SVEUČILIŠTE U ZAGREBU

FILOZOFSKI FAKULTET

Željko Agić

PRISTUPI OVISNOSNOM PARSANJU HRVATSKIH TEKSTOVA

DOKTORSKI RAD

Zagreb, 2012.

SVEUČILIŠTE U ZAGREBU

FILOZOFSKI FAKULTET

Željko Agić

PRISTUPI OVISNOSNOM PARSANJU HRVATSKIH TEKSTOVA

DOKTORSKI RAD

Zagreb, 2012.

UNIVERSITY OF ZAGREB

FACULTY OF HUMANITIES AND SOCIAL SCIENCES

Željko Agić

APPROACHES TO DEPENDENCY PARSING OF CROATIAN TEXTS

DOCTORAL THESIS

Zagreb, 2012.

SVEUČILIŠTE U ZAGREBU

FILOZOFSKI FAKULTET

Željko Agić

PRISTUPI OVISNOSNOM PARSANJU HRVATSKIH TEKSTOVA

DOKTORSKI RAD

Mentori:

dr. sc. Zdravko Dovedan Han, red. prof.

dr. sc. Marko Tadić, red. prof.

Zagreb, 2012.

UNIVERSITY OF ZAGREB

FACULTY OF HUMANITIES AND SOCIAL SCIENCES

Željko Agić

APPROACHES TO DEPENDENCY PARSING OF CROATIAN TEXTS

DOCTORAL THESIS

Supervisors:

dr. sc. Zdravko Dovedan Han, red. prof.

dr. sc. Marko Tadić, red. prof.

Zagreb, 2012.

Zahvala

Zahvaljujem voditeljima, profesoru Zdravku Dovedanu Hanu i profesoru Marku Tadiću, što su me upoznali s područjem jezičnih tehnologija i dali mi unutar njega slobodu, poticaj i podršku da slijedim vlastitu znanstvenu znatiželju.

Kolegama s Odsjeka za informacijske i komunikacijske znanosti, Odsjeka za lingvistiku i Zavoda za lingvistiku Filozofskoga fakulteta te suradnicima iz Laboratorija za tehnologije znanja Fakulteta elektrotehnike i računarstva zahvaljujem na dosadašnjoj suradnji i brojnim razmjenama ideja. Posebno hvala Boži Bekavcu, Nikoli Ljubešiću i Kristini Vučković. Hvala Daši Berović, Teni Gnjatović, Idi Raffaelli, Krešimiru Šojatu i brojnim studentima lingvistike pri Filozofskom fakultetu na dosadašnjem razvoju Hrvatske ovisnosne banke stabala. Hvala kolegici Nives Mikelić-Preradović na CROVALLEX-u. Bez tih jezičnih resurasa, ovdje prikazano istraživanje bilo bi nemoguće izvesti. Hvala Joakimu Nivreu sa Sveučilišta u Uppsali na raspravama o hrvatskom jeziku i pristupima ovisnosnomu parsanju. Osobito sam zahvalan Danijeli Merkler, koja je gotovo dvaput pročitala ranije inačice rada i pobrinula se da jezik kojim je on pisan zaista bude hrvatski.

Zahvaljujem najbližima na velikom strpljenju i razumijevanju. Posebno zahvaljujem roditeljima, koji su me nagovorili na pune dvadeset i dvije godine formalnoga obrazovanja, a potom mi uz puno samoodricanja svakoga njihovog dana pružali sve oblike potpore dosezanju konačnoga cilja. Nadate se, vjerujem, da je ovim radom on dosegnut.

Zagreb, srpanj 2012.

Željko Agić

Sadržaj

Popis algoritama

Popis primjera

Popis slika

Popis tablica

1 Uvod

Mogućnost razumijevanja prirodnoga jezika i sporazumijevanja prirodnim jezikom u pisanome i govorenome obliku jedna je od osnovnih značajki ljudske inteligencije. Ljudsko razumijevanje prirodnoga jezika promatraju s različitih gledišta znanstvena područja jezikoslovlja i računalne znanosti, odnosno područja umjetne inteligencije te računalnoga jezikoslovlja i strojne obradbe prirodnoga jezika. Načelno se raznorodna gledišta prema pojavi ljudskoga razumijevanja jezika koja pripadaju pojedinim područjima mogu razdijeliti dvočlanom razredbom. U toj razredbi prvo gledište je ono kod kojega se za osnovni zadatak postavlja pronalaženje znanstvenih spoznaja o načinima i utjecajima na ljudsko razumijevanje jezika. Kod drugoga gledišta nastoji se matematički i potom računalno modelirati postupak razumijevanja jezika svojstven ljudima, prvenstveno s ciljem izrade što učinkovitijih strojnih, odnosno umjetno- ili računalno-inteligentnih postupaka obradbe prirodnoga jezika. Primjetno je pritom kako gledišta iz postavljene jednostavne razredbe nisu odvojena budući da spoznaje usvojene nekim odabranim gledištem na pojavu razumijevanja jezika mogu i povijesno jesu značajno utjecale na usvajanje novih spoznaja, neovisno o različitome odabranom gledištu. Pojava razumijevanja jezika promatra se najčešće, prema načelima jezikoslovnoga pristupa, po razinama jezičnoga opisa, pa se istražuje i modelira ljudsko razumijevanje jezika, između ostalih, na morfološkoj, sintaktičkoj i semantičkoj razini. Sve razine jezikoslovnoga opisa i razumijevanja prirodnoga jezika prožete su pojavom jezične višeznačnosti, odnosno mogućnošću višestrukoga tumačenja pojedinih jezičnih elemenata. Osnovni problem pri razmatranju strojnoga modeliranja postupka razumijevanja prirodnoga jezika time posljedično postaje problem razrješivanja jezične višeznačnosti, odnosno objašnjavanje i modeliranje postupka usvajanja obavijesti iz višeznačne jezične poruke. Na sintaktičkoj razini, odnosno razini uspostavljanja odnosa među riječima i spojevima riječi kao elementima rečeničnoga ustroja, razrješivanje mogućih višestrukih tumačenja pojedinih rečeničnih elemenata – poput riječi ili spojeva riječi koji predstavljaju rečenične predikate, subjekte i objekte – od osobitoga je značaja za razumijevanje rečenice kao obavijesti.

U ovome istraživanju razmatraju se pristupi modeliranju i računalnoj izvedbi postupaka strojne obradbe rečenica hrvatskoga jezika na sintaktičkoj razini jezičnoga opisa, odnosno pristupi jednoznačnomu strojnom otkrivanju uloga riječi i spojeva riječi u rečeničnome ustroju za tekstove hrvatskoga jezika. Taj se postupak općenito naziva sintaktička analiza ili parsanje, pa se u ovome istraživanju govori o parsanju tekstova hrvatskoga jezika. Parsanje je

dobro istražen problem u području računalne znanosti, unutar kojega se razmatra sintaktička struktura jezika koji su eksplicitno definirani nekim formalnim modelom. Ti se jezici nazivaju formalnim jezicima i opisani su formalnim gramatikama. Izravna primjena postojećih znanja o parsanju formalnih jezika na tekstove prirodnoga jezika netrivijalna je zbog složenosti opisa prirodnoga jezika jednim formalnim modelom i posljedične prevelike složenosti algoritamske obradbe tekstova primjenom formalnoga modela s gledišta točnosti i učinkovitosti te zbog nemogućnosti razrješivanja sintaktičke višeznačnosti unutar toga teorijskog okvira. Stoga se pristupi parsanju tekstova prirodnoga jezika najčešće oslanjaju na približno modeliranje sintaktičkih pojava i na razrješivanje sintaktičke višeznačnosti sustavnom primjenom skupova pravila za razrješavanje višeznačnosti izvedenih iz nekoga uzorka opažanja valjanih ljudskih razrješivanja višeznačnosti. Načelno se razlikuje pristup parsanju temeljen na ručno zapisanim pravilima i pristup temeljen na podatcima. Potonji pristup u proteklih nekoliko godina bilježio je značajne pomake u opaženoj točnosti i učinkovitosti parsanja tekstova velikoga broja raznorodnih prirodnih jezika, pa se stoga u ovome istraživanju promatra upravo taj pristup. Kod pristupa parsanju tekstova prirodnoga jezika temeljenih na podatcima računalni model sintaktičke strukture toga jezika usvaja se statističkim izvođenjem iz poželjno velikoga skupa rečenica nad kojima su stručnjaci prethodno proveli sintaktičku analizu u skladu s prethodno zadanim i također stručno odabranim formalizmom za sintaktički opis. Budući da je većina sintaktičkih formalizama definirana tako da nad rečenicama gradi strukture koje se prema računalnoj znanosti nazivaju stablima, skup rečenica kojima su ovim postupkom dodijeljena sintaktička stabla naziva se najčešće bankom stabala. S obzirom na odabranu sintaktičku teoriju, odnosno formalizam sintaktičkoga opisa ugrađenoga u banku stabala, razlikuju se u pojednostavljenoj razredbi banke fraznih stabala – one koje sadrže stabla izgrađena sustavnom primjenom sintaktičke teorije zasnovane na fraznoj rečeničnoj strukturi – i banke ovisnosnih stabala – one koje sadrže ovisnosna stabla, odnosno sintaktička stabla izgrađena sustavnom primjenom sintaktičke teorije zasnovane na postavljanju riječi u ovisnosne odnose unutar rečenica. Ovisnosne teorije sintakse osobito su pogodne za opisivanje sintaktičke strukture jezika s većim stupnjem slobode s obzirom na redoslijed riječi u elementima rečeničnoga ustroja, pa se ovo istraživanje usmjerava parsanju tekstova hrvatskoga jezika temeljenom na ovisnosnoj sintaksi, odnosno ovisnosnome parsanju temeljenom na podatcima. Pritom se za izgradnju modela statističkim izvođenjem odabire Hrvatska ovisnosna banka stabala kao jedini dostupni uzorak valjanih razrješivanja sintaktičke višeznačnosti u hrvatskim tekstovima u trenutku provođenja ovoga istraživanja. Statistički pristupi, odnosno pristupi ovisnosnomu parsanju temeljeni na podatcima načelno se dijele u dvije skupine: na pristupe

temeljene na spoznajama iz teorije grafova i pristupe temeljene na nekim postavkama preuzetima iz teorije formalnih jezika, odnosno na apstraktnome modelu sustava temeljenoga na prijelazima ili prijelazničkoga sustava. Ove pristupe ovisnosnomu parsanju temeljene na podatcima načelno je moguće dopuniti povezivanjem s jezičnim resursima ili alatima koji su dostupni za ciljani prirodni jezik i smisleni s gledišta njegove sintaktičke analize, i to s ciljem postizanja veće točnosti i učinkovitosti pri parsanju tekstova toga jezika. Pristupi povezivanju metoda parsanja temeljenih na podatcima – koje se smatraju jezično neovisnima budući da se izgrađuju iz banaka ovisnosnih stabala – s jezičnim resursima i alatima razvijenima specifično za opisivanje jednoga prirodnog jezika većim su dijelom neistraženi, a posebno u usporedbi s općim pristupima temeljenima na podatcima.

Dva su osnovna zadatka ovoga istraživanja. Prvi je zadatak proučiti primjenjivost nekih znanih pristupa ovisnosnomu parsanju temeljenih na podatcima – s naglaskom na pristupe dokazane na zadatku parsanja srodnih jezika – pri parsanju tekstova hrvatskoga jezika. Drugi zadatak je istražiti neke mogućnosti povezivanja tih pristupa ovisnosnomu parsanju s jezičnim resursima i alatima dostupnima za obradbu hrvatskih tekstova s ciljem dosezanja veće razine točnosti ovisnosnoga parsanja hrvatskih tekstova u usporedbi s općim metodama. Za okvir ostvarenja postavljenih ciljeva dana je Hrvatska ovisnosna banka stabala kao korpus tekstova hrvatskoga jezika razdvojenih na rečenice i pojavnice i označenih na morfosintaktičkoj i sintaktičkoj razini u skladu s odabranim ovisnosno-sintaktičkim formalizmom. U istraživanju se iz pregleda pristupa parsanju formalnoga jezika izvodi formalna definicija parsanja i ovisnosnoga parsanja te formalni modeli parsera s obzirom na sintaktički formalizam i banku ovisnosnih stabala, predstavljaju se pristupi ovisnosnomu parsanju temeljeni na podatcima i uvodi okvir za formalno vrjednovanje značajki točnosti i učinkovitosti ovisnosnoga parsanja s pomoću banke ovisnosnih stabala. Uporabom Hrvatske banke ovisnosnih stabala vrjednuju se različite izvedbe modela parsera temeljenih na teoriji grafova i prijelazničkih parsera. Predlaže se i razvija izvorni model ovisnosnoga parsanja hrvatskih tekstova, temeljen na teoriji grafova i povezivanju s valencijskim rječnikom hrvatskih glagola CROVALLEX. Taj se prototipni model, izveden u obliku prototipnoga računalnog sustava radnoga naziva CroDep, vrjednuje na Hrvatskoj ovisnosnoj banci stabala. Do provođenja ovdje prikazanoga istraživanja nisu zabilježeni eksperimenti s ovisnosnim parsanjem hrvatskih tekstova temeljenim na podatcima i na povezivanju s dostupnim jezičnim resursima i/li alatima za obradbu hrvatskih tekstova.

Opis istraživanja u ovome je tekstu oblikovan četirima poglavljima. U uvodnome se poglavlju ocrtava problematika ovisnosnoga parsanja, odabrani pristupi ovisnosnomu parsanju hrvatskih tekstova i pristupi njegovu vrjednovanju. U drugome poglavlju izložen je teorijski okvir za ovisnosno parsanje tekstova prirodnoga jezika. To izlaganje uključuje opću definiciju parsanja prirodnoga jezika i izvođenje formalne definicije problema ovisnosnoga parsanja preko povijesnoga pregleda općih pristupa parsanju formalnoga jezika. U njemu se definira parsanje teksta u opreci s parsanjem gramatikom, postavljaju tražena svojstva i opći kriteriji za vrjednovanje parsera teksta te se formalno definiraju pristupi ovisnosnomu parsanju temeljeni na podatcima, odnosno na teoriji grafova i na prijelazničkim sustavima. Treće poglavlje predstavlja prikaz dvaju eksperimenata s ovisnosnim parsanjem hrvatskih tekstova iz Hrvatske ovisnosne banke stabala. Ono započinje prikazom osnovnih značajaka te banke ovisnosnih stabala s gledišta ovisnosnoga parsanja temeljenoga na podatcima i dosadašnjih i budućih pristupa njezinoj izgradnji. Slijedi prikaz postavki i rezultata prvoga eksperimenta s ovisnosnim parsanjem hrvatskih tekstova korištenjem dvaju sustava za izgradnju ovisnosnih parsera temeljenih na grafovima i prijelazničkih ovisnosnih parsera. Potom se prikazuje računalni model i izvedba prototipnoga izvornog sustava za ovisnosno parsanje hrvatskih tekstova temeljenoga na teoriji grafova i povezivanju s valencijskim rječnikom hrvatskih glagola CROVALLEX. Model se vrjednuje u drugome prikazanom eksperimentu usporedbom s modelima vrjednovanima u prethodnome eksperimentu prema tamo definiranim postavkama eksperimenta. Četvrto pogavlje sadrži zaključna razmatranja o provedenim eksperimentima i neke smjernice za buduća istraživanja ovisnosnoga parsanja hrvatskih tekstova.

2 Ovisnosno parsanje prirodnoga jezika

U ovome se poglavlju predstavlja i formalizira problem ovisnosnog parsanja tekstova pisanih prirodnim jezikom te predstavljaju neki pristupi rješavanju toga problema. Najprije se definira parsanje s gledišta formalnoga i prirodnoga jezika te s gledišta lingvistike i računalne znanosti. Potom se kratko ocrtava potreba za parsanjem i njegova korisnost u području obradbe prirodnoga jezika, crpljenju obavijesti i jezičnim tehnologijama općenito. U kratkoj raspravi o pristupima sintaktičkoj analizi, uvodi se i matematički formalizira problem ovisnosnoga parsanja te se pojašnjava njegova složenost s gledišta osmišljavanja računalno izvedivog modela parsera i njegove izvedbe. Predstavljaju se suvremeni pristupi rješavanju problema ovisnosnoga parsanja prirodnoga jezika, s naglaskom na jezično-neovisne pristupe, temeljene na podatcima, odnosno implicitnom zadavanju gramatičkoga formalizma parseru putem sintaktički obilježenih računalnih korpusa – banaka stabala. Detaljno se pojašnjavaju jezično-neovisni, na podatcima temeljeni pristupi, oni zasnovani na postavkama teorije grafova i oni zasnovani na formalnim automatima, odnosno tablicama prijelaza. Dan je također i kratak pregled nekih od preostalih pristupa, vrijednih spomena s gledišta najnovijih saznanja o točnosti ovisnosnoga parsanja u tim teorijskim okvirima. Vezano uz to, u ovome se poglavlju također raspravlja i o vrjednovanju ovisnosnoga parsanja s gledišta robustnosti, razrješivanja sintaktičke višeznačnosti, točnosti parsanja i računalne učinkovitosti parsera, a također se kratko prikazuju, razvrstani po paradigmi parsanja, rezultati trenutno najboljih ovisnosnih parsera pri primjeni na različitim jezicima. Poglavlje završava raspravom o korisnosti i primjenjivosti ovisnosnih parsera u području obradbe prirodnoga jezika, crpljenju obavijesti i jezičnim tehnologijama općenito.

2.1 Definicija ovisnosnoga parsanja

Ovdje se raspravlja o ovisnosnome parsanju kao optimizacijskom problemu. Definira se parsanje općenito, postavlja razlika među pristupima parsanju s obzirom na gramatički formalizam te obrazlaže odabir ovisnosnoga pristupa u ovome specifičnom istraživanju. Veći dio potpoglavlja bavi se matematičkom formalizacijom problema ovisnosnoga parsanja, s naglaskom na gledište računalne znanosti, kao pripremom za formalizaciju pristupa rješavanju toga problema.

2.1.1 Što je parsanje?

Naziv *parsanje* (eng. *parsing*) je pojednostavljeni, pa stoga i u nekoj mjeri kolokvijalni naziv koji se uobičajeno koristi u računalnoj znanosti i lingvistici umjesto formalnijega i znatno preciznijega naziva *sintaktička* (ili *sintaksna*) *analiza*. Parsanje je, proizlazi iz toga, zapravo sintaktička analiza. Analogijom bi se moglo pogrešno ustvrditi da je sintaktička analiza, nadalje, zapravo analiza sintakse, što je samo djelomično točno. Stoga je ovdje potrebno, prije svake konkretnije rasprave o parsanju, pojasniti što je – u kontekstu ovoga rada – sintaksa, a što sintaktička analiza.

2.1.1.1 Definicije parsanja

Dvije se definicije sintakse mogu izvesti iz dostupne literature, a upućuju na dva različita gledišta prema tome pojmu.

Jedna definicija, karakteristična za gramatike i gramatičare, odnosno jezikoslovce koji se bave proučavanjem konkretnih jezika i njihovih pravilnosti po razinama lingvističkoga opisa, najčešće kaže da je sintaksa (grč. *sýntaxis*: red, slaganje, razmještaj, itd.) dio gramatike koji "opisuje rečenično ustrojstvo" i postavlja "pravila o slaganju riječi u rečenice" (Barić i dr. 2003:391) te proučava odnose među "riječima i njihovim oblicima", "spojevima riječi" te među "surečenicama u složenoj rečenici" (Silić i Pranjković 2005:183). Uz sintaksu kao dio gramatike kojim se opisuje rečenični ustroj, odnosno pravila o slaganju rečenica iz riječi – te time kao donju granicu sintakse postavlja riječ, a kao gornju postavlja rečenicu (usp. Barić i dr. 2003) – spominje se i pojam sintakse teksta, odnosno proučavanja odnosa među rečenicama u većim jezičnim cjelinama, čime se prekoračuje ranije postavljena gornja granica proučavanja. Uobičajeno se pod sintaksom u ovoj definiciji misli na sintaksu rečenice, koja je i u presjeku s predmetom ovoga istraživanja, pa se dalje u tekstu više ne osvrće na sintaksu teksta, već samo na sintaksu rečenice. Također, sintaksa rečenice može se promatrati, načelno, na dva razdvojena načina: s obzirom na sintagmatske i s obzirom na paradigmatske odnose među jezičnim jedinicama unutar rečenice (Silić i Pranjković 2005:183). Proučavati sintagmatske odnose pritom znači baviti se isključivo pravilnostima razmještaja jezičnih jedinica bez promatranja njihova obavijesnog gledišta, dok proučavanje paradigmatskih odnosa uključuje i obavijesnu komponentu te se pita o implikacijama uporabe svih mogućih jezičnih jedinica na određenim mjestima u rečenici s obzirom na obavijesni sadržaj te

rečenice. Kao i ranije, ishodište ovoga istraživanja u presjeku je sa sintagmatskim proučavanjem rečenične sintakse.

Druga definicija sintakse odnosi se na skup svih pravila iz prethodne definicije kojima se opisuje rečenično ustrojstvo pojedinoga jezika, pa se kaže, primjerice, "sintaksa hrvatskoga jezika". U toj definiciji, sintaksa ne predstavlja dio gramatike, disciplinu ili razinu lingvističkoga opisa, već unaprijed izvedeni skup pravila kojima je opisano ustrojstvo nekoga jezika, primjerice hrvatskoga.

Sintaksa je, dakle, ili disciplina koja proučava zakonitosti kojima se iz riječi stvaraju rečenice prirodnoga jezika ili skup pravila koji je nastao ranijim proučavanjem i prema kojemu se za neki zadani jezik iz riječi toga jezika stvaraju valjane rečenice toga jezika. Mogao bi se stoga taj termin upotrijebiti[1] na neki od sljedećih načina: (1) *sintaksa* (kao jedna od razina jezičnoga opisa, odnosno proučavanje zakonitosti stvaranja valjanih rečenica), (2) *hrvatska sintaksa* ili *sintaksa hrvatskoga jezika* (kao skup pravila proizvedenih analizom iz prethodne točke), (3) *jedna sintaksa hrvatskoga jezika* (kao jedna od mogućih prezentacija prethodno izvedenih pravila) i (4) *sintaksa jedne rečenice hrvatskoga jezika* (kao jedna instanca pravila definiranih u prethodne dvije točke). Moguće je također precizno definirati sintaksu s gledišta teorije formalnih jezika, odnosno računalne znanosti (usp. Aho i Ullman 1972, Aho i dr. 2006, Slonneger i Kurtz 1995), no ta je definicija ovdje izostavljena jer se trivijalno implicira prethodnima; dovoljno je zamijeniti u prethodnim definicijama riječ *jezik* terminom *formalni jezik* ili *programski jezik*. Prethodne su definicije utoliko potpunije jer i formalni i programski jezici, kao i prirodni jezici, pripadaju na dovoljno visokoj razini apstrakcije istomu skupu. Također, presjek teorije formalnih jezika s obradbom prirodnih jezika – s isključivim naglaskom na razinu sintakse – dan je u idućem potpoglavlju, pa je ovdje izostavljen.

Iz danih definicija sintakse mogu se sada izvesti definicije sintaktičke analize (i analize sintakse), općenito i za potrebe ovoga istraživanja. Analizirati sintaksu, s gledišta četiri prethodno ocrtane uporabe termina *sintaksa*, moglo bi značiti: (1) proučavati kako se proučavaju zakonitosti stvaranja valjanih rečenica u jezicima, (2) proučavati skupove pravila koji objašnjavaju jezične fenomene kojima se iz riječi stvaraju rečenice jezika, (3) proučavati jednu od izvedbi tih skupova pravila ili (4) raščlaniti jednu ili više rečenica nekoga jezika kako bi se otkrilo koja su ih od svih mogućih sintaktičkih pravila danih za taj jezik stvorila,

[1] http://dictionary.reference.com/browse/syntax (2012-01-19)

odnosno koja od njih su primjenjiva za dane rečenice. S druge strane, *sintaktička analiza* je kao izraz olakšavajuće ograničena jer implicira samo jednu od četiri predložene mogućnosti: *sintaktički analizirati* znači provesti analizu s gledišta sintakse. Pritom se implicira sintaktička analiza rečenice ili skupa rečenica koji predstavlja tekst pisan nekim jezikom. (Ovaj se rad ograničava na sintaktičku analizu pisanoga teksta, a većina literature (usp. Nivre 2006:12) također termine *parsanje* i *sintaktička analiza* ograničava na pisani jezik, dok se za govoreni najčešće koristi drugačija terminologija.) Iz dane argumentacije proizlazi da je za raščlambu rečenica nekoga jezika na razini sintakse opravdanija uporaba termina *sintaktička* (ili *sintaksna*) *analiza*, nego termina *analiza sintakse*.

Preostaje još pojasniti zašto se, unatoč preciznosti i prethodno ilustriranoj obrazloživosti termina, sintaktička analiza svejedno najčešće[2] naziva parsanjem. Naziv *parsanje* (Vučković 2009) preuzet je iz engleskog jezika (en. *to parse*), a oba posljedično crpe iz latinskoga[3] *pars orationis*, što u doslovnome prijevodu znači *vrsta riječi* (Nivre 2006:12). Naime, sintaktička analiza, posebno ona uobičajena na ranijim razinama podučavanja jezičnoga opisa, obično započinje analizom uloge pojedinih riječi u rečenici s gledišta njihove leksičke višeznačnosti. Sintaktička raščlamba rečenice u tome okviru započinje prepoznavanjem vrsta riječi koje je sačinjavaju, a one potom dalje impliciraju određene sintaktičke uloge. Stoga je valjano započeti sintaktičku analizu rečenice pitajući se "Quae pars orationis?", odnosno "Koje je vrste ta riječ?". Osim tradicijski, uporaba termina *parsanje* kao zamjene za *sintaktičku analizu* opravdana je i budući da latinski *pars* (en. *part*) znači *dio*, a sintaktička analiza implicitno rastavlja rečenicu na dijelove, do razine riječi, promatrajući potom sintaktičke uloge (ili funkcije) tih dijelova. U literaturi se – najčešće onoj s područja računalne znanosti (usp. Srbljić 2000) – može pronaći[4] i termin *parsiranje*, koji je jednakovrijedan terminu *parsanje*, no ovdje se prednost daje potonjemu, budući da se preferira izravno crpljenje iz latinskoga u odnosu na posredno crpljenje iz njemačkoga (s obzirom na sufiks *–irati* od njem. *–ieren*; usp. Skok 1955).

Ovdje se usvaja da je sintaktička analiza ili parsanje teksta pisanoga nekim (prirodnim) jezikom – u užemu smislu, pogodnom za potrebe ovoga teksta i istraživanja – raščlamba

[2] Unatoč nepreciznosti takve argumentacije, ilustracije radi, dovoljno je pogledati, primjerice, razliku u redovima veličina za upite "syntactic analysis" i "parsing" na nekim tražilicama (http://www.google.hr, 2012-01-19) ili bibliografskim indeksima objavljenih radova iz lingvistike, računalne znanosti te njihovih presjeka u vidu obradbe prirodnoga jezika, računalne lingvistike, jezičnih tehnologija, itd.
[3] http://dictionary.reference.com/browse/parse (2012-01-19)
[4] Iako neki ipak inzistiraju na formalnijem terminu sintaksna analiza (usp. Dovedan 2003).

rečenica toga teksta od razine rečenice do razine riječi, u skladu s nekim prethodno zadanim okvirom za sintaktički opis toga jezika, odnosno u skladu s nekim sintaksnim formalizmom. Nadalje, fokus je ovoga istraživanja na različitim metodama automatske, odnosno strojne sintaktičke analize prirodnoga jezika i njihovoj primjeni na tekstove pisane hrvatskim jezikom. Utoliko je bitno napomenuti kako se, u skladu s prethodno ustanovljenim nazivljem, računalni algoritam (i njegova posljedična računalna izvedba) koji izvodi automatsku sintaktičku analizu teksta u skladu s nekim sintaksnim, odnosno gramatičkim formalizmom, naziva *parser*. Dalje u tekstu, termini *parsanje* i *parser* koristit će se stoga isključivo da označe automatsku, odnosno strojnu sintaktičku analizu i njenu pojavnost u obliku računalnoga algoritma izvedivoga na digitalnome računalu. U odnosu na ručnu sintaktičku analizu, za koju se pretpostavlja apsolutna točnost, parsanje računalom mora se promatrati i s nekoliko različitih gledišta koja razmatraju točnost i učinkovitost parsera. U idućim potpoglavljima raspravlja se detaljnije o sintaktičkim formalizmima, različitim pristupima parsanju u njihovim okvirima i različitim pristupima vrjednovanju parsera.

2.1.1.2 Elementi rečeničnog ustroja

Prije formalizacije problema parsanja razmatranjem različitih sintaktičkih formalizama i pripadajućih algoritama za sintaktičku analizu, odnosno parsera, potrebno je – u svrhu dodatnoga ilustrativnog pojašnjenja toga problema i predstavljanja dijela terminologije koja će se učestalo koristiti dalje u tekstu – kratko razmotriti proces ručnoga parsanja, od ranije spomenutoga pitanja latinskih učenjaka "Quae pars orationis?" nadalje. Postavlja se, dakle, pitanje: Što čovjek radi kad parsa, odnosno što želi doznati kad provodi sintaktičku analizu neke rečenice prirodnoga jezika?

Gramatike hrvatskoga jezika (Barić i dr. 2003:396, Silić i Pranjković 2005:284) kažu da, u skladu s ovdje danim definicijama, sintaktički analizirati rečenicu znači koristiti gramatičko svojstvo "članjivosti rečenice", koje kaže da se rečenica može članiti na "dijelove koji su međusobno u određenim gramatičkim odnosima". Skup odnosa među pronađenim dijelovima pritom naziva "gramatičkim ustrojstvom rečenice", a sami su dijelovi "članovi rečeničnoga ustrojstva". Ručna sintaktička analiza rečenice, odnosno parsanje od strane čovjeka (stručnjaka), cilja odrediti gramatičko ustrojstvo rečenice. Kao osnovni elementi gramatičkoga ustrojstva rečenice pritom se navode (Barić i dr. 2003:398, Silić i Pranjković 2005:279) predikat, subjekt, objekt i priložna oznaka. Kaže se da su ti elementi "samostalni" jer mogu "stajati samostalno u rečenici", za razliku od nesamostalnih elemenata (Silić i

Pranjković 2005:308), atributa i apozicije, koji "pobliže značenjski određuju" samostalne elemente te se "u rečenično ustrojstvo ne uvrštavaju izravno, nego preko temeljnih članova". Samostalni se elementi gramatičkoga ustrojstva definiraju na sljedeći način.

1. Predikat je (Silić i Pranjković 2005:289) "član rečeničnoga ustrojstva koji nije ovisan o drugim članovima", odnosno (Barić i dr. 2003:400) "riječ u rečenici koja sama sebi otvara mjesto". Jednostavnije, kada se kaže da predikat sam sebi otvara mjesto, misli se na činjenicu da je predikat riječ ili izraz koji za svoju rečenicu svjesno odabire pisac teksta, ovisno o "naravi obavijesti koja se želi prenijeti rečenicom". Kao takav, predikat posljedično nije ovisan o drugim članovima rečeničnoga ustroja te njima, svojim svojstvima, u rečenici otvara mjesto i implicitno im zadaje specifične zahtjeve. Predikat je stoga u rečenici nemoguće predvidjeti jer je ograničen samo morfologijom riječi koje ga sačinjavaju i ovisi samo o (Barić i dr. 2003:400) "tome što se u kojoj prigodi želi komu reći"; on "otvara mjesto ostalim riječima u rečenici, onima koje se po njemu u rečenicu uvrštavaju". Kaže se još i da su ostali članovi rečeničnoga ustrojstva ovisni o predikatu; kasnije se u ovome radu preciznije pojašnjava bitnost te tvrdnje s gledišta sintaktičkih formalizama i posljedičnih pristupa parsanju.

   Predikat ima gramatička svojstva po kojima upravlja otvaranjem mjesta za druge članove rečeničnoga ustrojstva, odnosno po kojima ih otvarajući ujedno ograničava, a koje uvodi tvorac rečenice s ciljem prenošenja upravo željene obavijesti. Navodi se da su ta gramatička svojstva, koja se još nazivaju i predikatnim kategorijama:

   a. lice, vrijeme, način i vid u (Barić i dr. 2003:400) ili
   b. lice, broj, vrijeme, način, vid i prijelaznost u (Silić i Pranjković 2005:286).

   Ovdje se ne raspravlja pobliže o razlikama u izboru skupa predikatnih kategorija među pojedinim izvorima i odnosa morfologije i sintakse koji ih definiraju, već se samo ilustrativno predstavljaju. Preko predikatnih kategorija ostvaruju se poveznice s drugim članovima rečeničnoga ustrojstva, odnosno, one predstavljaju mehanizam preko kojega predikat na određene načine otvara njima određena mjesta u rečenici. Ovdje je još bitno spomenuti da se predikat izriče s pomoću jedne ili više predikatnih riječi, pa se prema tome razlikuje glagolski od imenskoga predikata. Izricanje predikata jedna je od pojavnosti sučelja između morfologije i sintakse

jezika, odnosno mjesto na kojem sintaktički opis poseže za morfološkim opisom[5] budući da se tu prvi put ranija implikacija povezanosti predikata kao sintaktičke i glagola kao morfološke kategorije putem predikatnih kategorija eksplicitno navodi. Kaže se da je predikat glagolski (Barić i dr. 2003:401) ukoliko je izrečen predikatnim riječima koje predstavljaju glagol u jednostavnom ili složenom glagolskom vremenu, a imenski ukoliko je izrečen pomoćnim glagolom i nekom imenskom predikatnom riječju. Gramatička razrada predikata složenija je od onoga predstavljenog ovdje, pa valja i dalje imati na umu ilustrativnu svrhu ovoga kratkog predstavljanja.

2. Subjekt se određuje (Silić i Pranjković 2005:294) kao predmet predikata, "pokretač radnje označene predikatom" ili "vršitelj radnje". Subjekt je najčešće imenska riječ[6], pa mu, kao rečeničnomu članu, time pripadaju gramatičke kategorije svojstvene imenicama: rod, broj i padež. Budući da u rečenici predikat otvara mjesto subjektu, ako je subjekt ostvaren, oni se moraju slagati u licu i broju. Na mjestu subjekta mogu, osim imenica, stajati i druge imenske riječi koje imaju određene implikacije na sročnost subjekta s predikatom, a također se subjekt ne mora nužno ostvariti u rečenici, no nijedno nije predmet proučavanja u ovome radu.
3. Objekt je (Silić i Pranjković 2005:299) "predmet koji je zahvaćen glagolskom radnjom", odnosno "predmet u vezi s kojim se ta radnja vrši". Etimologija (od lat. *obiectum*, u jednom značenju, *izložen* (Anić 2004)) mu implicira izloženost glagolskoj radnji koju pokreće subjekt; objekt se stoga pojednostavljeno može definirati kao trpitelj radnje koju vrši subjekt. Kao i u slučaju subjekta, mjesto objektu u rečenici također otvara i njime upravlja predikat, a on sam posjeduje, ponovno poput subjekta, imenične gramatičke kategorije. Objekti se, između ostaloga, razlikuju po izravnosti (izravni i neizravni) te po stupnju obaveznosti, a obje su kategorije također određene izborom predikata.
4. Priložna oznaka izriče (Barić i dr. 2003:428) "okolnosti u kojima se zbiva predikatna radnja", odnosno, obavijesno usmjerenije, (Silić i Pranjković 2005:304) "različite okolnosti događanja o kojima se priopćuje rečenicom". Priložnu oznaku u

[5] Navedene gramatike hrvatskoga jezika (Barić i dr. 2003, Silić i Pranjković 2005) izbjegavaju izravno povezati glagol kao vrstu riječi, odnosno morfološku kategoriju, s predikatom kao sintaktičkom kategorijom, vjerojatno kako bi se izbjegla moguća zabuna, odnosno nevaljano jednačenje predikata i glagola i posljedično prekoračenje razina lingvističkoga opisa, prije svega zbog udžbeničke naravi tih tekstova.

[6] Zanimljivo je primijetiti da se opažanje iz prethodne bilješke ne odnosi na povezanost imenice kao vrste riječi i subjekta kao elementa rečeničnoga ustroja, odnosno sintaktičke kategorije. Vjerojatno se pri ranijem opažanju radi o opreznosti autora gramatika s obzirom na važnost razlikovanja glagolskih od imenskih predikata, koja pri objašnjavanju subjekta ipak nije bila nužna.

rečenicu uvodi predikat, a navodi se u (Silić i Pranjković 2005) da je nužno vezan uz glagolski predikat, dok (Barić i dr. 2003) izričito navodi da je mjesto priložnoj oznaci otvoreno "samom prisutnošću predikata, bez obzira na riječ i oblik riječi kojom je predikat izrečen". Ova i druge primijećene nepodudarnosti u citiranim tekstovima, iako zabilježene, nisu predmet ovoga istraživanja[7]. Navodi se nekoliko vrsta priložnih oznaka uz naznaku o njihovoj učestalosti (priložna oznaka vremena, mjesta, načina, pa potom uzroka, namjere, itd.), grupiranih prema "razlučivanju njihova obavijesnog sadržaja", bez utjecaja na "sintaktičke odnose u rečeničnom ustrojstvu".

Nesamostalni elementi rečeničnoga ustroja – atribut i apozicija – opisani su kao "članovi rečeničnih članova" (Silić i Pranjković 2005:308) koji se u rečenicu "ne uvrštavaju izravno, nego preko temeljnih članova". Atribut i apozicija vezuju se uz imenice da ih pobliže označe te mogu biti u rečenici uvršteni uz svaku imenicu, neovisno o tome kojem članu rečeničnoga ustrojstva ona pripada. Može se reći da su atributi i apozicije ovisni o imenicama preko kojih su uvršteni u rečenicu. Razlikuje se sročni od nesročnoga atributa: prvi je izražen pridjevom koji je sročan, tj. slaže se u rodu, broju i padežu s imenicom koju pobliže označuje, a drugi nije ograničen na pridjeve i nije sročan s imenicom, već je s njome u odnosu pridruživanja (Silić i Pranjković 2003:309). Apozicija je uvijek imenica, u pravilu sročna s imenicom koja je uvrštava te se time razlikuje od obje vrste atributa.

Prethodnim definicijama samostalnih i nesamostalnih elemenata rečeničnoga ustroja implicitno je definirana jednostavna rečenica. Naime, jednostavna je rečenica ona sintaktička jedinica "s jednim temeljnim rečeničnim ustrojstvom" (Silić i Pranjković 2005:315), bila ona "raščlanjena" (ili "dvočlana", u kojoj postoji eksplicitan odnos između predikata i subjekta) ili "neraščlanjena" (ili "jednočlana", u kojoj ne postoji subjektni, nego samo predikatni skup). Složene se rečenice potom grade iz jednostavnih sklapanjem (Barić i dr. 2003:455, Silić i Pranjković 2005:319). Načelno se razlikuju dvije vrste sklapanja[8], koje rezultiraju dvjema vrstama odnosa među jednostavnim rečenicama koje se sklapaju u složenu rečenicu i pritom se nazivaju surečenicama:

[7] Može se primijetiti da (Barić i dr. 2003:431) vežu priložnu oznaku uz neglagolski predikat samo u jednome primjeru, gdje se uporaba takva priloga može također objasniti i u smislu pojačivača, a ne priložne oznake.
[8] Ovdje je navedena samo "eksplicitna koordinacija i subordinacija" (Silić i Pranjković 2005:320). Implicitna se uvodi bez veznika, a umjesto njih se koristi interpunkcija, dok se iskazane pravilnosti u velikoj mjeri zadržavaju neovisno o eksplicitnosti, odnosno implicitnosti sklapanja.

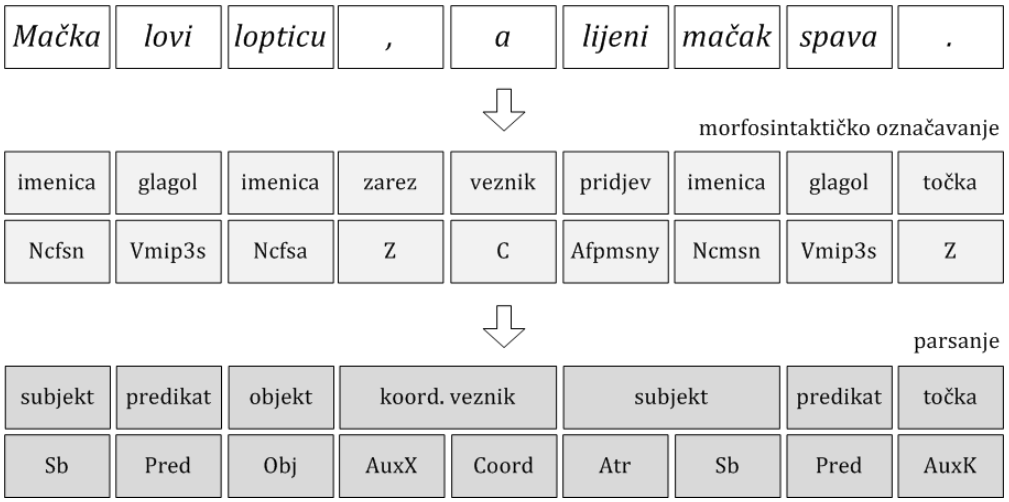
1. sklapanje povezivanjem svojstveno je složenim rečenicama s nezavisnim odnosom (ili koordinacijom), odnosno nezavisno-složenim rečenicama, a
2. sklapanje uvrštavanjem svojstveno je složenim rečenicama sa zavisnim odnosom (ili subordinacijom), odnosno zavisno-složenim rečenicama.

Budući da se i nezavisno- i zavisno-složene rečenice sastavljaju eksplicitno, s pomoću veznika, ne razlikuju se samo po tipu sklapanja, već i po odabiru veznika. Pritom "veznici nezavisno-složenih rečenica ne pripadaju nijednoj od surečenica koje povezuju", a "veznici zavisno-složenih rečenica sastavni su dio zavisnih surečenica" (Silić i Pranjković 2005:320), a također se razlikuju i po obavijesnosti. Razredba nezavisno- i zavisno-složenih rečenica hrvatskoga jezika u literaturi (usp. Barić i dr. 2003:457, Silić i Pranjković 2005:322) dalje popisuje načine na koje se izborom veznika mogu jednostavne rečenice sklapati u složene. Ovdje se ne raspravlja detaljnije o vrstama zavisno- i nezavisno-složenih rečenica, a razredba je dana u (Barić i dr. 2003:582, Silić i Pranjković 2005:328, Silić i Pranjković 2005:356).

Prema ovdje ocrtanim gramatičkim pravilima, složene se rečenice sklapaju od jednostavnih, a jednostavne se dalje razlažu po gramatičkome ustrojstvu na samostalne i nesamostalne ustrojbene jedinice (predikate, subjekte, objekte i priložne oznake te atribute i apozicije) koje se potom mogu konačno raščlaniti do razine samih (predikatnih, subjektnih, objektnih, itd.) riječi. Tako se provodi sintaktička analiza, odnosno parsanje tih rečenica. Parsati rečenicu, s toga gledišta, znači odrediti ulogu svake riječi u pripadajućem elementu gramatičkoga ustrojstva, ulogu svakoga od tih elemenata u pripadajućoj jednostavnoj rečenici te eventualno (ako je parsana rečenica složena) način povezivanja jednostavnih rečenica u složenu rečenicu. Parsanje u tome okviru ilustrirano je primjerom 2-1.

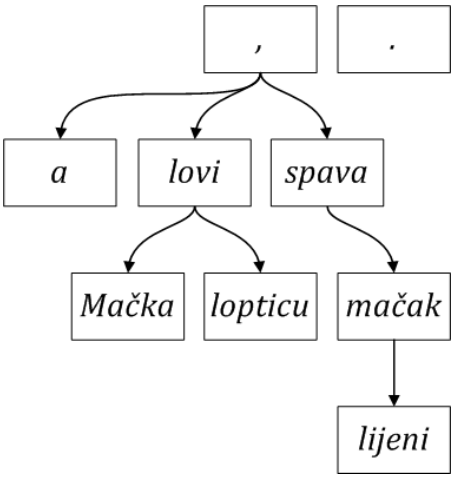
Primjerom se ilustrira međuovisnost morfološke analize – odnosno morfosintaktičke analize budući da se višeznačnost pojedinih riječi ovdje razrješava u kontekstu rečenice, kao npr. u slučaju riječi "lovi" (glagol "loviti" ili imenica "lova"?) ili "mačka" (ženski rod, nominativ ili muški rod, genitiv?), o čemu se još raspravlja kasnije u tekstu – i sintaktičke analize, a također daje i neke konkretne instancije prethodno postavljenih pravila za pojedine elemente rečeničnoga ustroja. Tako se može konkretno uočiti da je subjekt imenica u nominativu, da se imenična riječ subjekta s pridjevskim (odnosno sročnim) atributom slaže u rodu, broju i padežu, da je objekt imenica u akuzativu, da su predikati izraženi jednom glagolskom riječju te da je koordinacija ostvarena spajanjem veznika i interpunkcijskog

znaka[9]. Također je pokazano kako pojedine riječi u višerječnim jedinicama (ovdje, konkretno, subjektnoj riječi i koordinacijskom vezniku) također mogu imati zasebne sintaktičke funkcije (atribut, subjekt)[10].



Primjer 2-1 Parsanje nezavisno-složene rečenice

Primjer 2-1 može se također pobliže opisati i s gledišta redoslijeda uvođenja u rečenicu, odnosno otvaranja mjesta pojedinim elementima rečeničnoga ustroja. To je ilustrirano primjerom 2-2.



Primjer 2-2 Redoslijed uvođenja riječi u rečenicu

[9] Vidljivo primjerice iz morfosintaktičkih oznaka Ncmsn (en. *noun*, *common*, *masculine*, *singular*, *nominative*) i Afpmsny (en. *adjective*, itd.), koje su preuzete iz specifikacije Multext East v3/v4 za hrvatski jezik, o kojoj se raspravlja kasnije u tekstu (Erjavec 2004, 2008).

[10] Kao u prethodnoj napomeni, ovdje su korištene oznake sintaktičkih funkcija iz Praške ovisnosne banke stabala (Hajič i dr. 2000), o kojima će se također raspravljati naknadno.

Na primjeru se vidi da predikati otvaraju mjesto za subjekte i objekte u surečenicama ("lovi" za "Mačka" i "lopticu" te "spava" za "mačak") te da subjekt druge surečenice otvara mjesto za atribut ("mačak" za "lijeni"). Budući da se radi o nezavisno-složenoj rečenici, spajanje surečenica opisano je postavljanjem koordinacijskoga veznika za element koji u rečenicu uvodi predikate pripadajućih surečenica.

2.1.1.3 Jezična višeznačnost kao optimizacijski problem

Međutim, svrha primjera 2-1 ovdje nije samo pokazati prethodno izložena pravila o izgradnji rečenice iz riječi i višerječnih jedinica koje predstavljaju elemente njezina sintaktičkog ustroja, već i implicirati prirodu procesa sintaktičke analize prema danim pravilima kad je provodi čovjek, kako bi se pomoću te implikacije ocrtala složenost problema parsanja računalom te smjestio taj problem u odgovarajuće znanstvene okvire. Uz ranije pitanje ("Što čovjek želi doznati kad provodi sintaktičku analizu?"), postavlja se dodatno pitanje: Kako čovjek provodi sintaktičku analizu (usp. Jurafsky i Martin 1999:463), odnosno kako u rečenici jednoznačno prepoznaje riječi u elementima rečeničnoga ustroja te same elemente i njihove međuovisnosti i uloge? Postavlja se, dakle, pitanje o tome kako čovjek razumije neku rečenicu, tekst ili jezik općenito, bilo u svrhu analize, sinteze ili prihvaćanja obavijesti. Pitanjima usvajanja, razumijevanja i proizvodnje jezika bavi se psiholingvistika (usp. Aitchison 1998). Na tome području, postoje različiti teorijski modeli i pripadajući eksperimenti (usp. Rayner i dr. 1983, Fraizer 1987, Trueswell i Tanenhaus 1994, Lewis 1999, Townsend i Bever 2001) koji govore o tome kako ljudi obrađuju rečenice, odnosno kako usvajaju obavijesti koje su u njima sadržane. Načelno se razlikuje modularni (kod kojega se rečenica obrađuje nizom modula, najčešće serijski, tj. tako da izlaz iz jednoga modula predstavlja ulaz u drugi modul, pa npr. sintaktički modul po obradbi prosljeđuje podatke semantičkom modulu, itd.) od interaktivnoga modela razumijevanja (u kojemu se obradba obavlja istovremeno na više razina). Modelima je u pravilu zajednički cjeloviti pristup obradbi ulaznih podataka, odnosno misaoni proces u primatelju jezične poruke koji podrazumijeva povezivanje njegova leksičkog, sintaktičkog, semantičkog i pragmatičkog aparata te znanja o svijetu (usp. Vučković 2009:7, Žic-Fuchs 1991) s ciljem razumijevanja primljene poruke, u ovome slučaju valjane rečenice nekoga prirodnog jezika. Pretpostavlja se, dakle, uporaba svih raspoloživih razina jezične svijesti kako bi se ostvarila komunikacijska svrha jezične poruke, što je također opravdano i očekivano u kontekstu komunikacijskih modela općenito (usp. Craig 1999, Miller 2005, Barnlund 2008) te, specifičnije, njihove

ekonomičnosti. Pojednostavljeno, gledano s gledišta modeliranja razmjene obavijesti, očekivano je da će čitav čovjekov jezični aparat biti iskorišten, odnosno da će svaka njegova komponenta biti aktivirana u svakom procesu razmjene obavijesti jer bi u protivnome bio suboptimalan u ostvarenju svoje osnovne svrhe.

Modeli razumijevanja jezika usko su vezani uz problem jezične višeznačnosti, odnosno činjenicu da je jezik višeznačan, budući da ona predstavlja problem s gledišta ovoga istraživanja – ukoliko u jeziku ne bi postojala višeznačnost, čovjekov jezični aparat te teorijski i računalni modeli koji iz njega proizlaze bili bi vjerojatno znatno jednostavniji. Istraživači s jednoga gledišta (usp. Vučković 2009:77) tvrde da je "govoreni jezik rijetko višeznačan" budući da se njegova višeznačnost razrješuje "dodatnim informacijama koje proizlaze iz zajedničkoga kontekstnog znanja". Međutim, budući da je s gledišta pisanoga jezika i računalnoga modeliranja problema jezične obradbe nužno razmatrati jezične podatke bez dodatnih podataka koji proizlaze iz vanjezičnoga konteksta, opravdano je smatrati jezik višerazinski višeznačnim. Razlikuju se najčešće (Jurafsky i Martin 1999:4, Manning i Schütze 2003:17), po razinama jezičnoga opisa, tri vrste jezične višeznačnosti, ovdje s naglaskom na jezik u pismu.

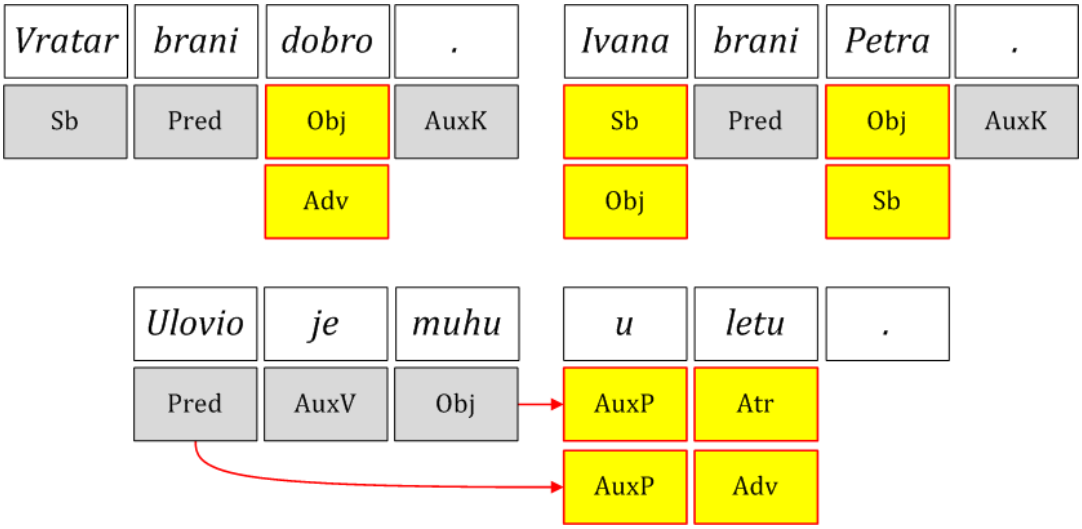
1. Leksička višeznačnost, odnosno svojstvo pojedinih riječi ili izraza da u nekome jeziku imaju više različitih značenja najčešće se navodi kao prva vrsta jezične višeznačnosti. Načelno se razlikuje homonimija i polisemija, odnosno homonimi od polisema. Homonimi su one riječi koje dijele pisanu i izgovornu formu, ali su različitoga značenja. (Dakle, s gledišta pisanoga jezika, razmatraju se samo homografi – riječi istoga pisanog oblika, ali različitih značenja, koji se nazivaju i istopisnice (Tadić 1994, Tadić 2003:122), a sama pojava istopisnost.) Polisemi se definiraju koristeći potpuno istu definiciju – i polisemi su riječi koje dijele formu, no ne i značenje, a jedina razlika između homonima i polisema je u postanku riječi: (pravim) homonimima se smatraju riječi koje su nepovezane etimološki, tj. postankom, dok polisemi dijele i etimologiju (usp. Raffaelli 2008, Raffaelli 2009, Hudeček i Mihaljević 2009, Hurford i Heasley 1983:123, Palmer 1981:100). Razlikuje se dodatno unutarnja od vanjske istopisnosti (Tadić 2003:122): unutarnje su istopisne dvije istopisne pojavnice (oblika) koje imaju istu lemu (polazni oblik), ali različiti morfosintaktički opis, dok su vanjski istopisne one koje potječu od različitih polaznih oblika. Unutarnja istopisnost je karakteristična za jezike poput

hrvatskoga, pa se navodi (usp. Tadić 2003:123), a očekivan je, pa stoga i višestruko razmatran njezin utjecaj na rješavanje nekih problema iz obradbe hrvatskoga jezika, poput lematizacije i morfološke normalizacije (usp. Agić i dr. 2009, Šnajder 2010). Kako je ranije navedeno, u rečenici iz primjera 2-1 leksički su višeznačni, odnosno istopisni, oblici "lovi" (lema može biti imenica "lova" i glagol "loviti", a također za svaku od navedenih lema postoje po dva različita tumačenja padeža, odnosno glagolskog lica i vremena) i "mačka" (imenica muškoga ili ženskoga roda, u različitim padežima), ali i "veliki" (različiti oblici pridjeva "velik" kao primjer unutarnje istopisnosti). Ova tumačenja rečenice iz primjera 2-1 preuzeta su s Hrvatskoga lematizacijskog poslužitelja[11] (Tadić 2005, Tadić 2006), zasnovanog na Hrvatskome morfološkom leksikonu (Tadić i Fulgosi 2003), o kojem će se još govoriti kasnije u tekstu. Leksičku višeznačnost, odnosno istopisnost, nije moguće razriješiti bez rečeničnoga konteksta u kojem se višeznačna riječ ostvaruje.

2. Sintaktička višeznačnost je svojstvo pojedinih fraza, odnosno rečenica, da mogu imati više različitih gramatičkih, odnosno sintaktičkih tumačenja. Pojednostavljeno, sintaktički su višeznačne one fraze i/li rečenice koje se mogu, u skladu s prethodno postavljenim sintaktičkim formalizmom, valjano parsati na više različitih načina (usp. Kroeger 2005:26). Sintaktička višeznačnost može, ali ne mora nužno biti uzrokovana leksičkom višeznačnošću; svojstveno joj je da proizlazi iz odnosa riječi koje se sklapaju u elemente rečeničnoga ustroja. Budući da rečenica iz primjera 2-1 nije sintaktički višeznačna, ovaj je oblik višeznačnosti ilustriran[12] nekim hrvatskim rečenicama na primjeru 2-3. Primjerom su prikazana dva različita načina uvođenja sintaktičke višeznačnosti u rečenice hrvatskoga jezika – u jednome je načinu višeznačnost posljedica istopisnosti, a u drugome je nevezana uz morfosintaksu jer proizlazi isključivo iz činjenice da se riječ ili izraz može u kontekstu rečenice tumačiti na više različitih načina, odnosno da može tvoriti više različitih elemenata rečeničnoga ustroja, a da pritom svako tumačenje bude valjano za tu rečenicu s gledišta odabranoga sintaktičkog formalizma, odnosno gramatičkih pravila. Tako je prva rečenica iz primjera ("Vratar brani dobro.") sintaktički višeznačna jer riječ "dobro" može u toj rečenici biti objekt (ukoliko se radi o obliku opće imenice srednjega roda "dobro", primjerice, kao u "Vratar brani opće dobro.") ili priložna oznaka načina (ukoliko se radi o prilogu "dobro", kao u "Vratar jako dobro brani.").

[11] Dostupan na URL-u http://hml.ffzg.hr/.
[12] Primjeri za engleski jezik npr. u (Zimmer 2010), v.URL http://en.wikipedia.org/wiki/Syntactic_ambiguity.

U drugoj rečenici, "Ivana" (žensko ime, nominativ) može biti subjekt samo ukoliko je "Petra" (muško ime "Petar", akuzativ) objekt i obratno: "Ivana" može biti objekt (muško ime "Ivan", akuzativ) samo ako je "Petra" subjekt (žensko ime, nominativ). Osim istopisnosti, u obje se rečenice može primijetiti i da je sintaktička višeznačnost uvedena i relativno slobodnim redoslijedom riječi, karakterističnim za hrvatski jezik (usp. Barić i dr. 2003:583, Manning i Schütze, 2003:96). Primjerice, ovisno o tumačenju, odnosno odabiru jednoga od dva ishoda parsanja druge rečenice, redoslijed riječi može biti *subjekt-predikat-objekt* (koristi se u gramatikama skraćenica *SPO*, a često u literaturi i *SVO*, od en. *subject-verb-object* (usp. Tomlin 1986)), ali i *objekt-predikat-subjekt* (*OPS*, *OVS*). Redovi riječi u rečenicama hrvatskoga jezika koji odudaraju od poretka SVO nazivaju se stilski obilježenima (usp. Barić i dr. 2003:590) ili afektiranima budući da se poredak namjerno mijenja kako bi se istaknule neke riječi ili skupine riječi u svrhu prenošenja i naglašavanja obavijesti prema autorovoj namjeri.



Primjer 2-3 Tri sintaktički višeznačne rečenice

Treća rečenica predstavlja primjer mogućnosti višestrukoga tumačenja nekoga izraza kao nekog od elemenata rečeničnoga ustroja budući da se izraz "u letu" u rečenici "Ulovio je muhu u letu." može odnositi na predikat (onaj koji je ulovio muhu, ulovio ju je dok je letio, kao u "U letu je ulovio muhu."), ali i na objekt (muha je ulovljena dok je letjela, kao u "Ulovio je muhu dok je letjela."). U prvome slučaju, ako višeznačni element u rečenicu uvodi predikat, radi se o priložnoj oznaci načina (jer pobliže označava način na koji je radnja izvršena), a ako ga uvodi objekt,

uvodi ga kao atribut (jer pobliže označava objekt, kao u "Ulovio je leteću muhu."). Sva tri primjera uvode višeznačnost koju je sintaktičkom analizom nemoguće razriješiti: svaka od njih ima po dva različita i jednakovrijedna tumačenja koja su u potpunosti u skladu s gramatičkim pravilima hrvatskoga jezika.

3. Semantička višeznačnost je svojstvo jezičnih elemenata – u ovdje razmatranom slučaju riječi, izraza i rečenica – da mogu prenositi više različitih i jednakovrijednih značenja. Primjerice, u rečenici "Svatko je pojeo ručak.", nejasno je je li svaka osoba pojela svoj ručak (kao u "Svatko je pojeo po jedan ručak.") ili su sve osobe jele od istoga posluženog ručka. Slično, u rečenici "Na stolu je bio sat i on ga je pogledao.", nije jasno je li vršitelj radnje iz druge surečenice pogledao sat ili stol iz prve surečenice. Semantička višeznačnost nije usko vezana uz ovo istraživanje, pa se o njoj dodatno ne raspravlja.

Iz prikazanoga je vidljivo da sva tri razreda jezične višeznačnosti dijele jedno zajedničko svojstvo. Naime, leksičku je višeznačnost nemoguće razriješiti bez rečeničnoga okruženja u kojem se ostvaruje, sintaktičku bez tumačenja značenja pojedinih ustrojbenih elemenata rečenica, a semantičku višeznačnost nije moguće razriješiti bez razmatranja širega obavijesnoga konteksta jezične poruke. Poopćeno, razrješivanje višeznačnosti jezika u nekom njegovom ostvaraju (riječi, izrazu, rečenici) na jednoj razini jezičnoga opisa zahtijeva posezanje za analizom istoga ostvaraja na višoj (ili višim) razinama jezičnoga opisa. U skladu s ranijim razmatranjem obavijesne uloge jezika, može se reći da se uporaba cjelokupnoga čovjekova jezičnog aparata s ciljem razumijevanja obavijesti sadržane u jezičnoj poruci prije svega odnosi na razrješivanje višeznačnosti koje ta poruka može sadržavati kako bi se od svih u njoj sadržanih tumačenja odabralo onu valjanu (ili podskup valjanih), odnosno onu koju je pošiljatelj obavijesti želio prenijeti. Potreba za višeznačnošću u jeziku i dalje je predmet proučavanja, a najnovije spoznaje (usp. Piantadosi i dr. 2012) upućuju na činjenicu da se postojanje jezične višeznačnosti – i, štoviše, njezine neizostavne prisutnosti u svim jezicima i na svim razinama jezičnoga opisa – može obrazložiti upravo preko njezine komunikacijske uloge, a svrhu pronaći u povećanju učinkovitosti razmjene obavijesti.

Višeznačnost u jeziku postoji i njegovo je neizostavno svojstvo te je podložna analizi i mjerenjima koja su pokazala da udjelom i značajem na svim razinama jezičnoga opisa predstavlja važan čimbenik u ostvarenju komunikacijske uloge jezika te da doprinosi povećanju njezine učinkovitosti. Iz toga proizlazi, donekle suprotno intuiciji, da višeznačnost

olakšava čovjekovu jezičnom aparatu razumijevanje jezičnih poruka, odnosno da je on posebno prilagođen baš za rukovanje višeznačnošću. Primjerice, pri parsanju rečenice ljudskim jezičnim aparatom uspješnost prihvaćanja obavijesti sadržane u toj rečenici implicira uspješno razrješivanje svih leksičkih, sintaktičkih i semantičkih višeznačnosti koje je sadržavala. Na sintaktičkoj razini razumijevanje poruke i uspješno razrješenje svih višeznačnosti implicira da je odabrano i valjano sintaktičko tumačenje rečenice iz skupa svih mogućih njezinih tumačenja. Ukoliko se želi izraditi računalni sustav, odnosno parser, kojim je također moguće od svih mogućih parsanja odabrati ono točno – ili bilo kakav računalni model i/li sustav koji je u stanju jednoznačno raščlanjivati jezik na bilo kojoj od razina jezičnoga opisa – onda u model toga sustava mora nužno biti ugrađeno i rukovanje višeznačnošću.

2.1.1.4 Parser kao inteligentni računalni sustav

Kaže se da je cilj svakoga računalnog sustava, koji obrađuje jezične podatke s ciljem razumijevanja jezične poruke na nekim razinama jezičnoga opisa, odrediti na nekoj razini strukturu ulaznoga teksta. Primjerice, parserom se najčešće želi pronaći u rečenici ili tekstu odgovor na jednostavno pitanje "Tko je učinio što komu?" (usp. Manning i Schütze 2003:17), koristeći pritom elemente rečeničnoga ustroja kao implikatore semantičkih uloga (subjekt kao vršitelja radnje, predikat kao radnju, objekt kao trpitelja radnje, itd.). Međutim, da bi se semantičke uloge uopće pokušalo valjano implicirati, potrebno je točno odrediti elemente rečeničnoga ustroja u rečenicama prirodnoga jezika kao proizvoljno složenim nositeljima obavijesti proizvoljne složenosti. Sintaktički formalizmi najčešće su takvi (usp. Manning i Schütze 2003:18) da njihovo prebacivanje u računalni model i njegova izvedba rezultira parserima koji i za jednostavne rečenice uvijek daju višestruka tumačenja, čiji se broj značajno uvećava usložnjavanjem rečenica. Primjerice, navodi se da (Manning i Schütze 2003:18) za rečenicu engleskoga jezika "List the sales of the products produced in 1973 with the products produced in 1972." jedan parser (Martin i dr. 1987) nudi 455 različitih sintaktičkih tumačenja. Iz toga je očigledna potreba za razrješivanjem višeznačnosti unutar računalnih modela i sustava za obradbu jezika, neovisno o ciljanoj razini jezičnoga opisa. Ti modeli i sustavi stoga moraju na neki način modelirati, odnosno simulirati ili aproksimirati rukovanje višeznačnošću svojstveno jezičnomu aparatu čovjeka, kojemu je, nadalje, svojstvena inteligencija (cf. Dennet 1994) što posljedično implicira da spomenuti računalni

modeli i sustavi moraju za razrješivanje višeznačnosti modelirati ljudsko ponašanje, odnosno ljudsku inteligenciju. Ovime se bavi područje umjetne inteligencije.

U (Russell i Norvig 2009:2) dano je osam definicija područja umjetne inteligencije koje razmatraju četiri različita pristupa modeliranju ljudskoga ponašanja, odnosno inteligencije: pristup orijentiran ljudskom razmišljanju, ljudskom djelovanju, racionalnom razmišljanju i racionalnom djelovanju. S gledišta iznesenoga u ovome radu, indikativna je definicija iz pristupa orijentiranoga ljudskom djelovanju koja kaže (Kurzweil 1992) da se područje umjetne inteligencije bavi "stvaranjem strojeva koji" usporedivo dobro "izvršavaju zadatke za koje ljudi koriste inteligenciju kad ih izvršavaju". Utoliko područje umjetne inteligencije, koje se ponekad naziva granom računalne znanosti, crpi, između ostaloga, i znanja iz (usp. Russell i Norvig 2009:5) filozofije, računalne znanosti i tehnologije, matematike, ekonomije, psihologije, kognitivne znanosti, neuroznanosti i lingvistike. Značajno s gledišta ovoga rada – budući da je jeziku svojstvena inteligencija (i obratno), pa se područje umjetne inteligencije samim time bavi i jezikom – ističe se u literaturi (usp. Russell i Norvig 2009:16) i da su područja umjetne inteligencije i suvremene lingvistike – koja se danas presijecaju u vidu računalne lingvistike te obradbe prirodnoga jezika, odnosno jezičnih tehnologija – stvorena otprilike u isto vrijeme, 1950-ih, te da su mnoge spoznaje s jednoga područja utjecale na drugo i obratno. Pritom se načelno razlikuje (usp. Tadić 2003:10) računalna lingvistika od računalne obradbe prirodnoga jezika prvenstveno u motivaciji istraživača: računalna lingvistika kao "zasebna lingvistička grana definirana svojom odjelitom metodologijom" cilja "izraditi računalne modele funkcioniranja pojedinih jezičnih podsustava ili sustava u cjelini" u svrhu "što potpunijega opisa nekoga jezika", dok se kod strojne obradbe prirodnoga jezika ("jezičnih podataka") na prvo mjesto stavlja "obradba podataka" i cilja na "što učinkovitiju i što bržu obradbu uz što manji utrošak računalnih resursa", pa su jezični podatci pritom "samo još jedna vrsta podataka čija obradba potpada pod iste zahtjeve".

Može se zaključiti da su načela izrade računalnih modela i sustava koji simuliraju ljudsku inteligenciju neovisna o načinu na koji će se oni po izradi koristiti, bilo za stjecanje novih spoznaja o jeziku i ljudskom ponašanju ili za učinkovitiju obradbu podataka. Budući da je ishodište ovoga istraživanja izrada i vrjednovanje jednoga takva sustava – parsera koji je u stanju učinkovito parsati rečenice hrvatskoga jezika – dalje se, neovisno o krajnjoj svrsi toga sustava, raspravlja upravo o općim načelima izrade, odnosno različitim pristupima izradi parsera prirodnoga jezika.

2.1.2 Parsanje formalnoga i prirodnoga jezika

Parser je prethodno definiran kao računalni sustav kojim se provodi sintaktička analiza rečenica nekoga jezika u skladu sa zadanim sintaktičkim formalizmom. Ovdje se raspravlja o različitim pristupima osmišljavanju i izvedbi računalnoga modela parsera s obzirom na izvedbenu paradigmu i sintaktički formalizam.

2.1.2.1 Formalne definicije

Jedna općenita definicija (Grune i Jacobs 1998:11) kaže da je parsanje "postupak otkrivanja jezične strukture u nekoj njezinoj slijedom predstavljenoj pojavnosti, u skladu s danom gramatikom". Definicija je općenita utoliko što su *jezik*, *struktura*, *slijedna pojavnost* i *gramatika* u njoj općeniti termini koji se mogu odnositi, primjerice, na prirodni jezik (za koji bi se radilo o sintaktičkoj strukturi, slijedu riječi, odnosno fraza i/li rečenica te sintaksi predmetnoga jezika), jezik za programiranje, ali i notni zapis ili model ponašanja životinja, odnosno "bilo koji slijed elemenata u kojemu prethodni elementi slijeda na neki način ograničavaju izbor idućega elementa u slijedu". Stoga se ovdje definiraju navedeni pojmovi – jezik, slijed, struktura i gramatika – najprije kao apstraktni entiteti pa onda i s gledišta parsanja prirodnoga jezika.

U prethodno predstavljenome značenju jedan jezični slijed, odnosno jedno moguće ostvarenje neke jezične strukture jedan je element jezika. Dakle, i jezična struktura i jezični slijed koji iz nje proizlazi dijelovi su jezika, a jezik je pritom u potpunosti opisan nekom gramatikom. U tako postavljenim odnosima gramatika definira jezičnu strukturu koja je takva da se iz nje mogu ostvariti svi oni jezični sljedovi koji pripadaju opisanomu jeziku. Jezik je pritom skup svih sljedova ostvarenih iz svih jezičnih struktura koje može opisati zadana gramatika toga jezika. Tako definiran jezik naziva se formalnim jezikom, a gramatika formalnom gramatikom (usp. Dovedan 2003). Slijedi njihova formalna definicija i formalne definicije svih uključenih pojmova.

2.1.2.1.1 Rječnik, riječ, rečenica i formalni jezik

Neka je skup Σ skup svih elemenata l_i: $\forall i, l_i \in \Sigma$ koje je moguće poredati u slijed. Skup Σ naziva se abecedom, a l_i predstavlja jedan simbol ili slovo iz abecede Σ. Bilo koji konačni slijed elemenata l_i iz abecede Σ zove se riječ i najčešće se označava kao w. Skup svih riječi označava se kao Σ^*. Ovakva notacija i definicije abecede, slova i riječi svojstveni su teoriji

formalnih jezika u kojoj su slova i riječi apstraktni pojmovi čije mjesto u konkretnim primjenama mogu zauzeti, kako je ranije navedeno, bilo koji entiteti kojima je svojstveno slijedno uvjetovanje. U takvome pojmovnom okruženju, s obzirom na ilustrativnu definiciju jezika iz prethodnoga odlomka, gramatika bi opisivala sve one jezične strukture koje se ostvaruju u obliku svih riječi koje pripadaju nekomu jeziku, a riječi bi se pritom gradile nizanjem slova iz abecede. Međutim, s gledišta ovoga istraživanja – te s obzirom na raspravu o gramatikama, rečenicama i riječima u elementima rečeničnoga ustroja – odabir pojmovlja u kojem gramatika opisuje strukture preko koje se slova spajaju u riječi nije opravdana ni intuitivna, pa se bira pojmovlje tako da gramatika opisuje strukture preko kojih se riječi spajaju u rečenice nekoga jezika. Ovdje se rječnik, riječ, rečenica i jezik definiraju upravo na taj način. Slijede definicije.

Rječnik $\mathcal{V}$ je skup svih elemenata $w_i : \forall i, w_i \in \mathcal{V}$ koje je moguće poredati u slijed. Jedan element w_i rječnika $\mathcal{V}$ pritom se naziva *riječ*. Rječnik može biti konačan ili beskonačan skup riječi.

Riječi rječnika moguće je poredati u sljedove. *Rečenica* je neki konačni slijed riječi iz rječnika, $s = \{w_i\}, m \leq i \leq n, 0 \leq m, n \leq |\mathcal{V}|$.

Skup svih podskupova rječnika $\mathcal{V}$ predstavlja stoga skup svih rečenica koje je moguće izraditi postavljajući riječi $w_i \in \mathcal{V}, \forall i$ u sljedove te se najčešće označava kao $\mathcal{V}^*$, a ponekad – u skladu s notacijom matematičke teorije skupova – i kao $2^{\mathcal{V}}$. Budući da $\mathcal{V}^*$ sadrži sve rečenice koje je moguće izraditi riječima iz $\mathcal{V}$, svaka rečenica koju je moguće sastaviti s pomoću riječi iz $\mathcal{V}$ pripada $\mathcal{V}^*$, odnosno, uz definiciju rečenice, $\forall i, 0 \leq i \leq |\mathcal{V}^*|, s_i \in \mathcal{V}^*$.

Formalni jezik je neki podskup skupa svih mogućih rečenica, odnosno $L \subseteq \mathcal{V}^*$. Nad rječnikom $\mathcal{V}$, odnosno skupom $\mathcal{V}^*$ svih rečenica koje je moguće izraditi nad tim skupom, moguće je definirati načelno beskonačan skup formalnih jezika tako da unija svih tih jezika, odnosno podskupova rečenica, bude jednaka $2^{\mathcal{V}^*}$, dakle, $\bigcup_{i=0}^{\infty} L_i = 2^{\mathcal{V}^*}$. Svaki tako definiran jezik je također načelno beskonačan skup rečenica, iako su same rečenice definirane kao konačni sljedovi riječi. Definicije su ilustrirane primjerom 2-4 u kojem je prikazan skup riječi koji predstavlja jednostavni rječnik, skup svih rečenica koje je moguće sastaviti nizanjem tih riječi te jedan izdvojeni podskup toga skupa, odnosno jedan mogući jezik nad tim rječnikom.

$\mathcal{V}$ = {pada, snijeg}

$\mathcal{V}^*$ = {{pada}, {pada, pada}, {pada, pada, pada}, ..., {snijeg}, {snijeg, snijeg}, ..., {pada snijeg}, {pada pada snijeg}, ...}

L = {{pada snijeg}, {pada pada snijeg}, ..., {snijeg pada}, {snijeg pada pada}, ...}

Primjer 2-4 Beskonačni jezik nad konačnim rječnikom

U primjeru se vidi kako je moguće iz konačnoga rječnika izraditi beskonačan jezik postavljanjem riječi iz rječnika u slijed. Međutim, iz primjera je, kao i iz prethodnih definicija, izostavljen zbog jednostavnosti jedan detalj. Naime, osim definicijama određenih i primjerom ocrtanih rečenica koje je moguće izraditi nizanjem riječi rječnika, uvijek je – za svaki rječnik i formalni jezik – moguće izraditi i praznu rečenicu, odnosno rečenicu koja ne sadrži nijednu riječ. Prazna se rečenica označava simbolom ε i uvijek se dodaje skupu $\mathcal{V}^*$ kao njegov prvi element.

Budući da je svaki formalni jezik načelno beskonačan skup (usp. Grune i Jacobs 1998:18), njega se u pravilu ne definira – ili ga se, točnije, tada u pravilu ni ne može definirati – popisivanjem svih rečenica koje mu pripadaju. Formalni jezik definira se implicitno, formalizmom koji opisuje njegove rečenice. Taj formalizam naziva se formalna gramatika.

Formalna gramatika je, dakle, skup pravila kojima se definiraju rečenice nekoga jezika. Primjerice (usp. Grune i Jacobs 1998:22), pravila se mogu definirati kao u primjeru 2-5.

Ovim tekstnim opisom definirana je gramatika koja opisuje rečenice poput "trokut.", "trokut, trokut.", "trokut, kvadrat, krug.", "krug, kvadrat, kvadrat.", itd. Dani tekstni opis pravila može se zapisati i formalnije, u obliku pravila za preslikavanje iz danih apstraktnih skupova u njihova konkretna ostvarenja, kao u primjeru 2-6.

1. Neka je skup riječi $\mathcal{V} = \{\text{trokut}, \text{krug}, \text{kvadrat}, .\}$.
2. Riječi trokut, krug i kvadrat, kao podskup od $\mathcal{V}$, neka se zbirno zovu geometrijskim likovima, Likovi = {trokut,krug,kvadrat}, Likovi $\subseteq \mathcal{V}^*$.
3. Neka se jedan element iz skupa Likovi naziva Lik, $\forall i, 0 \leq i \leq |\text{Likovi}|, \text{Lik}_i \in \text{Likovi}$. Taj element predstavlja apstrakciju geometrijskoga lika, a njegove konkretne instance

(ostvarenja) su elementi skupa Likovi.

4. Jedan geometrijski lik je jedna rečenica.
5. Jedan geometrijski lik koji slijedi zarez pa još jedan geometrijski lik također je jedna rečenica.
6. Rečenica završava kad su svi apstraktni simboli zamijenjeni konkretnima.
7. Završetak rečenice označava se točkom.

Primjer 2-5 Tekstni opis pravila za definiranje rečenica

1. Dan je skup $\mathcal{V} = \{\text{trokut}, \text{krug}, \text{kvadrat}\}$.
2. Dan je skup $\text{Likovi} = \{\text{trokut}, \text{krug}, \text{kvadrat}\}$, $\text{Likovi} \subseteq \mathcal{V}^*$.
3. $\text{Lik}, \forall i, 0 \leq i \leq |\text{Likovi}|, \text{Lik}_i \in \text{Likovi}$.
4. Neka se apstraktni elementi Rečenica, Likovi, Lik i Kraj ostvaruju na sljedeći način:
    a. Rečenica → Lik Kraj
    b. Rečenica → Likovi Kraj
    c. Likovi → Lik
    d. Likovi → Likovi , Lik
    e. Lik → trokut | krug | kvadrat
    f. Kraj → .
5. Strjelica predstavlja operator ostvarenja apstraktnoga elementa.

Primjer 2-6 Formalniji opis pravila za definiranje rečenica

$$\text{Rečenica} \overset{b}{\Rightarrow}$$
$$\overset{b}{\Rightarrow} \text{Likovi Kraj} \overset{d}{\Rightarrow} \text{Likovi, Lik Kraj} \overset{d}{\Rightarrow} \text{Likovi, Lik, Lik Kraj} \overset{c}{\Rightarrow} \text{Lik, Lik, Lik Kraj} \overset{e}{\Rightarrow}$$
$$\overset{e}{\Rightarrow} \text{krug, Lik, Lik Kraj} \overset{e}{\Rightarrow} \text{krug, kvadrat, Lik Kraj} \overset{e}{\Rightarrow} \text{krug, kvadrat, kvadrat Kraj} \overset{f}{\Rightarrow}$$
$$\overset{f}{\Rightarrow} \text{krug, kvadrat, kvadrat.}$$

$$\text{Rečenica} \overset{*}{\Rightarrow} \text{krug, kvadrat, kvadrat.}$$

Primjer 2-7 Ostvarenje jedne rečenice jezika preko definiranih pravila

Tako definiranim pravilima, mogu se – slijednom zamjenom apstraktnih za konkretne elemente – graditi rečenice jezika. Kaže se da tako zadana gramatika generira rečenice jezika, odnosno da generira jezik. Primjerice, ostvarenje rečenice "krug, kvadrat, kvadrat." prikazano je u primjeru 2-7.

Iz primjera je vidljivo da se rečenice jezika gramatikom stvaraju zamjenom apstraktnih elemenata konkretnim elementima, odnosno riječima rječnika. Konkretni elementi, odnosno riječi, zovu se stoga *završni* (*terminalni*) *simboli*, dok se apstraktni elementi, u skladu s time, zovu *nezavršni* (*neterminalni*) *simboli*. U skupu nezavršnih simbola potrebno je izdvojiti *početni nezavršni simbol* kojim nužno počinje svaki ostvaraj rečenice jezika iz gramatike. U primjeru 2-7 početni nezavršni simbol je simbol Rečenica. Gramatikom se ovako generira rečenica jezika, ali također generira i čitav jezik. Naime, u primjeru 2-7 vidi se ostvaraj jedne moguće rečenice jednom putanjom po pravilima dane gramatike, što je ilustrirano kodnim slovima za pravila preslikavanja iz primjera 2-6 iznad operatora ostvaraja koji se naziva i operatorom izravnoga izvođenja. Primjerice, nezavršni simbol Rečenica preslikan je u primjeru 2-7 u skup nezavršnih simbola Likovi Kraj pravilom b danim u primjeru 2-6. Ukoliko se želi istaknuti da je neka gramatika iz početnoga nezavršnog simbola generirala neku rečenicu, a da se pritom ne želi ilustrirati primjenu svakoga pojedinog pravila, koristi se operator izravnoga izvođenja uz naznaku (broja) ponavljanja, kao na kraju primjera 2-7. U primjeru se također može vidjeti da se nezavršni simboli prevode u završne primjenom pravila slijeva, odnosno redoslijedom uzimanja i prevođenja prvoga nezavršnog simbola s desne strane operatora izvođenja. Moguće je jednakovrijedno izvršavati zamjenu i zdesna.

Primjer 2-7 ilustracija je zapisa formalne gramatike i njezine uporabe za generiranje rečenica jezika koji implicitno definira. Slijedi formalna definicija.

2.1.2.1.2 Formalna gramatika

Formalna gramatika je uređena četvorka $\mathrm{G} = (\mathrm{N}, \mathcal{V}, \mathrm{P}, \mathrm{S})$, gdje je N skup svih nezavršnih (neterminalnih) simbola, $\mathcal{V}$ skup svih završnih (terminalnih) simbola, odnosno rječnik, P je skup svih pravila za prevođenje nezavršnih u završne simbole, odnosno skup *produkcija*, a $\mathrm{S} \in \mathrm{N}$ je početni nezavršni simbol gramatike.

Budući da se nezavršni simboli prevode u završne, nužno vrijedi da je presjek skupa nezavršnih i skupa završnih simbola prazan skup, odnosno da ta dva skupa ne dijele nijedan element: $\mathrm{N} \cap \mathcal{V} = \emptyset$.

Pravila prevođenja, odnosno produkcije definirana su formalno kao preslikavanja iz bilo kojega niza nezavršnih i završnih simbola u bilo koji niz nezavršnih i završnih simbola, uz uvjet nepraznosti niza iz kojega se izvodi: $\forall p \in P : p = u \rightarrow v, u \in (N \cup \mathcal{V})^* N (N \cup \mathcal{V})^*, v \in (N \cup \mathcal{V})^*$. Nepraznost niza iz kojega se izvodi osigurana je zahtijevanjem jednoga nezavršnog simbola u njemu budući da prevođenje jednoga završnog simbola nema smisla s gledišta svrhe produkcija, odnosno činjenice da upravo preko njih gramatika generira jezik. Operator strjelice u zapisu produkcije predstavlja potencijal generiranja, čije se ostvarenje prikazuje ranije definiranim operatorom izvođenja. Za ovako definirane produkcije kaže se često da su neograničene budući da se od njih zahtijeva samo nepraznost niza iz kojega se vrši prevođenje. Budući da su upravo produkcije sredstvo kojim gramatika generira jezik, njihov oblik, struktura i složenost, odnosno zahtjevi ili ograničenja koja se postave na njihovo oblikovanje, određuju strukturu, složenost i ograničenja jezika koji se generira.

Prema ograničenjima u definiranju produkcija razlikuju se hijerarhijski četiri razreda formalnih gramatika i četiri razreda formalnih jezika koje te gramatike respektivno generiraju. Ta se hijerarhija naziva Chomskyjevom hijerarhijom (Chomsky 1959).

1. Skup gramatika tipa 0 obuhvaća sve formalne gramatike, pa se naziva i skupom *neograničenih gramatika*. Njihove produkcije nemaju ograničenja, osim nepraznosti lijeve strane produkcije, kao što je ranije definirano. Jezici koje generiraju nazivaju se *rekurzivno prebrojivim jezicima*.
2. Gramatikama tipa 1 produkcije su ograničene na sljedeći način. One moraju biti oblika $\alpha A \beta \rightarrow \alpha \gamma \beta, A \in N, \alpha, \beta \in (N \cup \mathcal{V})^*, \gamma \in (N \cup \mathcal{V})^+$. Zahtijeva se, dakle, da neterminalni simbol A bude neprazan, kao i niz terminalnih i neterminalnih simbola γ, dok α i β mogu biti prazni. Također, početni nezavršni simbol gramatike ne smije se nalaziti s desne strane ni u jednoj produkciji. Za ovakav se oblik produkcija kaže da se A preslikava u γ u kontekstu α i β, pa se gramatike ovoga tipa nazivaju i *kontekstno-ovisnim gramatikama*, a jezici *kontekstno-ovisnim jezicima*.
3. Gramatike tipa 2 nazivaju se *beskontekstnim gramatikama* (Dovedan 2003) ili kontekstno-neovisnim gramatikama, a jezici beskontekstnim (ili *kontekstno-neovisnim*) *jezicima*. Produkcije su u njima ograničene na oblik $A \rightarrow \gamma, A \in N, \gamma \in (N \cup \mathcal{V})^+$, odnosno, definiraju preslikavanje iz jednoga neterminala u neprazni niz terminala i neterminala, a također zahtijevaju i da početni simbol nikad ne bude s desne strane.

4. Gramatike tipa 3 nazivaju se *regularnim gramatikama* i generiraju skup *regularnih jezika*. Za produkcije vrijedi pravilo o početnom simbolu, a ograničene su tako da vrijedi $A \rightarrow a, B \rightarrow aC \oplus B \rightarrow Ca, A, B, C \in N, a \in \mathcal{V}$. Dakle, jedan se neterminal preslikava u jedan terminal ili jedan terminal koji slijedi ili kojemu prethodi jedan neterminal. Izbor mjesta neterminalnoga simbola na desnoj strani produkcije je isključiv: ukoliko je smješten slijeva, mora biti smješten slijeva za sve produkcije, i obratno za smještaj zdesna.

Skupovi formalnih jezika generiranih formalnim gramatikama četiriju opisanih tipova nisu međusobno isključivi. Vrijedi $\text{Tip } 3 \subset \text{Tip } 2 \subset \text{Tip } 1 \subset \text{Tip } 0$. Dakle, postoje jezici koji su beskontekstni, a nisu regularni, jezici koji su kontekstno-ovisni, a nisu beskontekstni te jezici koji su neograničeni, a nisu kontekstno-ovisni. Obratno, svi regularni jezici su ujedno beskontektsni, svi beskontekstni jezici su kontekstno-ovisni, a svi kontekstno-ovisni jezici su ujedno i rekurzivno prebrojivi jezici.

Prema danim definicijama i razredbama, formalne gramatike produkcijama generiraju, odnosno stvaraju formalne jezike. Ukoliko je potrebno saznati, s druge strane, pripada li rečenica nekom formalnom jeziku sa zadanom gramatikom, dva su moguća pristupa. Jedan se izvodi korištenjem zadane formalne gramatike, a drugi korištenjem formalnoga automata komplementarnog toj gramatici. Slijedi definicija formalnoga automata, a potom i objašnjenje oba pristupa provjeri gramatičnosti rečenice.

2.1.2.1.3 Formalni automat

Svrha formalne gramatike je generiranje (rečenica) nekoga formalnog jezika, pa se kaže i da je formalna gramatika generator jezika. S druge strane, za formalni automat se kaže da je akceptor jezika, odnosno, svrha mu je provjeravati pripadaju li neke rečenice nekomu formalnom jeziku ili ne (usp. Hopcroft i dr. 2006). Formalni se automat može promatrati kao istinosna funkcija koja kao parametar prima rečenicu bilo kojega jezika, a povratna joj je vrijednost podatak o pripadnosti te rečenice jeziku prema kojemu je taj automat definiran. Za dani rječnik $\mathcal{V}$ i jezik L može se reći da za automat M definiran nad jezikom L vrijedi: $M: 2^{\mathcal{V}^*} \longrightarrow \{0, 1\}, \forall s \in 2^{\mathcal{V}^*}, s \in L \Leftrightarrow M(s) = 1, s \notin L \Leftrightarrow M(s) = 0$. Dakle, automat M vraća vrijednost 1 ukoliko rečenica s pripada jeziku L, a u protivnom vraća vrijednost 0.

Slično kao kod formalnih gramatika koje jezik definiraju preko produkcija, formalni automati također su vezani uz određene formalne jezike nekim svojstvima. Također, opet kao

i kod produkcija formalnih gramatika, različitim tipovima formalnih jezika iz Chomskyjeve hijerarhije svojstveni su različiti tipovi formalnih automata. Ipak, svaki formalni automat načelno provjerava rečenice na isti način. Automat čita ulaznu rečenicu, riječ po riječ, te za svaku riječ provjerava, u skladu sa svojim modelom formalnoga jezika uz koji je vezan, može li u tome trenutku pročitani dio ulazne rečenice biti elementom toga jezika, odnosno postoji li i dalje mogućnost da se iz toga pročitanog dijela ulazne rečenice nekim nizom riječi, koje tek imaju biti pročitane, može sastaviti valjana rečenica danoga jezika. Pri provjeri svake od riječi, u skladu s modelom jezika, automat mijenja svoje unutarnje stanje. Ukoliko automat pročita sve riječi s ulaza te pritom dođe u neko od završnih stanja, odnosno onih stanja kojima je svrha potvrditi valjanost ulazne rečenice, automat završava s radom uz povratnu vrijednost 1 i ulazni se niz znakova potvrđuje kao valjana rečenica danoga jezika; u protivnom, automat završava s radom uz povratnu vrijednost 0 i niz se ne prihvaća kao rečenica jezika. Slijedi formalizacija.

Općenito, *formalni automat* definira se kao petorka $\mathrm{M} = (\mathrm{Q}, \mathcal{V}, \delta, \mathrm{q}_0, \mathrm{F})$, gdje je Q konačni *skup stanja automata*, $\mathcal{V}$ je ulazni rječnik danoga jezika, q_0 je *početno stanje automata*, F je skup svih *završnih stanja automata*, a δ predstavlja funkciju prijelaza.

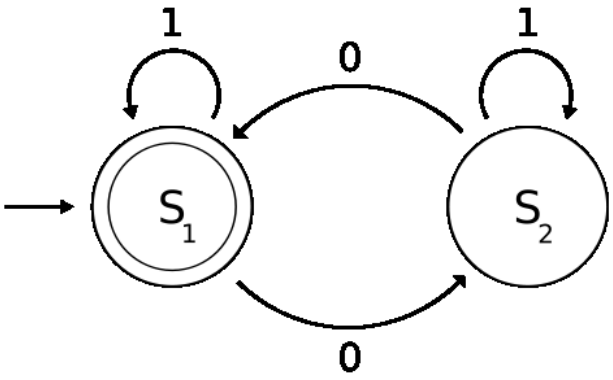
Automat prihvaća rečenicu jezika izmjenom stanja po čitanju riječi te rečenice. Izmjenama stanja po čitanju riječi ostvaruje se svrha formalnoga automata. Te izmjene stanja definirane su funkcijom prijelaza automata. Općenito, *funkcija prijelaza* deklarira se kao $\delta\colon (\mathrm{Q} \times \mathcal{V}) \longrightarrow \mathrm{Q}$, odnosno kao preslikavanje iz produkta skupa stanja i rječnika u skup stanja. Može se također proširiti domena funkcije prijelaza, $\hat{\delta}\colon (\mathrm{Q} \times \mathcal{V}^*) \longrightarrow \mathrm{Q}$, tako da ne prima samo jednu riječ iz rječnika, nego čitave nizove, odnosno rečenice. Funkcija prijelaza definirana je kao $\delta(\mathrm{q_i}, \mathrm{w_j}) = \mathrm{q_k}, \mathrm{q_i}, \mathrm{q_k} \in \mathrm{Q}, \mathrm{w_j} \in \mathcal{V}$ ili $\hat{\delta}(\mathrm{q_i}, \mathrm{s_j}) = \mathrm{q_k}, \mathrm{q_i}, \mathrm{q_k} \in \mathrm{Q}, \mathrm{s_j} \in \mathcal{V}^*$. Dakle, funkcija prijelaza za trenutno stanje automata i pročitanu riječ s trake mijenja stanje automata za svaku riječ ulazne rečenice za koju je definirana. Jedna trojka sačinjena od trenutnoga stanja, ulazne riječi i ciljanoga stanja, predstavlja jedan prijelaz funkcije prijelaza. Ukoliko funkcija prijelaza nije definirana – odnosno, ako nema nijednoga prijelaza – za neku riječ rečenice, ta rečenica ne pripada jeziku koji automat prihvaća.

Funkcija prijelaza formalnoga automata može se smatrati, s gledišta pripadajućega formalnog jezika, jednakovrijednom skupu produkcija formalne gramatike budući da gramatika produkcijama generira upravo onaj jezik koji automat prijelazima prihvaća. Početno stanje automata pritom se podudara s početnim nezavršnim simbolom gramatike, a

dosezanje završnoga stanja radom automata podudarno je sa zamjenom svih nezavršnih simbola završnim simbolima izravnim izvođenjem. Kaže se da je jezik koji prihvaća dani konačni automat jednak $L = \{s \in \mathcal{V}^*: \hat{\delta}(q_0, s) \in F\}$. To je, dakle, skup svih onih rečenica koje se mogu izraditi iz $\mathcal{V}$, a za koje dani automat, pročitavši sve njihove riječi, dosegne neko od završnih stanja.

Prethodna definicija predstavlja uopćeni formalni automat budući da su njegova stanja i prijelazi definirani apstraktno, na način kojim se implicira proizvoljna složenost modela stanja i prijelaza konkretnoga razreda automata. Ukoliko se definicija čita doslovno, ona predstavlja definiciju *konačnoga automata*. Konačni automati su razred automata koji prihvaća onu skupinu jezika koju generiraju regularne gramatike; konačni automati prihvaćaju, dakle, regularne jezike. Prihvat svakoga idućeg razreda formalnih jezika zahtijeva složenije modele stanja i prijelaza kod pripadajućih automata. Slijedi razredba.

1. Konačni automati prihvaćaju regularne jezike. Definicija konačnoga automata dana je ranije. Razlikuje se deterministička od nedeterminističke varijante: u potonjoj jedan prijelaz ne mora biti definiran kao prijelaz u jedno stanje, već i kao prijelaz u skup mogućih stanja, što predstavlja nedeterminizam. Također, nedeterministički konačni automat može imati i prijelaze koji se ostvaruju bez čitanja ulazne riječi, odnosno ε-prijelaze (epsilon-prijelaze). Sve tri vrste konačnih automata jednako su vrijedne, prihvaćaju isti razred formalnih jezika te stoga predstavljaju isti model izračunljivosti. Konačni se automati često prikazuju dijagramom stanja koji predstavlja grafički prikaz rada konačnoga automata. Jedan primjer dijagrama stanja[13] dan je na slici 2-1.



Slika 2-1 Dijagram stanja konačnoga automata

[13] Preuzeto s http://hr.wikipedia.org/wiki/Datoteka:DFAexample.svg (2012-02-07).

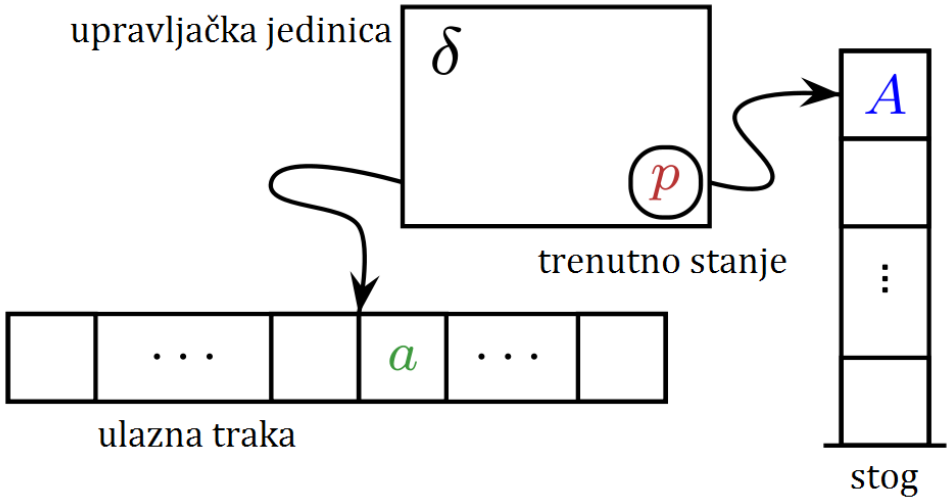
Dijagram stanja prikazuje početno i završno stanje automata (u ovome slučaju, s_1 je i početno i završno stanje), sva stanja (s_1 i s_2) te prijelaze iz jednoga stanja u drugo preuzimanjem simbola s trake (ulazni simboli su 0 i 1).

2. Potisni automati prihvaćaju beskontekstne jezike. Potisni automat je proširenje konačnoga automata radnom memorijom u obliku potisne strukture podataka, odnosno stoga na koji je moguće postavljati posebne stogovne simbole pa ih sa stoga potom i čitati. Definicija automata i funkcije prijelaza stoga mijenja se kako bi se u nju uključilo pisanje na stog i čitanje sa stoga, a također se prilagođava i definicija prihvaćanja ulazne rečenice: potisni automat može prihvatiti ulaznu rečenicu postizanjem završnoga stanja ili pražnjenjem stoga.

   Formalno, *potisni automat* je uređena sedmorka $M = (Q, \mathcal{V}, \Gamma, \delta, q_0, Z, F)$. Uz prethodno definirane elemente (skup svih stanja automata, ulazni rječnik, funkciju prijelaza, početno stanje i skup završnih stanja), Γ predstavlja rječnik stoga – odnosno skup svih riječi koje je moguće staviti na stog ili pročitati sa stoga – te $Z \in \Gamma$ predstavlja početnu riječ na stogu. Funkcija prijelaza definirana je tako da prijelazi automata uključuju stog: $\delta: Q \times (\mathcal{V} \cup \{\varepsilon\} \times \Gamma) \longrightarrow Q \times \Gamma^*$. Piše se $\delta(q, \alpha) \in \delta(p, a, A)$ ako postoji prijelaz iz trenutne konfiguracije automata (automat je u stanju p, ima pročitati riječ a s ulaza te na vrhu slova ima A) u novu konfiguraciju (mijenja stanje u q te umjesto A na stog piše α i uzima novu riječ s ulaza). Bitno je istaknuti kako je preslikavanje iz jednoga u drugo stanje potisnoga automata ograničeno na konačne podskupove skupa $Q \times \Gamma^*$. Za precizniji opis rada potisnoga automata koristi se operator $\vdash_M$ kojim se predstavlja jedna promjena konfiguracije automata u diskretnom vremenu. Tako je za ranije dani prijelaz $\delta(q, \alpha) \in \delta(p, a, A)$ jednakovrijedan zapis $(p, ax, A\gamma) \vdash_M (q, x, \alpha\gamma), x \in \mathcal{V}^*, y \in \Gamma^*$. Koristi se operator $\vdash_M^*$ da prikaže neki diskretan broj prijelaza operatorom $\vdash_M$, slično kao operatori $\Rightarrow$ i $\overset{*}{\Rightarrow}$ u opisu izvođenja formalnom gramatikom. Kao što je ranije spomenuto, potisni automat prihvaća ulazni niz riječi – i implicitno prihvatom definira pripadajući formalni jezik – završnim stanjem ($L(M) = \{w \in \mathcal{V}^*: (q_0, w, Z) \vdash_M^* (f, \varepsilon, \gamma), f \in F, \gamma \in \Gamma\}$) ili pražnjenjem stoga ($N(M) = \{w \in \mathcal{V}^*: (q_0, w, Z) \vdash_M^* (q, \varepsilon, \varepsilon), q \in Q\}$). Za svaki potisni automat M koji prihvaća jezik $L(M)$ završnim stanjem može se definirati potisni automat M' koji prihvaća taj isti jezik pražnjenjem stoga, odnosno tako da vrijedi $L(M) = N(M')$. Pri definiciji konačnoga automata potrebno je, međutim, istaknuti je li namijenjen za prihvat postizanjem završnoga stanja ili

pražnjenjem stoga jer u pravilu za automat M vrijedi $L(M) \neq N(M)$. Model potisnoga automata s ilustracijom[14] ulazne vrpce i stoga dan je na slici 2-2.

Razlikuju se deterministički od nedeterminističkih potisnih automata. Za razliku od konačnih automata, ova dva razreda automata nisu jednakovrijedna, odnosno ne prihvaćaju isti razred jezika. Vrijedi da se za svaku beskontekstnu gramatiku G može definirati (moguće nedeterministički) potisni automat M takav da $L(G) = N(M)$, pa time i automat M' takav da je $L(G) = L(M')$. Detaljna se razrada odnosa deterističkih i nedeterminističkih potisnih automata te beskontekstnih gramatika, uključujući formalne dokaze (ne)jednakovrijednosti, može pronaći u (Autebert i dr. 1997:111, Sipser 1997:101). Nekad se potisni automati definiraju i tako da, umjesto jedne riječi, s ulaza i sa stoga čitaju i pišu nizove riječi; oni se nazivaju uopćenim potisnim automatima.



Slika 2-2 Model potisnoga automata

3. Linearno ograničeni automati prihvaćaju kontektsno-ovisne jezike. Linearno ograničeni automat inačica je Turingova stroja, ograničena na ulaznu traku konačne duljine čiji je početak i kraj označen posebnim simbolima.
4. Turingovi strojevi prihvaćaju rekurzivno prebrojive jezike. Turingov stroj je zamišljen (Turing 1936, Turing 1948:31) kao "hipotetsko računalo s beskonačnom memorijom u obliku beskonačne ulazne trake na kojoj su označene ćelije u koje se može pisati i s kojih se može čitati" i kao "stroj koji može čitati i mijenjati simbol

[14] Preuzeto s http://en.wikipedia.org/wiki/File:Pushdown-overview.svg (2012-02-06) i preuređeno.

upisan u bilo koju od ćelija ulazne trake", a pritom "simbol s kojim se vrši interakcija utječe na daljnje ponašanje stroja" i stroj također "može čitati s vrpce i pisati na vrpcu neograničeno, u oba smjera". U odnosu na konačni i potisni automat, dakle, Turingov stroj nije ograničen smjerom, a također nije ograničen ni na čitanje s trake, već na nju može pisati, ali također i brisati s nje. Turingov stroj je od osobite važnosti za računalnu znanost – budući da je pokazano (usp. Davis 1965:128) kako je moguće, koristeći formalizam Turingova stroja, izraditi jedan automat kojim se može izračunati bilo koji izračunljivi izraz, pa time i modelirati bilo koji drugi automat, kao i digitalno računalo – no ovdje se ne proučava dodatno budući da – kako će kasnije biti pokazano – rekurzivno prebrojivi jezici, koje prihvaća, nisu ovdje u središtu zanimanja.

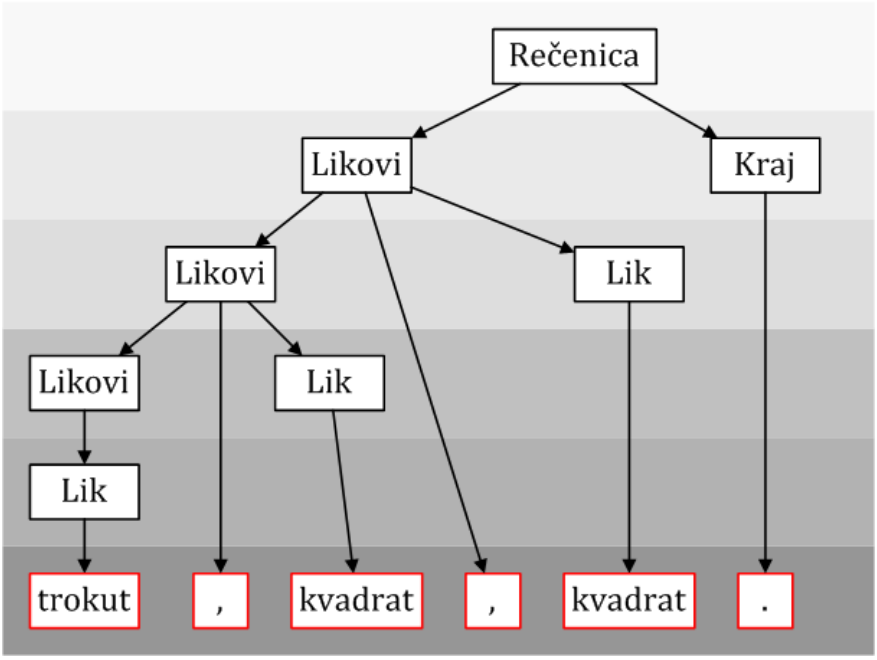
Iako se u pravilu formalnom gramatikom jezik generira, a formalnim automatom prihvaća, moguće je tu logiku i obrnuti pa s pomoću produkcija formalne gramatike provjeravati gramatičnost rečenica, a neprekinutim radom automata po svim putanjama od početnoga do nekoga od završnih stanja implicitno generirati rečenice jezika. Oba pristupa otkrivaju – automat funkcijom prijelaza, a gramatika produkcijama – način na koji se neka rečenica jezika ostvarila preko sintaktičkoga modela toga jezika, sadržanog u definiciji pripadajućega automata, odnosno gramatike.

2.1.2.1.4 Parsno stablo

Primjer 2-7 pokazuje kako sintaktički model, sadržan u produkcijama gramatike opisane primjerima 2-5 i 2-6, preslikavanjem u diskretnom vremenu, u ukupno osam koraka zamjene nezavršnih simbola završnima, određuje jednu rečenicu jezika. Postupak zamjene nezavršnih simbola završnima može se prikazati i grafički, kao u primjeru 2-8 koji se odnosi na rečenicu "trokut, kvadrat, kvadrat." iz primjera 2-7.

Ovaj grafički prikaz prati korake izmjene nezavršnih simbola završnima, kako su definirani produkcijama gramatike, i čita se od gore prema dolje. Dakle, najprije je početni nezavršni simbol Rečenica zamijenjen nezavršnim simbolima Likovi i Kraj. Potom je, zamjenama krajnjih lijevih nezavršnih simbola s desne strane produkcija, dvaput preslikan Likovi u niz Likovi, Lik te su, konačno, svi nezavršni simboli Lik preslikani u završne. Svako je preslikavanje grafički prikazano u obliku razlaganja jednoga nezavršnog simbola u neki broj nezavršnih ili završnih simbola s desne strane produkcije, vezanih uz onoga s lijeve

strane pomoću strjelica i smještenih ispod njega. Posljedično, samo iz nezavršnih simbola mogu se povući strjelice u druge nezavršne ili završne simbole, a završni su simboli nužno na završnoj razini preslikavanja, nakon koje se ne može dalje preslikavati. Ovako izrađen prikaz veze između formalne gramatike i rečenice jezika koju generira naziva se *parsnim stablom*. Kaže se da parsno stablo prikazuje sintaktičku strukturu rečenice prema zadanoj formalnoj gramatici koja ju je generirala. Ako se, primjerice, tumači sintaktička struktura rečenice prema primjeru 2-8, može se zaključiti sljedeće: (1) trokut je Lik, (2) kvadrat je Lik, (3) točka je Kraj, (4) trokut i kvadrat su Likovi, (5) trokut, kvadrat i kvadrat su također Likovi, (6) navedeni Likovi i Kraj čine jednu Rečenicu. Vidi se da parsno stablo omogućuje tumačenje sintaktičkih svojstava riječi i skupina riječi u danoj rečenici, odnosno da predstavlja čitljiv prikaz ishoda postupka parsanja dane rečenice. Može se utoliko reći da se postupkom parsanja želi doznati parsno stablo neke rečenice budući da je iz njega vidljiva čitava njezina sintaktička struktura.



Primjer 2-8 Parsno stablo za rečenicu "trokut, kvadrat, kvadrat."

Parsno stablo ne naziva se stablom samo zbog izgleda grafičkoga prikaza, iako je sam njegov naziv nastao upravo tako, nego prvenstveno zbog toga što odgovara definiciji strukture koja se naziva stablom u računalnoj znanosti, matematici i teoriji grafova (usp. Cormen i dr. 2009:286). O spomenutome se detaljno raspravlja kasnije u ovome radu, no zasad se može napomenuti kako se, u skladu s postavkama tih teorija, elementi parsnoga stabla nazivaju *čvorovima* (to su ovdje nezavršni i završni simboli) i *vezama* (strjelice). U teoriji grafova kaže

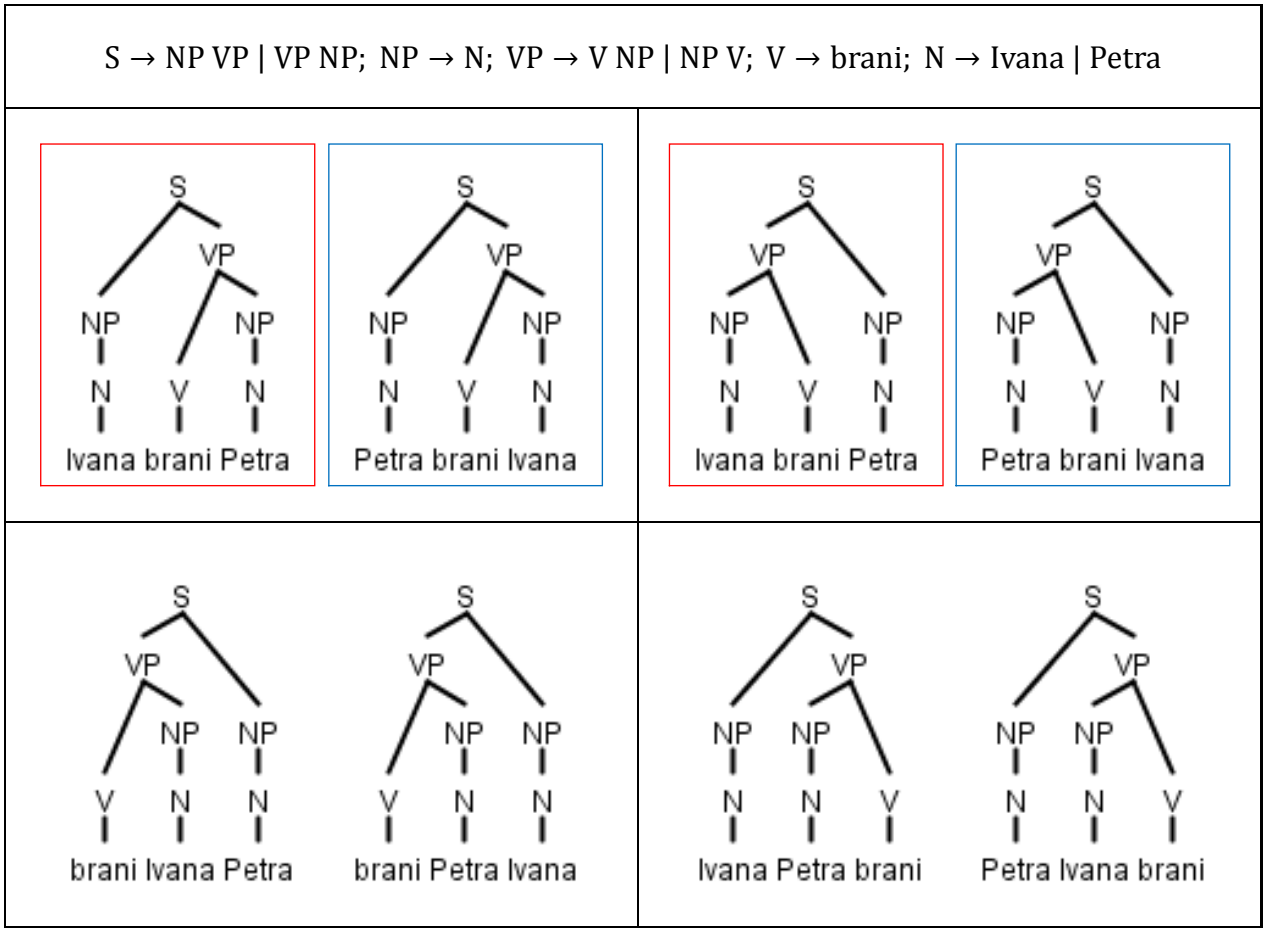
se da je stablo "graf kod kojega su bilo koja dva čvora povezana točno jednom acikličnom putanjom". Svojstva parsnoga stabla kao grafa razmatraju se naknadno. Početni nezavršni simbol u tome se smislu naziva *korijenskim čvorom*, a svi se ostali čvorovi nazivaju *granama*. Čvorovi koji predstavljaju završne simbole nazivaju se i *listovima* budući da iz njih ne može nastati novo grananje. Nadređeni čvor naziva se *roditelj*, a podređeni *dijete*. U primjeru 2-8 korijen stabla je Rečenica te je također roditelj čvoru Likovi i čvoru Kraj na jednoj razini ispod korijenskoga čvora. Svaki čvor Lik roditelj je jednom završnomu simbolu, a svaki od njih predstavlja jedan list stabla. Iako roditeljstvo pojedinih čvorova nad listovima može nastati na nekoj od razina stabla prije posljednje razine, uobičajeno je da se svi listovi umjetno postave na posljednju (najdublju) razinu stabla kako bi rečenica bila prikazana slijedno. U svrhu prikazivanja slijeda nekad se koriste strjelice, kao u primjeru 2-8, ali nisu obavezne budući da se slijed, odnosno smjer ionako implicira čitanjem stabla od vrha prema dnu, odnosno od korijenskoga čvora prema listovima.

Primjeri 2-6, 2-7 i 2-8 prikazuju jednostavnu formalnu gramatiku, način na koji ona generira jednu rečenicu te parsno stablo toga generiranja. Primjer 2-9 utoliko mu je sličan, no prikazuje[15] gramatiku i rečenicu svojstveniju prirodnomu jeziku.

Dana gramatika svojstvenija je prirodnomu jeziku utoliko što njezini nezavršni simboli nose značenje s gledišta sintaktičke analize: početni nezavršni simbol S predstavlja rečenicu, nezavršni simbol NP predstavlja subjektni skup (imeničnu frazu, en. *noun phrase*), simbol VP predstavlja predikatni skup (glagolski skup, en. *verb phrase*), a simboli N i V imenicu i glagol. Gramatika može generirati osam različitih rečenica[16], ako ih se promatra samo kao nizove riječi. To su rečenice "Ivana brani Petra", "Ivana brani Ivana", "Petra brani Ivana", "Petra brani Petra", "brani Ivana Petra", "brani Ivana Ivana", "brani Petra Ivana" i "brani Petra Petra". S druge strane, produkcijama ih može generirati na četiri različita načina, prikazana četirima različitim parsnim stablima, odnosno njihovim grupiranjima u primjeru[17]. U sva četiri parsna stabla rečenica se sastoji od imeničnoga i glagolskoga skupa. Razlika među njima jest u redoslijedu pojavljivanja ($S \Rightarrow NP\ VP$ ili $S \Rightarrow VP\ NP$) i u načinu na koji se ostvaruje glagolski skup ($VP \Rightarrow NP\ V$ ili $VP \Rightarrow V\ NP$); Kartezijev produkt tih načina ostvarivanja predstavlja četiri moguća parsna stabla. Listovi stabla, odnosno riječi nastaju iz

[15] Za prikaz ovih i nekih drugih stabala, korišten je program Parse Tree Application (PTA, autor Keith Trnka), preuzet s http://www.eecis.udel.edu/~trnka/pta/ (2012-02-07).
[16] Sve su vezane uz raniji primjer 2-3.
[17] Zbog jednostavnosti prikaza, neke su rečenice izostavljene iz primjera jer je on fokusiran na prikazivanje svih različitih parsnih stabala, a ne svih različitih rečenica kao nizova riječi.

nezavršnih simbola V i N, i to na način da se V može preslikati samo u glagol "brani", a N se može preslikati u dvije različite imenice, "Ivana" (osobna imenica; ženski rod, nominativ ili muški rod, akuzativ) i "Petra" (također osobna imenica; ženski rod, nominativ ili muški rod, akuzativ). Budući da postoje četiri parsna stabla i dvije imenice s dva različita ostvarenja, ovom se gramatikom može – ako se uzima u obzir sama pojavnost rečenice kao niza riječi, ali i pojavnost njezina sintaktičkog ustroja – ostvariti 16 različitih rečenica[18].



Primjer 2-9 Parsna stabla i sintaktička višeznačnost

S obzirom na parsna stabla iz primjera 2-9, može se, primjerice, promatrati rečenica, odnosno niz riječi "Ivana brani Petra". U primjeru 2-9 taj se niz riječi u danoj gramatici može izvesti iz dva niza produkcija, pa mu pripadaju i dva parsna stabla, koja su u primjeru označena crvenom bojom. U prvom parsnom stablu riječ "Ivana" dio je imeničnoga, odnosno subjektnoga skupa, pa predstavlja subjekt i stoga je imenica ženskoga roda u nominativu, dok je riječ "Petra" dio predikatnoga skupa, pa predstavlja objekt i stoga je imenica muškoga roda

[18] U primjeru su, zbog jednostavnosti prikaza, dane samo one rečenice u kojima su imenice ostvarene produkcijom NP → N različite. Budući da je njih osam, ima ih još osam u kojima su imenice jednake, primjerice, "Ivana brani Ivana" i "brani Petra Petra".

u akuzativu. To parsno stablo implicira uloge navedenih osoba u rečenici pa je u njoj "Ivana" osoba koja "brani Petra". U drugom parsnom stablu obrnut je redoslijed riječi u odnosu na intuiciju: "Ivana" je dio predikatnoga skupa, pa predstavlja objekt, odnosno imenicu muškoga roda u akuzativu, dok je "Petra" dio subjektnoga skupa i time imenica ženskoga roda u nominativu. Utoliko je u drugom parsanju "Petra" osoba koja "brani Ivana"[19].

S gledišta značenja sadržanoga u višeznačnim rečenicama iz primjera, odnosno implikacija koje njihova sintaktička struktura ima na semantičke uloge pojedinih elemenata te strukture, načelno se razlikuju dvije vrste formalne sintaktičke višeznačnosti (Grune i Jacobs 1991:62): prividna i prava višeznačnost. Prava višeznačnost je pritom ona sintaktička višeznačnost koja izravno utječe na semantičke uloge rečeničnih elemenata, dok je prividna višeznačnost ona koja se odražava samo na sintaktičku razinu, odnosno ne mijenja semantičke uloge višeznačnih elemenata višeznačnim sintaktičkim tumačenjem. U primjeru 2-9 rečenice povezane bojama (crveno-crveno i plavo-plavo) istinski su višeznačne jer isti niz ("Ivana brani Petra" i "Petra brani Ivana") prenosi različitu obavijest ovisno o pripadajućem parsnom stablu, odnosno sintaktičkome tumačenju.

Ovim je primjerom ilustriran postupak parsanja jednostavnih rečenica prirodnoga jezika tumačenjem zadanoga sintaktičkog formalizma. Ta ilustracija povlači dva pitanja.

Prvo je pitanje dijelom već uvedeno, a tiče se sintaktičke višeznačnosti: ukoliko dosljedna primjena zadanoga sintaktičkog formalizma – ovdje primjerom predstavljenog u obliku produkcija beskontekstne gramatike – pruža za neku rečenicu više od jednoga tumačenja, kako za tu rečenicu odabrati jedno od tih tumačenja kao točno? Ukratko, kako za svaku rečenicu pronaći sva moguća parsna stabla i kako potom svakoj rečenici ispravno dodijeliti samo jedno od njih?

Drugo pitanje vezano je uz složenost sintaktičkoga modela i utjecaj te složenosti na postupak parsanja. Beskontekstna gramatika iz primjera 2-9 sastoji se od samo osam različitih produkcija i može generirati samo šesnaest (oblikom ili parsnim stablom) različitih rečenica, pa se može reći da je vrlo jednostavna. Može se zamisliti, unutar paradigme beskontekstne gramatike i jezika, ali i izvan nje, sintaktičke modele puno veće složenosti, izrađene s ciljem objašnjavanja većega skupa sintaktičkih pojava u nekome prirodnom jeziku. U tome se slučaju postavlja pitanje složenosti postupka pronalaženja valjanoga parsnog stabla za neki

[19] Slično vrijedi i za parsna stabla označena plavom bojom, kao i za rečenice iz prethodne napomene.

proizvoljno odabrani niz riječi. Kako, dakle, učinkovito provjeriti je li neki niz riječi valjana rečenica modeliranoga jezika i, ako jest, kako joj učinkovito dodijeliti parsno stablo[20] i time je sintaktički analizirati?

Neki od mogućih odgovora na postavljena pitanja dani su dalje u tekstu, prvo unutar paradigme beskontekstne gramatike pa potom i izvan nje.

2.1.2.2 Parsanje beskontekstnih jezika

Povijesni razlozi – točnije, vremenska podudarnost razvoja teorije izračunljivosti i računalne znanosti s jedne strane (usp. Church 1936, Turing 1936, Turing 1938, Kleene 1952) te formalnih modela prirodnoga jezika s druge strane (usp. Bloomfield 1933, Harris 1951, Chomsky 1957), kao i uska vezanost razreda formalnih automata, formalnih gramatika i fomalnih jezika – učinili su upravo beskontekstnu gramatiku prvim sintaktičkim formalizmom za računalno parsanje prirodnoga jezika. Naime, formalizam beskontekstne gramatike (i postisnoga automata) pokazao se dovoljno ekspresivnim (usp. Chomsky 1959) da opiše veći dio sintaktičkih fenomena određenoga prirodnog jezika, a istovremeno s računalnoga gledišta dovoljno dobro strukturiranim da omogući učinkovite računalne izvedbe parsera. S druge strane, pokazano je da razred regularnih jezika ne pokriva dovoljan broj sintaktičkih jezičnih fenomena, ali je višestruko iskorišten u računalnom modeliranju morfologije prirodnoga jezika (usp. Karttunen i Beasley 2005), dok su se metode sintaktičke analize kontekstno-ovisnih jezika pokazale – uz izuzetak formalizama nastalih ograničavanjem kontekstno-ovisnih gramatika na produkcijski pojednostavljene njihove podskupove – (usp. Boullier 1995, Martin 2002) računalno neučinkovitima za stvarnu primjenu u računalnim sustavima (usp. Jurafsky i Martin 1999:492). Pritom se uglavnom smatra da je opis sintaktičke složenosti prirodnih jezika moguć u razredu "blago kontekstno-ovisne gramatike", s odvojenim fenomenima opisivima beskontekstnom gramatikom te fenomenima opisivima samo složenijim sintaktičkim formalizmima (usp. Savitch i dr. 1987, Kallmeyer 2010:23).

Budući da su spomenuti povijesni razlozi – posljedično, razvojem formalno-jezičnih formalizama i računalnih modela za njihovu obradu te primjenom tih formalizama i modela na opis i obradbu prirodnih jezika – značajno utjecali i na razvoj parsera za prirodne jezike,

[20] Ovdje treba primijetiti da se misli na čitavo parsno stablo za neku rečenicu. Postoje i parseri koji ne izvode traženje čitavoga parsnoga stabla, već samo nekog njegova dijela, odnosno ne parsaju sve do razine listova, nego s postupkom staju na nekoj ranijoj razini stabla; o njima se kratko raspravlja naknadno.

dalje se prikazuju neki pristupi parsanju beskontekstnih jezika koji su od posebnoga značaja s gledišta primjene u parsanju prirodnih jezika, pa potom i sami postupci primjene[21].

2.1.2.2.1 Razredba parsera beskontekstnih jezika

Parseri beskontekstnih jezika načelno se razvrstavaju u razrede prema tri različita kriterija, odnosno tri gledišta na pristup ulaznomu nizu i formalnoj gramatici.

1. Prema načinu sučeljavanja ulaznoga niza riječi s produkcijama zadane formalne gramatike razlikuje se silazno parsanje (usp. Dovedan 2003), odnosno parsanje od vrha prema dnu parsnoga stabla (en. *top-down parsing*) i uzlazno parsanje, odnosno parsanje od dna prema vrhu parsnoga stabla (en. *bottom-up parsing*).

   Kod silaznoga parsanja nastoji se oponašati postupak izvođenja formalnom gramatikom od početnoga nezavršnog simbola prema listovima parsnoga stabla, pa se stoga – budući da ide od vrha parsnoga stabla prema njegovu dnu[22] – takva analiza i naziva silaznom.

   Obrnuto, uzlazno parsanje nastoji rekonstruirati postupak izvođenja od posljednje produkcije unatrag, odnosno od listova parsnoga stabla pa do početnoga nezavršnog simbola. Uz ova dva pristupa, postoji i hibridni pristup koji se naziva *parsanje iz lijevoga kuta* (en. *left-corner parsing*). U tome pristupu svaka se desna strana produkcije gramatike dijeli na lijevi i desni dio, pa se lijevi dio raščlanjuje uzlaznom, a potom desni dio silaznom analizom (usp. Grune i Jacobs 1991:79).
2. Prema načinu čitanja ulazne rečenice razlikuje se usmjereno parsanje, odnosno čitanje niza riječi s lijeva na desno (en. *directional parsing*) i neusmjereno parsanje, odnosno čitanje niza riječi nekim drugim slijedom koji pogoduje izvršavanju algoritma (en. *non-directional parsing*).

   Kod neusmjerenoga parsanja način pristupanja ulaznomu nizu podređen je cilju izgradnje parsnoga stabla. Čitava ulazna rečenica stoga mora biti u memoriji parsera kako bi on njezinim riječima mogao pristupati neograničenim redoslijedom, odnosno neusmjereno. Postoji i silazna i uzlazna inačica neusmjerenoga parsanja.

[21] Prije toga, vrijedi napomenuti kako je kontradikcijom dokazano da je gramatikama tipa 0 nemoguće parsati, odnosno da je nemoguće izraditi algoritam koji kao ulaz prima bilo koju gramatiku tipa 0 i bilo koji niz znakova i u konačnome vremenu određuje pripada li zadani niz toj gramatici (usp. Grune i Jacobs 1991:70). Također, pokazano je da je – iako je parser za njih teorijski uvijek izradiv – parsanje gramatikama tipa 1 (kontekstno-ovisnima) nužno računalno neučinkovito s gledišta vremenske složenosti (usp. Grune i Jacobs 1991:72). Postoje i iznimke od ovoga pravila, stvorene uvođenjem ograničenja na produkcije gramatika tipova 0 i 1.

[22] Kažu (Grune i Jacobs 1991:64) da u računalnoj znanosti "stabla rastu od njihovih korijena prema dolje".

Tipični predstavnik skupine silaznih neusmjerenih parsera opisan je u (Unger 1968) i najčešće se naziva Ungerovim parserom (usp. Grune i Jacobs 1991:75), dok se kao primjer skupine uzlaznih neusmjerenih parsera najčešće uzima takozvani parser CYK (Sakai 1962, Younger 1967, Kasami i Torii 1969, Cocke i Schwartz 1970). Parser CYK uobičajen je u literaturi za ilustraciju parsanja beskontekstnih jezika budući da je njegova izvedba nad beskontekstnim gramatikama lako objašnjiva i prikaziva, a usto je računalno učinkovitiji[23] od Ungerova parsera.

Usmjereni parseri čitaju ulazni niz riječi redom, s lijeva na desno[24], pa stoga za njihov rad nije potrebno čitavu rečenicu pohraniti u memoriju. Sve poznate metode usmjerenoga parsanja modeliraju se i izvode u obliku parsnih automata. S obzirom na formalnu gramatiku, također postoji uzlazna i silazna inačica. Ukoliko usmjereni parser izvodi silazno parsanje, naziva ga se automatom koji predviđa i uspoređuje (en. *prediction/match parser*), a ukoliko parsa uzlazno, naziva ga se automatom koji izvodi pomak i redukciju (en. *shift/reduce parser*). Tipični predstavnici pojedinih skupina dani su u idućoj podjeli budući da su pojedine izvedbe usmjerenih parsera usko vezane uz način pretraživanja.

3. Prema načinu pretraživanja skupa produkcija zadane gramatike u potrazi za objašnjenjem ulaznoga niza razlikuje se pretraživanje po dubini (en. *depth first search*) od pretraživanja po širini (en. *breadth first search*).

   Metode pretraživanja detaljno su prikazane u (Cormen i dr. 2009) budući da se radi o generičkim metodama pretraživanja grafova, odnosno tehnikama dinamičkoga programiranja koje se koriste u brojnim algoritmima, posebno u teoriji grafova i za stablaste strukture. U skladu s nazivom, kod pretraživanja po dubini traži se neka optimalna putanja tako da se stablo ispituje od korijenskoga čvora do prvoga lista s njegove lijeve ili desne strane. Pri svakome grananju pamti se točka grananja, pa se algoritam po dosezanju prvoga lista vraća natrag (en. *backtracking*) na posljednju nađenu točku grananja te nju i ostale dalje ispituje slijedeći isto načelo. Kod pretraživanja po širini ispituju se svi susjedni čvorovi korijenskoga čvora pa potom svi susjedni čvorovi za svaki susjedni čvor korijenskoga čvora i istim načelom do dosezanja svih listova stabla.

[23] Barem u njihovome izvornome obliku budući da se oba algoritma mogu prilagoditi do identične izvedbene učinkovitosti (usp. Grune i Jacobs 1991:75).

[24] Ponekad također i obrnuto, s desna na lijevo, budući da usporedno izvođenje oba postupka, odnosno parsanje u oba smjera pokazuje izvjesne korisnosti.

Primjerice, silazni parseri s pretraživanjem po dubini nazivaju se parserima s tehnikom rekurzivnoga spusta (en. *recursive descent parsers*), a uzlazni parseri s pretraživanjem po širini najčešće pripadaju skupini Earleyjevih parsera, prema (Earley 1970). Detaljniji prikaz ostalih metoda, koje su od manje važnosti za ovo istraživanje, dan je u (Grune i Jacobs 1991).

Uz iznesenu razredbu, vrijedi napomenuti kako većina algoritama za parsanje, danih za primjere pojedinih skupina prema odabranim dimenzijama, nije računalno učinkovita, posebno s gledišta parsanja velikih ulaznih nizova riječi, karakterističnih za jezike za programiranje[25]. Za parsanje s takvim posebnim zahtjevima u pravilu se do neke mjere ograničavaju produkcije beskontekstne gramatike, a time i rezultirajući formalni jezici[26] kako bi se mogli za njih izraditi učinkovitiji parseri, poput LL(k), LR(k) i sličnih parsera, koji se ovdje dalje ne opisuju pobliže (usp. Grune i Jacobs 1991:78).

Pregled parsera prema navedenim dimenzijama – načinu rekonstrukcije izvođenja produkcija, pristupu ulaznomu nizu te načinu pretraživanja produkcijske stablaste strukture – dan je u tablici 2-1.

Tablica 2-1 Pregled pristupa parsanju beskontekstnih jezika

	silazno parsanje	**uzlazno parsanje**
neusmjereno parsanje	Ungerov parser	parser CYK
usmjereno parsanje	automat za predviđanje i usporedbu rekurzivni spust	automat za pomak i redukciju Earleyjev parser
linearno usmjereno parsanje s pretraživanjem po širini	LL(k) parser	LR(k), LALR(1), itd.
učinkovito (nelinearno) usmjereno parsanje s pretraživanjem po širini	/	Tomitin parser (Tomita 1986)

[25] Jedna prosječna rečenica hrvatskoga jezika ima otprilike 25 riječi (usp. Tadić 2009), a jednom "rečenicom" nekoga programskoga jezika može se smatrati čitav jedan program proizvoljne složenosti.
[26] Takva odluka stoga može biti valjana za izgradnju parsera programskih, ali ne i prirodnih jezika budući da je formalizam beskontekstne gramatike ionako nedovoljno ekspresivan za opisivanje sintaktičkih fenomena prirodnih jezika.

S obzirom na primjene u parsanju prirodnoga jezika, načelnu jednostavnost i ilustrativnost te uvođenje bitnih koncepata, dalje se dodatno pojašnjava jedan neusmjereni uzlazni parser (CYK) i jedan usmjereni uzlazni parser (Earleyjev parser), kao primjer automata za pomak i redukciju. Prikazuje se njihova izvedba i računalna učinkovitost, a oba se koriste za izgradnju konceptualnoga modela parsera beskontekstnih jezika koja će se potom prenijeti i na model parsera prirodnoga jezika.

2.1.2.2.2 Parser CYK

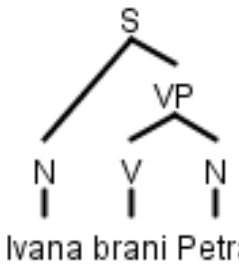
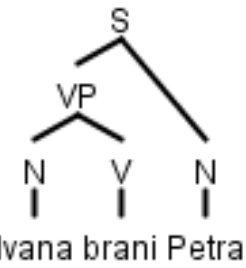
Parser CYK je računalni algoritam za parsanje koji kao ulaz uzima beskontekstnu gramatiku i jedan niz riječi, odnosno jednu rečenicu-kandidat za koju treba odrediti parsno stablo (ili parsna stabla) prema zadanoj gramatici. Za potrebe parsera CYK ulazna gramatika mora biti u *Chomskyjevu normalnom obliku* (en. *Chomsky normal form*, CNF).

Svaka produkcija beskontekstne gramatike u Chomskyjevu normalnom obliku oblika je $A \rightarrow BC|a$ ili $S \rightarrow \varepsilon, A, B, S \in N, a \in \Sigma$. Dakle, svaki se nezavršni simbol može preslikati u dva nezavršna simbola ili jedan završni simbol, a samo se početni nezavršni simbol može preslikati u prazan simbol. Svaka beskontekstna gramatika može se algoritamski prepisati u Chomskyjev normalni oblik[27]. Gramatika u Chomskyjevu normalnom obliku produkcijama može opisati samo binarna stabla (ona kod kojih se u svakom grananju stvaraju nužno dvije nove grane, osim ako se radi o listu stabla). Usporedbe radi, gramatika iz primjera 2-6 nije u Chomskyjevu normalnom obliku (zbog produkcije $\text{Likovi} \rightarrow \text{Likovi}\ ,\ \text{Lik}$), što je vidljivo i iz pripadajućega parsnoga stabla iz primjera 2-8, kao ni gramatika iz primjera 2-9, budući da produkcija $NP \rightarrow N$ ne udovoljava pravilu o preslikavanju u nezavršne simbole.

Prije formalizacije algoritma, ovdje se metoda rada parsera CYK prikazuje s pomoću formalne gramatike i rečenice iz primjera 2-9. Tu je gramatiku potrebno najprije prebaciti u Chomskyjev normalni oblik. Postoji učinkovit algoritam za prebacivanje gramatika u Chomskyjev normalni oblik (usp. Hopcroft i dr. 2006:272), no ovdje je, zbog jednostavnosti gramatike iz primjera 2-9, moguće to prebacivanje obaviti i ručno, ukidanjem jedinične produkcije $NP \rightarrow N$ i zamjenom svakoga nezavršnog simbola NP simbolom N. Nova gramatika prikazana je na primjeru 2-10, uz ponovljeni primjer sintaktičke višeznačnosti rečenice "Ivana brani Petra". U odnosu na parsna stabla iz primjera 2-9, ukinuti su svi čvorovi

[27] Prebacivanje beskontekstne gramatike u Chomskyjev normalni oblik nužno povećava broj produkcija što može uzrokovati probleme algoritmima čije vrijeme izvođenja i memorijski zahtjevi ovise o tome broju (usp. Lange i Leiß 2009).

NP, ali je sintaktička struktura rečenice zadržana, utoliko što implicirana semantika ostaje ista – osobne imenice "Ivana" i "Petra" predstavljaju subjekt, odnosno objekt rečenice, ovisno o produkciji iz početnoga nezavršnog simbola (S → N VP ili S → VP N).

$S \rightarrow N\ VP \mid VP\ N;\ VP \rightarrow V\ N \mid N\ V;\ V \rightarrow brani;\ N \rightarrow Ivana \mid Petra$	
	

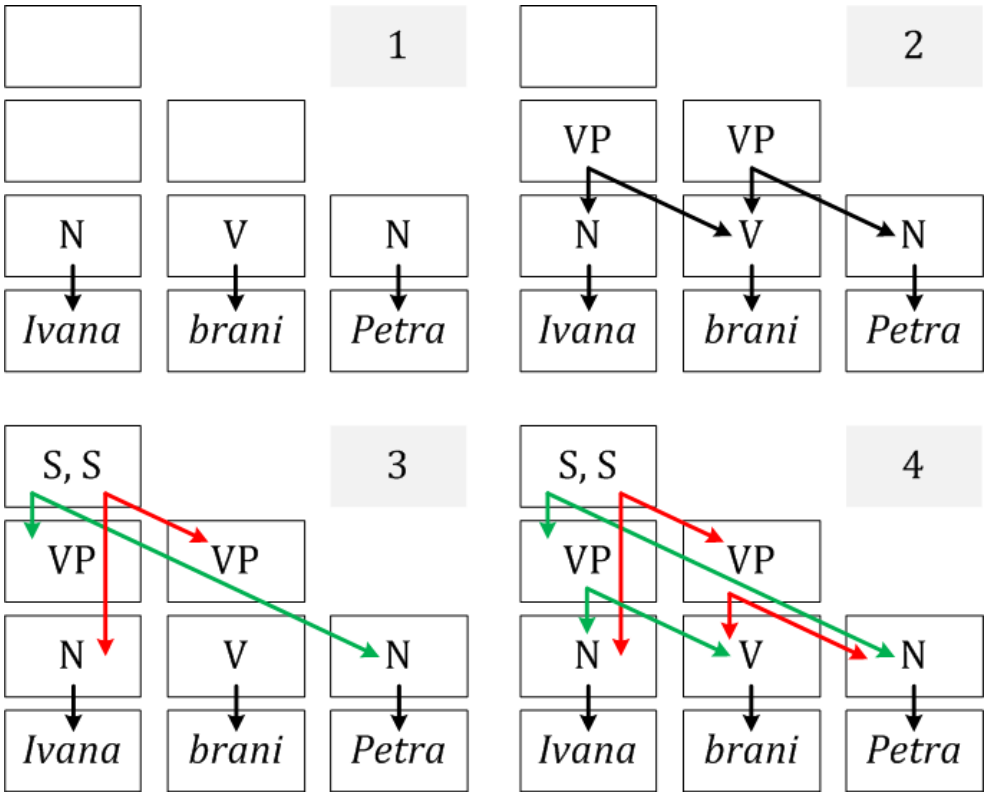
Primjer 2-10 Chomskyjev normalni oblik i sintaktička višeznačnost

Algoritam CYK neusmjereno pristupa ulaznomu nizu riječi, a izvršava uzlazno parsanje. Čitava rečenica stoga mora biti učitana u njegovu memoriju, kao polazna točka analize. Rad algoritma najčešće se prikazuje s pomoću dvodimenzinalne kvadratne matrice (donje trokutaste) čije su dimenzije određene duljinom ulaznoga niza, odnosno brojem riječi u rečenici. Jedan takav prikaz[28] njegova rada dan je u primjeru 2-11.

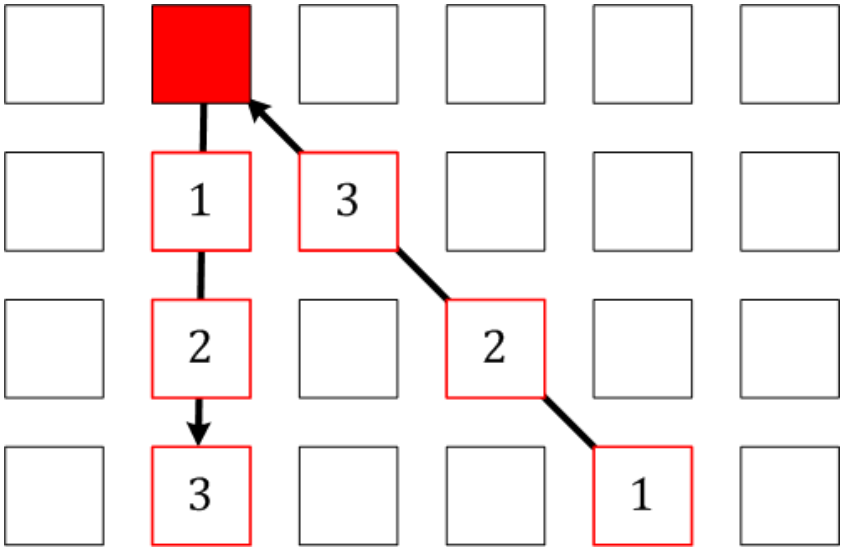
U primjeru je rad algoritma grafički opisan u četiri koraka. Nad ulaznim se nizom riječi gradi donja trokutasta matrica prema njegovoj duljini, pa u ovome slučaju ta matrica ima tri retka i tri stupca. Svaki od redaka, od zadnjega (koji ima tri elementa) prema prvome (koji ima samo jedan element) predstavljaju razine sintaktičkoga predstavljanja rečenice. Na posljednjoj razini tumače se sintaktički, s pomoću produkcija zadane formalne gramatike, listovi parsnoga stabla, odnosno riječi rečenice, a svaka iduća razina – sve do posljednje, koja uvijek predstavlja početni nezavršni simbol gramatike – predstavlja jednu razinu parsnoga stabla od listova prema korijenskome čvoru.

Algoritam s radom započinje u prvom elementu posljednjega retka svorene trokutaste matrice. Na toj posljednjoj razini svaka od riječi rečenice pokušava se objasniti s pomoću onoga podskupa produkcija zadane gramatike koji sadrži samo one produkcije kojima desna strana sadržava isključivo završne simbole, odnosno riječi.

[28] Pogledati također interaktivnu ilustraciju algoritma na http://www.diotavelli.net/people/void/demos/cky.html (2012-02-14).



Primjer 2-11 Ilustracija rada algoritma CYK



Slika 2-3 Uparivanje lijevih i desnih strana produkcija u algoritmu CYK

Na sve iduće razine, odnosno elemente matrice primjenjuje se isto algoritamsko pravilo koje se može kolokvijalno nazvati pravilom vertikalnoga spusta i dijagonalnoga uspona. Naime, u idućim elementima, odnosno na idućim razinama matrice traže se lijeve strane produkcija koje odgovaraju desnima, predstavljenima u svim redcima ispod trenutnoga. Budući da je zadana gramatika u Chomskyjevu normalnom obliku, uparivanje lijevih i desnih

strana produkcija svodi se na provjeravanje parova elemenata u nižim redcima te pokušaj njihova objašnjavanja nekima od produkcija iz zadane gramatike. Princip provjeravanja parova elemenata prikazan je na slici 2-3. Prema toj slici za traženu lijevu stranu produkcije – prikazanu crveno obojenim elementom matrice – desne se strane, odnosno parovi nezavršnih simbola provjeravaju tako da se uparuju redom elementi iz istoga stupca s elementima na istoj dijagonali u svim redcima ispod trenutnoga, isključujući redak u kojemu su opisane same riječi. Nadalje, uparivanje elemenata stupca i dijagonale odvija se obrnutim redoslijedom: element stupca koji je najbliži onomu koji predstavlja lijevu stranu produkcije – dakle, onaj u istome stupcu, ali jedan redak ispod – uparuje se s elementom dijagonale koji je najdalje od onoga koji predstavlja lijevu stranu produkcije – dakle, s onim dijagonalnim elementom u predzadnjem retku matrice. Uparivanje se dalje vrši s jediničnim pomakom, kako je prikazano brojevima koji identificiraju parove nezavršnih simbola na slici 2-3. Ukoliko se pronađe neka lijeva strana produkcije koja izvođenjem daje neki od algoritamski uparenih nezavršnih simbola, lijeva se strana produkcije zapisuje u aktivnu ćeliju matrice. Ukoliko je više pronađenih kandidata, u element matrice se upisuju svi, a ukoliko kandidata nema, algoritam završava i ulazni se niz riječi proglašava nevaljanom rečenicom jezika opisanoga zadanom formalnom gramatikom. Ukoliko rečenica s ulaza pripada danoj gramatici, algoritam završava u prvome retku i stupcu trokutaste matrice, odnosno prvom njezinom elementu te u njega upisuje početni nezavršni simbol koji se dobiva istim postupkom uparivanja nezavršnih simbola.

Parsno stablo rekonstruira se iz rada algoritma CYK, odnosno iz traga ostavljenoga uparivanjima lijevih i desnih strana produkcija – svako uparivanje stvara jednu roditeljsku granu stabla i dvije njoj podređene grane na razini ispod. Rekonstrukcija parsnoga stabla prikazana je primjerom 2-11, gdje je jedno moguće parsno stablo označeno zelenom bojom, a drugo parsno stablo crvenom bojom (osim izvođenja listova iz nevišeznačnih produkcija, koje je označeno crnom bojom). Dva parsna stabla iz primjera 2-11 očekivano su identična onima iz primjera 2-10 budući da je primjer 2-11 ilustracija rada parsera CYK za danu rečenicu, a također su praktički jednaka onima iz primjera 2-9, s obzirom na jednostavnu prilagodbu tamo prikazane gramatike u skladu s Chomskyjevim normalnim oblikom. Pojednostavljeno, ovim se algoritmom ulazni niz parsa parsanjem podnizova ulaznoga niza riječi, od jedinične duljine podniza pa do čitave rečenice. Parser CYK prikazan je pseudokodom kao algoritam 2-1. Izračunska složenost algoritma 2-1, odnosno vrijeme potrebno za parsanje ulazne rečenice algoritmom CYK u njegovu izvornom obliku, prema asimptotskome zapisu (usp. Cormen i

dr. 2009:43), iznosi u najgorem slučaju $\Theta(n^3 \cdot |G|)$, gdje n predstavlja duljinu ulaznoga niza, odnosno broj riječi u rečenici, a $|G|$ broj produkcija gramatike G. Stoga se kaže da je parser CYK algoritam polinomske složenosti[29].

neka je ulazna rečenica $S = (w_1, \ldots, w_n), 1 \leq i \leq n, w_i \in \mathcal{V}$
neka ulazna gramatika G sadrži r nezavršnih simbola R_1, ..., R_r.
neka je P[n, n, r] matrica istinosnih vrijednosti i neka je početno $\forall i, p_i \in P, p_i = 0$
za svaki $i = 1 \ldots n$
    za svaku jediničnu produkciju $R_j \rightarrow a_i$
        neka je $P[i, 1, j] = 1$
za svaki $i = 2 \ldots n$
    za svaki $j = 1 \ldots n - i + 1$
        za svaki $k = 1 \ldots i - 1$
            za svaku produkciju $R_A \rightarrow R_B R_C$
                ako su P[j, k, B] i $P[j + k, i - k, C] = 1$, onda $P[j, i, A] = 1$
ako $\exists x, 1 \leq x \leq r, P[1, n, x] = 1$ onda je $S \in L(G)$
u protivnom, $S \notin L(G)$

Algoritam 2-1 Pseudokod parsera CYK

2.1.2.2.3 Earleyjev parser

Earleyjev parser (Earley 1970) je usmjereni, uzlazni parser s pretraživanjem po širini[30] koji vrši operacije pomaka i redukcije nad ulaznim nizom riječi te ne zahtijeva da ulazna beskontekstna gramatika bude u Chomskyjevu normalnom obliku. Poput parsera CYK, Earleyjev parser također koristi zadanu formalnu gramatiku za dohvat nezavršnih simbola iz podnizova završnih simbola, odnosno podskupova niza riječi koji predstavlja ulaznu rečenicu. Međutim, umjesto primjene produkcija na nizove riječi rastuće duljine, Earleyjev parser čita riječi s ulaza redom, jednu po jednu riječ, pa primjenjuje za tu riječ svaku produkciju gramatike koja i dalje barem djelomično objašnjava trenutno pročitani niz riječi. Naime, ovaj parser za svaku pročitanu riječ pohranjuje u popis sve produkcije gramatike koje se za tu riječ

[29] Postoje prilagodbe algoritma CYK, koje mu umanjuju složenost na $\Theta\left(n^{3-\frac{\varepsilon}{3}}\right)$, za neki $\varepsilon \in \mathbb{R}_0^+$ (usp. Lee 2002).
[30] Preciznije, Earleyjev parser je zapravo "algoritam koji obavlja silaznu analizu s uzlaznim filtriranjem" (usp. Jurafsky i Martin 1999:354).

barem djelomično ostvaruju, odnosno imaju potencijal ostvarivanja, od početne produkcije pa do jediničnih preslikavanja u završne simbole. Po čitanju iduće riječi, koje se naziva i pomakom parsera (en. *shift*), promatra se ostvaruje li ona dodatno neku od ranije djelomično ostvarenih produkcija. Ukoliko se jedna ili više produkcija na ovaj način u potpunosti ostvari, njezina se desna strana briše i na njezino se mjesto stavlja nezavršni simbol s njezine lijeve strane. Potom se ranije opisani postupak ponavlja tako da se preostala nedovršena pravila sada provjeravaju za upisani nezavršni simbol koji i njih potencijalno dovršava. Ovim se postupkom produkcije, odnosno njihovi popisi reduciraju ili umanjuju (en. *reduce*). Ako algoritam pročita sve riječi iz ulazne rečenice, svaki niz nezavršnih simbola koji objašnjava čitav niz ulaznih riječi predstavlja valjano parsno stablo za ulaznu rečenicu.

Asimptotska vremenska složenost Earleyjeva algoritma je $O(n^3)$ u najgorem slučaju, no – ovisno o svojstvima, odnosno mogućim pojednostavljenjima općenito neograničene ulazne beskontekstne gramatike – može također raditi u kvadratnom ($O(n^2)$, za nevišeznačne gramatike) ili čak linearnom vremenu ($O(n)$, za gramatike koje se može također parsati i LL(k) i sličnim parserima). Može se stoga reći da je Earleyjev parser generički, utoliko što može parsati bilo kojom beskontekstnom gramatikom.

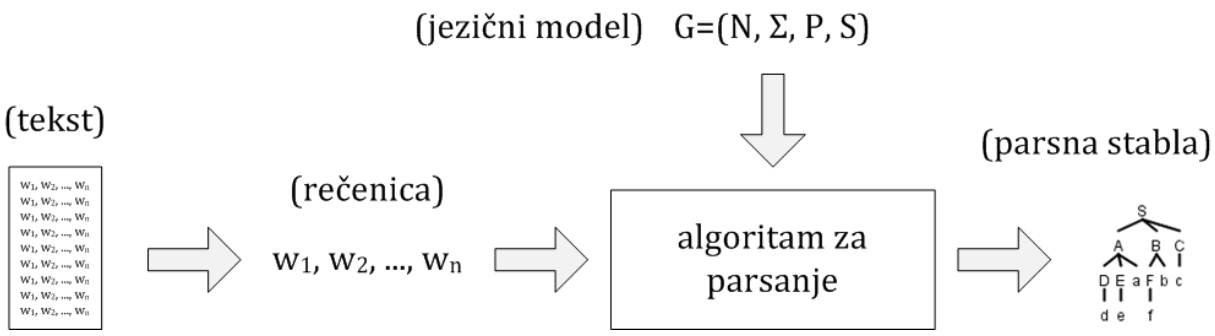
2.1.2.3 Formalne metode i parsanje prirodnoga jezika

Oba prikazana parsera koji za ulaz primaju neku beskontekstnu gramatiku i rečenicu-kandidat – parser CYK i Earleyjev parser – pripadaju skupini tabličnih parsera (en. *chart parsers*, usp. Kay 1986)[31] budući da parsaju pohranjujući implicitno, u tabličnome obliku, parcijalna parsna stabla rastućih podskupova riječi ulazne rečenice. Također, oba parsera mogu se konceptualno svesti na isti model parsanja u kojem parsni algoritam prima dva parametra na ulazu – beskontekstnu gramatiku i niz riječi, a na izlazu daje nijedno, jedno ili više parsnih stabala izvedenih iz ulazne gramatike nad ulaznim nizom riječi. Izostanak parsnoga stabla pritom implicira negramatičnost rečenice, a prisutnost više od jednoga parsnog stabla njezinu sintaktičku višeznačnost. Takav jednostavni model parsera kao "crne kutije", odnosno funkcije s određenim brojem i tipovima ulaznih i izlaznih parametara prikazan je slikom 2-4.

[31] Vrijedi primijetiti vremenski odmak od izuma Earleyjevog parsera (Earley 1970, odnosno Earley 1968) do prikaza tabličnoga parsanja u (Kay 1986) – potonji prikaz odnosi se na obradbu prirodnih jezika, dok je Earleyjev parser zamišljen isključivo kao parser beskontekstnom gramatikom, neovisno o primjeni.

Ovaj općeniti model parsera sačinjavaju tri elementa: algoritam za parsanje, jezični model – odnosno sintaktički model nekoga formalnog (beskontekstnog), a moguće i prirodnog jezika – koji parser zahtijeva za rad te modul za predobradbu ulaznoga teksta koji tekst pretvara u skup rečenica koje potom daje parseru na obradbu. Model je općenit, pa se može primijeniti, kao ranije, na beskontekstne jezike, ali i na prirodne jezike. Kod primjene na beskontekstne jezike jezični model je neka beskontekstna gramatika s pripadajućim skupom nezavršnih i završnih simbola te skupom pravila izvođenja, a za parsni algoritam bira se neki od ranije prikazanih algoritama, ovisno o svojstvima dane beskontekstne gramatike. Može se, primjerice, sa sigurnošću za parsni algoritam odabrati Earleyjev parser budući da on zna bez ograničenja parsati beskontekstnom gramatikom. Modul za predobradbu ulaznoga teksta u skup rečenica ostaje nejasan budući da opća definicija rečenice formalnoga jezika, osim kao niza nezavršnih simbola, odnosno riječi, ne postoji. Utoliko se taj modul može smatrati nepostojećim ili trivijalnim, pa se čitav ulazni tekst smatra jednim nizom riječi, odnosno jednom potencijalnom rečenicom. Valja također primijetiti kako je parsni algoritam usko vezan uz sintaktički formalizam: parseri se modeliraju i izvode tako da mogu parsati prema danome sintaktičkom formalizmu, pa je tako parser CYK bio vezan uz beskontekstne gramatike u Chomskyjevu normalnom obliku, a Earleyjev parser uz neograničene beskontekstne gramatike.

Ako se želi prikazani model parsera primijeniti za parsanje prirodnoga jezika, potrebno je preciznije opisati svojstva modula za predobradbu te, s posebnim naglaskom, pristupe sintaktičkom modeliranju jezika za potrebe parsanja i pristupe izradi algoritama za parsanje tim sintaktičkim modelima. U preostalome dijelu ovoga poglavlja raspravlja se upravo o različitim računalno primjenjivim sintaktičkim modelima prirodnoga jezika te pripadajućim algoritmima za parsanje prirodnoga jezika.



Slika 2-4 Konceptualni model parsera

2.1.2.3.1 Parsanje prirodnoga jezika

Kao i kod parsanja formalnih jezika – gdje je povijesna podudarnost razvoja teorije formalnih jezika i teorije izračunljivosti, odnosno računalne znanosti utjecala na pristupe parsanju formalnih jezika – tako je i na parsanje prirodnih jezika utjecala podudarnost u razvoju pristupa parsanju formalnih jezika te čitavoga područja umjetne inteligencije koje, kako je ranije spomenuto, kao bitan element izrade inteligentnih računalnih sustava uključuje i računalno modeliranje razumijevanja prirodnoga jezika. Kronološki se, naime, uvelike podudaraju prva sustavna znanstvena razmatranja fenomena prirodnih jezika prema razinama lingvističkoga opisa unutar okvira teorije formalnih jezika (Chomsky 1957, Chomsky 1959, usp. Jurafsky i Martin 1999:346, Jurafsky i Martin 1999:349), pojava prvih računalno učinkovitih modela parsera formalnih jezika (poput ranije predstavljenih parsera koji u najgorem slučaju parsaju u kubnom, odnosno polinomskom vremenu) te pojava prvih zahtjeva za modeliranjem i razumijevanjem prirodnih jezika od strane strojeva koje se želi nazivati inteligentnima, bilo za modeliranje ljudskoga ponašanja (usp. Turing 1950) ili za obavljanje zadataka koje učinkovito obavljaju ljudi (usp. Russel i Norvig 2009:2). Tim logičnim slijedom misli i događanja, očekivano, prvi parseri prirodnoga jezika bili su zapravo ranije prikazani parseri beskontekstnom gramatikom.

Za potrebe parsanja prirodnoga jezika beskontekstnom gramatikom pred potencijalne parsere postavljaju se isti opći zahtjevi kao kod parsanja formalnoga jezika. Naime, jezik je u toj paradigmi modeliranja i obradbe definiran isključivo formalnom gramatikom, pa se utoliko i prirodni jezik u njoj beziznimno tretira kao formalni. S gledišta modela prikazanoga slikom 2-4, ulazna beskontekstna gramatika predstavlja model jezika čije se (potencijalne) rečenice parsaju, a time predstavlja i jedinu dodirnu točku između modela parsera i pojavnosti jezika za koji se modelira i izvodi parser. Primjerice, može se zamisliti scenarij u kojem postoji beskontekstna gramatika u kojoj su – alatom nezavršnih i završnih simbola te produkcija u propisanome obliku – zapisane i potpuno objašnjene sve pojave vezane uz sintaksu rečenica hrvatskoga jezika. Ta se beskontekstna gramatika može dati kao ulazni parametar nekom parseru beskontekstnom gramatikom, primjerice, parseru CYK ili Earleyjevu parseru, zajedno s rečenicom koju se želi parsati. U polinomskom vremenu parser će kao rezultat obradbe dati neki broj parsnih stabala te rečenice u skladu sa zadanom gramatikom i time ujedno potvrditi njezinu gramatičnost ili će je, s druge strane, opovrgnuti izostankom parsnoga stabla na izlazu. U tim okvirima problem parsanja prirodnoga jezika

svodi se na problem izrade što boljega modela toga prirodnog jezika u obliku beskontekstne gramatike. Dakle, osnovni pristup parsanju prirodnoga jezika onaj je u kojem se sintaksa prirodnoga jezika tretira generičkim izračunskim aparatom iz teorije formalnih jezika, uz pretpostavku da se sve sintaktičke pojave (ili zadovoljavajuće velik njihov podskup, s obzirom na željenu primjenu) mogu opisati pomoću formalizma beskontekstne gramatike.

2.1.2.3.2 Parsanje gramatikom

Kao ranije, kod prikaza parsanja beskontekstnom gramatikom, parsanje prirodnoga jezika opisivanjem sintaktičkih fenomena toga jezika unutar nekoga formalnogramatičkog formalizma (ovdje se samo ilustrativno, a i zbog prikaza povijesnih podudarnosti, koristi formalizam beskontekstne gramatike) i primjenom parsnih algoritama koji pripadaju tomu formalizmu naziva se *parsanje gramatikom* (usp. Nivre 2006:13). Taj naziv implicira središnju ulogu formalne gramatike u postupku parsanja: ako je zadana formalna gramatika G, prirodni jezik L(G) koji se parsa u potpunosti je definiran gramatikom G. Štoviše, može se reći i da se tako ne parsa neki prirodni jezik, nego se provjerava jesu li ulazne rečenice usklađene s danim sintaktičkim formalizmom G, odnosno jesu li ulazne rečenice u skupu L(G). Ranije u tekstu raspravljalo se (usp. Kallmeyer 2010) o tome da gramatički formalizmi prema Chomskyjevoj hijerarhiji – od regularnih do kontekstno-ovisnih gramatika – posjeduju određene razine opisne moći koje su prikladne za opis prirodnih jezika na nekim razinama jezičnoga opisa. Također, zbog složenosti prirodnoga jezika na svim razinama jezičnoga opisa, uzrokovane prije svega jezičnom višeznačnošću i njezinom ulogom u osiguranju učinkovitosti i robustnosti postupka razmjene obavijesti, svaki sintaktički model ostvaren nekom formalnom gramatikom nužno je restriktivan[32] u odnosu na stvarnu pojavnost jezika u pismu i govoru. Jednostavnije, model jezika u obliku neke formalne gramatike G, neovisno o odabranome gramatičkom formalizmu, precizno definira upravo jezik L(G) koji nužno predstavlja pravi podskup prirodnoga jezika koji se nastoji opisati. Dakle, modeliranje i parsanje prirodnoga jezika formalnom gramatikom nužno je ograničavajuće budući da se parsna stabla mogu nekim algoritmom učinkovito dohvatiti isključivo za rečenice jezika L(G), dok će sve preostale rečenice, izostavljene nepotpunošću sintaktičkoga modela, biti ocijenjene negramatičnima. Može se također zamisliti i neke slučajeve u kojima ulazna rečenica nije u

[32] Primjerice, u (Nivre 2006:21), izričito se tvrdi kako su "pristupi parsanju prirodnoga jezika parsanjem gramatikom redom zasnovani na pretpostavci kako se može izraditi formalna gramatika koja predstavlja vjerodostojan opis ili dovoljno dobru aproksimaciju ciljanoga prirodnoga jezika" te da je, u praksi, očigledno kako "većina, ako ne i sve do danas razvijene formalne gramatike ne uspijevaju doseći taj cilj" jer skup L(G) u pravilu predstavlja "vrlo restriktivan podskup ciljanoga prirodnoga jezika".

skupu L(G), a nije ni u potpunosti valjana rečenica parsanoga prirodnog jezika, no moglo bi joj se svejedno dodijeliti smisleno sintaktičko tumačenje.

Vezano uz nepodudarnost pojavnosti prirodnoga jezika i njegova modela, razlikuju se načelno dva problema parsanja prirodnoga jezika formalnom gramatikom (usp. Nivre 2006:21): *problem pokrivenosti* i *problem robustnosti*. Robustnost je (usp. Nivre 2006:21) "sposobnost parsera da parsa bilo koji ulazni niz riječi", odnosno da za bilo koji niz riječi pokuša pružiti smislenu sintaktičku analizu. Parseri se stoga mogu uspoređivati prema robustnosti, pa će se "jedan parser smatrati robusnijim od drugoga ukoliko može parsati više ulaznoga teksta bez prekidanja rada". S obzirom na ranije iskazanu prirodu parsanja formalnom gramatikom, robustnost u tim okvirima očigledno predstavlja bitan problem. Robustnost parsanja formalnom gramatikom usko je vezana uz pokrivenost tom formalnom gramatikom, odnosno udio nekoga stvarnog prirodnog jezika koji je njome uspješno opisan. Načelno postoje dva pristupa (usp. Samuelsson i Wirén 2000) rješavanju ovih problema: izrada manje ograničavajućih gramatičkih pravila te izrada pravila za opis nekoga dohvatljivog – odnosno iz nekoga vanjskog razloga korisnog ili zanimljivog – podskupa sintaktičkih fenomena prirodnoga jezika. Za prvi se pristup u pravilu pokazalo (usp. Nivre 2006:22) da uzrokuje problem značajnoga povećanja broja valjanih parsanja ulaznih nizova, odnosno da može učiniti gramatiku višeznačnom do razine beskorisnosti. Drugi pristup vodi nepotpunoj sintaktičkoj analizi, odnosno (usp. Abney 1996, Vučković 2009:36, Vučković 2009:62) *razdjeljivanju* i *plitkom parsanju*. Taj je pristup osobito detaljno istražen (usp. Nivre 2006:22) i sustavi koji mu podliježu u pravilu postižu mjerljivo dobre rezultate s gledišta brzine, robustnosti i učinkovitosti, žrtvujući pritom potpunost parsanja.

Uz probleme robustnosti i pokrivenosti, često se za parsanje formalnom gramatikom ističe i problem višeznačnosti (usp. Nivre 2006:23)[33], odnosno još ranije, primjerom 2-9 oslikana činjenica da formalne gramatike nude višestruka tumačenja sintaktičke strukture ulaznoga niza. U predstavljenim okvirima teorije formalnih jezika – odnosno bez pomoći nekoga izvanteorijskog modela ljudskoga pristupa razrješivanju jezične višeznačnosti na sintaktičkoj razini i njegove računalne izvedbe – nemoguće je riješiti problem višeznačnosti parsanja, odnosno odabrati jedno od ponuđenih parsnih stabala kao točno.

[33] I citat otamo, vezan uz prekomjerno generiranje formalnom gramatikom, koji kaže, u prijevodu s engleskoga jezika, da "sve gramatike cure" (en. *all grammars leak*).

S obzirom na raznolikost ocrtanih pristupa rješavanju problema parsanja formalnom gramatikom[34] te raznolikost samih rješenja koja potencijalno izlaze izvan okvira sintaktičke analize, kako je ona definirana ranije u ovome tekstu, potrebno je prije daljnje rasprave o parsanju odrediti kriterije, odnosno svojstva kojima promatrani pristupi parsanju moraju udovoljiti kako bi ih se dalje i detaljnije razmatralo.

2.1.2.3.3 Tražena svojstva parsera prirodnoga jezika

Prema definiciji parsera iz ovoga teksta, od njega se zahtijeva izvođenje sintaktičke analize prirodnoga jezika, a ovdje se – prema (Nivre 2006:41) i u skladu s postavljenom svrhom izrade parsera kao računalno-inteligentnoga alata s gledišta definicije umjetne i računalne inteligencije – pred tu sintaktičku analizu postavljaju četiri osnovna kriterija: *robustnost* (en. *robustness*), *razrješivanje višeznačnosti* (en. *disambiguation*), *točnost* (en. *accuracy*) i *učinkovitost* (en. *efficiency*).

2.1.2.3.3.1 Robustno razrješivanje višeznačnosti

Robustnost je kriterij koji uzima u obzir pojavnosti prirodnoga jezika izvan nekoga izravno usmjerenog jezičnog modela poput formalne gramatike. Prema (Chanod 2001, Nivre 2006:41), robustnost se odnosi na "razmatranje svih jezičnih konstrukcija koje ljudi zaista oblikuju u tekstovima prirodnoga jezika". Utoliko se potpuno robusnim parserom naziva onaj parser koji je u stanju bez prekida rada pružiti sintaktičko tumačenje svakoga ulaznog niza, neovisno o tome pripada li on željenomu prirodnom jeziku, nekomu drugom prirodnom jeziku ili uopće ne pripada prirodnomu jeziku. Smislenost parsanja u ovim drugim slučajevima postaje upitna, pa postavlja pred module za predobradbu teksta, prema slici 2-4, određene zahtjeve za pripremu ulaznoga teksta kako bi se osigurala smislenost utroška računalnih resurasa za parsanje toga teksta.

Parser P jezika L robustan je ako i samo ako za svaki tekst $T = (s_1, \dots, s_n), T \subseteq L$ svakoj rečenici $s_i \in T$ dodijeli barem jedno parsno stablo.

Iz definicije je vidljivo kako robustnost nije kriterij kojim se moglo samostalno vrjednovati neki parser budući da je apsolutnu robustnost lako postići: dovoljno je da parser slučajnim odabirom svakoj ulaznoj rečenici $s_i \in T$ dodijeli neko parsno stablo. Robustnost se stoga koristi kao zahtjev, odnosno preduvjet koji se ispituje prije svakoga drugog kriterija

[34] Ti pristupi detaljno su opisani u (Samuelsson i Wirén 2000) i (Carroll 2003).

vrjednovanja. Mogući drugi kriteriji robustnosti, koji se ovdje ne usvajaju i ne predstavljaju dodatno, uzimaju u obzir i kvalitetu ulaznih podataka, pa robustnost parsera promatraju kao funkciju (ne)kvalitete ulaznoga teksta.

Kriterij razrješivanja višeznačnosti vodi se za ranije iznesenim – s gledišta umjetne inteligencije, a i modeliranja procesa razmjene obavijesti – općim ciljem izrade parsera kao računalnoga sustava koji kvalitetno obavlja zadatke koje kvalitetno obavljaju ljudi. Budući da uspješna razmjena obavijesti implicira prihvat upravo one obavijesti koju je pošiljatelj naumio odaslati, unatoč mogućoj – štoviše, vjerojatnoj, s obzirom na prirodu jezika – višeznačnosti tekstne poruke, postupak prihvaćanja obavijesti jezičnim aparatom uključuje i razrješivanje višeznačnosti. Stoga se i od parsera zahtijeva da zna razriješiti sintaktičku višeznačnost u rečenicama ulaznoga teksta, odnosno da zna među svim mogućim parsnim stablima neke rečenice odabrati jedno parsno stablo – ono točno, odnosno valjano s obavijesnoga gledišta.

Parser P jezika L razrješuje sintaktičku višeznačnost ako i samo ako za svaki tekst $T = (s_1, \dots, s_n), T \subseteq L$ svakoj rečenici $s_i \in T$ dodijeli najviše jedno parsno stablo.

Različiti parseri, kako će biti pokazano kasnije, pristupaju problemu razrješivanja višeznačnosti na različite načine, između ostaloga, isključivom dodjelom samo jednoga parsnog stabla nekim deterministčkim postupkom koji uvijek bira tumačenje koje smatra najboljim ili, s druge strane, izradom popisa parsnih stabala s padajućom ocjenom valjanosti pojedinoga parsnog stabla. U svakom slučaju, neovisno o pristupima pojedinih parsera, nužno je primijetiti da ni kriterij razrješivanja višeznačnosti, kao ni kriterij robustnosti, nije samostalan budući da je moguće izraditi parser koji apsolutno udovoljava tomu kriteriju, a na izlazu nikad ne daje parsno stablo, neovisno o ulaznome tekstu.

S obzirom na nedovoljnu ograničenost, odnosno nesamostalnost kriterija robustnosti i kriterija razrješivanja višeznačnosti, ali i s obzirom na njihovu komplementarnost, uvodi se vezani kriterij koji se naziva kriterijem *robustnoga razrješivanja višeznačnosti* (Nivre 2006:42) i zahtijeva od parsera zadovoljavanje oba prethodna kriterija.

Kaže se da parser P jezika L robustno razrješuje sintaktičku višeznačnost ako i samo ako za svaki tekst $T = (s_1, \dots, s_n), T \subseteq L$ svakoj rečenici $s_i \in T$ dodijeli, spojeno iz sastavnih kriterija, "barem jedno i najviše jedno parsno stablo", što znači da svakoj rečenici $s_i \in T$ dodijeli uvijek samo jedno parsno stablo. Kriterij robustnosti pritom jamči da se parsno stablo uvijek mora pojaviti, a kriterij razrješivanja višeznačnosti jamči da se neće pojaviti više od

jednoga parsnog stabla. Dakle, traženi parseri prirodnoga jezika ovdje uvijek moraju dati za ulaznu rečenicu točno jedno parsno stablo.

2.1.2.3.3.2 Točnost parsanja

Neovisno o očitoj smislenosti spajanja robustnosti i razrješivanja višeznačnosti u jedan povezani kriterij robustnoga razrješivanja višeznačnosti, ni na taj se način ne može u potpunosti sagledati kvaliteta nekoga parsera. Naime, parser može apsolutno robustno razrješivati višeznačnost, odnosno za svaku ulaznu rečenicu dati na izlazu jedno parsno stablo, a da to parsno stablo pritom nije valjana sintaktička analiza ulazne rečenice. Stoga se uvodi kriterij točnosti kojim se uspostavlja veza između parsanja nekoga prirodnog jezika i zadanoga sintaktičkog formalizma toga jezika. Kriterij točnosti pretpostavlja da za svaku rečenicu ulaznoga teksta postoji, prema sintaktičkome formalizmu i u obavijesnome kontekstu, samo jedno valjano sintaktičko tumačenje[35].

Parser P jezika L je točan, odnosno izvodi ispravno parsanje prema zadanome sintaktičkom formalizmu, ako i samo ako za svaki tekst $T = (s_1, \ldots, s_n), T \subseteq L$ svakoj rečenici $s_i \in T$ dodijeli baš ono parsno stablo koje predstavlja točno tumačenje te rečenice prema zadanome formalizmu.

Apsolutna točnost parsera smatra se asimptotskim ciljem (usp. Nivre 2006:42), posebno za načelno neograničene tekstove prirodnoga jezika te u kombinaciji s kriterijem robustnoga razrješivanja višeznačnosti. Također, hipotetski apsolutno točni parser se nikakvim formalnim metodama vrjednovanja uopće ne bi mogao utvrditi, odnosno prepoznati njegovo postojanje budući da ovdje razmatrani prirodni jezik L zapravo nije formalni jezik, već skup koji predstavlja sve postojeće tekstove za koje se smatra da pripadaju tomu prirodnom jeziku, neovisno o stupnju gramatičnosti. Vrjednovanje točnosti parsera po ovoj se definiciji stoga provodi nad nekim reprezentativnim uzorkom jezika L, odnosno nekim referentnim tekstom koji je prethodno apsolutno točno parsan – poželjno od strane mjerodavnih stručnjaka za sintaksu toga jezika – prema određenom sintaktičkom formalizmu.

[35] (Nivre 2006:42) razmatra i mogućnost višestrukih valjanih sintaktičkih tumačenja jedne ulazne rečenice, no svejedno se ograničava na jedno tumačenje s pomoću kriterija robustnoga razrješivanja višeznačnosti. Naime, ukoliko postoji više valjanih parsnih stabala iste rečenice, dani kriteriji svejedno osiguravaju točno tumačenje na izlazu iz parsera. Moglo bi se razmatrati izostavljanje ostalih tumačenja problemom pokrivenosti parsera, no ne za potrebe ovoga istraživanja.

Točnost predstavlja optimizacijski kriterij, utoliko što se želi pronaći parser koji je što točniji, a da pritom ne narušava kriterij robustnoga razrješivanja višeznačnosti. S obzirom na definiciju kriterija točnosti, testno okruženje u kojem se pokušava utvrditi točnost parsera nužno uključuje izdvojeni skup od povjerenja koji sadrži dovoljan broj apsolutno točno parsanih rečenica koje istovremeno predstavljaju reprezentativan uzorak prirodnoga jezika za koji se traži parser, odnosno aproksimaciju kojom se asimptotski teži idealu apsolutnoga vrjednovanja parsera u zatvorenom sustavu formalnoga jezika i pripadajuće formalne gramatike. O kriterijima veličine, točnosti i reprezentativnosti toga uzorka teksta detaljnije će se raspravljati kasnije.

2.1.2.3.3.3 Učinkovitost parsanja

Potpuno točan parser koji u potpunosti udovoljava kriteriju robustnoga razrješivanja višeznačnosti može se na neki način smatrati idealnim parserom budući da će svakoj rečenici ulaznoga teksta uvijek dodijeliti samo jedno ispravno sintaktičko tumačenje. Međutim, s gledišta umjetne i računalne inteligencije i primjenjivosti njezinih rješenja, takav se parser ne mora nužno smatrati i savršenim parserom. Naime, uz raniju definiciju umjetne inteligencije, usmjerenu izradi "strojeva koji su u stanju obavljati zadatke, za koje je kvalitetno obavljanje svojstveno samo ljudima, usporedivo kvalitetno kao ljudi" (Kurzweil 1992, usp. Russell i Norvig 2009:2), veže se i gledište korisnosti rada takvih strojeva, odnosno računalnih algoritama i računalnih rješenja koja iz njih proizlaze, u usporedbi s ljudskim radom. Računala podatke, između ostalih prednosti, mogu obrađivati brže i konstantnije nego ljudi (usp. Russell i Norvig 2009:12). Ukoliko se, dakle, razmatra izvedba računalnoga algoritma, u ovome slučaju parsera koji simulira ljudsku inteligenciju za rješavanje nekoga problema i izvođenje nekoga zadatka, poželjno je njegovo svojstvo – a to mu svojstvo daje i korisnost u primjenama – da taj problem i zadatak rješava u najkraćem mogućem vremenu te, poželjno, uz utrošak najmanje količine ostalih računalnih resurasa. U protivnome, korisnost bi takvoga sustava izvan potreba znanstvenih istraživanja, odnosno uporabe računala kao simulatora ljudske inteligencije, bila uvelike umanjena. Takvo rasuđivanje nužno vodi optimizacijskom kriteriju učinkovitosti parsera.

Ranije je pokazano da su parseri neograničenim beskontekstnim gramatikama, odnosno parseri beskontekstnih jezika uglavnom polinomske (kubne) složenosti u najgorem slučaju, odnosno da parsaju ulazne rečenice brzinom koja je polinomskom (kubnom) funkcijom vezana uz duljinu ulaznoga niza. Neka ograničenja ulaznih beskontekstnih gramatika mogu

umanjiti tu vremensku složenost na kvadratnu (kao kod Earleyjeva parsera i nevišeznačnih gramatika) ili čak linearnu (Earleyjev parser, LL(k)). Dakle, uz povezivanje kriterija učinkovitosti s prethodnim kriterijima, može se reći da savršeni parser za ulaznu rečenicu nekoga jezika na izlazu uvijek daje jedno točno parsno stablo te rečenice u skladu sa zadanim sintaktičkim formalizmom, u najkraćem mogućem vremenu.

Kriterij učinkovitosti očito je optimizacijski kriterij budući da se načelno teži linearnoj vremenskoj složenosti[36], a također je poželjna i linearna prostorna složenost, odnosno linearni memorijski zahtjevi s obzirom na veličinu ulaznih podataka.

Parser P jezika L je učinkovit ako i samo ako za svaki tekst $T = (s_1, \dots, s_n), T \subseteq L$ svaku rečenicu $s_i \in T$ iz toga teksta parsa u linearnom vremenu $O(|s_i|)$.

S obzirom na praktične primjene inteligentnih računalnih sustava, u slučaju računalnih parsera treba razlikovati (usp. Nivre 2006:43) teoretsku složenost i učinkovitost korištenoga parsnog algoritma od praktične složenosti i učinkovitosti samoga parsera. Teoretska, odnosno asimptotska složenost utvrđuje se analizom parsnoga algoritma (usp. Cormen i dr. 2009), dok se praktična učinkovitost samoga parsera utvrđuje mjerenjima određenoga skupa svojstava – primjerice, procesorskih i memorijskih zahtjeva – pri stvarnoj uporabi računalne izvedbe modela parsera u nekom testnom okruženju. U praksi se pokazuje, primjerice, da polinomska $(O(n^k), k \in \mathbb{R}^+)$ asimptotska složenost algoritma – iako se u računalnoj znanosti i teoriji izračunljivosti smatra poželjnom s obzirom na postojeće skupove općih izračunskih problema i pripadajućih rješenja – ne predstavlja nužno jamstvo učinkovitosti računalnoga sustava u kojem je taj algoritam izveden. Stoga se pred parsere – suprotno intuiciji s obzirom na ranije predstavljene parsere formalnom gramatikom te s obzirom na složenost prirodnih jezika s gledišta modeliranja formalnim aparatom – ovdje eksplicitno navodi zahtjev težiti linearnoj asimptotskoj složenosti, s ciljem posljedičnoga dosezanja zadovoljavajuće razine praktične učinkovitosti.

2.1.2.3.3.4 Unutarnje i vanjsko vrjednovanje

Četiri iznesena kriterija predstavljaju okvire za optimizaciju parsera. Potrebno je u potpunosti osigurati robustno razrješivanje višeznačnosti i pritom što točnije (poželjno, uvijek

[36] Vrijedi primijetiti da idealna brzina izvođenja nekoga algoritma nije linearna $O(n)$, nego konstantna $O(1)$. Konstantna složenost, međutim, nije moguća budući da svaki stvarni algoritam koji provodi stvarnu obradbu podataka mora te podatke barem pročitati, a sama ta operacija je načelno linearne prostorne i vremenske složenosti.

u potpunosti točno) i što brže parsati (poželjno, uvijek u asimptotski linearnome prostoru i vremenu te mjerljivo uz minimalan utrošak računalnih resursa, odnosno procesorskoga vremena i memorijskoga prostora pri korištenju sustava za parsanje temeljenoga na parsnome algoritmu) rečenice ulaznoga teksta. Intuicija nalaže kako ova dva optimizacijska kriterija, točnost i učinkovitost, moraju biti u obrnuto proporcionalnome odnosu, pa se traži parser koji robustno razrješuje sintaktičku višeznačnost ulaznoga teksta, a pritom nastoji dati što kvalitetnije parsanje s obzirom na vremenska i prostorna ograničenja ili dati što brže parsanje s obzirom na minimalne zahtjeve za kvalitetom rezultirajućih parsnih stabala. Ta dva suprotna gledišta upućuju na potrebu za vanjskim vrjednovanjem parsera: u teoriji se istovremeno želi pronaći najbrži i najtočniji parser, a u praksi, budući da su ti kriteriji vezani odnosom obrnute proporcionalnosti, ciljana primjena parsera u većem sustavu za obradbu prirodnoga jezika uvjetuje priklanjanje jednomu od dvaju kriterija, ovisno o tome je li rezultirajućem većem sustavu bitnija brzina ili točnost njegovih komponenata. Utoliko se na parsere, kao i na druge sustave za obradbu prirodnoga jezika, može načelno primijeniti isti princip unutarnjega (intrinsičnoga) i vanjskoga (ekstrinsičnoga) vrjednovanja (usp. Šnajder 2010:99).

Unutarnje vrjednovanje prema utvrđenim općim kriterijima predstavlja vrjednovanje parsera kao takvoga, odnosno prema kvaliteti parsanja ulaznoga teksta u odnosu na neki prethodno parsani referentni tekst. Unutarnje vrjednovanje nije vezano uz doprinos parsera kvaliteti rada nekoga drugog računalnog sustava u kojem se parser može upotrijebiti kao komponenta i prema kriterijima za vrjednovanje toga drugog sustava.

Vanjsko vrjednovanje mjeri doprinos parsera kvaliteti sustava u kojem se koristi i provodi se uspostavljanjem uzročno-posljedične veze između kvalitete samoga parsera i kvalitete sustava čiji je dio. Unutarnje vrjednovanje parsera u vanjskome se vrjednovanju smatra njegovim nerazdvojnim svojstvom, a može se stoga mjeriti utjecaj promjene kvalitete rada parsera, opisane rezultatima njegova unutarnjeg vrjednovanja, na promjenu kvalitete rada sustava preko kojega se parser izvanjski vrjednuje.

Ovdje je iznesen formalni okvir za vrjednovanje parsera koji uključuje dva općenita formalna preduvjeta (povezana u obuhvatni kriterij robustnoga razrješivanja višeznačnosti) i dva općenita formalna kriterija unutarnjega vrjednovanja (točnost i učinkovitost) te općeniti pristup njihovoj primjeni u unutarnjem i vanjskom vrjednovanju. Vrjednovanje parsera dodatno je razrađeno dalje u tekstu.

2.1.2.3.4 Kriteriji vrjednovanja i parsanje gramatikom

Sada se može razmatrati parsanje formalnom gramatikom s pomoću navedenih kriterija i time produbiti ranije iznesenu početnu ocjenu o njihovoj načelnoj neprikladnosti u rješavanju problema parsanja prirodnoga jezika. Gledište robustnoga razrješivanja višeznačnosti već je raspravljeno: zbog nemogućnosti opisivanja svih sintaktičkih fenomena nekoga prirodnog jezika unutar paradigme formalne gramatike i restriktivnosti formalnom gramatikom opisanoga podskupa toga prirodnog jezika, parsanje formalnom gramatikom prema ovdje postavljenoj definiciji nije robustno jer nedovoljna obuhvatnost gramatike posljedično nužno onemogućava parsanje određenoga broja ulaznih rečenica. Parsanje formalnom gramatikom ne razrješuje sintaktičku višeznačnost budući da paradigma formalne gramatike po definiciji ne predviđa kvantitativno vrjednovanje pojedinih analiza, odnosno pojedinih parsnih stabala koja proizvede parsni algoritam. Parsanje formalnom gramatikom stoga ne zadovoljava kriterij robustnoga razrješivanja višeznačnosti.

Parsanje formalnom gramatikom kriteriju točnosti ili udovoljava u potpunosti ili mu udovoljava djelomično, ovisno o gledištu prema robustnosti (i pokrivenosti) te razrješivanju višeznačnosti. Naime, budući da parsanje formalnom gramatikom nije robustno – ne vraća barem jedno tumačenje za svaku ulaznu rečenicu; utoliko mu je svojstven i problem manjka pokrivenosti – i ne razrješuje višeznačnost – jer nudi višestruka tumačenja budući da "sve formalne gramatike cure" i svojstven im je problem prekomjernoga generiranja rečenica i sintaktičkih analiza – kriteriju točnosti udovoljava tako da za svaku rečenicu za koju je robustno u skupu svih ponuđenih parsnih stabala sigurno sadrži i ono jedno traženo parsno stablo određeno kriterijem razrješivanja višeznačnosti. Međutim, treba ukupnu točnost parsanja formalnom gramatikom računati na čitavome ulaznom tekstu, a ne samo na onim rečenicama koje su pokrivene formalnom gramatikom. Također treba uzeti u obzir i šum na izlazu iz parsera koji je uzrokovan potrebom da se iz skupa svih ponuđenih parsnih stabala odabere jedno valjano parsno stablo.

Točnost parsanja formalnom gramatikom je, dakle, funkcijski ovisna o pokrivenosti gramatikom na način da je najveća moguća točnost takva parsera ograničena pokrivenošću proporcionalnim odnosom: što je veća pokrivenost, to je veća i ukupna točnost parsanja budući da svaka pokrivena rečenica dobije (i) valjano parsno stablo, a svaka nepokrivena uopće ne dobije parsno stablo. Može se reći: ako je točnost broj između 0 (potpuna netočnost) i 1 (potpuna točnost), najveća moguća točnost jednaka je pokrivenosti gramatikom koja je

također broj između 0 (ne pokriva nijednu rečenicu) i 1 (pokriva sve rečenice). Nadalje, najveća moguća točnost može se postići samo ako parser formalnom gramatikom za svaku pokrivenu rečenicu daje samo jedno parsno stablo i time udovoljava kriteriju robustnoga razrješivanja višeznačnosti. Šum koji uvodi nemogućnost udovoljavanja tomu kriteriju, a koja je svojstvena formalnim gramatikama, dodatno umanjuje stvarnu točnost označavanja u odnosu na najveću moguću. Za svako višeznačno tumačenje najveća se moguća točnost umanjuje proporcionalno broju višeznačnih tumačenja. Stvarna se točnost parsanja formalnom gramatikom s obzirom na pokrivenost rečenica ulaznoga teksta i razrješivanje višeznačnosti stoga može izraziti općenitom formulom.

Neka je parser P jezika L formalnom gramatikom G označen kao $P(G)$ i neka parsa rečenice iz teksta $T = (s_1, \dots, s_n), T \subseteq L$. Postupkom parsanja, na izlazu parser $P(G)$ daje skup $A = (a_{11}, \dots, a_{km})$ svih parsnih stabala za sve uspješno parsane rečenice teksta. Neka je značenje svakog elementa $a_{ij} \in A$ jednako "j-to parsno stablo i-te rečenice $s_i \in T$" te neka je svakom elementu $a_{ij} \in A$ pridijeljena vrijednost $a_{ij} = 1$. U tome okviru, pokrivenost teksta T gramatikom G i ukupna točnost parsera mogu se definirati kao:

$$\text{pokrivenost} = \frac{\sum_i a_{i1}}{|T|} \left(= \frac{\text{broj prihvaćenih rečenica}}{\text{broj rečenica ulaznoga teksta}} \right)$$

$$\text{točnost} = \frac{\sum_i a_{i1}}{|T|} \cdot \frac{|A| - \sum_i \left(\sum_j (a_{ij}) - 1\right)}{|A|} \; (= \text{pokrivenost} * \text{nevišeznačnost})$$

Formula pokazuje odnos između točnosti parsanja formalnom gramatikom, broja ulaznih rečenica, pokrivenosti formalnom gramatikom i višeznačnosti parsanja. Pokrivenost ulaznoga teksta gramatikom pritom je definirana kao omjer broja rečenica koje je parser gramatikom prepoznao i ukupnoga broja rečenica. Tako definirana pokrivenost predstavlja najveću moguću točnost parsera koja se potom u stvarnu točnost pretvara množenjem s koeficijentom nevišeznačnosti. Koeficijent nevišeznačnosti definiran je kao omjer broja nevišeznačnih parsnih stabala i ukupnoga broja parsnih stabala. Ukoliko je svakoj rečenici dodijeljeno samo jedno parsno stablo, parser postiže potpunu točnost. U protivnome, točnost je određena brojem prepoznatih rečenica i brojem višeznačnih tumačenja.

S obzirom na nerazdvojivost paradigme formalne gramatike od sintaktičke višeznačnosti u analizi, odnosno od prekomjernoga generiranja parsnih stabala, očigledno je kod parsanja prirodnoga jezika stvarnom formalnom gramatikom nemoguće dosegnuti

najveću moguću točnost. Najveća je moguća točnost pritom dodatno omeđena pokrivenošću ulaznoga teksta gramatikom. Proizlazi da, u stvarnim uvjetima, manjkavost parsera formalnom gramatikom, koja proizlazi iz neudovoljavanja uvjetima robustnosti i razrješivanja viženačnosti, u potpunosti onemogućava postizanje potpune točnosti pri parsanju prirodnoga jezika formalnom gramatikom.

Učinkovitost parsanja formalnom gramatikom proizlazi iz korištenoga algoritma za traženje parsnih stabala, odnosno nizova produkcija formalne gramatike koji opisuju kako gramatika generira ulaznu rečenicu. Ako se koristi formalizam beskontekstne gramatike i ako je ta beskontekstna gramatika neograničena – a očekivano je da mora biti takva budući da je i bez uvođenja ograničenja nedovoljno izražajna da objasni sve sintaktičke pojave u prirodnim jezicima – pripadajući algoritmi, poput CYK-a i Earleyjeva parsera, polinomske (kubne) su složenosti. Ukoliko se upotrijebi gramatički formalizam koji je složeniji od beskontekstne gramatike, složenost pripadajućih algoritama se uvećava[37]. S obzirom na ranije predstavljene zahtjeve za linearnom teorijskom složenošću i posljedičnom umanjenom stvarnom zahtjevnošću izvođenja, parsanje formalnom gramatikom ne može biti učinkovito.

Razmatranjem upravo uspostavljenih kriterija robustnoga razrješivanja višeznačnosti te optimizacije točnosti i učinkovitosti, pokazano je kako parsanje formalnom gramatikom ne može za potrebe ovoga istraživanja biti korišteno kao paradigma unutar koje se parsaju tekstovi prirodnoga jezika. Dalje se razmatraju neki od pristupa parsanju koji udovoljavaju navedenim kriterijima.

2.1.2.3.5 Parsanje teksta

Ranije je – postavljanjem kriterija za izvedbu i odabir parsera – ocrtana razlika između aproksimacije prirodnoga jezika $L(G)$ definirane formalnom gramatikom G te prirodnoga jezika L ostvarenoga u tekstu T. Prije razmatranja parsera koji parsaju tekstove nekoga jezika, potrebno je detaljnije pojasniti razliku između parsanja (aproksimacije) prirodnoga jezika (gramatikom) i parsanja tekstova nekoga prirodnog jezika.

Pri parsanju prirodnoga jezika L nekom formalnom gramatikom G, čije su manjkavosti upravo prikazane i koje je zbog tih manjkavosti odbačeno iz razmatranja za potrebe ovoga istraživanja, parsanje se zapravo provodi unutar modela toga prirodnog jezika, izvedenoga u

[37] Primjerice, (Nivre 2006:16) kaže da složenost parsanja jednostavnim kontekstno-ovisnim gramatikama ostaje polinomska, najčešće $O(n^6)$ pa do algoritama eksponencijalne složenosti $O(k^n)$ u najgorem slučaju.

obliku neke gramatike G. Utoliko parser P(G) ne parsa prirodni jezik L, nego – uslijed nepotpunosti modela – njegov pravi podskup L(G), pa posljedično nije prikladan za parsanje neograničenih tekstova jezika L budući da je onemogućen izborom modela. Ako se želi napraviti parser tekstova nekoga prirodnog jezika, usklađen s prethodno postavljenim formalnim kriterijima vrjednovanja, on mora na neki način biti usmjeren upravo tekstovima toga jezika, a ne fiksiranomu računalnom modelu sintakse toga jezika. Ta usmjerenost tekstu, radije nego nekom sintaktičkom standardu toga jezika i njegovu računalnom modelu, smislena je zbog, između ostaloga, neograničenosti prirodnoga jezika izvan dosega računalnoga modeliranja te zbog uvođenja tolerancije prema pogrješkama u jeziku koje ne mijenjaju nužno sintaktičku i semantičku strukturu tekstne poruke.

Parsanje teksta prirodnoga jezika odnosi se (usp. Nivre 2006:16) na "konkretna ostvarenja nekoga prirodnog jezika, bez razmatranja mogućnosti da on bude formalni jezik". Prirodni se jezik, dakle, na ovaj način promatra kao skup svih njegovih ostvarenja u tekstu, a ne kao formalni sustav opisan nekim računalno izvedivim formalizmom (ili fomalizmima) na nekoj razini (ili razinama) jezičnoga opisa. Problem parsanja nekoga prirodnog jezika stoga se s ovoga gledišta pobliže naziva *problemom parsanja tekstova* toga jezika. Zadatak parsera utoliko je, za dani tekst $T = (s_1, \dots, s_n), T \subseteq L$ prirodnoga jezika L, parsanjem dati točno sintaktičko tumačenje, odnosno po jedno parsno stablo za svaku rečenicu $s_i \in T$.

Konceptualni model parsera sa slike 3-4 zahtijeva od svakoga parsera teksta modul za predobradbu ulaznoga teksta, model prirodnoga jezika koji se parsa te parsni algoritam. Modul za predobradbu ulaznoga teksta može se, na neki način, i dalje smatrati trivijalnim i pretpostaviti da je razdvajanje teksta na rečenice (ili segmentacija na rečenice (usp. Boras 1998)) već dostupno u ulaznome tekstu[38]. Potrebno je, dakle, za svaki od pristupa parsanju teksta – prema općoj definiciji toga problema i u skladu s općim kriterijima vrjednovanja – opisati njegov jezični model i pripadajući algoritam za parsanje. Veza između jezičnoga modela i algoritma za parsanje ostaje čvrsta kao ranije kod formalnih gramatika i pripadajućih algoritama. Međutim, prije opisa konkretnih modela i algoritama, treba detaljnije razmotriti opću definiciju problema parsanja teksta.

[38] (Nivre 2006:17) raspravlja o netrivijalnosti problema segmentacije ulaznoga teksta na rečenice, s gledišta težine samoga problema, ali i zbog činjenice da tekstni podatci često sadrže i nerečenične strukturne elemente. Unatoč tome, tamo predstavljeno istraživanje također je usmjereno parsanju, a ne predobradbi pa također završava tu raspravu "jednostavnim zanemarivanjem problema segmentacije".

Za razliku od parsanja formalnom gramatikom, parsanje teksta prirodnoga jezika nije formalno precizno definiran problem (usp. Nivre 2006:17). Naime, kod parsanja formalnom gramatikom, definicija problema proizlazi iz same činjenice da se koristi formalna gramatika kao uređena četvorka koja se sastoji od skupa svih završnih simbola, skupa svih nezavršnih simbola, skupa produkcija te izdvojenoga početnog nezavršnog simbola. Skup završnih simbola predstavlja, s gledišta prirodnoga jezika koji se parsa, popis svih prihvatljivih riječi toga jezika, a skup nezavršnih simbola skup svih sintaktičkih kategorija, dok skup produkcija predstavlja sintaktičke zakonitosti toga jezika. S obzirom na zatvorenost problema parsanja, osiguranu gramatikom, u tome je okviru moguće precizno definirati:

1. ulazne i izlazne skupove podataka – budući da su sve prihvatljive ulazne rečenice upravo oni nizovi riječi u kojima je svaka riječ element skupa svih završnih simbola gramatike, odnosno nijedna ulazna riječ ne smije biti izvan toga skupa jer u protivnom parser neće za takvu rečenicu moći izvršiti parsanje;
2. vezu između parsanja i prepoznavanja – budući da parsanje formalnom gramatikom ujedno vrši i funkciju određivanja gramatičnosti rečenice: ako je parsanje formalnom gramatikom završilo uspješno, odnosno ako je parser vratio jedno ili više parsnih stabala za ulazni niz riječi, to ujedno znači i da je ulazni niz riječi valjana rečenica modela prirodnoga jezika opisanog formalnom gramatikom; i
3. okvire obradbe – budući da se parsanje formalnom gramatikom, činjenicom da za sintaktički višeznačne rečenice vraća sva moguća parsna stabla koja ih objašnjavaju, definira isključivo na sintaktičkoj razini, nemogućnost razrješivanja višeznačnosti, koju je načelno moguće razrješivati samo na višoj razini jezične obradbe, jamči da parsanje formalnom gramatikom predstavlja isključivo problem sintaktičke obradbe prirodnoga jezika, odnosno da ne sadrži semantičke i druge implikacije.

Formalna gramatika kao jezični model, dakle, u potpunosti određuje okvire problema parsanja. Kod parsanja bez formalne gramatike, odnosno korištenjem nekoga drugog jezičnog modela, koji na neki način proizlazi iz teksta kao ostvaraja nekoga jezika, problem načelno nije ograničen ni na jednoj od predstavljenih razina.

1. Ulazni i izlazni skupovi podataka nisu definirani. Budući da je ranijom definicijom problema parsanja teksta dopušteno da ulaz u parser bude bilo koji tekst nekoga prirodnog jezika, ne postoji popis potvrđenih riječi toga jezika iz kojega je dozvoljeno graditi nizove, odnosno rečenice. Takva definicija ulaznih podataka

nerestriktivna je prema prirodnome jeziku koji se obrađuje – budući da dozvoljava izvjesne jezične fenomene, poput jezične evolucije u vidu uvođenja novih riječi – i utoliko poželjna, pogotovo u usporedbi s restriktivnošću formalne gramatike koja, pojednostavljeno, problem parsanja prirodnoga jezika svodi na problem parsanja nekoga njegova vrlo ograničenog podskupa. S druge strane, ova otvorenost (ili nepreciznost) definicije ulaznoga skupa podataka može se pokazati otežavajućom okolnošću pri definiranju pristupa parsanju teksta koji se temelji na tekstnim podatcima kao ostvarajima ciljanoga prirodnog jezika. U tome slučaju, ulazni skup podataka koji je definiran implicitno, samim ulaznim tekstom, na neki način implicira poželjnost jezičnoga modela parsera koji se također definira implicitno, nad nekim većim skupom tekstova, po mogućnosti što sličnijem ulaznomu tekstu ili što vjernijem i obuhvatnijem uzorku obrađivanoga prirodnog jezika. S druge strane, nemogućnost razlikovanja valjanih od nevaljanih ulaznih nizova riječi kao neke vrste predfiltriranja nizova riječi koji zbog samoga nevaljanog leksika sigurno neće moći izgraditi valjane rečenice parsanoga jezika svakako otežava posao parseru budući da postoji mogućnost da parser tekstova hrvatskoga jezika pokušava nesvjesno i očekivano neuspješno parsati engleske tekstove. S obzirom na to, može se pokazati potrebnim u postupku predobradbe po nekom vanjskom kriteriju filtrirati ulazne nizove riječi i propustiti parseru samo one koji po tome kriteriju predstavljaju razumne kandidatske rečenice.

Što se tiče izlaznoga skupa podataka, odnosno skupa svih mogućih parsnih stabala kod parsanja formalnom gramatikom, on je zadan skupom produkcija u kojem skup nezavršnih simbola zapravo predstavlja skup svih sintaktičkih funkcija koje se mogu dodijeliti nekim riječima ili skupovima riječi s obzirom na njihovu službu u rečenici. Kod parsanja bez eksplicitne formalizacije sintaktičkih pravila, sintaktičke se uloge i funkcije definiraju iz podataka, odnosno iz teksta, što implicira da neki uzorak tekstova – primjerice, ranije spomenuti "što je moguće veći, vjerniji i obuhvatniji uzorak obrađivanoga prirodnog jezika" – mora, uz razinu samoga teksta kao ostvarenja jezika, sadržavati i eksplicitni opis sintakse toga teksta, odnosno njegovu sintaktičku analizu. Iz toga opisa, parser koji se temelji na tekstu može izvesti vlastiti model sintaktičkih pravila toga jezika koji će onda predstavljati njegov skup izlaznih podataka. Takav pristup izradi sintaktičkoga jezičnog modela implicira da se sintaktički model jezika mijenja ovisno o tekstu iz kojega se izrađuje. Ovdje također treba razlikovati tekst iz kojega se gradi sintaktički jezični model nekoga

prirodnog jezika i ulazni tekst koji se parsa, odnosno na kojem se usvojeni jezični model potom primijenjuje. O ovim se skupovima tekstova kasnije u više navrata dodatno raspravlja.

2. Jezični model parsera koji ne sadrži formalnu gramatiku kao eksplicitno izvedenu vezu između leksika i morfosintakse sa sintaksom parsanoga prirodnog jezika nužno ne može uz funkciju parsanja izvršavati i ulogu prepoznavača gramatičnih rečenica toga jezika. Naime, formalna gramatika produkcijama preko nezavršnih simbola prema završnima, generiranjem svih mogućih rečenica, u potpunosti ostvaruje apstraktni sintaktički model toga jezika. Može se reći da formalna gramatika generira čitav jezik, a problem prepoznavanja gramatičnih ulaznih rečenica formalnom gramatikom svodi se pritom na jednostavnu usporedbu ulazne rečenice sa skupom svih rečenica koje je gramatika generirala. Kod jezičnoga modela koji ne sadrži eksplicitnu izvedbu sintaktičkih pravila parsanoga prirodnog jezika ne može se govoriti o gramatičnosti (ili sintaktičkoj ispravnosti) rečenica, posebno ako se taj jezični model zasniva na tekstovima toga jezika budući da ti tekstovi mogu sadržavati i rečenice koje su sintaktički nevaljane prema nekom vanjskom sintaktičkom opisu toga jezika, ali su svejedno u jeziku prisutne, pa čak i često korištene. S druge strane, slično kao kod nedefiniranosti ulaznoga i izlaznoga skupa podataka, jezično sintaktičko modeliranje iz samoga teksta dopušta i na sintaktičkoj razini izvjesnu slobodu i "opuštenost jezičnih pravila" koja dalje dopušta mogućnost jezične promjene i promatra tu jezičnu promjenu izravno u njezinu tekstnom ostvaraju, a ne kompenzira za nju posredno, putem moguće krute i zatvorene izvedbe sintaktičkih pravila. Prilagodba ovakvoga modela na nove tekstove, odnosno nove sintaktičke fenomene promatranoga prirodnog jezika, vrši se iz samih tekstova, dok se kod parsanja formalnom gramatikom može vršiti samo posredno, prilagodbom pravila po uočavanju novih fenomena, moguće po opažanju dodatne degradacije kvalitete parsera formalnom gramatikom prema kriteriju robustnosti, odnosno pokrivenosti. Valja primijetiti kako i prilagodba pravila i prilagodba modela koji se crpi iz tekstova može zahtijevati nezanemarivu količinu (ljudskih) resurasa.
3. Kriterij robustnoga razrješivanja višeznačnosti, ranije općenito postavljen pred parsanje, pa tako i parsanje teksta, zahtijeva od parsera da svakoj rečenici dodijeli isključivo jedno, poželjno ispravno parsno stablo. Ispravnost parsnoga stabla pritom znači točno tumačenje ulazne rečenice na razini sintakse, unatoč samoj prirodi

(prave) sintaktičke višeznačnosti koja je takva da se ne može razriješiti bez izvan- ili iznadrečeničnoga konteksta, odnosno obradbe na višoj razini jezičnoga opisa. Parseri formalnom gramatikom, kako je već rečeno, ne mogu udovoljiti ovomu zahtjevu bez vanjskih modula za razrješivanje višeznačnosti. Parseri teksta temeljeni na jezičnim podatcima, odnosno tekstu, mogu čitav tekst na kojem grade sintaktički jezični model koristiti kao aproksimaciju vanrečeničnoga konteksta ili obradbe na višoj razini jezičnoga opisa. Naime, može se sintaktičke pravilnosti toga reprezentativnog uzorka jezika koristiti kao unutarnji modul za odabiranje neke od mogućih analiza kao najvjerojatnije putem prebrojavanja određenih pojavnosti na razini sintaktičkoga opisa. Kriterij razrješivanja višeznačnosti kod parsera s jezičnim modelom temeljenim na jezičnim podatcima stoga je broj pojavljivanja određenih sintaktičkih fenomena, odnosno kriterij učestalosti – ako se neki fenomen pojavljuje češće nego neki drugi, onda je on vjerojatniji kandidat za opisivanje strukture parsane ulazne rečenice.

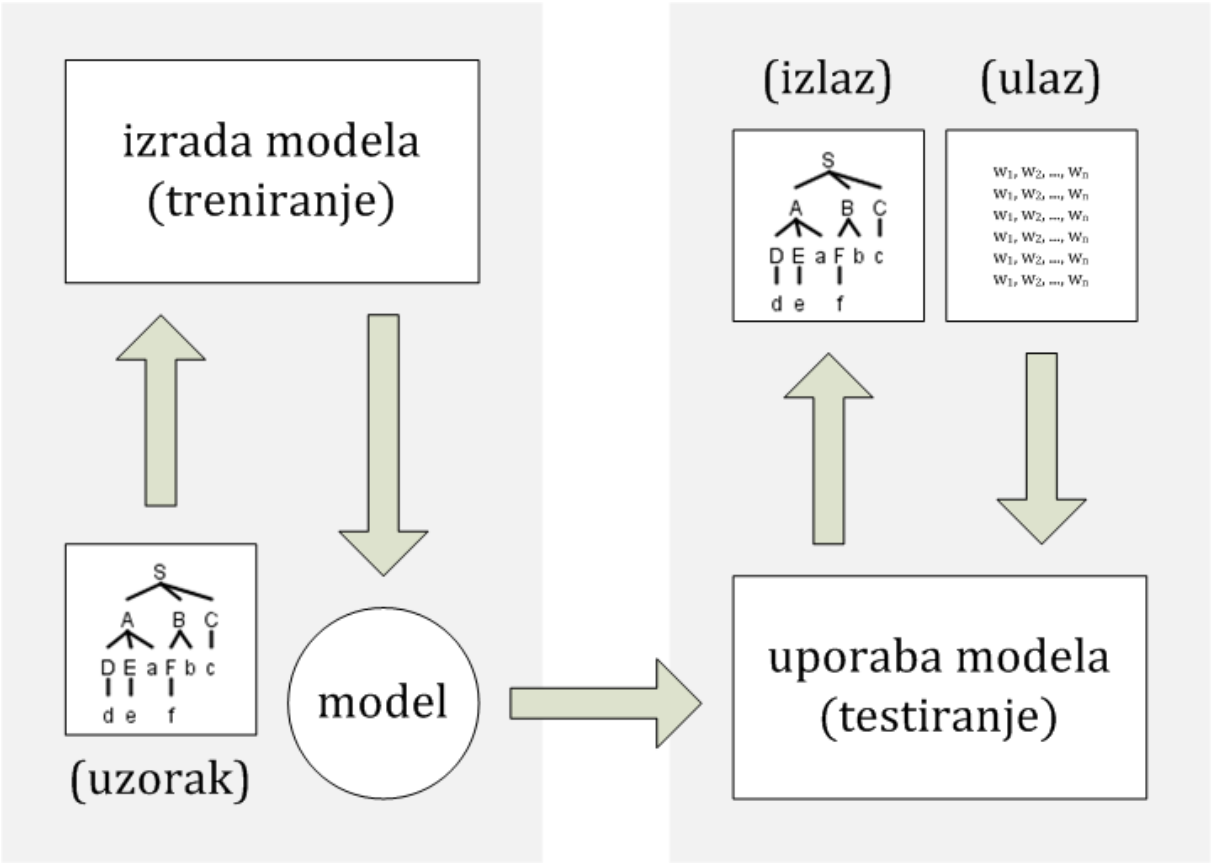
Uz ova tri navedena pogleda na razlike između definicije problema parsanja formalnom gramatikom i parsanja izvan paradigme formalne gramatike, među njima postoji također i jasna razlika s gledišta vrjednovanja točnosti parsanja. Ranije je ilustrirano kako pristupati vrjednovanju točnosti parsanja formalnom gramatikom: formalna je gramatika zatvoreni sustav unutar kojega samo postojanje parsnoga stabla jamči njegovu točnost prema danome sintaktičkom formalizmu, pa se vrjednovanje točnosti svodi na prebrojavanje parsnih stabala, odnosno uspješno parsanih rečenica (pokrivenost), uz uzimanje u obzir pojave mogućih višestrukih parsnih stabala za pojedine uspješno parsane rečenice (višeznačnost). Budući da je parsanje teksta izvan paradigme formalne gramatike, u obliku u kojem je upravo ocrtano, zasnovano na sintaktičkome formalizmu proizvedenom iz jezičnih podataka, ne postoji neki dobro definirani postupak kojim se ono može vrjednovati samo po sebi, odnosno bez jezičnih podataka. Stoga se vrjednovanju takvih parsera teksta pristupa po principu usporedbe parserom parsanoga teksta s nekim referentnim parsanjem toga istog teksta.

2.1.2.4 Parseri temeljeni na podatcima

Parsanje teksta van paradigme formalne gramatike, kako je ovdje predstavljeno – uz zadane kriterije i prema navedenim svojstvima – uključuje, i dalje u skladu s konceptualnim modelom sa slike 2-4, jezični model koji se gradi iz jezičnih podataka, odnosno tekstova jezika za koji je parser namijenjen, te referentno parsanje nekoga uzorka teksta toga jezika na

kojem se provodi vrjednovanje parsera. Utoliko se kaže da je takav parser zapravo inteligentni računalni sustav za obradbu prirodnoga jezika temeljen na podatcima, odnosno *parser temeljen na podatcima* (en. *data-driven parser*, usp. Kübler i dr. 2009:7), za razliku od onoga zasnovanog na formalnoj gramatici (en. *grammar-based*, usp. Kübler i dr. 2009:7). Kad se kaže da je rad nekoga inteligentnog računalnog sustava temeljen na podatcima ili *upravljan podatcima* (en. *data-driven*), pretpostavlja se da su podatci – svojstveni problemu koji taj sustav nekim računalno-inteligentnim postupcima rješava – izravno uključeni u sve faze korištenja toga sustava, od izrade modela pa do primjene na novim podatcima, odnosno praktičnim zadatcima i problemima. Može se reći da je sustav temeljen na podatcima gotovo u cijelosti definiran upravo iz podataka.

Rad sustava za obradbu prirodnoga jezika temeljenih na podatcima odvija se načelno preko dva međuovisna postupka: postupka izrade modela, koji se također naziva i *treniranje modela* (en. *training*) te postupka uporabe modela, zvanoga i *testiranje modela* (en. *testing*). Postupci su prikazani na modelu sa slike 2-5.



Slika 2-5 Model inteligentnoga računalnog sustava temeljenoga na podatcima

Zadatak prvoga postupka, odnosno treniranja ili izrade modela, jest stvoriti iz podataka računalni model primjenjiv na rješavanje predmetnoga problema inteligentnoga računalnog sustava kojemu pripada. Postupak treniranja, ako ga se razmatra kao funkciju, ima jedan ulazni parametar te jednu povratnu vrijednost, računalni model. Ulazni parametar za postupak treniranja modela su podatci iz kojih se model stvara, a ti su podatci svojom prirodom svojstveni predmetnomu problemu, odnosno, najčešće su oblikovani upravo tako da izgledaju poput očekivanih rezultata obradbe podataka pri uporabi izrađenoga modela, odnosno pri testiranju modela u stvarnim zadatcima. Dakle, ukoliko je predmetni problem parsanje tekstova nekoga prirodnog jezika, onda će ulazni podatci za izradu računalnoga modela biti prethodno parsani tekstovi toga jezika, po mogućnosti odabrani tako da predstavljaju što reprezentativniji uzorak toga prirodnog jezika, odnosno njegova ostvaraja u tekstu. Postupak izgradnje modela, dakle, uzima tako pripremljene podatke i iz njih stvara jedan ostvaraj nekoga računalnog modela. Izbor samoga modela ovisi o predmetnome problemu i prikladnosti određenih računalnih (matematičkih) modela koji ga najbolje opisuju. Odabir dobroga računalnog modela za ostvarivanje iz ulaznih podataka svodi se stoga na prepoznavanje nekoga postojećeg formalizma koji dobro – a pritom se pod dobrotom misli na računalnu izvedivost i algoritamsku izračunljivost – opisuje predmetni problem, odnosno dobro apstrahira fenomene koji su predmetnomu problemu svojstveni. Primjerice, kod rješavanja problema dodjele podataka o vrsti riječi u rečeničnome kontekstu, odnosno strojnoga morfosintaktičkog označavanja tekstova prirodnoga jezika (usp. Manning i Schütze 2003:341), uočeno je kako se ovisnost među morfosintaktičkim oznakama najčešće proteže na razini slijeda riječi, i to najčešće u rasponu do tri riječi. Stoga je kao jedan od modela za rješavanje toga problema odabran matematički aparat Markovljevih lanaca ili Markovljevih modela (usp. Manning i Schütze 2003:345) kojemu je svrha upravo modelirati ovisnost neke pojave (ovdje morfosintaktičke oznake neke riječi) u nekome diskretnom vremenu o nekom broju pojava koje su joj prethodile (ovdje morfosintaktičkim oznakama riječi koje se nalaze prije nje u rečenici). Takav morfosintaktički označivač temeljen na podatcima iz prethodno (točno, poželjno od strane stručnjaka) morfosintaktički označenih ulaznih rečenica u postupku treniranja stvara jedan (skriveni) Markovljev model iz prebrojavanja supojavljivanja sljedova morfosintaktičkih oznaka te supojavljivanja pojedinih riječi i oznaka. Taj se model potom koristi u postupku testiranja za označavanje teksta prepoznavanjem sljedova ulaznih riječi u modelu te heurističkim postupcima kad neke od ulaznih riječi modelu nisu poznate (usp. Manning i Schütze 2003:351). Dobar odabir računalnoga modela za problem označavanja

vrsta riječi, odnosno morfosintaktičkoga označavanja, dokazan je primjenom na široku skupinu jezika uz visoku točnost i učinkovitost (usp. Brants 2000, Agić i dr. 2008).

Ova ilustracija služi kako bi ukazala na zahtjeve koji se postavljaju pred postupak treniranja modela u nekome stvarnom sustavu temeljenom na podatcima, odnosno u parseru temeljenome na podatcima. Potrebno je osigurati sljedeće:

1. Pripremiti uzorak već obrađenih podataka iz kojega će se izraditi model. Taj uzorak podataka predstavlja neku vrstu primjera obradbe iz kojega računalni sustav uči kako riješiti predmetni problem. Ukoliko se radi o parseru nekoga prirodnog jezika, uzorak mora biti skup sintaktički označenih (parsanih) tekstova toga jezika, odnosno provjereno točan skup rečenica i pripadajućih parsnih stabala izrađenih u skladu s nekim sintaktičkim formalizmom. Poželjno je također, osim točnoga parsanja i izbora sintaktičkoga formalizma prikladnoga za opisivanje parsanoga jezika, učiniti i da sama zbirka teksta bude na neki način reprezentativna za predmetni prirodni jezik, odnosno da predstavlja *korpus* (en. *corpus*) tekstova toga jezika. Takav sintaktički označeni korpus često se naziva i *banka stabala* (en. *treebank*). O načelima izrade računalnih korpusa nekoga prirodnog jezika ovdje se ne raspravlja, već se naglasak stavlja na banke stabala, odabir sintaktičkoga formalizma i posebno na računalno modeliranje. Više o računalnim korpusima i njihovoj izradi može se pronaći u (McEnery i Wilson 2001, Tadić 2003).
2. Odabrati računalni model koji dobro opisuje predmetni problem. Ranije, u povijesno uvjetovanoj raspravi o parsanju beskontekstnom gramatikom, upravo je formalizam beskontekstne gramatike predstavljao računalni model koji donekle dobro opisuje predmetni problem parsanja tekstova prirodnoga jezika budući da je predstavljao donekle dobar model nekih sintaktičkih fenomena prirodnoga jezika. S obzirom na uspostavljene kriterije vrjednovanja parsera i usmjerenje parsanju teksta, potrebno je odrediti skup računalnih modela koji mogu dobro opisati sintaktičke fenomene prirodnoga jezika, a pritom se algoritamski stvarati iz banaka stabala.
3. Osmisliti i izraditi računalni algoritam koji iz uzorka već obrađenih podataka stvara instancije računalnoga modela. Parser temeljen na podatcima računalni je model pa posljedično i računalni sustav koji se sastoji od dva modula (treniranje, testiranje) te se pred oba modula postavljaju slični zahtjevi, ovisno o primjenjivosti pojedinih zahtjeva. Modul za treniranje mora moći točno i učinkovito proizvesti iz banke

stabala računalni model koji će se potom koristiti u modulu za testiranje, odnosno u parsanju teksta. Potrebno je za svaki par banke stabala (koja pretpostavlja određeni sintaktički formalizam te ga usto implicitno definira parsnim stablima) i računalnoga modela definirati učinkovite algoritme koji ih povezuju.

Zadatak drugoga postupka, odnosno modula za korištenje ili testiranje, koji zapravo predstavlja osnovnu funkcionalnost čitavoga sustava, jest primijeniti treniranjem dobiveni računalni model na predmetnome zadatku. Za parser temeljen na podatcima to znači uzeti kao ulazni parametar model izrađen iz banke stabala u postupku treniranja i upotrijebiti ga za parsanje novih rečenica. Tako opisan, problem parsanja teksta zapravo je problem približnoga određivanja ili *problem aproksimacije* (usp. Nivre 2006:17). Naime, ovako zamišljen parser zapravo je sustav koji promatra (poželjno veliku) količinu već parsanih rečenica te opaženo potom nastoji primijeniti na nove rečenice[39]. Dakle, za dani skup primjernih preslikavanja iz rečenica u parsna stabla, sadržan u banci stabala, za novi se ulazni skup rečenica nastoji izvršiti preslikavanje po uzoru na ono primjerno. Postupak korištenja jezičnoga modela dobivenoga iz banke stabala u prethodnome postupku mora osigurati sljedeće.

1. Osmisliti i izraditi računalni algoritam koji zna primijeniti jezični model na novim podatcima. Kao i kod uparivanja banaka stabala (odnosno pripadajućih sintaktičkih formalizama) s jezičnim modelima, potrebno je također uparivati dobivene jezične modele s pripadajućim algoritmima koji ih znaju koristiti. Osim same primjene jezičnoga modela na ulazni tekst, računalni algoritam za korištenje parsera mora znati rukovati iznimnim slučajevima, odnosno izvjesnim pojavnostima u ulaznome tekstu koje nisu pokrivene jezičnim modelom. Naime, za razliku od modeliranja jezika formalnom gramatikom, gdje je čitav problem parsanja zapisan u jednome sintaktičkom formalizmu, modeliranje prirodnoga jezika s pomoću jednoga primjernog uzorka jezika (banke stabala) i primjena toga modela na nekome drugom uzorku teksta povlači problem nepotpune pokrivenosti toga drugog uzorka modelom. Budući da je sintaktički model parsanoga prirodnog jezika u ovim okvirima u potpunosti definiran bankom stabala – a ona, neovisno o veličini i složenosti, nikad ne može sadržavati sve rečenice nekoga prirodnog jezika, samom činjenicom da je prirodni jezik prebrojivo beskonačan skup rečenica – algoritam za parsanje mora sadržavati neku vrstu statističkoga zaključivanja o ulaznim podatcima

[39] (Nivre 2006:17) kaže da se parsanjem teksta temeljenim na podatcima "pokušava aproksimirati preslikavanje iz novih rečenica u nova parsna stabla na osnovi velikoga, ali konačnoga broja već utvrđenih preslikavanja".

koji nisu u potpunosti sadržani u jezičnome modelu. To statističko zaključivanje i dalje se temelji na jezičnome modelu, no često uključuje i heurističke postupke.

2. Udovoljiti u najvećoj mogućoj mjeri svim kriterijma vrjednovanja. Algoritam za korištenje jezičnoga modela upravo je ona komponenta na koju se primjenjuju svi ranije predstavljeni kriteriji budući da ona predstavlja izvedbu parsanja. Poželjno je i da postupak izgradnje jezičnoga modela bude, primjerice, što učinkovitiji, no točna i učinkovita primjena dobivenoga jezičnog modela na parsanje novoga teksta ishodište je svakoga vrjednovanja sustava za parsanje. Takvo razlučivanje opravdano je utoliko što se očekuje da će se kod parsera temeljenih na podatcima jednom izrađeni jezični model višestruko koristiti, odnosno da će jedno pokretanje postupka treniranja biti praćeno višestrukim pokretanjima postupka testiranja.

Pri osmišljavanju i izradi parsera temeljenoga na podatcima, s obzirom na raspravu o zahtjevima i željenim svojstvima postupka izrade modela (treniranje) i postupka njegova korištenja na neviđenome tekstu (testiranje), potrebno je – s gledišta konkretnoga problema parsanja tekstova nekoga prirodnog jezika – precizno odrediti:

1. željena svojstva banke stabala i pripadajućega sintaktičkoga formalizma s obzirom na svojstva parsanoga prirodnog jezika,
2. jezični model primjenjiv za parsanje i prikladan s obzirom na odabir banke stabala i sintaktičkoga formalizma,
3. postupak izgradnje jezičnoga modela iz banke stabala i
4. postupak primjene jezičnoga modela na parsanje tekstova toga prirodnog jezika, u najvećoj mogućoj mjeri usklađen s kriterijima vrjednovanja.

Slijedi rasprava o pojedinim stavkama s ovoga popisa, odnosno precizno određivanje sastavnih dijelova parsera teksta temeljenih na tekstu.

2.1.2.4.1 Banke stabala i sintaktički formalizmi

Banka stabala (en. *treebank*) je sintaktički označeni računalni korpus[40] tekstova nekoga prirodnog jezika (usp. Abeillé 2003, Wallis 2008). Sintaktička označenost, uključena u samu definiciju banke stabala, implicira neka njezina tražena svojstva. Budući da se parsanje

[40] Ovdje se ne raspravlja o računalnim korpusima prirodnoga jezika općenito, već samo o bankama stabala. Definicije računalnih korpusa (usp. McEnery i Wilson 2001, Tadić 2003, Wallis 2008) slažu se da je računalni korpus zbirka tekstova nekoga jezika, odabrana tako da predstavlja njegov reprezentativni uzorak, moguće izrađen za usmjeravanje nekim ciljanim primjenama.

prirodnoga jezika provodi na rečeničnoj razini, banka stabala mora biti razdvojena na rečenice. Ona je utoliko skup rečenica nekoga prirodnog jezika, takav da je svakoj rečenici pridruženo jedno parsno stablo koje valjano opisuje njezinu sintaktičku strukturu. Dakle, u bankama stabala nužno su označene rečenične granice. Nadalje, može se također postaviti i pitanje osmišljavanja postupka dodjele parsnih stabala rečenicama. Najčešće se u definiciju banke stabala (usp. Wallis 2008) eksplicitno uključuje pojam dodjele i provjere sintaktičke analize od strane čovjeka, odnosno stručnjaka za sintaksu predmetnoga prirodnog jezika. Takve definicije kažu da je banka stabala računalni korpus razdvojen na rečenice tako da je svakoj rečenici dodijeljeno ispravno parsno stablo, a ispravnost parsnih stabala potvrđena je od strane stručnjaka. Pritom je za krajnji ishod nebitno jesu li stručnjaci u postupku izrade banke stabala ručno dodjeljivali parsna stabla rečenicama ili su provjeravali i dorađivali rezultate neke automatske analize; definicija zahtijeva samo da banka stabala bude reprezentativna za sintaktičku strukturu predstavljenoga prirodnog jezika. Taj zahtjev također implicira i provjeru prethodnih razina obradbe banke stabala, primjerice razdvajanja teksta na rečenice. Načelno se i za ovaj korak zahtijeva ručna obradba ili ručna provjera strojne obradbe budući da je provođenje sintaktičke analize u pravilu smisleno samo unutar rečeničnih granica. Dodjela parsnih stabala rečenicama i zapisivanje tih parova u računalno čitljivom obliku zahtijeva i jasnu definiciju zapisa banke stabala, odnosno definiciju zapisa toga korpusa na svim razinama – razini tekstova, rečenica i parsnih stabala. Dakle, treba definirati kako se računalno zapisuje korpus i kako se unutar njega računalno obilježavaju njegove rečenice i pripadajuća parsna stabla. Definicija zapisa parsnoga stabla, s druge strane, postavlja pitanja o sintaktičkom formalizmu koji se zapisuje. Kako bi se uopće moglo pristupiti izradi banke stabala nekoga prirodnog jezika, za taj prirodni jezik moraju biti dostupne računalno čitljive formalizacije sintaktičkoga opisa toga jezika. Pojednostavljeno, sva sintaktička pravila toga jezika moraju biti sadržana u nekome računalnom modelu koji je jednoznačno primjenjiv na rečenice toga jezika i zapisiv unutar zapisa korpusa.

Pri izradi banaka stabala, s ozbirom na prethodnu raspravu, postavljaju se načelno sljedeća pitanja (usp. Abeillé 2003:xv):

1. Kako se odabire korpus iz kojega će nastati banka stabala?
2. Kako se odabire sintaktički formalizam i osmišljava njegov računalni zapis?
3. Kako se pristupa sintaktičkoj analizi korpusa?

4. Kako se odabire računalni zapis korpusa i pripadajućih metapodataka, odnosno podataka o rečeničnim granicama, parsnim stablima pojedinih rečenica i mogućim drugim razinama jezičnoga opisa predmetnoga korpusa?

Ovdje se ne raspravlja detaljnije o prvome pitanju, svojstvenom korpusnoj lingvistici i pojašnjenom primjerice u (McEnery i Wilson 2001), već se polazišno pretpostavlja valjan odabir tekstova nekoga jezika za uključivanje u korpus, odnosno pretpostavlja se postojanje računalnoga korpusa toga jezika koji nije sintaktički označen.

Pred računalni zapis banke stabala mogu se postaviti neki okvirni zahtjevi. Zapis mora biti jednoznačan s obzirom na sve dostupne metapodatke (koji se u pravilu odnose na razine jezičnoga opisa toga korpusa) te računalno čitljiv i pretraživ, po mogućnosti s uvažavanjem učinkovitosti čitanja i pretraživanja kod pojedinih primjena. Pod računalnom čitljivošću i pretraživošću pretpostavlja se široki raspon mogućih namjena, pa se odabir računalnoga zapisa banke stabala može uvjetovati i prema budućoj namjeni. Veći broj postojećih specifikacija zapisa računalnih korpusa, primjerice XCES (Ide i dr. 2000) i TEI P5 (Burnard i Bauman 2007)[41] uključuje i razinu sintaktičkoga opisa (usp. Rehbein i van Genabith 2007, Przepiórkowski 2008), a većina je temeljena na jeziku XML[42], takozvanom proširivom jeziku za označavanje teksta (en. *extensible markup language*), čija primjena rezultira tekstom koji je istovremeno čitljiv i čovjeku i računalu.

Kako bi se sintaktička analiza, odnosno parsna stabla pojedinih rečenica mogla zapisati, odnosno pridružiti rečenicama u nekome računalnom zapisu korpusa, potrebno je učiniti zapisivim sintaktički formalizam u skladu s kojim je parsanje izvršeno. Prema ranije danim okvirnim načelima zapisivanja korpusa, i računalni zapis sintaktičke analize rečenica, sadržan u parsnim stablima pojedinih rečenica, mora biti jednoznačan i računalno čitljiv, a također i optimiziran s obzirom na moguće primjene, primjerice treniranje jezičnih modela parsera temeljenih na podatcima. Zapisivanje sintaktičke analize, odnosno njezino uključivanje u neki računalni zapis korpusa tekstova predmetnoga jezika, podrazumijeva ove međuovisne korake: odabir sintaktičkoga formalizma kojim se želi opisati promatrani jezik, izradu računalno čitljivoga zapisa, odnosno modela toga formalizma te eventualno i njegovu prilagodbu na

[41] Ovdje se ne raspravlja o povijesnom razvoju različitih standarda za zapisivanje računalnih korpusa. S obzirom na primjerne navedene standarde XCES i TEI P5, vrijedi pogledati i njihove specifikacije na http://www.xces.org/ i http://www.tei-c.org/index.xml (2012-02-26).

[42] XML je jedna primjena, odnosno jedan profil primjene SGML-a (standardiziranoga uopćenoga jezika za označavanje, en. *standard generalized markup language*), definiranoga međunarodnim standardom ISO 8879 (http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=16387 (2012-02-26)).

zahtjeve računalnoga zapisa korpusa u koji će se uključiti kako bi taj korpus postao banka stabala. Kao ranije, kod pojašnjenja pristupa jezičnomu modeliranju za parsanje, i ovdje se može zaključivati analogijom prema postupku morfosintaktičkoga označavanja. U rečenici danoj u primjeru 2-1 ustanovljeno je kako riječ "Mačka" predstavlja opću imenicu ženskoga roda u nominativu jednine i time je dan njezin *morfosintaktički opis*, odnosno točan opis njezinih morfoloških osobina određenih rečeničnim kontekstom. Međutim, unatoč potpunoj točnosti i jednoznačnosti opisa, zapisivanje tako opisane rečenice iz primjera 2-1 u računalnome obliku, odnosno uključivanje nje i pripadajućega joj morfosintaktičkog opisa u neki računalni korpus bilo bi računalno neučinkovito budući da bi opisi pojedinih riječi zahtijevali puno više računalnoga prostora za pohranu od riječi samih, a također bi bili nedovoljno čitljivi s gledišta strojne obradbe. Stoga se, kako je ilustrirano i primjerom 2-1, za potrebe računalnoga zapisivanja morfosintaktičkoga opisa uvode *morfosintaktičke oznake*, odnosno jednoznačni, računalno čitljivi i optimizirani kodovi pojedinih morfosintaktičkih opisa. Skup svih morfosintaktičkih oznaka namijenjenih označavanju nekoga jezika može se utoliko shvatiti kao kodna tablica kojom se jednoznačno povezuju svi mogući morfosintaktički opisi smisleni za morfologiju (morfosintaksu) toga jezika i njihovi pripadajući kodovi. Jedan primjer takve kodne tablice je standard Multext East (MTE, trenutno važeće inačice 3 i 4, Erjavec 2004, Erjavec 2010)[43], u koju je uključen i hrvatski jezik i koja je korištena upravo u primjeru 2-1. Primjerice, riječ "Mačka", opisana kao opća imenica ženskoga roda u nominativu jednine, nosi prema standardu Multext East morfosintaktičku oznaku "Ncfsn" (treba čitati slova kao kodove: imenica, opća, ženski rod, jednina, nominativ, en. *noun*, *common*, *feminine*, *singular*, *nominative*)[44]. Postupak izrade kodne tablice morfosintaktičkih oznaka mora proizvesti skup oznaka koji uz najmanji utrošak u smislu zahtjeva kodova za pohrambenim prostorom jednoznačno opisuje sve morfosintaktičke opise. S jednakim se načelima pristupa i izradi sintaktičkih oznaka koje predstavljaju kodnu tablicu sintaktičkoga opisa nekoga prirodnog jezika. Međutim, u usporedbi s morfosintaktičkim opisom, sintaktički je opis prirodnoga jezika složeniji – što je ranije ocrtano i u Chomskyjevoj hijerarhiji gdje se morfološke pojave mogu većim dijelom opisati regularnim gramatikama, a sintaktičke se ne mogu u potpunosti opisati ni beskontekstnim gramatikama – pa je složeniji i postupak njegove pretvorbe u optimizirani

[43] Inačica Multext East v4 dostupna je na http://nl.ijs.si/ME/V4/ (2012-02-26).

[44] Ne raspravlja se detaljnije o načelima izrade standarda morfosintaktičkih oznaka, ali vrijedi primijetiti na primjeru 2-1 da je standard Multext East pozicijski, odnosno da se morfosintaktička svojstva pojedinih riječi ovisno o vrsti riječi (zapisanoj prvim slovom oznake) tumače prema položaju slova u oznaci. Primjerice, MTE v4 oznaka za opću imenicu srednjeg roda u nominativu jednine bila bi "Ncnsn" pa se značenje pojedinoga slova tumači ovisno o položaju ("n" na trećem mjestu je "neuter", a na petome "nominative").

skup sintaktičkih oznaka i načela dodjele tih oznaka elementima rečeničnoga ustroja. U primjeru 2-1 vidi se kako je struktura niza morfosintaktičkih oznaka linearna s obzirom na niz riječi u rečenici – svakoj riječi u rečenici uvijek pripada jedna morfosintaktička oznaka. S druge strane, parsno stablo koje predstavlja sintaktičku analizu neke rečenice nije linearna struktura – što je prikazano u primjerima od 2-8 do 2-11 – te pojedini njegovi elementi obuhvaćaju nizove od više riječi. Štoviše, u sintaktičkoj je analizi moguće da grupiranja riječi opisana parsnim stablima nisu nužno slijedna u samoj rečenici. Upravo ovdje prethodno napisana rečenica ("...u sintaktičkoj je analizi moguće...") predstavlja takav primjer budući da je pomoćni glagol dislociran, odnosno umetnut među elemente jednoga imeničnog skupa, a sastavni je element predikata rečenice. S obzirom na ocrtanu složenost, izrada računalno čitljivoga sintaktičkog formalizma za neki prirodni jezik netrivijalan je problem kojim su se jezikoslovci – za potrebe računalnoga modeliranja, ali i za potrebe modeliranja sintaktičkih pojava općenito – sustavno bavili otkad je jezikoslovlje usvojilo znanstvenu metodu (Saussure 1916, usp. Saussure 2002), ali i ranije.

2.1.2.4.2 Gramatika fraznih struktura i ovisnosna gramatika

S obzirom na ciljanu primjenu u jezičnome modeliranju sintaktičkih pojava za potrebe parsanja tekstova prirodnoga jezika, odnosno u sintaktičkome označavanju za potrebe izrade banaka stabala, razlikuju se (usp. Nivre 2006:10, Kübler i dr. 2009:2) dva pristupa izradi sintaktičkih formalizama.

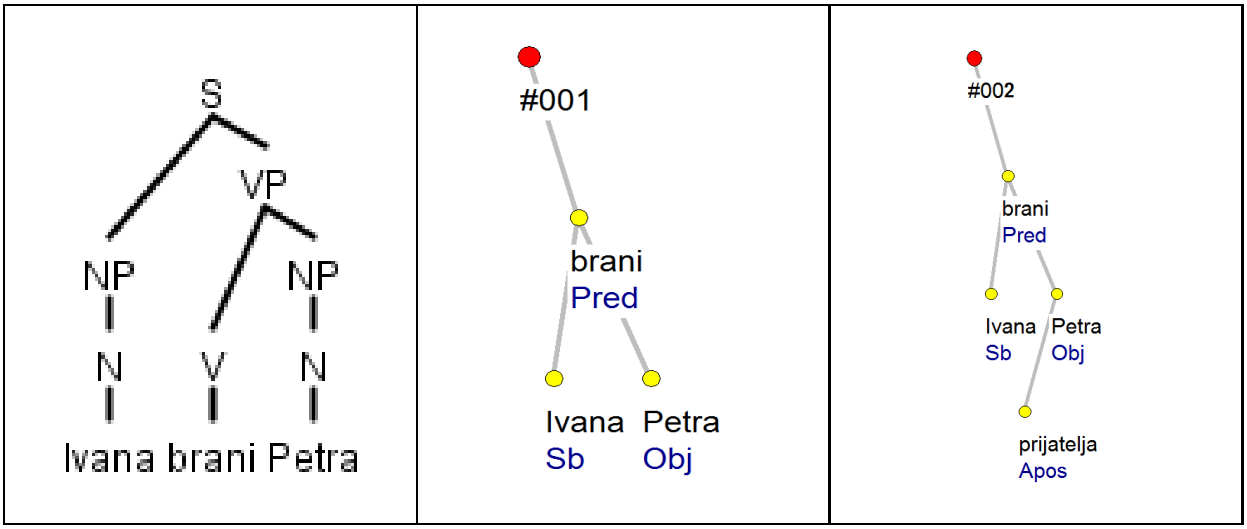
Jedan je zasnovan na koncepciji frazne strukture rečenice, upravo onakve kakva je definirana generativnom (formalnom) gramatikom (Chomsky 1957) i opisana ranije u tekstu, a naziva se *gramatikom frazne strukture* (en. *phrase structure grammar*), a ponekad i *konstituentskom gramatikom* i/li sintaksom (en. *constituency*). Kod gramatike frazne strukture kao modela sintakse prirodnoga jezika – za koju se navodi (usp. Kübler i dr. 2009:2) kako je "najšire korišteni razred sintaktičkih formalizama u teorijskoj i računalnoj lingvistici", prije svega zbog velikoga broja modela engleskoga jezika izrađenih unutar toga razreda – pojedini elementi rečeničnoga ustroja (primjerice, predikat, subjekt, objekt, itd.) opisani su grupiranjima riječi u međuovisne fraze s rastućim brojem sastavnih riječi na koje se potom primjenjuje unaprijed definirana razredba strukturalnih kategorija rečeničnoga ustroja, poput imeničnih fraza i glagolskih fraza, odnosno imeničnih i glagolskih skupova, kako je prikazano ranije u primjeru 2-9. Navodi se (usp. Nivre 2006:10) kako se u ovome modelu rečenične sintakse "rečenica rekurzivno raščlanjuje na manje elemente, koji se nazivaju konstituentima

ili frazama, kojima je obično dodijeljena neka razredba prema unutarnjoj strukturi i koja najčešće uključuje imenične fraze, glagolske fraze, itd.". Postoji veliki broj konkretnih sintaktičkih teorija zasnovanih na općem modelu gramatike fraznih struktura; najčešće korištene među njima sustavno su navedene u (Nivre 2006:10). Model gramatike fraznih struktura (Nivre 2006:10) "potječe iz strukturalističke tradicije predstavljene u (Bloomfield 1933) i (Chomsky 1957)" te je uobičajen u sintaktičkome modeliranju engleskoga i srodnih mu jezika.

Drugi pristup zasnovan je na binarnoj relaciji ovisnosti među riječima u rečenici i ima također "dugu tradiciju u deskriptivnoj lingvistici, posebno u Europi i za europske jezike" (Nivre 2006:10) te nadalje "posebno u domeni sintaktičkoga opisa klasičnih jezika i slavenskih jezika" (Nivre 2006:46). Kod ove skupine sintaktičkih modela rečenica se parsa povezivanjem njezinih riječi "imenovanim binarnim nesimetričnim relacijama, koje se nazivaju ovisnostima i koje su razvrstane prema rečeničnim ulogama ili funkcijama, u kategorije poput subjekta, predikata, objekta i slično". Model rečenične strukture i model sintaktičkoga ustrojstva nekoga prirodnog jezika temeljen na binarnoj nesimetričnoj relaciji ovisnosti među riječima naziva se *ovisnosnom sintaksom* ili *ovisnosnom gramatikom* (en. *dependency syntax*, *dependency grammar*). Navodi se (usp. Nivre 2006:11) da "povijest ovisnosne sintakse seže u antiku", no za polazišnu točku njezina suvremenog razvoja, na kojoj su dalje razvijani opisi pojedinih jezika, posebno europskih i, još posebnije, slavenskih jezika, uzima se (Tesnière 1959, usp. Šojat 2008:10). S obzirom na raznolikost skupina jezika za koje je sintaksa opisivana razredom ovisnosnih gramatika, postoji i veliki broj različitih teorija koje pripadaju tomu razredu. Detaljan popis dan je u (Nivre 2006:50). Ovdje se ne raspravlja detaljno o ovisnosnome opisu sintaktičkih pojava s gledišta jezikoslovlja, pa se ne analiziraju pojedine teorije, već se pod ovisnosnim pristupom i ovisnosnom sintaksom podrazumijeva – osim ako je naznačeno drugačije – čitav razred teorija kojima je zajednička osnova, odnosno vezivanje leksičkih jedinica u ovisnosni odnos imenovan sintaktičkom funkcijom.

Zapis sintaktičkoga opisa rečenice u okviru razreda gramatika frazne strukture rezultira parsnim stablom koje se naziva *fraznim stablom* ili *konstituentskim stablom*. Banka stabala koja sadrži takva stabla može se nazivati bankom fraznih stabala ili bankom konstituentskih stabala, ali se – ponajviše iz povijesnih razloga budući da su prve banke stabala bile upravo konstituentske i razvijene za engleski jezik (usp. Abeillé 2003) – najčešće naziva samo bankom stabala. Najpoznatiji predstavnik ove skupine banaka stabala je banka stabala

engleskoga jezika, naziva *Penn Treebank* (Marcus i dr. 1993)[45]. S druge strane, ukoliko je rečenica sintaktički opisana (ili zapisana) uspostavljanjem imenovanih binarnih relacija ovisnosti među njezinim riječima, rezultirajuće se stablo naziva *ovisnosnim stablom* (en. *dependency tree*), a banka stabala koja sadrži rečenice i pripadajuća ovisnosna stabla naziva se *bankom ovisnosnih stabala* ili *ovisnosnom bankom stabala* (en. *dependency treebank*)[46]. Praška ovisnosna banka stabala (en. *Prague Dependency Treebank*, *PDT*, Hajič i dr. 2000) najpoznatiji je predstavnik ove skupine banaka stabala[47]. Primjer 2-12 prikazuje frazno stablo i ovisnosno stablo rečenice hrvatskoga jezika iz primjera 2-9 kao jednakovrijedne zapise njezina sintaktičkog opisa[48].



Primjer 2-12 Frazno stablo i ovisnosno stablo kao sintaktički opis rečenice

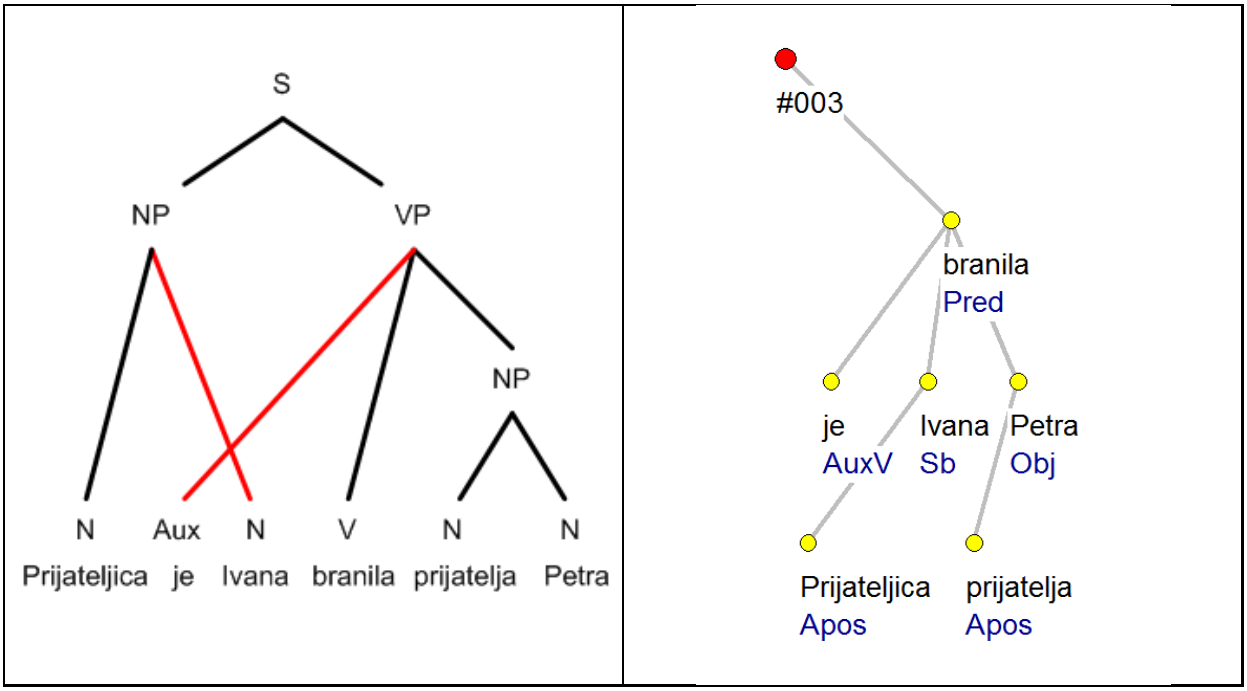
Osnovna razlika između fraznoga (konstituentskoga) i ovisnosnoga opisa (i zapisa, u skladu s ranije postavljenom razlikom između opisa i oznake) sintaktičke strukture može se izvesti upravo iz primjera 2-12. U fraznom je stablu eksplicitno prikazana frazna struktura rečenice, odnosno podjela na imenični i glagolski skup te pripadajuće daljnje podjele tih fraza na sastavne podfraze. S druge strane, uloge pojedinih riječi i fraza u rečenici s gledišta njezina gramatičkog ustroja dana je samo implicitno budući da parsno stablo ne sadrži konkretne oznake subjekta, predikata i objekta. Te kategorije pohranjene su u parsnome stablu implicitno, preko strukture na prvoj razini parsnoga stabla (odnosno razini razdjeljivanja, en.

[45] Vidjeti i URL http://www.cis.upenn.edu/~treebank/ (2012-03-03).
[46] Primjetno je kako je u engleskome jeziku izraz *treebank* nastao po načelu jezične ekonomičnosti iz *tree* i *bank* pa tako u izrazu *dependency treebank* zapravo *dependency* pobliže označava *tree*, iako implicitno označava i skup takvih stabala, odnosno *treebank*. Utoliko je izravno ispravno reći *banka ovisnosnih stabala*, iako je implicitno točno i *ovisnosna banka stabala*. Ovdje se koriste i smatraju jednakovrijednima oba izraza.
[47] Vidjeti i URL http://ufal.mff.cuni.cz/pdt2.0/ (2012-03-03).
[48] Ovo i iduća ovisnosna stabla u ovakvome grafičkome obliku, osim ako je naznačeno drugačije, izrađena su s pomoću računalnoga paketa TrEd, autora Petra Pajasa (v. http://ufal.mff.cuni.cz/tred/).

chunking, usp. Vučković 2009) budući da se njome implicira, primjerice, da je eksplicitno opisana imenična fraza (NP) zapravo subjektni skup, pa je u njemu sadržan subjekt te da je glagolska fraza (VP) predikatni skup koji sadrži predikat. Suprotno tomu, u ovisnosnome su stablu eksplicitno navedene rečenične uloge ili službe pojedinih riječi, a frazna je struktura prikazana implicitno. Ovisnosno stablo dane rečenice izgrađeno je s pomoću triju binarnih relacija ovisnosti među njezinim riječima, koje se mogu predstaviti u obliku uređenih trojki koje povezuju dvije riječi sintaktičkom ulogom ili sintaktičkom funkcijom koja opisuje odnos prve prema drugoj riječi, odnosno opisuje prirodu njihove međuovisnosti: ("Ivana", "brani", Sb), ("Petra", "brani", Obj), ("brani", rečenica, Pred). Pritom oznake "Sb", "Obj" i "Pred" predstavljaju kodove sintaktičkih funkcija subjekta, objekta i predikata. Utoliko je ovisnosnim stablom eksplicitno navedeno što je u rečenici subjekt, što objekt, a što predikat, odnosno eksplicitno su definirani u njoj ostvareni elementi rečeničnoga ustroja, onako kako su opisani u ranijim poglavljima. S druge strane, frazna struktura rečenice je implicitna, utoliko što riječi u rečenici nisu eksplicitno grupirane u fraze, već se ta grupiranja iz ovisnosnoga stabla rekonstruiraju spustom niz granu stabla označenu nekom sintaktičkom funkcijom, odnosno od više k nižoj razini apstrakcije izvedene sintaktičkim funkcijama. Budući da je rečenica iz primjera 2-12, za koju je usporedno dano frazno i ovisnosno stablo, prejednostavna za tu ilustraciju, popraćena je proširenom inačicom u kojoj je objektu pridružena imenica koja ga dodatno opisuje.

U tome se primjeru može vidjeti kako je, na višoj razini apstrakcije, objekt rečenice svaka riječ koja se nalazi u grani stabla gdje je prvi put uvedena oznaka objekta. Dakle, objekt rečenice je "prijatelja Petra". Na nižoj razini apstrakcije ovisnosna relacija ("prijatelja", "Petra", Apos) pojašnjava da je objekt pobliže označen apozicijom. S obzirom na opisivanje redoslijeda uvođenja riječi u rečenicu – odnosno načelo otvaranja mjesta pojedinim elementima rečeničnoga ustroja u kojem, kako je ranije raspravljeno, predikat otvara mjesto ostalim ustrojbenim elementima, a samostalni elementi uvode u rečenicu ovisne elemente – ovisnosno stablo, opet, za razliku od fraznoga stabla, nudi eksplicitan opis načina uvođenja. Pritom sama priroda ovisnosti upućuje na činjenicu da u takvome uređenju postoji podređeni i nadređeni element, odnosno onaj koji ovisi i onaj o kojem je neki element ovisan. O tome se dodatno raspravlja kasnije u tekstu. U primjeru 2-12, dakle, jasno se može pročitati da predikat "brani" otvara mjesto subjektu "Ivana" i objektu "Petra".

Osim navedenih razlika između frazne i ovisnosne strukture, koje su zapravo formalne prirode – budući da se svaka implicitnost pojedinih elemenata opisa sadržanih u strukturi parsnih stabala, poput sastavnih riječi u elementima rečeničnoga ustroja ili uloge pojedinih riječi u tim elementima, može nekim računalnim modelom pretvoriti u eksplicitnost – postoji i jedna temeljna razlika. Ona je opisana primjerom 2-13.



Primjer 2-13 Frazno i ovisnosno parsno stablo rečenice s umetanjem

U primjeru je jednostavna rečenica hrvatskoga jezika, proširena i izmijenjena u skladu s ranijim primjerima, koja glasi "Prijateljica je Ivana branila prijatelja Petra". Ova rečenica je jednostavna ilustracija relativno slobodnoga redoslijeda riječi u rečenicama hrvatskoga jezika budući da je u njoj pomoćni glagol "biti", kao sastavni dio predikata u "je branila", dislociran u odnosu na glavni glagol "braniti", odnosno umentut između dviju imenica koje sačinjavaju subjekt "Prijateljica Ivana". S obzirom na činjenicu da frazno parsno stablo predstavlja model rečenične strukture u kojemu su fraze, odnosno podskupovi rečenice uređeni kontinuirano, odnosno bez prekida u strukturi pojedinih fraza, njime je kao modelom sintaktičke strukture otežan prikaz strukture s prekidom, kao u "Prijateljica je Ivana branila", gdje se imenična i glagolska fraza preklapaju. Budući da je opis sintaktičkoga ustroja s pomoću frazne strukture povijesno vezan uz modeliranje i parsanje engleskoga jezika za koji je karakterističan fiksni redoslijed riječi u rečenici, on nije predviđen ni namijenjen za opisivanje pojave relativno ili potpuno slobodnoga redoslijeda riječi u rečenici (usp. Covington 1990, Nivre 2006:66). Iz toga se može zaključiti da je frazna struktura prikladnija za parsanje tekstova jezika s fiksnim

poretkom riječi (poput engleskoga jezika), dok je ovisnosna struktura i ovisnosna sintaksa bolji izbor[49] za opis rečenica onih jezika sa slobodnim ili relativno slobodnim poretkom riječi (poput slavenskih jezika, odnosno hrvatskoga jezika). S obzirom na isključivu usmjerenost ovoga istraživanja parsanju tekstova hrvatskoga jezika, dalje se razmatra isključivo ovisnosna gramatika, odnosno ovisnosni pristup modeliranju sintaktičkih pravila. Dalje u tekstu, dodatno će se opravdati ili implicirati opravdanost odabira ovisnosnoga pristupa sintaksi kao formalizma za parsanje tekstova hrvatskoga jezika, i to prvenstveno s obzirom na ranije postavljene formalne zahtjeve i kriterije za vrjednovanje parsera.

2.1.2.4.3 Ovisnosna sintaksa i ovisnosno parsanje

Ovisnosna sintaksa – kao sintaktički formalizam općenito te s primjenom u izradi banaka stabala – zasnovana je na pojmu ovisnosti. Prema (Tesnière 1959), prevedenome u (Nivre 2006:47), on je definiran na sljedeći način: "Rečenica je ustrojena cjelina, a riječi su njezine sastavnice. Uključivanjem u rečenicu svaka riječ prestaje biti izdvojena cjelina poput unosa u rječniku. Um među riječima u rečenici uočava veze, a ukupnost svih tih veza oblikuje strukturu rečenice. Strukturne veze među riječima uspostavljaju među njima ovisnosni odnos. Načelno, svaka ovisnosna veza povezuje nadređenu riječ s podređenom riječju. Primjerice, u rečenici *Alfred govori...*, riječ *govori* je nadređena, a *Alfred* podređena riječ".

Ovisnosna sintaktička analiza rečenica nekoga jezika stoga se svodi na uspostavljanje relacije ovisnosti među riječima u rečenici, i to na takav način da u svakoj relaciji postoji nadređena i podređena riječ. Nadređena se riječ najčešće naziva *glavom*, a podređena riječ *dependentom*, iako postoji i niz sličnih izraza (*upravitelj-modifikator* ili *regens-dependens*, usp. Nivre 2006:47). Ključno pitanje izrade ovisnosnoga modela sintakse nekoga prirodnog jezika jesu opća pravila prema kojima se određuju ovisnosni odnosi – odnosno podređenost i nadređenost – i pripadajuće sintaktičke funkcije među riječima u rečenici. Iako nisu u užem zanimanju ovoga rada, ovdje su prema (Nivre 2006:48) iznesena neka od tih načela[50].

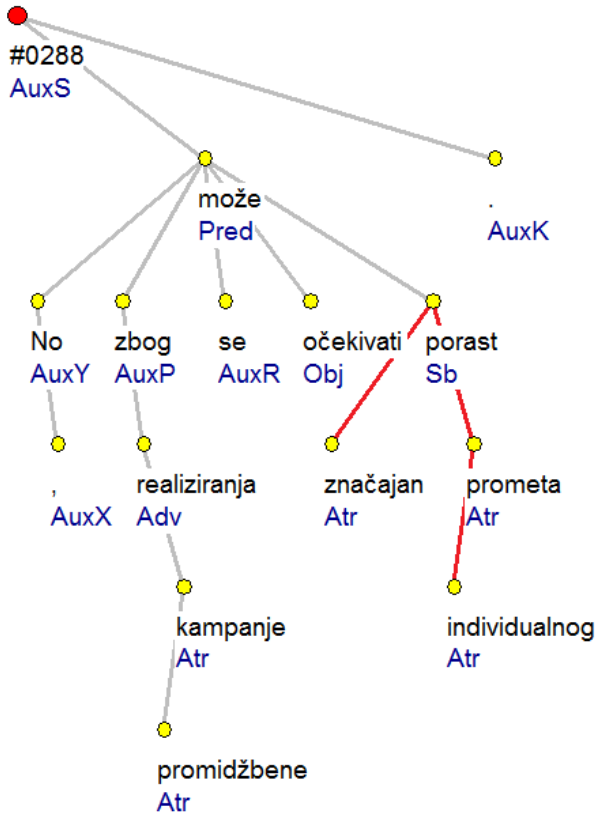
Neka je C neki podskup ili fraza rečenice R koja sadrži ovisnosnu relaciju između nadređene riječi ili glave H te podređene riječi ili dependenta D. Vrijedi sljedeće.

1. H određuje sintaktičku funkciju od C i često može zamijeniti C u rečenici.

[49] Iz parsnoga stabla iz primjera 2-13 ne vidi se izvorni redoslijed riječi u rečenici. On se ipak može lako pronaći ako se pojedina riječ indeksira rednim brojem koji predstavlja njeno mjesto u rečenici.
[50] Točka 2 iz (Nivre 2006:48) ovdje je rastavljena na dvije točke zbog dodatnoga pojašnjenja uloge riječi D u semantičkom ustrojstvu izraza C i rečenice R.

2. H određuje semantičku ulogu od C.
3. D dodatno, odnosno pobliže opisuje već određenu semantičku ulogu od C.
4. H je obavezan element u C i R, a D nije.
5. H uvodi D u rečenicu i određuje je li D obvezan ili ne.
6. Pojavnost riječi D s obzirom na morfosintaksu ovisi o pojavnosti riječi H.
7. Položaj riječi D u slijednom ustrojstvu C i R ovisi o položaju riječi H.

Dakle, kriteriji za određivanje glave i dependenta mogu biti morfosintaktički, sintaktički i semantički, a također su usklađeni s prethodno raspravljanim općim načelima izgradnje elemenata rečeničnoga ustroja. Rasprava o pojedinim kriterijima s obzirom na pojedine razine jezičnoga opisa može se pronaći u (Nivre 2006:48) i pojedinim teorijama ovisnosne sintakse (usp. Nivre 2006:50).



No, zbog realiziranja promidžbene kampanje može se očekivati značajan porast individualnog prometa.

Primjer 2-14 Glave i dependenti u ovisnosnim relacijama

Neka od zadanih pravila ilustrirana su primjerom 2-14. Primjer predstavlja ovisnosno stablo rečenice hrvatskoga jezika koja glasi "No, zbog realiziranja promidžbene kampanje može se očekivati značajan porast individualnog prometa". U svrhu pojašnjavanja prethodno navedenih pravila za uspostavljanje relacije ovisnosti među riječima može se uzeti, primjerice, fraza "značajan porast individualnog prometa". Štoviše, u svrhu jednostavnosti i ilustrativnosti, može se iz nje izdvojiti samo fraza "značajan porast".

Ako se odabrana fraza promatra s gledišta gore navedenih sedam pravila za određivanje glave i dependenta u ovisnosnoj relaciji, može se zaključiti sljedeće.

1. U frazi "značajan porast" samo "porast" može zamijeniti čitavu frazu u rečenici. Zamjena izraza riječju "značajan" čini rečenicu negramatičnom i neobavijesnom u odnosu na njezin izvorni oblik.
2. Riječ "porast" određuje osnovu obavijesnosti izraza "značajan porast", dok je riječ "značajan" dodatno opisuje. (Ovdje su na jednome mjestu povezana pravila 2 i 3 iz prethodnoga popisa.)
3. Riječ "porast" ne može se izostaviti iz rečenice, dok riječ "značajan" može. Prva je stoga obavezan element rečenice, a druga nije.
4. Riječ "značajan" uvedena je u rečenicu preko riječi "porast" budući da se radi o atributu subjekta, odnosno neobaveznome elementu rečeničnoga ustroja.
5. Morfosintaktički oblik pridjeva "značajan" ovisan je o morfosintaktičkome obliku imenice "porast". Budući da se radi o atributu i subjektu, oni se moraju slagati i slažu se u rodu, broju i padežu.
6. Položaj riječi "značajan" ovisan je o položaju riječi "porast" budući da se radi o njezinu atributu. S obzirom na ulogu prenošenja obavijesti, poželjan je njegov položaj što bliže subjektu, no može se zamisliti i rečenica oblika "značajan se može očekivati porast" u kojoj je atribut također vezan uz subjekt, iako je dislociran.

S obzirom na određenje s pomoću sedam pravila za pronalaženje glave i dependenta, može se zaključiti da je u rečenici iz primjera 2-14 riječ "porast" glava dvočlanoga izraza "značajan porast", dok je riječ "značajan" o njoj ovisna, pa stoga predstavlja dependent. Slično vrijedi i za preostale elemente u širem izrazu "značajan porast individualnog prometa" budući da izraz "individualnog prometa" dodatno određuje riječ "porast", slično kao što u podizrazu "individualnog prometa" riječ "individualnog" predstavlja dodatnu odrednicu, odnosno atribut riječi "prometa". Stoga ta grana ovisnosnoga stabla rečenice iz primjera 2-14

(označena crvenom bojom) ima kao glavu riječ "porast", a ostale je riječi pobliže označavaju, i to sa sintaktičkom funkcijom atributa ("Atr").

Izrada banke stabala prema pravilima ovisnosne sintakse, odnosno prema sintaktičkome modelu zasnovanom na relaciji ovisnosti među riječima, uključuje sljedeće preduvjete.

1. Uz granice rečenica, u polazišnome korpusu moraju biti označene i granice riječi budući da će se ovisnosne relacije uspostavljati među riječima. Mora biti odabran zapis korpusa koji podržava razgraničenje rečenica i riječi te zapisivanje ovisnosnih relacija među riječima.
2. Mora biti definiran konačan skup svih sintaktičkih funkcija koje se mogu dodijeliti pojedinim ovisnosnim relacijama. Sintaktičke funkcije moraju biti u potpunosti usklađene s pripadajućom sintaktičkom teorijom zasnovanom na relaciji ovisnosti. Sintaktička funkcija unutar ovisnosne relacije pritom uvijek predstavlja funkciju dependenta u odnosu na glavu relacije, odnosno pobliže opisuje prirodu ovisnosti dependenta o glavi.
3. Moraju biti postavljena pravila koja osiguravaju da se ovisnosnom analizom bilo koje rečenice kao rezultat dobije valjano parsno stablo. Valjanost parsnoga stabla, osim točnosti u skladu sa sintaktičkom teorijom, znači i valjanost stablaste strukture, odnosno povezanost svih riječi i postojanje samo jednoga korijenskog čvora.

Postojanje modela ovisnosne sintakse nekoga prirodnog jezika, odnosno na ovisnosnoj relaciji zasnovanoga i računalno izvedivoga modela sintaktičkih pojava toga jezika, preduvjet je za izgradnju ovisnosne banke stabala toga prirodnog jezika. Prema ranije navedenome, s druge strane, postojanje banke stabala nad tekstovima nekoga jezika preduvjet je za modeliranje i izvedbu na podatcima temeljenih parsera toga prirodnog jezika. Parser prirodnoga jezika koji koristi jezični model zasnovan na ovisnosnoj sintaksi naziva se *ovisnosnim parserom* (en. *dependency parser*). Dalje u tekstu, formalizira se problem ovisnosnoga parsanja – uz formalnu definiciju relacije ovisnosti, parsnoga stabla i samoga parsanja – te se potom opisuju neki značajni pristupi ovisnosnom parsanju temeljeni na podatcima, odnosno na treniranju stohastičkoga jezičnog modela nad bankom ovisnosnih stabala.

2.1.3 Ovisnosno parsanje kao optimizacijski problem

Ovdje se formalno opisuje problem ovisnosnoga parsanja tekstova prirodnoga jezika kao polazišna točka za određivanje različitih računalnih modela ovisnosnih parsera temeljenih na podatcima iz banaka stabala. Preuzimaju se ranije definicije riječi, rečenice i teksta pa se na njima gradi definicija relacije ovisnosti među riječima, ovisnosnoga grafa i ovisnosnoga stabla, uz navođenje bitnih formalnih svojstava ovisnosnih stabala s obzirom na opće gledište ovisnosne teorije sintakse. S pomoću tih pojmova postavlja se formalna definicija ovisnosnoga parsanja kao optimizacijskoga problema.

S obzirom na ranije navedena svojstva ovisnosnoga modela sintakse, sve postavljene formalne definicije slijedit će (usp. Nivre 2006:67) sljedeća opća načela, odnosno ograničenja koja su implicirana tim općim načelima. Budući da se ovdje raspravlja o parsanju tekstova prirodnoga jezika kao problemu obradbe prirodnoga jezika, ne uzimaju se eksplicitno u obzir moguće implikacije tih općih načela ili ograničenja na pojedine ovisnosne sintaktičke teorije s gledišta jezikoslovlja. Od definicija se zahtijeva sljedeće.

1. Ograničenost parsanja, odnosno sintaktičke analize na ovisnosnu strukturu. Pritom se dopušta parserima za izvođenje ovisnosne strukture koristiti ostale potencijalno dostupne podatke iz banke stabala na kojoj treniraju model.
2. Parsanjem se svakoj ulaznoj rečenici – u skladu s prethodno određenim kriterijem robustnoga razrješivanja višeznačnosti – dodijeli samo jedno ovisnosno stablo.
3. Ovisnosno stablo definirano je kao skup čvorova i veza među njima. Čvorovi ovisnosnoga stabla su riječi ulazne rečenice. Ne postoji nijedan čvor ovisnosnoga stabla koji nije riječ ulazne rečenice, osim korijenskoga čvora koji predstavlja rečenicu samu.
4. Veze među čvorovima ovisnosnoga stabla predstavljaju relacije ovisnosti među riječima u rečenici. Svaka veza među čvorovima definira relaciju ovisnosti između dvije riječi, a također – ovisno o nadređenosti, odnosno podređenosti tih dviju riječi, određene implicitno i vizualnom podređenošću u ovisnosnome stablu – prirodu ovisnosti među tim riječima.
5. Priroda ovisnosti među riječima predstavlja sintaktičku funkciju podređene prema nadređenoj riječi, odnosno dependenta prema glavi. Sintaktičke funkcije biraju se iz konačnoga skupa u skladu s odabranom sintaktičkom teorijom i pripisuju se vezama

među čvorovima ovisnosnoga stabla. Svakoj vezi mora biti pripisana točno jedna sintaktička funkcija.

6. Svako ovisnosno stablo mora imati korijenski čvor. Korijenski čvor je, kako je prethodno navedeno, jedini čvor koji nije predstavljen nekom riječju iz rečenice, odnosno jedini meta-čvor.
7. U svakome ovisnosnom stablu mora postojati neka putanja vezama s pomoću koje se može dohvatiti svaki čvor stabla. Drugim riječima, svakoj riječi u rečenici mora biti dodijeljena jedna sintaktička funkcija.
8. Definicije koje su ovdje ocrtane i koje slijede dalje u tekstu ne bave se problemima sintaktičkoga modeliranja prirodnoga jezika u okvirima ovisnosne sintakse. Njihova je svrha definirati – odnosno osigurati preduvjete za njihovo definiranje – pristupe parsanju tekstova prirodnoga jezika izgradnjom parsera iz sintaktički obilježenih podataka sadržanih u banci ovisnosnih stabala toga prirodnog jezika.

S obzirom na navedene napomene, može se zaključiti da parsanje tekstova prirodnoga jezika u ovim okvirima podrazumijeva i određeni broj dodatnih preduvjeta. Primjerice, nužan preduvjet za rad ocrtanoga parsera – uz dostupnost banke stabala, odabir valjanoga modela parsanja i ostale čimbenike o kojima se raspravlja dalje u tekstu – jest razdvajanje ulaznoga teksta na rečenice i riječi, što je u modelu parsera sa slike 2-4 ilustrirano modulom za predobradbu. Dalje u tekstu pojedini pristupi ovisnosnomu parsanju postavljat će dodatne specifične zahtjeve ovoga tipa od modula za predobradbu, ulaznih podataka modula za treniranje i modula za testiranje te zahtjeve od samoga jezičnog modela. Prije te rasprave slijede definicije koje vode formalizaciji problema ovisnosnoga parsanja.

2.1.3.1 Tekst, rečenica i riječ

Definicija rečenice i riječi već je postavljena pri opisu formalne gramatike, a definicija teksta implicirana je pri postavljanju općih zahtjeva i kriterija za odabir parsera prirodnoga jezika. Ovdje se riječ i rečenica ponovno definiraju, ovaj put s gledišta parsanja teksta, a ne parsanja gramatikom, dok se implicitna definicija teksta zapisuje eksplicitno.

Tekst je niz rečenica $T = (s_1, \ldots, s_n)$. Ova definicija teksta pretpostavlja, za potrebe izgradnje ovisnosnoga parsera, prethodnu razdvojenost teksta na rečenice. Ovdje se ne raspravlja o tome kako je na računalu izvedeno to razdvajanje, već se pretpostavlja njegova najveća moguća (poželjno potpuna) točnost.

Rečenica je niz riječi $s = (w_1, \ldots, w_n)$. Kao kod definicije teksta, ovdje se pretpostavlja prethodna razdvojenost ulazne rečenice na riječi.

Riječ je niz slova $w = (c_1, \ldots, c_n)$. Razdvojenost riječi na slova smatra se ovdje na neki način trivijalnom, pa se o njoj također ne raspravlja. Pretpostavlja se da ona proizlazi iz razdvojenosti rečenice na riječi i činjenice da su metode zapisa slova i ostalih znakova na digitalnom računalu dobro poznate. U ovoj definiciji riječ je opisana kao neka pojavnost u tekstu, pa se stoga radi o *pojavnici* (en. *token*, *word form*), a ne o polaznom obliku neke riječi, odnosno *lemi* (en. *lemma*) ili rječničkoj natuknici. Osim samih pojavnica, odnosno pojavnih oblika riječi, ova definicija riječi pokriva također i interpunkciju te moguće druge posebne znakove koji se mogu pojaviti u tekstovima. S druge strane, ovako neograničavajuća definicija riječi dopušta modulima za predobradbu ulaznoga teksta grupiranje riječi u višerječne jedinice (en. *multi-word units*, *MWU*) ili razdvajanje riječi na morfosintaktički smislene podrazine (usp. Nivre 2006:68). Definicija je s razlogom široka budući da se može zamisliti korisnost treniranja ili korištenja parsera nad višerječnim jedinicama (budući da višerječne jedinice mogu imati dijeljenu sintaktičku funkciju preuzetu od njezine glave, kako je prethodno pojašnjeno), kao i nad jedinicama koje sačinjavaju riječi (budući da, primjerice, prefiksi i sufiksi mogu sintaktički dodatno opisivati osnovni oblik neke riječi).

(Nivre 2006:68) također definira i dvije pomoćne funkcije koje preslikavaju (1) indekse riječi u rečenici u same riječi, odnosno (2) indekse riječi u rečenici u metapodatke koji su tim riječima pridruženi. Prva funkcija definirana je kako slijedi.

$$w_s(i) = \begin{cases} w_i, 1 \leq i \leq n \\ \text{nedefinirano u protivnom} \end{cases}$$

Riječima mogu biti pridruženi brojni metapodatci, no ovdje se uvodi ograničenje na razinu s osnovnim oblicima riječi (razinu lematizacije, odnosno svođenja na osnovni oblik) i razinu koja sadrži morfosintaktičke oznake pojedinih riječi (usp. Tadić 1994, Tadić 2003, Agić i dr. 2009). Funkcije vraćaju vrijednost – riječ ili oznaku te riječi na nekoj razini jezičnoga opisa – pod zadanim indeksom, ako je indeks u granicama duljine rečenice (odnosno broja riječi u rečenici), a izvan tih granica nisu definirane. Korisnost tih funkcija u kasnijem modeliranju parsera očituje se kod modela koji crpe znanje iz metapodataka, odnosno povezuju sintaktičke funkcije pojedinih riječi s njihovim kontekstom koji se potencijalno ostvaruje (i implicira ih) u metapodatcima poput morfosinaktičkih kategorija, a ne samo u pojavnostima samih riječi. Za dohvat svake razine metapodataka preko indeksa

riječi kojoj pripadaju definira se jedna nova funkcija nalik prethodno definiranoj za dohvat riječi preko indekasa. Primjerice, (Nivre 2006:68) daje funkciju za dohvat podataka o vrstama riječi (morfosintaktičkih oznaka) kako slijedi, ovdje uz neke izmjene.

$$\mathrm{msd_s(i)} = \begin{cases} \mathrm{msd_i}, 1 \leq \mathrm{i} \leq \mathrm{n} \\ \text{nedefinirano u protivnom} \end{cases}$$

Ovakva definicija implicira da je svaka riječ u rečenici popraćena odgovarajućom morfosintaktičkom oznakom. To se može tumačiti dvojako: (1) svaka riječ u rečenici može biti predstavljena kao uređeni par koji se sastoji od pojavnice i njezine morfosintaktičke oznake ili se (2) može smatrati da uz razinu pojavnica kojima se definiraju rečenice i kojima se onda definira tekst, postoji i dodatna, skrivena ulazna razina u kojoj se propagiraju leme, morfosintaktičke oznake i ostali metapodatci. Budući da su riječi, rečenice i tekstovi već određeni definicijama, ovdje se pretpostavlja model u kojemu se metapodatci smatraju dodatnim ulaznim podatcima za module parsera, iako su u praksi oni u pravilu uključeni u ulazni tekst preko nekoga standarda za zapisivanje korpusa, pa bi se utoliko ulazni podatci mogli smatrati višedimenzionalnima, uz broj dimenzija definiran brojem razina metapodataka kojima je opisana nulta razina ulaza, odnosno sami podatci.

2.1.3.2 Ovisnosni graf i ovisnosno stablo

Graf je uređeni par $\mathrm{G} = (\mathrm{V}, \mathrm{E})$, gdje je V skup čvorova, a E skup veza među parovima čvorova, odnosno skup dvočlanih podskupova iz V (usp. Cormen i dr. 2009:589), $\mathrm{E} \subseteq \mathrm{V} \times \mathrm{V}$.

Ovisnosni graf je označeni usmjereni graf na koji se kasnije uvode dodatna ograničenja kako bi ga se učinilo ovisnosnim stablom (usp. Nivre 2006:69). Čvorovi ovisnosnoga grafa su indeksi koji predstavljaju riječi u rečenici. Označenost pritom znači da je svakoj vezi dodijeljena neka oznaka iz skupa oznaka R koji predstavlja vrste ovisnosti, odnosno skup svih mogućih sintaktičkih funkcija prema zadanome sintaktičkom formalizmu. Usmjerenost grafa znači da veze imaju usmjerenje[51] ("strjelicu") koja u ovome slučaju predstavlja smjer relacije ovisnosti od depedenta prema glavi. Ovisnosni je graf, dakle, označeni usmjereni graf $\mathrm{G} = (\mathrm{V}, \mathrm{E}, \mathrm{L})$, gdje je pritom $\mathrm{V} = \mathbb{Z}_{\mathrm{n}+1}$, odnosno skup svih indekasa[52] riječi u rečenici i

[51] Ako se pretpostavi da neki element uređenoga para, ovisno o položaju, predstavlja glavu izraza, onda graf ne mora biti usmjeren, odnosno, usmjeren je implicitno pa ga se ne modelira poput usmjerenoga grafa.
[52] (Nivre 2006:70) napominje kako je korisno umjesto riječi u rečenici rabiti pripadajuće im indekse budući da se onda redoslijed riječi u rečenici može matematički modelirati operatorima usporedbe koji se inače koriste nad skupom cijelih brojeva. Tako se zna da, primjerice, riječ s indeksom j slijedi nakon riječi s indeksom i ako i

nultoga indeksa[53] koji predstavlja korijenski čvor, $E \subseteq V \times V$ kao i ranije, a $L: E \longrightarrow R$ funkcija koja dodjeljuje svakoj vezi oznaku u vidu neke od sintaktičkih funkcija. Skup svih veza E definiran je tako da svaki element predstavlja uređeni par dvaju indekasa riječi iz V, takav da je $\forall e \in E, e = (i, j), i, j \in V$, i to tako da prvi element uređenoga para predstavlja u kontekstu ovisnosne relacije glavu, a drugi element dependenta.

Ovisnosni graf G je dobro oblikovan ako i samo ako zadovoljava svojstvo početnosti korijenskoga čvora i svojstvo povezanosti.

1. Svojstvo početnosti korijenskoga čvora kaže da čvor indeksiran indeksom 0 mora biti korijenski čvor. Dakle, nulta riječ rečenice, koja se označava i kao w_0, je meta-riječ koja predstavlja samu rečenicu. (S obzirom na ovisnosnu sintaksu, pokazat će se kasnije kako se na nultu riječ, odnosno korijenski čvor načelno veže samo predikat budući da predikat otvara mjesto svim drugim sintaktičkim elementima rečenice.)
2. Svojstvo povezanosti kaže da svaki čvor ovisnosnoga grafa mora biti dohvatljiv nekom putanjom ovisnosnih veza. Ovdje se pritom misli na neusmjerenu putanju što predstavlja slabu povezanost, za razliku od jake kod koje putanja preko koje se čvorovi dohvaćaju mora biti usmjerena, odnosno ostvarena poštujući oznaku smjera na vezama.

S obzirom na definiciju skupa svih veza E i skupa sintaktičkih funkcija (ili ovisnosnih relacija) R, svaka od veza iz E može se dodatno opisati i s pomoću neke relacije iz R tako da zapis oblika $i \xrightarrow{r} j$ znači da u ovisnosnome grafu postoji ovisnosni odnos $r \in R$ između riječi $i, j \in V$ koji je opisan vezom $e \in E, e = (i, j)$ i funkcijom $L: E \longrightarrow R, L(e) = r$. Notacija nalik ovoj može se koristiti i neovisno o relaciji, odnosno sintaktičkoj funkciji, pa se može reći i samo da su dvije riječi u ovisnosnome odnosu, $i \rightarrow j$, dakle, bez točnoga navođenja sintaktičkih svojstava toga ovisnosnog odnosa, ali uz navođenje uloge glave i dependenta. Zbog jednostavnosti, ovdje se taj zapis smatra, izostavljanjem funkcije za obrnuto

---

samo ako je $i < j$. Međutim, ovdje se skup riječi i skup pripadnih indekasa naizmjenično koriste pod istom oznakom V.

[53] Također se može razlikovati (usp. Nivre 2006:70) i skup svih čvorova i skup svih čvorova osim meta-čvora koji predstavlja korijenski čvor, odnosno samu rečenicu. Notacijski, označava se V skupom svih čvorova, dakle, $|V| = n + 1$ te V^+ skupom svih čvorova koji predstavljaju stvarne riječi rečenice, bez korijenskoga ili meta-čvora, odnosno $|V^+| = n$.

indeksiranje, jednakovrijednim zapisu $w_i \overset{r}{\rightarrow} w_j$, odnosno $w_i \rightarrow w_j$ ako se želi umjesto indekasa koristiti same riječi.

Ovisnosno stablo je svaki dobro oblikovani ovisnosni graf $G = (V, E, L)$ koji je ujedno i usmjereno stablo s korijenskim čvorom indeksa 0 (usp. Kübler i dr. 2009:13). U teoriji grafova (usp. Cormen i dr. 2009) stablo je graf koji zadovoljava svojstvo povezanosti, koje je opisano ranije, i svojstvo acikličnosti. Svojstvo acikličnosti kaže da svi čvorovi u grafu moraju biti povezani upravo tako da su svaka dva čvora povezana samo jednom putanjom bez ponavljanja čvorova. Drugim riječima, svaka dva čvora moraju biti povezana neciklično, odnosno moraju biti onemogućene kružne putanje. Dakle, ovisnosno stablo je svaki dobro oblikovani ovisnosni graf koji je ujedno i acikličan. S obzirom na to da svi čvorovi takvoga grafa moraju biti povezani, i to samo jednom, proizlazi da je broj veza u ovisnosnome stablu vezan uz broj njegovih čvorova, i to tako da je $|E| = |V| - 1$.

Za neku rečenicu prirodnoga jezika postoji konačan broj ovisnosnih stabala – dobro oblikovanih acikličnih ovisnosnih grafova – koja je moguće izraditi povezujući njezine riječi u ovisnosne parove i označavajući te veze ovisnosnim relacijama (usp. Kübler i dr. 2009:14). Može se reći da je zadatak svakoga ovisnosnog parsera odabrati jedno ovisnosno stablo iz skupa svih mogućih ovisnosnih stabala za svaku rečenicu ulaznoga teksta koje predstavlja ispravnu sintaktičku analizu te rečenice.

2.1.3.3 Svojstva ovisnosnoga stabla

Navodi se najčešće šest svojstava ovisnosnih stabala (usp. Kübler i dr. 2009) koja su korisna za ograničavanje opisa problema ovisnosnoga parsanja. Neka od tih svojstava već su ranije opisana, no ovdje se sustavno navode.

1. Svojstvo postojanja korijenskoga čvora (en. *root property*) zahtijeva da ne postoji nijedan čvor koji predstavlja glavu korijenskomu čvoru, odnosno $\nexists i \in V, i \rightarrow 0$. Zahtjev za korijenskim čvorom koji ne predstavlja neku od riječi iz rečenice lingvistički je i računalno utemeljen (usp. Kübler i dr. 2009:14) budući da on istovremeno jamči zadržavanje algoritamski obradive stablaste strukture u parsanju te modeliranje rečenica u kojima ima više od jedne riječi koja otvara mjesto drugim elementima rečeničnoga ustrojstva (primjerice, kod složenih rečenica, odnosno

rečenica s više od jednoga predikata ili kod rečenica bez predikata, u kojima je netrivijalno odrediti redoslijed uvođenja).

2. Svojstvo prostiranja preko svih dostupnih čvorova (en. *spanning property*), poput svojstva postojanja korijenskoga čvora, proizlazi iz definicije ovisnosnoga stabla. S jezikoslovnoga gledišta ovo svojstvo uvažava činjenicu da svaka riječ ima određenu sintaktičku funkciju u rečenici[54].
3. Svojstvo povezanosti čvorova (en. *connectedness*) ranije je opisano, ali se može formulirati preciznije. Neka je u tu svrhu definirana relacija neusmjerene ovisnosti, $i \leftrightarrow j \Leftrightarrow i \rightarrow j \vee j \rightarrow i$, dakle, relacija koja govori o tome da postoji ovisnost između dviju riječi, neovisno o tome koja je riječ glava, a koja dependent. Neka također postoji i njezina refleksivno-tranzitivna inačica $i \leftrightarrow^* j \Leftrightarrow i \rightarrow^* j \vee j \rightarrow^* i$ koja kaže da je neka riječ posredno u relaciji ovisnosti s nekom drugom riječju preko nekoga broja posredničkih riječi, odnosno posredničkih ovisnosti. (Kao, primjerice, u izrazu "porast individualnog prometa" iz primjera 2-14 u kojemu je riječ "individualnog" posredno ovisna o riječi "porast", i to preko riječi "prometa".) Sada se svojstvo povezanosti lako definira kao $\forall i, j \in V, i \leftrightarrow^* j$. Dakle, (usp. Kübler i dr. 2009:14) nužno postoji putanja koja, ukoliko se zanemari usmjerenost stabla[55], spaja bilo koje dvije riječi u stablu.
4. Postojanje jedne glave po dependentu (en. *single head property*) je svojstvo koje onemogućava višestruke ovisnosti, odnosno ovisnosti nekoga dependenta o više različitih glava: $\forall i, j \in V, i \rightarrow j \Rightarrow \nexists k \neq i, k \rightarrow j$. Ovo je svojstvo osigurano time što je ranije dopušteno da svaki čvor ima samo jednu dolaznu vezu[56].
5. Svojstvo necikličnosti (en. *acyclicity*) okvirno je definirano ranije i kaže da do svakoga čvora mora postojati samo jedna putanja. Formalnije, koristeći relacije ovisnosti, ovo svojstvo glasi: $\forall i, j \in V, i \rightarrow j \Rightarrow \neg(j \longrightarrow^* i)$. Ovo svojstvo je smisleno za svaku lingvističku teoriju zasnovanu na relaciji ovisnosti budući da bi njegova negacija značila kako neka riječ može posredno ovisiti o samoj sebi.

[54] S obzirom na raniju nerestriktivnu definiciju riječi koja uključuje interpunkciju i druge posebne znakove, (Kübler i dr. 2009:14) raspravlja o (nepostojećoj) sintaktičkoj funkciji tih posebnih znakova u rečenici s gledišta svojstva prostiranja i nerestriktivnosti toga svojstva s obzirom na posebne znakove. Također, budući da povezanost svih riječi u rečenici s gledišta sintaktičke analize ne mora biti podržana u svim ovisnosnim teorijama, odnosno mogu u nekoj teoriji biti podržane fragmentacije rečenice u nespojene cjeline, svojstvo postojanja korijenskoga čvora opet ih povezuje u isti formalni okvir sa svojstvom povezanosti.

[55] Radi se o ranije spomenutoj slaboj povezanosti.

[56] (Kübler i dr. 2009:15) raspravlja o implikacijama ograničenja na postojanje jedne glave u frazi s obzirom na, primjerice, koordinaciju, odnosno nezavisno-složene rečenice kod kojih se može smatrati da pojedine fraze ovise o oba glagola, iako se taj problem obično rješava uporabom koordinacijskoga veznika kao nadređenoga čvora i time se zadržava i svojstvo postojanja jedne glave.

6. Svojstvo kardinalnosti skupa svih veza, odnosno ovisnosnih relacija (en. *arc size property*) proizlazi iz definicije ovisnosnoga stabla i već je navedeno. To svojstvo kaže da je broj svih veza, odnosno ovisnosnih relacija u ovisnosnome stablu jednak $|\mathrm{E}| = |\mathrm{V}| - 1$, što proizlazi iz zahtjeva za korijenskim čvorom i postojanjem jedne glave po dependentu.

Definirana svojstva odabrana su zbog uspostavljanja veze između teorije grafova – koja je dobro istražena i dokumentirana te su u njezinim okvirima dostupni mnogi učinkoviti algoritmi za pretraživanje i optimizaciju (usp. Cormen i dr. 2009) – i ovisnosnoga parsanja u vidu objašnjavanja ovisnosnoga stabla kao posebne vrste grafa. Utoliko se može reći da su navedena svojstva zapravo ograničenja matematičkoga (i računalnoga) modela koji se naziva grafom, koja su učinila da taj model predstavlja ovisnosno stablo, odnosno matematički (i računalni) model koji je prikladan za modeliranje ovisnosne strukture nad rečenicama, odnosno za modeliranje strukture ovisnosnoga parsnog stabla. Ovako definiran model ovisnosnoga stabla pogodan je za modeliranje unutar svake sintaktičke teorije zasnovane na relaciji ovisnosti koja pritom podržava ranije navedena lingvistički usmjerena svojstva, poput svojstva prostiranja i svojstva postojanja samo jedne glave po dependentu. S obzirom na izračunljivost, odnosno izračunsku složenost, ovaj je formalizam po opisnim mogućnostima složeniji od beskontekstne gramatike budući da se njime mogu modelirati diskontinuirane fraze (kao u primjeru 2-13), pa je utoliko očekivano kako bi asimptotska složenost parsanja formalnom gramatikom ovoga tipa bila veća od složenosti parsanja beskontekstnom gramatikom. Međutim, budući da je problem ovisnosnoga parsanja ovdje postavljen kao problem parsanja (iz) teksta, a ne kao problem parsanja gramatikom, pravila određena za parsanje gramatikom na njega se više ne odnose. Ipak, s obzirom na određena sintaktička svojstva pojedinih jezika, a s ciljem dodatnoga pojednostavljenja problema ovisnosnoga parsanja tekstova tih pojedinih jezika, moguće je uvesti u model ovisnosnoga stabla dodatna ograničavajuća svojstva. Jedno od tih svojstava jest projektivnost.

2.1.3.4 Projektivna i neprojektivna ovisnosna stabla

Ovisnosno stablo $\mathrm{G} = (\mathrm{V}, \mathrm{E}, \mathrm{L})$ je *projektivno* (en. *projective dependency tree*) ako i samo ako vrijedi da su sve veze iz E projektivne. Veza $\mathrm{e} \in \mathrm{E}, \mathrm{e} = (\mathrm{i}, \mathrm{j}), \mathrm{L}(\mathrm{e}) = \mathrm{r}$ ovisnosnoga stabla G je projektivna ako i samo ako je $\forall \mathrm{k}, \big((\mathrm{i} < j) \wedge (\mathrm{i} < k < j)\big) \vee \big((\mathrm{j} < i) \wedge (\mathrm{j} < k < i)\big) \Rightarrow (\mathrm{i} \rightarrow^{*} \mathrm{k})$. Dakle, jedna veza $\mathrm{i} \rightarrow \mathrm{j}$ ovisnosnoga stabla je projektivna ako postoji usmjerena putanja od glave $\mathrm{w_i}$ do svih riječi $\mathrm{w_k}$ između $\mathrm{w_i}$ i $\mathrm{w_j}$. Jednostavnije rečeno,

projektivnost ovisnosnoga stabla zahtijeva da se ovisnost između dvaju razdvojenih elemenata neke fraze propagira putanjom preko svih riječi koje ih razdvajaju, odnosno zahtijeva da fraza bude kontinuirani niz riječi u rečenici.

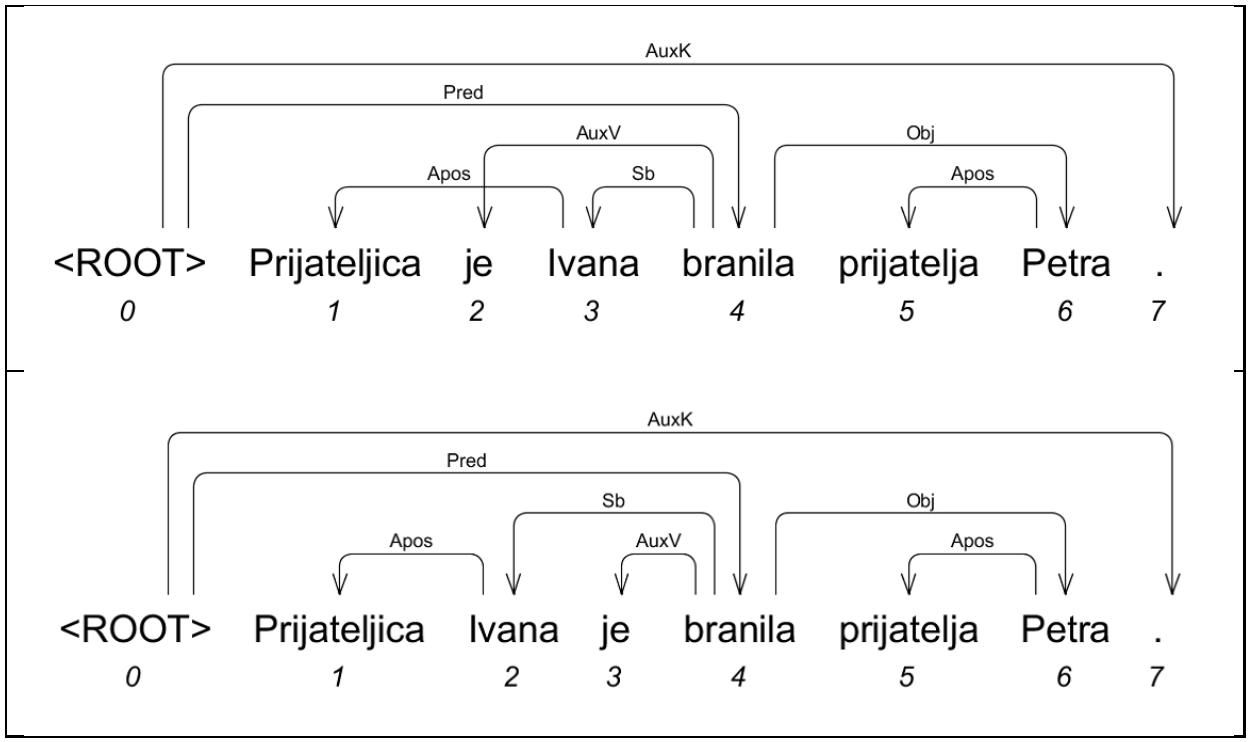
Obrnuto, ovisnosno stablo je *neprojektivno* (en. *non-projective dependency tree*) ako je ovisnosno stablo i ako nije projektivno, odnosno ako u njemu postoji barem jedna veza za koju ne vrijedi svojstvo projektivnosti.

Projektivnost ovisnosnoga stabla znači razdvojenost rečenice na kontinuirane fraze, odnosno implicira nepostojanje dislociranih riječi koje pripadaju pojedinim frazama. Primjer 2-13 rečenice hrvatskoga jezika s dislociranim elementom fraze jasno ilustrira kako projektivnost može biti ograničavajuća po sintaktički formalizam nekoga prirodnog jezika. Međutim, budući da je neprojektivnost u sintaktičkoj strukturi nekih prirodnih jezika, poput engleskoga, vrlo neuobičajena (usp. Kübler i dr. 2009:16), a projektivnost ovisnosnoga stabla kao ograničavajuće svojstvo implicira dodatna svojstva koja ga čine lakše algoritamski obradivim, često ga se razmatra i eksplicitno uvodi pri parsanju tih prirodnih jezika kojima je neprojektivnost nesvojstvena. S druge strane, nekim je jezicima[57] neprojektivnost svojstvena, najčešće jezicima sa složenijom morfologijom i posljedičnim manje ograničenim redoslijedom riječi u rečenici. Stoga se projektivnost kao zahtjev pri treniranju i korištenju ovisnosnih parsera, pa implicitno i kao zahtjev pri izgradnji banaka stabala, ovdje ne uvodi, već se samo iznosi činjenica o njegovu postojanju koja može biti korisna i korištena kao jedno od mogućih pojednostavljenja problema ovisnosnoga parsanja s obzirom na neke moguće daljnje primjene toga postupka koje ne zahtijevaju potpunu njegovu točnost.

Primjer 2-15 je ilustracija[58] neprojektivnosti rečenice iz primjera 2-13 te prikaz njezine projektivne inačice s obirom na razdvojenost subjektnoga skupa "prijateljica Ivana". Slijedni prikaz odabran je u primjeru 2-15 upravo zato što je, za razliku od okomitoga prikaza ovisnosnoga stabla od korijenskoga čvora prema listovima danoga primjerom 2-13 za istu rečenicu, vizualno pogodan za razotkrivanje neprojektivnosti ovisnosnih stabala. Naime, ukoliko je ovisnosno stablo neprojektivno, u slijednome se prikazu ovisnosne strukture nužno presijecaju ovisnosne relacije. U primjeru 2-15 presijeca se relacija ("branila", "je", AuxV) s

[57] (Kübler i dr. 2009:16) navodi češki, nizozemski i turski jezik kao primjere jezika s čestom neprojektivnošću, a kasnije se u ovome radu raspravlja i o neprojektivnosti u hrvatskome jeziku.
[58] Ova i druge ilustracije ovisnosnih stabala u kojima su riječi poredane onako kako se pojavljuju u rečenici, dakle, ilustracije s linearnim prikazom rečenica, izrađene su alatom MaltEval (Nilsson i Nivre 2008), v. URL alata http://w3.msi.vxu.se/users/jni/malteval/.

relacijom ("Ivana", "Prijateljica", Apos) budući da su subjektni i predikatni skup u slijedu riječi "Prijateljica je Ivana branila" pomiješani s obzirom na redoslijed riječi. Dodatno je dana inačica iste rečenice u kojoj su ovi skupovi razdvojeni, pa nema projektivnosti, što se na ilustraciji očituje u izostanku presijecanja među ovisnosnim relacijama.



Primjer 2-15 Neprojektivnost ovisnosnoga stabla

Grafički prikaz neprojektivnosti zapravo je ilustracija jednoga svojstva projektivnih ovisnosnih stabala koje se naziva svojstvom *planarnosti* (en. *planar property*) koje kaže da je za projektivno ovisnosno stablo moguće grafički urediti sve veze u stablu u prostoru iznad rečenice bez križanja grafičkoga prikaza veza. Za neprojektivnost se utoliko kaže suprotno: da je nemoguće grafičko uređenje stabla bez presijecanja ovisnosnih relacija.

Projektivna ovisnosna stabla također posjeduju svojstvo ugniježđenosti (en. *nested property*) koje kaže da za $\forall i, j \in V, \{w_j : w_i \rightarrow^* w_j\}$ vrijedi da je tako definiran skup riječi neprekinuti niz riječi iz rečenice. Ugniježđenost je prikazana i intuitivno čitljiva iz stabla projektivne rečenice iz primjera 2-15.

Dokazano je da projektivna ovisnosna stabla po složenosti i opisnoj snazi odgovaraju beskontekstnim gramatikama (usp. Kübler i dr. 2009:18) pa se mnogi sustavi za ovisnosno parsanje ograničavaju upravo na projektivne strukture, pogotovo ako se odnose na parsanje tekstova engleskoga i sličnih jezika. Takva odluka je u tim slučajevima utemeljena budući da

je beskontekstna gramatika, zajedno s njezinim ekvivalentima, dobro istražena i algoritamski učinkovito obradiva, pa se takva algoritamska podrška može očekivati i za projektivne ovisnosne strukture. Međutim, s obzirom na usmjerenost ovoga istraživanja tekstovima hrvatskoga jezika – znajući kako hrvatski jezik ima složenu morfologiju i relativno slobodan redoslijed riječi u rečenici – ovdje se nipošto ne uvodi eksplicitno ograničenje formalizma na projektivne ovisnosne strukture, no i one se svakako razmatraju.

2.1.3.5 Definicija ovisnosnog parsanja

Ranije je navedeno kako je ovisnosno parsanje problem odabira jednoga ovisnosnog stabla iz skupa svih mogućih ovisnosnih stabala koja je moguće izgraditi nad nekom ulaznom rečenicom uz dani skup ovisnosnih relacija, odnosno sintaktičkih funkcija. Ovdje se, prema (Nivre 2006) i (Kübler i dr. 2009), matematički formalizira upravo ta definicija, i to s gledišta modela sustava za parsanje temeljenoga na podatcima. Stoga formalna definicija koja slijedi uzima u obzir problem učenja, odnosno treniranja modela iz banke stabala, i problem samoga parsanja, odnosno testiranja modela na novim ulaznim podatcima.

Prema (Kübler i dr. 2009:18), *model ovisnosnoga parsera* (i parsanja, en. *dependency parsing model*) definiran je kao $\mathrm{M} = (\Gamma, \lambda, \mathrm{h})$, gdje je Γ skup ograničenja koja dodatno preciziraju prostor za definiranje skupa svih mogućih parsnih stabala za danu ulaznu rečenicu, λ je skup parametara koji predstavljaju jezični (sintaktički) model ovisnosnoga parsera, a h je parsni algoritam[59]. Načelno, skup ograničenja Γ je skup implicitno zadanih sintaktičkih pravila kojima se skup sintaktičkih funkcija R povezuje s riječima parsane rečenice i u praksi se svodi na ranije predstavljeni skup ograničenja koji od grafa gradi ovisnosno stablo, no načelno može uključivati i složenije ograničavajuće mehanizme poput projektivnosti.

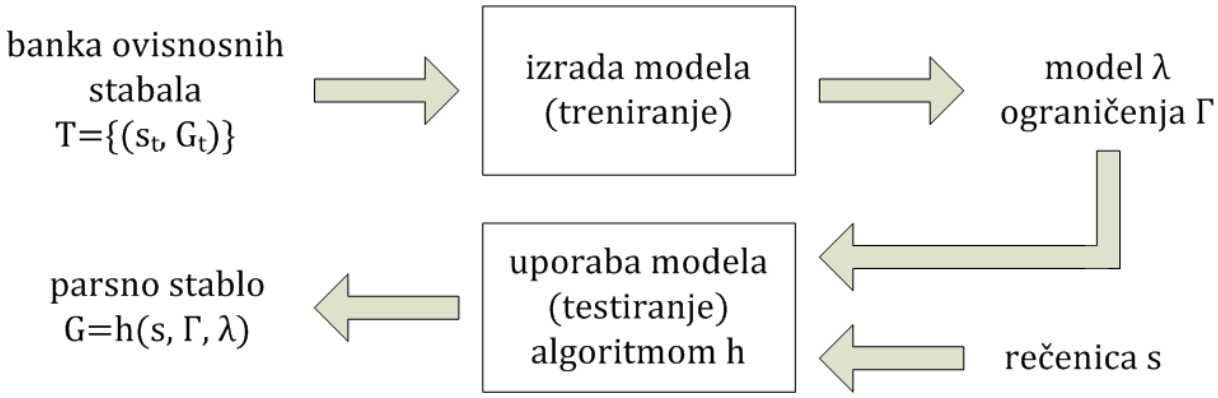
U radu parsera temeljenoga na podatcima, kao u početnome modelu takvoga parsera sa slika 2-4 i 2-5, razlikuje se faza treniranja, odnosno učenja, i faza testiranja, odnosno korištenja naučenoga jezičnog modela.

Pri treniranju se nastoji iz ulaznih podataka – prethodno sintaktički obilježene i ručno provjerene banke ovisnosnih stabala T – izgraditi skup parametara, odnosno jezični model λ. Pritom se banka stabala T definira kao skup uređenih parova rečenica i pripadajućih

[59] Model definiran u (Kübler i dr. 2009:18) pritom ne definira algoritam kojim se stvara jezični model λ, nego ga podrazumijeva, odnosno veže implicitno uz metode svojstvene određenim paradigmama strojnoga učenja. Mogao bi se stoga razlikovati algoritam $\mathrm{h_L}$ od algoritma $\mathrm{h_P}$ pa tako modelirati parser kao uređenu četvorku $\mathrm{M} = (\Gamma, \lambda, \mathrm{h_L}, \mathrm{h_P})$, no ovdje se ta distinkcija zbog jednostavnosti ipak ne uvodi.

ovisnosnih stabala: $T = \{(s_t, G_t)\}_{t=1}^{|T|}$. Preciznija definicija postupka treniranja ovisi o izboru matematičkoga formalizma jezičnoga modela λ i dostupnim algoritmima za određivanje numeričkih parametara unutar toga formalizma, pa se ovdje ne definira (i ne može definirati) pobliže, već se određuje u opisu pojedinih konkretnih pristupa ovisnosnomu parsanju. Taj postupak može, primjerice, uključivati (usp. Kübler i dr. 2009:19) "procjenu vjerojatnosti dodjele neke ovisnosne relacije nekomu paru riječi u rečenici".

Postupak parsanja, odnosno korištenja ili testiranja jezičnoga modela provodi se u vidu primjene nekoga parsnog algoritma nad modelom izgrađenim u fazi treniranja. Uz zadana ograničenja Γ i jezični model λ, algoritam za ulaznu rečenicu s mora vratiti – prema svim ranije zadanim zahtjevima i ograničenjima – jedno parsno stablo koje predstavlja valjani ovisnosno-sintaktički opis te rečenice: $G = h(s, \Gamma, \lambda)$. Algoritam h stoga se može smatrati funkcijom koja, poštujući zadana ograničenja Γ, pretražuje prostor svih mogućih ovisnosnih stabala za rečenicu s i vrjednuje ih prema jezičnome modelu λ, pa prema njemu određuje i najboljega kandidata. S obzirom na te definicije, smislena je preinaka početnih modela sa slika 2-4 i 2-5 u model ovisnosnoga parsera prikazan slikom 2-6[60].



Slika 2-6 Model ovisnosnoga parsera temeljenoga na podatcima

U idućem poglavlju, pobliže se definiraju neki pristupi ovisnosnomu parsanju, odnosno izradi ovisnosnih parsera temeljenih na podatcima. S gledišta formalne definicije postupka treniranja i testiranja jezičnoga modela i s gledišta kriterija odabira i vrjednovanja, nastoji se precizno definirati dobar model parsera tekstova prirodnoga jezika temeljenoga na podatcima $M = (\Gamma, \lambda, h)$ preko preciznoga određivanja svojstava njegovih sastavnica.

[60] Vrijedi primijetiti kako slika prikazuje model λ i ograničenja Γ kao ishode postupka treniranja. Ipak, ograničenja Γ ne moraju nužno biti rezultatom postupka treniranja, već mogu biti i izvanjski zadana, primjerice, odabirom formalizma i pripadajućega skupa sintaktičkih funkcija za dodjelu ovisnosnim relacijama.

2.2 Pristupi ovisnosnom parsanju

Mogu se načelno zamisliti dva osnovna razdvojena pristupa opisivanju svojstava različitih postojećih metoda ovisnosnoga parsanja u okvirima koji su ranije postavljeni preduvjetima, kriterijima vrjednovanja i općim modelom ovisnosnoga parsera.

U prvome pristupu, proučavanjem literature i razmatranjem rezultata ranijih istraživanja, prikazala bi se smislena shema opće i specifične razredbe postojećih pristupa ovisnosnomu parsanju, pa bi se – kao ranije u ovome tekstu, kod parsanja beskontekstnom gramatikom, odnosno parsera CYK i Earleyjeva parsera – prikazali neki karakteristični predstavnici pojedinih skupina i njihova svojstva. Tim pristupom, moglo bi se, primjerice, prema (Kübler i dr. 2009:7), navesti da postoje dvije osnovne skupine ovisnosnih parsera:

1. ovisnosni parseri koji parsaju ovisnosnom gramatikom i
2. ovisnosni parseri temeljeni na podatcima.

Ta podjela je već razmotrena ranije u ovome tekstu te su, s obzirom na postavljene opće preduvjete i kriterije vrjednovanja parsera, za daljnje razmatranje odabrani ovisnosni parseri temeljeni na podatcima, odnosno oni s ciklusom treniranja i testiranja sintaktičkoga modela. Opet prema (Kübler i dr. 2009:7), u toj se skupini razlikuju, ovisno o izboru jezičnoga modela i pripadajućega parsnoga algoritma, sljedeće metode:

1. ovisnosni parseri temeljeni na teoriji grafova i
2. ovisnosni parseri temeljeni na teoriji automata, odnosno na prijelazima.

Prije daljnje rasprave potrebno je istaknuti kako na podatcima, odnosno na procesu treniranja i testiranja mogu biti temeljeni i pristupi parsanju preko pravila, odnosno s pomoću nekoga modela formalne ovisnosne gramatike koji se izgrađuje iz podataka[61]. Međutim, iako njihovo postojanje jest podržano teorijski, dostupna literatura potvrđuje (usp. Nivre 2006:20, Kübler i dr. 2009) kako ti pristupi nisu brojni i ne udovoljavaju ovdje zadanim formalnim preduvjetima i kriterijima vrjednovanja, pa se dalje ne razmatraju.

Idući korak u ovakvoj raspravi bio bi odabir i razmatranje pojedinih konkretnih pristupa parsanju, odnosno pojedinih ovisnosnih parsera koji pripadaju kategoriji ovisnosnih parsera

[61] Ovdje je nužno primijetiti razliku između parsanja formalnom gramatikom (en. *grammar parsing*) i parsanja teksta gramatičkim modelom (en. *grammar-driven text parsing*) izgrađenim iz banke stabala.

temeljenih na teoriji grafova te ovisnosnih parsera temeljenih na prijelazima. Kod ranijega prikaza parsera beskontekstnom gramatikom (CYK i Earley) odabir prikazanih parsera bio je vođen njihovom ilustrativnošću i različitošću s obzirom na razredbu parsera iz tablice 2-1, odnosno željom da se pokažu dva razdvojena pristupa problemu parsanja, ali također i njihovom karakterističnom polinomskom (kubnom) vremenskom složenošću. Kod prikaza ovisnosnih parsera tekstova prirodnoga jezika odabirom opet može upravljati ilustrativnost s obzirom na upravo postavljenu binarnu razredbu (teorija grafova, teorija formalnih automata) ovisnosnih parsera temeljenih na podatcima i rezultati nekoga njihova ranije prikazanog vrjednovanja prema zadnome općem okviru za vrjednovanje ovisnosnih parsera prema točnosti i učinkovitosti parsanja.

Upravo odabir karakterističnih modela ovisnosnih parsera za detaljnije razmatranje u ovome tekstu, prema njihovoj točnosti i učinkovitosti, vodi prema drugome mogućem pristupu opisivanju pristupa ovisnosnomu parsanju. Drugi pristup vezan je uz činjenicu da (usp. Kübler i dr. 2009:1) ovisnosno parsanje, iako je "mnogo godina igralo razmjerno marginalnu ulogu u obradbi prirodnoga jezika", u zadnje vrijeme – "zbog iskoristivosti u mnogim primjenama jezičnih tehnologija, poput strojnoga prevođenja i crpljenja obavijesti" i zbog "prikladnosti u opisu jezika sa slobodnim ili fleksibilnim redoslijedom riječi", ali najviše zbog "razvoja točnih i učinkovitih ovisnosnih parsera temeljenih na strojnome učenju i lako prilagodljivih, odnosno neovisnih o parsanom jeziku, zadanom isključivo implicitno, preko banke ovisnosnih stabala" – privlači "značajnu pozornost u istraživačkoj zajednici". Iz te "značajne pozornosti" istraživača, vjerojatno procijenjene u (Kübler i dr. 2009) brojem istraživanja vezanih uz ovisnosno parsanje u zadnjih desetak godina (usp. Kübler i dr. 2009:97), proizašla su i natjecanja (često nazivana dijeljeni zadatci, en. *shared tasks*) u ovisnosnome parsanju na značajnim skupovima s područja jezičnih tehnologija. Dva najpoznatija održana su u sklopu skupa CoNLL (en. *Conference on Computational Natural Language Learning*)[62], i to 2006. i 2007. godine (Buchholz i Marsi 2006, Nivre i dr. 2007)[63], upravo na temu višejezičnoga ovisnosnog parsanja (en. *multilingual dependency parsing*), dopunjenu u zadatku na natjecanju 2007. godine dodatnim zadatkom prilagodbe na domenu teksta (en. *domain adaptation*). U idućim godinama (Surdeanu i dr. 2008, Hajič i dr. 2009) zadatci su dodatno prošireni na združeno višejezično parsanje (odnosno pronalaženje)

[62] Popis skupova i na njima objavljenih istraživanja te tekstovi samih istraživanja dostupni su preko sustava ACL Anthology, URL http://aclweb.org/anthology-new/signll.html (2012-03-04).
[63] Vidjeti također http://nextens.uvt.nl/~conll/ i http://nextens.uvt.nl/depparse-wiki/SharedTaskWebsite, odnosno URL-ove samih natjecanja (2012-03-04).

sintaktičkih i semantičkih ovisnosti u tekstu. Budući da je u tim natjecanjima – pritom se posebno misli na natjecanja usmjerena isključivo ovisnosnomu parsanju, dakle 2006. i 2007. godine – sudjelovao veliki broj istraživača te je prikazan veliki broj ovisnosnih parsera temeljenih na različitim teorijskim postavkama ili paradigmama, oni se mogu uzeti kao dobra ishodišna točka za promatranje i odabir metoda koje je smisleno detaljnije prikazati u ovome istraživanju.

Dakle, pristup koji se u ovome istraživanju bira za prikaz metoda ovisnosnoga parsanja temeljenih na podatcima je sljedeći. Prikazuju se osnovne postavke i rezultati dvaju dostupnih natjecanja na temu višejezičnoga ovisnosnog parsanja te se razmatranjem tih rezultata (i njihovim dovođenjem u odnos s ovdje postavljenim općim kriterijima vrjednovanja) odabiru parseri s najboljom izvedbom – prvenstveno u smislu točnosti, a onda i učinkovitosti parsanja, ali i u smislu performansi s obzirom na karakteristične skupine jezika koji su sudjelovali u natjecanju – pa se detaljno opisuju teorijski modeli na kojima su oni zasnovani. Ovime se osigurava odabir u skladu s već provedenim vrjednovanjem prema objektivnim kriterijima, odnosno promatranje samo dokazano dobrih – provjereno točnih i učinkovitih, pa stoga i smislenih – pristupa ovisnosnomu parsanju s gledišta ovoga istraživanja.

2.2.1 Natjecanja u ovisnosnome parsanju CoNLL 2006 i 2007

Natjecanja u višejezičnom ovisnosnom parsanju u sklopu znanstvenoga skupa CoNLL 2006. i 2007. godine (Buchholz i Marsi 2006, Nivre i dr. 2007) zamišljena su na sljedeći način. U pripremi natjecanja dostavljaju se ovisnosne banke stabala ili reprezentativni uzorci banaka stabala za što veći broj jezika. Potom se ti podatci dijele na dio za treniranje modela i dio za testiranje, odnosno posljedično vrjednovanje. Na natjecanje se potom prijavljuju sustavi za ovisnosno parsanje koji se treniraju na skupovima podataka za treniranje i testiraju na izdvojenim skupovima podataka za testiranje. Za svaki od sustava i svaki od jezika rezultati testiranja uspoređuju se s izdvojenim točnim parsanjima skupova podataka za testiranje prema nekoj unaprijed određenoj mjeri za vrjednovanje. Rezultati natjecanja, kao i prikazi pojedinih parsera koji su sudjelovali u natjecanju, potom se objavljuju i javno prikazuju u sklopu pripadajućega znanstvenog skupa.

Da bi se ovdje – s ciljem odabira za daljnje razmatranje dokazano najboljih metoda višejezičnoga ovisnosnog parsanja temeljenoga na podatcima – prikazali i tumačili rezultati ovih natjecanja, potrebno je pojasniti postupak pripreme podataka za natjecanje, postavke ili

pravila pokusa i metodu vrjednovanja rezultata. Sažeti prikaz rezultata ovih dvaju natjecanja – detaljno opisanih u (Buchholz i Marsi 2006) i (Nivre i dr. 2007a), odnosno izdanjima sa znanstvenoga skupa CoNLL – iznesen je također u (Kübler i dr. 2009:82).

2.2.1.1 Zapis podataka

Kako bi se velikim brojem različitih ovisnosnih parsera temeljenih na podatcima mogli graditi i koristiti jezični modeli iz velikoga broja različitih banaka ovisnosnih stabala, za potrebe natjecanja CoNLL 2006 i 2007 osmišljen je poseban oblik zapisa kojemu su se morale prilagoditi sve banke ovisnosnih stabala korištene u natjecanju. Taj zapis je opisan na sljedeći način. Ovisnosna banka stabala sastoji se od jedne ili više datoteka. Svaka datoteka sadrži tekstove nekoga prirodnog jezika, razdvojene na rečenice. Svaka je rečenica zapisana tako da se svaka njezina riječ nalazi u novoj liniji datoteke. Jedna riječ opisana je s pomoću osam pridruženih obilježja, odnosno metapodataka:

1. brojačem ili identifikatorom riječi, koji kreće od 1 do ukupnoga broja riječi u rečenici i predstavlja jedinstveni identifikator dane riječi u toj rečenici (ID),
2. pojavnim oblikom riječi u rečenici (FORM),
3. polaznim oblikom navedenoga pojavnog oblika, odnosno lemom (LEMMA),
4. općenitom morfosintaktičkom oznakom (en. *coarse part-of-speech tag*), često samo oznakom vrste riječi (CPOSTAG),
5. detaljnom morfosintaktičkom oznakom, koja može biti i jednaka općenitoj oznaci, ovisno o dostupnome skupu, odnosno standardu morfosintaktičkih oznaka za dani jezik (POSTAG),
6. skupom sintaktičkih i/li semantičkih značajki pripadajuće riječi, ukoliko takav skup ili standard postoji za dani jezik (FEATS),
7. unutarrečeničnim identifikatorom glave zadane riječi u ovisnosnoj relaciji, ukoliko riječ ima glavu, ili vrijednošću 0, ukoliko riječ nema glavu, pa se veže uz korijenski čvor (HEAD) i
8. sintaktičkom funkcijom, odnosno ovisnosnom relacijom zadane riječi prema glavi zadanoj u prethodnoj točki, iz skupa svih ovisnosnih relacija, odnosno sintaktičkih funkcija u banci ovisnosnih stabala zadanoga jezika (DEPREL).

Uz ovih osam obaveznih značajki kojima u zadanome zapisu mora biti opisana svaka riječ u rečenici, zapisom su predviđene još dvije dodatne značajke koje se odnose isključivo

na projektivnost ovisnosnoga stabla, s obzirom na činjenicu da se rečenice kojima prema zadanome sintaktičkom formalizmu pripadaju neprojektivna ovisnosna stabla mogu parsati i na projektivan način, pretvorbom neprojektivnih u projektivna ovisnosna stabla (usp. Kübler i dr. 2009:37). U ovome zapisu, dodatne značajke vezane uz projektivnost su:

9. projektivna glava zadane riječi (PHEAD) i
10. sintaktička funkcija prethodnom značajkom uvedene ovisnosne relacije zadane riječi prema projektivnoj glavi (PDEPREL).

1	Prijateljica	prijateljica	Z	Z	_	3	Apos	_	_
2	je	biti	V	V	_	4	AuxV	_	_
3	Ivana	Ivana	N	N	_	4	Sb	_	_
4	branila	braniti	V	V	_	0	Pred	_	_
5	prijatelja	prijatelj	N	N	_	6	Apos	_	_
6	Petra	Petar	N	N	_	4	Obj	_	_
7	.	.	Z	Z	_	0	AuxK	_	_
1	Prijateljica	prijateljica	Z	Z	_	2	Apos	_	_
2	Ivana	Ivana	N	N	_	4	Sb	_	_
3	je	biti	V	V	_	4	AuxV	_	_
4	branila	braniti	V	V	_	0	Pred	_	_
5	prijatelja	prijatelj	N	N	_	6	Apos	_	_
6	Petra	Petar	N	N	_	4	Obj	_	_
7	.	.	Z	Z	_	0	AuxK	_	_

Primjer 2-16 Zapis rečenica iz primjera 2-15 po pravilima s natjecanja CoNLL 2006 i 2007

Rečenice su u ovome zapisu razdvojene jednom praznom linijom datoteke. Zapis podržava dodatne metapodatke, poput identifikatora rečenica, paragrafa, dokumenata i bilo kojih ostalih metapodataka naslijeđenih iz izvornoga zapisa ovisnosne banke stabala, u obliku XML-oznaka, koje se također moraju zapisati u zasebnim redcima datoteke. Tako definirani zapis banke ovisnosnih stabala naziva se zapisom CoNLL ili formatom CoNLL. Dvije rečenice hrvatskoga jezika zapisane formatom CoNLL prikazane su primjerom 2-16. U primjeru su, u skladu s uputama zapisivanja banaka stabala u formatu CoNLL, izostavljene ili

nepostojeće značajke (poput popisa dodatnih morfosintaktičkih značajki, projektivne glave i projektivne ovisnosne relacije) označene donjom crtom (en. *underscore*). Također se vidi kako su predikati vezani ovisnosnim relacijama za nulte, odnosno korijenske čvorove, kako je prikazano i slijednim ovisnosnim stablom u primjeru 2-15. Osim predikata, na korijenske su čvorove direktno povezani i znakovi za završetak rečenica.

Svaki ovisnosni parser morao je za sudjelovanje u dijeljenim zadatcima CoNLL 2006 i 2007 znati rukovati bankama stabala u formatu CoNLL, kao ulaznim i kao izlaznim formatom podataka. Dakle, modul parsera za treniranje jezičnoga modela morao je znati čitati banke stabala u formatu CoNLL, a modul za korištenje jezičnoga modela morao je zapisivati rezultate parsanja ulaznoga teksta u istome formatu. Sam ulazni tekst za parsanje također je zapisan u formatu CoNLL, ali sadrži samo prvih šest stupaca, odnosno, iz njega su uklonjeni i izdvojeni za kasnije vrjednovanje očekivani rezultati parsanja. Prema (Kübler i dr. 2009:83), "prvih šest stupaca predstavlja jezične značajke koje se stavljaju na raspolaganje sustavima za parsanje kako bi iz njih ili nekoga njihova podskupa izradili jezični model", a sedmi i osmi stupac (glava i sintaktička funkcija) predstavljaju "podatke koje treba naučiti", odnosno povezati jezične značajke iz prvih šest stupaca – koji predstavljaju pojavnost teksta i njegova morfosintaktička svojstva – s ovisnosno-sintaktičkim značajkama.

2.2.1.2 Postavke pokusa

U natjecanju CoNLL 2006 sudjelovalo je, odnosno, pokusne banke stabala dostavljene su za ukupno (usp. Kübler i dr. 2009:83) "trinaest jezika iz sedam obitelji jezika: arapski, bugarski (kao dodatni, odnosno izborni jezik koji nije bilo nužno uključiti u postupak parsanja), češki, danski, japanski, kineski, nizozemski, njemački, portugalski, slovenski, španjolski, švedski, turski jezik". Svaka banka stabala podijeljena je na uzorak za treniranje modela i uzorak za testiranje modela, odnosno vrjednovanje rezultata. Uzorci za testiranje sadržavali su otprilike 5.000 riječi, dok je veličina uzorka za treniranje modela ovisila o veličini dostavljene banke ovisnosnih stabala, prema načelu izdvajanja testnoga uzorka od oko 5.000 riječi i korištenja preostaloga dijela banke stabala za postupak treniranja modela. Veličina uzorka za treniranje stoga je varirala od otprilike 29.000 riječi u otprilike 1.500 rečenica (Slovenska banka ovisnosnih stabala[64] (Ledinek i Žele 2005, Džeroski i dr. 2006)) pa do 1.290.000 riječi u otprilike 72.000 rečenica iz Praške banke ovisnosnih stabala (Buchholz i Marsi 2006:155). S obzirom na značajne razlike u veličinama uzoraka za treniranje modela na

[64] Vidjeti i internetsku stranicu SDT-a, URL http://nl.ijs.si/sdt/ (2012-03-05).

ovome natjecanju i s obzirom na njihov očekivani utjecaj na kvalitetu proizašlih jezičnih modela parsera, veličina uzorka za treniranje ograničena je za natjecanje CoNLL 2007 na najviše 500.000 riječi, dok je zadržana mjera od 5.000 riječi za testne uzorke. Na natjecanju CoNLL 2007 sudjelovalo je ukupno deset banaka ovisnosnih stabala iz devet skupina jezika: arapski, baskijski, češki, engleski, grčki, katalonski, kineski, mađarski, talijanski i turski. Razlike među uzorcima za treniranje svejedno su ostale velike – od 2.700 rečenica grčkoga jezika pa do 25.400 rečenica češkoga jezika (usp. Kübler i dr. 2009:84) – ali je uzorak za treniranje modela na Praškoj banci stabala značajno umanjen.

Na natjecanje CoNLL prijavljeno je 2006. godine 19, a 2007. godine sudjelovala su u njemu 23 različita ovisnosna parsera. Svaki od parsera morao je izgraditi jezični model na uzorku za treniranje i primijeniti ga na uzorku za testiranje. Rezultati parsanja na uzorku za testiranje potom su uspoređeni s izdvojenim točnim parsanjem uzorka za testiranje primjenom više različitih mjera vrjednovanja točnosti parsanja. Za svaku banku stabala testiranje je ponovljeno deset puta (en. *tenfold cross-validation*), i to podjelom svake banke stabala na deset nepreklapajućih dijelova, korištenjem devet od tih deset dijelova za postupak treniranja modela te 5.000 riječi kao desetog dijela za postupak testiranja modela.

2.2.1.3 Postupak vrjednovanja točnosti

Vrjednovanje točnosti ovisnosnih parsera provedeno je na natjecanjima CoNLL 2006 i 2007 usporedbom parsnih stabala koja su parseri predložili za dane testne rečenice s valjanim parsnim stablima koja su tim testnim rečenicama dodijeljena unutar banaka stabala. Pritom su se koristile dvije mjere za vrjednovanje: točnost povezivanja riječi u ovisnosne relacije uz dodjelu sintaktičkih funkcija (en. *labeled attachment score*, LAS) i točnost povezivanja riječi bez obzira na točnost dodjele sintaktičkih funkcija (en. *unlabeled attachment score*, UAS). Slijedi definicija ovih mjera vrjednovanja točnosti parsanja prema (Nivre 2006:127).

Neka je dan testni uzorak teksta $T = (s_1, \dots, s_n)$ iz banke ovisnosnih stabala i njegovo referentno parsanje, odnosno skup valjanih ovisnosnih stabala iz banke stabala za rečenice iz testnoga uzorka $A_R = (t_1^R, \dots, t_n^R)$. Neka parser P kao rezultat parsanja teksta T proizvede skup predloženih parsnih stabala $A_P = (t_1^P, \dots, t_n^P)$ kao rezultat primjene jezičnoga modela stvorenoga na uzorku za treniranje. Neka h_R predstavlja funkciju koja povezuje dependent i glavu za svaki dependent u nekoj rečenici $s_i = (w_1, \dots, w_k)$ referentnoga parsanja A_R, a neka

u skladu s time h_P predstavlja funkciju koja povezuje dependent i glavu za svaki dependent u nekoj rečenici $s_i = (w_1, \ldots, w_k)$ parsanja A_P dobivenoga primjenom parsera P.

Mjera točnosti povezivanja dependenata s glavama u ovisnosne relacije neovisno o dodjeli sintaktičkih funkcija – koja se ovdje i dalje u tekstu naziva *točnost neoznačenoga povezivanja* te se za nju uglavnom koristi skraćenica UAS – definira se za neku rečenicu $s = (w_1, \ldots, w_k)$ na sljedeći način:

$$\mathrm{UAS}(s) = \frac{1}{k}\sum_{i=1}^{k} \delta\big(h_R(i), h_P(i)\big), \qquad \delta(i) = \begin{cases} 1, \text{ako je } h_R(i) = h_P(i) \\ 0, \text{u protivnome} \end{cases}$$

Vrijedi primijetiti kako se za potrebe definiranja mjere UAS ne koristi nulta riječ rečenice, odnosno korijenski čvor budući da funkcija za povezivanje dependenata s glavama nije definirana za korijenski čvor jer on nikad nije dependent. Prema pojašnjenju iz (Nilsson i Nivre 2008), riječ se funkcijom δ "vrjednuje kao 1, odnosno kao točno parsana, ukoliko je njoj dodijeljena glava jednaka glavi koja joj je dodijeljena u banci stabala, a 0 u protivnome".

Mjera točnosti povezivanja dependenata s glavama u ovisnosne relacije uz zahtjev za točnošću dodijeljene sintaktičke funkcije – koja se naziva *točnost označenoga povezivanja* te se za nju uglavnom koristi skraćenica LAS – definira se, slično kao kod definicije točnosti neoznačenoga povezivanja, na sljedeći način:

$$\mathrm{LAS}(s) = \frac{1}{k}\sum_{i=1}^{k} \delta\big(h_R(i), h_P(i)\big) \cdot \delta\big(d_R(i), d_P(i)\big)$$

Funkcije d_P i d_R pritom su definirane, po uzoru na dane funkcije h_P i h_R, tako da kao parametar uzimaju neku riječ rečenice, a vraćaju oznaku sintaktičke uloge koja je toj riječi dodijeljena u banci stabala i u postupku parsanja. Opet prema pojašnjenju iz (Nilsson i Nivre 2008), riječ se ovdje "vrjednuje kao 1, odnosno kao točno parsana, ukoliko je njoj dodijeljena glava jednaka glavi koja joj je dodijeljena u banci stabala i ukoliko je toj vezi dodijeljena ona sintaktička funkcija koja joj je dodijeljena u banci stabala, a 0 u protivnome".

Budući da su obje mjere definirane nad jednom rečenicom, koriste se nad svakom od rečenica iz testnoga uzorka, pa se za točnost parsanja nad čitavim testnim uzorkom može uzeti prosječna vrijednost po rečenici (en. *mean per sentence*, *macro-average*) ili prosjek po riječi (en. *mean per word*, *micro-average*). Navodi se (usp. Nivre 2006:127) kako se najčešće

koristi prosjek po riječi budući da "rezultat dobiven za vrlo kratke rečenice može narušiti vjerodostojnost ukupnoga rezultata".

Prema definiciji mjera UAS i LAS, može se izvesti i mjera kojom se vrjednuje samo vezivanje dependenata i sintaktičkih funkcija (usp. Nilsson i Nivre 2008). Ta mjera naziva se *mjerom točnosti označavanja* (en. *label attachment score*, LA) i definira se iz definicije mjere koja je sadrži, odnosno mjere LAS:

$$\mathrm{LA(s)} = \frac{1}{k}\sum_{i=1}^{k} \delta\big(d_R(i), d_P(i)\big)$$

Dakle, mjera točnosti označavanja (ili dodjele sintaktičkih funkcija dependentima) kaže da se riječ "vrjednuje kao točno parsana ukoliko je njezina ovisnosna sintaktička funkcija jednaka onoj koja joj je dodijeljena u banci stabala".

Uz ove tri mjere, (Nivre 2006:126) navodi i intuitivnu mjeru koju naziva *potpunim poklapanjem* (en. *exact match*) kojom se ne uvažava djelomična valjanost parsnoga stabla u vidu uspješnoga povezivanja određenoga broja riječi u ovisnosne relacije, već se zahtijeva identičnost referentnoga parsnog stabla i onoga dobivenog parserom:

$$\mathrm{EM} = \frac{1}{n}\sum_{i=1}^{n} \delta\big(t_i^P, t_i^R\big)$$

Mjera potpunoga poklapanja govori o udjelu potpuno točno parsanih rečenica iz testnoga uzorka i najčešće se koristi samo kao dodatak prethodnim mjerama budući da (usp. Nivre 2006:126) "jednako vrjednuje pogrješku nastalu na samo jednoj riječi, pogrješku u parsanju fraze te pogrješku u vidu potpunoga izostanka parsnoga stabla".

S obzirom na ranije definirana svojstva ovisnosnih stabala – preciznije, svojstvo kojim se zahtijeva postojanje jedne glave po dependentu – moguće je valjano vrjednovati ovisnosno parsanje samo jednom od ovih mjera točnosti (usp. Kübler i dr. 2009:79) budući da se može svaka od ovih i bilo koja moguća druga mjera odnositi isključivo na postupak povezivanja riječi u ovisnosne relacije, odnosno postupak dodjele sintaktičkih metapodataka pojedinim riječima u rečenicama. Analogno morfosintaktičkom označavanju, može se stoga reći da se u ovisnosnome parsanju svakoj riječi dodjeljuje sintaktička oznaka koja sadrži dva podatka: glavu riječi i ovisnosnu funkciju te riječi prema glavi. Stoga se točnost ovisnosnoga parsanja

neke rečenice može vrjednovati na razini riječi: ukoliko je riječi dodijeljena ispravna glava i/li sintaktička funkcija, ukupna točnost parsanja se uvećava, a u protivnom, neuvećavanjem se ukupna točnost implicitno umanjuje. S obzirom na postupke vrjednovanja parsanja u okviru modela sintakse koji su zasnovani na fraznoj strukturi, može se reći da je vrjednovanje ovisnosnoga parsanja jednostavniji postupak (usp. Kübler i dr. 2009:79).

Budući da su mjere LAS i UAS, za razliku od mjere EM, definirane na razini rečenice, treba ih – za potrebe vrjednovanja parsanja tekstova prirodnoga jezika – prilagoditi kako bi se s pomoću njih izrazila točnost parsera na zadatku parsanja nekoga teksta. Slijede definicije mjera LAS i UAS za parsanje čitavoga teksta (Nivre 2006:140).

Neka je, slično kao i ranije, zadan testni uzorak $T = (s_1, \dots, s_m)$ koji se sastoji od m rečenica, a pritom se svaka rečenica $s_i = (w_1, \dots, w_{k_i})$ sastoji od k_i riječi i neka je pritom ukupan broj riječi u testnome uzorku T jednak n, odnosno $n = \sum_{i=1}^{m} k_i$. Neka su referentna parsna stabla i parsna stabla dobivena parsanjem s pomoću parsera P za rečenice uzorka T definirana skupovima parsnih stabala $A_R = (t_1^R, \dots, t_m^R)$ i $A_P = (t_1^P, \dots, t_m^P)$, kao i ranije. Neka je svako referentno ovisnosno stablo t_i^R definirano kao ovisnosni graf $t_i^R = (V_{s_i}, E_i^R, L_i^R)$ i neka slično vrijedi i za parsna stabla dobivena parsanjem, $t_i^P = (V_{s_i}, E_i^P, L_i^P)$. Neka su funkcije za povezivanje dependenata i glava i funkcije za dohvaćanje sintaktičkih funkcija pojedinih riječi zadane, opet u skladu s ranijim definicijama, na sljedeći način.

$$h_i^R(j) = l \Leftrightarrow (l, j) \in E_i^R, \qquad h_i^P(j) = l \Leftrightarrow (l, j) \in E_i^P$$
$$d_i^R(j) = r \Leftrightarrow \exists l: (l, j, r) \in L_i^R, \qquad d_i^P(j) = r \Leftrightarrow \exists l: (l, j, r) \in L_i^P$$

Točnost označenoga (LAS) i neoznačenoga (UAS) povezivanja s obzirom na parsanje čitavoga teksta, analogno definicijama za pojedine rečenice, opisuje se na sljedeći način.

$$\mathrm{LAS}(T) = \frac{1}{n} \sum_{i=1}^{m} \sum_{j=1}^{k_i} \delta\left(h_i^R(j), h_i^P(j)\right) \cdot \delta\left(d_i^R(j), d_i^P(j)\right)$$

$$\mathrm{UAS}(T) = \frac{1}{n} \sum_{i=1}^{m} \sum_{j=1}^{k_i} \delta\left(h_i^R(j), h_i^P(j)\right)$$

Vrijedi primijetiti kako su ove mjere točnosti definirane na razini riječi, a ne na razini rečenica. Nadalje, može se (usp. Nivre 2006:140) na sličan način definirati i mjera potpune točnosti (EM) na razini testnoga uzorka, u dvije inačice koje se nazivaju mjerom potpune

označene točnosti (LEM) i mjerom potpune neoznačene točnosti (UEM). Ranije prikazana mjera potpune točnosti (EM) u tome je smislu zapravo predstavljala mjeru potpune označene točnosti[65].

$$\mathrm{LEM} = \frac{1}{m}\sum_{i=1}^{m} \delta\left(h_i^R(j), h_i^P(j)\right) \cdot \delta\left(d_i^R(j), d_i^P(j)\right)$$

$$\mathrm{UEM} = \frac{1}{m}\sum_{i=1}^{m} \delta\left(h_i^R(j), h_i^P(j)\right)$$

Osim vrjednovanja ukupne točnosti parsanja nekoga teksta, može se – s pomoću upravo postavljenoga matematičkog uređaja – također mjeriti i vrjednovati točnost dodjele pojedinih sintaktičkih funkcija (usp. Kübler i dr. 2009:79). U tome se slučaju najčešće koristi okvir za vrjednovanje s pomoću izračuna *preciznosti* (en. *precision*, P) i *odziva* (en. *recall*, R) sustava te njihove harmonijske sredine, odnosno *F_β-mjere* (en. *F_β-measure*, Rijsbergen 1979). Za danu sintaktičku funkciju, odnosno ovisnosnu relaciju r taj se okvir ovdje definira na sljedeći način (usp. Nivre 2006:141).

$$P(r) = \frac{\left|\left\{w_j \in T: d_i^P(j) = d_i^R(j) = r, h_i^P(j) = h_i^R(j)\right\}\right|}{\left|\left\{w_j \in T: d_i^P(j) = r\right\}\right|}$$

$$R(r) = \frac{\left|\left\{w_j \in T: d_i^P(j) = d_i^R(j) = r, h_i^P(j) = h_i^R(j)\right\}\right|}{\left|\left\{w_j \in T: d_i^R(j) = r\right\}\right|}$$

$$F_\beta(r) = (1 + \beta^2) \cdot \frac{P(r) \cdot R(r)}{\beta^2 \cdot P(r) + R(r)}$$

Pri vrjednovanju točnosti ovisnosnoga parsanja mjera preciznosti označavanja nekom sintaktičkom funkcijom, odnosno ovisnosnom relacijom predstavlja mjeru točnosti parsera pri dodjeli te sintaktičke funkcije. Preciznost, dakle, iskazuje omjer točnih dodjela i svih dodjela neke sintaktičke funkcije od strane ovisnosnoga parsera. S druge strane, mjera odziva predstavlja mjeru pokrivenosti neke sintaktičke funkcije parserom u odnosu na banku stabala. Dakle, odziv iskazuje omjer svih točnih dodjela sintaktičke funkcije od strane ovisnosnoga parsera i svih dodjela te sintaktičke funkcije u referentnome uzorku. Budući da su preciznost i

[65] (Nivre 2006:140) definira mjeru točnosti povezivanja (AS) te mjeru potpune točnosti (EM) pa varijante koje se odnose na označenu i neoznačenu točnost označava indeksiranjem slovima L i U. Prema toj notaciji, dakle, postojale bi mjere AS_L, AS_U, EM_L i EM_U.

odziv nepotpune, odnosno nesamostalne mjere – savršenu preciznost moguće je postići što manjim brojem, a odziv što većim brojem dodjela neke sintaktičke funkcije – uvodi se njihova harmonijska sredina[66] kao jedinstvena i samostalna mjera koja povezuje preciznost i odziv i dovodi ih u odnos u postupku optimizacije. Harmonijska sredina u vidu mjere F_β definirana je tako da se preciznost i odziv mogu označiti kao jednakovrijedni ili se može među njima, ovisno o odabiru faktora β, uspostaviti nejednak odnos u kojemu se jedna mjera nagrađuje više od druge. S obzirom na vrjednovanje ovisnosnoga parsanja i ostale mjere za vrjednovanje, najčešće se postavlja $\beta = 1$, pa slijedi da je mjera F_1 za vrjednovanje točnosti parsanja nekom sintaktičkom funkcijom jednaka

$$F_1(r) = \frac{2P(r) \cdot R(r)}{P(r) + R(r)}$$

Također se može mjeriti (usp. Nivre 2006:141) i "koliko često je nekomu dependentu kojem pripada ovisnosna relacija, odnosno sintaktička funkcija r dodijeljena valjana glava". Ta se mjera naziva *točnošću neoznačenoga povezivanja za sintaktičku funkciju* (en. *unlabeled attachment score for dependency type*) i definirana je sljedećom formulom.

$$UAS(r) = \frac{\left|\{w_j \in T: d_i^R(j) = r, h_i^P(j) = h_i^R(j)\}\right|}{\left|\{w_j \in T: d_i^R(j) = r\}\right|}$$

Pri vrjednovanju parsera za potrebe natjecanja CoNLL 2006 i 2007, kako je ranije naznačeno, korištene su mjere točnosti LAS i UAS, odnosno mjera točnosti označenoga i neoznačenoga povezivanja na razini riječi za čitav testni uzorak.

2.2.1.4 Rezultati

Točnosti pojedinih parsera s natjecanja CoNLL 2006 i 2007 prema mjerama LAS i UAS navedene su u (Buchholz i Marsi 2006) te (Nivre i dr. 2007a), a zbirni komentar rezultata dan je u (Kübler i dr. 2009:85). Zbirno, rezultati su pokazali (usp. Kübler i dr. 2009:84) kako se u paradigmi temeljenoj na podatcima "ovisnosno parsanje može uspješno primijeniti na širok spektar različitih jezika", ali ipak "uz očigledne razlike u točnosti prema skupinama jezika". Primjerice, rezultati s natjecanja CoNLL 2007 podijeljeni su prema točnosti parsanja u tri vrijednosne skupine:

[66] Koristi se harmonijska sredina, a ne aritmetička sredina, isključivo iz povijesnih razloga, odnosno zbog uporabe u području crpljenja obavijesti, prema (Rijsbergen 1979). Aritmetička sredina preciznosti i odziva predstavljala bi jednakovrijednu mjeru točnosti.

1. skupinu s niskom točnošću parsanja, oko 0.76 LAS, u koju je svrstan arapski, baskijski i grčki jezik,
2. skupinu sa srednjom točnošću parsanja, oko 0.80 LAS, u koju je svrstan češki, mađarski i turski jezik, i
3. skupinu s visokom točnošću parsanja, između 0.84 i 0.90 LAS, u koju je svrstan katalonski, kineski, engleski i talijanski jezik.

S obzirom na ranije opisane značajne razlike u veličinama uzoraka za treniranje jezičnih modela, koje se ne prikazuju u ovoj razredbi točnosti parsanja, dolazi se do zaključka (usp. Kübler i dr. 2009:85) kako "točnost parsanja značajnije ovisi o svojstvima parsanih jezika, nego o svojstvima uzorka za treniranje" budući da skupini jezika s niskom i srednjom točnošću parsanja pripadaju jezici složene morfologije i relativno slobodnoga redoslijeda riječi u rečenici, dok skupini s visokom točnošću pripadaju jezici siromašnije morfologije i fiksnoga ili ograničenoga redoslijeda riječi. Smisleno je stoga razmatrati i pojedine ovisnosne parsere prema njihovu vrjednovanju na pojedinim jezicima, odnosno jezičnim uzorcima, a pritom imati na umu – za potrebe prikazivanja modela pojedinih parsera – one jezike koji dijele svojstva s hrvatskim jezikom[67].

Rezultati natjecanja CoNLL 2006 otkrivaju da su najbolji rezultati – prije svega u smislu najvećih postignutih vrijednosti za mjeru LAS – na jedanaest od dvanaest obaveznih jezika[68] razdijeljeni između dva sustava, odnosno ovisnosna parsera naziva MaltParser[69] (Nivre i dr. 2007b) i MSTParser[70] (McDonald i dr. 2006). Podjela rezultata među ovim parserima dana je u tablici 2-2, a masnim su slovima označeni najviši rezultati na natjecanju uz poštovanje statističke značajnosti rezultata.

Tablica 2-2 MaltParser i MSTParser na natjecanju CoNLL 2006

	ara	**kin**	**češ**	**dan**	**niz**	**njem**	**jap**	**por**	**slo**	**špa**	**šve**	**tur**	**ukupno**
MST	**66.9**	85.9	**80.2**	**84.8**	**79.2**	**87.3**	**90.7**	**86.8**	**73.4**	**82.3**	82.6	63.2	**80.3**
Malt	**66.7**	86.9	78.4	**84.8**	**78.6**	85.8	**91.7**	**87.6**	70.3	**81.3**	**84.6**	**65.7**	**80.2**

[67] Pritom se, s obzirom na ranije nabrojane jezike koji su sudjelovali u natjecanjima, prije svega misli na češki i slovenski jezik budući da pripadaju obitelji slavenskih jezika te dijele svojstva složene morfologije i relativno slobodnoga redoslijeda riječi.
[68] Bugarski je jezik izdvojen kao neobavezan, a navedeni sustavi nisu postigli najbolji rezultat pri parsanju kineskoga jezika.
[69] Dostupan na URL-u http://maltparser.org/ (2012-03-06).
[70] Dostupan na URL-u http://sourceforge.net/projects/mstparser (2012-03-06).

Iz tablice 2-2 vidljivo je kako su – osim na zadatku parsanja tekstova pisanih kineskim jezikom – parseri MaltParser i MSTParser bili uvjerljivo najbolji parseri na natjecanju. S obzirom na istraživanje koje se predstavlja u ovome tekstu, vrijedi primijetiti kako je sustav MSTParser na testiranju bio, u okvirima statističke značajnosti, bolji u parsanju slovenskoga i češkoga jezika. Nadalje, s obzirom na prikazane rezultate, oba parsera i pripadajući teorijski modeli na kojima su izgrađeni odabiru se ovdje za detaljno prikazivanje dalje u tekstu.

Tablica 2-3 Pet najboljih parsera s natjecanja CoNLL 2007

	ara	**bask**	**kat**	**kin**	**češ**	**eng**	**grč**	**mađ**	**tal**	**tur**	**ukupno**
Nilsson	79.79 (2)	**76.52 (1)**	**76.94 (1)**	**88.70 (1)**	75.82 (15)	77.98 (3)	88.11 (5)	74.65 (2)	**80.27 (1)**	**84.40 (1)**	**80.32 (1)**
Nakagawa	78.22 (5)	75.08 (2)	72.56 (7)	87.90 (3)	83.84 (2)	**80.19 (1)**	88.41 (3)	**76.31 (1)**	76.74 (8)	83.61 (3)	80.29 (2)
Titov	**79.81 (1)**	74.12 (6)	75.49 (3)	87.40 (6)	82.14 (7)	77.94 (4)	88.39 (4)	73.52 (10)	77.94 (4)	82.26 (6)	79.90 (3)
Sagae	75.91 (10)	74.71 (4)	74.64 (6)	88.16 (2)	**84.69 (1)**	74.83 (8)	89.01 (2)	73.58 (8)	79.53 (2)	83.91 (2)	79.90 (4)
Hall, J.	79.24 (3)	74.75 (3)	74.99 (5)	87.74 (4)	83.51 (3)	77.22 (6)	85.81 (12)	74.21 (6)	78.09 (3)	82.48 (5)	79.80 (5)

Rezultati s natjecanja u višejezičnome ovisnosnom parsanju CoNLL 2007 navedeni su u tablici 2-3 za pet parsera s najboljim prosječnim rezultatima prema mjeri točnosti LAS[71]. U sklopu toga natjecanja parseri nisu označeni pripadajućim imenima – budući da se neki od sustava za parsanje mogu nazivati i *generatorima parsera* (en. *parser generator*), pa je smislenije primjenu jednoga jezičnog modela proglasiti jednim parserom, utoliko što više različitih jezičnih modela može biti stvoreno jednim generatorom – već imenima istraživača koji su proveli eksperimente. Stoga vrijedi napomenuti kako su sustavi navedeni u tablici 2-3

[71] Za tablice 2-2 i 2-3 vrijedi napomenuti kako je mjera LAS izdvojena kao važnija budući da predstavlja obuhvatniju mjeru, no rezultati vrjednovanja mjerom UAS u potpunosti su usklađeni s njima.

pod nazivima "Nilsson" i "Hall, J." opisani u (Hall i dr. 2007) i predstavljaju inačice sustava MaltParser, koji je ranije izdvojen pri opisu rezultata s natjecanja CoNLL 2006. Ovisnosni parser naziva "Nakagawa" opisan je u (Nakagawa 2007), parser naziva "Titov" u (Titov i Henderson 2006, 2007) i predstavlja sustav naziva IDP[72] te je konačno parser naziva "Sagae" opisan u (Sagae i Lavie 2006) i (Sagae i Tsujii 2007).

Prema izvještaju s natjecanja CoNLL 2007 (Nivre i dr. 2007:923), ovisnosni parseri koji su sudjelovali u natjecanjima CoNLL 2006 i 2007 – što se u potpunosti odnosi na one najbolje, izdvojene i navedene u prethodnim tablicama s rezultatima – svrstavaju se u dvije skupine prema paradigmi unutar koje pristupaju problemu ovisnosnoga parsanja. Prema toj razredbi, dvije su skupine[73] učenja i zaključivanja kojima ovi parseri pripadaju:

1. parseri temeljeni na grafovima (en. *graph-based parsers*) i
2. parseri temeljeni na prijelazima (en. *transition-based parsers*).

Uzimajući u obzir rezultate s natjecanja CoNLL 2006 i 2007 i ovu razredbu ovisnosnih parsera temeljenih na podatcima, dalje se prikazuju ova dva pristupa – parsanje temeljeno na grafovima i parsanje temeljeno na prijelazima – te se za ilustraciju tih pristupa odabiru parseri MaltParser (temeljen na prijelazima) i MSTParser (temeljen na grafovima). Obje se metode promatraju s gledišta ranije (slika 2-6) predstavljenoga modela ovisnosnoga parsera, odnosno korištenoga jezičnog modela sintakse (en. *model*) i njegova stvaranja, odnosno učenja (en. *learning*), svojstava parsnoga algoritma koji za ulaznu rečenicu pronalazi najbolje parsno stablo, odnosno zaključuje o najboljemu rješenju problema parsanja rečenice s pomoću jezičnoga modela (en. *inference*).

2.2.2 Ovisnosno parsanje temeljeno na grafovima

Pristup ovisnosnomu parsanju koji se temelji na grafovima, odnosno teoriji grafova zasnovan je na promišljanju o ovisnosnim stablima kao grafovima i posljedičnoj obradbi ovisnosnih stabala algoritmima koji su dostupni za obradbu grafova. Naime, ako se ovisnosna stabla promatra kao grafove, s gledišta traženja valjanoga modela ovisnosnoga parsanja prema ranije postavljenim kriterijima, vrijede sljedeći sudovi (Kübler i dr. 2009:41).

[72] ISBN Dependency Parser, dostupan na URL-u http://cui.unige.ch/~titov/idp/ (2012-03-08).
[73] Navodi se (usp. Nivre i dr. 2007:923) kako izvještaj s CoNLL-a 2006 (Buchholz i Marsi 2006) naziva ove pristupe slijednima ili koračnima (en. *stepwise*) i onima koji obuhvaćaju sve parove (en. *all-pairs*).

1. Ovisnosna stabla su usmjereni, aciklički grafovi koji zadovoljavaju i određeni broj ranije definiranih i dodatno ograničavajućih svojstava.
2. U području teorije grafova kao grani matematike i računalne znanosti postoji veliki broj algoritama za obradbu svih vrsta grafova, a ti su algoritmi "među najbolje objašnjenima u računalnoj znanosti uopće".

Iz tih sudova logički slijedi zaključak kako se algoritmi dostupni za obradbu grafova mogu primijeniti i na ovisnosna stabla. Međutim, nejasno je jesu li ti algoritmi smisleni s gledišta ovisnosnoga parsanja, odnosno mogu li se povezati rezultati njihova rada s ciljem pronalaženja ovisnosnoga stabla koje najbolje objašnjava ulaznu rečenicu. Dakle, kod pristupa ovisnosnomu parsanju temeljenih na teoriji grafova postavlja se sljedeće pitanje (Kübler i dr. 2009:41): "Je li moguće izraditi ovisnosne parsere koji koriste postojeće algoritme za grafove i stabla?" Budući da ranija rasprava o natjecanjima CoNLL 2006 i 2007 na to pitanje odgovara potvrdno, ovdje se opisuju opća načela toga pristupa parsanju, pa se ona dodatno opisuju s pomoću njihove izvedbe u sustavu MSTParser.

2.2.2.1 Definicije

Kod ovisnosnoga parsanja temeljenoga na grafovima jezični se model nastoji izgraditi izravnim modeliranjem postupka stvaranja ovisnosnih stabala iz njegovih sastavnih dijelova – podstabala, odnosno podgrafova. Preciznije, u osnovi svih ovih metoda ovisnosnoga parsanja nalazi se vrjednovanje vjerojatnosti dodjele nekoga ovisnosnog stabla (odabranoga iz skupa svih mogućih ovisnosnih stabala) za danu rečenicu preko vrjednovanja vjerojatnosti pojavljivanja njegovih podgrafova s obzirom na samu rečenicu. To vrjednovanje provodi se pomoću funkcije za vrjednovanje (en. *score function*, *fitness function*). Slijedi njezina formalizacija.

Neka je zadana ulazna rečenica s i ovisnosni graf G koji predstavlja jedno ovisnosno stablo iz skupa svih mogućih ovisnosnih stabala za danu rečenicu, $G = (V, E, L), G \in G_s$. Neka je nadalje zadan skup svih mogućih podgrafova – odnosno sastavnih elemenata ovisnosnoga grafa G – kao Ψ_G te neka je jedan podgraf iz toga skupa definiran kao $\psi_i \in \Psi_G, 1 \leq i \leq |\Psi_G|$. Funkcija kojom se vrjednuje primjerenost toga ovisnosnog stabla za sintaktički opis ulazne rečenice definira se na sljedeći način.

$$\text{score} : G_s \longrightarrow \mathbb{R}, \qquad \text{score}(G) = \text{score}(V, E, L) \in \mathbb{R}$$

$$\text{score}(G) = f(\psi_1, \ldots, \psi_q), \forall \psi_i \in \Psi_G, \qquad \text{score}(G) = \sum_{i=1}^{|\Psi_G|} \lambda_{\psi_i}$$

Formalizacijom vrjednovanja ovisnosnih stabala kao grafova i s pomoću podgrafova u obliku funkcije za vrjednovanje score izneseno je sljedeće. Funkcija score definira se nad skupom svih ovisnosnih stabala koja je moguće dodijeliti nekoj rečenici, tako da prima kao parameter jedno ovisnosno stablo, a vraća neki broj iz skupa realnih brojeva koji predstavlja numeričko vrjednovanje toga ovisnosnog stabla. Nadalje, funkcija score za vrjednovanje jednoga ovisnosnog stabla – stoga definirana nad jednim ulaznim parametrom – pobliže se označava preko funkcije $f : \Psi_G \longrightarrow \mathbb{R}$, koja prima na ulazu onoliko parametara koliko je podgrafova u pripadajućemu ovisnosnom stablu i pri vrjednovanju ovisnosnoga stabla u obzir uzima doprinos svakoga njegova podgrafa. Budući da se funkcija za vrjednovanje podgrafova kod primjene na jedan podgraf ovisnosnoga stabla najčešće svodi na provjeru opisa toga podgrafa na jezičnome modelu, $f(\psi_i) = \lambda_{\psi_i}$, vrjednovanje čitavoga ovisnosnog stabla preko vrjednovanja podgrafova svodi se na zbrajanje provjera pojedinih podgrafova na jezičnome modelu.

Definicije navedenih funkcija za vrjednovanje prikladnosti stabala i podgrafova su općenite i ilustrativne, odnosno konceptualne budući da se njihove domene (parametri) i kodomene (povratne vrijednosti), kao i načini rada, mogu značajno razlikovati u pojedinim pristupima koji dijele svojstvo temeljenosti na grafovima. S obzirom na to, svaki sustav za ovisnosno parsanje koji se temelji na grafovima mora definirati barem sljedeće (usp. Kübler i dr. 2009:42):

1. skup svih podgrafova Ψ_G ovisnosnoga stabla G, što implicira i definiranje svojstava pojedinih podgrafova $\psi_i \in \Psi_G$,
2. jezični, odnosno sintaktički model preko kojega se izvršava vrjednovanje pojedinih ovisnosnih stabala, $\lambda = \{\lambda_\psi, \forall \psi \in \Psi_G, \forall G, \forall s\}$,
3. postupak izvođenja jezičnoga modela λ iz banke ovisnosnih stabala i
4. algoritam za parsanje, $h(s, \Gamma, \lambda) = \arg\max_{G \in G_s} \text{score}(G)$.

Najčešći pristup parsanju temeljenomu na grafovima jest onaj u kojem se podgrafovi $\psi_i \in \Psi_G$ – kao vrijednosti preko kojih se određuje mjera valjanosti nekoga ovisnosnog stabla kao parsanja neke ulazne rečenice – definiraju na trivijalan način, odnosno ograničenjem

prostiranja svakoga podgrafa na samo jednu ovisnosnu relaciju. Ovaj pristup modeliranju, u kojem se podgraf izjednačava s ovisnosnom relacijom, rezultira *relacijski uvjetovanim modelima* (en. *arc-factored models*) parsanja temeljenoga na grafovima.

Neka je ovisnosno stablo $G = (V, E, L)$ – s pripadajućim skupom veza E među riječima i funkcijom $L : E \longrightarrow R$ koja ovisnosnim relacijama dodjeljuje sintaktičke oznake definirano kao ranije. Neka su nadalje skup veza E i funkcija za dodjelu sintaktičkih oznaka L povezane u skup označenih ovisnosnih relacija A na sljedeći način:

$$\forall a \in A, a = (w_i, r, w_j) \Leftrightarrow e = (i, j) \in E, L(e) = r$$

Dakle, svrha skupa označenih ovisnosnih relacija jest povezati – zbog jednostavnosti i čitljivosti za potrebe definiranja parsanja temeljenoga na grafovima – skup veza E i njegovu funkcijsku poveznicu L sa sintaktičkim oznakama, ranije razdvojene kako bi se pojasnila dvojna struktura ovisnosnoga stabla i posljedično preciznije definirale označene i neoznačene mjere vrjednovanja točnosti. Sada vrijedi i $G = (V, A) \Leftrightarrow G = (V, E, L)$, pa se dalje u ovome tekstu može koristiti bilo koja od te dvije definicije ovisnosnoga grafa, odnosno stabla.

Kod relacijski uvjetovanoga modela parsanja temeljenoga na grafovima, prema ranijim definicijama grafa i podgrafova, ograničenje podgrafa na jednu ovisnosnu relaciju formalno je određeno na sljedeći način:

$$\Psi_G = A$$
$$\lambda_\psi = \lambda_{(w_i, r, w_j)}, \lambda_\psi \in \mathbb{R}, \forall (w_i, r, w_j) \in A$$

Opisno, skup svih mogućih podgrafova ovisnosnoga stabla G izjednačava se sa skupom svih ovisnosnih relacija u tome ovisnosnom stablu[74], a pojedini faktori u izgradnji funkcije za vrjednovanje predloženih ovisnosnih stabala izračunavaju se nad pojedinim ovisnosnim relacijama iz toga skupa. Iz ovih ograničenja i definicija slijedi da se u relacijski uvjetovanim modelima svakoj ovisnosnoj relaciji dodjeljuje neki realni broj koji predstavlja njegovu vrijednost. Jezični model λ u tome slučaju predstavlja skup svih ovisnosnih relacija iz A s pridruženim mjerama vrijednosti ili značaja (en. *weight*).

[74] Navodi se u (Kübler i dr. 2009:42) kako je ova formulacija zapravo "zlorabljenje notacije" budući da jedna ovisnosna relacija po definiciji nije ovisnosni (pod)graf. Jedna ovisnosna relacija $a = (w_i, r, w_j)$ morala bi se formalno pretvoriti u podgraf $G_a = (V_a, A_a), V_a = \{w_i, w_j\}, A_a = \{(w_i, r, w_j)\}$.

Funkcija za vrjednovanje predloženih ovisnosnih stabala za ulaznu rečenicu sada se može preciznije definirati na sljedeći način, koji zapravo govori kako se vrijednosna mjera nekoga ovisnosnog stabla računa iz vrijednosnih mjera ovisnosnih relacija među njegovim čvorovima:

$$\mathrm{score}(G) = \sum_{(w_i,r,w_j)\in A} \lambda_{(w_i,r,w_j)}$$

Ako se pretpostavi postojanje sintaktičkoga modela λ izgrađenoga iz banke ovisnosnih stabala, problem relacijski uvjetovanoga parsanja temeljenoga na grafovima može se proširivanjem ranije definicije preciznije označiti kao:

$$h(s,\Gamma,\lambda) = \arg\max_{G\in G_s} \mathrm{score}(G) = \arg\max_{G\in G_s} \sum_{(w_i,r,w_j)\in A} \lambda_{(w_i,r,w_j)}$$

Traži se, dakle, upravo ono ovisnosno stablo $G \in G_s$ za koje je zbroj svih vrjednovanja $\lambda_{(w_i,r,w_j)}$ ovisnosnih relacija $(w_i, r, w_j) \in A$ na nekome smislenom sintaktičkom modelu λ koji sadrži vrjednovanja ovisnosnih relacija najveći, odnosno najbolji.

Brojčana vrjednovanja $\lambda_{(w_i,r,w_j)}$ ovisnosnih relacija najčešće predstavljaju na banci ovisnosnih stabala zasnovanu procjenu vjerojatnosti pojavljivanja ovisnosnih relacija u sintaktičkome modelu parsanoga prirodnog jezika, $\lambda_{(w_i,r,w_j)} = \hat{p}(w_i, r, w_j) \in [0,1]$, te se vrijednosti u algoritmu za parsanje najčešće ne zbrajaju, nego množe kako bi i dalje predstavljale valjane vjerojatnosti. Stoga se definicija parsanja mijenja:

$$h(s,\Gamma,\lambda) = \arg\max_{G\in G_s} \prod_{(w_i,r,w_j)\in A} \lambda_{(w_i,r,w_j)}$$

Vjerojatnost je po definiciji realni broj definiran u zatvorenome intervalu $[0,1]$. Budući da jedan izračun vjerojatnosti prema definiciji parsanja može uključivati velik broj množenja vjerojatnosti, s ozbirom na mogući veliki broj ovisnosnih relacija po parsnome stablu i na moguće posljedice te pojave na računalne izvedbe takvih parsera[75], definicija se prilagođava uporabom logaritama vjerojatnosti umjesto samih vjerojatnosti:

[75] S obzirom na svojstvo broja ovisnosnih relacija ovisnosnoga stabla, $|E| = |V| - 1$, za rečenicu od 50 riječi bit će u ovisnosnome stablu 49 ovisnosnih relacija. Neka svaka relacija ima prosječnu vjerojatnost od 0.05 prema

$$h(s,\Gamma,\lambda) = \arg\max_{G\in G_s} \log \prod_{(w_i,r,w_j)\in A} \lambda_{(w_i,r,w_j)} = \arg\max_{G\in G_s} \sum_{(w_i,r,w_j)\in A} \log \lambda_{(w_i,r,w_j)}$$

Svojstva logaritamske funkcije u izmijenjenoj definiciji također omogućavaju uporabu zbrajanja umjesto množenja vjerojatnosti, a njome se implicira da u jezičnome modelu vrijedi za pojedinu ovisnosnu relaciju da je $\lambda_{(w_i,r,w_j)} \equiv \log \lambda_{(w_i,r,w_j)}$. Model relacijski uvjetovanoga ovisnosnog parsera temeljenoga na grafovima $M = (\Gamma, \lambda, h)$ definira se u ovome slučaju s pomoću

1. skupa ograničenja Γ, kojim se zahtijeva da izlazi iz algoritma za parsanje budu ograničeni na skup parsnih stabala mogućih za ulazne rečenice,
2. sintaktičkoga modela λ, koji predstavlja skup $\lambda = \{\lambda_{(w_i,r,w_j)}, \forall(w_i, r, w_j) \in A\}$ koji najčešće sadrži vjerojatnosti pojavljivanja pojedinih ovisnosnih relacija procijenjene prebrojavanjem u banci ovisnosnih stabala i
3. parsnoga algoritma h, koji rješava gore navedeni optimizacijski problem parsanja jedne rečenice, $h(s,\Gamma,\lambda) = \arg\max_{G\in G_s} \sum_{a\in A} \log \lambda_a$.

Osim relacijski uvjetovanih, postoje i drugi modeli ovisnosnih parsera temeljenih na grafovima, kod kojih se podgrafovi ne ograničavaju na ovisnosne relacije. Oni se u ovome tekstu ne razmatraju, ali vrijedi istaknuti (usp. Kübler i dr. 2009:56-61) kako zbog izračunske složenosti – budući da su "projektivne inačice takvih algoritama polinomske, a neprojektivne eksponencijalne složenosti" – najčešće ne udovoljavaju ranije postavljenom općem kriteriju vremenske i prostorne složenosti ovisnosnoga parsanja.

Slijedi prikaz pristupa izradi sintaktičkoga modela za relacijski uvjetovano ovisnosno parsanje temeljeno na grafovima prebrojavanjem pojavnosti iz banke ovisnosnih stabala pa potom prikaz algoritma za parsanje tim sintaktičkim modelom.

2.2.2.2 Algoritam za izradu jezičnoga modela

Izrada sintaktičkoga jezičnog modela u zadanim okvirima relacijski uvjetovanoga ovisnosnog parsanja temeljenoga na grafovima zahtijeva odrediti

---

jezičnome modelu. Izračunati 0.05^{49} i upotrijebiti taj izračun u daljnoj obradbi – koja može uključivati daljnja množenja – može se pokazati izvedbeno netrivijalnim s obzirom na memorijska ograničenja pri definiranju tipova podataka u programskim jezicima.

1. značenje jednoga parametra $\lambda_{(w_i,r,w_j)}$ preko kojega se utvrđuje poredak ovisnosnih relacija i posljedično poredak parsnih stabala kojima se može opisati neka ulazna rečenica i
2. način na koji se do modela $\lambda = \{\lambda_{(w_i,r,w_j)}\}$ dolazi iz banke ovisnosnih stabala, odnosno algoritam za izvođenje jezičnoga modela iz parsanoga teksta.

Vrijednost $\lambda_{(w_i,r,w_j)}$ ranije je – za potrebe opisivanja problema ovisnosnoga parsanja temeljenoga na grafovima i ograničenoga na jedinične podgrafove – definirana na skupu realnih brojeva, a potom povezana s procjenom vjerojatnosti pojavljivanja ovisnosne relacije i definirana na zatvorenom intervalu $[0, 1]$. Međutim, relacijski uvjetovani ovisnosni parseri temeljeni na grafovima u stvarnosti najčešće vrjednuju ovisnosnu relaciju preko unaprijed zadanoga skupa značajki, pa vrijednost $\lambda_{(w_i,r,w_j)}$ zapravo predstavlja jednu pojavnost toga skupa značajki, svojstvenu za pripadajuću ovisnosnu relaciju.

Neka je zadana funkcija $\mathbf{f} : X \longrightarrow Y, \mathbf{f}(x) = y$ koja preslikava neke ulazne podatke $x \in X$ u skup značajki $y = (y_1, \dots, y_m) \in Y$ koje opisuju te ulazne podatke. Funkcija $\mathbf{f}$ naziva se *funkcijom značajki* (en. *feature function*), a njezina povratna vrijednost $\mathbf{f}(x) = y$ najčešće se naziva *vektorom značajki* (en. *feature vector*) budući da sadrži niz značajki koje opisuju ulazne podatke[76]. Značajke pritom mogu biti brojčane ili simboličke, no najčešće se radi o realnim brojevima kojima su izdvojeno pridružena značenja. U tome slučaju vrijedi $\mathbf{f} : X \longrightarrow \mathbb{R}^m, \mathbf{f}(x) = y, y = (y_1, \dots, y_m) \in \mathbb{R}^m, \forall i, 1 \leq i \leq m, y_i \in \mathbb{R}$ (usp. Kübler i dr. 2009:18).

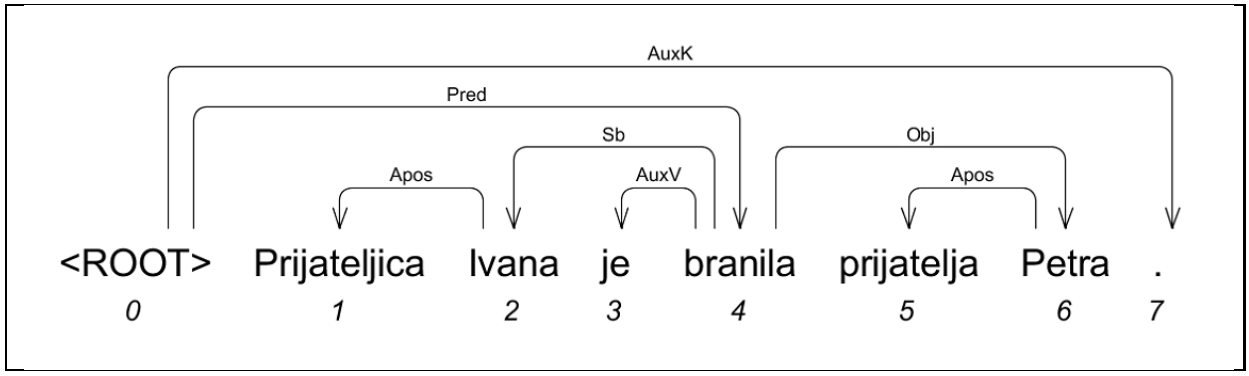
Neka je predefiniran skup značajki za opisivanje svake ovisnosne relacije i neka su pojedine značajke opisane realnim brojevima, $\mathbf{f}(w_i, r, w_j) \in \mathbb{R}^m$ i neka je tako zadanome vektoru značajki pridružen vektor $\mathbf{w} \in \mathbb{R}^m$ koji predstavlja brojčani zapis značajnosti, odnosno težine (en. *weight*) pojedinih značajki. Vrijedi:

$$\lambda_{(w_i,r,w_j)} = \mathbf{w} \cdot \mathbf{f}(w_i, r, w_j)$$

[76] Vektor se ovdje koristi u smislu matrice koja sadrži jedan stupac i m redaka, a ne u strogom smislu vektora u m-dimenzionalnome prostoru, iako su ti pojmovi zamjenjivi. Vektor se označava masnim slovima ili strjelicom kako bi se razlikovao od ostalih entiteta. Ovdje se koristi notacija s masnim slovima.

Skup od m značajki koje su predstavljene[77] vektorom realnih brojeva i pridruženim vektorom težina odabire se tako da opisuje "bilo koje relevantno svojstvo neke ovisnosne relacije" (usp. Kübler i dr. 2009:54). Tako se, primjerice, ovisnosna relacija ("branila", "Obj", "Petra") u projektivnoj rečenici iz primjera 2-15, koja se ovdje zbog ilustrativnosti ponavlja u primjeru 2-17, može opisati sljedećim skupom značajki:

1. pojavnim oblikom riječi w_i, odnosno glave ovisnosne relacije, $w_i = \text{"branila"}$,
2. pojavnim oblikom riječi w_j, odnosno dependenta relacije, $w_i = \text{"Petra"}$,
3. osnovnim oblikom riječi w_i, $\text{lemma}(w_i) = \text{"braniti"}$,
4. osnovnim oblikom riječi w_j, $\text{lemma}(w_j) = \text{"Petar"}$,
5. morfosintaktičkom oznakom riječi w_i, $\text{msd}(w_i) = \text{"V} * * * \text{"}$,
6. morfosintaktičkom oznakom riječi w_j, $\text{msd}(w_j) = \text{"Npmsa"}$,
7. ovisnosnom relacijom dependenta w_j prema glavi w_i, $r = \text{"Obj"}$,
8. osnovnim oblicima i morfosintaktičkim oznakama riječi u neposrednom okruženju od w_i i w_j, primjerice, w_{i-1}, w_{i+1}, w_{j-1} i w_{j+1} te
9. udaljenošću u broju riječi između w_i i w_j, $\text{distance}(w_i, w_j) = 1$.



Primjer 2-17 Rečenica s projektivnim ovisnosnim stablom

Ovaj popis značajki je ilustrativan, a skupovi značajki koji se koriste za opis ovisnosnih relacija u stvarnim sustavima za ovisnosno parsanje temeljenima na grafovima mogu biti i znatno složeniji, no najčešće uključuju i gore navedene značajke. U postupku pretvaranja ovih simboličkih značajki u brojčane vrijednosti razmatra se skup svih mogućih ostvarenja pojedine značajke. Primjerice, ako postoji trideset različitih ovisnosnih relacija u zadanome

[77] Postupak zapisivanja simboličkih značajki numeričkim vrijednostima naziva se *binarizacija* (en. *binarization*). Primjerice, ako treba simboličke vrijednosti *a*, *b*, *c* i *d* zapisati numerički, u tome im se postupku treba dodijeliti jedinstvene brojčane identifikatore, poput *00*, *01*, *10*, *11* (binarni) ili *-2*, *-1*, *1*, *2* (realni). Odabir numeričkih vrijednosti za pojedine simboličke vrijednosti ovisi o daljnjoj uporabi tih vrijednosti u izračunima.

sintaktičkom formalizmu, potrebno je svakoj dodijeliti jedinstvenu brojčanu vrijednost. Najčešće se za svaki par značajke i vrijednosti dodjeljuje binarna vrijednost 1 ako je par ostvaren i binarna vrijednost 0 ako par nije ostvaren[78].

S obzirom na raniju definiciju sintaktičkoga modela $\lambda = \left\{\lambda_{(w_i,r,w_j)}, \forall(w_i, r, w_j) \in A\right\}$ za relacijski uvjetovano ovisnosno parsanje temeljeno na grafovima i poveznicu pojedinoga parametra s popisom značajki, $\lambda_{(w_i,r,w_j)} = \mathbf{w} \cdot \mathbf{f}(w_i, r, w_j)$, sada vrijedi i da je čitav sintaktički jezični model definiran kao popis značajki za sve ovisnosne relacije koje se u njemu pojave, odnosno koje su opažene na banci stabala iz koje je izveden:

$$\lambda = \{\mathbf{w} \cdot \mathbf{f}(w_i, r, w_j), \forall(w_i, r, w_j) \in A\}$$

Prema toj definiciji jezičnoga modela, i problem parsanja može se definirati u obliku traženja globalnoga maksimuma nad skupom značajki i pripadajućih težina pojedinih ovisnosnih relacija koje su moguće za danu ulaznu rečenicu:

$$h(s, \Gamma, \lambda) = \arg\max_{G \in G_s} \sum_{(w_i,r,w_j) \in A} \log \lambda_{(w_i,r,w_j)} = \arg\max_{G \in G_s} \sum_{(w_i,r,w_j) \in A} \mathbf{w} \cdot \mathbf{f}(w_i, r, w_j)$$

Sintaktički model λ u ovako zadanome okruženju već sadrži vektore težina $\mathbf{w}$ za vektore značajki $\mathbf{f}$ koji su izvedeni iz banke ovisnosnih stabala. Osnovni algoritam za stvaranje takvoga jezičnog modela iz banke ovisnosnih stabala zasnovan je na rješavanju gore navedenoga optimizacijskog problema za svaku rečenicu iz banke stabala i uspoređivanju toga s referentnim rješenjem, a naziva se *algoritam perceptron* (en. *perceptron algorithm*) i pripada skupini *algoritama za učenje temeljeno na zaključivanju* (en. *inference-based learning*). Slijedi prikaz algoritma.

Neka je stoga zadana banka ovisnosnih stabala $T = \{(s_t, G_t)\}_{t=1}^{|T|}$ koja sadrži provjerene parove rečenica i pripadajućih ovisnosnih stabala. Algoritam perceptron uzima jedan po jedan par iz banke stabala i za svaki taj par rečenice i ovisnosnoga stabla

1. za rečenicu s_t predlaže ovisnosno stablo G' rješavanjem problema $h(s, \Gamma, \lambda)$ s pomoću nekoga od dostupnih algoritama iz teorije grafova,

[78] Posljedično, većini parova u opisu jedne značajke dodijeli se vrijednost 0 budući da se u opisu značajki najčešće ostvari samo jedna od mogućih pojavnosti. Stoga i vektor značajki sadrži većinom binarne 0, osim na mjestima ostvaraja, što omogućava pristupe izvedbi programskih rješenja koji se koriste tom činjenicom i koji se nazivaju oskudnim predstavljanjem i izračunom (en. *sparse representation and calculation*).

2. uspoređuje dobiveno kandidatsko ovisnosno stablo s ovisnosnim stablom G_t koje je rečenici s_t dodijeljeno u banci stabala,
3. ukoliko se predloženo ovisnosno stablo razlikuje od referentnoga, u vektor težina dodaju se vrijednosti vektora značajki za one ovisnosne relacije koje se ne razlikuju među stablima, a oduzimaju vrijednosti za ovisnosne relacije po kojima se stabla razlikuju,
4. postupak se ponavlja dok se predloženo ovisnosno stablo ne izjednači s referentnim ili dok se ne dosegne neki fiksni broj ponavljanja.

Algoritam je prikazan i detaljno raščlanjen u (Collins 2002). Navodi se (usp. Kübler i dr. 2009:56) kako on "za slijedno odvojive podatke jamči pronalazak vektora težina koji idealno odgovara danoj banci ovisnosnih stabala, odnosno idealno je klasificira". Njegov pseudokod prikazan je kao algoritam 2-2.

Po završetku rada ovoga algoritma za učenje sintaktički se model izračunava dodjelom vrijednosti pojedinim ovisnosnim relacijama, $\lambda_{(w_i,r,w_j)} = \mathbf{w} \cdot \mathbf{f}(w_i, r, w_j)$, za svaku ovisnosnu relaciju svake rečenice u banci stabala, $\lambda = \{\lambda_{(w_i,r,w_j)}\}$. Algoritam perceptron, kao algoritam za izradu jezičnoga modela parsera, zasnovan je na rješavanju problema parsanja $h(s, \Gamma, \lambda)$ nekim već dostupnim algoritmom s područja teorije grafova. Slijedi prikaz algoritama za ovisnosno parsanje temeljenih na teoriji grafova i ovako izgrađenome jezičnom modelu parsera.

perceptron(T)

neka je $T = \{(s_t, G_t)\}_{t=1}^{|T|}$ banka ovisnosnih stabala

neka je $\mathbf{w} = 0$

za svaki $n = 1 \ldots N$

za svaki $t = 1 \ldots |T|$

neka je $G' = h(s_t, \Gamma, \lambda) = \arg\max_{G' \in G_{s_t}} \sum_{(w_i,r,w_j) \in A'} \mathbf{w} \cdot \mathbf{f}(w_i, r, w_j)$

ako je $G' \neq G_t$

$\mathbf{w} = \mathbf{w} + \sum_{(w_i,r,w_j) \in A_t} \mathbf{f}(w_i, r, w_j) - \sum_{(w_i,r,w_j) \in A'} \mathbf{f}(w_i, r, w_j)$

vrati $\mathbf{w}$

Algoritam 2-2 Pseudokod algoritma perceptron

2.2.2.3 Algoritam za parsanje

Algoritmi za relacijski uvjetovano ovisnosno parsanje temeljeno na grafovima izvode se izravnim povezivanjem modela poznatoga iz teorije grafova koji se naziva *najveće prostiruće stablo* ili *najveće razapinjuće stablo* (en. *maximum spanning tree*, MST) i modela ovisnosnoga stabla te primjenom algoritama za pronalaženje najvećih prostirućih stabala (usp. Kübler i dr. 2009:43). Slijedi formalizacija toga postupka.

Neka je zadan usmjereni graf $DG = (V, A)$ i neka postoji neka funkcija za vrjednovanje težine veza, po uzoru na ranije definiranu funkciju $\text{score} : DG \longrightarrow \mathbb{R}$, koja pojedinim vezama iz A dodjeljuje realne brojeve koji predstavljaju njihove težine. Neka je težina nekoga podgrafa od DG jednaka zbroju težina svih veza koje ga sačinjavaju.

Najveće prostiruće stablo zadanoga grafa DG je onaj podgraf $DG' = (V', A')$ čija je težina s obzirom na funkciju za vrjednovanje težina veza najveća i koji pritom zadovoljava i sljedeća svojstva:

1. DG' sadrži sve čvorove iz grafa DG, odnosno $V' = V$, i
2. DG' je usmjereno stablo.

Sva stabla zadanoga grafa koja zadovoljavaju navedena svojstva, ali ne postižu najveću vrijednost težinske funkcije nazivaju se *prostirućim stablima* ili *razapinjućim stablima* (en. *spanning tree*).

Ovako zadani usmjereni graf DG zapravo se naziva usmjerenim multi-grafom ili multi-digrafom (en. *multi-digraph*, *quiver*) budući da podržava – s obzirom na činjenicu da su veze u njemu određene i ovisnosnom relacijom, a ne samo čvorovima koje povezuju – mogućnost višestrukoga povezivanja istih čvorova različitim ovisnosnim relacijama. Za ulaznu rečenicu $s = (w_0, \dots, w_n)$, skup ovisnosnih relacija $R = (r_1, \dots, r_m)$ i jezični model λ može se stoga definirati multi-digraf $DG_s = (V_s, A_s)$ takav da za njega vrijedi $V_s = \{w_0, \dots, w_n\}$ i $A_s = \{(w_i, r, w_j) : \forall w_i, w_j \in V_s, r \in R, j \neq 0\}$, odnosno da sadržava sve riječi ulazne rečenice i sve moguće ovisnosne veze među njima i s obzirom na skup svih ovisnosnih relacija, uz nemogućnost postavljanja korijenskoga čvora kao dependenta u nekoj od relacija.

Iz definicije ovisnosnoga stabla i prostirućega stabla (usp. Kübler i dr. 2009:44) slijedi da je za neku rečenicu s skup svih ovisnosnih stabala G_s jednak skupu svih prostirućih stabala

multi-digrafa DG_s. Stoga nadalje slijedi da je problem pronalaženja ovisnosnoga stabla koje prema jezičnome modelu λ najbolje opisuje ulaznu rečenicu, odnosno problem ovisnosnoga parsanja $h(s, \Gamma, \lambda)$ u paradigmi teorije grafova uz relacijsku uvjetovanost jednak problemu pronalaženja najvećega prostirućeg stabla multi-digrafa DG_s kojemu su za težine pojedinih veza dodijeljeni parametri $\lambda_{(w_i,r,w_j)}$ iz jezičnoga modela[79].

Uobičajeni algoritmi za pronalaženje najmanjega (pa implicitno i najvećega)[80] prostirućeg stabla – primjerice, algoritam Chu-Liu-Edmonds (Chu i Liu 1965, Edmonds 1967) asimptotske vremenske složenosti $O(n^3)$ i njegovo poboljšanje za primjenu nad potpunim grafovima, Tarjanov algoritam (Tarjan 1977), asimptotske vremenske složenosti $O(n^2)$ – odnose se na grafove u kojima su veze među čvorovima neoznačene. Stoga je potrebno prije njihove primjene problem relacijski uvjetovanoga ovisnosnog parsanja temeljenoga na grafovima, kako je sada definiran, reducirati izostavljanjem ovisnosnih relacija iz definicije ovisnosnoga grafa i njihovim naknadnim dodavanjem po primjeni algoritma za nalaženje najvećega prostirućega stabla.

Neka je stoga – u skladu s ranijom definicijom za DG_s – sada zadan usmjereni graf $DG'_s = (V'_s, A'_s)$ takav da je $V'_s = V_s$ i $A'_s = \{(w_i, w_j): \forall w_i, w_j \in V'_s, j \neq 0\}$. Tako definiran usmjereni graf je također potpun kao i DG_s, no više nije potencijalni multigraf budući da je svaki par čvorova povezan samo jednom vezom.

Skup pripadajućih parametara za veze iz jezičnoga modela sada se također nanovo definira uklanjanjem ovisnosne relacije:

$$\lambda_{(w_i, w_j)} = \max_r \lambda_{(w_i,r,w_j)}$$

Dakle, pri izgradnji jezičnoga modela neoznačenoga povezivanja riječi odabire se za svaku neoznačenu vezu samo najveća težina iz ranijega jezičnoga modela, a pritom se pamti ovisnosna relacija, odnosno sintaktička funkcija koja je bila dodijeljena toj vezi kako bi se

[79] Navedeno vrijedi samo za parsanje u kojemu se ne isključuje mogućnost neprojektivnih struktura. Ukoliko se uvede ograničenje u vidu projektivnosti parsnih stabala, algoritmi za pronalaženje MST-ova se ne mogu izravno primijeniti.

[80] Pronalaženje najmanjega prostirućega stabla tipičan je problem u računalnoj znanosti s gledišta pronalaženja najlakšega puta koji obilazi sve točke neke složene mreže koja se može predstaviti u obliku grafa. Zato se u literaturi u pravilu navode algoritmi za pronalaženje najmanjega prostirućega stabla; pronalaženje najvećega se stoga postiže množenjem svih težina pojedinih veza brojem -1, odnosno inverzijom problema.

rješenje problema neoznačenoga pretraživanja naknadnim mapiranjem sintaktičkih funkcija pretvorilo u rješenje problema ovisnosnoga parsanja.

Dakle, može se reći da za najveće prostiruće stablo $G = (V, A)$ multi-digrafa DG_s i najveće prostiruće stablo $G' = (V', A')$ digrafa DG'_s vrijedi:

$$\sum_{(w_i,r,w_j)\in A} \lambda_{(w_i,r,w_j)} = \sum_{(w_i,\, w_j)\in A'} \lambda_{(w_i,w_j)}$$

$$(w_i, r, w_j) \in A \Leftrightarrow (w_i,\ w_j) \in A' \wedge r = \arg\max_r \lambda_{(w_i,r,w_j)}$$

Iz jednakovrijednosti ovih jezičnih modela, s obzirom na njihovu poveznicu preko ovisnosne relacije za koju je pronađen parametar s najvećom težinskom vrijednošću, formalno je pokazano kako se problem ovisnosnoga parsanja svodi na problem pronalaženja najvećega prostirućega stabla usmjerenoga grafa.

Uz ranije navedeni algoritam Chu-Liu-Edmonds i Tarjanov algoritam, za rješavanje ovako postavljenoga problema može se, primjerice, upotrijebiti[81] i Kruskalov algoritam ($O(A \log A)$, Kruskal 1956) i Primov algoritam ($O(A + V \log V)$, Prim 1957)[82]. Zbog jednostavnosti, ali i ilustrativnosti s obzirom na preostale algoritme, ovdje se kratko prikazuje samo algoritam Chu-Liu-Edmonds.

Pseudokod algoritma preuzet je iz (Kübler i dr. 2009:47) i prikazan kao algoritam 2-4. Za njega se kaže da pripada skupini *pohlepnih* (en. *greedy*) i *rekurzivnih* (en. *recursive*) algoritama budući da izravno odabire veze i sadrži mogućnost rekurzivnoga poziva na obavljenoj transformaciji ulaznoga grafa s obzirom na cikluse koje može sadržavati. Algoritam za svaki čvor pronalazi vezu s najvećom težinom i iz grafa izbacuje sve ostale veze. Potom vrši provjeru acikličnosti. Ukoliko je rezultirajući graf acikličan, on predstavlja najveće prostiruće stablo i algoritam završava izvršavanje. U protivnome, ako je u tome grafu pronađen ciklus, on se sažima u jedan čvor pomoćnom procedurom i težine veza se u skladu s time preračunavaju. Algoritam se potom rekurzivno poziva na novonastalome grafu u kojemu je jedan ciklus zamijenjen jednim novim čvorom uz preračunavanje težina te se korištenjem pomoćnih varijabli izgrađuje najveće prostiruće stablo.

[81] Također (usp. Cormen i dr. 2009:631-638) za pregled navedenih algoritama.
[82] Valja primijetiti da se Primov algoritam odnosi na neusmjerene grafove pa njegova primjena zahtijeva određene preinake u ponovnome definiranju ovisnosnoga stabla.

Chu-Liu-Edmonds(G, λ)

neka je zadan graf $G = (V, A)$ i parametri $\lambda_{(w_i,w_j)} \in \lambda$

$A' = \left\{(w_i, w_j) : \ w_j \in V, w_i = \arg\max_{w_i} \lambda_{(w_i,w_j)}\right\}$

$G' = (V, A')$

ako je G' acikličan graf, vrati G' i završi izvođenje

u protivnom

pronađi ciklus A_c u grafu G'

$\langle G_c, w_c, ep\rangle = \text{contract}(G', A_c, \lambda)$

$G = (V, A) = \text{Chu-Liu-Edmonds}(G_c, \lambda)$

za $(w_i, w_c) \in A, ep(w_i, w_c) = w_j$, pronađi $(w_k, w_j) \in A_c$ za neki w_k

pronađi sve veze $(w_c, w_l) \in A$

$A = A \cup \{(ep(w_c, w_l), w_l), \forall(w_c, w_l) \in A\} \cup A_c \cup \{(w_i, w_j)\} - \{(w_k, w_j)\}$

$V = V$

vrati G

contract(G, C, λ)

neka je G_c podgraf od G bez čvorova iz C

u G_c dodaj čvor w_c tako da taj čvor predstavlja ciklus C

za $w_j \in V - C$ takav da $\exists(w_i, w_j) \in A, w_i \in C$

dodaj (w_c, w_j) u G_c tako da vrijedi

$ep(w_c, w_j) = \arg\max_{w_i \in C} \lambda_{(w_i,w_j)}$

$w_i = ep(w_c, w_j)$

$\lambda_{(w_c,w_j)} = \lambda_{(w_i,w_j)}$

za $w_i \in V - C$ takav da $\exists(w_i, w_j) \in A, w_j \in C$

dodaj (w_i, w_c) u G_c tako da vrijedi

$ep(w_i, w_c) = \arg\max_{w_j \in C} \left[\lambda_{(w_i,w_j)} - \lambda_{(a(w_j),w_j)}\right]$

$w_j = ep(w_i, w_c)$

$\lambda_{(w_i,w_c)} = \lambda_{(w_i,w_j)} - \lambda_{(a(w_j),w_j)} + \text{score}(C)$

uz $a(w) =$ prethodnik od w u C i $\text{score}(C) = \sum_{w \in C} \lambda_{(a(w),w)}$

vrati trojku $\langle G_c, w_c, ep\rangle$

Algoritam 2-3 Pseudokod algoritma Chu-Liu-Edmonds

Projektivna ovisnosna stabla, odnosno svaki sintaktički formalizam koji se na njima temelji po opisnoj je moći jednak beskontekstnoj gramatici (usp. Kübler i dr. 2009:18). Stoga se za projektivno ovisnosno parsanje u okviru relacijske uvjetovanosti i teorije grafova, odnosno traženja najvećega prostirućeg stabla kao optimalnoga ovisnosnog stabla, u pravilu koriste prilagođene inačice algoritama za parsanje beskontekstnom gramatikom. Literatura navodi (usp. Kübler i dr. 2009:49-54) sljedeće algoritme:

1. jednu inačicu algoritma CYK, asimptotske vremenske složenosti $O(n^5)$ i prostorne složenosti $O(n^3)$ i
2. Eisnerov algoritam (Eisner 1996a, 1996b), asimptotske vremenske složenosti $O(n^3)$ i prostorne složenosti $O(n^2)$.

Vrijedi primijetiti kako su navedeni algoritmi za projektivno ovisnosno parsanje u teorijskome okviru teorije grafova, donekle paradoksalno, zapravo složeniji[83] od algoritama za neprojektivno ovisnosno parsanje, iako je problem neprojektivnoga parsanja po definiciji složeniji.

Ovaj paradoks proizlazi iz činjenice da se problem ovisnosnoga parsanja svodi na problem nalaženja najvećega prostirućeg stabla, pa se time zanemaruje njegova inherentna težina s gledišta samoga parsanja. Može se stoga pretpostaviti da će algoritmi za ovisnosno parsanje bez ograničenja na projektivnost, iako nezahtjevniji od algoritama za projektivno parsanje, rezultirati s gledišta vrjednovanja točnosti lošijim parsanjem. Ta se pretpostavka razmatra dalje u ovome istraživanju, no tek nakon predstavljanja pristupa ovisnosnomu parsanju temeljenom na prijelazima. Sustav MSTParser (McDonald i dr. 2005a, 2005b, McDonald i Pereira 2006, McDonald i dr. 2006), kao primjer izvedbe ovdje prikazanoga pristupa, također se detaljnije prikazuje naknadno.

2.2.3 Ovisnosno parsanje temeljeno na prijelazima

Kod ovisnosnoga parsanja temeljenoga na prijelazima zamišlja se model ovisnosnoga parsera kao apstraktnoga stroja, odnosno formalnoga automata koji iz tablice prijelaza – koja predstavlja jezični model izgrađen iz banke ovisnosnih stabala – u diskretnome vremenu gradi ovisnosno stablo čitajući redom riječi ulazne rečenice. U tome se teorijskom okviru svaki

[83] Primjerice, Eisnerov je algoritam kubne vremenske složenosti $O(n^3)$, a Tarjanov algoritam, kao inačica Chu-Liu-Edmonds algoritma, kvadratne vremenske složenosti $O(n^2)$.

idući korak takvoga apstraktnog stroja zasniva na razmatranju pročitane riječi i prethodno izvršenih operacija, uz korištenje pohlepnoga (en. *greedy*) deterministčkog algoritma. Njegov se jezični model, odnosno tablica prijelaza zasniva na modelu jednoga prijelaza koji je načelno zamišljen kao funkcija koja – ovisno o trenutnome stanju apstraktnoga stroja, riječi na ulazu i mogućim drugim opažanjima unutarnjega stanja ili dodatnih značajki toga stanja ili ulaznih podataka – vrši izmjenu stanja apstraktnoga stroja i pokreće operacije nad mogućim drugim njegovim strukturama, kao kod formalnih automata.

Budući da se radi o modelu parsanja teksta, odnosno rješavanju problema $h(s, \Gamma, \lambda)$ iz ranije danoga modela ovisnosnoga parsanja, skup ograničenja Γ ostaje – kao kod ovisnosnoga parsanja temeljenoga na grafovima – takav da ograničava izlaznu strukturu parsera na skup svih valjanih ovisnosnih stabala koja se mogu izgraditi nad danom ulaznom rečenicom s pomoću skupa sintaktičkih funkcija R iz danoga sintaktičkog formalizma. Ovisnosno parsanje temeljeno na prijelazima prikazuje se ovdje prema (Kübler i dr. 2009:21-39), a usko je vezano uz izvedbu u sustavu MaltParser (Nivre 2003, 2006, 2007, 2008, Nivre i Nilsson 2005, Nivre i dr. 2006, Nivre i dr. 2007b), koji se prikazuje kasnije u tekstu.

2.2.3.1 Definicije

Prijelaznički sustav ili *prijelaznik* (en. *transition system*) je apstraktni stroj određen s pomoću skupa konfiguracija (ili stanja) i funkcije koja, najčešće ovisno o nekome ulazu, određuje njegovo prelaženje iz jedne u drugu konfiguraciju (ili iz jednoga stanja u drugo). Ova je definicija dovoljno uopćena da pokriva širok raspon mogućih apstraktnih strojeva, od deterministčkoga konačnog automata – kod kojega su skupovi ulaza i stanja određeni kao nedjeljivi (atomarni) elementi – pa do modela ovisnosnoga parsera, kod kojega ti skupovi sadrže složene elemente kojima se predstavlja postupak izgradnje ovisnosnoga stabla ulazne rečenice. Prema (Kübler i dr. 2009:21), osnovna je ideja u ovome pristupu modeliranju ovisnosnoga parsanja "slijedom valjanih prijelaza od početne konfiguracije, čitajući jednu po jednu riječi ulazne rečenice, dosegnuti neku završnu konfiguraciju i tim slijedom pritom ocrtati ovisnosno stablo koje predstavlja valjanu sintaktičku analizu ulazne rečenice". Model ovisnosnoga parsera temeljenoga na prijelazima koji se prikazuje ovdje zasnovan je na onome prikazanom u (Kudo i Matsumoto 2000, Yamada i Matsumoto 2003) i predstavlja jednostavni – a ujedno i najčešće korišteni – prijelaznički sustav, zasnovan na stogu i izvedbi jednoga oblika parsnoga algoritma s pomakom i redukcijom (en. *shift-reduce parser*).

Neka je zadan skup sintaktičkih funkcija R i ulazna rečenica $s = (w_0, \dots, w_n)$. Jedna *konfiguracija* (en. *configuration*) prijelazničkoga sustava definira se kao uređena trojka $c = (\sigma, \beta, A)$, gdje je σ stog (en. *stack*) riječi $w_i \in s$, β ulazna vrpca (en. *input tape*, *buffer*) s riječima $w_i \in s$, a A je skup ovisnosnih relacija $(w_i, r, w_j) \in V_s \times R \times V_s$ s obzirom na raniju definiciju skupa svih ovisnosnih stabala G_s koja je moguće izraditi nad rečenicom s.

Svaka konfiguracija predstavlja djelomično parsanje ulazne rečenice, i to na način da se na stog stavljaju djelomično obrađene riječi, dok se na ulaznoj vrpci nalaze još neobrađene riječi, a skup ovisnosnih relacija sadrži djelomično izgrađeno ovisnosno stablo za ulaznu rečenicu. Jedna konfiguracija c_j za danu ulaznu rečenicu s u diskretnome vremenu j prikazuje se kao $c_j(s) = \left([w_0, \dots, w_j]_\sigma, [w_{j+1}, \dots, w_n]_\beta, A\right)$, uz eventualno eksplicitno navođenje ovisnosnih relacija u skupu A. Često se koristi skraćena notacija $c_j(s) = (\sigma|w_j, w_{j+1}|\beta, A)$, kojom se prikazuje riječ na vrhu stoga i iduća riječ ulazne vrpce. Posebno se definira:

1. početna konfiguracija sustava, $c_0(s) = ([w_0]_\sigma, [w_1, \dots, w_n]_\beta, \emptyset)$ i
2. završna konfiguracija sustava, $c_n(s) = (\sigma, [\ \ ]_\beta, A), \forall\sigma, \forall A$.

Početna konfiguracija sustava ima nultu, odnosno korijensku meta-riječ na stogu i sve riječi rečenice na ulaznoj vrpci te je skup uspostavljenih ovisnosnih relacija prazan. Završnom se smatra svaka konfiguracija u kojoj je ulazna rečenica pročitana, neovisno o sadržaju stoga i izgrađenome skupu ovisnosnih relacija. Takva definicija donekle je podudarna s onom stogovnoga automata koji prihvaća ulazni niz dosezanjem završnoga stanja, za razliku od onoga koji prihvaća pražnjenjem stoga.

Prijelaznički sustav, kao i svaki drugi apstraktni stroj, predstavlja izračunski model koji izvršava operacije izmjenom konfiguracija pri čitanju ulaznih podataka. Izmjena konfiguracija definira se parcijalnom funkcijom prijelaza koja se preslikava iz jedne konfiguracije u drugu, no ne mora nužno biti definirana za svaku konfiguraciju. Svaki prijelaz predstavlja neku osnovnu radnju kod ovisnosnoga parsanja: dodavanje veze u ovisnosno stablo, rukovanje stogom i rukovanje ulaznom vrpcom. Prema (Kübler i dr. 2009:23), u ovome se modelu parsanja definiraju tri vrste prijelaza.

1. $\text{LeftArc}(r) \dots (\sigma|w_i, w_j|\beta, A) \Rightarrow (\sigma, w_j|\beta, A \cup \{(w_j, r, w_i)\}), i \neq 0$

Za svaku ovisnosnu relaciju $r \in R$ prijelazi iz razreda LeftArc uzimaju riječ w_i s vrha stoga i prvu riječ w_j s ulazne vrpce, uspostave ovisnosnu relaciju između riječi w_j s ulazne vrpce kao glave i riječi w_i sa stoga kao dependenta i obrišu riječ w_i sa stoga. Zahtijevaju neprazni stog i ulaznu vrpcu te riječ na vrhu stoga ne smije biti jednaka w_0, čime se osigurava svojstvo postojanja korijena ovisnosnoga stabla.

2. $\text{RightArc}(r) \ldots (\sigma|w_i, w_j|\beta, A) \Rightarrow (\sigma, w_i|\beta, A \cup \{(w_i, r, w_j)\})$

   Za svaku ovisnosnu relaciju $r \in R$ prijelazi iz razreda RightArc uzimaju riječ w_i s vrha stoga i prvu riječ w_j s ulazne vrpce, uspostave ovisnosnu relaciju između riječi w_i sa stoga kao glave i riječi w_j s ulazne vrpce kao dependenta i obrišu riječ w_i sa stoga te je stave na mjesto riječi w_j u ulaznoj vrpci kako bi se w_j mogla povezati s glavom w_i koja se nalazi lijevo od nje. Stog i ulazna vrpca moraju biti neprazni. Prijelazi iz razreda LeftArc i RightArc pronalaze ovisnosnu relaciju između glave i dependenta i zamijene je glavom relacije te predstavljaju operaciju redukcije.

3. $\text{Shift} \ldots (\sigma, w_i|\beta, A) \Rightarrow (\sigma|w_i, \beta, A)$

   Prijelazi iz razreda Shift uzimaju riječ w_i s ulazne vrpce i postavljaju je na vrh stoga. Ne dodaju novu ovisnosnu relaciju u skup ovisnosnih relacija A i zahtijevaju samo nepraznu ulaznu vrpcu.

Neka je zadan skup svih dopuštenih prijelaza $\mathcal{T}$ za neki prijelaznički sustav. *Slijed prijelaza* (en. *transition sequence*) ulazne rečenice $s = (w_0, \ldots, w_n)$ definira se kao slijed konfiguracija $c_{0,m} = (c_0, \ldots, c_m)$ takav da vrijedi

1. c_0 je početna konfiguracija $c_0(s)$ rečenice s,
2. c_m je završna konfiguracija rečenice s i
3. $\forall i, 1 \leq i \leq m, \exists t \in \mathcal{T}, c_i = t(c_{i-1})$.

Slijed prijelaza nekoga prijelazničkog sustava započinje od korijenskoga meta-čvora, odnosno početne konfiguracije i u diskretnome vremenu doseže završnu konfiguraciju, i to izmjenom konfiguracija u skladu sa skupom svih dopuštenih prijelaza $\mathcal{T}$ za dani sustav. Ovisnosno stablo izrađeno ovim slijedom prijelaza određeno je završnom konfiguracijom, prema definiciji konfiguracije, tako da vrijedi $G = G_{c_m} = (V_s, A_{c_m})$.

Svaki prijelaz u slijedu prijelaza definira ovisnosni graf koji zadovoljava sva svojstva tražena od ovisnosnoga stabla, osim svojstva povezanosti. Međutim, s obzirom na to da svaki

od njih zadovoljava svojstvo necikličnosti, dovoljno je naknadnom obradbom sve nepovezane čvorove povezati s korijenskim čvorom i osigurati zadovoljavanje toga svojstva. Kako pokazuje (Nivre 2008), model prijelazničkoga sustava omeđuje upravo skup projektivnih ovisnosnih stabala $G_s^P \subset G_s$. Dalje u tekstu stoga se prikazuje pristup jezičnomu modeliranju u ovome teorijskom okviru, model parsanja tim jezičnim modelom i prilagodba jezičnoga modela i algoritma za parsanje s obzirom na rukovanje neprojektivnim strukturama.

2.2.3.2 Algoritam za parsanje

Problem ovisnosnoga parsanja prijelazničkim sustavom svodi se na pretraživanje skupa svih dopuštenih prijelaza u potrazi za onima najboljima, odnosno onima koji će na kraju slijeda prijelaza proizvesti valjano ovisnosno stablo za ulaznu rečenicu.

Neka je dana funkcija koja za trenutnu konfiguraciju nalazi uvijek optimalni prijelaz u novu konfiguraciju, $t = \text{oracle}(c)$. Ta se funkcija naziva *tumač* (ili prorok, en. *oracle*) i za trenutnu konfiguraciju uvijek vraća samo točan prijelaz, odnosno onaj koji po dosezanju završne konfiguracije jamči valjano parsanje ulazne rečenice. Parsanje prijelazničkim sustavom uz uporabu funkcije tumača kao savršenoga jezičnog modela prikazano je kao algoritam 2-4. Kako navodi (Kübler i dr. 2009:25, Nivre 2006:77), algoritam za parsanje s pomoću funkcije tumač je jednostavan: počevši od početne pa sve do završne konfiguracije prijelazničkoga sustava, tumač uvijek pronalazi najbolji prijelaz, pa je stablo sadržano u završnoj konfiguraciji optimalno parsanje ulazne rečenice.

h$(s, \Gamma, \text{oracle})$
    neka je $c = c_0$ početna konfiguracija
    dok konfiguracija c nije završna konfiguracija
        neka je idući prijelaz $t = \text{oracle}(c)$
        neka je iduća konfiguracija $c = t(c)$
    vrati G_c

Algoritam 2-4 Parsanje prijelazničkim sustavom uz uporabu tumača

Rješavanje optimizacijskoga problema $t = \text{oracle}(c)$ računalno je težak problem i njegovu se rješavanju pristupa približnim metodama, uglavnom zasnovanima na postupcima strojnoga učenja (en. *machine learning*). U tome se smislu uvodi razlika između funkcije

tumač i funkcije vodič (en. *guide*): funkcija tumač uvijek pronalazi optimalno rješenje i kao takva je hipotetske prirode, dok funkcija vodič nastoji aproksimirati rješenje u skladu s modelom usvojenim nad prethodno obrađenim podatcima, odnosno podatcima prikupljenima iz banke ovisnosnih stabala. Ako se hipotetski pretpostavi da je funkcija tumač konstantne vremenske složenosti $O(1)$, vremenska složenost algoritma 2-4 je linearna s obzirom na ulazne podatke, $O(n)$. Svaka približna funkcija vodič bit će očekivane vremenske složenosti veće od one pretpostavljene za funkciju tumač, pa će nužno predstavljati ključni činitelj u ukupnoj vremenskoj složenosti ovisnosnoga parsanja temeljenoga na prijelazima. Funkcija vodič može se izvesti (usp. Kübler i dr. 2009:26) na mnoge načine, od uporabe formalne gramatike pa do niza heurističkih metoda. Pokazalo se, međutim, između ostaloga i na natjecanjima CoNLL 2006 i 2007, kako su najuspješniji pristupi izradi te funkcije temeljeni na podatcima i na strojnome učenju. Još konkretnije, funkcija tumač najčešće se aproksimira funkcijom vodič predstavljenim modelom *klasifikatora* (en. *classifier*) treniranoga na ovisnosnoj banci stabala. Stoga se kaže da su klasifikatori ključna komponenta parsanja temeljenoga na prijelazima te se posljedično čitava paradigma naziva *parsanje temeljeno na klasifikatorima* (en. *classifier-based parsing*).

Neka se jezični model λ u parsanju prijelazničkim sustavom temeljenom na klasifikatoru sastoji od elemenata $\lambda_c \in \lambda$, gdje svaki element predstavlja predviđanje idućega prijelaza za danu konfiguraciju c prema jezičnome modelu λ, odnosno što je moguće bolju aproksimaciju funkcije tumač $\mathrm{oracle}(c)$. Jezični se model λ stoga može zamisliti (usp. Kübler i dr. 2009:27) kao "velika tablica predviđenih prijelaza $\lambda_c \in \lambda$ za svaku moguću konfiguraciju c". Pritom se zadržava forma algoritma 2-4, osim što funkciju tumač mijenja jezični model u svojstvu funkcije vodič. Prilagođena inačica dana je kao algoritam 2-5.

h(s, Γ, λ)
    neka je $c = c_0$ početna konfiguracija
    dok konfiguracija c nije završna konfiguracija
        neka je idući prijelaz $t = \lambda_c$
        neka je iduća konfiguracija $c = t(c)$
    vrati G_c

Algoritam 2-5 Zamjena funkcije tumač jezičnim modelom

Ovaj algoritam predstavlja prijelaznički sustav za ovisnosno parsanje sa sintaktičkim modelom temeljenim na klasifikatoru. Preostaje definirati taj jezični model na neki opisno prikladan i istovremeno učinkovit način kako bi se ovisnosno parsanje prijelazničkim sustavom zadržalo u okvirima linearne vremenske složenosti.

Ranije je definirana funkcija, odnosno vektor značajki $\mathbf{f} : X \longrightarrow Y, \mathbf{f}(x) = y, y = (y_1, \ldots, y_m) \in Y$ koji nekoj kategoriji $x \in X$ dodjeljuje niz brojčanih ili simboličkih značajki, najčešće opisanih brojčanim vrijednostima $(y_1, \ldots, y_m), y_i \in \mathbb{R}, Y = \mathbb{R}^m$ koje tu kategoriju pobliže označavaju. Kod klasifikatorskoga prijelazničkog parsanja vektor značajki definira se tako da je $X = C$, gdje je C skup svih mogućih konfiguracija za bilo koje ulazne podatke. Dakle, funkcija $\mathbf{f} : C \longrightarrow Y$ svakoj konfiguraciji $c \in C$ sustava dodijeli m-dimenzionalni vektor $\mathbf{f}(c)$ koji ju opisuje.

Na temelju ove definicije vektora značajki definira se klasifikator kao funkcija $g : Y \longrightarrow \mathcal{T}$, gdje je $\mathcal{T}$ skup svih mogućih prijelaza za koje vrijedi $g(\mathbf{f}(c)) = \text{oracle}(c), \forall c \in C$. Prema (Kübler i dr. 2009:27), "drugim riječima, iz danoga uzorka za treniranje modela iz ovisnosne banke stabala, nastoji se stvoriti klasifikator koji predviđa prijelaz koji bi pronašla hipotetska funkcija $\text{oracle}(c)$ za svaku moguću konfiguraciju c i sa zadanim skupom značajki $\mathbf{f}(c)$".

Definicija klasifikatora povlači tri pitanja (usp. Kübler i dr. 2009:27) na koja je potrebno odgovoriti s gledišta ovisnosnoga parsanja temeljenoga na podatcima kako bi se postavljeni teorijski okvir klasifikatorskoga prijelazničkog parsanja zaista ostvario u nekoj konkretnoj izvedbi. Treba, dakle, definirati sljedeće:

1. opis konfiguracije prijelazničkoga sustava vektorom značajki,
2. algoritam za preoblikovanje podataka iz banke ovisnosnih stabala u oblik koji je smislen s gledišta konfiguracija i njihovih značajki i
3. pristup učenju klasifikatora.

Definirati ove elemente – model koji povezuje konfiguracije sa značajkama i metodu povezivanja podataka u banci stabala s modelom klasifikatora – znači pronaći postupak izrade jezičnoga modela za klasifikatorsko prijelazničko parsanje za izravnu (i jednostavnu) primjenu u algoritmu 2-5 za parsanje funkcijom tumač.

2.2.3.3 Algoritam za izradu jezičnoga modela

Prikazati neku konfiguraciju $c \in C$ pripadajućim vektorom značajki $\mathbf{f}(c)$ potrebno je kako bi se podatci o konfiguraciji mogli iskoristiti u nekome modelu klasifikatora $g(\mathbf{f}(c))$, a da se pritom u potpunosti zadrži opis same konfiguracije. Vektor značajki $\mathbf{f}(c) = y = (y_1, \dots, y_m)$ pritom se može opisati i preko niza jednostavnih značajki $\mathbf{f}_i(c) = y_i, 1 \leq i \leq m$, a svaka od jednostavnih značajki može biti određena nekim brojem proizvoljnih atributa koji opisuju svojstva dane konfiguracije. Značajkama se najčešće opisuju svojstva pojedinih riječi u rečenici, odnosno čvorova u ovisnosnome stablu. Stoga se značajke definiraju u obliku kompozicije dviju funkcija: funkcije za identifikaciju riječi u konfiguraciji kao riječi u rečenici i ovisnosnome stablu i funkcije za dohvat neke jednostavne značajke ili atributa te riječi. Slijedi formalizacija.

Neka je dana ulazna rečenica $s = (w_0, \dots, w_n)$ i skup svih mogućih ovisnosnih stabala G_s s pripadajućim skupom čvorova $V_s = \{w_0, \dots, w_n\}$ koje je moguće izgraditi nad njome. Kompozicija funkcija

$$(v \circ a)(c) : C \longrightarrow Y, \qquad a : C \longrightarrow V_s, \qquad v : V_s \longrightarrow Y$$

naziva se – odnosno, njezina se povratna vrijednost naziva – *konfiguracijskom značajkom riječi* (en. *configurational word feature*), a sastoji se od *adresne funkcije* (en. *address function*) koja povezuje konfiguraciju s riječju koja joj pripada i *atributske funkcije* (en. *attribute function*) koja povezuje tu riječ s nekom njezinom značajkom, najčešće oznakom s neke razine jezičnoga opisa. Obje funkcije koje sačinjavaju funkciju konfiguracijske značajke riječi također se mogu graditi kompozicijom jednostavnijih funkcija. Primjerice, može se zamisliti adresna funkcija koja (usp. Kübler i dr. 2009:29-30) pronalazi "k-tu riječ od vrha stoga nadolje", "povezuje riječ s njezinom glavom ili najdesnijim dependentom" ili "najdesnijega dependenta najlijevijega susjeda glave riječi na vrhu stoga". S druge strane, atributske funkcije pronalaze neke metapodatke promatranih riječi, poput lema, morfosintaktičkih oznaka ili sintaktičkih funkcija. Obje funkcije, pa tako i njihova kompozicija, u pravilu su parcijalne budući da ne moraju nužno biti definirane za svaku konfiguraciju, odnosno za svaku riječ ulazne rečenice.

Model konfiguracijskih značajki definira se obično u obliku tablice koja povezuje adresne i atributske funkcije. Primjer je dan u tablici 2-4.

Tablica 2-4 Model konfiguracijskih značajki prijelazničkoga parsera

	atributska funkcija				
adresna funkcija	form	lemma	postag	feats	deprel
stack[0]	+	+	+	+	
stack[1]			+		
ldep(stack[0])					+
buffer[1]	+		+		
rdep(buffer[0])					+

Atributske funkcije iz tablice 2-4 odgovaraju stupcima iz zapisa banke ovisnosnih stabala u formatu CoNLL. Dakle, svaka od atributskih funkcija za danu riječ vraća element iz stupca predstavljenoga imenom funkcije koji predstavlja oznaku dane riječi na nekoj od dostupnih razina lingvističkoga opisa. Adresne funkcije stack[i] i buffer[i] služe za dohvat i-te riječi sa stoga i ulazne vrpce, a funkcije $\mathrm{ldep}(w_i)$ i $\mathrm{rdep}(w_i)$ dohvaćaju krajnje lijevi i krajnje desni dependent dane riječi. Svaki element tablice predstavlja značajku kojoj se može pristupiti kompozicijom funkcija u stupcu i retku toga elementa. Primjerice, u danome se modelu značajki može dohvatiti sintaktička funkcija krajnje lijevoga dependenta riječi na vrhu stoga ili morfosintaktička oznaka druge riječi na ulaznoj vrpci. Tablica 2-4 predstavlja model značajki utoliko što se svaka konfiguracija sada opisuje upravo putem toga modela – točnije, promatranjem svojstava i međuovisnosti pojedinih riječi na stogu i ulaznoj vrpci u nekom diskretnom vremenu – te se taj opis koristi za treniranje jezičnoga modela.

Neka je za svaku konfiguraciju $c \in C$, s obzirom na ulaznu rečenicu, dostupan vektor značajki $\mathbf{f}(c)$ prema nekome unaprijed zadanom modelu konfiguracijskih značajki. Potrebno je svaku konfiguraciju povezati s točnim prijelazom oracle(c), odnosno izvršiti razredbu konfiguracija prema razredima prijelaza definiranih skupom $\mathcal{T}$ svih mogućih prijelaza. Taj problem naziva se *problemom klasifikacije* (en. *classification problem*) ili *razredbe* budući da se konfiguracije svrstavaju u razrede prema zadanoj razredbi. U skladu sa zadanim okvirima parsanja temeljenoga na podatcima, utoliko se zahtijeva da podatci za treniranje, odnosno podskup ovisnosne banke stabala namijenjen izradi jezičnoga modela prijelazničkoga parsera,

budu pripremljeni u obliku uređenih parova vektora značajki i pripadajućih točnih prijelaza u skladu s funkcijom tumač, $(\mathbf{f}(c), t), t = \text{oracle}(c)$.

Povezivanjem ranije formalne definicije banke ovisnosnih stabala kao skupa uređenih parova rečenica i pripadajućih ovisnosnih stabala s definicijom vektora značajki pojedinih konfiguracija, iz banke ovisnosnih stabala $T = \{(s_t, G_t)\}_{t=1}^{|T|}$ sada se može izraditi novi uzorak za treniranje, oblika:

$$T' = \{(f(c_d), t_d)\}_{d=1}^{|T'|}$$

Uzorak za treniranje T' oblikuje se izradom slijeda prijelaza $c_{0,m}^d = (c_0, \dots, c_m)$ za svaki uređeni par (s_d, G_d) iz banke stabala, tako da vrijedi $c_0 = c_0(s_d)$ i $G_t = (V_d, A_{c_m})$. Potom se za svaku nezavršnu konfiguraciju $c_i^d \in c_{0,m}^d$ u T' dodaje uređeni par $\left(f(c_i^d), t_i^d\right)$, i to tako da vrijedi $t_i^d(c_i^d) = c_{i+1}^d$. Dakle, za svaku se rečenicu izrađuje slijed prijelaza, pa se konfiguracije iz toga slijeda povezuju jediničnim prijelazima prema redoslijedu njihova pojavljivanja u diskretnome vremenu. Za svaki par (s_d, G_d), odnosno za svako ovisnosno stablo $G_d = (V_d, A_d)$ i svaki prijelaz iz konfiguracije u konfiguraciju, može se na sljedeći način odrediti[84] vrsta prijelaza koja ga je uzrokovala:

$$\text{oracle}(c = (\sigma, \beta, A)) = \begin{cases} \text{LeftArc}(r), \text{ako } (w_j, r, w_i) \in A_d, \sigma|w_i, w_j|\beta \\ \text{RightArc}(r), \text{ako } (w_i, r, w_j) \in A_d, \sigma|w_i, w_j|\beta \\ \text{i ako } \forall w, \forall r', (w_j, r', w) \in A_d, w_j|\beta \Rightarrow (w_j, r', w) \in A \\ \text{Shift}(r), \text{u protivnom} \end{cases}$$

Treniranje samoga klasifikatora je (usp. Kübler i dr. 2009:33) "dobro istražen problem u području strojnoga učenja", pa je za njegovo rješavanje "na raspolaganju veliki broj različitih algoritama". Uobičajeni pristup klasifikaciji u klasifikacijskome prijelazničkom parsanju jest uporaba *potpornih vektorskih strojeva* (en. *support vector machines*) s *polinomskim jezgrama* (en. *polynomial kernels*).

Na jezični model λ usvojen treniranjem klasifikatora, uz raniju izradu modela značajki i prilagodbu banke ovisnosnih stabala, primjenjuje se algoritam 2-5 i njime se – jednostavnim čitanjem pojedinih prijelaza iz jezičnoga modela i izmjenom konfiguracija prijelazničkoga

[84] Ova transformacija iz slijeda konfiguracija u slijed funkcija prijelaza moguća je samo za projektivna ovisnosna stabla (usp. Kübler i dr 2009:32).

sustava u diskretnome vremenu – rješava problem ovisnosnoga parsanja $G = h(s, \Gamma, \lambda)$ rečenice $s = (w_0, \ldots, w_n)$ u linearnome vremenu $O(n)$.

2.2.3.4 Neprojektivno parsanje

Različite inačice ovisnosnih parsera temeljenih na prijelazima – primjerice, s obzirom na izbor svojstava prijelazničkoga sustava, model algoritma za parsanje i odabir pristupa klasifikaciji – prikazane su u (Kübler i dr. 2009:34-37) i ovdje se ne raspravljaju dodatno. Međutim, s obzirom na raniju definiciju projektivnih i neprojektivnih ovisnosnih struktura, vrijedi još jednom napomenuti da je ovdje definirani model klasifikatorskoga prijelazničkog ovisnosnog parsera ograničen na projektivne strukture, pa vrijedi kratko razmotriti pristupe njegovoj prilagodbi za rukovanje neprojektivnim strukturama.

Jedan pristup omogućavanju uporabe neprojektivnih ovisnosnih stabala u postupcima izrade i korištenja modela u klasifikatorskom prijelazničkom parsanju – izveden, između ostaloga, i u sustavu MaltParser – naziva se *pseudo-projektivno parsanje* (en. *pseudo-projective parsing*, Nivre i Nilsson 2005). Taj se pristup svodi na postupak projektivizacije neprojektivnih ovisnosnih stabala u uzorku za treniranje, uz zapisivanje svih u tome postupku obavljenih preinaka ovisnosnih stabala u vidu dodatnih ovisnosnih relacija za potrebe kasnije rekonstrukcije. Potom se prethodno definirani projektivni parser trenira na takvome jezičnom uzorku i parsa ulazne rečenice uz ograničenje na projektivne strukture. Izlaz iz parsera se naposljetku pretvara, prema potrebi, iz projektivne u neprojektivnu strukturu, koristeći ranije pohranjene podatke za rekonstrukciju. Postupak se temelji na činjenici da je, prema definiciji ovisnosnoga stabla, svaku neprojektivnu ovisnosnu relaciju, odnosno svaku ovisnosnu relaciju (w_i, r, w_j) koja narušava postavljeno ograničenje na projektivne strukture moguće pretvoriti u projektivnu vezivanjem dependenta w_j te relacije uz najbližu posrednu ili neposrednu glavu glave relacije w_i za koju je nova relacija projektivna. Dakle, neprojektivna relacija (w_i, r, w_j) mijenja se projektivnom relacijom $(a(w_i), r', w_j)$, gdje je funkcija za dohvat projektivne glave definirana kao $a(w_i) = w_k, k \rightarrow^* i$.

Pokazano je da se postupkom deprojektivizacije "uspješno rekonstruira više od 90% neprojektivnih ovisnosnih relacija" (usp. Nivre i Nilsson 2005) uz zadržavanje složenosti parsanja u okvirima linearne vremenske složenosti, no uz gubitak točnosti i učinkovitosti kod izrade i kod korištenja jezičnoga modela (usp. Nivre 2008). Jedan pristup temeljen na dopuštanju samo ograničenih oblika neprojektivnosti uz očuvanje linearne vremenske

složenosti prikazan je u (Attardi 2006), a parsanje neograničenih neprojektivnih struktura uz kvadratnu vremensku složenost $O(n^2)$ prikazano je u (Covington 2001) i ugrađeno u sustav MaltParser prema (Nivre 2006b, Nivre 2007). Neki od tih pristupa neprojektivnom parsanju razmatraju se dalje u tekstu.

3 Neki pristupi ovisnosnom parsanju hrvatskih tekstova

U ovome se poglavlju prikazuju i raspravljaju rezultati eksperimenata s ovisnosnim parsanjem hrvatskih tekstova u ranije izloženome teorijskom okviru ovisnosnoga parsanja teksta parserima temeljenima na podatcima, odnosno temeljenima na grafovima i prijelazima. Detaljno se razmatraju različiti vanjski utjecaji na točnost i učinkovitost ovisnosnoga parsanja hrvatskih tekstova. Također se definira novi model ovisnosnoga parsera hrvatskih tekstova, temeljen na podatcima i uporabi dostupnih jezičnih resursa za hrvatski jezik. Budući da čitav teorijski okvir ovisnosnoga parsanja temeljenoga na podatcima počiva na referentnome označavanju iz kojega se potom treniraju sintaktički modeli za parsanje, odnosno na dostupnosti banke ovisnosnih stabala, pritom se kratko opisuje i najnovija inačica Hrvatske ovisnosne banke stabala, koja je dijelom razvijana i ciljano za potrebe ovoga istraživanja.

Najprije se, u poglavlju 3.1, opisuju postojeći pristupi parsanju hrvatskoga jezika. U trenutku provođenja ovoga istraživanja bio je dostupan mali broj dokumentiranih i vrjednovanih parsera hrvatskih tekstova, kao i inteligentnih računalnih sustava za obradbu hrvatskoga jezika na sintaktičkoj razini uopće. Ovdje se daje kratki osvrt na tri sustava: jedan je zasnovan na teorijskome okviru parsanja gramatikom, drugi predstavlja razdjelnik i plitki parser hrvatskih tekstova temeljen na pravilima i djelomično na ovisnosnoj sintaksi, a treći vrši sintaktičku obradbu s ciljem pronalaženja i razredbe naziva (ili imenovanih entiteta, en. *named entities*) u hrvatskim tekstovima i također je temeljen na pravilima.

U poglavlju 3.2 opisuju se dva skupa eksperimenata s ovisnosnim parsanjem hrvatskih tekstova. Budući da su oba zasnovana na Hrvatskoj ovisnosnoj banci stabala, najprije se predstavlja njezina najnovija inačica i iz nje izrađeni skupovi za treniranje modela parsera i njihovo testiranje.

Prvi skup eksperimenata, opisan u poglavlju 3.2.2, predstavlja testiranje točnosti i učinkovitosti sustava MaltParser i MSTParser na Hrvatskoj ovisnosnoj banci stabala u različitim testnim okruženjima s obzirom na postavke samih sustava te utjecaje izmjena u strukturi dostupnih metapodataka iz banke stabala.

U poglavlju 3.2.3, na osnovi testiranja postojećih sustava, prikazuje se novi model ovisnosnoga parsanja hrvatskih tekstova koji se gradi povezivanjem jednoga od postojećih pristupa s dostupnim jezičnim resursima, odnosno valencijskim rječnikom najučestalijih

hrvatskih glagola CROVALLEX. Nakon prikaza toga modela parsera slijedi opis njegove računalne izvedbe i testiranje točnosti i učinkovitosti u okruženju identičnome onom iz testiranja postojećih sustava, u svrhu usporedbe postignutih rezultata.

3.1 Postojeći pristupi

Pregled stanja s jezičnim tehnologijama za hrvatski jezik u (Tadić 2003:49) navodi za sintaktičku razinu jezične obradbe kako "nema parsera (osim jednog ili dva prototipa u akademskim institucijama), nema sustava za prepoznavanje rečeničnih dijelova (osim također prototipnoga sustava za prepoznavanje imeničnih fraza), postoji sustav za segmentaciju teksta na rečenice (također prototip u akademskoj ustanovi)". Navedeni prototipni sustavi iz toga pregleda redom su opisani u (Seljan 2003), (Lauc 2001) i (Boras 1998). Šest godina kasnije (Vučković 2009:88) navodi kako se "LFG model za rečeničnu analizu" prikazan u (Seljan 2003) i "sustav OZANA" (Bekavac 2005) mogu "smjestiti na početke računalne sintaksne analize hrvatskoga jezika". S obzirom na navedeno, ovdje se ukratko i slijedom pojavljivanja opisuju tri postojeća pristupa obradbi hrvatskoga jezika na razini sintakse:

1. sustav iz (Seljan 2003), parser hrvatskih tekstova u okviru formalizma leksičko-funkcionalne gramatike (en. *Lexical Functional Grammar*, LFG),
2. sustav OZANA iz (Bekavac 2005), sustav za pronalaženje i razredbu naziva u hrvatskim tekstovima, temeljen na pravilima i
3. razdjelnik i plitki parser hrvatskih tekstova prikazan u (Vučković i dr. 2008, Vučković 2009, Vučković i dr. 2010), temeljen na pravilima i djelomično na ovisnosnoj sintaksi.

Za sustav iz (Seljan 2003) navodi se u (Tadić 2007:85-86) kako se radi o "pokušaju izrade parsera temeljenoga na formalizmu leksičko-funkcionalne gramatike" koji je "ostao na razini prototipa" te se "zasigurno ne bi mogao upotrijebiti za razmjerniju obradbu i izgradnju banaka stabala, budući da je nedovoljno robustan i pokriva samo ograničeni skup odabranih sintaktičkih struktura te se ne može nositi sa svim vrstama složenih rečenica". Navedeni je sustav zapravo uporabljen kao ilustracija teorijskoga računalnog modela nekih sintaktičkih pojava u hrvatskome jeziku u okviru formalizma LFG i ovdje se ne razmatra.

Sustav OZANA (Bekavac 2005, Bekavac i Tadić 2007) je sustav za pronalaženje i razredbu naziva (en. *named entity recognition and classification*, NERC). Razvijen je kao

sustav temeljen na pravilima, odnosno na kaskadnoj uporabi konačnih pretvarača (en. *finite state transducers*, FST, usp. Vučković 2009) nad Hrvatskim morfološkim leksikonom i specijaliziranim popisima naziva, i to kao modul unutar lingvističkoga razvojnog okruženja INTEX (Silberztein 1999, 1999b). Naknadno je razvijena i samostalna inačica toga sustava koja kao ulaz prima pravila razvijena u INTEX-u i izvezena iz njega, jezične resurse i tekst hrvatskoga jezika u kojemu je potrebno pronaći i razvrstati nazive (Bekavac i dr. 2009). Ručnim je vrjednovanjem utvrđeno kako sustav OZANA postiže zbirnu F_1-mjeru od 90% pri pronalaženju i svrstavanju u sedam razreda naziva[85] nad tekstovima iz informativne domene. Prepoznavanje i razredba naziva po ranijoj definiciji nije parsanje tekstova prirodnoga jezika, no zbog odabranoga se teorijskog okvira sustav OZANA može smatrati sustavom koji, poput razdjelnika (en. *chunker*), provodi neku vrstu razdjeljivanja rečenične strukture na nazive i dijelove rečenice koji nisu nazivi. Nadalje, sustav pronalazi nazive u tekstu razdvojenome na rečenice, odnosno – poput parsera teksta u ranijoj definiciji – pretpostavlja rečenicu ili skup rečenica hrvatskoga jezika kao ulazne podatke, pa ga se može smatrati sustavom za obradbu isključivo unutar rečenične strukture. Također se radi o prvome sustavu za obradbu hrvatskih tekstova na rečeničnoj razini unutar teorijskoga okvira kaskada konačnih pretvarača, uz primjenu većine tada dostupnih jezičnih resursa za hrvatski jezik.

Sustav SynCro (Vučković i dr. 2010c) izgrađen je s ciljem proširivanja sustava OZANA "prepoznavanjem ostalih sintagmi u hrvatskom jeziku, uključujući imenske, glagolske, pridjevske i prijedložne sintagme" i pokušajem prikazivanja "barem nekih od njihovih međusobnih odnosa" (Vučković 2009:88). Sustav je početno zamišljen kao razdjelnik temeljen na pravilima (usp. Vučković i dr. 2008), izrađen u obliku kaskada konačnih pretvarača korištenjem razvojnog okruženja NooJ[86] (Silberztein 2004), nasljednika sustava INTEX. Sustav je kasnije u nekoliko navrata dopunjavan i pretvoren u (Vučković 2009:171) "djelomični plitki parser upravljan gramatikom" zasnovan na "gramatici ovisnosti i lokalnim gramatikama". Ranije inačice sustava parsale su samo jednostavne rečenice hrvatskoga jezika, a trenutno se razvijaju poboljšane inačice, s obzirom na rukovanje složenim rečenicama (usp. Štefanec i dr. 2010, Vučković i dr. 2010b, 2010c), kao i s obzirom na postupke predobradbe ulaznih podataka (usp. Vučković i dr. 2010). U prvome koraku sustav uporabom kaskada konačnih pretvarača, odnosno slijedom lokalnih regularnih gramatika razdjeljuje ulazni tekst

[85] Detalji u (Bekavac 2005). Radi se o ovim razredima naziva: imena osoba (PERSON), lokacija (LOCATION) i oganizacija (ORGANIZATION) te vremenski (DATE, TIME) i brojčani (MONEY, PERCENT) izrazi. Vidjeti i URL http://www.itl.nist.gov/iaui/894.02/related_projects/muc/proceedings/ne_task.html.
[86] Alat je dostupan na URL-u http://www.nooj4nlp.net.

na imenske, glagolske, prijedložne, atributske i apozicijske sintagme. To je razdjeljivanje po prirodi temeljeno na gramatici fraznih struktura budući da se razdvojenim rečeničnim elementima dodjeljuju grupne oznake sintagmi. Međutim, u drugome se koraku, opisanom u (Vučković i dr. 2010c), ranije razdvojeni elementi povezuju u strukturu nalik ovisnosnomu stablu. Tako dobivena struktura nije ovisnosno stablo prema ovdje iznesenoj definiciji, no modelira ovisnosni odnos među elementima rečeničnoga ustroja, dijelom prema osnovnim postavkama ovisnosne sintakse (Tesnière 1959). Ukupna točnost sustava, izražena zbirnom F_1-mjerom nad svim razredima sintagmi, iznosi 92.34%, a dobivena je ručnim vrjednovanjem (Vučković 2009:169-170) "nad 146 jednostavnih rečenica", ručno odabranih iz jednoga podkorpusa Hrvatskoga nacionalnog korpusa (Tadić 2002, 2009), o kojemu se više govori u opisu Hrvatske ovisnosne banke stabala dalje u tekstu. Točnost sustava je u postupku njegove izrade značajno podignuta (Vučković i dr. 2010e) povezivanjem s valencijskim rječnikom CROVALLEX (Mikelić Preradović 2008, Mikelić Preradović i dr. 2009).

Osim navedenih sustava i ranije spomenutoga sustava (Lauc 2001) koji je sadržavao prototipnu funkcionalnost pronalaženja imeničnih fraza, do provođenja ovoga istraživanja i prema dostupnim saznanjima nije zabilježena izvedba nijednoga drugog pristupa parsanju hrvatskih tekstova, osim onoga preliminarnog u (Berović i dr. 2012) koji je proizvod upravo rane faze ovdje predstavljenoga istraživanja. Budući da nijedan od tri ranije navedena sustava nije ovisnosni parser u smislu iznesene definicije ovisnosnoga parsanja i ovisnosnoga stabla, ne postoji mogućnost njihove primjene u ovome istraživanju. S obzirom na bliskost paradigmi ovisnosnoga parsanja, može se eventualno teorijski razmatrati sustav SynCro, no on ipak – s ozbirom na ovdje navedena svojstva ovisnosnih stabala i kriterije vrjednovanja ovisnosnoga parsanja – ne udovoljava u dovoljnoj mjeri ovdje postavljenim zahtjevima ili preduvjetima, posebno s gledišta kriterija robusnoga razrješavanja višeznačnosti.

Slijedi opis eksperimenata s ovisnosnim parsanjem hrvatskih tekstova temeljenim na podatcima, prema zadanim definicijama, svojstvima, ograničenjima i formalnim kriterijima vrjednovanja ovisnosnoga parsanja.

3.2 Ovisnosno parsanje hrvatskih tekstova

Ovdje se opisuju eksperimenti s ovisnosnim parsanjem hrvatskih tekstova korištenjem Hrvatske ovisnosne banke stabala. Prvo se predstavlja sama banka stabala, a potom dva skupa eksperimenata – jedan s postojećim ovisnosnim parserima temeljenim na podatcima, odnosno

teoriji grafova (MSTParser) i prijelazima (MaltParser) te jedan s uporabom novoga pristupa ovisnosnomu parsanju hrvatskih tekstova, temeljenoga na podatcima i korištenju valencijskoga rječnika hrvatskih glagola. Hrvatska ovisnosna banka stabala predstavlja se kratko, uz prikaz njezinih osnovnih statističkih značajki. Prikaz svakoga od eksperimenata sastoji se od prikaza korištenih sustava, testnoga okruženja i plana eksperimenta te rasprave rezultata. Prikazu i raspravi rezultata u drugome skupu eksperimenata prethodi formalni opis novopredloženoga modela ovisnosnoga parsanja hrvatskih tekstova.

3.2.1 Hrvatska ovisnosna banka stabala

Plan izgradnje Hrvatske ovisnosne banke stabala (HOBS)[87] – kao prvoga sintaktički označenoga korpusa hrvatskih tekstova – iznesen je prvotno u (Tadić 2006b, 2007), zajedno s prikazom prve inačice korpusa koja se sastojala od 100 rečenica. U (Tadić 2007:85-87) navodi se kako je "s obzirom na postojanje banaka stabala slavenskih jezika" – poput primjerice češkoga (PDT, Hajič i dr. 2000), bugarskoga, ruskoga, poljskoga i slovenskoga (Džeroski i dr. 2006) – "provedeno istraživanje nad tim bankama stabala s ciljem pronalaženja najprikladnijega sintaktičkoga formalizma za primjenu na hrvatskim tekstovima, s obzirom na svojstva hrvatskoga jezika, ali i na vanjezična projektna ograničenja". U skladu s ranijom raspravom o fraznoj i ovisnosnoj strukturi rečenice, za HOBS je odabran pristup temeljen na ovisnosnoj sintaksi i formalizmu naziva *funkcionalni generativni opis* (en. *Functional Generative Description*, FGD, Sgall i dr. 1986), odnosno pristup korišten u PDT-u.

Za korpus hrvatskih tekstova nad kojim se izgradio HOBS odabran je podkorpus Hrvatskoga nacionalnog korpusa (HNK, Tadić 2002, 2009)[88] pod nazivom CW2000, odnosno CW100. Korpus CW100, odnosno korpus *Croatia Weekly 100 kw* (usp. Agić i Tadić 2006) "sastoji se od članaka izdvojenih iz sedam izdanja časopisa *Croatia Weekly*, koji je objavljivan od 1998. do 2000. godine od strane Hrvatskoga informativno-kulturnoga zavoda (HIKZ) te je također i dio hrvatske strane Hrvatsko-engleskoga paralelnoga korpusa (Tadić 2000)". Korpus je "ručno lematiziran i morfosintaktički označen prema standardu Multext East v3 (Erjavec 2004) te zapisan u formatu XCES (Ide i dr. 2000)" i sadrži ukupno 118,529 pojavnica u 4,626 rečenica. Detaljan opis izrade korpusa CW100 može se pronaći u (Tadić 2000), a njegove osnovne statističke značajke u (Agić i Tadić 2006). Taj je podkorpus odabran jer je u trenutku pokretanja izrade HOBS-a bio jedini dostupan korpus hrvatskoga

[87] Vidjeti i URL Hrvatske ovisnosne banke stabala, http://hobs.ffzg.hr.
[88] Vidjeti i URL Hrvatskog nacionalnog korpusa, http://hnk.ffzg.hr.

jezika s obavljenim ručnim razdvajanjem na rečenice i pojavnice koji je ručno morfosintaktički označen i lematiziran, pa ga se smatralo logičnim izborom s obzirom na ograničenost ljudskih resursa raspoloživih za HOBS, ali i potrebu za što većim brojem opisanih jezičnih razina, ukoliko bi se banka stabala koristila za treniranje parsera temeljenih na podatcima. Također, korpus CW100 sadrži tekstove iz informativne domene koja se tada smatrala (i još se uvijek smatra, primjerice, s obzirom na usložnjavanje zahtjeva krajnjih korisnika vezano uz pronalaženje podataka s Interneta i njihovu inteligentnu daljnju obradbu) domenom od osobite važnosti za strojnu obradbu hrvatskih tekstova.

Najnovija inačica HOBS-a, prvi put predstavljena i upotrebljena u ovome istraživanju, izgrađuje se nad ručno lematiziranim i morfosintaktički označenim korpusom CW100 ručnim označavanjem – povezivanjem riječi u ovisnosne odnose i dodjelom sintaktičkih funkcija tim ovisnosnim odnosima – pomoću računalnoga paketa TrEd (Pajas 2000), razvijenoga upravo za potrebe izgradnje PDT-a. HOBS se ljudskim naporom i dalje neprekidno proširuje u paketima od 35 rečenica iz korpusa CW100, koje po ručnom označavanju prolaze i dodatnu dvostruku provjeru od strane stručnjaka za sintaksu hrvatskoga jezika koji se brinu za sustavnost primjene i prilagodbe formalizma iz PDT-a na hrvatske tekstove. Jedan prikaz značajnijih prilagodbi praškoga formalizma na posebnosti hrvatskih tekstova za potrebe razvoja HOBS-a dan je u (Berović i dr. 2012), praćen opisom jedne starije inačice HOBS-a. Po sintaktičkom označavanju čitavoga korpusa CW100, prema planu iz (Tadić 2007), u HOBS će se početi uključivati rečenice iz hrvatskoga prijevoda Orwellova romana *1984.*, odnosno hrvatskoga dijela višejezičnoga paralelnog korpusa iz domene književnosti, javno dostupnoga u sklopu četvrte inačice standarda Multext East (Erjavec 2010, usp. Agić i dr. 2011). Najnovija inačica HOBS-a, korištena u ovome istraživanju, prikazana je osnovnim statističkim značajkama u tablici 3-1 i dodatno slikom 3-1.

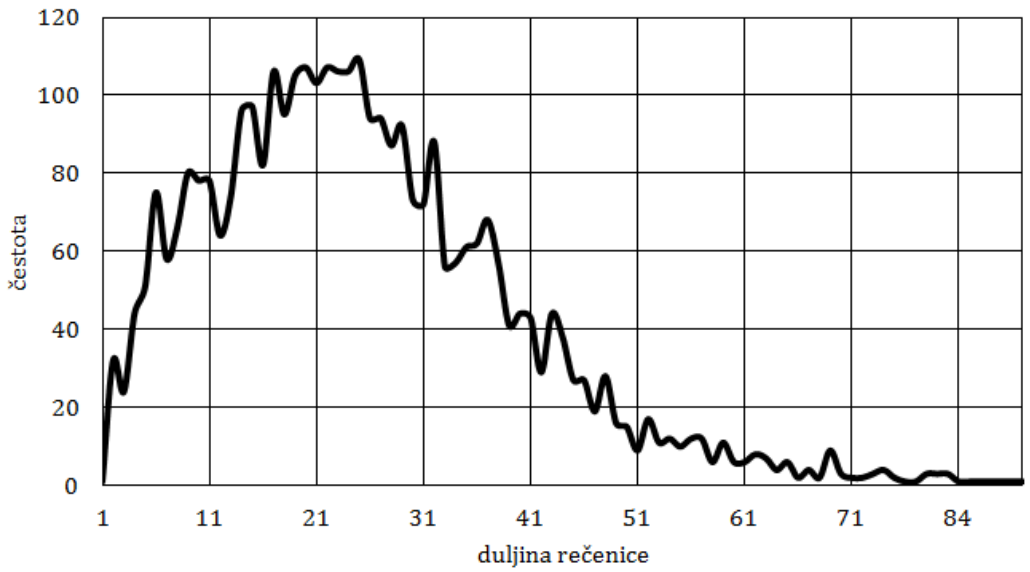
Od ukupno 4,626 rečenica iz korpusa CW100, u HOBS je trenutno uneseno 3,465 rečenica (74.9%), odnosno 88,045 od 118,529 pojavnica (74.3%). U HOBS-u se pojavljuje ukupno 828 od 896 morfosintaktičkih oznaka korištenih u korpusu CW100 (92.4%). Usporedbe radi, u Hrvatskom morfološkom leksikonu (Tadić i Fulgosi 2003), odnosno Hrvatskom lematizacijskom poslužitelju (Tadić 2005, 2006), postoji ukupno 1,475 različitih morfosintaktičkih oznaka za oko 4 milijuna pojavnica izvedenih iz oko 110,000 lema[89]. Slika 3-1 dodatno opisuje rečenice u HOBS-u prema duljini mjerenoj brojem pojavnica koje

[89] Dakle, u CW100 se pojavljuje 60.75% MSD-oznaka iz HML-a, a u HOBS-u 56.14%.

sadržavaju. U skladu s tablicom 3-1, prosječna rečenica HOBS-a sadrži 25.41 riječi. Slika 3-1, kao grafički prikaz čestote pojavljivanja rečenica s određenim duljinama u broju pojavnica, potvrđuje taj prosjek. Iz nje se, primjerice, može pročitati da se rečenice duljine između 15 i 25 pojavnica pojavljuju u HOBS-u redom više od 100 puta, a rečenice dulje od 50 pojavnica manje od 10 puta[90]. Takva je razdioba karakteristična upravo za tekstove iz informativne domene, poput onih u korpusu CW100.

Tablica 3-1 Osnovne statističke značajke HOBS-a

značajka	broj
rečenica	3,465
pojavnica	88,045
oblik	20,703
lema	10,481 (10,527)
morfosintaktička oznaka	828
sintaktička funkcija	26 (69)[91]



Slika 3-1 Čestota pojedinih duljina rečenica u HOBS-u

[90] Točnije, budući da je tablica s duljinama rečenica prevelika za prikazivanje u cijelosti, u razdiobi se može vidjeti da je u HOBS-u bilo 78 rečenica duljine od 10 pojavnica, 107 rečenica duljine 20 pojavnica i 109 rečenica duljine 25 pojavnica. Potonji par je ekstrem razdiobe. Najdulja rečenica u HOBS-u ima 148 riječi. Ostale statističke značajke korpusa CW100 mogu se pronaći u (Agić i Tadić 2006).
[91] Osnovne i proširene sintaktičke funkcije opisane su dalje u tekstu.

Za sintaktičku analizu rečenica u HOBS-u korišteno je ukupno 69 različitih sintaktičkih funkcija – koje se prema sintaktičkomu formalizmu FGD nazivaju i *analitičkim funkcijama* (en. *analytical function*) – preuzetih izravno iz priručnika za sintaktičko označavanje PDT-a (Hajič i dr. 1999), uz dogovornu prilagodbu formalizma na hrvatske tekstove tijekom postupka izrade HOBS-a. Korištene sintaktičke funkcije mogu se načelno podijeliti na osnovne i proširene. Prema (Hajič i dr. 1999:14), osnovne su sljedeće sintaktičke funkcije.

1. Pred – predikat, odnosno čvor koji ne ovisi ni o jednome drugom čvoru, nego se veže za korijenski čvor rečenice,
2. Sb – subjekt,
3. Obj – objekt,
4. Adv – priložna oznaka,
5. Atv – komplement,
6. AtvV – predikatni proširak,
7. Atr – atribut,
8. Pnom – imenski predikat, odnosno imenski dio imenskoga predikata,
9. AuxV – pomoćni glagol,
10. Coord – oznaka za koordinacijski element kod nezavisno-složenih rečenica,
11. Apos – apozicija,
12. AuxT – povratna zamjenica,
13. AuxR – povratna zamjenica kod refleksivnoga pasiva, odnosno obezličenja,
14. AuxP – prijedlog koji uvodi prijedložno-padežni izraz,
15. AuxC – oznaka za koordinacijski element koji uvodi zavisno-složene rečenice,
16. AuxO – modalna čestica,
17. AuxZ – pojačivač,
18. AuxX – svaki zarez, osim onoga koji se koristi kao koordinacijski veznik,
19. AuxG – nezavršni grafički simboli,
20. AuxY – nesvrstani prilozi i čestice,
21. AuxS – korijenski čvor ovisnosnoga stabla,
22. ExD – vanjska ovisnost, oznaka za glavni element rečenice bez predikata,
23. AtrAtr – atribut bilo koje od nekoga broja imenica,
24. AtrAdv – oznaka za sintaktičku višeznačnost s obzirom na priložnu i imeničnu ovisnost, koja ne uvodi semantičku višeznačnost,
25. AdvAtr – obrnuto od prethodne,

26. AtrObj – sintaktička višeznačnost između objektne i druge imenične ovisnosti, koja ne uvodi semantičku višeznačnost i
27. ObjAtr – obrnuto od prethodne.

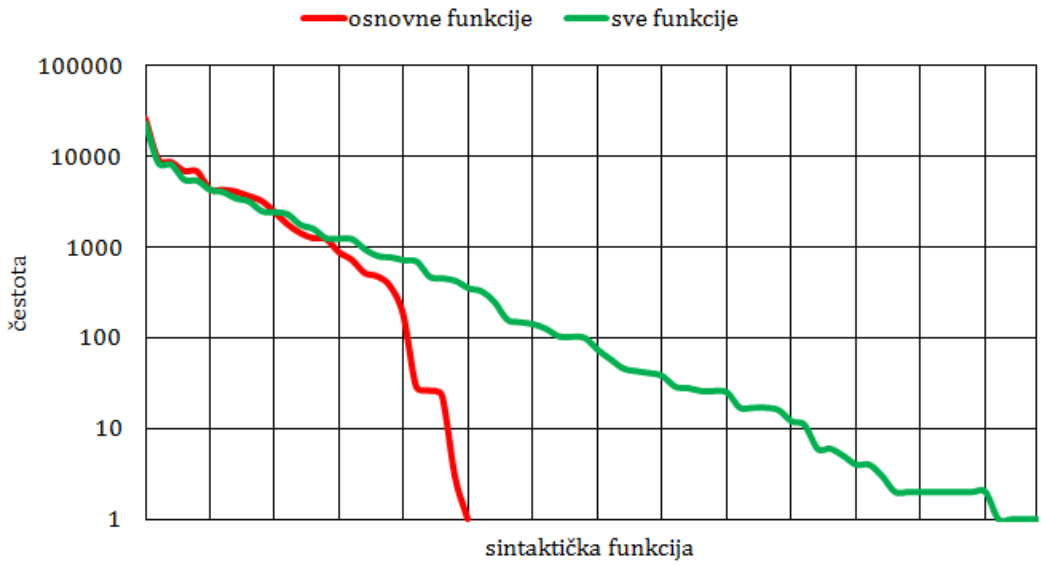
Iz osnovnih se sintaktičkih funkcija – koje odgovaraju ranije izloženim samostalnim i nesamostalnim elementima rečeničnoga ustroja te nezavisno- i zavisno-složenim rečenicama – dalje stvaraju proširene sintaktičke funkcije, i to dodavanjem pobliže opisujućega sufiksa na postojeću osnovnu funkciju. Primjerice, ukoliko se neka nezavisno-složena rečenica sastoji od dvije surečenice, svaki od predikata (Pred) tih surečenica bit će označen sintaktičkom funkcijom predikata koji pripada nezavisno-složenoj rečenici, uvedenoj koordinacijskim veznikom (Pred_Co).

Tablica 3-2 Razdioba osnovnih sintaktičkih funkcija u HOBS-u

Atr	**Adv**	**AuxP**	**Sb**	**Obj**	**AuxX**	**Pred**
25816	9150	8607	6874	6776	4316	4255
AuxV	**Coord**	**AuxK**	**AuxG**	**AuxC**	**Pnom**	**AuxZ**
4075	3627	3219	2441	1782	1422	1248
ExD	**AuxY**	**AuxR**	**Apos**	**AuxT**	**Atv**	**AtvV**
1225	876	721	516	473	366	178
AtrAdv	**AuxO**	**AdvAtr**	**AtrObj**	**ObjAtr**		
29	26	23	3	1		

Budući da se u HOBS-u korijenski čvor ne navodi eksplicitno, u označavanju rečenica korištene su sve osnovne sintaktičke funkcije, osim funkcije za korijenski čvor (AuxS), njih ukupno 26. Ukupno je korišteno 69 različitih sintaktičkih funkcija, što znači da su se koristile i 43 proširene funkcije. Čestota osnovnih sintaktičkih funkcija u HOBS-u dana je u tablici 3-2, a čestota svih sintaktičkih funkcija prikazana je tablicom 3-3. Pri navođenju osnovnih sintaktičkih funkcija u tablici 3-2, svakoj su osnovnoj funkciji pridružene i čestote onih proširenih funkcija koje joj pripadaju. Tako je, prema ranijem primjeru, čestoti osnovne sintaktičke funkcije Pred pridružena i čestota proširene sintaktičke funkcije Pred_Co. Ove tablice popraćene su i slikom 3-2 koja grafički prikazuje čestotu osnovnih i svih sintaktičkih funkcija, padajuću linearnu funkciju na logaritamskoj ljestvici, što implicira eksponencijalnu

padajuću funkciju na odgovarajućoj linearnoj ljestvici, odnosno očekivanu uporabu manjega broja sintaktičkih funkcija u velikom broju ovisnosnih relacija te rijetku uporabu većega broja sintaktičkih funkcija. Primjerice, s obzirom na sliku 3-2, u tablici 3-2 može se vidjeti relativno česta uporaba sintaktičke funkcije subjekta (Sb) u odnosu na predikatni proširak (AtvV) ili modalnu česticu (AuxO).



Slika 3-2 Čestota sintaktičkih funkcija u HOBS-u

Budući da se HOBS neprekidno dopunjava novim rečenicama, pa se u postupak razvoja može u svakome trenutku intervenirati, vrijedi primijetiti neke moguće nepravilnosti u dodjeli sintaktičkih funkcija koje se impliciraju razdiobom njihovih pojavljivanja[92]. Primjerice, u tablici 3-2 može se vidjeti kako je osnovna sintaktička funkcija atributa (Atr) dodijeljena ukupno 25,816 puta.

Prema definiciji ovisnosnoga stabla i ovisnosnoga parsanja, sintaktičke se funkcije u izvedbi dodjeljuju jednoj od riječi u ovisnosnoj relaciji, najčešće dependentu. Dakle, dodjela sintaktičkih funkcija ima onoliko koliko je riječi u rečenici, odnosno u banci ovisnosnih stabala. Stoga vrijedi promatrati broj dodjela određenih sintaktičkih funkcija u odnosu prema broju svih dodjela, odnosno broju pojavnica u HOBS-u.

[92] Ishodište ovoga istraživanja ipak je u postupcima parsanja tekstova hrvatskoga jezika pa iznesena analiza HOBS-a kao banke ovisnosnih stabala korištene za izradu jezičnih modela parsera temeljenih na podatcima nipošto nije opširna, već samo ilustrativna i eventualno uporabljiva.

Tablica 3-3 Razdioba svih sintaktičkih funkcija u HOBS-u

Atr	**AuxP**	**Adv**	**Sb**	**Obj**	**AuxX**	**AuxV**
23207	8527	8179	5572	5438	4316	4074
Coord	**AuxK**	**Pred**	**AuxG**	**Atr_Co**	**AuxC**	**Pred_Co**
3468	3219	2512	2441	2316	1771	1597
Pnom	**AuxZ**	**Obj_Co**	**Sb_Co**	**Adv_Co**	**AuxY**	**AuxR**
1255	1244	1230	948	799	776	721
ExD	**AuxT**	**Apos**	**ExD_Co**	**Sb_Ap**	**Atv**	**Atr_Ap**
695	473	454	427	354	328	250
AtvV	**Pnom_Co**	**Pred_Pa**	**Adv_Ap**	**Coord_Ap**	**Obj_Ap**	**AuxY_Pa**
159	149	142	126	104	103	100
ExD_Pa	**Apos_Co**	**Adv_Pa**	**Atr_Pa**	**AuxP_Ap**	**Coord_Co**	**ExD_Ap**
74	58	46	43	41	38	29
AuxP_Co	**AuxO**	**Atv_Co**	**AtrAdv**	**Coord_Pa**	**AtvV_Co**	**AdvAtr**
28	26	26	25	17	17	17
Pnom_Ap	**Atv_Pa**	**AuxP_Pa**	**AuxC_Co**	**AdvAtr_Co**	**Obj_Pa**	**AtrAdv_Co**
16	12	11	6	6	5	4
Pred_Ap	**AuxC_Pa**	**Apos_Pa**	**AtrObj_Ap**	**AuxC_Ap**	**Pnom_Pa**	**Apos_Ap**
4	3	2	2	2	2	2
AtvV_Pa	**AuxZ_Pa**	**AuxZ_Co**	**ObjAtr**	**AtrObj**	**AuxV_Co**	
2	2	2	1	1	1	

U skladu s time, od ukupno 88,045 riječi, u HOBS-u je 25,816 riječi (29.32%) opisano sintaktičkom funkcijom atributa. Imajući na umu ranije izloženu raspravu o samostalnim i nesamostalnim elementima rečeničnoga ustroja rečenica hrvatskoga jezika, vrijedi čestotu funkcije atributa usporediti s čestotom funkcije apozicije (Apos) zbog slične prirode i zbog očekivane česte uporabe ovih funkcija u novinskim tekstovima. U HOBS-u je ukupno 516 riječi (oko 0.59%) označeno apozicijama. Dakle, otprilike je 50 puta više atributa nego apozicija prema referentnome ovisnosnom parsanju dostupnom u HOBS-u.

Tablica 3-4 Razdioba sintaktičkih funkcija po vrstama riječi u HOBS-u

	pridjev (A)	veznik (C)	broj (M)	imenica (N)	zamjenica (P)	prilog (R)	prijedlog (S)	glagol (V)
Adv	299	127	359	4800	401	2421	85	647
Apos	1	17	0	1	4	33	1	3
Atr	9477	4	752	11209	1586	221	5	1644
AuxC	0	1517	0	2	126	57	28	5
AuxP	2	100	0	190	0	46	8260	1
Coord	0	3141	0	2	14	32	6	1
Obj	120	2	138	3644	860	42	1	1927
Pnom	517	0	37	670	32	59	2	102
Pred	63	0	0	1	3	0	0	4188
Sb	82	1	196	4853	1179	41	2	337

Ova se nepodudarnost s jezičnom intuicijom dodatno potvrđuje tablicom 3-4 koja predstavlja razdiobu odabranih sintaktičkih funkcija prema podatcima o vrstama riječi, odnosno prema odabranim morfosintaktičkim oznakama pripadajućih pojavnica. Tablica 3-4, iako joj je svrha prikazati samu razdiobu, također implicitno potvrđuje postojanje problema u označavanju ovisnosnih relacija sintaktičkom funkcijom apozicije budući da je samo jedna od svih apozicija u HOBS-u po vrsti riječi imenica, što je u potpunoj suprotnosti s ranije iznesenom definicijom apozicije. Ovdje se pretpostavlja – s obzirom na uočenu veliku razliku u broju atributa i apozicija – kako je veći dio riječi koje su u sintaktičkoj strukturi morale biti označene kao apozicije označen oznakom za atribut[93]. Ta se hipoteza dodatno potvrđuje činjenicom da je među atributima više imenica (43.42%) nego pridjeva (36.7%), iako se intuitivno očekuje upravo obratno. Također vrijedi primijetiti i 102 imenska predikata (7.17% od svih imenskih predikata) koji su prema vrsti riječi glagoli, što je ponovno u suprotnosti s definicijom imenskoga predikata[94].

[93] Primjerice, u HOBS-u je čest izraz "predsjednik X", gdje je X neka osobna imenica. Riječ "predsjednik" je u tome izrazu sustavno označavana funkcijom atributa, iako se po definiciji radi o apoziciji.

[94] Ovo opažanje vezano je uz razinu morfosintaktičkih oznaka, preuzetu iz CW100. Primjerice, glagolski pridjev trpni – koji može sačinjavati imenski predikat – označava se prema standardu Multext East i HML-u kao glagol, a ne kao pridjev.

Tablica 3-4 otkriva i neke manje učestale pogrješke o kojima se ovdje ne raspravlja, kao što se ne raspravlja ni o vrlo rijetkoj uporabi nekih osnovnih i proširenih sintaktičkih funkcija, koja bi mogla implicirati smislenost njihove zamjene nekom od funkcija s više razine uopćenosti. Neke značajke koje bi mogle upućivati na odudaranje od definicije pojedinih sintaktičkih funkcija, odnosno elemenata rečeničnoga ustroja zapravo se najčešće odnose na uvođenje zavisno-složenih rečenica. Primjerice, glagoli kojima je značajan broj puta pridružena sintaktička funkcija objekta zapravo uvode objektne rečenice. Istraživanje (Berović i dr. 2012b) predlaže stoga podsustav zasebnih sintaktičkih funkcija kojima se u glavnu rečenicu uvode zavisne surečenice kako bi se izbjegla ova nepodudarnost sustava sintaktičkih funkcija prema pripadajućemu sintaktičkom opisu hrvatskoga jezika.

Unatoč kratkoj raspravi o mogućim nedostatcima HOBS-a – koji su dijelom očekivani, s obzirom na to da se radi o jezičnome resursu koji se još uvijek nalazi u fazi razvoja i budući da su se tek nedavno (usp. Berović i dr. 2012) počeli ocrtavati formalni pristupi razrješavanju problema s primjenom formalizma iz PDT-a na sintaktičku strukturu hrvatske rečenice – ovo istraživanje nije usmjereno daljnjem preispitivanju njegove kvalitete. S gledišta ovisnosnoga parsanja temeljenoga na podatcima, banka stabala nad kojom se provodi eksperiment – nad kojom se trenira i testira jezični model parsera – predstavlja sliku parsanoga prirodnog jezika kojoj se bira u potpunosti vjerovati, neovisno o eventualnome vrjednovanju njezine kvalitete. Budući da je HOBS u ovome trenutku jedina dostupna banka ovisnosnih stabala hrvatskoga jezika, ona će se u ovdje predstavljenim eksperimentima s ovisnosnim parsanjem temeljenim na podatcima smatrati valjanim implicitnim opisom hrvatskoga jezika na razini ovisnosne sintakse. S obzirom na to, o svojstvima HOBS-a se u ovome istraživanju kratko tehnički raspravlja još samo u opisu eksperimenata s ovisnosnim parsanjem, gdje se skup rečenica HOBS-a dijeli na skupove za treniranje i skupove za testiranje jezičnih modela, pa je potrebno opisati osnovne statističke značajke tih skupova. Slijedi opis prvoga skupa eksperimenata s parsanjem hrvatskih tekstova iz HOBS-a ovisnosnim parserima temeljenima na podatcima i rasprava o postignutim rezultatima.

3.2.2 Eksperiment s postojećim ovisnosnim parserima

Svrha eksperimenta s uporabom odabranih (i javno dostupnih) na podatcima temeljenih ovisnosnih parsera na zadatku parsanja hrvatskih tekstova s pomoću HOBS-a je višestruka. Ovdje se tim eksperimentom nastoji:

1. utvrditi primjenjivost – u smislu vrjednovanja točnosti i učinkovitosti – ovisnosnih parsera temeljenih na grafovima i ovisnosnih parsera temeljenih na prijelazima na parsanje tekstova hrvatskoga jezika i
2. odabrati jedan od tih teorijskih okvira kao polazišnu točku za izradu modela (i) ovisnosnoga parsera ciljanoga specifično tekstovima hrvatskoga jezika.

Opis eksperimenta uključuje redom kratki opis ovisnosnih parsera – točnije, jezičnih modela i algoritama za parsanje dobivenih s pomoću odabranih generatora parsera, odnosno računalnih sustava za ovisnosno parsanje – korištenih u eksperimentu, opis postavki eksperimenta – skupova za treniranje i testiranje dobivenih iz HOBS-a, odabira mjera za vrjednovanje te promatranih unutarnjih i vanjskih utjecaja na pojedine vrjednovane značajke ovisnosnih parsera – te prikaz i raspravu o rezultatima.

3.2.2.1 Ovisnosni parseri

Ovisnosno parsanje temeljeno na podatcima ranije je predstavljeno unutar dva različita teorijska okvira, i to kao parsanje temeljeno na grafovima i prijelazničko parsanje. U prikazu i raspravi rezultata s natjecanja u ovisnosnome parsanju CoNLL 2006 i CoNLL 2007, sustavi MaltParser i MSTParser izdvojeni su kao dva sustava s najboljim rezultatima vrjednovanja točnosti, ali i kao predstavnici dvaju teorijskih okvira parsanja temeljenoga na podatcima. U ovome se eksperimentu stoga koriste upravo sustavi MaltParser i MSTParser. Oba se sustava smatraju stvaračima parsera (en. *parser generator*) budući da nude prilagodbu određenoga broja parametara koji utječu na izradu jezičnoga modela iz banke ovisnosnih stabala i odabir algoritma za parsanje ulaznoga teksta tim jezičnim modelom. Pojednostavljeno, ovisno o izboru parametara jezičnoga modela i parsnoga algoritma, definira se s pomoću sustava MaltParser i MSTParser jedan par sačinjen od jednoga jezičnog modela i jednoga parsnog algoritma koji predstavlja jedan ovisnosni parser koji se može primijeniti na tekstu i vrjednovati njegovu točnost i učinkovitost.

3.2.2.1.1 MaltParser

MaltParser[95] je stvarač prijelazničkih ovisnosnih parsera temeljenih na podatcima (usp. Nivre i dr. 2007b) i predstavlja izvedbu ranije prikazanoga općeg modela ovisnosnoga

[95] Ranije je navedeno kako je dostupan na URL-u http://maltparser.org/ (2012-03-28). Njegova je funkcionalnost predstavljena nizom istraživanja, a neka od njih navedena su na URL-u http://maltparser.org/publications.html (2012-03-28).

parsanja temeljenoga na prijelazima. Trenutno važeća inačica podržava sedam algoritama za prijelazničko ovisnosno parsanje (usp. Nivre i Hall 2012:2):

1. algoritmi za projektivno ovisnosno parsanje: algoritam *Nivre eager* (MaltNe), algoritam *Nivre standard* (MaltNs), algoritam *Covington projective* (MaltCp) i algoritam *stack projective* (MaltSp),
2. algoritmi za neprojektivno ovisnosno parsanje: algoritam *Covington non-projective* (MaltCn), algoritam *stack eager* (MaltSe) i algoritam *stack lazy* (MaltSl),
3. algoritmi za projektivno parsanje s dodatnim ograničenjima *planar* i *2-planar*.

Navedeni su algoritmi inačice ranije predstavljenoga općeg algoritma za prijelazničko ovisnosno parsanje i svi su temeljeni na funkciji tumač, a razlikuju se prema metodama pristupa stogu i ulaznoj vrpci za vrijeme parsanja ulazne rečenice. Većina tih algoritama detaljno je prikazana u (Covington 2001) i (Nivre 2008), a detaljan popis dodatnih njihovih prikaza i formalnih vrjednovanja njihove točnosti i učinkovitosti može se pronaći u (Nivre i Hall 2012). Svi gore navedeni algoritmi za projektivno parsanje mogu se u okviru pseudo-projektivnoga parsanja (usp. Nivre i Nilsson 2005) koristiti za neprojektivno parsanje.

MaltParser nudi mogućnost izbora algoritama za treniranje klasifikatora – i to putem sustava LIBSVM (Chang i Lin 2001) ili sustava LIBLINEAR (Fan i dr. 2008) i navođenja postavki treniranja specifičnih za pojedini sustav – i mogućnost eksplicitnoga definiranja značajki za treniranje prema ranije izloženim načelima. Podržan je i niz drugih postavki koje se odnose na specifičnosti ulaznih podataka, a također – osim osnovnih načina rada (treniranje i korištenje modela) – i postupci projektivizacije i deprojektivizacije banaka stabala.

U ovome se istraživanju vrjednuje sustav MaltParser sa svim dostupnim algoritmima za projektivno i neprojektivno ovisnosno parsanje, uz uporabu pseudo-projektivnoga parsanja za projektivne algoritme. Budući da (Nivre i Nilsson 2005:4) navodi kako biblioteka LIBSVM omogućava implicitno povezivanje značajki i posljedično jednostavniju optimizaciju značajki za treniranje jezičnoga modela – unatoč napomeni kako je biblioteka LIBLINEAR vremenski i prostorno učinkovitija – ovdje se koristi upravo biblioteka LIBSVM. Taj se izbor opravdava ciljem postizanja što veće točnosti ovisnosnoga parsanja, uz pretpostavku da se jezični model izveden korištenjem MaltParsera i LIBSVM-a može jednoznačno pretvoriti u jezični model koji koristi LIBLINEAR za moguće praktične primjene u kojima se zahtijeva veća brzina treniranja jezičnih modela.

```
POS     STACK
POS     INPUT
POS     INPUT   1
POS     INPUT   2
POS     STACK   1
POS     STACK   0   1
POS     INPUT   0   -1
DEP     STACK
DEP     STACK   0   0   0   -1
DEP     STACK   0   0   0   1
DEP     INPUT   0   0   0   -1
DEP     STACK   0   0   0   -1  1
LEX     STACK
LEX     INPUT
LEX     INPUT   1
LEX     STACK   0   0   1
CPOS    STACK
CPOS    INPUT
CPOS    STACK   2
LEMMA   STACK
```

Primjer 3-1 Jedna definicija značajki za treniranje jezičnoga modela MaltParserom

Skup značajki ulaznih podataka za treniranje jezičnoga modela preuzet je i prilagođen iz dostupnih definicija za češki i slovenski jezik s natjecanja CoNLL 2006 i 2007[96], kao i parametri za treniranje klasifikatora povezivanjem s odabranom bibliotekom LIBSVM. Skup značajki za treniranje prikazan je primjerom 3-1. Specifikacija jezika za definiranje značajki, ostvarenoga primjerom 3-1, dana je u prikazima sustava MaltParser[97]. Trenutna inačica sustava pruža mogućnost zapisivanja svih postavki eksperimenta s parsanjem, pa tako i definiranja značajki, korištenjem XML-zapisa. MaltParser razvijen je u programskome jeziku Java i podržava uključivanje u druge sustave na razini izvornoga koda i razvijanje dodatnih modula. Izvorni mu je kod javno dostupan za korištenje u istraživačke svrhe pod licencijom ciljano sastavljenom za taj sustav.

[96] Mogu se pronaći na URL-u http://maltparser.org/conll.html (2012-03-28). Vrijedi napomenuti kako su identične definicije dobivene uporabom sustava MaltOptimizer, URL http://nil.fdi.ucm.es/maltoptimizer/.

[97] A može se pronaći, zajedno s primjerima, u nekome opisu starije inačice sustava MaltParser, primjerice na URL-u http://w3.msi.vxu.se/~nivre/research/MaltParser.html (2012-03-28).

3.2.2.1.2 MSTParser

MSTParser (od en. *maximum spanning tree parser*) je stvarač ovisnosnih parsera temeljenih na teoriji grafova, odnosno na ranije prikazanome teorijskom okviru uspostavljanja analogije između optimalnih parsnih stabala i najvećih prostirućih stabala koja je moguće izgraditi za danu ulaznu rečenicu. Sustav je detaljno opisan u (McDonald i dr. 2005a, 2005b, McDonald i Pereira 2006, McDonald i dr. 2006). Podržava sljedeće algoritme:

1. projektivno parsanje korištenjem Eisnerova algoritma (MstEis) i
2. neprojektivno parsanje korištenjem algoritma Chu-Liu-Edmonds (MstCle).

Podržan je ranije prikazan relacijski uvjetovan (en. *arc-factored*) pristup modeliranju, kao i pristup modeliranju uvjetovan parovima ovisnosnih relacija, odnosno korištenje relacijski uvjetovanoga sintaktičkog modela drugoga reda (en. *second order arc-factored language model*). U trenutno važećoj inačici podržana je i mogućnost dodavanja brojčane mjere pouzdanosti svakoj ovisnosnoj relaciji proizvedenoj algoritmom za parsanje, prema modelu izloženome u (Mejer i Crammer 2010), pa se prema toj mjeri može izvršiti i dodatno vrjednovanje ovisnosnih relacija s obzirom na pouzdanost pri njihovoj dodjeli.

U ovome istraživanju korištena su oba navedena algoritma za parsanje (projektivni MstEis, neprojektivni MstCle) i oba jezična modela (model prvoga reda, model drugoga reda), odnosno četiri različita ovisnosna parsera iz sustava MSTParser.

MSTParser je razvijen u programskome jeziku Java i javno je dostupan za istraživačke svrhe pod inačicom 2.0 licencije Apache[98].

3.2.2.2 Plan eksperimenta

Testiranje prikazanih ovisnosnih parsera na HOBS-u zamišljeno je ovdje u skladu sa smjernicama testiranja provedenoga za potrebe natjecanja u ovisnosnome parsanju CoNLL 2006 i CoNLL 2007. HOBS je zapisan u CoNLL formatu[99] i programskim rješenjem definiran kao skup rečenica iz kojega su slučajnim odabirom izdvojena ovisnosna stabla u odvojene skupove od najmanje i što bliže broju od 5,000 pojavnica, ovisno o broju pojavnica u slučajno odabranim rečenicama, odnosno ovisnosnim stablima. Tako je dobiveno 17

[98] Vidjeti URL http://www.apache.org/licenses/ (2012-03-28).
[99] Definirano je sedam elemenata svake pojavnice: identifikator unutar rečenice, oblik, lema, podatak o vrsti riječi, podatak o morfosintaktičkim kategorijama, glava i sintaktička funkcija.

nepreklapajućih skupova slučajno odabranih rečenica i pripadajućih ovisnosnih stabala iz HOBS-a. Oni su dalje iskorišteni za izradu 17 parova sastavljenih od jednoga skupa za treniranje jezičnih modela i jednoga skupa za testiranje jezičnih modela, po principu spajanja 16 skupova slučajno odabranih rečenica i ovisnosnih stabala u jedan skup za treniranje i odabira preostaloga jednog skupa u jedan skup za testiranje. Zbirne statističke značajke dobivenih 17 parova uzoraka za treniranje i testiranje jezičnih modela dane su u tablici 3-5. Svi podatci u toj i budućim tablicama koje se odnose na zbirne značajke dani su, tamo gdje je taj podatak smislen, uz naznaku pripadajućih 95-postotnih intervala pouzdanosti.

Tablica 3-5 Osnovne statističke značajke uzoraka za treniranje i testiranje modela

značajka	skup za treniranje		skup za testiranje	
rečenica	3261.18 ± 4.20		203.82 ± 4.20	
pojavnica	82865.88 ± 6.87		5179.12 ± 6.87	
oblik	19927.06 ± 15.71		2594.06 ± 12.26	
lema	10166.00 ± 9.19		1909.00 ± 14.12	
morfosintaktička oznaka	817.94 ± 1.40		368.35 ± 4.41	
sintaktička funkcija	69.00 ± 0.00	26. 00 ± 0.00	48.12 ± 0.84	23. 24 ± 0.43

Iz tablice se može vidjeti kako skupovi za treniranje modela sadrže otprilike 3,261 od ukupno 3,465 rečenica iz HOBS-a (94.12%), što ostavlja 5.88% rečenica HOBS-a skupovima za testiranje modela. Otprilike se isti postotci odnose i na brojeve pojavnica, dok brojevi oblika, lema, morfosintaktičkih oznaka i sintaktičkih funkcija slijede logaritamsku funkciju rasta u odnosu na rast broja pojavnica, odnosno rečenica. Tako od ukupnoga broja različitih oblika iz HOBS-a u skupovima za treniranje ostaje oko 96.25%, a u skupovima za testiranje oko 12.53%. Od ukupnoga broja različitih lema iz HOBS-a, skupovi za treniranje zadržavaju oko 96.99%, a skupovi za testiranje oko 18.21%, dok su morfosintaktičke oznake zadržane u trenažnim i testnim skupovima u 98.79- i 44.49-postotnoj količini. U trenažnim su skupovima zadržane sve sintaktičke funkcije iz HOBS-a, dok testni skupovi zadržavaju oko 48 (69.74%) od ukupno 69 funkcija.

Ovako definirani skupovi pretpostavljaju uporabu skupa svih sintaktičkih funkcija u postupku treniranja i testiranja jezičnoga modela. Međutim, s obzirom na raniju podjelu na

osnovne i proširene sintaktičke funkcije, definiran je i uzorak za treniranje i testiranje modela koji sadržava samo 26 osnovnih sintaktičkih funkcija. Taj je uzorak dobiven uklanjanjem proširenoga dijela iz svih sintaktičkih funkcija u svim parovima iz testnoga uzorka koji koristi sve sintaktičke funkcije. Stoga se sve statističke značajke iz tablice 3-5 odnose i na uzorak s osnovnim sintaktičkim funkcijama, osim podatka o broju sintaktičkih funkcija koji je u toj tablici razdvojen. Tako se može vidjeti kako skupovi za treniranje modela, budući da su zadržali sve sintaktičke funkcije, zadržavaju i sve osnovne sintaktičke funkcije, dok je u skupovima za testiranje zadržano oko 89.38% osnovnih sintaktičkih funkcija.

Od 17 parova sačinjenih od jednoga skupa za treniranje i jednoga skupa za testiranje, odabrano je nasumično 10 parova nad kojima su trenirani, testirani i vrjednovani jezični modeli i parsni algoritmi prikazanih ovisnosnih parsera. Postupak se odnosio na skupove za treniranje i testiranje modela sa svim sintaktičkim funkcijama i s osnovnim sintaktičkim funkcijama. Tako je stvoreno 10 uređenih parova (skup za treniranje, skup za testiranje) koji koriste sve sintaktičke funkcije i 10 uređenih parova (skup za treniranje, skup za testiranje) koji koriste samo osnovne sintaktičke funkcije, odnosno ukupno 20 parova nad kojima se provodilo treniranje i testiranje parsera.

Formalno, ako je HOBS definiran kao banka ovisnosnih stabala $T = \{(s_t, G_t)\}_{t=1}^{|T|}$ prema ranijoj definiciji banke ovisnosnih stabala, iz nje slučajnim odabirom izdvojen skup parova $S_R = \{(U_i^{tr}, U_i^{te})\}_{i=1}^{10}$, gdje $\forall i, (U_i^{tr}, U_i^{te})$ predstavlja uređeni par sačinjen od skupa uređenih parova rečenica i pripadajućih referentnih ovisnosnih stabala za treniranje jezičnoga modela i skupa referentnih ovisnosnih stabala za korištenje modela, odnosno parsanje. Skup S_R je definiran i s obzirom na skup svih sintaktičkih funkcija R iz pripadajućega sintaktičkog formalizma HOBS-a. Formalno su skupovi U_i^{tr} i U_i^{te} definirani kao

$$U_i^{tr} = \left\{\left(s_{1_i}, t_{1_i}^R\right), \dots, \left(s_{m_i}, t_{m_i}^R\right)\right\}, \qquad U_i^{te} = \left\{\left(s_{m+1_i}, t_{m+1_i}^R\right), \dots, \left(s_{n_i}, t_{n_i}^R\right)\right\}$$

Pritom za njih vrijedi $\forall i, U_i^{tr} \cap U_i^{te} = \emptyset, U_i^{tr} \cup U_i^{te} = T$, odnosno $|U_i^{tr}| + |U_i^{te}| = |T|, |U_i^{te}| \approx \frac{5.88}{100}|T|$ u skladu sa statističkim značajkama HOBS-a. S ozbirom na postojanje dva skupa ovisnosnih relacija, odnosno sintaktičkih funkcija – skupa svih sintaktičkih funkcija $R_a, |R_a| = 69$ i skupa osnovnih sintaktičkih funkcija $R_b, |R_b| = 26$ prema statističkim značajkama HOBS-a – skup za testiranje ovisnosnih parsera ovim se eksperimentom prema njima konačno definira iz HOBS-a kao $S = \{S_R\}_{R \in \{R_a, R_b\}}$, gdje su $\{S_R\}$ skupovi parova

(U_i^{tr}, U_i^{te}) izdvojenih iz banke stabala s obzirom na skup sintaktičkih funkcija. Takva formalna definicija podržava i dodavanje novih skupova sintaktičkih funkcija koje je moguće izvesti iz skupa svih funkcija R_a trenutno dostupnih u HOBS-u.

Za svaki od parova (U_i^{tr}, U_i^{te}) iz S, odnosno za svaki od parova sačinjenih od jednoga skupa ovisnosnih stabala za treniranje modela i jednoga skupa ovisnosnih stabala za testiranje modela, trenirao se iz skupa za treniranje jezični model za svaki od ranije predstavljenih parsera, pa se taj model s pomoću odgovarajućega parsnog algoritma primijenio na skupu rečenica za testiranje modela. Formalno, definiran je skup $P = \{p_k = (\Gamma_k, \lambda_k, h_k)\}_{k=1}^{|P|}$, gdje svaki element $p_k \in P$, odnosno uređena trojka $p_k = (\Gamma_k, \lambda_k, h_k)$ predstavlja jedan sustav za ovisnosno parsanje (neka od postavki sustava MaltParser i MSTParser) s obzirom na posebnosti jezičnoga modela λ_k i parsnoga algoritma h_k. Skup ograničenja Γ_k u ovome je eksperimentu pritom definiran implicitno, kao u definiciji ovisnosnoga parsanja temeljenoga na grafovima i ovisnosnoga parsanja temeljenoga na prijelazima. Dakle, njime se ograničavaju ulazne i izlazne strukture na skup svih ovisnosnih stabala ulazne rečenice. Za svaki od sustava p_k, primjenom algoritma za izradu jezičnoga modela λ_k^i za svaki skup ovisnosnih stabala U_i^{tr}, dobiveni su parseri p_k^i s parsnim algoritmima h_k^i. Parsni algoritmi potom su primijenjeni na sve rečenice svakoga skupa U_i^{te}. Formalno, postupak izrade modela i postupak parsanja definirani su kao

$$\forall i, k, U_i^{tr} = \{(s_{1_i}, t_{1_i}^R), \dots, (s_{m_i}, t_{m_i}^R)\}, \qquad \lambda_k^i \leftarrow \{t_{j_i}^R\}_{j=1}^{m}$$

$$\forall i, k, U_i^{te} = \{(s_{m+1_i}, t_{m+1_i}^R), \dots, (s_{n_i}, t_{n_i}^R)\}, \qquad U_i^{p_k^i} = \left\{t_{j_i}^{p_k^i} = h_k^i(s_{j_i}, \Gamma_k, \lambda_k^i)\right\}_{j=m+1}^{n}$$

Postupkom parsanja $h_k^i(s_{j_i}, \Gamma_k, \lambda_k^i)$ za svaku od rečenica svakoga od testnih skupova U_i^{te} dobiveni su parovi rečenica i predloženih ovisnosnih stabala $U_i^{p_k^i}$. Ovisnosna stabla iz toga skupa potom su uspoređena s ovisnosnim stablima iz testnoga skupa U_i^{te} korištenjem ranije definiranih mjera za vrjednovanje točnosti ovisnosnoga parsanja. Preciznije, korištene su sljedeće mjere za vrjednovanje:

1. mjere točnosti LAS (označeno povezivanje), UAS (neoznačeno povezivanje) i LA (označavanje) za vrjednovanje ukupne točnosti pojedinih parsera,

2. mjera točnosti UAS(r) za vrjednovanje točnosti neoznačenoga povezivanja pojedinom sintaktičkom funkcijom $r \in R$ i
3. mjera $F_1(r)$, odnosno mjera preciznosti i odziva pri označavanju pojedinom sintaktičkom funkcijom $r \in R$.

Mjere za vrjednovanje primijenjene su, ovisno o smislenosti, pri vrjednovanju ukupne točnosti parsanja, ali i pri vrjednovanju određenih njezinih značajki s obzirom na ulazne podatke. Vrjednovana je točnost ovisnosnoga parsanja:

1. ovisno o duljini rečenice, odnosno o broju riječi koje rečenice sadrže,
2. na podskupovima definiranima duljinom rečenice, kako bi se utvrdili podskupovi točnosti prema duljini rečenica,
3. prema vrsti riječi kojoj je dodijeljena sintaktička funkcija,
4. prema duljini ovisnosne relacije, odnosno udaljenosti među ovisnim riječima,
5. prema smjeru ovisnosne relacije s obzirom na slijed riječi u rečenicama i
6. prema (ne)projektivnosti.

Kako je ranije opisano, neki od algoritama za parsanje izvedenih u sklopu odabranih sustava za ovisnosno parsanje mogu proizvesti samo projektivna ovisnosna stabla, a pritom koriste jezične modele izgrađene isključivo nad projektivnim strukturama te se za neprojektivno ovisnosno parsanje oslanjaju na postupak projektivizacije i deprojektivizacije (usp. Nivre i Nilsson 2005). Za testiranje te skupine algoritama, odnosno parsera korišten je sljedeći postupak:

1. sva ovisnosna stabla iz uzoraka za treniranje i testiranje (U_i^{tr}, U_i^{te}) iz skupa za testiranje ovisnosnih parsera $S = \{S_R\}_{R \in \{R_a, R_b\}}$ u predobradbi su pretvorena u projektivna ovisnosna stabla postupkom projektivizacije, dostupnim u sklopu sustava MaltParser;
2. treniranje i testiranje jezičnih modela, odnosno pseudo-projektivnih ovisnosnih parsera izvršeno je nad novonastalom projektivnom bankom ovisnosnih stabala (projektivnim HOBS-om);
3. projektivna ovisnosna stabla dobivena nekim projektivnim ovisnosnim parserom pretvorena su, ponovno korištenjem sustava MaltParser i tamo izvedenoga postupka deprojektivizacije, u neprojektivna ovisnosna stabla.

Vrjednovanje projektivnih ovisnosnih parsera izvršeno je isključivo nad neprojektivnim HOBS-om, odnosno nakon provedenoga postupka deprojektivizacije. Formalno, skupu za testiranje ovisnosnih parsera $S = \{S_R\}_{R\in\{R_a,R_b\}}$ pridružuje se skup $S^P = \{S_R^P\}_{R\in\{R_a,R_b\}}$ uzoraka (U_i^{tr}, U_i^{te}) za koje vrijedi da je svako ovisnosno stablo svake njihove rečenice projektivno. Korišten je postupak (de)projektivizacije temeljen na pristupu u kojemu se neprojektivne ovisnosne relacije pretvaraju u projektivne uspostavljanjem zamjenske ovisnosne relacije i dodjelom sintaktičke funkcije obogaćene metapodatcima za rekonstrukciju glave i putanje (en. *head+path scheme*, usp. Nivre i Nilsson 2005:102) budući da se u testiranju (usp. Nivre i Nilsson 2005:104-106) pokazala najboljim pristupom za pseudo-projektivno parsanje čeških tekstova. Tim je postupkom izmjenjena – promjenom glave ovisnosne relacije ili promjenom sintaktičke funkcije – ukupno 1,801 pojavnica iz HOBS-a (2.06%) u ukupno 761 od 3465 rečenica (21.96%). Ti postotci predstavljaju udio neprojektivnih ovisnosnih relacija, odnosno neprojektivnih ovisnosnih stabala u HOBS-u i usklađeni su s onima prikazanima u (Nivre i Nilsson 2005:104) za PDT i očekivanima s obzirom na sintaktičko ustrojstvo hrvatskoga jezika. Pseudo-projektivni parseri ovdje se, dakle, vrjednuju kao neprojektivni parseri.

Ranija vrjednovanja točnosti ovisnosnoga parsanja pokazala su kako je njegova ukupna točnost obrnuto proporcionalna broju korištenih sintaktičkih funkcija (usp. McDonald i Nivre 2007). Također, iz predstavljanja rezultata s natjecanja CoNLL 2007 i CoNLL 2007 može se zaključiti kako korištenje relativno maloga uzorka za treniranje (kakav je HOBS) zajedno s velikim skupom sintaktičkih funkcija (kakav je skup svih sintaktičkih funkcija iz HOBS-a) obično ishoduje ukupnu točnost ovisnosnoga parsanja koja nije usporediva s točnostima parsanja jezika s većim uzorcima za treniranje i/li manjim skupovima sintaktičkih funkcija. Primjerice, ovisnosno parsanje tekstova iz PDT-a – banke stabala koja je, prema podatcima s natjecanja CoNLL, bila preko 10 puta veća od HOBS-a po broju rečenica – provedeno je isključivo korištenjem skupa osnovnih sintaktičkih funkcija. Budući da je HOBS izgrađen upravo prema načelima izgradnje PDT-a, a također i s obzirom na razliku u veličini, i u ovome se eksperimentu koristio samo skup osnovnih sintaktičkih funkcija. Dakle, prema ranijoj notaciji, za ovaj eksperiment vrijedi $S = S_{R_b}$.

Prema (Nivre 2006:141), učinkovitost se vrjednovala mjerenjem triju značajki uporabe sustava za ovisnosno parsanje, odnosno pojedinih parsera. Te su značajke:

1. trajanje postupka treniranja, odnosno vrijeme izrade jezičnoga modela iz dostupnih skupova za treniranje modela,
2. trajanje postupka primjene modela, odnosno vrijeme potrebno za parsanje skupa za testiranje modela (pritom se može procijeniti i vrijeme parsanja jedne rečenice ili jedne pojavnice, s obzirom na statističke značajke skupova za testiranje) i
3. memorijski zahtjevi parsera, odnosno količina računalne memorije uporabljene za vrijeme trajanja postupka treniranja parsera i samoga parsanja.

Rezultati mjerenja vremenske učinkovitosti parsanja, odnosno trajanja pojedinih parsnih postupaka izraženi su u (mili)sekundama ili minutama, dok su memorijski zahtjevi izraženi u megabajtima radne memorije računalnoga sustava kojim je provedeno testiranje. Pri mjerenju učinkovitosti naglasak je stavljen na vrjednovanje postupka parsanja budući da se postupak izrade modela može smatrati jednokratnim u usporedbi s postupkom parsanja, posebno s gledišta moguće primjene ovisnosnih parsera u složenijim sustavima. Naglasak je također stavljen na vrjednovanje vremenske učinkovitosti budući da se ona – s obzirom na dostupnost radne memorije u današnjim računalnim sustavima – uglavnom smatra važnijom od prostorne složenosti. Sva su testiranja provedena korištenjem računala s procesorom Intel Core 2 Quad Q6600 (2.40 GHz, 8 MB cache, 1066 MHz FSB), 6 GB radne memorije (DDR2, 1066 MHz) i 64-bitnom inačicom operacijskoga sustava Windows 7 Professional.

Vrjednovanje točnosti tehnički je izvedeno korištenjem računalnoga paketa MaltEval (Nilsson i Nivre 2008), a vrjednovanje vremenske i memorijske učinkovitosti (usp. Nivre 2006:142) korištenjem standardnih naredaba dostupnih unutar korištenoga operacijskog sustava. Statistička značajnost razlika među pojedinim rezultatima utvrđena je korištenjem standardnih testova, a rezultati su uspoređeni s rezultatima natjecanja u ovisnosnome parsanju CoNLL 2006 i CoNLL 2007 za srodne jezike (češki, slovenski) i jezike s usporedivim veličinama banaka ovisnosnih stabala.

3.2.2.3 Rezultati

Tablica 3-6 prikazuje rezultate vrjednovanja ukupne točnosti parsanja hrvatskih tekstova iz HOBS-a ovisnosnim parserima temeljenima na grafovima i prijelazničkim ovisnosnim parserima prema mjerama vrjednovanja LAS (označeno povezivanje), UAS (neoznačeno povezivanje) i LA (dodjela oznaka). Parseri su u tablici svrstani u četiri skupine. To su, redom iz tablice, ranije opisani: projektivni prijelaznički parseri (MaltNe, MaltNs,

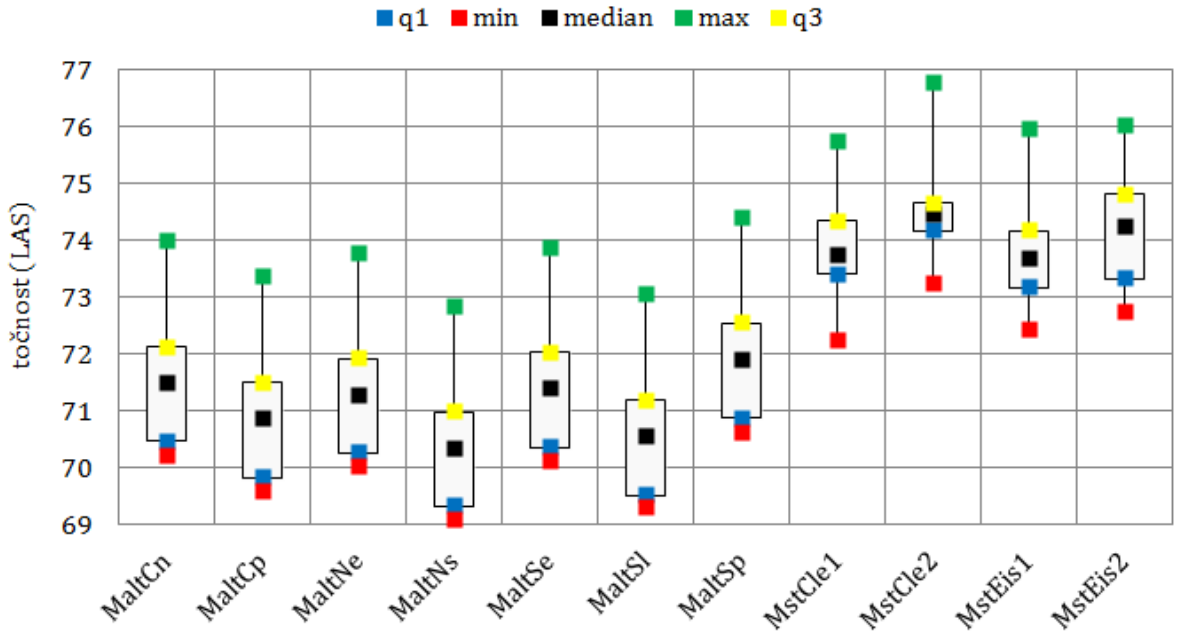
MaltCp, MaltSp), neprojektivni prijelazni̇čki parseri (MaltCn, MaltSe, MaltSl), projektivni parseri temeljeni na grafovima, prvoga i drugoga reda s obzirom na korišteni sintaktički model (MstEis1, MstEis2) i neprojektivni parseri temeljeni na grafovima, također prvoga i drugoga reda (MstCle1, MstCle2). Posebno su označeni parseri s postignutim najvećim točnostima prema mjerama za vrjednovanje u sve četiri skupine.

Tablica 3-6 Ukupna točnost ovisnosnoga parsanja

parser	LA	LAS	UAS
MaltNe	83.74 ± 0.46	71.29 ± 0.74	77.13 ± 0.71
MaltNs	83.16 ± 0.47	70.35 ± 0.73	76.44 ± 0.70
MaltCp	83.46 ± 0.48	70.87 ± 0.73	76.80 ± 0.69
MaltSp	**84.05 ± 0.44**	**71.91 ± 0.74**	**77.59 ± 0.73**
MaltCn	**83.88 ± 0.46**	**71.50 ± 0.74**	**77.30 ± 0.72**
MaltSe	83.75 ± 0.42	71.39 ± 0.73	77.23 ± 0.72
MaltSl	83.28 ± 0.48	70.56 ± 0.73	76.54 ± 0.71
MstEis1	85.57 ± 0.36	73.73 ± 0.65	80.92 ± 0.61
MstEis2	**85.64 ± 0.39**	**74.17 ± 0.64**	**81.27 ± 0.59**
MstCle1	85.76 ± 0.35	73.88 ± 0.58	80.99 ± 0.50
MstCle2	**85.87 ± 0.38**	**74.53 ± 0.57**	**81.69 ± 0.44**

Iz tablice 3-6 mogu se stoga izdvojiti parseri s najvećim točnostima s obzirom na korišteni jezični model, a također se može izdvojiti najtočniji parser neovisno o teorijskome okviru. Parser MstCle2 – parser izrađen sustavom MSTParser, neprojektivni ovisnosni parser temeljen na grafovima i jezičnome modelu s parovima ovisnosnih relacija i algoritmu za pronalaženje najvećega prostirućeg stabla Chu-Liu-Edmonds kao algoritmu za parsanje – postigao je najveću točnost pri parsanju tekstova iz HOBS-a prema svim odabranim mjerama za vrjednovanje ukupne točnosti. Najtočniji projektivni ovisnosni parser temeljen na grafovima je parser MstEis2 (Eisnerov algoritam, model uvjetovan jediničnim ovisnosnim relacijama), dok su najtočniji prijelazni̇čki parseri MaltCn (neprojektivni, Covingtonov algoritam) i MaltSp (projektivni, temeljen na stogu). Iz tablice je također primjetna usklađenost svih triju mjera za vrjednovanje točnosti budući da izdvojeni parseri bilježe

najviše vrijednosti u svojim skupinama s obzirom na sve tri mjere. Tablicu 3-6 prati slika 3-3 koja prikazuje točnost pojedinih parsera iz tablice prema mjeri LAS u obliku dijagrama s pravokutnicima (en. *box plot*). Iz dijagrama se mogu pročitati statističke značajke izvršenih mjerenja koje upućuju na statističku značajnost razlika među njima.



Slika 3-3 Ukupna točnost ovisnosnoga parsanja

Slika 3-3 ukazuje na značajnost razlike u točnosti parsera temeljenih na grafovima i točnosti prijelazničkih parsera. Testiranje statističke značajnosti pokazalo je da su razlike u točnostima svih parsera iz skupine temeljenih na grafovima u odnosu na prijelazničke parsere statistički značajne. S druge strane, točnosti među parserima unutar skupine prijelazničkih parsera nisu različite u statistički značajnoj mjeri, iako se prema ukupnoj točnosti iz te skupine izdvajaju parseri MaltSp (71.91 LAS) i MaltCn (71.50 LAS). Razlike u postignutoj točnosti parsanja nisu statistički značajne ni u skupini parsera temeljenih na grafovima, no u njoj se po postignutoj točnosti izdvajaju parseri MstCle2 (74.53 LAS) i MstEis2 (74.17 LAS), odnosno parseri s jezičnim modelima temeljenima na parovima ovisnosnih relacija.

Iz tablice 3-6 i slike 3-3 može se zaključiti kako je ovisnosno parsanje temeljeno na grafovima – prema opažanjima izvedenima iz uporabe HOBS-a i prema ovim postavkama eksperimenta – bolji teorijski okvir za parsanje hrvatskih tekstova od prijelazničkoga ovisnosnog parsanja s obzirom na mjere za vrjednovanje njegove ukupne točnosti. Međutim, vrijedi napomenuti kako ovdje nisu istražene sve moguće postavke pojedinih prijelazničkih

ovisnosnih parsera temeljenih na sustavu MaltParser budući da mnogobrojnost tih postavki čini da takav eksperiment opsegom nadilazi ovdje postavljene ciljeve. S druge strane, ovdje opažena (relativno velika) razlika u točnostima među ovim skupinama parsera i posljedična statistička značajnost te razlike, kao i razlike među njima opažene na natjecanjima CoNLL 2006 i 2007 za jezike srodne hrvatskomu, upućuju na valjanost zaključka o ovisnosnome parsanju temeljenom na grafovima kao najvjerojatnijemu najboljem izboru za ovisnosno parsanje hrvatskih tekstova.

Prema postignutim općim točnostima, za daljnji se prikaz dodatnih značajki točnosti odabiru ukupnom točnošću najbolji parseri iz pojedinih skupina. Tablica 3-7 prikazuje točnost parsanja izdvojenih parsera prema mjerama LA, LAS i UAS s obzirom na podatak o vrsti riječi pojavnice koja se povezuje u ovisnosnu relaciju. Parser s najvišim ocjenama na najvećem broju vrsta riječi – i najbrojnijim vrstama riječi prema razdiobi u HOBS-u – očekivano je ujedno i parser s najvećom ukupnom točnošću (MstCle2) budući da upravo pojavnice koje pripadaju tim vrstama riječi najviše doprinose ukupnoj točnosti.

Tablica 3-7 Točnost parsanja s obzirom na vrstu riječi

parser	mjera	A	C	M	N	P	R	S	V	Z
MaltCn	LA	91.32	77.88	70.21	82.08	79.71	79.73	95.43	77.55	88.66
	LAS	88.00	51.92	61.09	73.94	75.00	65.70	69.59	65.18	71.40
	UAS	89.88	56.37	73.60	83.29	85.75	73.08	70.41	70.50	73.83
MaltSp	LA	91.38	77.36	**70.68**	82.36	79.48	79.90	95.62	78.05	88.78
	LAS	88.02	51.99	**61.23**	74.45	75.12	65.87	69.79	66.13	71.79
	UAS	89.82	56.32	73.43	83.73	86.18	72.67	70.58	71.02	74.21
MstCle2	LA	**92.96**	87.94	68.19	81.84	**80.79**	**80.77**	98.57	**80.65**	**91.13**
	LAS	**89.96**	62.09	59.99	**74.50**	**76.08**	**68.19**	**74.72**	**71.81**	73.26
	UAS	**92.69**	64.31	76.39	**86.60**	**89.22**	**77.84**	75.35	**79.11**	75.40
MstEis2	LA	92.25	**88.12**	67.69	**81.85**	79.92	80.45	**98.58**	80.54	90.78
	LAS	88.73	**62.12**	61.00	74.33	74.32	66.81	74.63	71.54	**73.55**
	UAS	91.28	**64.34**	**77.59**	86.31	87.45	76.03	**75.36**	78.87	**75.61**

Točnost pojedinih parsera s gledišta pojedinih sintaktičkih funkcija, odnosno točnost povezivanja pojavnica u ovisnosne relacije s pripadajućim funkcijama (LAS) i točnost samoga povezivanja za fiksiranu sintaktičku funkciju (UAS) prikazana je u tablici 3-8. Iz nje je primjetno kako su u parsanju obavijesno najvažnijih kategorija (Pred, Sb, Obj) najbolje rezultate postigla dva parsera temeljena na grafovima, dok su u dodjeli najbrojnije sintaktičke funkcije iz HOBS-a (Atr) bolje rezultate postigli prijelazнички parseri. Ti su rezultati u potpunosti usklađeni s ranijim istraživanjima značajki pogrešaka ovih dvaju razreda parsera (usp. McDonald i Nivre 2007) u kojima je utvrđeno kako prijelazнички parseri u pravilu bolje parsaju sintaktičkim funkcijama koje su vrlo česte u banci stabala i kojima se najčešće vežu susjedne riječi, a oba su svojstva ranije uočena za sintaktičku funkciju atributa (Atr) u HOBS-u. S druge strane, parseri temeljeni na grafovima u pravilu bolje povezuju osnovne elemente rečeničnoga ustroja, odnosno predikate sa subjektima i objektima.

Tablica 3-8 Točnost parsera s obzirom na sintaktičku funkciju

parser	mjera	Adv	Apos	Atr	AuxC	AuxP	Coord	Obj	Pnom	Pred	Sb
MaltCn	LAS	70.67	**45.88**	83.77	74.36	71.99	46.28	67.40	**66.55**	36.45	69.14
	UAS	83.16	**50.45**	88.40	75.81	72.46	46.92	79.94	70.33	43.82	76.73
MaltSp	LAS	**71.31**	44.73	**83.98**	**75.68**	72.08	46.96	68.15	66.35	37.33	70.12
	UAS	83.41	48.50	**88.59**	**77.10**	72.53	47.79	80.08	70.43	44.33	77.53
MstCle2	LAS	69.01	37.40	81.80	71.94	**74.35**	**56.49**	**69.38**	65.18	68.10	72.51
	UAS	**85.58**	43.48	87.78	74.07	**75.06**	**57.73**	**84.64**	**77.52**	75.36	**81.67**
MstEis2	LAS	68.38	39.34	81.46	73.21	74.15	55.05	68.29	62.47	**69.09**	**72.63**
	UAS	84.67	44.23	87.44	74.86	74.90	56.41	83.95	74.38	**76.06**	81.34

Upravo je za uspostavu ovisnosnih relacija sa sintaktičkom funkcijom atributa, zbog brojnosti i svojstvene manje duljine, zabilježena najviša točnost prema mjerama LAS i UAS. Najniža je točnost u sva četiri parsera zabilježena za sintaktičku funkciju koja označava apoziciju (Apos), što potvrđuje raniju tvrdnju o mogućem nevaljanom ručnom označavanju atributa (prekobrojnih) i apozicija (malobrojnih) u HOBS-u. Osim apozicija, koordinacijski veznici (Coord) također su uzrokovali probleme svim parserima, što upućuje na sustavni problem u parsanju (nezavisno-)složenih rečenica u odnosu na parsanje surečenica. Svojstva

pogrešaka pojedinih parsera na pojedinim elementima rečeničnoga ustroja valjalo bi stoga dodatno promotriti s jezikoslovnoga gledišta, no takva raščlamba prelazi zadani doseg ovoga istraživanja i postavljenoga formalnog uređaja za vrjednovanje parsanja.

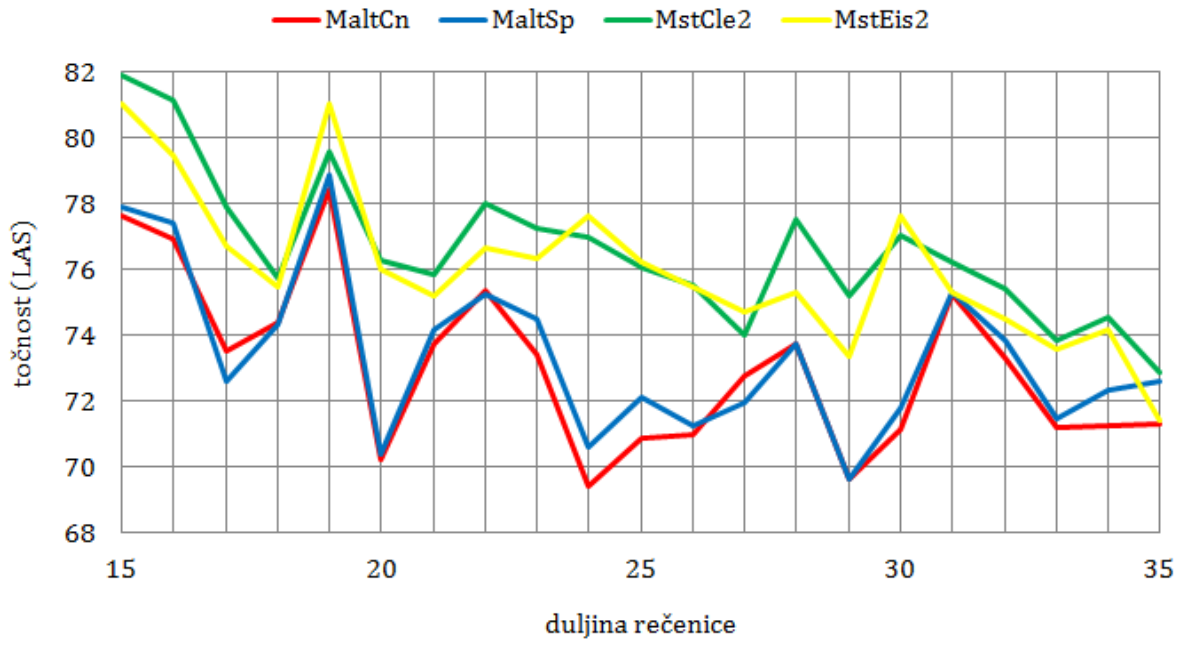
Tablica 3-9 dopunjuje tablicu 3-8 razmatranjem preciznosti i odziva dodjele sintaktičkih funkcija pojedinim pojavnicama u testnim uzorcima. Naime, u skladu s ranije izloženom formalnom definicijom ovisnosnoga parsanja, sintaktičke se funkcije dodjeljuju pojedinim pojavnicama – konvencijom se radi o dependentima, a ne glavama relacija – dok se same relacije potom uspostavljaju dodatnom oznakom. Dakle, ovisnosna je relacija u potpunosti označena s pomoću dva podatka: sintaktičke funkcije dependenta i identifikatora glave. Prvim se podatkom pritom predstavlja slijedno označavanje pojavnica u tekstu, poput označavanja vrsta riječi ili morfosintaktičkoga označavanja, dok drugi podatak definira ovisnosno stablo. Stoga se sama dodjela sintaktičkih funkcija može promatrati upravo kao dodjela podataka o vrstama riječi, pa se za nju može mjeriti točnost (LA), ali i preciznost i odziv (P, O) s obzirom na dodjelu pojedine sintaktičke funkcije. Prema ranijoj definciji, preciznost u tome slučaju predstavlja omjer pojavnica kojima je parserom točno dodijeljena neka sintaktička funkcija i svih pojavnica kojima je dodijeljena ta sintaktička funkcija, a odziv predstavlja omjer točno označenih pojavnica prema svim pojavnicama kojima je ta sintaktička funkcija dodijeljena u referentnome uzorku. Može se reći kako tablica 3-9 predstavlja vrjednovanje ovisnosnoga parsanja metodom vrjednovanja svojstvenom za morfosintaktičko označavanje.

Tablica 3-9 Preciznost i odziv s obzirom na dodjelu sintaktičke funkcije (LA)

parser	mjera	Adv	Apos	Atr	AuxC	AuxP	Coord	Obj	Pnom	Pred	Sb
MaltCn	P	78.72	58.94	90.17	**94.04**	98.69	88.01	75.63	**69.95**	50.68	81.76
	O	77.24	36.63	91.48	83.90	94.50	67.14	72.55	49.35	82.69	83.35
MaltSp	P	**79.09**	58.70	**90.20**	93.67	**98.79**	87.96	**76.31**	69.46	50.89	82.32
	O	**77.50**	37.78	**91.76**	84.15	94.67	67.28	**73.57**	47.06	82.76	83.47
MstCle2	P	75.63	**64.29**	88.20	91.51	98.09	**90.22**	76.24	69.18	79.66	82.98
	O	76.04	**51.80**	91.69	88.81	**97.83**	83.74	71.37	54.36	**84.89**	**86.15**
MstEis2	P	75.44	60.07	87.97	92.31	97.96	89.37	75.71	66.29	**80.26**	**83.53**
	O	75.62	47.35	91.58	**89.44**	97.77	**84.87**	71.03	**57.10**	83.62	85.83

Rezultati u tablici 3-9 uglavnom slijede raniji obrazac točnosti parsanja (LAS, UAS) s obzirom na sintaktičku funkciju, prikazan tablicom 3-8. Vrijedi primijetiti kako su postignute visoke točnosti označavanja za većinu sintaktičkih funkcija, primjerice, predikate je parser MstCle2 – koji i u ovoj razdiobi točnosti bilježi najbolje prosječne rezultate – prepoznavao s F_1-mjerom od oko 82.19%, subjekte s F_1-mjerom od oko 84.54, a objekte s nešto nižom F_1-mjerom od oko 73.72%. S obzirom na veličinu HOBS-a, odnosno uzoraka za treniranje i testiranje i na broj sintaktičkih funkcija, može se reći kako je ukupna točnost dodjele sintaktičkih funkcija usporediva s očekivanom točnošću MSD-označavanja uz isti broj MSD-oznaka i veličinu uzoraka pri korištenju nekoga uobičajenog teorijskog okvira za MSD-označavanje, poput skrivenih Markovljevih modela.

Slika 3-4 predstavlja opaženu funkcijsku ovisnost ukupne točnosti ovisnosnoga parsanja prema mjeri LAS s obzirom na duljinu rečenice, odnosno broj pojavnica u rečenici, za četiri izdvojena parsera.



Slika 3-4 Točnost parsanja s obzirom na duljinu rečenice

Tu funkcijsku ovisnost treba promatrati i s obzirom na razdiobu rečenica prema broju riječi u HOBS-u koja je dana na slici 3-1. Tamo je pokazano kako najveći broj rečenica sadrži između 15 i 35 pojavnica, pa je razdioba točnosti na slici 3-4 dana upravo za taj podskup rečenica iz HOBS-a. Ranija istraživanja razlika između prijelazničkih parsera i parsera temeljenih na grafovima (usp. McDonald i Nivre 2007) pokazala su da temeljna razlika u tim

teorijskim okvirima – prema kojima parsanje temeljeno na grafovima predstavlja neusmjereni pristup parsanju s algoritmom za traženje najboljega parsnog stabla na razini čitave rečenice, odnosno globalnome dosegu s obzirom na ulaznu rečenicu, a prijelazničko parsanje usmjereni pristup s lokalnim, odnosno fraznim dosegom pri pronalaženju ovisnosnih relacija – obično uzrokuje manji pad točnosti parsera temeljenih na grafovima s rastom duljine rečenice u odnosu na pad točnosti prijelazničkih parsera. Taj se raniji rezultat u ovome istraživanju potvrđuje djelomično budući da je opaženi pad točnosti usporediv za obje skupine parsera, ali nešto manje značajan za prijelazničke parsere[100]. Međutim, u rasponu duljina rečenica od 20 do 25 pojavnica ipak se može opaziti nešto mjerljiviji pad točnosti kod prijelazničkih parsera, a taj se raspon može smatrati najznačajnijim s obzirom na broj rečenica koje sadrži prema slici 3-1. Vrijedi napomenuti kako je ranije istraživanje iz (McDonald i Nivre 2007) provedeno korištenjem banaka ovisnosnih stabala koje su znatno veće od HOBS-a, pa su i tamo opažene razlike statistički značajnije. Na slici 3-4 jasno se vidi pripadnost parsera pojedinim skupinama s obzirom na karakterističnu oblik funkcije.

Tablica 3-10 prikazuje točnost – izraženu preciznošću i odzivom – dodjele smjera ovisnosne relacije unutar rečenice za pojedine parsere. Prikazuje se, dakle, koliko su točni pojedini parseri kad povezuju dependente s glavama kad se glave nalaze lijevo i desno od dependenata i kad su pojavnice ovisne o fiktivnim korijenskim čvorovima rečenica.

Tablica 3-10 Točnost povezivanja s obzirom na smjer ovisnosti unutar rečenice

parser	mjera	lijevo	desno	korijen
MaltCn	preciznost	90.16	88.71	60.47
	odziv	86.22	86.23	86.17
MaltSp	preciznost	90.34	88.67	60.56
	odziv	86.27	86.35	86.19
MstCle2	preciznost	**91.28**	**90.59**	87.75
	odziv	**91.43**	**90.45**	87.51
MstEis2	preciznost	91.16	90.10	**87.99**
	odziv	91.04	90.17	**88.32**

[100] Linearne aproksimacije funkcijskih ovisnosti točnosti prikazanih na slici 3-4 padajuće su linearne funkcije s koeficijentima smjera -0.277 (MstCle2) i -0.199 (MaltSp).

Iz tablice je primjetno kako svi odabrani parseri nešto točnije povezuju pojavnice s glavama koje su u rečenici lijevo od njih i kako parseri temeljeni na grafovima značajno preciznije povezuju pojavnice s korijenskim čvorovima, iako su im odzivi za te relacije usporedivi s odzivima prijelazničkih parsera. Problem točnosti povezivanja s korijenskim čvorom karakterističan je za prijelazničke parsere budući da se kod njih te veze najčešće uspostavljaju naknadno, odnosno povezivanjem svih nepovezanih pojavnica s korijenskim čvorom po završetku postupka parsanja pripadajućim parsnim algoritmom, uz dodjelu neke ranije odabrane sintaktičke funkcije (u ovome slučaju, Pred). Opaženi pad preciznosti stoga je uzrokovan povezivanjem prevelikoga broja pojavnica s korijenskim čvorovima, što također objašnjava zadržavanje visokoga odziva.

U tablici 3-11 dane su točnosti (preciznosti i odzivi) povezivanja pojavnica u ovisnosne relacije s obzirom na udaljenost tih pojavnica unutar rečenice, mjerenu brojem pojavnica koje se nalaze između njih. Točnost povezivanja stoga je identična onoj iz prethodne tablice, pa se ovdje o njoj ne raspravlja dodatno. Iz tablice se vidi kako je pad točnosti parsera temeljenih na grafovima s rastom udaljenosti među pojavnicama manje značajan od pada točnosti prijelazničkih parsera[101].

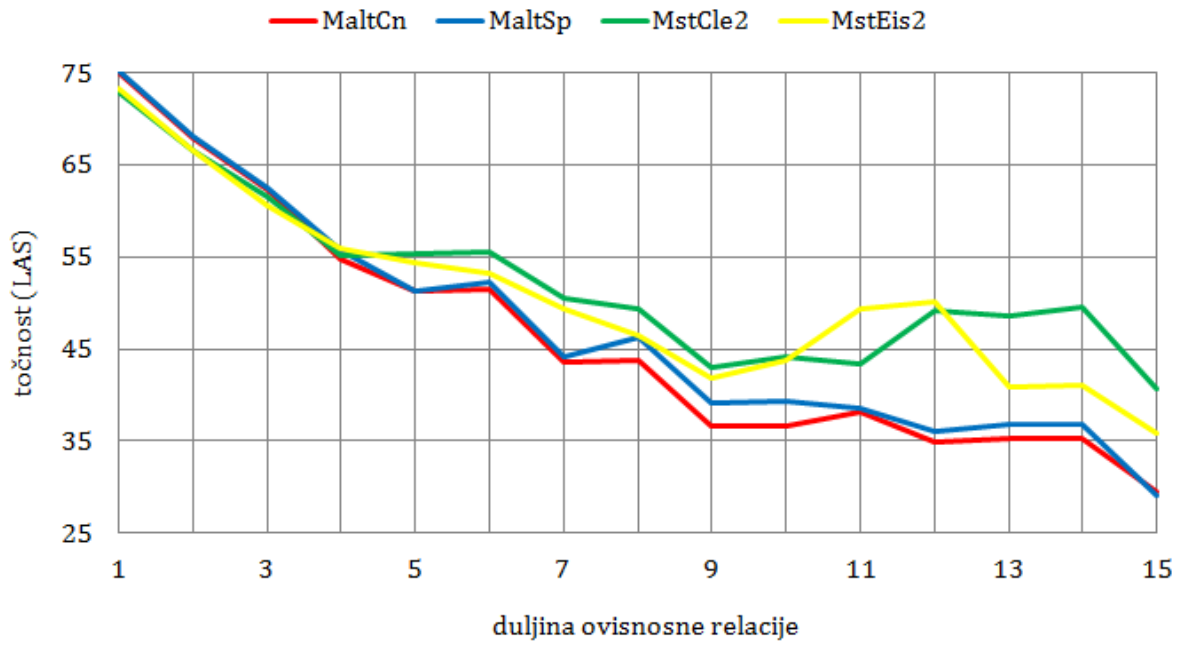
Tablica 3-11 Točnost povezivanja s obzirom na udaljenost među pojavnicama

parser	mjera	1	2	3-6	7-...	korijen
MaltCn	preciznost	90.80	83.14	78.30	60.15	60.47
	odziv	90.40	80.24	70.44	55.33	86.17
MaltSp	preciznost	**90.98**	83.08	78.53	60.77	60.56
	odziv	90.46	80.33	70.57	56.24	86.19
MstEis2	preciznost	90.60	**83.26**	79.14	69.02	**87.99**
	odziv	93.03	83.50	75.87	**63.87**	**88.32**
MstCle2	preciznost	90.16	83.07	**80.15**	**72.89**	87.75
	odziv	**93.92**	**85.40**	**76.13**	58.86	87.51

[101] Radi se o padu od oko 90% do oko 70% preciznosti i odziva za parsere temeljene na grafovima i padu od oko 90% do oko 60% preciznosti i odziva za prijelazničke parsere.

Ta je razlika između parsera temeljenih na grafovima i prijelazničkih parsera također uzrokovana razlikom u pripadajućim parsnim algoritmima, odnosno proizlazi iz razlike između neusmjerenoga i usmjerenoga pristupa parsanju općenito, a uočena je i u istraživanjima s većim bankama ovisnosnih stabala (usp. McDonald i Nivre 2007).

Tablica 3-11 dopunjena je slikom 3-5 koja prikazuje ukupnu točnost parsanja prema mjeri LAS s obzirom na udaljenost među povezanim pojavnicama. Izdvojene su samo ovisnosne relacije s udaljenostima od 1 do 15 budući da preostale udaljenosti (veće od 15 pojavnica) nisu dovoljno učestale, pa bi narušile značajnost opažanja. Slika 3-5 pokazuje ono raslojavanje parsera prema teorijskome okviru koje, moguće zbog veličine HOBS-a, nije u dovoljnoj mjeri ilustrirano slikom 3-4. Dakle, na slici 3-5 jasno se vidi značajniji pad točnosti[102] prijelazničkih parsera u usporedbi s parserima temeljenima na grafovima kod povezivanja udaljenijih pojavnica u ovisnosne odnose.



Slika 3-5 Točnost parsanja s obzirom na udaljenost među pojavnicama

S obzirom na prikazane rezultate i statističku značajnost u njima opaženih različitosti i s obzirom na svojstva hrvatskih tekstova sadržanih u HOBS-u, ovdje se zadržava zaključak iznesen u raspravi o točnostima pojedinih parsera prema općim mjerama za vrjednovanje. Dakle, ovdje se smatra kako opaženi rezultati upućuju na ovisnosno parsanje temeljeno na

[102] Pripadajući koeficijenti smjera linearnih aproksimacija prikazanih funkcijskih ovisnosti ovdje su obrnuti u odnosu na one za sliku 3-4, oko -1.722 (MstCle2) i oko -2.813 (MaltSp).

grafovima kao najbolji izbor za ovisnosno parsanje hrvatskih tekstova, posebno u usporedbi s prijelazničkim ovisnosnim parsanjem. Vrjednovanje učinkovitosti prikazanih parsera dano je u opisu rezultata eksperimenta prikazanoga u idućemu poglavlju.

3.2.3 Jedan model ovisnosnoga parsanja hrvatskih tekstova

Opisani rezultati eksperimenta s ovisnosnim parsanjem hrvatskih tekstova iz HOBS-a postojećim parserima temeljenima na grafovima i prijelazničkim parserima uporabljeni su u ovome istraživanju kao polazište za izradu hibridnoga ovisnosnog parsera usmjerenoga isključivo parsanju hrvatskih tekstova. Izrada takvoga parsera usmjerena je podizanju ukupne točnosti parsanja hrvatskih tekstova, s posebnim naglaskom na obavijesno najvažnijim elementima rečeničnoga ustroja – predikatima, subjektima i objektima. U ovome poglavlju izložen je model toga parsera, njegovo vrjednovanje na HOBS-u i usporedba s najboljim parserima iz dvaju izdvojenih teorijskih okvira ovisnosnoga parsanja, prikazanima u prethodnome eksperimentu.

3.2.3.1 Model i izvedba parsera

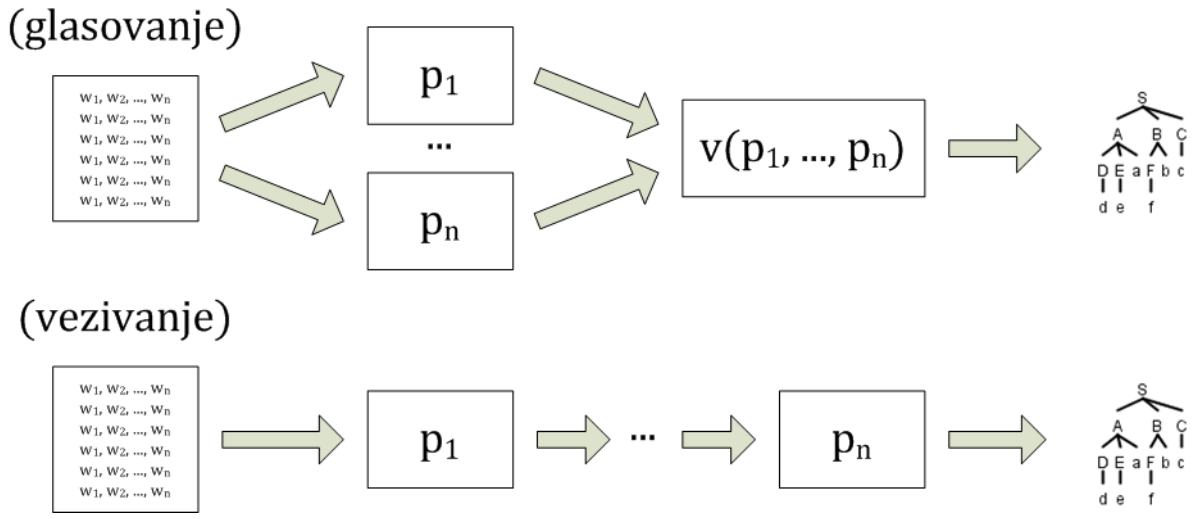
Pristupi poboljšavanju točnosti ovisnosnoga parsanja temeljenoga na podatcima načelno se mogu podijeliti u dvije glavne skupine pristupa:

1. pristupe temeljene na slaganju pojedinih ovisnosnih parsera i
2. pristupe temeljene na povezivanju ovisnosnih parsera s jezičnim resursima i/li alatima za ciljani jezik.

Slaganje pojedinih ovisnosnih parsera u složene sustave za ovisnosno parsanje načelno se dalje dijeli na glasovanje (en. *voting*) i vezivanje (en. *stacking*). Pristupi su ilustrirani slikom 3-6. Kod glasovanja ovisnosnih parsera prikupljaju se parsna stabla svih ovisnosnih parsera prisutnih u složenome sustavu i iz njih se izgrađuje novo ovisnosno stablo. Ovaj se pristup naziva glasovanje[103] jer se pojedina ovisnosna stabla razmatraju kao skupovi ovisnosnih relacija, pa se promatraju preklapanja i razlike u skupovima ovisnosnih relacija koje su pojedini parseri predložili za povezivanje pojavnica u ulaznim rečenicama. Novo se ovisnosno stablo, dakle, izrađuje glasovanjem o ovisnosnim relacijama, na način da se u njega ugrađuju ovisnosne relacije koje su prisutne u većini pojedinih parsnih stabala. Primjerice, neka je u sustavu za parsanje glasovanjem pet različitih ovisnosnih parsera. Svaki od njih

[103] Točnije, jednostavno većinsko glasovanje (en. *simple majority vote*).

parsa neku ulaznu rečenicu. Na izlazu se dobije pet načelno različitih parsnih stabala, na način da je svakoj pojavnici iz ulazne rečenice svaki parser dodijelio sintaktičku funkciju i identifikator glave. Jednostavno glasovanje može se izvesti tako da se za svaku pojavnicu prebrojavaju – odvojeno ili združeno – dodijeljene sintaktičke funkcije i identifikatori glava te se u ishodišno parsno stablo biraju one funkcije i identifikatori (ili parovi istih) koji su potvrđeni najvećim brojem parsera.



Slika 3-6 Slaganje ovisnosnih parsera

Primjerice, ako je nekoj pojavnici triput dodijeljena sintaktička funkcija apozicije, a dvaput atributa, pojavnica će u ishodišnome ovisnosnom stablu nositi sintaktičku funkciju apozicije. Ovaj ilustrativni pristup moguć je, ipak, samo u slijednom označavanju pojavnica, poput lematizacije, morfosintaktičkoga označavanja ili dodjele sintaktičkih funkcija (u smislu mjere LA), no ne i kod izgradnje stablastih struktura poput ovisnosnih stabala. Naime, može se dogoditi da se većinskim glasovanjem o sintaktičkim funkcijama, identifikatorima glava ili uređenim parovima tih značajki izgradi struktura koja nije ovisnosno stablo, primjerice, na način da ne udovoljava svojstvu acikličnosti ili jednoga korijenskog čvora. U pristupima s glasovanjem, prema (Sagae i Lavie 2006) i (Hall i dr. 2007), najčešće se nad skupom svih ovisnosnih relacija svih parsera uključenih u sustav za parsanje glasovanjem većinsko glasovanje izvodi u obliku dodatnoga pokretanja nekoga od algoritama za izgradnju parsnoga stabla (poput algoritma Chu-Liu-Edmonds za pronalaženje najvećega prostirućeg stabla) nad tim skupom. Glasovanje se pritom može dodatno prilagoditi dodjelom težina pojedinim parserima za pojedine razrede ovisnosnih relacija, odnosno dinamičkim uvođenjem nejednakih vrijednosti glasova u većinsko glasovanje prema nekom izdvojenom skupu za

procjenu težina. Upravo taj pristup prikazan je u (Sagae i Lavie 2006) i ugrađen u sustav za glasovanje MaltBlender[104], prikazan u (Hall i dr. 2007), te su u oba istraživanja zabilježena statistički značajna povećanja ukupnih točnosti ovisnosnoga parsanja. Sustav iz (Hall i dr. 2007) također je bio najbolji sustav na natjecanju CoNLL 2007.

Pristup s vezivanjem parsera temelji se na ideji da se ovisnosna stabla ulaznih rečenica dobivena jednim parserom mogu upotrijebiti kao značajke za izgradnju jezičnoga modela drugoga parsera i najčešće se primijenjuju za raznorodne parsere, poput povezivanja parsera temeljenih na grafovima i prijelazničkih parsera. Jedan takav eksperiment s povezivanjem MaltParsera i MSTParsera prikazan je u (Nivre i McDonald 2008). U tome su istraživanju razmotrena tri modela – jedan temeljen na povezivanju značajki, drugi na izgradnji modela za prijelazničko parsanje iz rezultata parsanja modelom temeljenim na grafovima i treći na izgradnji modela za parsanje temeljeno na grafovima iz rezultata prijelazničkoga parsanja. Najbolje rezultate u tome istraživanju postigao je upravo treći model, onaj u kojemu je MSTParser uz uobičajene značajke koristio i značajke izvedene iz ovisnosnih stabala dobivenih MaltParserom. I tim su pristupom postignuta statistički značajna povećanja ukupne točnosti ovisnosnoga parsanja.

Pristupi temeljeni na povezivanju generičke metode ovisnosnoga parsanja temeljenoga na podatcima s dostupnim alatima i resursima za ciljani prirodni jezik najčešće su usmjereni znatnijem povećanju točnosti parsanja tekstova toga specifičnog jezika, uz posljedični gubitak općenitosti polazišne metode. U tim pristupima, generički ovisnosni parseri – i drugi alati za obradbu tekstova nekoga jezika na drugim razinama jezičnoga opisa – dopunjuju se dodatnim algoritamskim postupcima, koji koriste specifično jezično znanje sadržano u dostupnim jezičnim resursima za ciljani jezik ili u predobradnim postupcima, kako bi se povećala točnost obradbe tim parserom ili drugim alatom za jezičnu obradbu. Tim pristupima se načelno dobivaju hibridni alati koji su vezani uz onaj jezik ili skupinu jezika za koje su dostupni jezični resursi i/li alati o kojima taj hibridni alat ovisi, ali za taj jezik pružaju razinu točnosti koju generički alati ne mogu ponuditi. Takvi alati nazivaju se hibridnima jer za polazišnu točku najčešće uzimaju neku generičku metodu temeljenu na podatcima, a putem jezično-specifičnih resursa i/li alata implicitno u tu metodu uvode komponentu temeljenu na pravilima jezičnoga opisa, stvarajući time alat temeljen na međuovisnosti svojstava modela temeljenih na pravilima i temeljenih na podatcima.

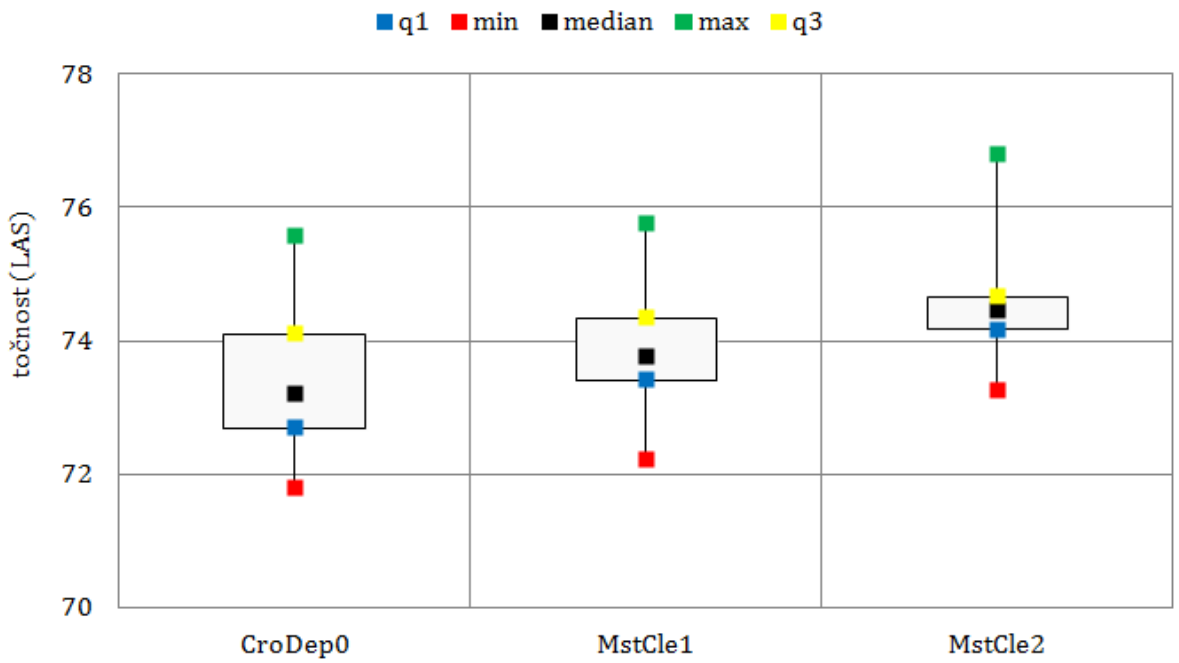
[104] Vidjeti i URL http://w3.msi.vxu.se/users/jni/blend/ (2012-04-11).

S obzirom na vrjednovanje točnosti postojećih ovisnosnih parsera na HOBS-u, posebno u usporedbi s vrjednovanjem u sklopu natjecanja CoNLL 2006 i 2007 za srodne jezike[105], u ovome je istraživanju odabran upravo smjer koji uključuje hibridizaciju nekoga od postojećih pristupa ovisnosnomu parsanju s dostupnim jezičnim resursima za hrvatski jezik, s ciljem – poželjno što značajnijega – povećanja ukupne točnosti parsanja hrvatskih tekstova. Pritom je bilo potrebno odabrati dobar polazišni pristup ovisnosnomu parsanju temeljen na podatcima, razmotriti poželjna svojstva dostupnih jezičnih resursa i alata za opis i obradbu hrvatskih tekstova s gledišta ovisnosnoga parsanja te odabrati resurse i/li alate i pristup njihovu povezivanju s polazišnim parserom.

Kod izbora polazišnoga pristupa parsanju za daljnju hibridizaciju razmotreni su upravo rezultati ranije prikazanoga eksperimenta s prijelazničkim parsanjem i parsanjem temeljenim na grafovima, i to s gledišta ukupne točnosti, izvedivosti i jednostavnosti odabranoga pristupa s obzirom na potrebu za njegovom kasnijom prilagodbom na specifičnosti odabranih vanjskih jezično-specifičnih modula. Rezultati eksperimenta s parsanjem iz prethodnoga poglavlja jasno su ukazali na modele temeljene na grafovima kao najbolji od ponuđenih izbora teorijskoga okvira za parsanje. Za računalnu izvedbu odabran je stoga neprojektivni i relacijski uvjetovani pristup ovisnosnomu parsanju temeljenom na grafovima, odnosno pristup čija je izvedba ranije vrjednovana u vidu parsera MstCle1 iz paketa MSTParser. Taj je teorijski okvir odabran – unatoč činjenici da parser MstCle1 nije bio najbolji od ovisnosnih parsera temeljenih na grafovima s obzirom na prosječnu točnost parsanja – iz dva razloga. Prvenstveno, statističkim testiranjem u prethodnome eksperimentu uočeno je kako razlike među pojedinim parserima temeljenima na grafovima nisu statistički značajne, pa se ti parseri do neke mjere mogu smatrati jednakovrijednima. Razlika od 0.65% LAS između parsera MstCle1 i MstCle2 može se stoga smatrati zanemarivom, posebno s gledišta primjene parsera u složenijim sustavima za obradbu jezika. Također, relacijski uvjetovani (en. *arc factored*) jezični model – ili model prvoga reda s obzirom na onaj s parovima relacija koji se naziva modelom drugoga reda i vrjednovan je u parseru MstCle2 – i pripadajući algoritam za Chu-Liu-Edmonds za parsanje pronalaženjem najvećega prostirućeg stabla izvedbeno su takvi da omogućavaju izravnije izmjene nego jezični model drugoga reda i pripadajući algoritam za parsanje razmatranjem parova ovisnosnih relacija. Odabrani model, koji je detaljno izložen u ranijem poglavlju s osnovnim teorijskim postavkama relacijski uvjetovanoga ovisnosnog

[105] Točnost parsanja hrvatskih tekstova iz HOBS-a usporediva je s točnošću postignutom za slovenski jezik, unatoč trostruko većoj banci stabala, a statistički je značajno lošija od točnosti postignute za češki jezik.

parsanja temeljenoga na grafovima i koji je prisutan u programskome paketu MSTParser, ovdje je za potrebe istraživanja i za buduće eksperimente s parsanjem hrvatskih tekstova nanovo izveden po uzoru na izvedbu iz MSTParsera. Tako dobiveni prototipni parser hrvatskih tekstova ovdje se naziva CroDep0. Izveden je u programskome jeziku Java te se za vrijeme provođenja eksperimenta s hibridizacijom nalazio u ranoj razvojnoj fazi. Također, tijekom provođenja ovoga istraživanja nije pružao korisniku nikakve mogućnosti prilagodbe značajki izrade jezičnoga modela i njegove uporabe – njime je bilo predviđeno da korisnik u postupku treniranja navede datoteku s uzorkom za treniranje i naziv rezultirajućega jezičnog modela, a u postupku treniranja samo navede datoteku s uzorkom za testiranje i naziv izlazne datoteke s rezultatima. Budući da je ishodište ovoga dijela istraživanja bilo u hibridizaciji, odnosno povezivanju parsera CroDep0 s jezičnim resursima i/li alatima za obradbu hrvatskih tekstova, on nije podvrgnut opširnomu testiranju prema ranijemu testnom planu. Međutim, provedeno je ipak vrjednovanje njegove ukupne točnosti prema mjeri LAS kojim je zabilježena ukupna točnost parsanja od oko 73.27% LAS, odnosno 0.61% manje od sustava MstCle1 i izvan dosega statističke značajnosti. Rezultat je ocrtan i slikom 3-7 koja prikazuje statističke značajke ukupne točnosti sustava CroDep0 u odnosu na srodne sustave MstCle1 i MstCle2 prema njihovu ranijem vrjednovanju. Detaljniji opis parsera CroDep0 izložen je dalje u tekstu uz opis njegove hibridizacije.

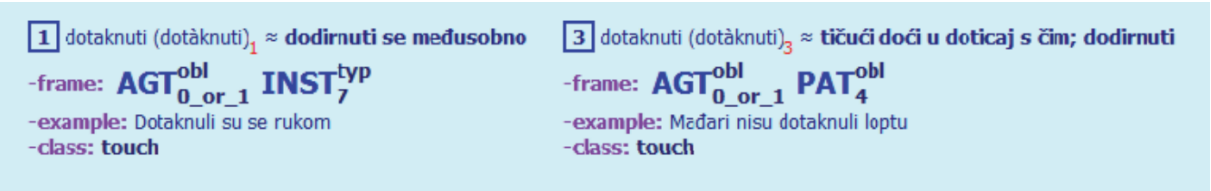


Slika 3-7 Ukupna točnost parsera CroDep0, MstCle1 i MstCle2

Za hibridizaciju parsera CroDep0 odabran je, slijedeći osnovnu ideju izloženu u (Zeman 2002), valencijski rječnik hrvatskih glagola – CROVALLEX (usp. Mikelić Preradović 2008, Mikelić Preradović i dr. 2009), izgrađen po uzoru na valencijski rječnik čeških glagola VALLEX (Lopatková i dr. 2006)[106]. Naime, prema ranije izloženim osnovnim postavkama ovisnosne sintakse i sintaktičkoga, odnosno rečeničnoga ustrojstva uopće, rečeničnu strukturu stvara svojstvo nekoga glagola da drugim elementima rečeničnoga ustroja otvara u rečenici mjesto. Prema (Tesnière 1959), svojstvo pojedinoga glagola da otvara određeni broj mjesta za određene razrede elemenata rečeničnoga ustroja naziva se *valentnost* ili *valencija glagola* (en. *verb valency*, *verb valence*). Pojednostavljeno, valentnost nekoga glagola može se promatrati u obliku dvije značajke:

1. brojčane vrijednosti koja predstavlja broj elemenata rečeničnoga ustroja kojima taj glagol otvara mjesto u rečenici i
2. pripadajućega formalnog opisa zahtijevanih dodatnih svojstava kojima ti elementi rečeničnoga ustroja moraju udovoljiti kako bi bili uvedeni u rečenicu.

Značajke su oslikane primjerom 3-2 koji predstavlja dva izdvojena valencijska okvira iz CROVALLEX-a za glagol *dotaknuti*, preuzeta iz (Agić i dr. 2010). U njemu se vidi kako u prvome valencijskom okviru glagol otvara u rečenici mjesto za dva elementa – vršitelja radnje (agens, AGT) i predmet izvršenja radnje (instrument, INST) – dok u drugome okviru otvara mjesto vršitelju radnje i trpitelju radnje (pacijens, PAT). Dodatni zahtjevi za elemente kojima je otvoreno mjesto dani su uz identifikatore pojedinih elemenata (tzv. *funktore*). Tako se, primjerice, od predmeta izvršenja radnje (INST) u prvome okviru zahtijeva da bude u instrumentalu (7), a od trpitelja radnje (PAT) u drugome da bude u akuzativu (4).



Primjer 3-2 Neki valencijski okviri glagola *dotaknuti* u CROVALLEX-u

Valentnost glagola u hrvatskome jeziku detaljno se razmatra u (Mikelić Preradović 2008) s gledišta izrade valencijskoga rječnika i posebno u (Šojat 2008) s općega gledišta sintaktičke i semantičke strukture hrvatskih tekstova, dok se za potrebe ovoga istraživanja

[106] Vidjeti i URL projekta http://ufal.mff.cuni.cz/vallex/ (2012-04-12).

razmatraju samo praktične implikacije postojanja pojave valentnosti glagola i dostupnosti jezičnoga resursa koji je opisuje za glagole hrvatskoga jezika. Spomenuti pristup hibridizaciji temeljen na osnovnim postavkama istraživanja iz (Zeman 2002) usko je vezan uz valenciju glagola i dostupnost neke vrste valencijskoga rječnika parsanoga prirodnog jezika. Naime, može se pretpostaviti sljedeće.

1. Valencija glagola implicitno je prisutna za sve glagole u nekoj banci ovisnosnih stabala budući da ovisnosna stabla pokazuju konkretna ostvarenja[107] pojedinih valencijskih okvira. Iz ovisnosnoga se stabla za neki glagol-predikat može pročitati koliko je mjesta otvorio, za koje sintaktičke funkcije i za koje vrste riječi i dodatne morfosintaktičke značajke. Štoviše, može se na temelju čestote pojedinih otvaranja i pojedinih ograničenja za pojedine glagole (polu-)automatski iz banke stabala crpiti jezični resurs poput valencijskoga rječnika ili dograđivati postojeći. Taj je pristup jednim dijelom ocrtan i istražen za hrvatski jezik na starijoj inačici HOBS-a u (Agić i dr. 2010) i (Šojat i dr. 2010).
2. Ukoliko banka ovisnosnih stabala nije velika, očekivano je da neće obuhvatiti ni veliki broj glagola nekoga jezika, a posebno u smislu čestote ostvaraja pojedinih valencijskih okvira. Stoga se može smatrati kako podatci u valencijskome rječniku, ako takav resurs postoji za promatrani jezik, mogu na neki način nadopuniti podatke iz banke stabala s gledišta primjene u sustavu poput ovisnosnoga parsera. Utoliko se banka stabala i valencijski rječnik mogu smatrati komplementarnima s obzirom na valenciju glagola.
3. Ovisnosni parseri temeljeni na podatcima povezuju pojavnice u ovisnosne odnose koristeći jezični model naučen iz banke stabala. Taj jezični model nužno sadržava i neku vrstu implicitnoga zapisa valentnosti pojedinih glagola. Budući da je model izrađen iz banke stabala, odnosno izdvojenoga uzorka za izradu jezičnoga modela, nepotpunost banke stabala s obzirom na valencijski opis glagola preslikava se i na jezični model parsera. To vodi pojavi lančanoga prostiranja pogrješaka (en. *error propagation*) budući da se nedostatci banke stabala prikazuju uvećano kao pogrješke pri parsanju sustavom koji je izgrađen iz nje. Stoga se može pretpostaviti kako bi se takav model ovisnosnoga parsanja mogao izmijeniti tako da koristi podatke o valenciji glagola iz valencijskoga rječnika u postupku izrade modela i/li u

[107] Valentnost glagola može se u banci ovisnosnih stabala promatrati preko čestote ovisnosnih relacija, odnosno svojstava i sintaktičkih funkcija izravnih (i nekih neizravnih) dependenata tih glagola.

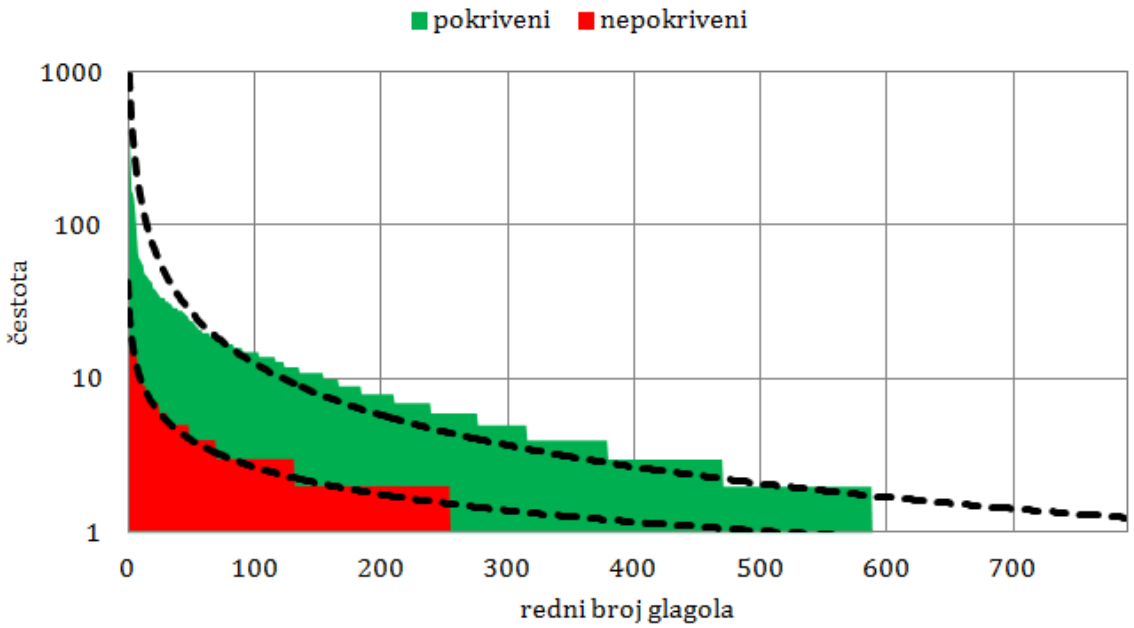
postupku njegove primjene, s ciljem povećanja točnosti povezivanja osnovnih elemenata rečeničnoga ustroja s glagolima-predikatima koji im u rečenici otvaraju mjesta.

Postupak povezivanja ovisnosnoga parsera CroDep0 s valencijskim rječnikom hrvatskih glagola CROVALLEX i posljedična hibridizacija toga parsera sveden je ovime na odabir pristupa ugradnje znanja o hrvatskim glagolima iz CROVALLEX-a u jezični model parsera i/li u postupak parsanja, odnosno algoritam za parsanje Chu-Liu-Edmonds. Razmatranju tih preinaka prethodila je analiza CROVALLEX-a s obzirom na HOBS kako bi se utvrdilo je li njegovo povezivanje s parserom smisleno. Smislenost povezivanja svodi ovdje se definira prema preklapanju glagola iz CROVALLEX-a i HOBS-a: ukoliko CROVALLEX sadrži valencijske okvire za nezanemariv broj glagola iz HOBS-a, onda je logično pretpostaviti mogućnost i svrhovitost korištenja toga dodatnog znanja u parsanju.

U HOBS-u je pronađeno ukupno 1,525 lema glagola za ukupno 12,958 pojavnih oblika, odnosno oko 14.55% od ukupnoga broja osnovnih oblika u HOBS-u ili otprilike 14.72% od ukupnoga broja pojavnica u HOBS-u. Inačica CROVALLEX-a 2.008, korištena u ovome eksperimentu, sadržavala je ukupno 1,797 unosa, odnosno osnovnih oblika glagola i ukupno 5,188 pripadajućih valencijskih okvira. Iz toga su popisa potom uklonjeni glagoli za koje je u CROVALLEX-u navedena čestota 0 i svi povratni glagoli[108], pa je iz rječnika za potrebe eksperimenta zadržano ukupno 1,455 glagola, odnosno ukupno 4,090 valencijskih okvira. Preklapanje HOBS-a i CROVALLEX-a s obzirom na leme glagola, odnosno pokrivenost glagola HOBS-a valencijskim rječnikom iznosi oko 51.87% – u CROVALLEX-u se nalazi 791 od 1,525 glagola iz HOBS-a. S druge strane, ukupno 664 glagola opisanih CROVALLEX-om nije pronađeno u HOBS-u, odnosno oko 45.64% od ukupnoga broja glagola opisanih tim rječnikom. Ručnim je pregledom opaženo kako se radi uglavnom o glagolima širokoga raspona čestota koji nisu svojstveni tekstovima iz informativne domene, što predstavlja odraz CROVALLEX-a kao statičkoga opisa hrvatskoga jezika s jedne strane i HOBS-a kao jednoga – relativno nevelikoga brojem pojavnica, s obzirom, primjerice, na postojeće korpuse hrvatskih tekstova označene na razini morfosintakse (usp. Tadić 2009) i na

[108] Povratni su glagoli uklonjeni zato što su u CROVALLEX-u navedeni kao višerječne jedinice koje se sastoje od glagola i povratne zamjenice (primjerice, *dogoditi se*), a ovisnosno parsanje u ovdje odabranome teorijskom okviru ne podržava eksplicitno rukovanje višerječnim jedinicama. Moguće je ipak zamisliti teorijski model u kojemu se s pomoću banke ovisnosnih stabala prepoznavaju povratni glagoli ili, uopćeno, vrši dodjela sintaktičkih funkcija višerječnim jedinicama. Takav model podržan je i korištenim zapisom HOBS-a, no nije ugrađen u algoritme za parsanje.

ranije spominjane dostupne banke ovisnosnih stabala za druge jezike – ostvarenja sintaktičke strukture hrvatskoga jezika s druge strane. Opažena 50-postotna pokrivenost glagola iz HOBS-a CROVALLEX-om upućuje na smislenost korištenja znanja iz CROVALLEX-a pri parsanju budući da su za preko 50% svih glagola iz uzoraka za treniranje i testiranje dostupni valencijski okviri, odnosno opći podatci o valentnosti tih glagola i dodatne značajke, odnosno zahtjevi za elemente rečeničnoga ustroja kojima ti glagoli otvaraju mjesta u rečenici.



Slika 3-8 Pokrivenost glagola iz HOBS-a CROVALLEX-om

Pokrivenost glagola iz HOBS-a CROVALLEX-om dodatno je ocrtana slikom 3-8 koja pokazuje pokrivenost CROVALLEX-om s obzirom na čestotu glagola u HOBS-u. Iz nje se jasno vidi kako je stvarna pokrivenost – mjerena brojem očekivanih preklapanja u rečenicama iz HOBS-a – veća od 51.87% izmjerenih usporedbom dvaju popisa glagola budući da većina nepokrivenih glagola iz HOBS-a ima čestotu manju od 10, dok s druge strane CROVALLEX pokriva većinu učestalijih glagola. Naknadnim je mjerenjem utvrđeno kako u rečenicama iz HOBS-a CROVALLEX ne sadrži valencijske okvire za ukupno 1,406 pojavnica glagola od ukupno 15,219 pojavnica glagola, odnosno da stvarna nepokrivenost CROVALLEX-om iznosi oko 9.24%. Dakle, otprilike ima 90.76% od ukupnoga broja pojavnica glagola iz HOBS-a kojima se može pridružiti barem jedan valencijski okvir iz CROVALLEX-a.

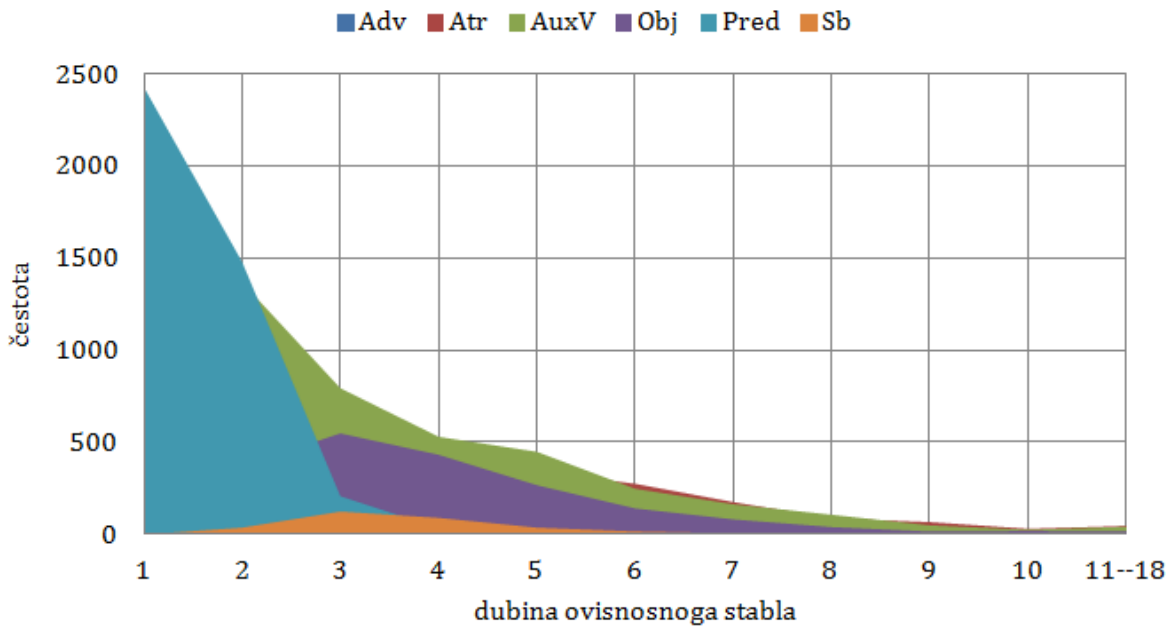
Prikazana razdioba čestota glagola HOBS-a pokrivenih CROVALLEX-om pokazala je, dakle, opravdanost posezanja za CROVALLEX-om kao jezičnim resursom kojim se može

poboljšati ukupna točnost parsanja hrvatskih tekstova. Preostalo je odabrati dobar pristup povezivanju CROVALLEX-a s ovisnosnim parserom CroDep0, odnosno njegovim jezičnim modelom temeljenim na ovisnosnim relacijama i/li algoritmom za parsanje. Dobrim se pritom smatrao svaki pristup koji povećava ukupnu točnost parsanja, a pritom ne uzrokuje značajan pad u opaženoj (posebno vremenskoj) učinkovitosti sustava. Budući da je jezični model parsera izgrađen iz HOBS-a, pretpostavljeno je kako on u sebi već implicitno sadrži i neku vrsru modela valencije pojedinih glagola, izgrađen iz prebrojavanja – i dodjele pripadajućih težina – poveznica između tih glagola i elemenata rečeničnoga ustroja kojima su oni otvorili mjesto u promatranim rečenicama. Dakle, postupak za izgradnju jezičnoga modela kao ulaz prima referentna, odnosno točna ovisnosna stabla ulaznih rečenica koja sadrže točna parsanja svih neposrednih okruženja svih tamo sadržanih glagola. Ovdje je stoga pretpostavljeno kako se postupak izrade jezičnoga modela ne može obogatiti valencijskim rječnikom, no da je moguće izmijeniti postupak parsanja korištenjem opažanja o svojstvima okruženja glagola u HOBS-u i uporabom valencijskih okvira u algoritmu za parsanje. Tablica 3-12 i slika 3-9 predstavljaju jedan znakovit pogled na glagole iz HOBS-a.

Tablica 3-12 Sintaktičke funkcije prema položaju u ovisnosnome stablu za glagole iz HOBS-a

dubina	Adv	Atr	AuxV	Obj	Pred	Sb
1	0	0	14	3	2429	1
2	17	5	1369	367	1480	38
3	229	225	795	544	209	125
4	174	384	531	429	34	91
5	87	344	450	266	13	38
6	43	275	249	141	9	18
7	46	176	165	82	6	8
8	22	88	108	41	6	8
9	14	68	50	17	1	2
10	3	32	27	20	0	2
11-18	12	47	43	17	1	1

Tablica 3-12 i slika 3-9 prikazuju sintaktičke funkcije glagola u HOBS-u s obzirom na položaj tih pojavnica u ovisnosnim stablima. Položaj pojavnica izražen je dubinom pojavnice u ovisnosnome stablu, odnosno udaljenošću mjerenom brojem ovisnosnih relacija koje je potrebno prijeći putanjom od pojavnice do korijenskoga čvora. Tako se vidi da su glagoli na dubini 1 gotovo isključivo predikati (oko 99%), a na dubini 2 predikati (45.18%) i pomoćni glagoli (41.18%), uz doprinos oznake za objekt (11.2%) kojom se u pravilu uvodi objektna zavisna surečenica. Predikati na dubini 2 pritom su obično predikati surečenica nezavisno-složenih rečenica. Na dubini 3 razdioba među sintaktičkim funkcijama potpuno je ujednačena i predstavlja uvođenja različitih vrsta zavisnih rečenica. Na dubinama od 4 i više glagolima uglavnom nisu dodijeljene funkcije predikata, a i te se dodjele (njih oko 1.49% od ukupnoga broja) načelno mogu smatrati pogrješkama u ručnome označavanju. Također je primjetno kako glagola na dubinama od 5 nadalje ima značajno manje (ukupno oko 24.29%) nego na dubinama bližima korijenskomu čvoru (oko 75.71% na dubinama od 1 do 4, odnosno 62.60% samo na prve tri dubine).



Slika 3-9 Sintaktičke funkcije prema položaju u ovisnosnome stablu za glagole iz HOBS-a

Slika 3-9 dodatno pojašnjava razdiobu iz tablice 3-12. Iz nje je jasno vidljiv značajan udio predikata u razdiobi glagola na prve tri dubine ovisnosnoga stabla i značajan pad toga udjela na ostalim razinama. Također je vidljiva i dubina stabla na kojoj se uvode zavisne surečenice dodjelom opisnih sintaktičkih funkcija tih surečenica njihovim predikatima, kao i stalna prisutnost pomoćnih glagola na svim dubinama ovisnosnoga stabla. Nadalje, tamo gdje

su glagoli označeni sintaktičkom funkcijom predikata – dakle, na prve dvije ili najviše prve tri dubine ovisnosnoga stabla – njihova je uloga uvođenje samostalnih elemenata rečeničnoga ustroja u rečenicu. Dakle, očekivano je povezivanje subjekata, objekata i priložnih oznaka s predikatima na razini jednostavne rečenice ili surečenice i povezivanje predikata zavisnih surečenica kojima je dodijeljena neka nepredikatna sintaktička funkcija zbog opisa vrste uvedene surečenice[109]. Ta je razdioba prikazana tablicom 3-13.

Tablica 3-13 Razdioba sintaktičkih funkcija pojavnica direktno ovisnih o predikatima

Sb	**AuxP**	**AuxV**	**Obj**	**Adv**	**AuxC**	**Pnom**
19.87%	16.38%	15.47%	12.17%	10.00%	5.34%	4.27%
Coord	**AuxR**	**AuxY**	**AuxX**	**AuxT**	**AuxG**	**Apos**
3.93%	2.01%	2.00%	2.00%	1.61%	1.42%	1.19%
AtvV	**Pred**	**ExD**	**AuxZ**	**AuxK**	**AuxO**	**Atr**
0.82%	0.65%	0.40%	0.35%	0.05%	0.05%	0.03%

Iz tablice 3-13 jasno se vidi kako predikati najčešće otvaraju mjesto subjektima (skoro 20% od svih ovisnosnih relacija u kojima je predikat glava relacije), prijedlozima koji uvode prijedložno-padežne izraze (16.38%), pomoćnim glagolima (15.47%), objektima (12.17%) i prilozima ili priložnim oznakama (10%). Pet navedenih sintaktičkih funkcija, koje okvirno odgovaraju osnovnim elementima rečeničnoga ustroja, sačinjava ukupno 73.89% svih ovisnosnih relacija koje za glavu imaju predikat.

Iz prikazanih pravilnosti HOBS-a s obzirom na glagole, odnosno predikate i elemente rečeničnoga ustroja kojima oni otvaraju mjesta u rečenici, i s obzirom na dopunjavanje znanja implicitno sadržanoga u HOBS-u eksplicitno zapisanim znanjem o valencijskim okvirima glagola u CROVALLEX-u, ovisnosni parser CroDep0 prilagođen je uvođenjem niza pravila, odnosno neobvezujućih ograničenja na svojstva ovisnosnih stabala koja se izgrađuju parsnim algoritmom. Formalno, ranije je pokazano kako se opći model parsera $\mathrm{M} = (\Gamma, \lambda, \mathrm{h})$ odnosi na parsere temeljene na grafovima i na prijelazničke parsere te se u slučaju pojedinih parsera

[109] Ranije je spomenuto istraživanje (Berović i dr. 2012b) u kojemu se razmatra povezivanje surečenica dodjelom opisnih funkcija sintaktičkim veznicima i zadržavanje funkcije predikata na glagolima. To bi rješenje olakšalo ovisnosno parsanje predikata i omogućilo bolje crpljenje obavijesti iz rečenica postupcima nadgradnje na rezultate ovisnosnoga parsanja.

pobliže označuje definicijom jezičnoga modela λ i algoritma za parsanje h. Pritom je za oba razreda parsera navedeno kako se skup ograničenja Γ definira tako da izlazne strukture iz pojedinih algoritama za parsanje ograniči na ovisnosna stabla prema formalnoj definiciji i traženim svojstvima ovisnosnih stabala. Za potrebe vezivanja ovisnosnoga parsera CroDep0 s valencijskim rječnikom CROVALLEX pobliže se definira upravo skup ograničenja Γ, i to tako da se njime uvode dodatni zahtjevi za oblikovanje ovisnosnih stabala koja za ulazne rečenice stvara njegov algoritam za parsanje. Postavljeni skup ograničenja povlači i određene zahtjeve kojima odabrani algoritam za parsanje mora udovoljiti. Ograničenja su postavljena na sljedeći način.

1. Glagolski se predikati unutar ovisnosnoga stabla moraju nalaziti na jednoj od prve tri razine, odnosno prve tri dubine s obzirom na korijenski čvor ovisnosnoga stabla.
2. Najveći i najmanji broj izravno ovisnih pojavnica koje se vezuju uz te predikate određen je navedenim brojem elemenata rečeničnoga ustroja iz valencijskih okvira tih predikata u CROVALLEX-u.
3. Morfosintaktička svojstva elemenata rečeničnoga ustroja, odnosno pojavnica koje se vezuju uz glagolske predikate podržane u CROVALLEX-u moraju u najvećoj mogućoj mjeri odgovarati ograničenjima koja su za ovisne elemente uvedene tim glagolskim predikatima propisana u CROVALLEX-u.

Budući da se navedeni zahtjevi odnose na izlazne strukture, odnosno ovisnosna stabla koja za ulazne rečenice proizvodi algoritam za parsanje, nužno je omogućiti vrjednovanje tih struktura s obzirom na navedene zahtjeve za vrijeme rada samoga algoritma. Razmotrena su teorijski dva razdvojena pristupa: prilagodba izvedbe algoritma Chu-Liu-Edmonds iz parsera CroDep0 i uvođenje zamjenskoga algoritma za parsanje. Pritom je osnovna postavka kojom se uvažavaju postavljena ograničenja zamišljena u vidu unutarnjega glasovanja, odnosno parsanja u kojemu parsni algoritam za svaku ulaznu rečenicu ponudi određeni broj izlaznih ovisnosnih stabala, pa se ta stabla nekom dodatnom procedurom vrjednuju kako bi se odabralo najbolje ovisnosno stablo (en. *k-best parsing, reranking*) ili se preuređuju s obzirom na neki skup pravila u jedno ovisnosno stablo. Dakle, od parsera se zahtijeva da za ulaznu rečenicu proizvede *k* ovisnosnih stabala koja predstavljaju kandidate svrstane prema pouzdanosti parsera s obzirom na jezični model, a potom se dodatnom procedurom svako od tih stabala vrjednuje s obzirom na povezivanje predikata prema propisanim ograničenjima te se za izlazno stablo odabire ono koje u najvećoj mjeri udovoljava tim ograničenjima. Budući

da je ranije pokazano (usp. McDonald i dr. 2005b:527) kako su proširenja algoritma Chu-Liu-Edmonds za pronalaženje većeg broja ovisnosnih stabala svrstanih po razini valjanosti vremenski neučinkovita do razine neupotrebljivosti u stvarnim sustavima, ovdje je odabran pristup s uvođenjem novoga algoritma u sustav CroDep0. Točnije, odabran je algoritam za pronalaženje nekoga broja poredanih prostirućih stabala opisan u (Camerini i dr. 1980) i uspješno korišten za ovisnosno parsanje u (Hall 2007)[110]. Algoritam za svaku ulaznu rečenicu daje *k* ovisnosnih stabala – u ovdje predstavljenoj prototipnoj inačici pronalazi se 10 stabala budući da (Hall 2007) bilježi najveći rast točnosti s pomakom od 1 do 10, veći nego od 10 do 500, uz najmanji pad učinkovitosti – i ta se ovisnosna stabla potom vrjednuju prema tri navedena ograničenja za svako povezivanje pojavnica s glagolskim predikatima. Svakoj takvoj ovisnosnoj relaciji dodjeljuje se istinosna vrijednost koja predstavlja ocjenu relacije s obzirom na svako zadano ograničenje. Tim se vrjednovanjem gradi ljestvica valjanosti pojedinih stabala s obzirom na ograničenja te se taj popis uspoređuje s ljestvicom valjanosti koju je proizveo algoritam za parsanje. Za najbolje ovisnosno stablo i konačni izlaz iz algoritma za parsanje bira se ono ovisnosno stablo koje bilježi najvišu prosječnu ocjenu u oba popisa. Ukoliko više ovisnosnih stabala bilježi istu prosječnu ocjenu, odabire se ono stablo s najvišom ocjenom iz popisa dobivenoga algoritmom za parsanje. Ova nadgradnja parsera CroDep0 naziva se CroDep, a njegovo vrjednovanje na HOBS-u izloženo je dalje u tekstu. Prije vrjednovanja, ovdje je izložen kratki formalni opis jezičnoga modela i parsnoga algoritma predloženoga parsera CroDep.

Parser CroDep sastoji se od tri osnovna modula, po uzoru na već prikazane parsere temeljene na podatcima i njihov konceptualni model – sadrži modul za treniranje jezičnoga modela, modul za pronalaženje *k* najboljih ovisnosnih stabala za ulaznu rečenicu i modul za ponovno vrjednovanje tih stabala pomoću valencijskoga rječnika CROVALLEX.

Po uzoru na izvedbu relacijski uvjetovanoga ovisnosnog parsanja temeljenoga na teoriji grafova unutar sustava MSTParser, za parser CroDep0 razvijen je postupak za treniranje relacijski uvjetovanoga jezičnog modela algoritmom perceptron te je taj postupak izravno preuzet iz parsera CroDep0 i ugrađen u parser CroDep. Jezični je model, kao u prikazu relacijski uvjetovanoga ovisnosnog parsanja, definiran na sljedeći način.

$$\lambda = \left\{\lambda_{(w_i,r,w_j)}, \forall(w_i, r, w_j) \in A\right\}, \lambda_{(w_i,r,w_j)} = \mathbf{w} \cdot \mathbf{f}(w_i, r, w_j)$$

[110] Vidjeti i izvedbu toga parsera, URL http://web.me.com/khallbobo/KeithHall/Home.html (2012-04-27)

$$\sum_{(w_i,r,w_j)\in A} \lambda_{(w_i,r,w_j)} = \sum_{(w_i,\, w_j)\in A'} \lambda_{(w_i,w_j)}$$

$$(w_i, r, w_j) \in A \Leftrightarrow (w_i,\ w_j) \in A' \wedge r = \arg\max_r \lambda_{(w_i,r,w_j)}$$

Pretpostavlja se, dakle, dodjela težina pojedinim značajkama koje u jezičnome uzorku, odnosno banci ovisnosnih stabala, pobliže određuju pojedine ovisnosne relacije te naknadna rekonstrukcija ovisnosnih relacija, koje se za postupak parsanja izostavljaju zbog specifičnih zahtjeva algoritama iz teorije grafova za pronalaženje najvećih prostirućih stabala, kako je opisano ranije. Parsanje je formalno definirano kao pretraživanje jezičnoga modela u potrazi za onim od svih mogućih ovisnosnih stabala za ulaznu rečenicu koje ulaznu rečenicu prema modelu najbolje opisuje:

$$h(s, \Gamma, \lambda) = \arg\max_{G\in G_s} \sum_{(w_i,r,w_j)\in A} \log \lambda_{(w_i,r,w_j)} = \arg\max_{G\in G_s} \sum_{(w_i,r,w_j)\in A} \mathbf{w}\cdot\mathbf{f}(w_i, r, w_j)$$

Kako je ranije navedeno, parser CroDep0 za rješavanje ovoga optimizacijskog problema koristi algoritam CLE. Budući da je taj algoritam dokazano suboptimalan za pronalaženje nekog broja najboljih rješenja, nanovo je implementiran u parseru CroDep algoritam iz (Camerini 1980, Hall 2007). Prema (Hall 2007), taj algoritam je izveden u obliku tri vezana algoritma. Algoritam 3-1, dodatno opisujućega naziva *first-best* (hr. *prvi najbolji*), pronalazi u jezičnome modelu najveće prostiruće stablo za ulaznu rečenicu i po toj je funkcionalnosti identičan algoritmu CLE. Vremenska složenost algoritma 3-1 je $O(m \log n)$, odnosno $O(n^2)$ za guste ili potpuno povezane grafove. Algoritam 3-2 pronalazi iduće najbolje rješenje, odnosno ono prostiruće stablo nad ulaznom rečenicom koje predstavlja najveće prostiruće stablo ukoliko se izuzme optimalno rješenje pronađeno algoritmom *first-best*. Stoga se algoritam 3-2 naziva još i algoritmom *next-best* (hr. *idući najbolji*). Algoritam pronalazi iduće najbolje ovisnosno stablo pronalaženjem jedne ovisnosne relacije iz najboljega rješenja problema parsanja – odnosno izlaza iz algoritma *first-best* – koju je potrebno onemogućiti u najboljemu rješenju kako bi se ponovljenim radom algoritma *first-best* dobilo novo rješenje. Vremenska složenost ovoga algoritma također je $O(m \log n)$, odnosno $O(n^2)$ za guste ili potpuno povezane grafove. Treći algoritam, ovdje algoritam 3-3 ili algoritam *k-best* (hr. *k najboljih*) koristi prethodna dva algoritma za onemogućavanje nekog broja *k* ovisnosnih relacija iz slijeda optimalnih rješenja i posljedično pronalaženje popisa od *k* najboljih ovisnosnih stabala za ulaznu rečenicu. Složenost trećeg algoritma je konstantna pa je ukupna

složenost prikazanoga parsnog algoritma jednaka $O(km \log n)$, odnosno $O(kn^2)$ za guste ili potpuno povezane grafove.

firstbest(G, Y, Z)

neka je $G = (G \cup Y) - Z, G = (V, E), V = \{v_1, \dots, v_n\},\ E = \{e_{11}, e_{nn}\}$

$B = \emptyset, C = V$

za svaki neposjećeni čvor $v_i \in V$

označi v_i posjećenim

dohvati najbolju ulaznu vezu $b \in \{e_{jk} : k = i\}$ za v_i

$B = B \cup b, \beta(v_i) = b$

ako B sadrži neki ciklus C

stvori novi čvor $v_{n+1}, C = C \cup v_{n+1}$

neka su svi čvorovi iz C djeca od v_{n+1} u C

sažmi sve čvorove iz C u čvor v_{n+1}

dodaj v_{n+1} u popis neposjećenih čvorova

$n = n + 1, B = B - C$

proširi C uklanjanjem ciklusa

vrati najbolji $A = \{b \in E \mid \exists v \in V : \beta(v) = b\}, C$

Algoritam 3-1 Pronalaženje prvoga od *k* najvećih prostirućih stabala (*first-best* MST)

nextbest(G, Y, Z, A, C)

$\delta = +\infty$

za svaki neposjećeni čvor $v \in V$

dohvati najbolju ulaznu vezu b za čvor v

ako je $b \in A - Y$

f = alternativna veza u čvor v

ako zamjena veze b vezom f ishoduje manji δ

ažuriraj δ, neka je $e = f$

ako b oblikuje ciklus, razriješi ciklus kao u algoritmu firstbest(G, Y, Z)

vrati e i δ

Algoritam 3-2 Pronalaženje idućega najvećeg prostirućeg stabla (*next-best* MST)

kbest(G, k)

A, C ← firstbest(E, V, ∅, ∅)

e, δ ← nextbest(E, V, ∅, ∅, A, C)

popis ← A

Q ← dodaj(score(A) − δ, e, A, C, ∅, ∅)

za svaki j = 2 ... k

s, e, A, C, Y, Z = ukloni(Q)

Y' = Y ∪ e, Z' = Z ∪ e

A', C' ← best(E, V, Y, Z')

popis ← A'

e', δ' ← nextbest(E, V, Y', Z, A', C'), Q ← dodaj(score(A) − δ', e', A', C', Y', Z)

e', δ' ← nextbest(E, V, Y, Z', A', C'), Q ← dodaj(score(A) − δ', e', A', C', Y, Z')

vrati popis

Algoritam 3-3 Pronalaženje *k* najvećih prostirućih stabala (*k-best* MST)

Treći je modul parsera CroDep onaj za ponovno vrjednovanje predloženih ovisnosnih stabala dobivenih algoritmom *k-best* pomoću valencijskoga rječnika CROVALLEX, prema osnovnim načelima prikazanima ranije u tekstu. Neka je dan valencijski rječnik kao skup uređenih parova lema glagola i pripadajućih valencijskih okvira i neka je također uz njega definirana funkcija koja za neki polazni oblik pretražuje valencijski rječnik.

$$\text{vallex} = \{(\text{verb}_i, \text{frames}_i)\}_{i=1}^{L}$$

$$\forall i, \text{frames}_i = \left\{\left(\min_i, \{c_1, \dots, c_{\min_i}\}\right)\right\}_{j=1}^{|\text{frames}_i|}, \forall j, k, \min_j \in \mathbb{N}, c_k = (\text{obl}_k, \text{msd}_k)$$

$$\text{obl}_k = \begin{cases} 1, \text{ako je značajka obvezna} \\ 0, \text{u protivnom} \end{cases}, \qquad \text{msd}_k \dots \text{zahtijevana MSD značajka}$$

$$\text{vf}: V \longrightarrow \text{frames}, \qquad \text{vf}(v) = \{f_1, \dots, f_l\}$$

Pojedini unos $(\text{verb}_i, \text{frames}_i)$ iz valencijskoga rječnika predstavlja jednu lemu glagola verb_i i niz valencijskih okvira frames_i. Svaki okvir se sastoji od broja $\min_i$ preko kojega je za odabrani glagol definiran najmanji broj dependenata koje može uvesti u rečenicu i skupa uređenih parova $(\text{obl}_k, \text{msd}_k)$ u kojemu prvi element predstavlja istinosnu funkciju

obveznosti definiranoga ograničenja, a drugi element jedno konkretno morfosintaktičko svojstvo prema standardu Multext East koje se zahtijeva od uvedenoga dependenta. Dana je i funkcija vf za pretraživanje valencijskoga rječnika za bilo koji čvor ovisnosnoga grafa, odnosno za bilo koji osnovni oblik riječi iz ulaznoga teksta.

lexrerank$\left(\{G_i\}_{i=1}^{k}, \text{vallex}\right)$
za svaki $G_i = (V, A)$
za svaki čvor $w \in V$ ovisnosnoga grafa G_i
ako je $\text{vf}\left(\text{lemma}(w)\right) \neq \emptyset$
za svaku ovisnosnu relaciju $a = \left(w, r, w_j\right) \in A$ i
za svaki valencijski okvir $f \in \text{frames}_w$
ako je $\text{msd}(w_j) = \text{msd}_f$ i ako je $\text{obl}_f = 1$
uvećaj $\text{score}(G_i)$
ako je $\min(\text{min}_f) \leq \left|\{w_j\}\right| \leq \max(\text{min}_f)$, uvećaj $\text{score}(G_i)$
popis $\leftarrow (G_i, \text{score}(G_i))$
vrati popis

Algoritam 3-4 Vrjednovanje ovisnosnih stabala valencijskim rječnikom (*lexical-reranker*)

Algoritam 3-4 predstavlja ponovno vrjednovanje kandidatskih ovisnosnih stabala predloženih algoritmom *k-best* pomoću valencijskoga rječnika. Svaka ovisnosna relacija koja za glavu ima neku pojavnicu čiji se polazni oblik nalazi u valencijskom rječniku vrjednuje se usporedbom njezinih MSD-značajki sa MSD-značajkama propisanima rječnikom. Ukoliko se značajke podudaraju i ukoliko je broj mjesta otvorenih pronađenom pojavnicom između najmanjeg i najvećeg broja mogućih otvorenih mjesta za tu pojavnicu prema valencijskom rječniku, uvećava se ocjena ovisnosnoga stabla. Tako se vrjednuju sve ovisnosne relacije svih ulaznih ovisnosnih stabala pa algoritam završava vraćajući popis njihovih ukupnih ocjena. Popis iz algoritma 3-4 – također nazvanoga i *lexical-reranker* (hr. *ponovni vrjednovatelj leksikonom*) – uspoređuje se s popisom iz algoritma *k-best* prema ranije opisanom postupku te se tako odabire jedno ovisnosno stablo kao konačni izlaz iz parsnoga algoritma. Slijedi opis vrjednovanja parsera CroDep.

3.2.3.2 Plan eksperimenta

Vrjednovanje hibridnoga ovisnosnog parsera CroDep provedeno je u skladu s ranije izloženim planom eksperimenta s usporedbom ovisnosnih parsera temeljenih na grafovima i prijelazničkih ovisnosnih parsera. Jezični model parsera CroDep izgrađen je na trenažnim skupovima HOBS-a i primijenjen na testnim skupovima. Primijenjene su mjere LA, LAS i UAS i, gdje je primjenljivo, preciznost i odziv, za vrjednovanje ukupne točnosti i točnosti s obzirom na vrste riječi i sintaktičke funkcije. Promatran je utjecaj duljine rečenice, smjera i duljine ovisnosne relacije na točnost parsanja i izmjerena je vremenska i prostorna složenost izgradnje i korištenja njegova jezičnog modela.

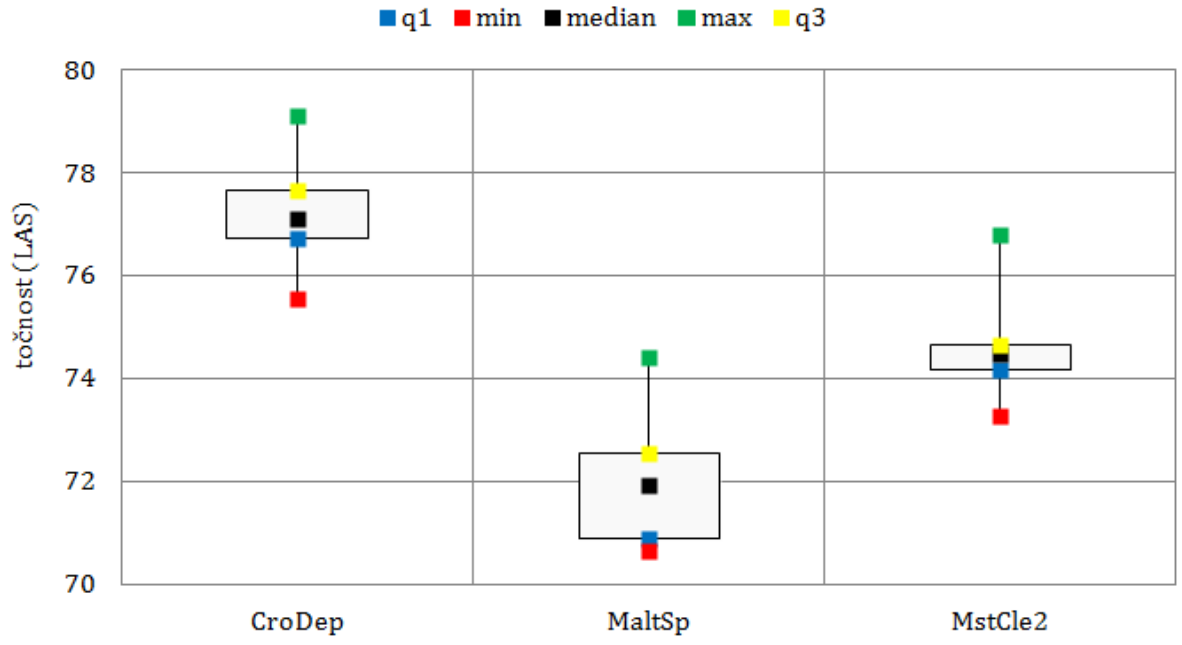
Iz prethodnoga su eksperimenta odabrana dva parsera za usporedbu s parserom CroDep. Odabrani su parseri predstavnici dviju skupina s obzirom na teorijski okvir, odnosno parser temeljen na grafovima (MstCle2) i prijelaznički parser (MaltSp) s najvećim zabilježenim ukupnim točnostima pri parsanju HOBS-a. Ti su parseri uspoređeni s parserom CroDep prema svim vrjednovanim značajkama.

3.2.3.3 Rezultati

Tablica 3-14 predstavlja točnost parsera CroDep prema mjerama za vrjednovanje LA, LAS i UAS i točnost po istim mjerama za izdvojene vrste riječi. Zabilježena je točnost prema mjeri LAS od 77.21%, što predstavlja porast točnosti od 2.68% u odnosu na parser MstCle2, najbolji iz prethodnoga eksperimenta. Razlika između točnosti tih dvaju parsera statistički je značajna s obzirom na sve tri mjere za vrjednovanje. Tablicu dopunjava slika 3-10 kojom je dana usporedba statističkih značajki točnosti triju izdvojenih sustava prema mjeri LAS i ocrtani intervali povjerenja.

Tablica 3-14 Točnost parsera CroDep s obzirom na vrstu riječi i ukupno

mjera	N	V	Z	A	S	C	P	R	ukupno
LA	**85.34**	**87.89**	**91.20**	92.67	**98.64**	87.12	**84.38**	80.14	**88.27 ± 0.30**
LAS	**80.10**	**82.85**	73.48	86.40	71.20	**63.24**	76.04	65.77	**77.21 ± 0.59**
UAS	**90.16**	**86.84**	**75.73**	89.13	71.92	**67.06**	84.84	75.30	**83.05 ± 0.50**



Slika 3-10 Ukupna točnost parsera CroDep, MaltSp i MstCle2

Najviši zabilježeni rezultati parsanja parserom CroDep u usporedbi s parserima iz prethodnoga eksperimenta dodatno su označeni u tablici 3-14 i svim idućim tablicama. Tako se u tablici 3-14 vidi kako je za porast ukupne točnosti parsanja zaslužan isključivo statistički značajan porast točnosti parsanja imenica i glagola. S obzirom na ranije prikazane značajke povezivanja glagolskih predikata u HOBS-u, opaženi porast točnosti pri povezivanju tih vrsta riječi u ovisnosne relacije pokazuje opravdanost uporabe jezičnoga znanja sadržanoga u valencijskim okvirima iz CROVALLEX-a u ovisnosnome parsanju.

Tablica 3-15 Točnost parsera CroDep s obzirom na sintaktičku funkciju

mjera	Adv	Apos	Atr	AuxC	AuxP	Coord	Obj	Pnom	Pred	Sb
LAS	70.69	34.49	83.94	69.80	70.59	49.41	**83.17**	**71.46**	**82.12**	**85.01**
UAS	84.81	40.99	**88.90**	71.53	71.48	50.87	**93.12**	**79.92**	**86.81**	**91.35**
P (LA)	78.96	58.92	**91.21**	91.96	97.86	89.72	**84.12**	**77.06**	**84.36**	**86.78**
O (LA)	74.11	41.59	90.94	87.77	97.74	81.60	**94.75**	49.73	**97.21**	**97.50**

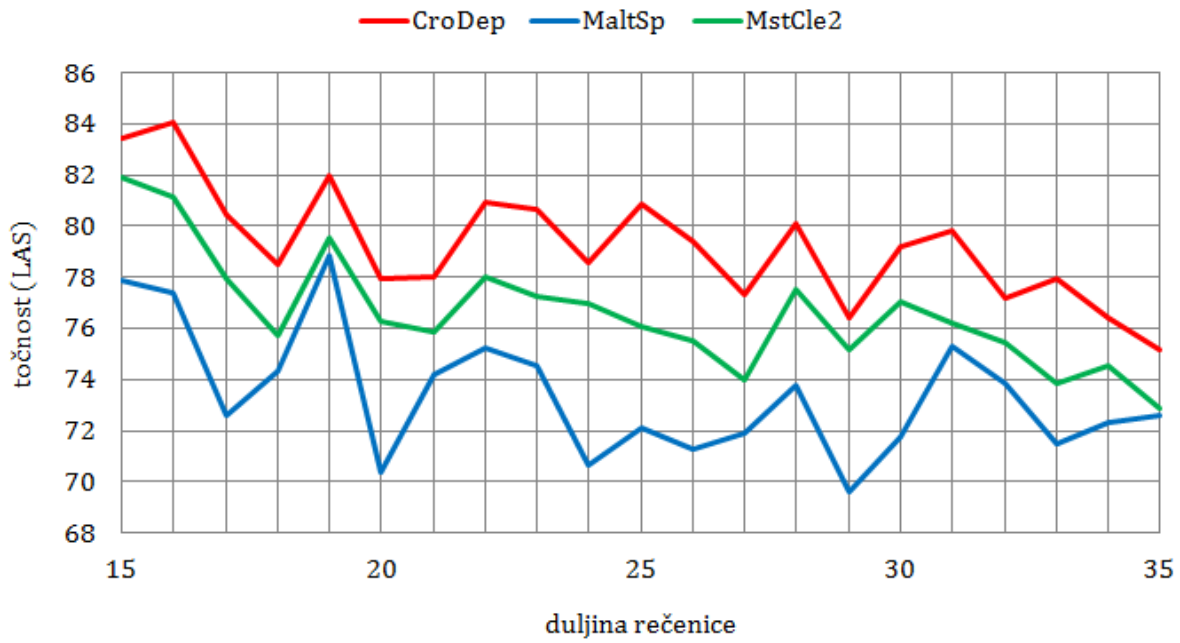
Tablica 3-15 pokazuje točnost parsanja parserom CroDep s obzirom na sintaktičku funkciju i predstavlja spoj tablica 3-8 i 3-9 iz prikaza rezultata prethodnoga eksperimenta. Iz nje se prvenstveno može iščitati porast točnosti povezivanja i dodjele za sintaktičke funkcije

koje predstavljaju predikate, subjekte, objekte i u nekoj mjeri atribute. Prema mjerama LAS i UAS parser CroDep u usporedbi s parserom MstCle2 bilježi u pravilu povećanje točnosti od preko 10% za predikate, subjekte i objekte, što dodatno potvrđuje smislenost povezivanja CROVALLEX-a i parsera temeljenoga na grafovima.

Tablica 3-16 Točnost parsera CroDep s obzirom na smjer ovisnosti

mjera	lijevo	desno	korijen
preciznost	**92.18**	**91.09**	**90.25**
odziv	**91.52**	**91.54**	**91.67**

Tablica 3-16, poput tablice 3-10 iz prikaza rezultata prethodnoga eksperimenta, govori o preciznosti i odzivu parsera CroDep pri povezivanju pojavnica u ovisnosne relacije s obzirom na njihov relativni položaj u rečenicama. Slično kao i kod svih ostalih vrjednovanih parsera, CroDep također neznatno točnije povezuje promatrane pojavnice s pojavnicama koje se u rečenici pojavljuju ranije.



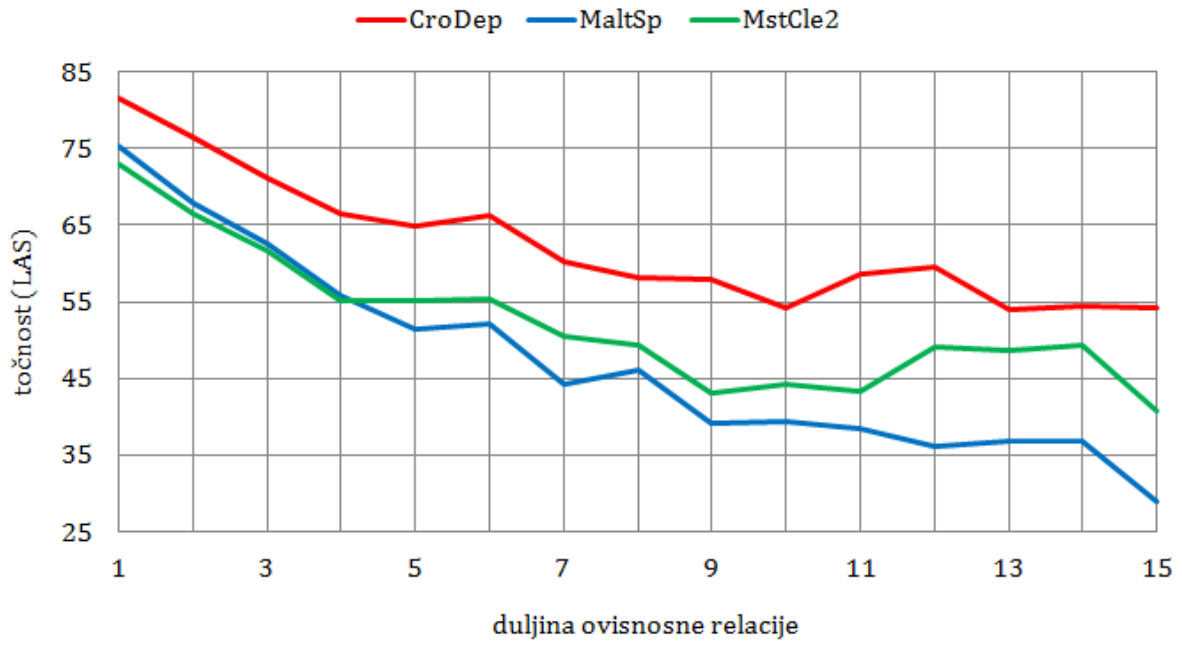
Slika 3-11 Točnost parsera CroDep, MaltSp i MstCle2 s obzirom na duljinu rečenice

Slika 3-11 predstavlja usporedbu točnosti parsera CroDep i izdvojenih parsera MaltSp i MstCle2 prema mjeri LAS s obzirom na duljinu rečenice mjerenu brojem pojavnica. Može se

primijetiti vrlo slična funkcijska ovisnost točnosti o duljini rečenice za parsere CroDep i MstCle2 koja je dijelom očekivana budući da se radi – unatoč razlici u izboru algoritma za parsanje i u redu jezičnoga modela – o parserima temeljenima na grafovima koji postupak izrade jezičnoga modela temelje na istome algoritmu. Kod ovoga pogleda na točnost parsanja također vrijedi ranija napomena o mogućemu negativnom utjecaju veličine HOBS-a na uvjete mjerenja i mjerodavnost dobivenih rezultata.

Tablica 3-17 Točnost parsera CroDep s obzirom na duljinu ovisnosne relacije

mjera	1	2	3-6	7-...	korijen
preciznost	90.63	**84.60**	**81.64**	**75.14**	**90.25**
odziv	**94.14**	**85.88**	**77.14**	63.70	**91.67**



Slika 3-12 Točnost parsera CroDep, MaltSp i MstCle2 s obzirom na duljinu ovisnosne relacije

Tablica 3-17 i slika 3-12 prikazuju točnost parsera CroDep pri rekonstrukciji ovisnosnih relacija iz HOBS-a s obzirom na udaljenost među vezanim pojavnica mjerenu brojem drugih pojavnica koje se nalaze među njima i usporedbu s istim značajkama ukupne točnosti kod parsera MaltSp i MstCle2.

Vidljiv je rast preciznosti i odziva kod parsanja parserom CroDep u odnosu na parsere MaltSp i MstCle2 u većini skupina ovisnosnih relacija prema udaljenosti među pojavnicama, a slika 3-12 pokazuje i ponešto blaži pad točnosti parsera CroDep s rastom duljine ovisnosne relacije u odnosu na parser MstCle2. Kao i ranije, slika 3-12 ocrtava raslojavanje između parsera temeljenih na grafovima i prijelazničkih parsera s obzirom na rastuću duljinu rečenice i uz nju vezani porast broja ovisnosnih relacija koje se prostiru preko većega broja pojavnica. Međutim, budući da se uvedene izmjene algoritma za parsanje u parseru CroDep odnose prije svega na poboljšanje točnosti na određenim razredima ovisnosnih relacija, sintaktičkim funkcijama i određenim vrstama riječi, statističke značajke prikazane slikom 3-11 i 3-12 i tablicama 3-16 i 3-17 očekivano odražavaju samo ukupni porast točnosti, a ne pokazuju uzrok toga porasta jednako jasno kao prikazi koji se odnose na točnost s obzirom na vrstu riječi i sintaktičku funkciju.

Tablica 3-18 prikazuje rezultate mjerenja brzine izvođenja i potrošnje računalne radne memorije za parsere CroDep, MaltSp i MstCle2. Prema provedenom vrjednovanju, parser s najmanjim memorijskim zahtjevima je prijelaznički parser MaltSp koji je izveden iz sustava MaltParser i koristi potporne vektorske strojeve za izgradnju jezičnoga modela. Budući da se radi o jezičnome modelu temeljenom na lokalnome, odnosno fraznome okruženju pojedinih pojavnica, on je očekivano manje složen od jezičnih modela parsera temeljenih na grafovima koji su zasnovani na neusmjerenome pretraživanju. Također, značajka je biblioteke LIBSVM za potporne vektorske strojeve, koja je korištena u ovim eksperimentima, niži memorijski otisak i sporiji postupak izgradnje modela u odnosu na biblioteku LIBLINEAR. Korištenjem druge biblioteke memorijski otisak parsera MaltSp značajno se povećava (na preko 3.5 GB), no vrijeme izgradnje modela značajno se smanjuje (na oko 60 min). Najbrže se izgrađuje jezični model parsera CroDep budući da se radi o najjednostavnijemu od triju jezičnih modela, onome relacijski uvjetovanom, odnosno modelu prvoga reda. Model drugoga reda, temeljen na parovima ovisnosnih relacija i korišten u parseru MstCle2, trenira se gotovo 2.5 puta duže od modela iz parsera CroDep, dok je vrijeme treniranja parsera MaltSp usporedivo s vremenom treniranja parsera CroDep. Parser MstCle2 najbrži je pri parsanju ulaznih rečenica, očekivano, budući da parsa primjenom učinkovite inačice algoritma Chu-Liu-Edmonds. Parser CroDep gotovo je 2.5 puta sporiji od njega, što odražava znatno veću složenost njegova algoritma za parsanje koji u međukoraku za ulaznu rečenicu daje 10 različitih kandidatskih ovisnosnih stabala i pripadajućih ocjena valjanosti te odabire jedno od tih stabala s obzirom na dodatno vrjednovanje prema ograničenjima iz CROVALLEX-a.

Najsporiji je prijelaznički parser MaltSp, gotovo 3.3 puta sporiji od parsera MstCle2, a to je usporenje uzrokovano brzinom učitavanja njegova jezičnog modela, a ne brzinom samoga algoritma za parsanje, no ponovno su njegovi memorijski zahtjevi najmanji. Najveće memorijske zahtjeve pri primjeni algoritma za parsanje – uzrokovane složenošću jezičnoga modela temeljenoga na parovima ovisnosnih relacija – bilježi parser MstCle2.

Tablica 3-18 Vremenski i prostorni zahtjevi parsera CroDep, MaltSp i MstCle2

postupak	Mjera	CroDep	MaltSp	MstCle2
treniranje	min	**137.79 ± 3.26**	143.9 ± 2.85	328.07 ± 12.16
	MB	~ 2300	**~ 1800**	~ 2800
testiranje	sec	351.74 ± 4.22	470.56 ± 12.11	**143.25 ± 2.18**
	MB	~ 1850	**~ 750**	~ 2200

4 Zaključak

Ovisnosno parsanje je strojna sintaktička analiza tekstova prirodnoga jezika prema sintaktičkome formalizmu temeljenom na ovisnosnoj gramatici. U ovome radu istraženi su neki pristupi ovisnosnomu parsanju hrvatskih tekstova unutar teorijskoga okvira ovisnosnoga parsanja temeljenoga na podatcima i s pomoću Hrvatske ovisnosne banke stabala kao uzorka hrvatskih tekstova i implicitnoga modela ovisnosne sintakse hrvatskoga jezika. Za osnovni zadatak prikazanoga istraživanja postavljeno je:

1. utvrditi razinu primjenjivosti postojećih pristupa ovisnosnomu parsanju temeljenih na podatcima pri parsanju tekstova hrvatskoga jezika i
2. istražiti neke mogućnosti povezivanja tih pristupa ovisnosnomu parsanju s jezičnim resursima i/li alatima dostupnima za obradbu hrvatskih tekstova s ciljem povećanja ukupne točnosti parsanja hrvatskih tekstova.

S tim ciljem postavljen je najprije teorijski okvir za uvođenje eksperimenta s parsanjem hrvatskih tekstova. Formalno je definirano ovisnosno parsanje preko definicije parsanja općenito i kao problema svojstvenoga području računalne inteligencije, postavljanja zahtjeva i općih kriterija za vrjednovanje parsera te razmatranja odnosa parsanja i formalnih modela za opisivanje sintaktičke strukture prirodnoga jezika. Uvedena je razlika između parsanja teksta i parsanja formalnom gramatikom te parsanja temeljenoga na pravilima i parsanja temeljenoga na podatcima. U odnos je postavljeno parsanje teksta temeljeno na podatcima i dostupnost sintaktički označenih korpusa tekstova prirodnoga jezika, odnosno banaka ovisnosnih stabala. Formalna je definicija ovisnosnoga parsanja i ovisnosnoga parsera izgrađena prikazivanjem njezinih građevnih elemenata – formalnoga sintaktičkog modela i algoritma za parsanje, odnosno primjenu toga modela na ulaznim tekstovima – s formalnoga, ali i s razvojnoga, odnosno povijesnoga gledišta. Potom su izdvojeni pristupi ovisnosnomu parsanju temeljenom na podatcima, za koje je ranijim istraživanjima utvrđena postojanost visoke razine točnosti na velikom skupu raznorodnih jezika: pristup ovisnosnomu parsanju temeljen na teoriji grafova i pristup temeljen na prijelazničkim sustavima. Izložene su formalne definicije njihovih jezičnih modela i algoritama za parsanje, kao i opažena njihova svojstva s obzirom na ranija istraživanja ovisnosnoga parsanja tekstova prirodnoga jezika. Također je postavljen i formalni okvir za vrjednovanje točnosti i učinkovitosti ovisnosnoga parsanja nad bankama ovisnosnih stabala.

Razmatranje mogućih pristupa ovisnosnomu parsanju hrvatskih tekstova temeljenom na podatcima dano je u obliku dva eksperimenta s parsanjem hrvatskih tekstova iz Hrvatske ovisnosne banke stabala. Predstavljena je trenutna inačica Hrvatske ovisnosne banke stabala, odnosno njezine opće statističke značajke i značajke smatrane važnima s gledišta ovisnosnoga parsanja te načela njezina dosadašnjega i budućega razvoja.

Prvi eksperiment utvrđuje razinu primjenjivosti pristupa ovisnosnomu parsanju temeljenih na grafovima i pristupa temeljenoga na prijelazničkim sustavima za ovisnosno parsanje hrvatskih tekstova putem vrjednovanja niza značajki njihove točnosti i učinkovitosti s pomoću postavljenih mjera za vrjednovanje na Hrvatskoj ovisnosnoj banci stabala i time je usklađen s prvim od dvaju osnovnih zadataka ovdje predstavljenoga istraživanja. Vrjednovani su ovisnosni parseri izgrađeni sustavom MSTParser za parsanje temeljeno na teoriji grafova i sustavom MaltParser za prijelazničko parsanje. Također je postavljen opći praktični okvir za vrjednovanje točnosti i učinkovitosti ovisnosnoga parsanja na dostupnoj inačici Hrvatske ovisnosne banke stabala. Utvrđena je statistički značajna razlika ukupne točnosti parsanja tekstova iz Hrvatske ovisnosne banke stabala za dva odabrana teorijska okvira, i to u korist ovisnosnoga parsanja temeljenoga na teoriji grafova. Za najbolji parser iz toga istraživanja zabilježena je ukupna točnost parsanja od 74.53% točnih povezivanja pojavnica ovisnosnim relacijama uz dodjelu točne sintaktičke funkcije.

Drugi eksperiment usklađen je stoga s drugim osnovnim zadatkom. Za njegove potrebe izrađen je – za specifične potrebe hrvatskih tekstova i s ciljem povećanja točnosti njihova parsanja – novi model ovisnosnoga parsanja. Taj je model ovisnosnoga parsanja zasnovan na pristupu ovisnosnomu parsanju temeljenom na teoriji grafova i potom proširen povezivanjem s valencijskim rječnikom glagola hrvatskoga jezika CROVALLEX s ciljem povećanja ukupne točnosti ovisnosnoga parsanja hrvatskih tekstova povećanjem točnosti povezivanja glagolskih predikata i samostalnih elemenata rečeničnoga ustroja koje u rečenice hrvatskoga jezika uvode ti predikati. Time je izrađen prototipni model hibridnoga ovisnosnog parsanja hrvatskih tekstova, koji je izveden u obliku prototipnoga računalnog sustava radnoga naziva CroDep. Sustav CroDep vrjednovan je prema okviru za vrjednovanje iz prvoga eksperimenta, prema zadanome formalnom okviru za vrjednovanje točnosti i učinkovitosti i u usporedbi s najboljim prijelazničkim parserom i najboljim parserom temeljenom na grafovima s obzirom na rezultate prvoga eksperimenta. Utvrđena je statistički značajna razlika između točnosti parsanja hibridnim parserom CroDep i svih parsanja ostalim ovisnosnim parserima iz prvoga

eksperimenta, i to u korist hibridnoga pristupa ovisnosnomu parsanju. Zabilježena je ukupna točnost parsanja ovisnosnim parserom CroDep od oko 77.21% točnih povezivanja pojavnica ovisnosnim relacijama uz dodjelu točne sintaktičke funkcije, što predstavlja povećanje od oko 2.68% u odnosu na najbolji parser iz prvoga eksperimenta. To je povećanje uzrokovano značajnim – točnije, u prosjeku najmanje 10-postotnim – povećanjem točnosti ovisnosnoga povezivanja i dodjele sintaktičkih funkcija predikata, subjekata i objekata glagolima i imenicama iz Hrvatske ovisnosne banke stabala. Time je opravdan odabir pristupa zasnovanoga na povezivanju modela ovisnosnoga parsanja temeljenoga na teoriji grafova s valencijskim rječnikom hrvatskih glagola CROVALLEX i izvršen drugi osnovni zadatak postavljen pred prikazano istraživanje.

Postoje mnogi smjerovi za moguća buduća istraživanja ovisnosnoga parsanja hrvatskih tekstova. U eksperiment s Hrvatskom bankom ovisnosnoih stabala i izdvojenim modelima ovisnosnoga parsanja temeljenoga na podatcima mogu se uključiti i modeli parsanja koji nisu obuhvaćeni teorijskim okvirima parsanja temeljenoga na grafovima ili prijelazničkim sustavima, poput modela temeljenoga na uvećavajućim sigmoidnim mrežama povjerenja (en. *incremental sigmoid belief networks*, ISBN) prikazanoga u (Titov i Henderson 2006, 2007) i izvedenoga u obliku parsera IDP[111]. Također se u takav eksperiment mogu uključiti i druge izvedbe parsera temeljenih na grafovima ili prijelazničkih parsera, poput parsera DeSR[112], opisanoga u (Attardi 2006). U ranijim eksperimentima s parsanjem temeljenim na podatcima pokazano je kako se točnost parsanja može povećati glasovanjem i vezivanjem ovisnosnih parsera. Vrijedilo bi za hrvatske tekstove izraditi eksperimente s glasovanjem po uzoru na (Sagae i Lavie 2006) i (Hall i dr. 2007) i eksperimente s vezivanjem raznorodnih ovisnosnih parsera po uzoru na (Nivre i McDonald 2008, 2011). Također se eksperiment s postojećim pristupima ovisnosnomu parsanju može obogatiti razmatranjem razlika između parsanja teksta s lijeva na desno i s desna na lijevo (usp. Passarotti i Dell'Orletta 2010) i lingvistički usmjerenijim odabirom združenih značajki za jezične modele pojedinih parsera temeljenih na podatcima. Moguće je razmotriti i primjenjivost nekih sintaktičko-semantički usmjerenih i na podatcima temeljenih modela povezivanja pojavnica u ovisnosne odnose (usp. Yuret 1998) s gledišta ovisnosnoga parsanja. Hibridizacija modela ovisnosnoga parsanja može se dodatno proširiti: neizravno – razvojem Hrvatske ovisnosne banke stabala i njezinim povezivanjem s rječnikom CROVALLEX putem metoda crpljenja valencijskih okvira glagola iz ovisnosnih

[111] Vidjeti i URL parsera IDP http://cui.unige.ch/~titov/idp/ (2012-04-14).
[112] Vidjeti i URL parsera DeSR https://sites.google.com/site/desrparser/ (2012-04-14).

stabala (usp. Agić i dr. 2010, Šojat i dr. 2010) – ili izravno – razmatranjem uvođenja dodatnih pravila i ograničenja temeljenih na valencijskim okvirima hrvatskih glagola i drugim zapisima jezičnoga znanja u različite razrede algoritama za ovisnosno parsanje. Valencijske značajke glagola također se mogu uključiti i u model značajki prijelazničkoga parsanja. Vrjednovanje točnosti i učinkovitosti ovisnosnoga parsanja može se također proširiti uvođenjem dodatnih mjera za vrjednovanje (Nivre i dr. 2010) i prilagodbom svih raspoloživih mjera točnosti parsanja jezikoslovnomu gledištu.

5 Literatura

Abeillé, A. (ur.) (2003). Treebanks: Building and Using Parsed Corpora. Text, Speech and Language Technology, volume 20, Springer, 2003.

Abney, S. (1996). Partial Parsing via Finite-State Cascades. Journal of Natural Language Engineering, 2(4), 1996, pp. 337-344.

Agić, Ž.; Tadić, M. (2006). Evaluating Morphosyntactic Tagging of Croatian Texts. Proceedings of the 5th International Conference on Language Resources and Evaluation, Genova, ELRA, 2006.

Agić, Ž.; Tadić, M.; Dovedan, Z. (2008) Improving Part-of-Speech Tagging Accuracy for Croatian by Morphological Analysis. Informatica. 32(4), 2008, pp. 445-451.

Agić, Ž.; Tadić, M.; Dovedan, Z. (2009). Evaluating Full Lemmatization of Croatian Texts. Recent Advances in Intelligent Information Systems, Academic Publishing House EXIT, Warsaw, 2009. pp. 175-184.

Agić, Ž.; Šojat, K.; Tadić, M. (2010). An Experiment in Verb Valency Frame Extraction from Croatian Dependency Treebank. Proceedings of the 32nd International Conference on Information Technology Interfaces, SRCE University Computer Centre, University of Zagreb, 2010, pp. 55-60.

Agić, Ž.; Merkler, D.; Berović, D.; Tadić, M. (2011). Development and Applications of the Croatian 1984 Corpus for the Multext-East Resources. Proceedings of The Second Conference on Slavic Corpora (SlaviCorp 2011), Dubrovnik, Croatia (u tisku).

Aho, A. V.; Ullman, J. D. (1972). The Theory of Parsing, Translation and Compiling, Volume I: Parsing. Prentice Hall, Englewood Cliffs, New Jersey, 1972.

Aho, A. V.; Lam, M. S.; Sethi, R; Ullman, J. D. (2006). Compilers: Principles, Techniques, and Tools. Prentice Hall, 2006.

Aitchison, Jean. (1998). The Articulate Mammal: An Introduction to Psycholinguistics. Routledge, 1998.

Anić, V. (2004). Veliki rječnik hrvatskoga jezika. Novi liber, Zagreb, 2004. URL http://hjp.srce.hr (2012-01-20)

Attardi, G. (2006). Experiments with a multilanguage non-projective dependency parser. Proceedings of the 10th Conference on Computational Natural Language Learning (CoNLL), New York, NY, pp. 166-170.

Autebert, J.-M.; Berstel, J.; Boasson, L. (1997). Context-Free Languages and Push-Down Automata. Handbook of Formal Languages, vol. 1, Springer-Verlag, 1997, pp. 111-174.

Barić, E.; Lončarić, M.; Malić, D.; Pavešić, S.; Peti, M.; Zečević, V.; Znika, M. (2003). Hrvatska gramatika. Školska knjiga, Zagreb, 2003.

Barnlund, D. C. (2008). A transactional model of communication. Communication Theory, New Brunswick, New Jersey, 2008, pp. 47-57.

Bekavac, B. (2005). Strojno prepoznavanje naziva u suvremenim hrvatskim tekstovima. Doktorska disertacija, Sveučilište u Zagrebu, Filozofski fakultet, Zagreb, 2005.

Bekavac, B.; Tadić, M. (2007). Implementation of Croatian NERC system. Proceedings of the Workshop on Balto-Slavonic Natural Language Processing (BSNLP 2007), Special Theme: Information Extraction and Enabling. Association for Computational Linguistics (ACL), Prague, pp. 11-18.

Bekavac, B.; Agić, Ž.; Tadić, M. (2009). Interacting Croatian NERC System and Intex/NooJ Environment. Applications of Finite-State Language Processing: Selected Papers from the 2008 International NooJ Conference, Cambridge Scholars Publishing, Newcastle upon Tyne, United Kingdom, pp. 21-29.

Berović, D.; Agić, Ž.; Tadić, M. (2012). Croatian Dependency Treebank: Recent Development and Initial Experiments. Proceedings of The Eighth International Conference on Language Resources and Evaluation (LREC 2012), Istanbul, Turkey (u tisku).

Berović, D.; Merkler, D.; Tadić, M. (2012b). Depedent Clause Annotation in Croatian Dependency Treebanks. Proceedings of The Eighth International Conference on Language Resources and Evaluation (LREC 2012), Istanbul, Turkey (u postupku recenzije).

Bloomfield, L. (1933). Language. Henry Holt, New York, 1933.

Boras, D. (1998). Teorija i pravila segmentacije teksta na hrvatskom jeziku. Doktorska disertacija, Sveučilište u Zagrebu, Filozofski fakultet, 1998.

Boullier, P. (1995.) Yet Another O(n^6) Recognition Algorithm for Mildly Context-Sensitive Languages. Raport de recherche no. 2730, Programme 3, Intelligence artificielle, systemes cognitifs et interaction homme-machine, Projet Atoll, INRIA, 1995.

Brants, T. (2000). TnT - A Statistical Part-of-Speech Tagger. The 6th Applied Natural Language Processing Conference, Association for Computational Lingusitics, Seattle, USA, 2000, pp. 224-231.

Buchholz, S.; Marsi, E. (2006). CoNLL-X Shared Task on Multilingual Dependency Parsing. Proceedings of the 10th Conference on Computational Natural Language Learning (CoNLL), New York, NY, pp. 149-164.

Burnard, L.; Bauman, S. (ur.) (2007). TEI P5: Guidelines for Electronic Text Encoding and Interchange. URL http://www.tei-c.org/Guidelines/P5/ (2012-02-26).

Camerini, P. M.; Fratta, L.; Maffioli, F. (1980). The k-Best Spanning Arborescences of a Network. Networks 10, pp. 91-110.

Carroll, J. (2003). Parsing. The Oxford Handbook of Computational Linguistics, Oxford University Press, 2003, pp. 233-248.

Chang, C. C.; Lin, C. J. (2001). LIBSVM : A Library for Support Vector Machines. URL http://www.csie.ntu.edu.tw/~cjlin/libsvm (2012-03-28).

Chanod, J.-P. (2001). Robust Parsing and Beyond. Robustness in Language and Speech Technology, Kluwer Academic Publishers, 2001, pp. 187.204.

Chomsky, N. A. (1957). Syntactic Structures. Mouton, The Hague, 1957.

Chomsky, N. A. (1959). On certain formal properties of grammars. Inform. Control, vol. 2, 1959, pp. 137-167.

Chu, Y. J.; Liu, T. H. (1965). On The Shortest Arborescence of a Directed Graph. Science Sinica, 1965(14), pp. 1396-1400.

Church, A. (1936). An Unsolvable Problem of Elementary Number Theory. American Journal of Mathematics, 58 (1936), pp. 345–363.

Cocke, J.; Schwartz, J. T. (1970). Programming languages and their compilers: Preliminary notes. Technical report, Courant Institute of Mathematical Sciences, New York University.

Collins, M. (2002). Discriminative Training Methods for Hidden Markov Models: Theory and Experiments With Perceptron Algorithms. Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), Philadelphia, PA, 2002.

Cormen, T. H.; Leiserson, C. E.; Rivest, R. L.; Stein, C. (2009). Introduction to Algorithms. MIT Press i McGraw-Hill, treće izdanje, 2009.

Covington, M. A. (1990). Parsing Discontinuous Constituents in Dependency Grammar. Computational Linguistics, 16(4), 1990, pp. 234-236.

Covington, M. A. (2001). A Fundamental Algorithm for Dependency Parsing. Proceedings of the 39th Annual ACM Southeast Conference, pp. 95-102.

Craig, R. T. (1999). Communication Theory as a Field. Communication Theory, 9(2), International Communication Association, Blackwell Publishing Ltd., 1999, pp. 119-161.

Davis, M. (1965). The Undecidable. Raven Press, Hewlett, New York, 1965.

Davis, M. (1995). Influences of Mathematical Logic on Computer Science. The universal Turing machine: a half-century survey. Springer, 1995.

Dennett, D. C. (1994). The Role of Language in Intelligence. What is Intelligence?, Cambridge University Press, Cambridge, 1994, pp. 161-178.

Dovedan, Z. (2003). Formalni jezici: sintaksna analiza. Zavod za informacijske studije, Zagreb, 2003.

Džeroski, S.; Erjavec, T.; Ledinek, N.; Pajas, P.; Žabokrtský, Z.; Žele, A. (2006). Towards a Slovene Dependency Treebank. Proceedings of the Fifth International Conference on Language Resources and Evaluation, ELRA, Paris-Genoa, 2006.

Earley, J. (1968). An Efficient Context-Free Parsing Algoritm. Doktorska disertacija, Carnegie Mellon University, Pittsburgh, PA, 1968.

Earley, J. (1970). An Efficient Context-Free Parsing Algorithm. Communications of the ACM, vol. 13, no. 2, 1970, pp. 94-102.

Edmonds, J. (1967). Optimum Branchings. Journal of Research of the National Bureau of Standards, 1967(71B), pp. 233-240.

Eisner, J. M. (1996a). An Empirical Comparison of Probability Models for Dependency Grammar. Technical Report IRCS-96-11, Institute for Research in Cognitive Science, University of Pennsylvania.

Eisner, J. M. (1996b). Three New Probabilistic Models for Dependency Parsing: An Exploration. Proceedings of the 16th International Conference on Computational Linguistics (COLING), Copenhagen, Denmark, pp. 340-345.

Erdeljac, V. (2009). Mentalni leksikon: modeli i činjenice. Ibis grafika, Zagreb, 2009.

Erjavec, T. (2004). Multext-East Version 3: Multilingual Morphosyntactic Specifications, Lexicons and Corpora. Proceedings of the Fourth International Conference on Language Resources and Evaluation. ELRA, Lisbon-Paris, 2004, pp. 1535-1538.

Erjavec, T. (2010). MULTEXT-East Version 4: Multilingual Morphosyntactic Specifications, Lexicons and Corpora. Proceedings of the Seventh Conference on International Language Resources and Evaluation, ELRA, Valletta-Paris, 2010, pp. 2544-2547.

Fan, R. E.; Chang, K. W.; Hsieh, C. J.; Wang, X. R.; Lin, C. J. (2008). LIBLINEAR: A Library for Large Linear Classification. Journal of Machine Learning Research 9(2008), pp. 1871-1874.

Frazier, L. (1987). Sentence processing: A tutorial review. Attention and Performance XII: The Psychology of Reading, Lawrence Erlbaum Associates, 1987.

Frege, G. (1879). Begriffsschrift: eine der arithmetischen nachgebildete Formelsprache des reinen Denkens. Halle, 1879.

Grune, D.; Jacobs, C. (1998). Parsing Techniques: A Practical Guide. Ellis Horwood Ltd., Chichester, England, 1998.

Hajič, J.; Panevová, J.; Buráňová, E.; Urešová, Z.; Bémová, A. (1999). Annotations at Analytical Level: Instructions for Annotators. UK MFF ÚFAL, Praha, Czech Republic. URL http://ufal.mff.cuni.cz/pdt2.0/doc/manuals/en/a-layer/pdf/a-man-en.pdf (2012-03-18).

Hajič, J.; Böhmová, A.; Hajičová, E.; Vidová Hladká, B. (2000). The Prague Dependency Treebank: A Three-Level Annotation Scenario. Treebanks: Building and Using Parsed Corpora, Amsterdam, Kluwer, 2000.

Hajič, J.; Ciaramita, M.; Johansson, R.; Kawahara, D.; Marti, M. A.; Marquez, L.; Meyers, A.; Nivre, J.; Pado, S.; Štepanek, J.; Stranak, P.; Surdeanu, M.; Xue, N.; Zhang, Y. (2009). The CoNLL-2009 Shared Task: Syntactic and Semantic Dependencies in Multiple Languages. Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL): Shared Task, ACL, Boulder, Colorado, 2009, pp. 1–18.

Hall, K. B. (2007). K-best Spanning Tree Parsing. Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, 2007, pp. 392-399.

Hall, J.; Nilsson, J.; Nivre, J.; Eryigit, G.; Megyesi, B,; Nilsson, M.; Saers, M. (2007). Single Malt or Blended? A Study in Multilingual Parser Optimization. Proceedings of the CoNLL Shared Task of EMNLP-CoNLL 2007, Prague, Czech Republic, 2007.

Harris, Z. S. (1951). Methods in structural linguistics. University of Chicago Press, Chicago, 1951.

Hopcroft, J. E.; Motwani, R.; Ullman, J. D. (2006). Introduction to Automata Theory, Languages, and Computation. Addison-Wesley, 2006.

Hudeček, L.; Mihaljević, M. (2009). Homonimija kao leksikografski problem. Rasprave Instituta za hrvatski jezik i jezikoslovlje, 35, 2009.

Hurford, J. R.; Heasley, B. (1983). Semantics: a coursebook, Cambridge University Press, 1983.

Ide, N.; Bonhomme, P.; Romary, L. (2000). XCES: An XML-based Encoding Standard for Linguistic Corpora. Proceedings of the Second International Language Resources and Evaluation Conference. European Language Resources Association (ELRA), Paris, 2000, pp. 825-830.

Jurafsky, D.; Martin, J. H. (1999). Speech and Language Processing – An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition. Prentice Hall Series in Artificial Intelligence. Prentice Hall, Englewood Cliffs, New Jersey, 1999.

Kay, M. (1986). Algorithm schemata and data structures in syntactic processing. Readings in Natural Language Processing, Morgan Kaufmann, Los Altos, 1986, pp. 35-70.

Kallmeyer, L. (2010). Parsing Beyond Context-Free Grammars. Springer-Verlag, Berlin, Heidelberg, 2010.

Karttunen, L.; Beesley, K. R. (2005). Twenty-Five Years of Finite-State Morphology. Inquiries Into Words: a Festschrift for Kimmo Koskenniemi on his 60th Birthday, 2005, pp. 71-83.

Kasami, T.; Torii, K. (1969). A syntax-analysis procedure for unambiguous context-free grammars. Journal of the ACM, vol. 16, no. 3, 1969, pp. 423-431.

Kleene, S. C. (1952). Introduction to Metamathematics. North-Holland, 1952.

Kroeger, P. R. (2005). Analyzing Grammar: An Introduction. Cambridge Textbooks in Linguistics, Cambridge University Press, Cambridge, 2005.

Kruskal, J. B. (1956). On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem. Proceedings of the American Mathematical Society, 7(1), pp. 48-50.

Kudo, T.; Matsumoto, Y. (2000). Japanese Dependency Structure Analysis Based on Support Vector Machines. Proceedings of the Joint SIGDAT Conference on Empirical Methods in NLP and Very Large Corpora, Hong Kong, pp. 18-25.

Kübler, S.; McDonald, R.; Nivre, J. (2009). Dependency Parsing. Synthesis Lectures on Human Language Technologies, lecture 2. Morgan i Claypool Publishers, 2009.

Kurzweil, R. (1992). The Age of Intelligent Machines. The MIT Press, 1992.

Lange, M.; Leiß, H. (2009). To CNF or not to CNF? An Efficient Yet Presentable Version of the CYK Algorithm. Informatica Didactica 8, 2009, pp. 1-21.

Lauc, T. (2001). Problemi obrade prirodnoga jezika u sustavima za pretraživanje obavijesti putem pretraživanja punoga teksta na hrvatskome književnom jeziku. Doktorska disertacija, Sveučilište u Zagrebu, Filozofski fakultet, Zagreb, 2001.

Ledinek, N.; Žele, A. (2005). Building of the Slovene Dependency Treebank Corpus According to the Prague Dependency Treebank Corpus. Grammar and Corpus Conference, Prague, Czech Republic, 2005.

Lee, L. (2002). Fast Context-Free Grammar Parsing Requires Fast Boolean Matrix Multiplication. Journal of the ACM 49 (1), 2002, pp. 1-15.

Lewis, R. (1999). Specifying architectures for language processing: Process, control, and memory in parsing and interpretation. Architecutres and mechanisms for language processing, Cambridge University Press, 1999.

Lopatková, M.; Žabokrtský, Z.; Skwarska, K. (2006). Valency Lexicon of Czech Verbs: Alternation-Based Model. Proceedings of the Fifth International Conference on Language Resources and Evaluation – LREC 2006, volume 3, pp. 1728-1733.

Manning, C. D.; Schütze, H. (2003). Foundations of Statistical Natural Language Processing. The MIT Press, Cambridge, Massachusetts, 2003.

Marcus, M. P.; Santorini, B.; Marcinkiewicz, M. A. (1993). Building a Large Annotated Corpus of English: The Penn Treebank. Computational Linguistics, 19(2), 1993, pp. 313-330.

Martin, J. (2002). Introduction to Languages and the Theory of Computation. McGraw-Hill Higher Education, 2002.

Martin, W. A.; Church, K. W.; Patil, R. (1987). Preliminary analysis of a breadth-first parsing algorithm: Theoretical and experimental results. Natural Language Parsing Systems. Springer Verlag, Berlin (i MIT LCS technical report TR-261).

McDonald, R.; Crammer, K.; Pereira, F. (2005a). Online Large-Margin Training of Dependency Parsers. Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics, ACL, 2005.

McDonald, R.; Pereira, F.; Ribarov, K.; Hajič, J. (2005b) Non-projective Dependency Parsing using Spanning Tree Algorithms. Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP), ACL, pp. 523-530.

McDonald, R. Pereira, F. (2006). Online Learning of Approximate Dependency Parsing Algorithms. Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics, EACL, pp. 81-88.

McDonald, R.; Lerman, K.; Pereira, F. (2006). Multilingual Dependency Parsing with a Two-Stage Discriminative Parser. Tenth Conference on Computational Natural Language Learning (CoNLL-X), 2006.

McDonald, R.; Nivre, J. (2007). Characterizing the Errors of Data-Driven Dependency Parsing Models. Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, ACL, Prague, pp. 122-131.

McDonald, R.; Nivre, J. (2011). Analyzing and Integrating Dependency Parsers. Computational Linguistics 37(1), pp. 197-230.

McEnery, T.; Wilson, A. (2001). Corpus Linguistics: An Introduction. Edinburgh Textbooks in Empirical Linguistics, Edinburgh University Press, Edinburgh, 2001.

Mejer, A.; Crammer, K. (2010). Confidence in Structured-Prediction using Confidence-Weighted Models. Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, pp. 971-981.

Mikelić Preradović, N. (2008). Pristupi izradi strojnog tezaurusa za hrvatski jezik. Doktorska disertacija, Sveučilište u Zagrebu, Filozofski fakultet, Zagreb, 2008.

Mikelić Preradović, N.; Boras, D.; Kišiček, S. (2009). CROVALLEX: Croatian Verb Valence Lexicon. Proceedings of the ITI 2009 31st International Conference on Information Technology Interfaces. SRCE, Zagreb, pp. 533-538.

Miller, K. (2005). Communication Theories: Perspectives, Processes, and Contexts. McGraw-Hill Higher Education, Boston, Massachussets, 2005.

Nakagawa, T. (2007). Multilingual Dependency Parsing Using Gibbs Sampling. Proceedings of the CoNLL Shared Task of EMNLP-CoNLL 2007, Prague, Czech Republic, 2007.

Nilsson, J.; Nivre, J. (2008). MaltEval: an Evaluation and Visualization Tool for Dependency Parsing. Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08), European Language Resources Association (ELRA), Paris-Marrakech, 2008, pp. 161-166.

Nivre, J. (2003). An Efficient Algorithm for Projective Dependency Parsing. Proceedings of the 8th International Workshop on Parsing Technologies (IWPT), Nancy, France, pp. 149-160.

Nivre, J.; Nilsson, J. (2005). Pseudo-Projective Dependency Parsing. Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL), Ann Arbor, MI, pp. 99-106.

Nivre, J. (2006). Inductive Dependency Parsing. Text, Speech and Language Technology, volume 34. Springer, Dordrecht, The Netherlands, 2006.

Nivre, J. (2006b). Constraints on non-projective dependency graphs, Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL),Trento, Italy, pp. 73-80.

Nivre, J.; Hall, J.; Nilsson, J.; Eryigit, G.; Marinov, S. (2006). Labeled Pseudo-Projective Dependency Parsing With Support Vector Machines. Proceedings of the 10th Conference on Computational Natural Language Learning (CoNLL), New York, NY, pp. 221-225.

Nivre, J. (2007). Incremental Non-Projective Dependency Parsing. Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL HLT), Rochester, NY, pp. 396-403.

Nivre, J.; Hall, J.; Kübler, S.; McDonald, R.; Nilsson, J.; Riedel, S.; Yuret, D. (2007a). The CoNLL 2007 Shared Task on Dependency Parsing. Proceedings of the CoNLL Shared Task of EMNLP-CoNLL 2007, Prague, Czech Republic, pp. 915-932.

Nivre, J.; Hall, J.; Nilsson, J.; Chanev, A.; Eryigit, G.; Kübler, S.; Marinov, S.; Marsi, E. (2007b). MaltParser: A Language-Independent System for Data-Driven Dependency Parsing. Natural Language Engineering, 13(2), 95-135.

Nivre, J. (2008). Algorithms for Deterministic Incremental Dependency Parsing. Computational Linguistics, 34(4), pp. 513-553.

Nivre, J.; McDonald, R. (2008). Integrating Graph-Based and Transition-Based Dependency Parsers. Proceedings of ACL-08: HLT, Association for Computational Linguistics, Columbus, Ohio, USA, pp. 950-958.

Nivre, J.; Rimell, L.; McDonald, R.; Gómez Rodríguez, C. (2010) Evaluation of Dependency Parsers on Unbounded Dependencies. Proceedings of the 23rd International Conference on Computational Linguistics (COLING 2010), pp. 833-841.

Nivre, J.; Hall, J. (2012). A Quick Guide to MaltParser Optimization. Vidjeti URL http://maltparser.org/userguide.html#opt (2012-03-26)

Pajas P. (2000). Tree Editor TrEd, Prague Dependency Treebank, Charles University, Prague. Vidjeti URL http://ufal.mff.cuni.cz/~pajas/tred (2012-03-18).

Palmer, F. R. (1981). Semantics. Cambridge University Press, Cambridge, 1981.

Passarotti, M.; Dell'Orletta, F. (2010). Improvements in Parsing the Index Thomisticus Treebank: Revision, Combination and a Feature Model for Medieval Latin. Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC 2010). La Valletta, Malta, ELRA, 2010, pp. 1964-1971.

Piantadosi, S. T.; Tily, H.; Gibson, E. (2012). The communicative function of ambiguity in language. Cognition, 122(3), 2012, pp. 280-291.

Prim, R. C. (1957). Shortest Connection Networks and Some Generalizations. Bell System Technical Journal, 36(1957), pp. 1389-1401.

Przepiórkowski, A. (2008). TEI P5 as an XML Standard for Treebank Encoding. Proceedings of of the Eighth International Workshop on Treebanks and Linguistic Theories (TLT8), Milan, Italy, pp. 149-160.

Raffaelli, I. (2008). Neka načela ustroja polisemnih leksema. Filologija, 48, 2008, pp. 135 - 172.

Raffaelli, I. (2009). Značenje kroz vrijeme : poglavlja iz dijakronijske semantike. Disput, Zagreb, 2009.

Rayner, K.; Carlson, M.; Frazier, L. (1983). The interaction of syntax and semantics during sentence processing: Eye movements in the analysis of semantically biased sentences. Journal of verbal learning and verbal behavior 22(3), 1983, pp. 358-374.

Rehbein, I.; van Genabith, J.(2007). Treebank Annotation Schemes and Parser Evaluation for German. Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Prague, Czech Republic, 2007, pp. 630–639.

Rijsbergen, C. J. (1979). Information Retrieval. Butterworths, London, 1979.

Russell, S.; Norvig, P. (2009). Artificial Intelligence: A Modern Approach. Pearson Education, Prentice Hall, 2009.

Sagae, K.; Lavie, A. (2006). Parser Combination by Reparsing. Proceedings of The Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics (HLT/NAACL).

Sagae, K.; Tsujii, J. (2007). Dependency Parsing and Domain Adaptation With LR models and Parser Ensembles. Proceedings of the CoNLL Shared Task of EMNLP-CoNLL 2007, Prague, Czech Republic.

Sakai, I. (1962). Syntax in Universal Translation. Proceedings of the 1961 International Conference on Machine Translation of Languages and Applied Language Analysis, Her Majesty's Stationery Office, London, 1962, pp. 593-608.

Samuelsson, C.; Wirén, M. (2000). Parsing Techniques. Handbook of Natural Language Processing, Marcel Dekker, 2000, pp. 59-91.

Saussure, F. (1916). Cours de linguistique générale. Payot, Lausanne-Paris, 1916. (prijevod Baskin, W. (1977). Course in General Linguistics, Fontana/Collins, Glasgow, UK)

Saussure, F. (2002) Écrits de linguistique générale Gallimard, Paris, 2002. (prijevod Writings in General Linguistics, Oxford University Press, Oxford, 2006)

Savitch, W. J.; Bach, E.; Marxh, W.; Safran-Naveh, G. (ur.) (1987). The Formal Complexity of Natural Language. Studies in Linguistics and Philosophy. Reidel, Dordrecht, Holland, 1987.

Seljan, S. (2003). Leksičko-funkcionalna gramatika hrvatskoga jezika: teorijski i praktični modeli. Doktorska disertacija, Sveučilište u Zagrebu, Filozofski fakultet, Zagreb, 2008.

Sgall, P.; Hajičová, E.; Panevová, J. (1986). The Meaning of the Sentence in Its Semantic and Pragmatic Aspects. Dordrecht, D. Reidel Publishing Company.

Silberztein, M. (1999). Text Indexing with INTEX. Computers and the Humanities 33(3), Kluwer Academic Publishers.

Silberztein, M. (1999b). INTEX: A Finite State Transducer Toolbox. Theoretical Computer Science 231(1), Elsevier Science.

Silberztein, M. (2004). NooJ: An Object-Oriented Approach. INTEX pour la Linguistique et le Traitement Automatique des Langues, Cahiers de la MSH Ledoux. Presses Universitaires de Franche-Comté, pp. 359-369.

Silić, J.; Pranjković, I. (2005). Gramatika hrvatskoga jezika za gimnazije i visoka učilišta. Školska knjiga, Zagreb, 2005.

Sipser, M. (1997). Introduction to the Theory of Computation. PWS Publishing, 1997.

Skok, P. (1955). O sufiksima -isati, -irati i -ovati. Jezik: časopis za kulturu hrvatskoga književnog jezika, volume 4, no. 2, 1955, pp. 36-43.

Slonneger, K.; Kurtz, B. (1995). Formal Syntax and Semantics of Programming Languages. Addison Wesley Longman, 1995.

Srbljić, S. (2000). Jezični procesori 1: Uvod u teoriju formalnih jezika, automata i gramatika. Element, Zagreb, 2000.

Surdeanu, M.; Johansson, R.; Meyers, A.; Marquez, L.; Nivre, J. (2008). The CoNLL-2008 Shared Task on Joint Parsing of Syntactic and Semantic Dependencies. Proceedings of the

12th Conference on Computational Natural Language Learning, ACL, Manchester, 2008, pp. 159-177.

Šnajder, J. (2010). Morfološka normalizacija tekstova na hrvatskome jeziku za dubinsku analizu i pretraživanje informacija. Doktorska disertacija. Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva, Zagreb, 2010.

Šojat, K. (2008). Sintaktički i semantički opis glagolskih valencija u hrvatskom. Doktorska disertacija, Sveučilište u Zagrebu, Filozofski fakultet, Zagreb, 2008.

Šojat, K.; Agić, Ž.; Tadić, M. (2010). Verb Valency Frame Extraction Using Morphological and Syntactic Features of Croatian. Proceedings of the 7th International Conference on Formal Approaches to South Slavic and Balkan Languages – FASSBL 7, Croatian Language Technologies Society, Faculty of Humanities and Social Sciences, Zagreb, 2010, pp. 119-126.

Štefanec, V.; Vučković, K.; Dovedan, Z. (2010). Towards Parsing Croatian Complex Sentences: Dependent Noun Clauses. NooJ 2010 International Conference and Workshop.

Tadić, M. (1994). Računalna obradba morfologije hrvatskoga književnoga jezika. Doktorska disertacija. Sveučilište u Zagrebu, Filozofski fakultet, Zagreb, 1994.

Tadić, M. (2000). Building the Croatian-English Parallel Corpus. Proceedings of the Second International Conference on Language Resources and Evaluation, ELRA, Paris-Athens 2000, pp. 523-530.

Tadić, M. (2002). Building the Croatian National Corpus. Proceedings of the Third International Conference on Language Resources and Evaluation (LREC2002), Paris-Las Palmas, ELRA, pp. 441-446.

Tadić, M. (2003). Jezične tehnologije i hrvatski jezik. Ex Libris znanstvena knjižnica, 3. knjiga, Ex Libris, Zagreb, 2003.

Tadić, M.; Fulgosi, S. (2003). Building the Croatian Morphological Lexicon. Proceedings of the EACL2003 Workshop on Morphological Processing of Slavic Languages, ACL, Budapest, 2003, pp. 41-46.

Tadić, M. (2005). The Croatian Lemmatization Server. Southern Journal of Linguistics. 29, 1/2, 2005, pp. 206-217.

Tadić, M. (2006). Croatian Lemmatization Server. Formal Approaches to South Slavic and Balkan Languages, Bulgarian Academy of Sciences, Sofia, 2006, pp. 140-146.

Tadić, M. (2006b). Croatian Dependecy Treebank in Multilingual Context. Readings in Multilinguality, Selected papers for young researchers, Bulgarian Academy of Science, Sofia, Bulgaria, pp. 125-128.

Tadić, M. (2007). Building the Croatian Dependency Treebank: the initial stages. Suvremena lingvistika, 63, pp. 85-92.

Tadić, M. (2009). New Version of the Croatian National Corpus. After Half a Century of Slavonic Natural Language Processing. Masaryk University, Brno, 2009, pp. 199-205.

Tarjan, R. E. (1977). Finding Optimum Branchings. Networks, 1977(7), pp. 25-35.

Tesnière, L. (1959). Éléments de syntaxe structurale. Editions Klincksieck.

Titov, I.; Henderson, J. (2006). Porting Statistical Parsers With Data-Defined Kernels. Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X).

Titov, I.; Henderson, J. (2007). Fast and Robust Multilingual Dependency Parsing With a Generative Latent Variable Model. Proceedings of the CoNLL Shared Task of EMNLP-CoNLL 2007, Prague, Czech Republic.

Tomita, M. (1986). Efficient Parsing for Natural Language: A Fast Algorithm for Practical Systems. Kluwer Academic Publishers, Boston, 1986.

Tomlin, R. (1986). Basic Word Order: Functional Principles. Croom Helm, London, 1986.

Townsend, D. J.; Bever, T. G. (2001). Sentence Comprehension: The Integration of Habits and Rules. MIT Press, 2001.

Trueswell, J.; Tanenhaus, M. (1994). Toward a lexical framework of constraint-based syntactic ambiguity resolution. Perspectives on sentence processing, 1994, pp. 155-179.

Turing, A. M. (1936). On Computable Numbers, with an Application to the Entscheidungsproblem. Proceedings of the London Mathematical Society, 2 42, pp. 230–265.

Turing, A. M. (1938). On Computable Numbers, with an Application to the Entscheidungsproblem: A correction" Proceedings of the London Mathematical Society 43, pp. 544–546.

Turing, A. M. (1948). Intelligent Machinery. Cybernetics: Key Papers, University Park Press, Baltimore, 1968.

Turing, A. M. (1950). Computing Machinery and Intelligence. Mind LIX (236), 1950, pp. 433–460.

Unger, S. H. (1968). A Global Parser for Context-Free Phrase Structure Grammars. Communications of the ACM, vol. 11, no. 4, 1968, pp. 240-247.

Vučković, K.; Tadić, M.; Dovedan, Z. (2008). Rule Based Chunker for Croatian. Proceedings of the Sixth International Conference on Language Resources and Evaluation, Marrakech-Paris, ELRA, 2008.

Vučković, K. (2009). Model parsera za hrvatski jezik. Doktorska disertacija. Sveučilište u Zagrebu, Filozofski fakultet, Zagreb, 2009.

Vučković, K.; Agić, Ž.; Tadić, M. (2010). Improving Chunking Accuracy on Croatian Texts by Morphosyntactic Tagging. Proceedings of the Seventh International Conference on Language Resources and Evaluation, Valletta, European Language Resources Association, 2010. pp. 1944-1949.

Vučković, K.; Bekavac, B.; Dovedan, Z. (2010b). Improved Parser for Simple Croatian Sentences. NooJ 2010 International Conference and Workshop.

Vučković, K.; Bekavac, B.; Dovedan, Z. (2010c). SynCro - Parsing Simple Croatian Sentences. Finite State Language Engineering: NooJ 2009 International Conference and Workshop, Touzeur, Centre de Publication Universitaire, 2010. pp. 207-217.

Vučković, K.; Tadić, M.; Bekavac, B. (2010d). Croatian Language Resources for NooJ. CIT – Journal of Computing and Information Technology. 18(2010), pp. 295-301.

Vučković, K.; Mikelić Preradović, N.; Dovedan, Z. (2010e). Verb Valency Enhanced Croatian Lexicon. Applications of Finite-State Language Processing, Selected Papers from the 2008 NooJ Conference. Cambridge Scholars Publishing, 2010, pp. 52-60.

Wallis, S. (2008). Searching Treebanks and Other Structured Corpora. Corpus Linguistics: An International Handbook, Handbücher zur Sprache und Kommunikationswissenschaft, Mouton de Gruyter, Berlin, 2008, poglavlje 36.

Yamada, H.; Matsumoto,Y. (2003). Statistical Dependency Analysis with Support Vector Machines. Proceedings of the 8th International Workshop on Parsing Technologies (IWPT), Nancy, France, pp. 195-206.

Younger, D. H. (1967). Recognition and parsing of context-free languages in time n^3. Information and Control 10(2), 1967, pp. 189-208.

Yuret, D. (1998). Discovery of Linguistic Relations Using Lexical Attraction. Doktorska disertacija, Massachussets Institute of Technology, 1998.

Zeman, D. (2002). Can Subcategorization Help a Statistical Dependency Parser? Proceedings of the 19th international conference on Computational linguistics – COLING 2002, volume 1, Association for Computational Linguistics, Stroudsburg, PA, USA, 2002, pp. 1-7.

Zimmer, B. (2010). On Language: Crash Blossoms. New York Times Magazine, 27.1.2010.

Žic-Fuchs, M. (1991). Znanje o jeziku i znanje o svijetu – semantička analiza glagola kretanja u engleskom jeziku. Filozofski fakultet, Zagreb, 1991.

Sažetak

Parsanje tekstova prirodnoga jezika – u presjeku teorijskih okvira onih znanstvenih disciplina koje sačinjavaju interdisciplinarno znanstveno područje jezičnih tehnologija – definira se kao strojna sintaktička analiza pojedinih rečenica tih tekstova, odnosno kao strojni postupak jednoznačnoga otkrivanja sintaktičkih uloga pojedinih njihovih riječi u izgradnji osnovnih elemenata rečeničnoga ustroja – poput rečeničnih predikata, subjekata i objekata – prema nekoj unaprijed zadanoj sintaktičkoj teoriji. Korisnost parsanja tekstova prirodnoga jezika očituje se u rješavanju niza problema obradbe prirodnoga jezika, od pronalaženja značenjskih odnosa u tekstovima do statističkoga strojnog prevođenja, a također i pri pronalaženju obavijesti i u proučavanju svojstava prirodnih jezika. U ovome istraživanju razmotreni su neki pristupi ovisnosnomu parsanju tekstova hrvatskoga jezika temeljeni na podatcima, odnosno neki pristupi strojnoj sintaktičkoj analizi hrvatskih tekstova prema implicitnome teorijskom modelu sintakse hrvatskoga jezika temeljenome na uspostavljanju ovisnosnih odnosa među elementima rečeničnoga ustroja i unutar njih te sadržanome u sintaktički obilježenome korpusu tekstova hrvatskoga jezika. Postavljena je definicija parsanja kao problema strojne obradbe prirodnoga jezika i kao problema oponašanja ljudske inteligencije računalnim postupcima. Preko teorijskoga okvira formalnih jezika i postavljanja općih kriterija za vrjednovanje postupaka parsanja izložena je definicija ovisnosnoga parsanja temeljenoga na podatcima i predstavljeni su neki pristupi rješavanju toga problema – modeli ovisnosnoga parsanja temeljeni na teoriji grafova i modeli temeljeni na prijelazničkim sustavima. Opisan je i izveden hibridni pristup ovisnosnomu parsanju hrvatskih tekstova, temeljen na teoriji grafova i naknadnome vrjednovanju predloženih rješenja povezivanjem s valencijskim rječnikom glagola hrvatskoga jezika CROVALLEX. Korištenjem Hrvatske ovisnosne banke stabala i definiranjem mjera za vrjednovanje točnosti i učinkovitosti parsera postavljeno je okruženje za vrjednovanje ovisnosnoga parsanja hrvatskih tekstova, i to parserima iz okvira teorije grafova i okvira prijelazničkih sustava te za vrjednovanje novopredloženoga hibridnog pristupa u usporedbi s prethodnima. Za hibridni pristup zabilježena je ukupna točnost ovisnosnoga parsanja od oko 77.21% točnih povezivanja riječi ovisnosnim relacijama uz dodjelu točne sintaktičke funkcije, odnosno statistički značajno povećanje od oko 2.68% u odnosu na najbolji postojeći model ovisnosnoga parsanja.

Ključne riječi: ovisnosno parsanje, ovisnosna sintaksa, hrvatski jezik, Hrvatska ovisnosna banka stabala, sustavi temeljeni na podatcima, hibridni pristup, jezične tehnologije.

Abstract

In the formal framework of language technologies – and the formal frameworks of respective scientific disciplines comprising it – natural language text parsing is defined as automatic syntactic analysis of its sentences or as an algorithmic procedure for unambiguous detection of syntactic roles of words in the construction of basic grammatical structures – such as sentence predicates, subjects and objects – with respect to a previously defined syntactic formalism of that specific natural language. Usefulness of natural language text parsing is reflected today in many other natural language processing tasks – such as question answering, semantic role detection and statistical machine translation – as well as in information retrieval and extraction, data mining and language research in general. This thesis investigated several approaches to data-driven dependency parsing of Croatian texts, i.e. approaches to automatic syntactic analysis of sentences written in Croatian in accordance with a predefined word-dependency-based computational model of Croatian syntax contained implicitly within a corpus of syntactically annotated Croatian texts. Parsing was firstly defined as a problem in the general domains of natural language processing and computational intelligence. By using the formal language theory framework and by defining general formal requests and evaluation criteria for natural language parsing, the problem of data-driven dependency parsing of natural language text was introduced. Two state-of-the-art general approaches to data-driven dependency parsing were described in detail, namely, graph theory based dependency parsing and transition based dependency parsing. A novel approach was envisioned and implemented specifically for dependency parsing of Croatian text by using the Croatian Dependency Treebank and a valency lexicon of Croatian verbs CROVALLEX. The approach was based on linking a graph-based data-driven dependency parser with the valency lexicon by re-ranking k-best dependency trees suggested by the data-driven module on basis of valency information encoded within the lexicon. An experiment was implemented by using the Croatian Dependency Treebank and defining a set of metrics for the evaluation of parsing accuracy and efficiency. The suggested hybrid parsing system scored the highest labeled attachment score (LAS) within the experiment, accurately parsing approximately 77.21% wordforms from the treebank. These scores were further shown to be significantly different, i.e. at least 2.68% higher than the highest scores for any of the data-driven parsing systems.

Keywords: dependency parsing, data-driven parsing, dependency syntax, Croatian language, Croatian Dependency Treebank, hybrid approach, language technologies.

Životopis

Željko Agić rođen je 16. travnja 1983. godine u Splitu. U Trogiru je pohađao osnovnu školu, a u Splitu III. gimnaziju, prirodoslovno-matematičkoga usmjerenja. Diplomirao je 2005. godine na studiju računarstva Fakulteta elektrotehnike, strojarstva i brodogradnje Sveučilišta u Splitu, diplomskim radom "Morfosintaktičko označavanje hrvatskoga jezika". Od 31. prosinca 2006. godine zaposlen je na Odsjeku za informacijske i komunikacijske znanosti Filozofskoga fakulteta Sveučilišta u Zagrebu kao znanstveni novak na znanstvenome projektu "Računalna sintaksa hrvatskoga jezika" pod vodstvom prof. dr. sc. Zdravka Dovedana Hana, u sklopu znanstvenoga programa "Računalnolingvistički modeli i jezične tehnologije za hrvatski jezik" pod vodstvom prof. dr. sc. Marka Tadića s Odsjeka za lingvistiku Filozofskoga fakulteta Sveučilišta u Zagrebu. Iste godine upisuje i poslijediplomski doktorski studij Odsjeka za informacijske i komunikacijske znanosti. Aktivno sudjeluje u izvođenju nastave i vođenju završnih i diplomskih radova na preddiplomskim i diplomskim studijima Odsjeka za informacijske i komunikacijske znanosti. Bio je stručni suradnik na EU FP7 i ICT-PSP znanstvenim projektima ACCURAT, CESAR, CLARA, CLARIN, Let'sMT! i XLike pri Sveučilištu u Zagrebu i pod vodstvom prof. dr. sc. Marka Tadića. Sudjelovao je u organizaciji i radu niza konferencija s područja jezičnih tehnologija i informacijskih znanosti. Pohađao je škole ESSLLI 2008, ESSLLI 2009 i CLARA 2010 i sudjelovao u izvođenju škole CLARA Career Course 2011 kao predavač. U suautorstvu je objavio dvadesetak znanstvenih radova na znanstvenim skupovima i u časopisima s međunarodnom recenzijom. Cjeloviti popis radova dostupan je na URL-u http://bib.irb.hr/lista-radova?autor=291312.