da je povratna vrijednost operatora `$$` omotana u tip razreda `MonadPlus`, konkretan tip povratne vrijednosti (koji može biti ili lista ili vrijednost tipa `Maybe`) potrebno je eksplicitno naznačiti (u programskom kodu to je rijetko potrebno budući da se tip povratne vrijednosti može automatski izvesti temeljem konteksta). Primjenu inverza preoblike ilustriraju sljedeći primjeri:

```
> inv(sfx "i" & sbl :: TTransf) $$ "slici" :: [] String
["slik"]
> inv(sfx "ača" & try(jat1) :: TTransf) $$ "cvjetača" :: [] String
["cvijet","cvjet"]
```

U drugom slučaju, budući da je preoblika `try` neinjektivna, to je sukladno izrazu (3.45) inverz preoblike višeznačan.

A.2.6 Modul `Hofm.Cond`

Modul `Hofm.Cond` definira uvjetnu funkciju i funkcije višeg reda koje se koriste za definiranje uvjetnih funkcija. Uvjetna funkcija implementirana je izravno kao funkcija programskoga jezika Haskell (to je moguće budući da nije potrebno izračunavati njezin inverz):

```
type Cond = String -> Bool
```

Za logičko kombiniranje uvjeta definirane su sljedeće funkcije:

```
land :: Cond -> Cond -> Cond
lor  :: Cond -> Cond -> Cond
neg  :: Cond -> Cond
```

Funkcije višeg reda za definiranje uvjetnih funkcija su sljedeće:

```
always :: Cond
ends   :: [String] -> Cond
starts :: [String] -> Cond
nends  :: [String] -> Cond
```

A.2.7 Modul `Hofm.IPattern`

Modul `Hofm.IPattern` implementira apstraktna sučelja razreda `IP` i `IPR`, odnosno definira tip flektivnog uzorka kao konkretan primjerak tih tipskih razreda. Flektivni uzorak predstavljen je algebarskim tipom `IPattern`, definiranim kako slijedi:

```
type Label = String
type TransfMsd t x = (t,[x])
data IPattern t x = IPattern Label Cond [TransfMsd t x]
```

Tip `IPattern t x` parametriziran je tipskom varijablom `t`, koja određuje tip preoblike koji se u uzorku koristi, te tipskom varijablom `x`, koja određuje tip morfosintaktičkog opisa. Sama podatkovna struktura `IPattern` sastoji se od triju vrijednosti: oznake (naziva uzorka), uvjetne funkcije te liste parova preoblika i morfosintaktičkih opisa. Ova struktura odgovara onoj danoj definicijom (3.31).

Tip `IPattern` definiran je kao primjerak razreda `IP` (puna definicija je izostavljena):

```
instance (Transf t,Msd x) => IP (IPattern t x) x where
  sWfsMsd ...
  sLemma ...
  msds ...
```

Kako bi tip `IPattern` mogao biti primjerkom razreda `IPR`, koji odgovara generativno-redukcijskome modelu, preoblika `t` mora pripadati razredu `InvTransf`, odnosno njezin inverz mora biti izračunljiv. Prema tome (puna definicija je izostavljena):

```
instance (InvTransf t, Msd x) => IPR (IPattern t x) x where
  lStem ...
  wStemMsd ...
```

Tipski razred `Msd` obuhvaća tipove koji se mogu koristiti za prikaz morfosintaktičkih opisa:

```
class (Eq x) => Msd x where
  combine :: x -> x -> x
```

Razred dakle deklarira samo jednu funkciju, `combine`, koja se kod definiranja flektivnih uzoraka koristi za kombiniranje djelomično zadanih morfosintaktičkih opisa. Prikaz morfosintaktičkih opisa u načelu je proizvoljan (pod uvjetom da tip pripada razredu `Eq`). U modulu je kao osnovni tip za prikaz morfosintaktičkih opisa predefiniran tip `String`, pa se dakle morfosintaktički opisi mogu prikazivati kao znakovni nizovi, što je sukladno normi MULTEX-East (v. odjeljak 4.2.4). U tom slučaju kombiniranje morfosintaktičkih opisa definirano je kao povezivanje (konkatenacija) znakovnih nizova, pri čemu se, dodatno, na mjestima koja su u drugom znakovnom nizu označena točkom (`.`) umeću, slijedno slijeva nadesno, znakovi drugog znakovnog niza. Primjerice:

```
> combine "N-m" "sn"
"N-msn"
> combine "Af....." "p...n"
"Afp...n"
```

Ovime se osigurava da vrijednosti pojedinih morfosintaktičkih kategorija u kombiniranom znakovnom nizu završe na ispravnim pozicijama (u normi MULTEX-East kategorija je određena položajem u znakovnome nizu).

Podrazumijevani tip flektivnoga uzorka definiran je na sljedeći način:

```
type IPatternDefault = IPattern TTransf String
```

Podrazumijevani tip koristi, dakle, preobliku tipa `TTransf`, dok za morfosintaktičke opise koristi znakovne nizove. Budući da tip `TTransf` pripada razredu `InvTransf`, to tip `IPatternDefault` pripada kako razredu `IP`, tako i razredu `IPR`.

Kako bi se pojednostavilo definiranje flektivnih uzoraka, definiran je sljedeći infiksni operator:

```
(#) :: (Transf t) => t -> [x] -> TransfMsd t x
```

Operator se može upotrijebiti za konstrukciju parova sastavljenih od preoblike `t` i (djelomično zadanih) morfosintaktičkih opisa iz liste `[x]`.

Također u svrhu pojednostavljivanja definiranja flektivnih uzoraka, u modulu je definirana sljedeća funkcija:

```
iPattern :: (Transf t, Msd x) =>
  Label -> x -> Cond -> [TransfMsd t x] -> IPattern t x
```

Funkcija `iPattern` istovjetna je tipskom konstruktoru `IPattern`, s tom razlikom da uzima još jedan dodatni argument: djelomični morfosintaktički opis koji je zajednički svim oblicima koje uzorak izvodi, a koji se kombinira sa specifičnijim morfosintaktičkim opisima koji su pridruženi pojedinim preoblikama.

Na primjer, flektivni uzorak (3.19) definiran je na sljedeći način:

```
n04 = iPattern "N04" "N-m"
  (ends velars `land` (neg(ends cgr)))
  [nul              # ["sn"],
   sfx "a"          # ["sg","sa","pg"],
   sfx "u"          # ["sl"],
   sfx "e" & plt    # ["sv"],
   sfx "om"         # ["si"],
   sfx "i" & sbl    # ["pn","pv"],
   sfx "ima" & sbl  # ["pd","pl","pi"],
   sfx "e"          # ["pa"]]
```

Ovim je uzorkom definirano 13 različitih parova preoblika i morfosintaktičkih opisa, koji se mogu izvesti primjenom funkcija `sWfsMsd` ili `lWfsMsd`. Na primjer, oblici leme *vojnik* jesu:

```
> lWfsMsd (n04::IPatternDefault) "vojnik"
[("vojnik","N-msn"),("vojnika","N-msg"),("vojnika","N-msa"),
("vojnika","N-mpg"),("vojniku","N-msl"),("vojniče","N-msv"),
("vojnikom","N-msi"),("vojnici","N-mpn"),("vojnici","N-mpv"),
("vojnicima","N-mpd"),("vojnicima","N-mpl"),("vojnicima","N-mpi"),
("vojnike","N-mpa")]
```

Svođenje oblika na osnovu ili lemu ostvaruje funkcija `wStemMsd` odnosno `wLemmaMsd`. Primjerice:

```
> wStemMsd (n04::IPatternDefault) "vojniče" :: Maybe (String,String)
Just ("vojnik","N-msv")
> wStemMsd (n04::IPatternDefault) "vojniki" :: Maybe (String,String)
Nothing
> wLemmaMsd (n04::IPatternDefault) "bubrezima" :: [] (String,String)
[("bubreg","N-mpd"),("bubreg","N-mpl"),("bubreg","N-mpi")]
```

Za definiranje flektivnih uzoraka koji su sastavljeni od poduzoraka, u modulu su definirana sljedeća dva pomoćna operatora:

```
(<&) :: (Transf t,Msd x) => [TransfMsd t x] -> t -> [TransfMsd t x]
(<#) :: (Transf t,Msd x) => [TransfMsd t x] -> [x] -> [TransfMsd t x]
```

Operator `<&` distribuira zadanu preobliku na listu parova preoblika i morfosintaktičkih opisa, čime se dobiva kompozicija preoblika u kojoj se pridodana preoblika nalazi sdesna operatoru kompozicije. Operator `<#` distribuira zadanu listu morfosintaktičkih opisa te ih kombinira s postojećim opisima. Prvi se operator koristi za primjenu preoblike na poduzorak, a drugi za specijaliziranje morfosintaktičkih opisa u poduzorku. Primjerice:

```
[sfx "i" # ["msn","msa"], sfx "og" # ["msg","nsg"]] <& sfx "ij" <# ["c..."]
```

ekvivalentno je s:

```
[sfx "i" & sfix "ij" ["cmsn","cmsa"], sfx "og" & sfix "ij" ["cmsg","cnsg"]]
```

Na primjer, flektivni uzorak dan izrazom (3.20) može se definirati na sljedeći način:

```
a01 = iPattern "A01" "Af....."
  (ends nonpals `land` (neg(ends cgr) `lor` ends ["st","št"]))
  (ai01 <# ["p...n"] ++
   ad01 <# ["p...y"] ++
   ad02 <& tc <# ["c...-"] ++
   ad02 <& tc <& ts <# ["s...-"])
  where tc = jot
        ts = pfx "naj"
```

gdje je `++` Haskellov operator povezivanja (konkatenacije) listi, dok su `ai01`, `ad01` i `ad02` poduzorci (liste preoblika) definirani sukladno (3.21). Primjerice, oblici pridjeva *brz* su sljedeći:

```
> lWfs (a01::IPatternDefault) "brz"
["brz","brza","brzu","brzim","brzi","brzih","brzima","brze","brzo","brzoj",
"brzom","brzog","brzoga","brzome","brzomu","brži","bržeg","bržega","bržem",
"bržom","bržemu","bržim","bržih","bržima","brže","brža","bržoj","bržu",
"najbrži","najbržeg","najbržega","najbržem","najbržom","najbržemu",
"najbržim","najbržih","najbržima","najbrže","najbrža","najbržoj","najbržu"]
```

Višeznačnost svođenja na lemu ilustrira sljedeći primjer (prema (3.27)):

```
> wLemmaMsd (a01::IPatternDefault) "bržeg" :: [] (String,String)
[("bržeg","Afpmsnn"),("bržeg","Afpmsan"),("bržeg","Afpmsvn"),
("brz","Afcmsg-"),("brz","Afcnsg-")]
```

A.2.8 Modul `Hofm.DPattern`

Modul `Hofm.DPattern` implementira apstraktno sučelje tipskog razreda `DP`, odnosno definira konkretan primjerak tog razreda. Derivacijski uzorak predstavljen je algebarskim tipom `DPattern`, definiranim na sljedeći način:

```
type Label = String
data DPattern t ip = DPattern Label t [ip] [ip]
```

Tip `DPattern t ip` parametriziran je tipskom varijablom `t`, koja određuje tip preoblike, te tipskom varijablom `ip`, koja određuje tip flektivnog uzorka. Podatkovna struktura `DPattern` sastoji se od četiri vrijednosti: oznake uzorka, preoblike koja obličnu osnovu preobličuje u lemu izvedenice te dviju listi flektivnih uzoraka koje odgovaraju kategorijama osnovne riječi i izvedenice.

Tip `DPattern` definiran je kao primjerak razreda `DP` (puna definicija je izostavljena):

```
instance (Transf t, IPR ip x) => DP (DPattern t ip) ip x [ip] where
  lDerive ...
```

Kako bi tip `DPattern` mogao biti primjerkom razreda `DP`, flektivni uzorak mora pripadati razredu `IPR`, tj. flektivna sastavnica modela podržavati redukcijski smjer izvođenja. To je potrebno zato što funkcija `lDerive` kao ishodišni oblik uzima lemu, dok je preoblika `t` definirana u odnosu na obličnu osnovu, pa je dakle lemu potrebno najprije svesti na obličnu osnovu, sukladno (3.29) odnosno (3.50).

Podrazumijevani tip derivacijskog uzorka definiran je na sljedeći način:

```
type DPatternDefault = DPattern TTransf IPatternDefault
```

Primjer derivacijskog uzorka jest sljedeći:

```
let d = DPattern "iu13"
        (sfx "ina" & try jat1 & try plt) mNouns fNouns
```

Primjenom funkcije `lDerive` na uzorak i lemu osnovne riječi dobivaju se odgovarajuće izvedenice i pripadne kategorije. Primjerice:

```
> lDerive d ("junak",n04::IPatternDefault) :: [] (String,[IPatternDefault])
[("junačina",[N28,N47])]
> lDerive d ("svijet",n04::IPatternDefault) :: [] (String,[IPatternDefault])
[("svjetina",[N28,N47])]
```

A.2.9 Modul `Molex`

Modul `Molex` implementira morfološki leksikon. Morfološki je leksikon predstavljen apstraktnim podatkovnim tipom `Molex ip`, koji je parametriziran tipom flektivnog uzorka `ip`.

Osnovne operacije nad morfološkim leksikonom su sljedeće:

```
readMolex  :: (IPR ip x) => [ip] -> FilePath -> IO (Molex ip)
writeMolex :: (IPR ip x) => Molex ip -> FilePath -> IO ()
size       :: Molex ip -> Int
lemmas     :: Molex ip -> [String]
wordforms  :: Molex ip -> [String]
patterns   :: (IPR ip x) => Molex ip -> [ip]
```

Funkcije `readMolex` i `writeMolex` služe za učitavanje leksikona iz datoteke odnosno njegovo pohranjivanje. Funkcija `size` vraća broj unosaka leksikona. Funkcije `lemmas`, `wordforms` i `patterns` vraćaju listu lema, listu oblika odnosno listu flektivnih uzoraka sadržanih u leksikonu.

Modul definira sljedećih sedam funkcija za generiranje oblika, lematizaciju i normalizaciju temeljem flektivnog leksikona:

```
wfs      :: (IPR ip x) => Molex ip -> String -> [String]
wfsMsd   :: (IPR ip x) => Molex ip -> String -> [(String, x)]
lemma    :: (MonadPlus m, IPR ip x) => Molex ip -> String -> m String
lemmaMsd :: (MonadPlus m, IPR ip x) => Molex ip -> String -> m (String,x)
lm       :: (IPR ip x, MonadPlus m) => Molex ip -> String -> m (String,ip)
lmMsd    :: (IPR ip x, MonadPlus m) => Molex ip -> String -> m (String,x,ip)
norm     :: (MonadPlus m, IPR ip x) =>
              Molex ip -> (Entry ip -> String) String -> m String
```

Funkcije `lemma`, `lemmaMsd` i `norm` odgovaraju funkcijama (6.1), (6.2) odnosno (6.3). Funkcije `wfsMsd` i `wfs` generiraju oblike zadane leme analogne su funkcijama (3.46) i (3.47) morfološkog modela.

A.2.10 Modul `Molex.Acquisition`

Podmodul `Molex.Acquisition` definira funkciju za akviziciju flektivnog leksikona iz korpusa, prema tablici 5.1. Tip funkcije je sljedeći:

```
acquire :: (IPR ip x) => AcqParams ip -> [ip] -> Corpus -> Molex ip
```

Akvizicijski parametri definirani su podatkovnom strukturom `AcqParams`, na sljedeći način:

```
data AcqParams ip = AcqParams {
  score  :: Score ip,
  wford  :: (String,Int) -> (String,Int) -> Ordering
  accept :: Corpus -> ((String,ip),Double) -> Bool,
  tie    :: Corpus -> (String,ip) -> (String,ip) -> Bool}
```

Funkcija `wford` definira potpuni uređaj nad oblicima riječi iz korpusa, čime je zapravo određen redoslijed pribavljanja oblika iz korpusa (u formalnom modelu to je ostvareno funkcijom *wfpref*). Funkcije `accept` i `tie` odgovaraju funkcijama (5.1) i (5.3). Tip `Score ip` odgovara mjeri ocjene i definiran je kao:

```
type Score ip = Corpus -> (String, ip) -> Double
```

A.2.11 Modul `Molex.Acquisition.Heuristics`

U podmodulu `Molex.Acquisition.Heuristics` definirane su mjere ocjene LU-parova opisane u odjeljcima 5.2 i 5.3:

```
score0   :: (IPR ip x) => Score ip
score1   :: (IPR ip x) => Score ip
score2   :: (IPR ip x) => Score ip
score3   :: (IPR ip x) => Score ip
score02  :: (IPR ip x) => Score ip
score12  :: (IPR ip x) => Score ip
scoreP1  :: (IPR ip x) => MsdProb ip x -> Score ip
scoreP2  :: (IPR ip x) => MsdProb ip x -> Score ip
scoreP3  :: (IPR ip x) => MsdProb ip x -> Score ip
score1P2 :: (IPR ip x) => MsdProb ip x -> Score ip
score1P3 :: (IPR ip x) => MsdProb ip x -> Score ip
```

Vjerojatnosne mjere kao dodatni argument uzimaju vrijednost tipa `MsdProb ip x`, odnosno funkciju koja izračunava vjerojatnost morfosintaktičkog opisa u ovisnosti o flektivnome uzorku.

A.2.12 Modul `Molex.Derivation`

Modul `Molex.Derivation` definira različite funkcije za konstrukciju flektivno-derivacijskog leksikona. Osnovna funkcija modula jest:

```
dMerge :: (IPR ip x, Ord ip, DP dp ip x [ip]) => [dp] -> Molex ip -> Molex ip
```

Funkcija ostvaruje grupiranje i stapanje unosaka flektivnog leksikona temeljem liste derivacijskih uzoraka, sukladno (6.4).

Dodatak B

Izvedba modela HOFM za hrvatski jezik

Ovdje je izložena cjelovita programska izvedba modela HOFM za hrvatski jezik. U nastavku su najprije izložene jezično specifične preoblike definirane na način opisan u odjeljku 4.1.1. U dijelu B.2 izloženi su imenički, glagolski i pridjevski uzorci, prema opisu u odjeljku 4.2. U dijelu B.3 izloženi su derivacijski uzorci modela za sufiksalnu tvorbu imenica, glagola i pridjeva, na način opisan u odjeljku 4.3. Tekst koji je u programskom kodu prefiksiran dvjema crticama (`--`) jest komentar.

B.1 Jezično specifične preoblike

```
trueCons = [
  "p","t","k","b","d","g","c","č","ć","dž","đ",
  "f","s","š","h","z","ž"]
sonants = [
  "r","v","m","n","nj","l","lj"]
cons =
  trueCons ++ sonants ++ ["j"]
pals = [
  "lj","nj","ć","đ","č","dž","š","ž","j"]
nonpals = [
  "v","r","l","m","n","p","b","f","t","d","s",
  "z","c","k","g","h"]
velars = [
  "k","g","h"]

caps = map (:[]) $ ['A'..'Z'] ++ "ĐŽŠČĆ"

-- consonant groups
cgr = [c1++c2 |
  c1 <- cons, c2 <- cons, c1++c2 `notElem` ["nj","lj","dž"]]

-----------------------------------------------------------------
-- MORFOLOŠKI UVJETOVANE ALTERNACIJE
-----------------------------------------------------------------

-- sibilarizacija
sbl = asfx [("k","c"), ("h","s"), ("g","z")]

-- palatalizacija
plt = asfx [("k","č"),("g","ž"),("h","š"),("c", "č"),("z","ž")]

-- jotacija
jot = asfx [
  ("k","č"),("g","ž"),("h","š"),("c","č"),("z","ž"),("s","š"),
  ("t","ć"),("d","đ"),("l","lj"),("n","nj"),("p","plj"),
  ("b","blj"),("m","mlj"),("v","vlj"),("f","flj")]

-- zamjena suglasničkih skupoca
acg = asfx [
  ("ht","šć"),("sk","šč"),("sk","šć"),("sl","šlj"),("sn","šnj"),
  ("st","šć"),("st","št"),("zd","žđ"),("zn","žnj")]

-- proširenje osnove samoglasnikom a/e
ex v cs = asfx [(c,v++c) | c <- cs]
exa = ex "a" ["nj","lj","dž"] .||. ex "a" cons
exe = ex "e" ["nj","lj","dž"] .||. ex "e" cons

-- alternacija refleksa jata
jat1 = rifx "ije" "je"
jat2 = rifx "ije" "e"
jat3 = rifx "ije" "i"

-----------------------------------------------------------------
-- FONOLOŠKI UVJETOVANE ALTERNACIJE
-----------------------------------------------------------------

-- ozvučivanje
ozv = [
  ("p","b",["dž"]),
  ("t","d",["b"]),
  ("s","z",["b","d","g"]),
  ("s","ž",["b"]),
  ("č","dž",["b"]),
  ("k","g",["dž"]),
  ("s","ž",["dž"]),
  ("z","ž",["dž"])]
```

```
67
68  -- obezvučivanje
69  obzv = [
70    ("b","p",["c","č","ć","f","h","k","s","š","t"]),
71    ("d","t",["f","h","k","p"]),
72    ("g","k",["c","č"]),
73    ("đ","ć",["k"]),
74    ("z","s",["c","f","h","k","p","t"]),
75    ("ž","š",["c","k"])]
76
77  -- jednačenje suglasnika po mjestu tvorbe
78  jmt = [
79    ("s","š",cs),
80    ("z","š",cs),
81    ("z","ž",cs),
82    ("h","š",cs),
83    ("n","m",["p","b"])]
84    where cs = ["č","ć","dž","đ","lj","nj"]
85
86  -- ispadanje suglasnika
87  isg = [
88    ("t","",cs),
89    ("d","",cs)] ++ concat
90    [[(a++"t",a,[b]),(a++"d",a,[b])] |
91       a <- ["s","z","š","ž"], b <- cons \\ ["r","v"]]
92    where cs = ["c","č","ština"]
93
94  pca = [(x++c,y++c) | (x,y,cs) <- alts, c <- cs]
95    where alts = obzv ++ ozv ++ jmt ++ isg
96
97  pca1 = asfx pca
98  pca2 = aifx pca
99
100 -- preoblika sufikacije s ugrađenom fonološkom alternacijom:
101 -- (Napomena: ova se preoblika u dijelu B.3 koristi pod nazivom 'sfx')
102
103 sfx' s = try pca2 & sfx s & try pca1
104
105
106
107
```

B.2 Definicije flektivnih uzoraka

B.2.1 Imenički uzorci

```
1   -- vrsta a, ništični nastavak:
2   -- uzorak 1 (252)
3   n01 = iPattern "N01" "N-m"   -- izvor, političar
4     (ends (nonpals ++ ["j","š","č","ž"]))
5     [nul        # ["sn","sa"],
6      sfx "a"    # ["sg","pg"],
7      sfx "u"    # ["sd","sl"],
8      sfx "e"    # ["sv","pa"],
9      sfx "om"          # ["si"],
10     sfx "i"    # ["pn","pv"],
11     sfx "ima"  # ["pd","pl","pi"]]
12
13  -- uzorak 2 (253)
14  n02 = iPattern "N02" "N-m"   -- nokat
15    (ends cgr `land` nends ["l"])
16    [exa            # ["sn","sa"],
```

```
    sfx "a" & t    # ["sg"],
    sfx "u" & t    # ["sd","sl"],
    sfx "e" & t    # ["sv","pa"],
    sfx "om" & t   # ["si"],
    sfx "i" & t    # ["pn","pv"],
    sfx "ima" & t  # ["pd","pl","pi"],
    sfx "a" & exa  # ["pg"]]
  where t = try pca1

-- uzorak 3 (257)
n03 = iPattern "N03" "N-m"   -- otocanin, bugarin
  always
  [sfx "in"    # ["sn"],
   sfx "ina"   # ["sg","sa"],
   sfx "inu"   # ["sd"],
   sfx "ine"   # ["sv"],
   sfx "inom"  # ["si"],
   sfx "i"     # ["pn","pv"],
   sfx "a"     # ["pg"],
   sfx "ima"   # ["pd","pl","pi"],
   sfx "e"     # ["pa"]]

-- uzorak 4 (260)
n04 = iPattern "N04" "N-m"   -- vojnik, bubreg, trbuh, kirurg
  (ends velars `land` (neg(ends cgr)))
  [nul              # ["sn"],
   sfx "a"          # ["sg","sa","pg"],
   sfx "u"          # ["sl"],
   sfx "e" & plt    # ["sv"],
   sfx "om"         # ["si"],
   sfx "i" & sbl    # ["pn","pv"],
   sfx "ima" & sbl  # ["pd","pl","pi"],
   sfx "e"          # ["pa"]]

-- uzorak 5 (261)
n05 = iPattern "N05" "N-m"   -- čvorak
  (ends velars `lor` ends cgr)
  [exa                # ["sn"],
   sfx "a" & t        # ["sg","sa"],
   sfx "u" & t        # ["sd","sl"],
   sfx "e" & plt & t  # ["sv"],
   sfx "om" & t       # ["si"],
   sfx "i" & sbl & t  # ["pn"],
   sfx "a" & exa      # ["pg"],
   sfx "ima" & sbl & t  # ["pd","pl","pi"],
   sfx "e" & t        # ["pa"]]
  where t = try pca1

-- uzorak 5B
n06 = iPattern "N06" "N-m"   -- napredak, odlazak, kralježak, zadatak
  (ends nonpals)
  [exa                       # ["sn","sa"],
   sfx "a" & t1              # ["sg"],
   sfx "u" & t1              # ["sd","sl","sv"],
   sfx "om" & t1             # ["si"],
   sfx "e" & t2 & plt & t1   # ["sv"],
   sfx "i" & t2 & sbl & t1   # ["pn","pv"],
   sfx "a" & exa             # ["pg"],
   sfx "e" & t1              # ["pa"],
   sfx "ima" & t2 & sbl & t1  # ["pd","pl","pi"]]
  where t1 = try pca1
        t2 = opt pca1

-- uzorak 6 (262)
n07 = iPattern "N07" "N-m"   -- panj, marš, grč
  (ends pals `lor` ends ["št","žđ"])
  [nul           # ["sn","sa"],
   sfx "a"       # ["sg"],
   sfx "u"       # ["sd","sv","sl"],
   sfx "em"      # ["si"],
   i & sfx "i"   # ["pn","pv"],
   i & sfx "a"   # ["pg"],
   i & sfx "ima"  # ["pd","pl","pi"],
   i & sfx "e"   # ["pa"]]
```

```
    where i = sfx "ev"

-- uzorak 6B (262)
n08 = iPattern "N08" "N-m"   -- ražanj, češalj
  ((ends pals `lor` ends ["št","žd"]) `land` ends cgr)
  [exa              # ["sn","sv"],
   sfx "a" & t      # ["sg"],
   sfx "u" & t      # ["sd","sv","sl"],
   sfx "em" & t     # ["si"],
   sfx "i" & t      # ["pn","pv"],
   sfx "evi" & t    # ["pn","pv"],
   sfx "a" & exa    # ["pg"],
   sfx "ima" & t    # ["pd","pl","pi"],
   sfx "evima" & t  # ["pd","pl","pi"],
   sfx "e" & t      # ["pa"],
   sfx "eve" & t    # ["pa"]]
  where t = try pca1

-- uzorak 7 (264)
n09 = iPattern "N09" "N-m"   -- stric
  (ends ["c"] `land` neg(ends cgr))
  [nul                 # ["sn"],
   sfx "a"             # ["sg","sa"],
   sfx "u"             # ["sd","sl"],
   sfx "e" & plt       # ["sv"],
   sfx "em"            # ["si"],
   sfx "i" & i & plt   # ["pn","pv"],
   sfx "a" & i & plt   # ["pg"],
   sfx "ima" & i & plt  # ["pd","pl","pi"],
   sfx "e" & i & plt   # ["pa"]]
   where i = sfx "ev"

-- uzorak 7B (264)
n10 = iPattern "N10" "N-m"   -- klinac, obrazac
  (ends ["c"])
  [exa               # ["sn","sa"],
   sfx "a" & t       # ["sg"],
   sfx "u" & t       # ["sd","sl"],
   sfx "e" & plt & t  # ["sv"],
   sfx "em" & t      # ["si"],
   sfx "a" & exa     # ["pg"],
   sfx "ima" & t     # ["pd","pl","pi"],
   sfx "i" & t       # ["pn","pv"],
   sfx "e" & t       # ["pa"]]
  where t = try pca1

-- uzorak 8 (268)
n11 = iPattern "N11" "N-m"   -- posjetilac (A/197)
  (ends ["c"])
  [rsfx "oc" "lac"   # ["sn"],
   sfx "a"           # ["sg","sa"],
   sfx "u"           # ["sd","sl"],
   sfx "e" & plt     # ["sv"],
   sfx "em"          # ["si"],
   sfx "i"           # ["pn","pv"],
   rsfx "oc" "laca"  # ["pg"],
   sfx "ima"         # ["pd","pl","pi"],
   sfx "e"           # ["pa"]]

-- uzorak 9 (269)
n12 = iPattern "N12" "N-m"   -- pepeo (A/198)
  (ends ["el"])
  [rsfx "l" "o"  # ["sn","sa"],
   sfx "a"       # ["sg","pg"],
   sfx "u"       # ["sd","sl"],
   sfx "e"       # ["sv","pa"],
   sfx "om"      # ["si"],
   sfx "i"       # ["pn","pv"],
   sfx "ima"     # ["pd","pl","pi"]]

-- uzorak 9B (269)
n13 = iPattern "N13" "N-m"   -- ugao (A/198b)
  (ends ["l"] `land` ends cgr)
  [rsfx "l" "ao"  # ["sn","sa"],
```

```
    sfx "a"        # ["sg"],
    sfx "u"        # ["sd","sl"],
    sfx "e"        # ["sv"],
    sfx "om"       # ["si"],
    sfx "ovi"      # ["pn","pv"],
    sfx "ove"      # ["pa"],
    sfx "ovima"    # ["pd","pl","pi"],
    sfx "ova"      # ["pg"]]

-- uzorak 10 (271)
n14 = iPattern "N14" "Ncm"   -- čovjek
  (ends ["čovjek"])
  [nul                      # ["sn"],
   sfx "a"                  # ["sg","sa"],
   sfx "u"                  # ["sd","sl"],
   sfx "e"                  # ["sv"],
   sfx "om"                 # ["si"],
   rsfx "čovjek" "ljudi"    # ["pn","pg","pv"],
   rsfx "čovjek" "ljude"    # ["pa"],
   rsfx "čovjek" "ljudima"  # ["pd","pl","pi"]]

-- uzorak 11 (272)
n15 = iPattern "N15" "N-m"   -- intervju
  (ends ["u", "e", "o"])
  [nul        # ["sn","sa"],
   sfx "a"    # ["sg","pg"],
   sfx "u"    # ["sd","sv","sl"],
   sfx "om"   # ["si"],
   sfx "i"    # ["pn","pv"],
   sfx "ima"  # ["pd","pl","pi"],
   sfx "e"    # ["pa"]]

-- uzorak 12 (273)
n16 = iPattern "N16" "N-m"   -- žiri, Emmy (A/201b)
  (ends ["i","y"])
  [nul         # ["sn","sa"],
   sfx "ja"    # ["sg","pg"],
   sfx "ju"    # ["sd","sv","sl"],
   sfx "jem"   # ["si"],
   sfx "ji"    # ["pn","pv"],
   sfx "jima"  # ["pd","pl","pi"],
   sfx "je"    # ["pa"]]

-- nastavak -o ili -e:

-- uzorak 13 (vlastita imena)
n17 = iPattern "N17" "N-m"   -- Danilo
  (ends nonpals)
  [sfx "o"    # ["sn","sv"],
   sfx "a"    # ["sg","sa","pg"],
   sfx "u"    # ["sd","sl"],
   sfx "om"   # ["si"],
   sfx "i"    # ["pn","pv"],
   sfx "ima"  # ["pd","pl","pi"],
   sfx "e"    # ["pa"]]

-- uzorak 13B (274)
n18 = iPattern "N18" "N-m"   -- Hrvoje
  (ends pals)
  [sfx "e"    # ["sn","sv"],
   sfx "a"    # ["sg","sa","pg"],
   sfx "u"    # ["sd","sl"],
   sfx "em"   # ["si"],
   sfx "i"    # ["pd","pl","pi"],
   sfx "ima"  # ["pa"]]

-- uzorak 14 (276)
n19 = iPattern "N19" "N-m"   -- rašćupanko (A/203b)
  (ends ["k"])
  [sfx "o"          # ["sn","sv"],
   sfx "a"          # ["sg","sa","pg"],
   sfx "u"          # ["sd","sl"],
   sfx "om"         # ["si"],
   sfx "e"          # ["pa"],
```

```
    sfx "i" & sbl    # ["pn","pv"],
    sfx "ima" & sbl  # ["pd","pl","pi"]]

n30 = iPattern "N30" "N-m"   -- zeko (A/248)
  always
  [sfx "o"    # ["sn","sv"],
   sfx "e"    # ["sg","pn","pa","pv"],
   sfx "i"    # ["sd","sl"],
   sfx "u"    # ["sa"],
   sfx "om"   # ["si"],
   sfx "a"    # ["pg"],
   sfx "ama"  # ["pd","pl","pi"]]

n37 = iPattern "N37" "N-m"   -- prijatelj
  (ends cons)
  [nul        # ["sn","sv"],
   sfx "a"    # ["sg","sa","pg"],
   sfx "u"    # ["sd"],
   sfx "em"   # ["si"],
   sfx "i"    # ["pn","pv"],
   sfx "ima"  # ["pd","pl","pi"],
   sfx "e"    # ["pa"]]

n38 = iPattern "N38" "N-m"   -- artikl
  (ends cgr `land` neg(ends ["n"] `lor` ends ["nc"]))
  [nul            # ["sn","sa","sv"],
   sfx "a"        # ["sg"],
   sfx "u"        # ["sd"],
   sfx "e"        # ["pa"],
   sfx "om"       # ["si"],
   sfx "i"        # ["pn","pv"],
   sfx "ima"      # ["pd","pl","pi"],
   sfx "a" & exa  # ["pg"]]

n41 = iPattern "N41" "N-m"   -- bod, akt, grb, cvijet, show, Byrne
  (ends (cons ++ ["y","x","w"]))
  [nul              # ["sn","sa"],
   sfx "a"          # ["sg"],
   sfx "u"          # ["sd","sl"],
   sfx "e"          # ["sv"],
   sfx "om"         # ["si"],
   sfx "ovi" & t    # ["pn","pv"],
   sfx "ova" & t    # ["pg"],
   sfx "ove" & t    # ["pa"],
   sfx "ovima" & t  # ["pd","pl","pi"]]
  where t = try jat1

n43 = iPattern "N43" "N-m"   -- dio
  always
  [rsfx "l" "o" & rifx "ije" "i" # ["sn","sa"],
   sfx "a"      # ["sg"],
   sfx "u"      # ["sd","sl"],
   sfx "om"     # ["si"],
   sfx "ovi"    # ["pn","pv"],
   sfx "ova"    # ["pg"],
   sfx "ove"    # ["pa"],
   sfx "ovima"  # ["pd","pl","pi"]]

n44 = iPattern "N44" "N-m"   -- kadar, zajam, pojam
  (ends cgr)
  [exa              # ["sn","sa"],
   sfx "a" & t      # ["sg"],
   sfx "u" & t      # ["sd","sl"],
   sfx "e" & t      # ["sv","pa"],
   sfx "om" & t     # ["si"],
   sfx "ovi" & t    # ["pn","pv"],
   sfx "ovima" & t  # ["pd","pl","pi"],
   sfx "ova" & t    # ["pg"],
   sfx "a" & exa    # ["pg"]]
  where t = try pca1

n45 = iPattern "N45" "N-m"   -- kolega, mladoženja
  (ends cons)
  [sfx "a"    # ["sn","pg"],
```

```
    sfx "e"    # ["sg","pn","pv","pa"],
    sfx "i"    # ["sd","sl"],
    sfx "u"    # ["sa"],
    sfx "o"    # ["sv"],
    sfx "om"   # ["si"],
    sfx "ama"  # ["pd","pl","pi"]]

n46 = iPattern "N46" "N-m"   -- mozak
  (ends ["zg"])
  [rsfx "g" "ak"  # ["sn","sa"],
   sfx "a"        # ["sg"],
   sfx "u"        # ["sd","sl"],
   sfx "om"       # ["si"],
   sfx "ovi"      # ["pn","pv"],
   sfx "ova"      # ["pg"],
   sfx "ove"      # ["pa"],
   sfx "ovima"    # ["pd","pl","pi"]]

n48 = iPattern "N48" "N-m"   -- žabac, otac, sudac
  (ends ["c"])
  [exa               # ["sn"],
   sfx "a" & t       # ["sg","sa"],
   sfx "u" & t       # ["sd","sl"],
   sfx "e" & plt & t # ["sv"],
   sfx "em" & t      # ["si"],
   sfx "a" & exa     # ["pg"],
   sfx "ima" & t     # ["pd","pl","pi"],
   sfx "i" & t       # ["pn","pv"],
   sfx "e" & t       # ["pa"]]
  where t = try pca1

-- srednji rod:

n20 = iPattern "N20" "N-n"   -- koljeno (A/218)
  (ends nonpals ‘land‘ neg(ends ["c","št"]))
  [sfx "o"    # ["sn","sa","sv"],
   sfx "a"    # ["sg","pn","pg","pa","pv"],
   sfx "u"    # ["sd","sl"],
   sfx "om"   # ["si"],
   sfx "ima"  # ["pd","pl","pi"]]

n21 = iPattern "N21" "N-n"   -- jedro (A/219)
  (ends cgr ‘land‘ neg(ends ["st","zd"]))
  [sfx "o"        # ["sn","sa","sv"],
   sfx "a"        # ["sg","pn","pa","pv"],
   sfx "u"        # ["sd","sl"],
   sfx "om"       # ["si"],
   sfx "a" & exa  # ["pg"],
   sfx "ima"      # ["pd","pl","pi"]]

n22 = iPattern "N22" "N-n"   -- polje (A/225)
  (ends ["j","lj","nj","ć","đ","c","št","šć","žđ","mor","tl"])
  [sfx "e"    # ["sn","sa","sv"],
   sfx "a"    # ["sg","pn","pg","pa","pv"],
   sfx "u"    # ["sd","sl"],
   sfx "em"   # ["si"],
   sfx "ima"  # ["pd","pl","pi"]]

n23 = iPattern "N23" "N-n"  -- sunce (A/226)
  (ends cgr ‘land‘ neg(ends ["št","šć","žđ"] ‘lor‘ ends ["j"]))
  [sfx "e"        # ["sn","sa","sv"],
   sfx "a"        # ["sg","pn","pa","pv"],
   sfx "u"        # ["sd","sl"],
   sfx "em"       # ["si"],
   sfx "a" & exa  # ["pg"],
   sfx "ima"      # ["pd","pl","pi"]]

n24 = iPattern "N24" "N-n"  -- uže (A/228)
  (ends ["pet","bet","met","tet","det","let","ret","net","set",
         "zet","šet","žet","čet","jajet"])
  [dsfx "t"   # ["sn","sa","sv"],
   sfx "a"    # ["sg","pn","pg","pa","pv"],
   sfx "u"    # ["sd","sl"],
   sfx "om"   # ["si"],
```

```
    sfx "ima"  # ["pd","pl","pi"]]

n25 = iPattern "N25" "N-n"  -- zvonce (A/230)
  (ends ["c","anc","ašc","ešc"])
  [sfx "e"        # ["sn","sa","sv"],
   sfx "a"        # ["sg","pn","pa","pv"],
   sfx "u"        # ["sd","sl"],
   sfx "em"       # ["si"],
   sfx "eta"      # ["sg"],
   sfx "etu"      # ["sd","sl"],
   sfx "etom"     # ["si"],
   sfx "ad"       # ["pn"],
   sfx "a" & exa  # ["pg"],
   sfx "ima"      # ["pd","pl","pi"]]

n26 = iPattern "N26" "N-n"  -- rame (A/231)
  (ends ["breme","ime","pleme","rame","sjeme","tjeme","vime","eme"])
  [nul         # ["sn","sa","sv"],
   sfx "na"    # ["sg","pn","pg","pa","pv"],
   sfx "nu"    # ["sd","sl"],
   sfx "nom"   # ["si"],
   sfx "nima"  # ["pd","pl","pi"]]

n27 = iPattern "N27" "N-n"  -- podne (A/232)
  (ends ["podnev"])
  [dsfx "v"   # ["sn","sa","sv"],
   sfx "a"    # ["sg","pn","pg","pa","pv"],
   sfx "u"    # ["sd","sl"],
   sfx "om"   # ["si"],
   sfx "ima"  # ["pd","pl","pi"]]

n40 = iPattern "N40" "N-n"  -- vrijeme
  (ends nonpals)
  [t0         # ["sn","sa","sv"],
   sfx "a"    # ["sg","pn","pg","pa","pv"],
   sfx "u"    # ["sd","sl"],
   sfx "om"   # ["si"],
    sfx "ima"  # ["pd","pl","pi"]]
  where t0 = inv (jat2 & sfx "n")

-- ženski rod:

n28 = iPattern "N28" "N-f"  -- žaba (A/241)
  (ends cons)
  [sfx "a"    # ["sn","pg"],
   sfx "e"    # ["sg","pn","pa","pv"],
   sfx "i"    # ["sd","sl"],
   sfx "u"    # ["sa"],
   sfx "o"    # ["sv"],
   sfx "om"   # ["si"],
   sfx "ama"  # ["pd","pl","pi"]]

n47 = iPattern "N47" "N-f"  -- crkva
  (ends cons)
  [sfx "a"        # ["sn","pg"],
   sfx "e"        # ["sg","pn","pa","pv"],
   sfx "i"        # ["sd","sl"],
   sfx "u"        # ["sa"],
   sfx "o"        # ["sv"],
   sfx "om"       # ["si"],
   sfx "ama"      # ["pd","pl","pi"],
   (sfx "a" & exa) .|. sfx "i" .|. sfx "a" # ["pg"]]

n29 = iPattern "N29" "N-f"  -- slika (A/242)
  (ends velars)
  [sfx "a"        # ["sn","pg"],
   sfx "e"        # ["sg","pn","pa","pv"],
   sfx "i" & sbl  # ["sd","sl"],
   sfx "i"        # ["sd","sl"],
   sfx "u"        # ["sa"],
   sfx "o"        # ["sv"],
   sfx "om"       # ["si"],
   sfx "ama"      # ["pd","pl","pi"]]
```

```
n31 = iPattern "N31" "N-f"  -- mati (A/248)
  (ends ["mater"])
  [rsfx "er" "i"  # ["sn","sv"],
   sfx "e"        # ["sg","pn","pa","pv"],
   sfx "i"        # ["sd","sl"],
   nul            # ["sa"],
   sfx "om"       # ["si"],
   sfx "a"        # ["pg"],
   sfx "ama"      # ["pd","pl","pi"]]

n32 = iPattern "N32" "N-f"  -- riječ (A/251)
  (ends ["č","ž","š","s","z","r","n"])
  [nul        # ["sn","sa"],
   sfx "i"    # ["sg","sd","sv","sl","si","pn","pg","pa","pv"],
   sfx "ima"  # ["pd","pl","pi"],
   sfx "ju"   # ["si"]]

n33 = iPattern "N33" "N-f"  -- kap, krv, svijest (A/252)
  (ends ["p","b","v","m","t","d", "n"])
  [nul            # ["sn","sa"],
   sfx "i"        # ["sg","sd","sv","sl","si","pn","pg","pa","pv"],
   sfx "ima"      # ["pd","pl","pi"],
   sfx "u" & try pca1 & jot  # ["si"]]

n34 = iPattern "N34" "N-f"  -- misao (A/252b)
  (ends ["l"])
  [rsfx "l" "ao"  # ["sn","sa"],
   sfx "i"        # ["sg","sd","sv","sl","si","pn","pg","pa","pv"],
   sfx "ima"      # ["pd","pl","pi"],
   sfx "u" & jot  # ["si"]]

n35 = iPattern "N35" "N-f"  -- noć, obitelj (A/253)
  (ends ["ć","đ","lj"])
  [nul        # ["sn","sa"],
   sfx "i"    # ["sg","sd","sv","sl","si","pn","pg","pa","pv"],
   sfx "ima"  # ["pd","pl","pi"],
   sfx "u"    # ["si"]]

n36 = iPattern "N36" "N-f"  -- kći (A/253)
  (ends ["kćer"])
  [rsfx "er" "i" # ["sn"],
   sfx "i"       # ["sg","sd","sv","sl","si","pn","pg","pa","pv"],
   nul           # ["sa"],
   sfx "ju"      # ["si"],
   sfx "ima"     # ["pd","pl","pi"]]

n39 = iPattern "N39" "N-f"  -- banka
  (ends velars ‘land‘ ends cgr)
  [sfx "a"        # ["sn","pg"],
   sfx "e"        # ["sg","pn","pa","pv"],
   sfx "i" & sbl  # ["sd","sl"],
   sfx "i"        # ["sd","sl"],
   sfx "u"        # ["sa"],
   sfx "o"        # ["sv"],
   sfx "om"       # ["si"],
   sfx "ama"      # ["pd","pl","pi"],
   sfx "a" & exa  # ["pg"]]

n49 = iPattern "N49" "N-f"  -- novine, hlače (pl. tantum)
  (ends cons)
  [sfx "e"    # ["pn","pa","pv"],
   sfx "a"    # ["pg"],
   sfx "ama"  # ["pd","pl","pi"]]
```

B.2.2 Glagolski uzorci

```
v01 = iPattern "V01" "Vm"  -- plesti (v1r1a)
  (ends ["t"])
  [rsfx "t" "sti"  # ["n"],
   rsfx "t" "st"   # ["n"],
   sfx "em"        # ["ip1s"],
   sfx "eš"        # ["ip2s"],
   sfx "e"         # ["ip3s"],
   sfx "emo"       # ["ip1p"],
   sfx "ete"       # ["ip2p"],
   sfx "u"         # ["ip3p"],
   sfx "i"         # ["m-2s"],
   sfx "imo"       # ["m-1p"],
   sfx "ite"       # ["m-2p"],
   sfx "ući"       # ["gp"],
   sfx "avši"      # ["gs"],
   rsfx "t" "o"    # ["pp-s"],
   rsfx "t" "la"   # ["pp-s"],
   rsfx "t" "lo"   # ["pp-s"],
   rsfx "t" "li"   # ["pp-p"],
   rsfx "t" "le"   # ["pp-p"]]

v02 = iPattern "V02" "Vm"  -- krasti (v1r1b)
  (ends ["d"])
  [rsfx "d" "sti"  # ["n"],
   rsfx "d" "st"   # ["n"],
   sfx "em"        # ["ip1s"],
   sfx "eš"        # ["ip2s"],
   sfx "e"         # ["ip3s"],
   sfx "emo"       # ["ip1p"],
   sfx "ete"       # ["ip2p"],
   sfx "u"         # ["ip3p"],
   sfx "i"         # ["m-2s"],
   sfx "imo"       # ["m-1p"],
   sfx "ite"       # ["m-2p"],
   sfx "ući"       # ["gp"],
   sfx "avši"      # ["gs"],
   rsfx "d" "o"    # ["pp-s"],
   rsfx "d" "la"   # ["pp-s"],
   rsfx "d" "lo"   # ["pp-s"],
   rsfx "d" "li"   # ["pp-p"],
   rsfx "d" "le"   # ["pp-p"]]

v03 = iPattern "V03" "Vm"  -- tresti (v1r2a)
  (ends ["s"] `land` nends["nes"])
  [sfx "ti"        # ["n"],
   sfx "t"         # ["n"],
   sfx "em"        # ["ip1s"],
   sfx "eš"        # ["ip2s"],
   sfx "e"         # ["ip3s"],
   sfx "emo"       # ["ip1p"],
   sfx "ete"       # ["ip2p"],
   sfx "u"         # ["ip3p"],
   sfx "i"         # ["m-2s"],
   sfx "imo"       # ["m-1p"],
   sfx "ite"       # ["m-2p"],
   sfx "ući"       # ["gp"],
   sfx "avši"      # ["gs"],
   sfx "ao"        # ["pp-s"],
   sfx "la"        # ["pp-s"],
   sfx "lo"        # ["pp-s"],
   sfx "li"        # ["pp-p"],
   sfx "le"        # ["pp-p"]]

v04 = iPattern "V04" "Vm"  -- don(es/ije)ti (v1r2b)
  (ends ["nes"])
  [sfx "ti"        # ["n"],
   rsfx "es" "ijeti" # ["n"],
   sfx "t"         # ["n"],
   rsfx "es" "ijet" # ["n"],
   sfx "em"        # ["ip1s"],
   sfx "eš"        # ["ip2s"],
   sfx "e"         # ["ip3s"],
   sfx "emo"       # ["ip1p"],
   sfx "ete"       # ["ip2p"],
   sfx "u"         # ["ip3p"],
   sfx "i"         # ["m-2s"],
   sfx "imo"       # ["m-1p"],
   sfx "ite"       # ["m-2p"],
   sfx "ući"       # ["gp"],
   sfx "avši"      # ["gs"],
   sfx "ao"        # ["pp-s"],
   sfx "la"        # ["pp-s"],
   rsfx "es" "io"  # ["pp-s"],
   rsfx "es" "ijela" # ["pp-s"],
   rsfx "es" "ijelo" # ["pp-s"],
   rsfx "es" "ijele" # ["pp-p"],
   rsfx "es" "ijeli" # ["pp-p"]]

v05 = iPattern "V05" "Vm"  -- gristi (v1r2c)
  (ends ["z"])
  [rsfx "z" "sti"  # ["n"],
   rsfx "z" "st"   # ["n"],
   sfx "em"        # ["ip1s"],
   sfx "eš"        # ["ip2s"],
   sfx "e"         # ["ip3s"],
   sfx "emo"       # ["ip1p"],
   sfx "ete"       # ["ip2p"],
   sfx "u"         # ["ip3p"],
   sfx "i"         # ["m-2s"],
   sfx "imo"       # ["m-1p"],
   sfx "ite"       # ["m-2p"],
   sfx "ući"       # ["gp"],
```

```
    sfx "avši"      # ["gs"],
    sfx "ao"        # ["pp-s"],
    sfx "la"        # ["pp-s"],
    sfx "lo"        # ["pp-s"],
    sfx "li"        # ["pp-p"],
    sfx "le"        # ["pp-p"]]

v06 = iPattern "V06" "Vm"  -- crpsti (v1r3a)
  (ends ["p"])
  [sfx "sti"       # ["n"],
   sfx "st"        # ["n"],
   sfx "em"        # ["ip1s"],
   sfx "eš"        # ["ip2s"],
   sfx "e"         # ["ip3s"],
   sfx "emo"       # ["ip1p"],
   sfx "ete"       # ["ip2p"],
   sfx "u"         # ["ip3p"],
   sfx "i"         # ["m-2s"],
   sfx "imo"       # ["m-1p"],
   sfx "ite"       # ["m-2p"],
   sfx "ući"       # ["gp"],
   sfx "avši"      # ["gs"],
   sfx "ao"        # ["pp-s"],
   sfx "la"        # ["pp-s"],
   sfx "lo"        # ["pp-s"],
   sfx "li"        # ["pp-p"],
   sfx "le"        # ["pp-p"]]

v07 = iPattern "V07" "Vm"  -- grepsti (v1r3b)
  (ends ["b"])
  [rsfx "b" "psti" # ["n"],
   rsfx "b" "pst"  # ["n"],
   sfx "em"        # ["ip1s"],
   sfx "eš"        # ["ip2s"],
   sfx "e"         # ["ip3s"],
   sfx "emo"       # ["ip1p"],
   sfx "ete"       # ["ip2p"],
   sfx "u"         # ["ip3p"],
   sfx "i"         # ["m-2s"],
   sfx "imo"       # ["m-1p"],
   sfx "ite"       # ["m-2p"],
   sfx "ući"       # ["gp"],
   sfx "avši"      # ["gs"],
   sfx "ao"        # ["pp-s"],
   sfx "la"        # ["pp-s"],
   sfx "lo"        # ["pp-s"],
   sfx "li"        # ["pp-p"],
   sfx "le"        # ["pp-p"]]

v08 = iPattern "V08" "Vm"  -- vući (v1r4a)
  (ends ["k"])
  [rsfx "k" "ći"    # ["n"],
   rsfx "k" "ć"     # ["n"],
   sfx "em" & plt   # ["ip1s"],
   sfx "eš" & plt   # ["ip2s"],
   sfx "e" & plt    # ["ip3s"],
   sfx "emo" & plt  # ["ip1p"],
   sfx "ete" & plt  # ["ip2p"],
   sfx "u"          # ["ip3p"],
   sfx "i" & sbl    # ["m-2s"],
   sfx "imo" & sbl  # ["m-1p"],
   sfx "ite" & sbl  # ["m-2p"],
   sfx "ući"        # ["gp"],
   sfx "avši"       # ["gs"],
   sfx "ao"         # ["pp-s"],
   sfx "la"         # ["pp-s"],
   sfx "lo"         # ["pp-s"],
   sfx "li"         # ["pp-p"],
   sfx "le"         # ["pp-p"]]

v09 = iPattern "V09" "Vm"  -- strići (v1r4b)
  (ends ["g"])
  [rsfx "g" "ći"     # ["n"],
   rsfx "g" "ć"      # ["n"],
   sfx "em" & plt  # ["ip1s"],
   sfx "eš" & plt  # ["ip2s"],
   sfx "e" & plt   # ["ip3s"],
   sfx "emo" & plt # ["ip1p"],
   sfx "ete" & plt # ["ip2p"],
   sfx "u"           # ["ip3p"],
   sfx "i" & sbl   # ["m-2s"],
   sfx "imo" & sbl # ["m-1p"],
   sfx "ite" & sbl # ["m-2p"],
   sfx "ući"         # ["gp"],
   sfx "avši"        # ["gs"],
   sfx "ao"          # ["pp-s"],
   sfx "la"          # ["pp-s"],
   sfx "lo"          # ["pp-s"],
   sfx "li"          # ["pp-p"],
   sfx "le"          # ["pp-p"]]

v10 = iPattern "V10" "Vm"  -- vrći (v1r4c)
  (ends ["h"])
  [rsfx "h" "ći"     # ["n"],
   rsfx "h" "ć"      # ["n"],
   sfx "em" & plt  # ["ip1s"],
   sfx "eš" & plt  # ["ip2s"],
   sfx "e" & plt   # ["ip3s"],
   sfx "emo" & plt # ["ip1p"],
   sfx "ete" & plt # ["ip2p"],
   sfx "u"           # ["ip3p"],
   sfx "i" & sbl   # ["m-2s"],
   sfx "imo" & sbl # ["m-1p"],
   sfx "ite" & sbl # ["m-2p"],
   sfx "ući"         # ["gp"],
   sfx "avši"        # ["gs"],
   sfx "ao"          # ["pp-s"],
   sfx "la"          # ["pp-s"],
   sfx "lo"          # ["pp-s"],
   sfx "li"          # ["pp-p"],
   sfx "le"          # ["pp-p"]]
```

```
v11 = iPattern "V11" "Vm"  -- obuti (v1r7a)
  (ends ["i", "u"])
  [sfx "ti"        # ["n"],
   sfx "t"         # ["n"],
   sfx "jem"       # ["ip1s"],
   sfx "ješ"       # ["ip2s"],
   sfx "je"        # ["ip3s"],
   sfx "jemo"      # ["ip1p"],
   sfx "jete"      # ["ip2p"],
   sfx "ju"        # ["ip3p"],
   sfx "j"         # ["m-2s"],
   sfx "jmo"       # ["m-1p"],
   sfx "jte"       # ["m-2p"],
   sfx "jući"      # ["gp"],
   sfx "vši"       # ["gs"],
   sfx "o"         # ["pp-s"],
   sfx "la"        # ["pp-s"],
   sfx "lo"        # ["pp-s"],
   sfx "li"        # ["pp-p"],
   sfx "le"        # ["pp-p"]]

v12 = iPattern "V12" "Vm"  -- smjeti (v1r7b)
  (ends ["je"])
  [sfx "ti"        # ["n"],
   sfx "t"         # ["n"],
   sfx "em" & t    # ["ip1s"],
   sfx "eš" & t    # ["ip2s"],
   sfx "e" & t     # ["ip3s"],
   sfx "emo" & t   # ["ip1p"],
   sfx "ete" & t   # ["ip2p"],
   sfx "u" & t     # ["ip3p"],
   t               # ["m-2s"],
   sfx "mo" & t    # ["m-1p"],
   sfx "te" & t    # ["m-2p"],
   sfx "ući" & t   # ["gp"],
   rsfx "je" "ivši" # ["gs"],
   sfx "o"         # ["pp-s"],
   sfx "lo"        # ["pp-s"],
   sfx "la"        # ["pp-s"],
   sfx "li"        # ["pp-p"],
   sfx "le"        # ["pp-p"]]
  where t = rsfx "je" "ij"

v13 = iPattern "V13" "Vm"  -- tonuti (v2)
  (ends ["n"])
  [sfx "uti"       # ["n"],
   sfx "ut"        # ["n"],
   sfx "em"        # ["ip1s"],
   sfx "eš"        # ["ip2s"],
   sfx "e"         # ["ip3s"],
   sfx "emo"       # ["ip1p"],
   sfx "ete"       # ["ip2p"],
   sfx "u"         # ["ip3p"],
   sfx "i"         # ["m-2s"],
   sfx "imo"       # ["m-1p"],
   sfx "ite"       # ["m-2s"],
   sfx "ući"       # ["gp"],
   sfx "uvši"      # ["gs"],
   sfx "uo"        # ["pp-s"],
   sfx "ula"       # ["pp-s"],
   sfx "ulo"       # ["pp-s"],
   sfx "uli"       # ["pp-p"],
   sfx "ule"       # ["pp-p"]]

v14 = iPattern "V14" "Vm"  -- vidjeti (v3r1)
  (ends cons)
  [sfx "jeti"      # ["n"],
   sfx "jet"       # ["n"],
   sfx "im"        # ["ip1s"],
   sfx "iš"        # ["ip2s"],
   sfx "i"         # ["ip3s","m-2s"],
   sfx "imo"       # ["ip1p","m-1p"],
   sfx "ite"       # ["ip2p","m-2p"],
   sfx "o"         # ["pp-s"],
   sfx "lo"        # ["pp-s"],
   sfx "la"        # ["pp-s"],
   sfx "li"        # ["pp-p"],
   sfx "le"        # ["pp-p"]]
  where t = rsfx "je" "ij"

v15 = iPattern "V15" "Vm"  -- držati (v3r2)
  (ends ["č","ž","j", "št", "žd"])
  [sfx "ati"       # ["n"],
   sfx "at"        # ["n"],
   sfx "im"        # ["ip1s"],
   sfx "iš"        # ["ip2s"],
   sfx "i"         # ["ip3s","m-2s"],
   sfx "imo"       # ["ip1p","m-1p"],
   sfx "ite"       # ["ip2p","m-2p"],
   sfx "e"         # ["ip3p"],
   sfx "ći"        # ["gp"],
   sfx "avši"      # ["gs"],
   sfx "ao"        # ["pp-s"],
   sfx "ala"       # ["pp-s"],
   sfx "alo"       # ["pp-s"],
   sfx "ali"       # ["pp-p"],
   sfx "ale"       # ["pp-p"]]

v16 = iPattern "V16" "Vm"  -- voziti (v4)
  always
  [sfx "iti"       # ["n"],
   sfx "it"        # ["n"],
   sfx "im"        # ["ip1s"],
   sfx "iš"        # ["ip2s"],
   sfx "i"         # ["ip3s","m-2s"],
   sfx "imo"       # ["ip1p","m-1p"],
   sfx "ite"       # ["ip2p","m-2p"],
   sfx "e"         # ["ip3p"],
   sfx "eći"       # ["gp"],
   sfx "jevši"     # ["gs"],
   sfx "io"        # ["pp-s"],
   sfx "jela"      # ["pp-s"],
   sfx "jelo"      # ["pp-s"],
   sfx "jeli"      # ["pp-p"],
   sfx "jele"      # ["pp-p"]]
```

```
    sfx "ećI"       # ["gp"],
    sfx "ivši"      # ["gs"],
    sfx "io"        # ["pp-s"],
    sfx "ila"       # ["pp-s"],
    sfx "ilo"       # ["pp-s"],
    sfx "ili"       # ["pp-p"],
    sfx "ile"       # ["pp-p"]]

v17 = iPattern "V17" "Vm"  -- pitati (v5r1)
  always
  [sfx "ati"       # ["n"],
   sfx "at"        # ["n"],
   sfx "am"        # ["ip1s"],
   sfx "aš"        # ["ip2s"],
   sfx "a"         # ["ip3s"],
   sfx "amo"       # ["ip1p"],
   sfx "ate"       # ["ip2p"],
   sfx "aju"       # ["ip3p"],
   sfx "aj"        # ["m-2s"],
   sfx "ajmo"      # ["m-1p"],
   sfx "ajte"      # ["m-2p"],
   sfx "ajući"     # ["gp"],
   sfx "avši"      # ["gs"],
   sfx "ao"        # ["pp-s"],
   sfx "ala"       # ["pp-s"],
   sfx "alo"       # ["pp-s"],
   sfx "ali"       # ["pp-p"],
   sfx "ale"       # ["pp-p"]]

v18 = iPattern "V18" "Vm"  -- glodati, lagati,
                           -- dozivati (v5r2a)
  always
  [sfx "ati"         # ["n"],
   sfx "at"          # ["n"],
   sfx "em" & jot  # ["ip1s"],
   sfx "eš" & jot  # ["ip2s"],
   sfx "e" & jot   # ["ip3s"],
   sfx "emo" & jot # ["ip1p"],
   sfx "ete" & jot # ["ip2p"],
   sfx "u" & jot   # ["ip3p"],
   sfx "i" & jot   # ["m-2s"],
   sfx "imo" & jot # ["m-1p"],
   sfx "ite" & jot # ["m-2p"],
   sfx "ući" & jot # ["gp"],
   sfx "avši"        # ["gs"],
   sfx "ao"          # ["pp-s"],
   sfx "ala"         # ["pp-s"],
   sfx "alo"         # ["pp-s"],
   sfx "ali"         # ["pp-p"],
   sfx "ale"         # ["pp-p"]]

v19 = iPattern "V19" "Vm"  -- dihati, vezati (v5r2b)
  always
  [sfx "ati"         # ["n"],
   sfx "at"          # ["n"],
   sfx "em" & jot  # ["ip1s"],
   sfx "eš" & jot  # ["ip2s"],
   sfx "e" & jot   # ["ip3s"],
   sfx "emo" & jot # ["ip1p"],
   sfx "ete" & jot # ["ip2p"],
   sfx "u" & jot   # ["ip3p"],
   sfx "i" & jot   # ["m-2s"],
   sfx "imo" & jot # ["m-1p"],
   sfx "ite" & jot # ["m-2p"],
   sfx "ući" & jot # ["gp"],
   sfx "avši"        # ["gs"],
   sfx "ao"          # ["pp-s"],
   sfx "ala"         # ["pp-s"],
   sfx "alo"         # ["pp-s"],
   sfx "ali"         # ["pp-p"],
   sfx "ale"         # ["pp-p"]]

v32 = iPattern "V32" "Vm"  -- zanijekati
                           -- (v18, varijanta)
  always
  [sfx "ati"         # ["n"],
   sfx "at"          # ["n"],
   sfx "em" & jot  # ["ip1s"],
   sfx "eš" & jot  # ["ip2s"],
   sfx "e" & jot   # ["ip3s"],
   sfx "emo" & jot # ["ip1p"],
   sfx "ete" & jot # ["ip2p"],
   sfx "u" & jot   # ["ip3p"],
   sfx "i" & jot   # ["m-2s"],
   sfx "imo" & jot # ["m-1p"],
   sfx "ite" & jot # ["m-2p"],
   sfx "ući" & jot # ["gp"],
   sfx "avši"        # ["gs"],
   sfx "ao"          # ["pp-s"],
   sfx "ala"         # ["pp-s"],
   sfx "alo"         # ["pp-s"],
   sfx "ali"         # ["pp-p"],
   sfx "ale"         # ["pp-p"]]

v20 = iPattern "V20" "Vm"  -- derati, hrvati,
                           -- očešati (v5r3a)
  (ends ["r","v","š"])
  [sfx "ati"       # ["n"],
   sfx "at"        # ["n"],
   sfx "em"        # ["ip1s"],
   sfx "eš"        # ["ip2s"],
   sfx "e"         # ["ip3s"],
   sfx "emo"       # ["ip1p"],
   sfx "ete"       # ["ip2p"],
   sfx "u"         # ["ip3p"],
   sfx "i"         # ["m-2s"],
   sfx "ite"       # ["m-1p"],
   sfx "imo"       # ["m-1p"],
   sfx "ući"       # ["gp"],
   sfx "avši"      # ["gs"],
   sfx "ao"        # ["pp-s"],
```

```
    sfx "ala"       # ["pp-s"],
    sfx "alo"       # ["pp-s"],
    sfx "ali"       # ["pp-p"],
    sfx "ale"       # ["pp-p"]]

v21 = iPattern "V21" "Vm"  -- prati (v5r3b)
  (ends ["r"])
  [rsfx "er" "rati"  # ["n----"],
   rsfx "er" "rat"   # ["n"],
   sfx "em"          # ["ip1s"],
   sfx "eš"          # ["ip2s"],
   sfx "e"           # ["ip3s"],
   sfx "emo"         # ["ip1p"],
   sfx "ete"         # ["ip2p"],
   sfx "u"           # ["ip3p"],
   sfx "i"           # ["m-2s"],
   sfx "ite"         # ["m-1p"],
   sfx "imo"         # ["m-1p"],
   sfx "ući"         # ["gp"],
   rsfx "er" "ravši" # ["gs"],
   rsfx "er" "rao"   # ["pp-s"],
   rsfx "er" "rala"  # ["pp-s"],
   rsfx "er" "ralo"  # ["pp-s"],
   rsfx "er" "rali"  # ["pp-p"],
   rsfx "er" "rale"  # ["pp-p"]]

v22 = iPattern "V22" "Vm"  -- zvati (v5r3c)
  (ends ["v"])
  [rsfx "ov" "vati"  # ["n"],
   rsfx "ov" "vat"   # ["n"],
   sfx "em"          # ["ip1s"],
   sfx "eš"          # ["ip2s"],
   sfx "e"           # ["ip3s"],
   sfx "emo"         # ["ip1p"],
   sfx "ete"         # ["ip2p"],
   sfx "u"           # ["ip3p"],
   sfx "i"           # ["m-2s"],
   sfx "ite"         # ["m-1p"],
   sfx "imo"         # ["m-2p"],
   sfx "ući"         # ["gp"],
   rsfx "ov" "vavši" # ["gs"],
   rsfx "ov" "vao"   # ["pp-s"],
   rsfx "ov" "vala"  # ["pp-s"],
   rsfx "ov" "valo"  # ["pp-s"],
   rsfx "ov" "vali"  # ["pp-p"],
   rsfx "ov" "vale"  # ["pp-p"]]

v23 = iPattern "V23" "Vm"  -- sijati (v5r4a)
  (ends ["j"])
  [sfx "ati"        # ["n"],
   sfx "at"         # ["n"],
   sfx "em"         # ["ip1s"],
   sfx "eš"         # ["ip2s"],
   sfx "e"          # ["ip3s"],
   sfx "emo"        # ["ip1p"],
   sfx "ete"        # ["ip2p"],
   sfx "u"          # ["ip3p"],
   nul              # ["m-2s"],
   sfx "mo"         # ["m-1p"],
   sfx "te"         # ["m-2p"],
   sfx "ući"        # ["gp"],
   sfx "avši"       # ["gs"],
   sfx "ao"         # ["pp-s"],
   sfx "ala"        # ["pp-s"],
   sfx "alo"        # ["pp-s"],
   sfx "ali"        # ["pp-p"],
   sfx "ale"        # ["pp-p"]]

v24 = iPattern "V24" "Vm"  -- kljuvati (v5r4b)
  always
  [sfx "vati"       # ["n"],
   sfx "vat"        # ["n"],
   sfx "jem"        # ["ip1s"],
   sfx "ješ"        # ["ip2s"],
   sfx "je"         # ["ip3s"],
   sfx "jemo"       # ["ip1p"],
   sfx "jete"       # ["ip2p"],
   sfx "ju"         # ["ip3p"],
   sfx "j"          # ["m-2s"],
   sfx "jmo"        # ["m-1p"],
   sfx "jte"        # ["m-2p"],
   sfx "jući"       # ["gp"],
   sfx "vavši"      # ["gs"],
   sfx "vao"        # ["pp-s"],
   sfx "vala"       # ["pp-s"],
   sfx "valo"       # ["pp-s"],
   sfx "vali"       # ["pp-p"],
   sfx "vale"       # ["pp-p"]]

v25 = iPattern "V25" "Vm"  -- kupovati (v6a)
  always
  [sfx "ovati"      # ["n"],
   sfx "ovat"       # ["n"],
   sfx "ujem"       # ["ip1s"],
   sfx "uješ"       # ["ip2s"],
   sfx "uje"        # ["ip3s"],
   sfx "ujemo"      # ["ip1p"],
   sfx "ujete"      # ["ip2p"],
   sfx "uju"        # ["ip3p"],
   sfx "uj"         # ["m-2s"],
   sfx "ujmo"       # ["m-1p"],
   sfx "ujte"       # ["m-2p"],
   sfx "ujući"      # ["gp"],
   sfx "ovavši"     # ["gs"],
   sfx "ovao"       # ["pp-s"],
   sfx "ovala"      # ["pp-s"],
   sfx "ovalo"      # ["pp-s"],
   sfx "ovali"      # ["pp-p"],
   sfx "ovale"      # ["pp-p"]]

v26 = iPattern "V26" "Vm"  -- kraljevati (v6b)
```

```
  always
  [sfx "evati"      # ["n"],
   sfx "evat"       # ["n"],
   sfx "ujem"       # ["ip1s"],
   sfx "uješ"       # ["ip2s"],
   sfx "uje"        # ["ip3s"],
   sfx "ujemo"      # ["ip1p"],
   sfx "ujete"      # ["ip2p"],
   sfx "uju"        # ["ip3p"],
   sfx "uj"         # ["m-2s"],
   sfx "ujmo"       # ["m-1p"],
   sfx "ujte"       # ["m-2p"],
   sfx "ujući"      # ["gp"],
   sfx "evavši"     # ["gs"],
   sfx "evao"       # ["pp-s"],
   sfx "evala"      # ["pp-s"],
   sfx "evalo"      # ["pp-s"],
   sfx "evali"      # ["pp-p"],
   sfx "evale"      # ["pp-p"]]

v27 = iPattern "V27" "Vm"  -- kazivati (v6c)
  always
  [sfx "ivati"      # ["n"],
   sfx "ivat"       # ["n"],
   sfx "ujem"       # ["ip1s"],
   sfx "uješ"       # ["ip2s"],
   sfx "uje"        # ["ip3s"],
   sfx "ujemo"      # ["ip1p"],
   sfx "ujete"      # ["ip2p"],
   sfx "uju"        # ["ip3p"],
   sfx "uj"         # ["m-2s"],
   sfx "ujmo"       # ["m-1p"],
   sfx "ujte"       # ["m-2p"],
   sfx "ujući"      # ["gp"],
   sfx "ivavši"     # ["gs"],
   sfx "ivao"       # ["pp-s"],
   sfx "ivala"      # ["pp-s"],
   sfx "ivalo"      # ["pp-s"],
   sfx "ivali"      # ["pp-s"],
   sfx "ivale"      # ["pp-s"]]

-- nepravilni glagoli:

v28 = iPattern "V28" "Vm"  -- htjeti (621)
  (ends ["hoć"])
  [rsfx "oć" "tjeti"  # ["n"],
   rsfx "oć" "tjet"   # ["n"],
   sfx "u"            # ["ip1s"],
   sfx "eš"           # ["ip2s"],
   sfx "e"            # ["ip3s","ip3p"],
   sfx "emo"          # ["ip1p"],
   sfx "ete"          # ["ip2p"],
   rsfx "oć" "otećі"  # ["gp"],
   rsfx "oć" "tijući" # ["gp"],
   rsfx "oć" "tjevši" # ["gs"],
   rsfx "oć" "tio"    # ["pp-s"],
   rsfx "oć" "tjela"  # ["pp-s"],
   rsfx "oć" "tjelo"  # ["pp-s"],
   rsfx "oć" "tjeli"  # ["pp-s"],
   rsfx "oć" "tjele"  # ["pp-s"]]

v29 = iPattern "V29" "Vm"  -- ići (623)
  always
  [sfx "ći"         # ["n"],
   sfx "ć"          # ["n"],
   sfx "dem"        # ["ip1s"],
   sfx "deš"        # ["ip2s"],
   sfx "de"         # ["ip3s"],
   sfx "demo"       # ["ip1p"],
   sfx "dete"       # ["ip2p"],
   sfx "du"         # ["ip3p"],
   sfx "di"         # ["m-2s"],
   sfx "dimo"       # ["m-1p"],
   sfx "dite"       # ["m-2p"],
   sfx "šavši"      # ["gs"],
   sfx "šao"        # ["pp-s"],
   sfx "šla"        # ["pp-s"],
   sfx "šlo"        # ["pp-s"],
   sfx "šli"        # ["pp-s"],
   sfx "šle"        # ["pp-s"]]

v30 = iPattern "V30" "Vm"  -- doći, naći (624)
  always
  [sfx "ći"         # ["n"],
   sfx "ć"          # ["n"],
   sfx "đem"        # ["ip1s"],
   sfx "đeš"        # ["ip2s"],
   sfx "đe"         # ["ip3s"],
   sfx "đemo"       # ["ip1p"],
   sfx "đete"       # ["ip2p"],
   sfx "đu"         # ["ip3p"],
   sfx "đi"         # ["m-2s"],
   sfx "đimo"       # ["m-1p"],
   sfx "đite"       # ["m-2p"],
   sfx "šavši"      # ["gs"],
   sfx "šao"        # ["pp-s"],
   sfx "šla"        # ["pp-s"],
   sfx "šlo"        # ["pp-s"],
   sfx "šli"        # ["pp-s"],
   sfx "šle"        # ["pp-s"]]

v31 = iPattern "V31" "Vm"  -- moći
  (ends ["g"])
  [rsfx "g" "ći"    # ["n"],
   sfx "u"          # ["ip1s","ip3p"],
   sfx "eš" & plt   # ["ip2s"],
   sfx "e" & plt    # ["ip3s"],
   sfx "emo" & plt  # ["ip1p"],
   sfx "ete" & plt  # ["ip2p"],
   sfx "avši"       # ["gs"],
   sfx "ao"         # ["pp-s"],
   sfx "la"         # ["pp-s"],
   sfx "lo"         # ["pp-s"],
   sfx "li"         # ["pp-p"],
   sfx "le"         # ["pp-p"]]
```

B.2.3 Pridjevski uzorci

```
-- poduzorci za neodređeni vid

-- poduzorak za osnove koje završavaju na nepčani sugl. (palatal)
ai01 = [
  nul        # ["msn","msa","msv"],
  sfx "a"    # ["msg","nsg","npn","npa","npv","fsn","fsv"],
  sfx "u"    # ["msd","msl","nsd","nsl","fsa"],
  sfx "im"   # ["msi","mpd","mpl","mpi","nsi","npd","npl",
             "npi","fpd","fpl","fpi"],
  sfx "i"    # ["mpn","mpv"],
  sfx "ih"   # ["mpg","npg","fpg"],
  sfx "ima"  # ["mpd","mpl","mpi","npd","npl","npi","fpd",
             "fpl","fpi"],
  sfx "e"    # ["mpa","fsg","fpn","fpa","fpv"],
  sfx "o"    # ["nsn","nsa","nsv"],
  sfx "oj"   # ["fsd","fsl"],
  sfx "om"   # ["fsi"]]

-- poduzorak za osnove koje završavaju na nenepčani suglasnik
ai02 = [
  nul        # ["msn","msa","msv"],
  sfx "a"    # ["msg","nsg","npn","npa","npv","fsn","fsv"],
  sfx "u"    # ["msd","msl","nsd","nsl","fsa"],
  sfx "im"   # ["msi","mpd","mpl","mpi","nsi","npd","npl",
             "npi","fpd","fpl","fpi"],
  sfx "i"    # ["mpn","mpv"],
  sfx "ih"   # ["mpg","npg","fpg"],
  sfx "ima"  # ["mpd","mpl","mpi","npd","npl","npi","fpd",
             "fpl","fpi"],
  sfx "e"    # ["mpa","nsn","nsa","nsv","fsg","fpn","fpa",
             "fpv"],
  sfx "oj"   # ["fsd","fsl"],
  sfx "om"   # ["fsi"]]

-- poduzorci za određeni vid

-- poduzorak za osnove koje završavaju na nenepčani suglasnik (nonpals)
ad01 = [
  sfx "i"    # ["msn","msa","msv","mpn","mpv"],
  sfx "og"   # ["msg","nsg"],
  sfx "oga"  # ["msg","nsg"],
  sfx "om"   # ["msd","msl","nsd","nsl","fsi"],
  sfx "ome"  # ["msd","msl","nsd","nsl"],
  sfx "omu"  # ["msd","msl","nsd","nsl"],
  sfx "im"   # ["msi","mpd","mpl","mpi","nsi","npd","npl",
             "npi","fpd","fpl","fpi"],
  sfx "ih"   # ["mpg","npg","fpg"],
  sfx "ima"  # ["mpd","mpl","mpi","npd","npl","npi","fpd",
             "fpl","fpi"],
  sfx "e"    # ["mpa","fsg","fpn","fpa","fpv"],
  sfx "o"    # ["nsn","nsa","nsv"],
  sfx "a"    # ["npn","npa","npv","fsn","fsv"],
  sfx "oj"   # ["fsl"],
  sfx "u"    # ["fsa"]]

-- poduzorak za osnove koje završavaju na nepčani sugl. (palatal)
ad02 = [
  sfx "i"    # ["msn","msa","msv","mpn","mpv","nsn","nsa",
             "nsv"],
  sfx "eg"   # ["msg","nsg"],
  sfx "ega"  # ["msg","nsg"],
  sfx "em"   # ["msd","msl","nsd","nsl"],
  sfx "om"   # ["fsi"],
  sfx "emu"  # ["msd","msl","nsd","nsl"],
  sfx "im"   # ["msi","mpd","mpl","mpi","nsi","npd","npl",
             "npi","fpd","fpl","fpi"],
  sfx "ih"   # ["mpg","npg","fpg"],
  sfx "ima"  # ["mpd","mpl","mpi","npd","npl","npi","fpd",
```

```
                "fpl","fpi"],
  sfx "e"    # ["mpa","nsn","nsa","nsv","fsg","fpn","fpa",
                "fpv"],
  sfx "a"    # ["npn","npa","npv","fsn","fsv"],
  sfx "oj"   # ["fsl"],
  sfx "u"    # ["fsa"]]

-- uzorci:

a01 = iPattern "A01" "Af....."  -- skup->skuplji, pust->pušći
  (ends nonpals ‘land‘
   (neg(ends (cgr ++ ["en","an","on","un","in"])) ‘lor‘
    ends ["st","št","rz","rn","rd"]))
  (ai01 <# ["p...n"] ++
   ad01 <# ["p...y"] ++
   ad02 <& tc <# ["c...-"] ++
   ad02 <& tc <& ts <# ["s...-"])
  where tc = jot & try jat1
        ts = pfx "naj"

a03 = iPattern "A03" "Af....."  -- tuđ->tuđi
  (ends pals ‘land‘
   (nends cgr ‘lor‘ ends ["st", "št"]) ‘land‘ nends ["ć"])
  (ai02 <# ["p...n"] ++
   ad02 <# ["p...y"] ++
   ad02 <& tc <# ["c...-"] ++
   ad02 <& tc <& ts <# ["s...-"])
  where tc = sfx "ij"
        ts = pfx "naj"

a04 = iPattern "A04" "Af....."  -- sretan, koristan, drzak, vrijedan
  (ends cgr)
  ((exa # ["msn","msa","msv"] : tail ai01 <& try pca1) <# ["p...n"] ++
   ad01 <& try pca1 <# ["p...y"] ++
   ad02 <& tc <& try pca1 <# ["c...-"] ++
   ad02 <& tc <& try pca1 <& ts <# ["s...-"])
  where tc  = sfx "ij" & opt (jat1 .|. jat2)
        ts  = pfx "naj"

a06 = iPattern "A06" "Af....."  -- star->stariji, loš->lošiji
  (ends (nonpals ++ ["š"]) ‘land‘
   (nends cgr ‘lor‘ ends ["st", "št", "rt","rn","rm"]))
  (ai01 <# ["p...n"] ++
   ad01 <# ["p...y"] ++
   ad02 <& tc <# ["c...-"] ++
   ad02 <& tc <& ts <# ["s...-"])
  where tc = sfx "ij"
        ts = pfx "naj"

a07 = iPattern "A07" "Af....."  -- lak->lakši, mek->mekši
  (ends ["lak","mek","lijep"])
  (ai01 <# ["p...n"] ++
   ad01 <# ["p...y"] ++
   ad02 <& tc <# ["c...-"] ++
   ad02 <& tc <& ts <# ["s...-"])
  where tc = sfx "š"
        ts = pfx "naj"

a08 = iPattern "A08" "Asp...y"  -- arapski
  (ends ["sk", "šk","čk","ćk"])
  ad01

a09 = iPattern "A09" "Asp...y"  -- kozji
  (ends ["j"] ‘land‘ nends ["nj"])
  ad02

a10 = iPattern "A10" "Asp...n"  -- sinov
  (ends ["ov", "ev", "ljev", "in"])
  (ai01 ++
   [sfx "om"  # ["msd","msl","nsd","nsl"],
    sfx "og"  # ["msd","nsg","nsl"],
    sfx "oga" # ["msd","nsg","nsl"]])

a11 = iPattern "A11" "Af....."  -- zao, podao
```

```
  (ends cgr ‘land‘ ends ["l"])
  ((t0 # ["msn","msv"] : tail ai01) <# ["p...n"] ++
   ad01 <# ["p...y"] ++
   ad02 <& tc <# ["c...-"] ++
   ad02 <& tc <& ts <# ["s...-"])
  where t0 = rsfx "l" "ao"
        tc = sfx "ij"
        ts = pfx "naj"

a12 = iPattern "A12" "Af....y"  -- sinji, idući, drugačiji
  (ends ["ć","n","v","k","nj","t","lj","r","đ","š","j"])
  ad02

a13 = iPattern "A13" "Af....."  -- desni, siroti, mali
  (ends ["ć","n","v","k","nj","t","l"])
  (ad01 <# ["p...y"] ++
   ad02 <& tc <# ["c...-"] ++
   ad02 <& tc <& ts <# ["s...-"])
  where tc = sfx "ij"
        ts = pfx "naj"

a14 = iPattern "A14" "Af....."  -- zreo, preminuo
  (ends ["el","ul","al","il","rl"])
  ((t0 # ["msn","msv"] : tail ai01) <# ["p...n"] ++
   ad01 <# ["p...y"] ++
   ad02 <& tc <# ["c...-"] ++
   ad02 <& tc <& ts <# ["s...-"])
  where t0 = rsfx "l" "o"
        tc = sfx "ij"
        ts = pfx "naj"

a15 = iPattern "A15" "Af....."  -- kratak, nizak, rijedak, težak, sladak
  (ends ["k"])
  ((exa # ["msn","msa","msv"] : tail ai01 <& try pca1) <# ["p...n"] ++
   ad01 <& try pca1 <# ["p...y"] ++
   ad02 <& tc <& dsfx "k" <# ["c...-"] ++
   ad02 <& tc <& dsfx "k" <& ts <# ["s...-"])
  where tc = try jot & try jat1
        ts = pfx "naj"
```

B.2.4 Flektivne kategorije

```
nouns = mNouns ++ nNouns ++ fNouns

verbs = tiVerbs ++ ciVerbs

adjectives = pAdjectives ++ qAdjectives

qAdjectives = qndAdjectives ++ qdAdjectives

mNouns = [
  n01,n02,n03,n04,n05,n06,n07,n08,n09,n10,
  n11,n12,n13,n14,n15,n16,n17,n18,n19,
  n30,n37,n38,n41,n43,n44,n45,n46,n48]

nNouns = [
  n20,n21,n22,n23,n24,n25,n26,n27,n40]

fNouns = [
  n28,n47,n29,n31,n32,n33,n34,n35,n36,n39,n49]

tiVerbs = [
  v01,v02,v03,v04,v05,v06,v07,
  v11,v12,v13,v14,v15,v16,v17,v18,v19,v20,
  v21,v22,v23,v24,v25,v26,v27,v28,v32]
```

```
ciVerbs = [v08,v09,v10,v29,v30,v31]

pAdjectives = [a08,a09,a10]

qndAdjectives = [a01,a03,a04,a06,a07,a11,a14,a15]

qdAdjectives = [a12,a13]
```

B.3 Definicije derivacijskih uzoraka

B.3.1 Sufiksalna tvorba imenica

```
-- Div : imenica m. rod kao vrsitelj radnje
dN01 = [
  DPattern "iv01"
    -- kositi -> kosac
    (sfx "ac") tiVerbs mNouns,
  DPattern "iv02"
    -- misliti -> mislilac
    (rsfx "ti" "lac") tiVerbs mNouns,
  DPattern "iv03"
    -- orati -> orač, pripovijedati -> pripovjedač
    (sfx "ač" & try jat1) tiVerbs mNouns,
  DPattern "iv04"
    -- čuvati -> čuvar
    (sfx "ar") tiVerbs mNouns,
  DPattern "iv05"
    -- kišobran -> kišobranar, mlijeko->mljekar, higijena->higijeničar
    (sfx "ar" & opt jat1) nouns mNouns,
  DPattern "iv06"
    -- servis -> serviser
    (sfx "er") mNouns mNouns,
  DPattern "iv07"
    -- hokej -> hokejaš
    (sfx "aš") mNouns mNouns,
  DPattern "iv08"
    -- nišan -> nišandžija, oklop -> oklobdžija
    (sfx "džija") mNouns mNouns,
  DPattern "iv09"
    -- voditi -> vodič
    (sfx "ič") tiVerbs mNouns,
  DPattern "iv10"
    -- bicikl -> biciklist
    (sfx "ist" ) mNouns mNouns,
  DPattern "iv11"
    -- finalan -> finalist
    (sfx "ist" ) qAdjectives mNouns,
  DPattern "iv12"
    -- potpis -> potpisnik, prijevoz -> prijevoznik
    (sfx "nik") mNouns mNouns,
  DPattern "iv13"
    -- pobjeda -> pobjednik, pravo -> pravnik
    (sfx "nik") (fNouns++nNouns) mNouns,
  DPattern "iv14"
    -- pripadati -> pripadnik, zapovijedati -> zapovjednik
    (sfx "nik" & try jat1) tiVerbs mNouns,
  DPattern "iv15"
    -- voditi -> voditelj, izvijestiti -> izvjestitelj
    (sfx "telj" & try jat1) tiVerbs mNouns,
  DPattern "iv16"
    -- tvornica -> tvorničar, biblioteka -> bibliotekar,
    -- mehanika->mehaničar, mlijeko -> mljekar
```

```
    (sfx "ar" & try jat1 & opt jot) (fNouns++nNouns) mNouns,
  DPattern "iv17"
    -- akademija -> akademik
    (rsfx "ij" "ik") fNouns mNouns,
  DPattern "iv18"
    -- alkemija->alkemičar
    (rsfx "ij" "ičar") fNouns mNouns,
  DPattern "iv19"
    -- alkohol->alkoholičar
    (sfx "ičar") mNouns mNouns,
  DPattern "iv20"
    -- alat->alatničar
    (sfx "ničar") mNouns mNouns,
  DPattern "iv21"
    -- dirigirati -> dirigent
    (rsfx "ir" "ent") tiVerbs mNouns]

-- Div : imenica m. rod kao nositelj osobine
dN02 = [
  DPattern "io01"
    -- lakom -> lakomac
    (sfx "ac") qAdjectives mNouns,
  DPattern "io02"
    -- sretan -> sretnik, glazben -> glazbenik
    (sfx "ik") qAdjectives mNouns,
  DPattern "io03"
    -- slobodan -> slobodnjak
    (sfx "ak" & jot) qAdjectives mNouns,
  DPattern "io04"
    -- afera -> aferaš
    (sfx "aš") fNouns mNouns,
  DPattern "io05"
    -- droga -> drogeraš
    (sfx "eraš") fNouns mNouns]

-- Dislj: imenica m roda kao sljedbenik
dN03 = [
  DPattern "islj01"
    -- dinamo -> dinamovac, hajduk -> hajdukovac
    (sfx "ovac") mNouns mNouns,
  DPattern "islj02"
    -- franjo -> franjevac, skoj -> skojevac
    (sfx "evac") mNouns mNouns,
  DPattern "islj03"
    -- parlamentaran -> parlamentarac
    (rsfx "n" "ac") qAdjectives mNouns]

-- Diz: imenice koje znace zensku osobu
dN04 = [
  DPattern "iz01"
    -- kum -> kuma
    (sfx "a") mNouns fNouns,
  DPattern "iz02"
    -- glupan -> glupača
    (rsfx "n" "ča") mNouns fNouns,
  DPattern "iz03"
    -- rukometaš -> rukometašica
    (sfx "ica") mNouns fNouns,
  DPattern "iz04"
    -- junak -> junakinja, arheolog -> arheologinja
    (sfx "inja") mNouns fNouns,
  DPattern "iz05"
    -- bolničar -> bolničarka, mljekar -> mljekarka
    (sfx "ka") mNouns fNouns,
  DPattern "iz06"
    -- državljanin -> državljanka
    (rsfx "in" "ka") mNouns fNouns,
  DPattern "iz07"
   -- stranac -> strankinja, sudac -> sutkinja
   -- stranac -. strankinja, sluga -> sluškinja
    (sfx "inja" & rsfx "c" "k") mNouns fNouns,
  DPattern "iz08"
    -- rob -> ropkinja
    (sfx "kinja") mNouns fNouns,
```

```
    DPattern "iz09"
      -- mahnit -> mahnitulja
      (sfx "ulja") qAdjectives fNouns,
    DPattern "iz10"
      -- klepet -> klepetuša
      (sfx "uša") mNouns fNouns,
    DPattern "iz11"
      -- disk -> diskašica, košarka -> košarkašica
      (sfx "ašica") (mNouns++fNouns) fNouns]

-- Dimzo: imenice koje znače podjednako mušku i žensku osobu
dN05 = [
    DPattern "imzo01"
      -- ništa -> ništarija
      (sfx "arija") (fNouns++nNouns) fNouns,
    DPattern "imzo02"
      -- pastir -> pastirče
      (sfx "če") mNouns mNouns,
    DPattern "imzo03"
      -- varati -> varalica
      (sfx "alica") tiVerbs mNouns,
    DPattern "imzo04"
      -- izdati -> izdajica
      (sfx "ajica") tiVerbs mNouns,
    DPattern "imzo05"
      -- piskarati -> piskaralo
      (sfx "alo") tiVerbs nNouns]

-- Dime: etici
dN06 = [
    DPattern "ime01"
      -- varaždin -> varaždinac
      (sfx "ac") mNouns mNouns,
    DPattern "ime02"
      -- afrika -> afrikanac
      (sfx "anac") fNouns mNouns,
    DPattern "ime03"
      -- beč -> bečanin, karlovac -> karlovčanin
      -- atena -> atenjanin
      (sfx "anin" & try jot) (mNouns++fNouns) mNouns,
    DPattern "ime04"
      -- slavonija -> slavonac
      (rsfx "ij" "ac") fNouns mNouns,
    DPattern "ime05"
      -- bjelovar -> bjelovarčanin, zagreb->zagrepčanin
      (sfx "čanin") mNouns mNouns,
    DPattern "ime06"
      -- hrvat -> hrvatica
      (sfx "ica") mNouns fNouns,
    DPattern "ime07"
      -- čeh -> čehinja
      (sfx "inja") mNouns fNouns,
    DPattern "ime08"
      -- rumunj -> rumunjka
      (sfx "ka") mNouns fNouns,
    DPattern "ime09"
      -- slavonac -> slavonka
      (rsfx "ac" "ka") mNouns fNouns,
    DPattern "ime10"
      -- danac -> dankinja
      (rsfx "c" "kinja") mNouns fNouns,
    DPattern "ime11"
      -- arapin -> arapkinja, srbin -> srpkinja,
      -- bugarin -> bugarkinja, francuz -> franzuskinja
      (sfx "kinja") mNouns fNouns]

-- Dimzb: životinje i bilje
dN07 = [
    DPattern "izb01"
      -- školjka -> školjkaš
      (sfx "aš") fNouns mNouns,
    DPattern "izb02"
      -- mrk -> mrkonja
      (sfx "onja") qAdjectives mNouns,
```

```
    DPattern "izb03"
      -- otrovan -> otrovnica, grabljiv -> grabljivica
      (sfx "ica") qAdjectives fNouns,
    DPattern "izb04"
      -- raditi -> radilica
      (sfx "ilica") tiVerbs fNouns,
    DPattern "izb05"
      -- kreketati -> kreketuša
      (sfx "uša") tiVerbs fNouns,
    DPattern "izb06"
      -- cvijet -> cvjetača
      (sfx "ača" & try jat1) mNouns fNouns]

-- Dis: stvari
dN08 = [
    DPattern "is01"
      -- brisati -> brisač, mijenjati -> mjenjač, peglati -> pegljač
      -- nositi -> nosač, upaliti -> upaljač
      (sfx "ač" & opt jot & try jat1) tiVerbs mNouns,
    DPattern "is02"
      -- kuhati -> kuhača
      (sfx "ača") tiVerbs fNouns,
    DPattern "is03"
      -- kapati -> kapaljka
      (sfx "aljka") tiVerbs fNouns,
    DPattern "is04"
      -- cijediti -> cjediljka
      (sfx "iljka" & try jat1) tiVerbs fNouns,
    DPattern "is05"
      -- bušiti -> bušilica, hraniti -> hranilica
      (sfx "ilica") tiVerbs fNouns,
    DPattern "is06"
      -- dizati -> dizalo
      (sfx "alo") tiVerbs fNouns,
    DPattern "is07"
      -- svinja -> svinjetina, tele -> teletina
      (sfx "etina") (fNouns++nNouns) fNouns,
    DPattern "is08"
      -- borov -> borovina
      (sfx "ina") pAdjectives fNouns,
    DPattern "is09"
      -- darovan -> darovnica
      (sfx "ica") qAdjectives fNouns,
    DPattern "is10"
      -- cijena -> cjenik
      (sfx "ik" & try jat1) fNouns mNouns,
    DPattern "is11"
      -- vježba -> vježbenica
      (sfx "enica") fNouns fNouns]

-- Dim: mjesne imenice
dN09 = [
    DPattern "im01"
      -- cigla -> ciglana
      (sfx "na") fNouns fNouns,
    DPattern "im02"
      -- čaj -> čajana
      (sfx "ana") mNouns fNouns,
    DPattern "im03"
      -- piliti -> pilana,
      (sfx "ana") tiVerbs fNouns,
    DPattern "im04"
      -- brijač -> brijačnica, cvjećar -> cvjećarnica
      (sfx "nica") mNouns fNouns,
    DPattern "im05"
      -- čekati -> čekaonica, gostiti -> gostionica,
      -- ispovijedati -> ispovjedaonica, štedjeti -> štedionica
      -- vježbati -> vježbaonica
      (sfx "aonica" & try jat1) tiVerbs fNouns,
    DPattern "im06"
      -- gostiti -> gostionica, štedjeti -> štedionica
      (sfx "ionica" & try jat2) tiVerbs fNouns,
    DPattern "im07"
      -- boj -> bojište
```

```
    (sfx "ište") mNouns nNouns,
  DPattern "im08"
    -- polaziti -> polazište, hvatati -> hvatište
    (sfx "ište") tiVerbs nNouns,
  DPattern "im09"
    -- graditi -> gradilište
    (sfx "ilište") tiVerbs nNouns]

-- Dia: apstraktne (mislene) imenice
dN10 = [
  DPattern "ia01"
    -- grub -> grubost, unutrašnji -> unutrašnjost,
    -- kiseo -> kiselost, suvisao -> suvislost, gibak -> gipkost
    (sfx "ost") qAdjectives fNouns,
  DPattern "ia02"
    -- prijatelj -> prijateljstvo, rob -> ropstvo,
    (sfx "stvo") mNouns nNouns,
  DPattern "ia03"
    -- lud -> ludilo
    (sfx "ilo") qAdjectives nNouns,
  DPattern "ia04"
    -- trijezan -> treznoća, bijesan -> bjesnoća, gladak -> glatkoća,
    -- kratak -> kratkoća, težak -> teškoća, gluh -> gluhoća
    (sfx "oća" & try (jat1 .|. jat2))
    qAdjectives fNouns,
  DPattern "ia05"
    -- brz -> brzina, bijel -> bjelina
    (sfx "ina" & try jat1) qAdjectives fNouns,
  DPattern "ia06"
    -- strah -> strahota
    (sfx "ota") mNouns fNouns,
  DPattern "ia07"
    -- dobar -> dobrota, vrijedan -> vrednota/vrjednota
    (sfx "ota" & try (jat1 .|. jat2)) qAdjectives fNouns,
  DPattern "ia08"
    -- divan -> divota
    (rsfx "an" "ota") qAdjectives fNouns,
  DPattern "ia09"
    -- gol -> golotinja
    (sfx "otinja") qAdjectives fNouns,
  DPattern "ia10"
    -- prljav -> prljavština
    (sfx "ština") qAdjectives fNouns,
  DPattern "ia11"
    -- raketa -> raketiranje
    (sfx "iranje") fNouns nNouns,
  DPattern "ia12"
    -- bogat -> bogatstvo
    (sfx "stvo") qAdjectives nNouns,
  DPattern "ia13"
    -- arhitekt -> arhitektura
    (sfx "ura") mNouns fNouns,
  DPattern "ia14"
    -- alkohol -> alkoholizam, turist -> turizam,
    -- barbarin -> barbarizam, boljševik -> boljševizam
    (sfx "izam" & try (dsfx "ist" .|. dsfx "ik")) mNouns mNouns,
  DPattern "ia15"
    -- avangardan -> avangardizam
    (rsfx "n" "izam") qAdjectives mNouns,
  DPattern "ia16"
    -- dokument -> dokumentacija
    (sfx "acija") mNouns fNouns,
  DPattern "ia17"
    -- drama -> dramatika
    (sfx "atika") fNouns fNouns,
  DPattern "ia18"
    -- centralan -> centralizacija, kolektivan -> kolektivizacija
    (rsfx "n" "izacija") qAdjectives fNouns]

-- Dig: glagolske imenice
dN11 = [
  DPattern "ig01"
    -- čuvati -> čuvanje
    (sfx "anje") tiVerbs nNouns,
```

```
1743    DPattern "ig02"
1744      -- otvoriti -> otvorenje, dojiti -> dojenje,
1745      -- unaprijediti -> unapređenje/unaprjeđenje,
1746      -- čistiti -> čišćenje, doseliti -> doseljenje,
1747      -- koristiti -> korištenje, bditi -> bdjenje
1748      (sfx "enje" & try (jat1 .|. jat2) &
1749       opt (acg .||. jot)) tiVerbs nNouns,
1750    DPattern "ig03"
1751      -- otkriti -> otkriće, raspeti -> raspeće,
1752      -- dospjeti -> dospijeće
1753      (sfx "će" & try jat1) tiVerbs nNouns,
1754    DPattern "ig04"
1755      -- izvoziti -> izvoz, upisati -> upis,
1756      -- izdahnuti->izdah, prelaziti -> prijelaz
1757      (opt (inv jat3)) tiVerbs mNouns,
1758    DPattern "ig05"
1759      -- boriti -> borba
1760      (sfx "ba") tiVerbs fNouns,
1761    DPattern "ig06"
1762      -- gnojiti -> gnojidba, kruniti -> krunidba,
1763      -- plijeniti -> pljenidba
1764      (sfx "dba" & try jat1) tiVerbs fNouns,
1765    DPattern "ig07"
1766      -- jecati -> jecaj, natjecati -> natječaj,
1767      -- odnositi -> odnošaj
1768      (sfx "aj" & opt (acg .|. jot)) tiVerbs mNouns,
1769    DPattern "ig08"
1770      -- dobiti -> dobitak, ostati -> ostatak
1771      (sfx "itak" .|. sfx "atak") tiVerbs mNouns,
1772    DPattern "ig09"
1773      -- izlaziti -> izlazak
1774      (sfx "ak") tiVerbs mNouns,
1775    DPattern "ig10"
1776      -- tutnjati -> tutnjava
1777      (sfx "ava") tiVerbs fNouns,
1778    DPattern "ig11"
1779      -- grabiti -> grabež
1780      (sfx "ež") tiVerbs fNouns,
1781    DPattern "ig12"
1782      -- graditi -> gradnja
1783      (sfx "nja") tiVerbs fNouns,
1784    DPattern "ig13"
1785      -- juriti -> jurnjava, pucati -> pucnjava
1786      (sfx "njava") tiVerbs fNouns,
1787    DPattern "ig14"
1788      -- govoriti -> govorancija, tjerati -> tjerancija
1789      (sfx "ancija") tiVerbs fNouns,
1790    DPattern "ig15"
1791      -- hvatati -> hvatanija
1792      (sfx "anija") tiVerbs fNouns,
1793    DPattern "ig16"
1794     -- održati -> održavanje, udovoljiti -> udovoljavanje
1795      (sfx "avanje") tiVerbs fNouns,
1796    DPattern "ig17"
1797      -- bolovati -> bolovanje
1798      (sfx "anje") tiVerbs fNouns,
1799    DPattern "ig18"
1800      -- tužiti -> tužilaštvo, izdavati -> izdavalaštvo
1801      (sfx "ilaštvo" .|. sfx "alaštvo") tiVerbs nNouns,
1802    DPattern "ig19"
1803      -- tužiti -> tužiteljstvo
1804      (sfx "iteljstvo") tiVerbs nNouns,
1805    DPattern "ig20"
1806      -- ukinuti -> ukinuće
1807      (sfx "uće") tiVerbs nNouns,
1808    DPattern "ig21"
1809      -- administrirati -> administracija
1810      (rsfx "ir" "acija") tiVerbs fNouns,
1811    DPattern "ig22"
1812      -- vladati -> vladavina
1813      (sfx "avina") tiVerbs fNouns,
1814    DPattern "ig23"
1815      -- opeći -> opeklina
1816      (sfx "lina") ciVerbs fNouns,
```

```
    DPattern "ig24"
      -- padati -> padalina
      (sfx "alina" ) tiVerbs fNouns]

-- Diu: umanjenice i uvećanice
dN12 = [
  DPattern "iu01"
    -- crv -> crvić, cvijet -> cvjetić,
    -- članak -> člančić, svezak -> sveščić
    (sfx "ić" & try jat1 & try plt) mNouns mNouns,
  DPattern "iu02"
    -- cvijet -> cvijetak, smijeh -> smiješak
    (sfx "ak" & try plt) mNouns mNouns,
  DPattern "iu03"
    -- plamen -> plamičak
    (rsfx "en" "ičak") mNouns mNouns,
  DPattern "iu04"
    -- crkva -> crkvica, svijeća -> svjećica
    (sfx "ica" & try jat1 & try plt) fNouns fNouns,
  DPattern "iu05"
    -- grana -> grančica, cijev -> cjevčica
    (sfx "čica" & try jat1) fNouns fNouns,
  DPattern "iu06"
    -- jezero -> jezerce, tijelo -> tijelce
    (sfx "ce") nNouns nNouns,
  DPattern "iu07"
    -- vrelo -> vreoce
    (rsfx "l" "oce") nNouns nNouns,
  DPattern "iu08"
    -- društvo -> društvance, bure -> burence
    (sfx "ance" .|. sfx "ence") nNouns nNouns,
  DPattern "iu09"
    -- mjesto -> mjestašce, jaje -> jajašce, dijete -> djetešce
    ((sfx "ašce" .|. sfx "ešce") & try jat1) nNouns nNouns,
  DPattern "iu10"
    -- put -> puteljak
    (sfx "eljak") mNouns mNouns,
  DPattern "iu11"
    -- crv -> crvuljak, brijeg -> brežuljak
    (sfx "uljak" & try jat2 & try plt) mNouns mNouns,
  DPattern "iu12"
    -- pjesma -> pjesmuljak
    (sfx "uljak") fNouns mNouns,
  DPattern "iu13"
    -- komad -> komadina, svijet -> svjetina, junak -> junačina
    (sfx "ina" & try jat1 & try plt) mNouns fNouns,
  DPattern "iu14"
    -- muškarac -> muškarčina, lanac -> lančina
    (rsfx "c" "čina") mNouns (mNouns++fNouns),
  DPattern "iu15"
    -- oficir -> oficirčina
    (sfx "čina") mNouns mNouns,
  DPattern "iu16"
    -- baba -> babetina, ruka -> ručetina
    (sfx "etina" & try plt) fNouns fNouns,
  DPattern "iu17"
    -- trava -> travurina
    (sfx "urina") fNouns fNouns,
  DPattern "iu18"
    -- kuća -> kućerina
    (sfx "erina") fNouns fNouns,
  DPattern "iu19"
    -- glava -> glavešina
    (sfx "ešina") fNouns fNouns]

-- Dia: zbirne imenice
dN13 = [
  DPattern "izbr01"
    -- prase -> prasad
    (sfx "ad") nNouns fNouns,
  DPattern "izbr02"
   -- biser -> biserje, otok -> otočje, okrug -> okružje, naziv -> nazivlje
    (sfx "je" & try jot) mNouns nNouns,
  DPattern "izbr03"
```

```
    -- naziv -> nazivlje, zvijezda -> zviježđe
     (sfx "e" & try (acg .||. jot)) (mNouns++fNouns) nNouns,
  DPattern "izbr04"
    -- sklop -> sklopovlje
    (sfx "ovlje") mNouns nNouns,
  DPattern "izbr05"
    -- slušatelj -> slušateljstvo
    (sfx "stvo") mNouns nNouns,
  DPattern "izbr06"
    -- svećenik -> svećenstvo
    (rsfx "ik" "stvo") mNouns nNouns,
  DPattern "izbr07"
    -- radnik -> radništvo
    (rsfx "ik" "ištvo") mNouns nNouns]

-- Dipzs: ostale pojedinačne značenjske skupine imenica
dN14 = [
  DPattern "ipzs01"
    -- slagati -> slagaljka
    (sfx "aljka") tiVerbs fNouns,
  DPattern "ipzs02"
    -- pošta -> poštarina
    (sfx "rina") fNouns fNouns,
  DPattern "ipzs03"
    -- najam -> najamnina
    (sfx "nina") mNouns fNouns,
  DPattern "ipzs04"
    -- otprema -> otpremnina
    (sfx "nina") fNouns fNouns,
  DPattern "ipzs05"
    -- kajkavski -> kajkavština
    (rsfx "sk" "ština") qAdjectives fNouns,
  DPattern "ipzs06"
    -- kajkavski -> kajkavica
    (rsfx "sk" "ica") qAdjectives fNouns]
```

B.3.2 Sufiksalna tvorba pridjeva

```
-- Dpo: opisni pridjevi
dA01 = [
  DPattern "po01"
    -- blato -> blatan, sunce -> sunčan, mlijeko -> mlječan
    (sfx "an" & try jat1 & try plt) nNouns qAdjectives,
  DPattern "po02"
    -- dvorište -> dvorišni
    (rsfx "t" "ni") nNouns qAdjectives,
  DPattern "po03"
   -- izvor -> izvoran, mrak -> mračan, luk -> lučni
   -- grijeh -> grešan/grješan, smijeh -> smiješan,
    ((sfx "an" .|. sfx "ni") &
     opt (jat1 .|. jat2) & try plt) mNouns qAdjectives,
  DPattern "po04"
    -- novac -> novčani, pijesak -> pješčan
    ((sfx "an" .|. sfx "ani") &
     opt jat1 & try plt) mNouns qAdjectives,
  DPattern "po05"
    -- noga -> nožni, zvijezda -> zvjezdan,
    -- bijeda -> bijedan, kartica -> kartični
    ((sfx "an" .|. sfx "ni") & opt jat1 & try plt)
     fNouns qAdjectives,
  DPattern "po06"
    -- udaranje -> udaran
    (rsfx "anj" "an") nNouns qAdjectives,
  DPattern "po07"
    -- borba -> borben, knjiga -> knjižen, muka -> mučen
    -- spas -> spašen, zdravstvo -> zdravstven
```

```
    (sfx "en" & opt jot) (fNouns++nNouns) qAdjectives,
  DPattern "po08"
    -- masa -> masovan, svijet -> svjetovni
    -- cijena -> cjenovni, banka -> bankovni
    ((sfx "ovan" .|. sfx "ovni") & try jat1) mNouns qAdjectives,
  DPattern "po09"
    -- duša -> duševan, knjiga -> književan
    (sfx "evan" & try plt) fNouns qAdjectives,
  DPattern "po10"
    -- rušiti -> ruševan, platiti -> plaćevan
    (sfx "evan" & try jot) tiVerbs qAdjectives,
  DPattern "po11"
    -- ekonomija -> ekonomičan
    (rsfx "j" "čan") fNouns qAdjectives,
  DPattern "po12"
    -- egoist -> egoističan
    (sfx "ičan") mNouns qAdjectives,
  DPattern "po13"
    -- koža -> kožnat, papir -> papirnat
    (sfx "nat") fNouns qAdjectives,
  DPattern "po14"
    -- anketa -> anketiran
    (sfx "iran") fNouns qAdjectives,
  DPattern "po15"
    -- kesten -> kestenjast, trbuh -> trbušast
    -- maslina -> maslinast, stepenica -> stepeničast
    -- vlakno -> vlaknast
    (sfx "ast" & opt jot) nouns qAdjectives,
  DPattern "po16"
    -- kovrča -> kovrčav
    (sfx "av") fNouns qAdjectives,
  DPattern "po17"
    -- krilo -> krilat, glava -> glavat, rep -> repat
    (sfx "at") nouns qAdjectives,
  DPattern "po18"
    -- milost -> milostiv
    (sfx "iv" & opt jot) fNouns qAdjectives,
  DPattern "po19"
    -- briga -> brižljiv, poruga -> porugljiv
    (sfx "ljiv" & try plt) fNouns qAdjectives,
  DPattern "po20"
    -- brdo -> brdovit, kiša -> kišovit
    (sfx "vit" & try (jat1 .|. jat2)) nouns qAdjectives,
  DPattern "po21"
    -- krš -> kršovit, lijek -> ljekovit, brijeg -> bregovit
    (sfx "ovit" & try (jat1 .|. jat2)) mNouns qAdjectives,
  DPattern "po22"
    -- sloj -> slojevit, stupanj -> stupnjevit
    (sfx "evit") mNouns qAdjectives,
  DPattern "po23"
    -- izbrojati -> izbrojiv, ostvariti -> ostvariv
    (sfx "iv") tiVerbs qAdjectives,
  DPattern "po24"
    -- ostvariti -> ostvarljiv, gledati -> gledljiv
    -- istezati -> istežljiv/istezljiv
    (sfx "ljiv" & opt plt) tiVerbs qAdjectives,
  DPattern "po25"
    -- čitati -> čitak
    (sfx "ak") tiVerbs qAdjectives,
  DPattern "po26"
    -- brljati -> brbljav
    (sfx "av") tiVerbs qAdjectives,
  DPattern "po27"
    -- pun -> puncat
    (sfx "cat") qAdjectives qAdjectives,
  DPattern "po28"
    -- rumen -> rumenkast
    (sfx "kast") qAdjectives qAdjectives,
  DPattern "po29"
    -- plav -> plavičast
    (sfx "ičast") qAdjectives qAdjectives,
  DPattern "po30"
    -- živ -> živahan, mlad -> mlađahan
    (sfx "ahan" & try jot) qAdjectives qAdjectives,
```

```
  DPattern "po31"
    -- pun -> punašan
    (sfx "ašan") qAdjectives qAdjectives,
  DPattern "po32"
    -- dug -> duguljast
    (sfx "uljast") qAdjectives qAdjectives,
  DPattern "po33"
    -- slab -> slabunjav
    (sfx "unjav") qAdjectives qAdjectives,
  DPattern "po34"
    -- koncept -> konceptualan
    (sfx "ualan") mNouns qAdjectives,
  DPattern "po35"
    -- kultura -> kulturalan, centar -> centralan
    (sfx "alan") nouns qAdjectives]

-- Dpp: posvojni pridjevi
dA02 = [
  DPattern "pp01"
    -- djed -> djedov
    (sfx "ov") mNouns pAdjectives,
  DPattern "pp02"
    -- bratić -> bratićev, jakov -> jakovljev, kupac -> kupčev
    -- mjesec -> mjesečev, pisac -> piščev, sunce -> sunčev
    (sfx "ev" & try jot) (mNouns++nNouns) pAdjectives,
  DPattern "pp03"
    -- jerko -> jerkov
    (sfx "ov") mNouns pAdjectives,
  DPattern "pp04"
    -- sin -> sinovljev
    (sfx "ovljev") mNouns pAdjectives,
  DPattern "pp05"
    -- tata -> tatin, mama -> mamin, ivica -> ivičin
    (sfx "in" & try plt) (mNouns++fNouns) pAdjectives,
  DPattern "pp06"
    -- grad -> gradski, klub -> klupski, englez -> engleski, rus -> ruski
    (sfx "ski" & try (asfx [("b","p"),("z",""),("s","")]))
    nouns pAdjectives,
  DPattern "pp07"
    -- slovenija -> slovenski
    (rsfx "ij" "ski") fNouns pAdjectives,
  DPattern "pp08"
    -- dalmacija -> dalmatinski
    (rsfx "cij" "tinski") fNouns pAdjectives,
  DPattern "pp09"
    -- gospić -> gospićki, junak -> junački,
    -- bijelac -> bjelački, brandenburg -> brandenburški
    -- antika -> antički, tvornica -> tvornički
    (sfx "ki" & try jat1 & try plt) nouns pAdjectives,
  DPattern "pp10"
    -- djed -> djedovski
    (sfx "ovski") mNouns pAdjectives,
  DPattern "pp11"
    -- dužd -> duždevski
    (sfx "evski") mNouns pAdjectives,
  DPattern "pp12"
    -- sestra -> sestrinski
    (sfx "inski") fNouns pAdjectives,
  DPattern "pp13"
    -- kapitalist -> kapitalistički
    (sfx "ički") mNouns pAdjectives,
  DPattern "pp14"
    -- zagreb -> zagrebački, valpovo -> valpovački
    (sfx "ački") nouns pAdjectives,
  DPattern "pp15"
    -- koza -> kozji, kukavica -> kukavičji,
    -- patka -> pačji, guska -> guščji
    -- pas -> pasji, bog -> božji
    (sfx "ji" & opt jot) nouns pAdjectives,
  DPattern "pp16"
    -- krava -> kravlji, mrav -> mravlji,
    -- govedo -> goveđi
    (sfx "i" & opt jot) nouns adjectives,
  DPattern "pp17"
```

```
    -- mrav -> mravinji, pčela -> pčelinji
    (sfx "inji") mNouns adjectives,
  DPattern "pp18"
    -- brijati -> brijaći
    (sfx "aći") tiVerbs adjectives,
  DPattern "pp19"
    -- turizam -> turistički
    (rsfx "izm" "istički") mNouns pAdjectives]

-- Dpgt: pridjevi glagolski trpni
dA03 = [
  DPattern "pgt01"
    -- tužiti -> tužen, izraziti -> izražen,
    -- baciti -> bačen, spasiti -> spašen,
    -- tresti -> tresen, tući -> tučen
    (sfx "en" & opt jot) verbs qAdjectives,
  DPattern "pgt02"
    -- adaptirati -> adaptiran, prolaziti -> prolazan
    (sfx "an") tiVerbs qAdjectives,
  DPattern "pgt03"
    -- spomenuti -> spomenut
    (sfx "ut") tiVerbs qAdjectives,
-- ]  spojeno s A04

-- Dpgs : pridjevi koji odgovaraju glagolskom prilogu sadašnjem
-- dA04 = [
  DPattern "pgs01"
    -- zastrašivati -> zastrašujući, iscrpljivati -> isrpljujući,
    -- omamljivati -> omamljujući, onečišćivati -> onečišćujući
    (sfx "ujući") tiVerbs qAdjectives,
  DPattern "pgs02"
    -- dodavati -> dodajući
    (sfx "jući") tiVerbs qAdjectives,
  DPattern "pgs03"
    -- dopunjavati -> dopunjavajući
    (sfx "ajući") tiVerbs qAdjectives,
  DPattern "pgs04"
    -- brisati -> brišući, puzati -> pužući
    (sfx "ući" & try jot) tiVerbs qAdjectives,
  DPattern "pgs05"
    -- dolaziti -> dolazeći, govoriti -> govoreći,
    -- letjeti -> leteći, lebdjeti -> lebdeći
    (sfx "eći") tiVerbs qAdjectives,
  DPattern "pgs06"
    -- bdjeti -> bdijući
    (sfx "ijući") tiVerbs qAdjectives]
```

B.3.3 Sufiksalna tvorba glagola

```
-- Dgv: glagoli iz glagola s promjenom u vidu
dV01 = [
  DPattern "gv01"
   -- baciti -> bacati, prikupiti -> prikupljati
   -- ispratiti -> ispraćati, izmisliti -> izmišljati
    (sfx "ati" & opt (acg .||. jot)) tiVerbs tiVerbs,
  DPattern "gv02"
    -- darovati -> darivati
    -- izvijestiti -> izvješćivati, poskupiti -> poskupljivati,
    (sfx "ivati" & try jat1 & try jot) tiVerbs tiVerbs,
  DPattern "gv03"
    -- odobriti -> odobravati, onečistiti -> onečišćavati,
    -- naseliti -> naseljavati, navodniti -> navodnjavati,
    (sfx "avati" & try jat1 & try jot) tiVerbs tiVerbs,
  DPattern "gv04" --
    -- naviknuti -> navikavati
    (rsfx "n" "avati") tiVerbs tiVerbs,
  DPattern "gv05"
```

```
    -- vikati -> viknuti
    (sfx "nuti") tiVerbs tiVerbs,
  DPattern "gv06"
    -- izliti -> izlijevati, sagoriti -> sagorijevati
    -- razumjeti -> razumijevati
    (sfx "ijevati") tiVerbs tiVerbs,
  DPattern "gv07"
    -- blijediti -> blijedjeti, bluditi -> bludjeti,
        -- bjesniti -> bjesnjeti, oboliti -> oboljeti
    (sfx "jeti") tiVerbs tiVerbs,
  DPattern "gv08"
    -- izroniti -> izranjati, nasloniti -> naslanjati
    (rsfx "on" "anjati") tiVerbs tiVerbs,
  DPattern "gv09"
    -- početi -> počinjati
    (sfx "injati") tiVerbs tiVerbs,
  DPattern "gv10"
    -- nagnuti -> naginjati,
    (rsfx "n" "injati") tiVerbs tiVerbs,
  DPattern "gv11"
    -- proizvoditi -> proizvesti
    (rsfx "oditi" "esti") tiVerbs tiVerbs,
  DPattern "gv12"
    -- napomenuti -> napominjati
    (rsfx "en" "injati") tiVerbs tiVerbs]

-- Dgdp: deminutivni i pejorativni glagoli
dV02 = [
  DPattern "gdp01"
    -- govoriti -> govorkati, šetati -> šetkati
    (sfx "kati") tiVerbs tiVerbs,
  DPattern "gdp02"
    -- pjevati -> pjevuckati
    (sfx "uckati") tiVerbs tiVerbs,
  DPattern "gdp03"
    -- piti -> pijuckati
    (sfx "juckati") tiVerbs tiVerbs,
  DPattern "gdp04"
    -- pjevati -> pjevušiti
    (sfx "ušiti") tiVerbs tiVerbs,
  DPattern "gdp05"
    -- pisati -> piskarati
    (sfx "karati") tiVerbs tiVerbs,
  DPattern "gdp06"
    -- smijati -> smijuljiti
    (sfx "uljiti") tiVerbs tiVerbs]

-- Dgi: glagoli iz imenica
dV03 = [
  DPattern "gi01"
    -- komad -> komadati, večera -> večerati
    (sfx "ati") nouns tiVerbs,
  DPattern "gi02"
    -- bol -> bolovati, tuga -> tugovati
    (sfx "ovati") nouns tiVerbs,
  DPattern "gi03"
    -- lak -> lakirati, adresa -> adresirati
    (sfx "irati") nouns tiVerbs,
  DPattern "gi04"
    -- filozofija -> filozofirati
    (rsfx "ij" "irati") fNouns tiVerbs,
  DPattern "gi05"
    -- grijeh -> griješiti, drug -> družiti, znak -> značiti
    -- zima -> zimiti, granica -> graničiti
    (sfx "iti" & try plt) nouns tiVerbs]

-- Dgp: glagoli iz pridjeva
dV04 = [
  DPattern "gp01"
    -- bijel -> bijeliti
    (sfx "iti") qAdjectives tiVerbs,
  DPattern "gp02"
    -- sitan -> sitniti
    (sfx "iti") qAdjectives tiVerbs]
```

Literatura

Adamson, G. & Boreham, J. 1974. The Use of an Association Measure Based on Character Structure to Identify Semantically Related Pairs of Words and Document Titles. *Information Processing and Management*, **10**(7/8), 253–260.

Agirre, E. & Edmonds, P. (ured.). 2007. *Word Sense Disambiguation: Algorithms and Applications*. 1. izdanje. Springer.

Agić, Ž., Tadić, M. & Bekavac, B. 2009. Evaluating Full Lemmatization of Croatian Texts. *Str. 133–144:* Tadić, M., Dalbelo Bašić, B. & Moens, M.-F. (ured.), *Technologies for the Processing and Retrieval of Semi-Structured Documents: Experience from the CADIAL Project*. Zagreb: Croatian Language Technologies Society.

Ahmad, F., Yussof, M. & Sembok, M. T. 1996. Experiments with a Stemming Algorithm for Malay Words. *Journal of the American Society for Information Science*, **47**(1), 909–918.

Alemayehu, N. & Willett, P. 2002. Stemming of Amharic Words for Information Retrieval. *Literary and Linguistic Computing*, **17**(1), 1–17.

Anić, Velimir. 2003. *Veliki rječnik hrvatskoga jezika*. Novi Liber.

Aronoff, M. & Fudeman, K. A. 2005. *What is morphology?* Wiley-Blackwell.

Baeza-Yates, R. 1992. *Information Retrieval: Data Structures & Algorithms*. Prentice-Hall.

Barić, E., Lončarić, M., Malić, D., Pavešić, S., Peti, M., Zečević, V. & Znika, M. 2005. *Hrvatska gramatika*. 4. izdanje. Školska knjiga.

Baroni, M., Matiasek, J. & Trost, H. 2002. Unsupervised discovery of morphologically related words based on orthographic and semantic similarity. *Str. 48–57: Proceedings*

of the ACL-02 workshop on Morphological and phonological learning. Morristown, NJ, USA: Association for Computational Linguistics.

Bernhard, D. 2010. MorphoNet: Exploring the Use of Community Structure for Unsupervised Morpheme Analysis. *In: Multilingual Information Access Evaluation Vol. I, 10th Workshop of the Cross-Language Evaluation Forum, CLEF 2009, Revised Selected Papers*. Springer. (U tisku.)

Birtić, M. 2006. Nazivlje u generativnoj morfologiji. *Filologija*, 1–18.

Carlberger, J., Dalianis, H., Hassel, M. & Knutsson, O. 2001. Improving Precision in Information Retrieval for Swedish Using Stemming. *Str. 21–22: Proceedings of NODALIDA'01.*

Clement, L., Sagot, B. & Lang, B. 2004. Morphology Based Automatic Acquisition of Large-coverage Lexica. *Str. 1841–1844: Proceedings of LREC'04.*

Creutz, M. & Lagus, K. 2002. Unsupervised discovery of morphemes. *Str. 21–30: Proceedings of the ACL-02 workshop on Morphological and phonological learning*. Morristown, NJ, USA: Association for Computational Linguistics.

Ćavar, D., Jazbec, I.-P. & Stojanov, T. 2008. CroMo – Morphological Analysis for Standard Croatian and its Synchronic and Diachronic Dialects and Variants. *In: Proceedings of FSMNLP 2008.*

Dalbelo Bašić, B., Bereček, B. & Cvitaš, A. 2005. Mining textual data in Croatian. *Str. 61–66: Proceedings of the 28th International Conference MIPRO 2005, Business Intelligence Systems.*

Delač, D., Krleža, Z., Šnajder, J., Dalbelo Bašić, B. & Šarić, F. 2009. TermeX: A Tool for Collocation Extraction. *Lecture Notes in Computer Science (Computational Linguistics and Intelligent Text Processing)*, **5449**, 149–157.

Dice, L. R. 1945. Measures of the amount of ecologic association between species. *Ecology*, **26**(3), 297–302.

Džeroski, S. & Erjavec, T. 2000. Learning to Lemmatise Slovene Words. *Str. 69–88: Learning language in logic, Lecture notes in computer science.*

Ekmekcioglu, F. C., Lynch, M. F. & Willett, P. 1996. Stemming and N-gram Matching for Term Conflation in Turkish Texts. *Information Research News*, **7**(1), 2–6.

Erjavec, T., Krstev, C., Petkevič, V., Simov, K., Tadić, M. & Vitas, D. 2003. The MULTEXT-East Morphosyntactic Specifications for Slavic Languages. *Str. 25–32: Proceedings of the EACL2003 Workshop on Morphological Processing of Slavic Languages.*

Erjavec, T. & Džeroski, S. 2004. Machine Learning of Language Structure: Lemmatising Unknown Slovene Words. *Applied Artificial Intelligence*, **18**(1), 17–41.

Feldman, R. & Sanger, J. 2006. *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data.* Cambridge University Press.

Figuerola, C. G., Gomez, R. & de San Roman, E. L. 2000. Stemming and N-grams in Spanish: an Evaluation of their Impact on Information Retrieval. *Journal of Information Science*, **26**, 461–467.

Forsberg, M. & Ranta, A. 2003. Functional Morphology. *Str. 213–223: Proceedings of the Ninth ACM SIGPLAN International Conference of Functional Programming ICFP'04.*

Forsberg, M., Hammarström, H. & Ranta, A. 2006. Morphological Lexicon Extraction from Raw Text Data. *Str. 488–499: FinTAL.*

Frakes, W. B. & Fox, C. J. 2003. Strength and similarity of affix removal stemming algorithms. *SIGIR Forum*, **37**(1), 26–30.

Freund, G. E. & Willett, P. 1982. Online Identification of Word Variants and Arbitrary Truncation Searching Using a String Similarity Measure. *Information Technology: Research and Development*, **1**, 177–187.

Frost, R. A. 2006. Realization of Natural Language Interfaces Using Lazy Functional Programming. *ACM Computing Surveys*, **38**(4).

Galvez, C. & Moya-Anegón, F. 2006. An Evaluation of Conflation Accuracy Using Finite-state Transducers. *Journal of Documentation*, **62**(3), 328–349.

Galvez, C., de Moya-Anegón, F. & Solana, V. H. 2005. Term Conflation Methods in Information Retrieval: Non-linguistic and Linguistic Approaches. *Journal of Documentation*, **61**(4), 520–547.

Gaussier, E. 1999. Unsupervised Learning of Derivational Morphology from Inflectional Lexicons. *In: ACL '99 Workshop Proceedings: Unsupervised Learning in Natural Language Processing.*

Gaustad, T. & Bouma, G. 2002. *Accurate Stemming of Dutch for Text Classification.*

Gelbukh, A., Alexandrov, M. & Han, S.-Y. 2004. Detecting Inflection Patterns in Natural Language by Minimization of Morphological Model. *Progress in Pattern Recognition, Image Analysis and Applications, LNCS*, **3287**, 432–438.

Goldsmith, J. 2000. Automatic Language-Specific Stemming in Information Retrieval. *Revised Papers from the Workshop of Cross-Language Evaluation Forum on Cross-Language Information Retrieval and Evaluation, LNCS*, **2069**, 273–284.

Goldsmith, J. 2001. Unsupervised Learning of the Morphology of a Natural Language. *Computational Linguistics*, **27**, 153–198.

Hafer, M. & Weiss, S. 1974. Word Segmentation by Letter Successor Varieties. *Information Processing and Management*, **10**(11/12), 371–386.

Hana, J. 2008. Knowledge- and labor-light morphological analysis. *Ohio State University Working Papers in Linguistics*, **58**, 52–84.

Hankin, C. 2004. *An Introduction to Lambda Calculi for Computer Scientists.* King's College Publications.

Harman, D. 1991. How Effective is Suffixing? *Journal of the American Society for Information Science*, **42**(1), 7–15.

Hockett, C. F. 1954. Two models of Grammatical Description. *Word*, **10**, 210–234.

Hudak, P. 1989. Conception, Evolution, and Application of Functional Programming Languages. *ACM Comput. Surv.*, **21**(3), 359–411.

Hudak, P. 2000. *The Haskell School of Expression.* Cambridge University Press.

Hull, D. A. 1996. Stemming Algorithms: A Case Study for Detailed Evaluation. *Journal of the American Society of Information Science*, **47**(1), 70–84.

Jones, S. P. 2003. *Haskell 98 Language and Libraries: The Revised Report.* Tech. rept.

Kalamboukis, T. Z. 1995. Suffix Stripping with Modern Greek. *Program*, **29**(3), 313–321.

Karttunen, L. 1995. The replace operator. *Str. 16–23: Proceedings of the 33rd annual meeting on Association for Computational Linguistics*. Morristown, NJ, USA: Association for Computational Linguistics.

Karttunen, L. 2001. Applications of Finite-State Transducers in Natural Language Processing. *Lecture Notes in Computer Science*, **2088**, 34–46.

Kolaković, Z. 2007. Zastupljenost padeža u hrvatskome jeziku u pisanim i govornim tekstovima. *Lahor: časopis za hrvatski kao materinski, drugi i strani jezik*, **2**(4), 242–270.

Koskenniemi, K. 1983. Two-Level Morphology: A General Computational Model for Word-Form Recognition and Production. Helsinki: Publications of the Department of General Linguistics, University of Helsinki.

Kovalenko, A. 2002. *Stemka: Morphological Analyser for Small Search Systems*.

Kraaij, W. & Pohlman, R. 1995. Evaluation of a Dutch Stemming Algorithm. *The New Review of Document and Text Management*, **1**, 25–43.

Kraaij, W. & Pohlmann, R. 1994. *Porter's Stemming Algorithm for Dutch*.

Kraaij, W. & Pohlmann, R. 1996. Viewing Stemming as Recall Enhancement. *Str. 40–48: Proc. of SIGIR '96*.

Krovetz, R. 1993. Viewing Morphology as an Inference Process. *Str. 191–203: Proceedings of the Sixteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Krstinić, D. & Slapničar, I. 2004. Web Indexing and Search with Local Language Support. *Str. 71–75: Proceedings of SoftCOM 2004*.

Kržak, M. 1988. Serbo-Croatian Morpho-spelling. *In: Proceedings of the 4th Conference Computer Processing of Language Data*.

Kržak, M. & Boras, D. 1985. Lexical Database of the Croatian Literary Language. *Informatologia Yugoslavica*, **17**(3-4), 223–242.

Kurimo, M., Virpioja, S., Turunen, V. & Hirsimäki, T. 2009. Morpho challenge: evaluation of algorithms for unsupervised learning of morphology in various tasks and languages. *Str. 13–16: NAACL '09: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Demonstration Session*. Morristown, NJ, USA: Association for Computational Linguistics.

Larkey, L., Ballesteros, L. & Connell, M. 2002. Improving Stemming for Arabic Information Retrieval: Light Stemming and Co-occurrence Analysis. *Str. 275–282: Proceedings of the 25th Annual international ACM SIGIR conference on research and development in information retrieval*.

Lauc, D., Lauc, T., Boras, D. & Ristov, S. 1998. Developing Text Retrieval System Using Robust Morphological Parsing. *Str. 61–65:* Damir Kalpić, Vesna Hljuz-Dobrić (ured.), *Proceedings of 20th International Conference on Information Technology Interfaces (ITI'98)*. SRCE, Zagreb.

Lauc, T. 2001. *Problemi obrade prirodnog jezika u sustavima za pretraživanje obavijesti putem pretraživanja punoga teksta na hrvatskome književnom jeziku, doktorska disertacija*.

Lenci, A. 2008. Distributional semantics in linguistic and cognitive research. *Italian Journal of Linguistics*, **20**(1), 1–31.

Leopold, E. & Kindermann, J. 2002. Text categorization with support vector machines: how to represent texts in input space? *Machine Learning*, **46**, 423–222.

Lesk, M. 1986. Automatic Sense Disambiguation: How to Tell a Pine Cone from an Ice Cream Cone. *Str. 24–26: Proceedings of the 1986 SIGDOC Conference*. ACM.

Liao, C., Alpha, S. & Dixon, P. 2003. Feature preparation in text categorization. *In: Proceedings of Autralasian Data Mining Workshop*.

Ljubešić, N. 2009. *Pronalaženje događaja u višestrukim izvorima informacija, doktorska disertacija*.

Ljubešić, N., Boras, D. & Kubelka, O. 2007. Retrieving Information in Croatian: Building a Simple and Efficient Rule-based Stemmer. *Str. 313–320: Digital information and heritage*. Zagreb: Odsjek za informacijske znanosti Filozofskog fakulteta u Zagrebu.

Lopina, V. 1992. Dvorazinski opis morfonoloških smjena u pisanome hrvatskom jeziku. *Suvremena lingvistika*, **34**, 185–194.

Lovins, J. B. 1968. Development of a Stemming Algorithm. *Translation and Computational Linguistics*, **11**(1), 22–31.

Majumder, P., Mitra, M. & Pal, D. 2007a. Hungarian and Czech Stemming using YASS. *In: Working Notes for the CLEF 2007 Workshop.*

Majumder, P., Mitra, M., Parui, S. K., Kole, G., Mitra, P. & Datta, K. 2007b. YASS: Yet another suffix stripper. *ACM Transactions on Information Systems*, **25**(4), 18:1–18:20.

Malenica, M., Šmuc, T., Šnajder, J. & Dalbelo Bašić, B. 2008. Language Morphology Offset: Text classification on a Croatian-English Parallel Corpus. *Information Processing and Management*, **41**(1), 325–339.

Manning, C. D., Raghavan, P. & Schütze, H. 2008. *Introduction to Information Retrieval.* Cambridge University.

Melucci, M. & Orio, N. 2003. A Novel Method for Stemmer Generation Based on Hidden Markov Models. *Str. 131–138: Proceedings of CIKM'2003.*

Mladenić, D. 2002. Learning Word Normalization Using Word Suffix and Context from Unlabeled Data. *Str. 427–434: Proceedings of the Nineteenth International Conference on Machine Learning, ICML 2002.*

Moreau, F., Claveau, V. & Sébillot, P. 2007. Automatic Morphological Query Expansion Using Analogy-Based Machine Learning. *Str. 222–233: ECIR.* Lecture Notes in Computer Science, vol. 4425. Springer.

Oliver, A. 2003. Use of Internet for Augmenting Coverage in a Lexical Acquisition System from Raw Corpora. *In: Workshop on Information Extraction for Slavonic and Other Central and Eastern European Languages (IESL 2003), RANLP.*

Oliver, A. & Tadić, M. 2004. Enlarging the Croatian Morphological Lexicon by Automatic Lexical Acquisition from Raw Corpora. *Str. 1259–1262: Proceedings of LREC'04.*

Paice, C. D. 1990. Another Stemmer. *ACM SIGIR Forum*, **24**, 56–61.

Paice, C. D. 1996. Method for Evaluation of Stemming Algorithms Based on Error Counting. *Journal of the American Society for Information Science*, **47**(8), 632–649.

Pecina, P. 2010. Lexical Association Measures and Collocation Extraction. *Multiword expressions: hard going or plain sailing? Journal of Language Resources and Evaluation.*

Pirkola, A. 2001. Morphological Typology of Languages for IR. *Journal of Documentation*, **57**(3), 330–348.

Plisson, J., Lavrac, N., Mladenic, D. & Erjavec, T. 2008. Ripple Down Rule learning for automated word lemmatisation. *AI Communications*, **21**(1), 15–26.

Ponte, J. M. & Croft, W. B. 1998. A language modeling approach to information retrieval. *Str. 275–281: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval.*

Popovic, M. & Willett, P. 1992. The Effectiveness of Stemming for Natural-language Access to Slovene Textual Data. *Journal of the American Society for Information Science*, **43**(5), 384–390.

Porter, M. F. 1980. An Algorithm for Suffix Stripping. *Program*, **14**(3), 130–137.

Roeck, A. D & Al-Fares, W. 2000. *A Morphologically Sensitive Clustering Algorithm for Identifying Arabic Roots.*

Sagot, B. 2005. Automatic Acquisition of a Slovak Lexicon from a Raw Corpus. *Lecture Notes in Computer Science*, **3658**, 156–163.

Saint-Dizier, P. & Viegas, E. (ured.). 1994. *Computational Lexical Semantics.* New York, NY, USA: Cambridge University Press.

Salton, G., Wong, A. & Yang, C. S. 1975. A vector space model for automatic indexing. *Commun. ACM*, **18**(11), 613–620.

Savoy, J. 1999. A Stemming Procedure and Stopword List for General French Corpora. *Journal of the American Society for Information Science*, **50**(10), 944–952.

Savoy, J. 2006. Light Stemming Approaches for the French, Portuguese, German and Hungarian languages. *Str. 1031–1035: SAC '06: Proceedings of the 2006 ACM symposium on Applied computing.* New York, NY, USA: ACM Press.

Schinke, R., Greengrass, M., Robertson, A. M. & Willett, P. 1996. A Stemming Algorithm for Latin Text Databases. *Journal of Documentation*, **52**, 172–187.

Schone, P. & Jurafsky, D. 2001. Knowledge-free Induction of Inflectional Morphologies. *Str. 1–9: Proceedings of the North American Chapter Of The Association For Computational Linguistics, NAACL 2001.*

Sebastiani, F. 2002. Machine Learning in Automated Text Categorization. *ACM Computing Surveys*, **34**(1), 1–47.

Slavić, A. 1998. Automatsko predmetno označivanje: od računalno potpomognutog predmetnog označivanja do znalačkih sustava. *Str. 98–115: Predmetna obradba: ishodišta i smjernice.* Zagreb: Hrvatsko bibliotekarsko društvo.

Spencer, A. 1991. *Morphological theory: an introduction to word structure in generative grammar.* Blackwell textbooks in linguistics. Wiley-Blackwell.

Stanković, R., Obradović, I. & Krstev, C. 2009. Proširivanje upita zasnovano na leksičkim resursima. *Str. 73–76: Zbornik radova naučno-stručnog skupa SNTPI'09.*

Stein, B. & Potthast, M. 2007. Putting Successor Variety Stemming to Work. *Str. 367–374: Advances in Data Analysis: Selected Papers from the 30th Annual Conference of the German Classification Society.* Springer.

Šilić, A., Chauchat, J.-H., Dalbelo Bašić, B. & Morin, A. 2007. N-Grams and Morphological Normalization in Text Classification: A Comparison on a Croatian-English Parallel Corpus. *Str. 671–682: Lecture Notes in Artificial Intelligence*, vol. 4874. Springer.

Šnajder, J. & Dalbelo Bašić, B. 2008. Higher-Order Functional Representation of Croatian Inflectional Morphology. *Str. 121–130: Proceedings of the 6th International Conference on Formal Approaches to South Slavic and Balkan Languages, FASSBL6.* Dubrovnik, Croatia: Croatian Language Technologies Society.

Šnajder, J. & Dalbelo Bašić, B. 2009. String Distance-Based Stemming of the Highly Inflected Croatian Language. *In: Proceedings of Recent Advances in Natural Language Processing (RANLP 2009).*

Šnajder, J. & Dalbelo Bašić, B. 2009. Lexicon-Based Morphological Normalisation and its Application to Croatian Language. *Str. 23–80:* Tadić, M., Dalbelo Bašić, B. & Moens, M.-F. (ured.), *Technologies for the Processing and Retrieval of Semi-Structured Documents: Experience from the CADIAL Project.* Zagreb: Croatian Language Technologies Society.

Šnajder, J., Dalbelo Bašić, B. & Tadić, M. 2008. Automatic Acquisition of Inflectional Lexica for Morphological Normalisation. *Information Processing and Management*, **44**(5), 1720–1731.

Tadić, M. 1992. Problemi računalne obrade imeničnih oblika u hrvatskome. *Suvremena lingvistika*, **34**, 301–308.

Tadić, M. 1994. *Računalna obrada morfologije hrvatskoga književnoga jezika, doktorska disertacija.*

Tadić, M. 2002. Building the Croatian National Corpus. *Str. 441–446: Proceedings of LREC'2002.*

Tadić, M. 2003. *Jezične tehnologije i hrvatski jezik.* Zagreb: Ex Libris.

Tadić, M. & Bekavac, B. 2006. Inflectionally Sensitive Web Search in Croatian using Croatian Lemmatization Server. *Str. 481–486:* Lužar-Stiffler, V. & Hljuz Dobrić, V. (ured.), *Proceedings of 26th International Conference on Information Technology Interfaces (ITI'06).* SRCE, Zagreb.

Tadić, M. & Fulgosi, S. 2003. Building the Croatian Morphological Lexicon. *Str. 41–46: Proceedings of EACL'2003.*

Tadić, M. 2005. The Croatian Lemmatization Server. *Southern Journal of Linguistics*, **29**(1/2), 206–217.

Thompson, S. 1999. *The Craft of Functional Programming.* Addison Wesley.

Tomlinson, S. 2003. Lexical and Algorithmic Stemming Compared for 9 European Languages with Hummingbird SearchServer at CLEF 2003. *Str. 286–300: CLEF.*

van Rijsbergen, C. J. 1979. *Informaton Retrieval.* Butterworths, London.

Weiss, D. 2005. A survey of freely available Polish stemmers and evaluation of their applicability in information retrieval. *Str. 216–223: Human Language Technologies as a Challange for Computer Science and Linguistics, Proceedings of the 2nd Language and Technology Conference.*

Xu, J. & Croft, W. B. 1998. Corpus-Based Stemming using Cooccurrence of Word Variants. *ACM Transactions on Information Systems*, **16**(1), 61–81.

Yang, Y. & Pedersen, J. O. 1997. A Comparative Study on Feature Selection in Text Categorization. *Str. 412–420:* Fisher, Douglas H. (ured.), *Proceedings of ICML-97, 14th International Conference on Machine Learning.* Nashville, US: Morgan Kaufmann Publishers, San Francisco, US.

Sažetak

Morfološka normalizacija tekstova na hrvatskome jeziku za dubinsku analizu i pretraživanje informacija

Riječi u tekstu pojavljuju se u različitim morfološkim varijantama, odnosno flektivnim i derivacijskim oblicima. Morfološka varijacija ima negativan utjecaj na djelotvornost sustava za pretraživanje informacija i dubinsku analizu teksta, naročito kod morfološki složenih jezika kao što je hrvatski. Negativne utjecaje morfološke varijacije moguće je ukloniti primjenom postupaka morfološke normalizacije, odnosno sažimanjem različitih morfoloških varijanti jedne ili više riječi na jedan reprezentativni oblik. U okviru ovog rada razvijen je postupak za flektivnu i flektivno-derivacijsku normalizaciju tekstova na hrvatskome jeziku temeljen na morfološkom leksikonu. Kako bi se zaobišao problem ograničenosti opsega leksikona te visoke cijene njegove izgradnje, razvijen je postupak za automatsku akviziciju flektivnoga leksikona iz neoznačenog korpusa. Akvizicija i normalizacija temelje se na računalnom generativno-redukcijskome modelu morfologije hrvatskoga jezika kojim je obuhvaćena fleksija i sufiksalna tvorba imenica, glagola i pridjeva. Model je inspiriran konceptima funkcijske programske paradigme, napose funkcijama višega reda kao načinu apstrakcije flektivnih i tvorbenih pravila. Provedeno je iscrpno eksperimentalno vrednovanje kojim je utvrđeno da postupak doseže visoku intrinzičnu kakvoću normalizacije, na flektivnoj razini usporedivu s onom ručno sastavljenog leksikona. Pristup opisan u ovome radu usredotočen je na hrvatski jezik, ali je primjenjiv i na druge, morfološki slične jezike.

Ključne riječi: Morfološka normalizacija, računalna obrada morfologije, obrada prirodnog jezika, hrvatski jezik, pretraživanje informacija, dubinska analiza teksta.

Abstract

Morphological Normalization of Texts in Croatian Language for Text Mining and Information Retrieval

Due to language morphology, words appear in text in various inflectional and derivational forms. Morphological variation has been shown to negatively affect the performance of most information retrieval and text mining systems, especially in the case of morphologically complex languages such as Croatian. Morphological variation may be reduced by performing morphological normalisation, i.e., the conflation of morphological variants of a word into a single representative form. This thesis describes a lexicon-based approach to morphological normalization that addresses both inflectional and derivational variation. To eliminate the problem of limited lexicon coverage and the immense effort required to compile a lexicon by hand, a procedure for acquiring automatically an inflectional morphological lexicon from raw corpora is devised. To this end, a computational model of Croatian inflectional and derivation morphology has been developed. The model, which is both generative and reductive, is inspired by the functional programming paradigm and makes use of higher-order functions to abstract inflectional and word-formation rules. Detailed experimental evaluation revealed that the developed procedure achieves high normalization quality, which at the inflectional level is comparable to the hand-crafted gold standard. Although the focus of this thesis is on Croatian language, the approach is general enough to be also applicable to other morphologically similar languages.

Keywords: Morphological normalization, computational morphology, natural language processing, Croatian language, information retrieval, text mining.

Životopis

Jan Šnajder rođen je 27. siječnja 1977. godine u Zagrebu. Diplomirao je 2002. godine na Fakultetu elektrotehnike i računarstva Sveučilišta u Zagrebu, smjer računarstvo, s temom "Izvedba MPEG-4 video algoritma". Magistrirao je 2006. godine na istome fakultetu, smjer jezgra računarstva, s temom "Arhitektura višeagentskog sustava temeljena na obojenim Petrijevim mrežama". Godine 2002. nagrađen je Rektorovom nagradom Sveučilišta u Zagrebu kao suautor studentskog rada. Od rujna iste godine zaposlen je na Zavodu za elektroniku, mikroelektroniku, računalne i inteligentne sustave Fakulteta elektrotehnike i računarstva kao znanstveni novak na projektu "Višeagentski sustavi za tumačenje dinamičkih scena", a od ožujka 2008. na projektu "Otkrivanje znanja u tekstnim podacima". Bio je uključen u nastavne aktivnosti Zavoda na predmetima Digitalna logika; Arhitektura računala; Projektiranje digitalnih sustava; Neizrazito, evolucijsko i neuroračunarstvo; Strojno učenje; Inteligentni sustavi i Umjetna inteligencija, a asistirao je u vođenju 12 diplomskih i završnih radova. Njegovi istraživački interesi obuhvaćaju područje obrade prirodnog jezika, pretraživanje informacija, dubinske analize teksta i funkcijskog programiranja. Stručno se usavršavao na trima ljetnim školima: ESSLLI'2008 i ESSLLI'2009 (European Summer School in Language, Logic and Information) te ESSIR'2009 (European Summer School in Information Retrieval). Surađivao je na četirima znanstvenoistraživačkim projektima Ministarstva znanosti, obrazovanja i športa te dvama međunarodnim projektima. U suautorstvu je objavio 14 radova na međunarodnim znanstvenim skupovima i šest radova u časopisima s međunarodnom recenzijom, od kojih tri u časopisima indeksiranima u bazama CC i SCI Expanded. Suautor je jednog poglavlja u knjizi te jedne sveučilišne skripte. Član je strukovnih udruga ACM, IEEE, ACL (Association of Computational Linguistics), FoLLI (European Association for Logic, Language and Information) i Hrvatskog društva za jezične tehnologije. Govori engleski i njemački jezik.

Biography

Jan Šnajder was born on January 27, 1977 in Zagreb, Croatia. He received his B.Sc. in computing from the University of Zagreb, Faculty of Electrical Engineering and Computing in 2002 (thesis title: "Implementation of MPEG-4 Video Algorithm") and M.Sc. in computer science from the same university in 2006 (thesis title: "Multiagent System Architecture Based on Coloured Petri Nets"). In 2002 he was awarded the University of Zagreb Rector Award as the co-author of the best student paper. From September of the same year he is employed at the Department of Electronics, Microelectronics, Computer and Intelligent Systems at the Faculty of Electrical Engineering and Computing, at first as a scientific novice on the project "Multiagent Systems for Dynamic Scene Interpeting", and from March 2008 on the project "Knowledge Discovery from Textual Data". He was involved in Department's educational activities within the courses on Digital Logic; Computer architecture; Digital System Design; Fuzzy, Evolutionar and Neurocomputing; Machine Learning; Intelligent Systems and Artificial Intelligence. He also assisted in supervising 12 bachelor's theses. His research interests include natural language processing, information retrieval, text analytics, and functional programming. He furthered his education by attending three summer schools: ESSLLI'2008 and ESSLLI'2009 (European Summer School in Language, Logic and Information), as well as ESSIR'2009 (European Summer School in Information Retrieval). He participated as a collaborator in four scientific projects funded by the Ministry of Science and two international projects. He has co-authored 14 conference papers and six journal papers, three of which in journals indexed in Current Contents and SCI Expanded. He has also co-authored one book chapter and one lecture notes booklet. He is a member of the ACM, IEEE Computer Society, ACL (Association of Computational Linguistics), FoLLI (Association for Logic, Language and Information), and the Croatian Language Technologies Society. He is fluent in English and German.