UNIVERZITET U BEOGRADU

PRIRODNO-MATEMATIČKI FAKULTET

Gordana Pavlović-Lažetić

BAZE PODATAKA I EKSPERTNI SISTEMI U UPRAVLJANJU TEKSTOM

DOKTORSKA DISERTACIJA

BEOGRAD 1987.

SADRŽAJ:

1. UVOD

Savremena generacija sistema za upravljanje bazama podataka (SUBP) je, uz sva svoja dostignuća, projektovana za rad sa podacima primitivnih tipova kao što su brojevi i niske karaktera [16]. Uvodjenje relacionog koncepta [10] uslovilo je razvoj generacije sistema potpune funkcionalnosti, koji pružaju korisniku mogućnosti direktnog pristupa podacima. Kao rezultat, omogućene su mnoge aplikacije ne-tradicionalne obrade podataka sa mnogo izraženijom semantičkom strukturom nego što su primitivni tipovi podataka.

Postoji nekoliko pristupa proširenju mogućnosti relacionih sistema za upravljanje bazama podataka (RSUBP) na rad sa podacima koje karakteriše viši nivo semantike. Jedan od tih pristupa je podrška opšteg apstraktnog tipa podataka u RSUBP, tako da se korisniku omogući lako definisanje novih tipova. Ovaj pristup oslobadja SUBP od "razumevanja" semantike tipova podataka koje korisnik definiše, kao i od evaluacije operatora nad tim podacima, koja se pomera u aplikativne programe.

Drugi pristup je proširenje upitnog jezika i odgovarajućeg procesora tako da SUBP direktno podržava neke specifične, neprimitivne tipove podataka koji su često u upotrebi; tekstovi i geometrijski podaci su očigledni kandidati za ovakvu podršku. Direktna nadgradnja kompleksnih podataka u SUBP ima očigledne prednosti, od kojih je najznačajnija efikasnost.

Problem kojim se ova teza bavi je proširenje domena podrške RSUBP-a na tekstuelne podatke. Istorijski, neke od važnih ideja u strukturama za smeštaj podataka i upravljanju bazama podataka dolaze iz oblasti pretraživanja informacija i srodnih aplikacija. Na primer, sekundarno indeksiranje i upitni jezici su prvi put

predloženi i korišćeni u kontekstu bibliografskog traženja na vezi (engl. "on-line"). Stoga se prirodno nameće potreba da se u bazama podataka upravljanje tekstuelnim podacima reši na kompletniji i efikasniji način od onog koji nudi reprezentacija teksta niskama karaktera, koja rezultuje gotovo potpunim gubitkom njegovog semantičkog sadržaja.

S obzirom da je reč prirodni atom teksta, prvi korak u upravljanju tekstom kao podacima je obezbediti zadovoljavajuću obradu na nivou reči. Stoga se tekstuelni podaci tretiraju leksički, tj. podatak je reč okarakterisana svojim bitnim morfološkim, sintaksnim i semantičkim svojstvima.

Nad rečima kao morfološko-sintaksno-semantičkim entitetima izgradjuju se aplikacije inteligentne obrade teksta. Zato su osnovni operatori nad rečima - leksički operatori (za razliku od operatora nad niskama simbola). Oni tretiraju odnose izmedju reči i način upotrebe reči. Na primer, "ljudi" je imenica u množini sa korenom reči "čovek". "Imenica", "množina", "čovek" su vrednosti različitih leksičkih operatora primenjenih na reč "ljudi". U radu (glava 2), definiše se leksički tip podataka u relacionoj bazi podataka, kao specifična reprezentacija reči kodirana celim brojem fiksne dužine, zajedno sa klasom operatora nad rečima. Za kodiranje reči potreban je rečnik posebne strukture i skup morfoloških pravila za prepoznavanje oblika reči. Kako skup leksičkih podataka treba da podržava unapred zadati skup leksičkih operatora, dobar izbor za strukturu leksičkog podatka je kompozicija vrednosti leksičkih operatora nad tim podatkom. Na taj način leksički tip ne samo da obezbedjuje kompresiju podataka (tj. efikasno čuvanje reči u relacionoj bazi podataka), već čini eksplicitnim njihova leksička svojstva. Leksički tip podataka objavljen je u radu [49].

Nad leksičkim tipom podataka moguće je izgraditi hijerarhiju operatora. Na najnižem nivou hijerarhije nalaze se osnovni leksički operatori neposredno podržani leksičkim tipom. Od ovih operatora moguće je, na sledećem nivou, komponovati operatore kojima se izvršavaju složene radnje nad elementima i skupovima leksičkog tipa, a koje odgovaraju važnim operacijama nad tekstom.

Jedan takav operator je razrešavanje leksičke višeznačnosti (RAMB - Resolving AMBiguity), koji se primenjuje pri izgradnji "leksikalizovanog" teksta, tj. teksta predstavljenog leksičkim tipom.

Primer leksičke višeznačnosti je reč "radi" koja ima bar dva udaljena značenja. Jasno je da je taj isti niz simbola potrebno preslikati u jedan od dva različita leksička podatka, u zavisnosti od nameravanog značenja koje se najčešće može izvesti iz konteksta. Operator RAMB preslikava par skupova leksičkih podataka u jedan leksički podatak. Svi leksički podaci iz prvog skupa domena odgovaraju istom nizu simbola (leksički višeznačnoj reči), dok leksički podaci iz drugog skupa domena odgovaraju raznim nizovima simbola - reči - iz specifičnog konteksta. Leksički podatak - slika- je jedinstveni element prvog skupa domena.

Drugi operator ovog nivoa je odredjivanje referenata zamenica tj. razrešavanje referencijalnog konflikta, PRONR (PRONoun Referencing). Operacija zamene surogata izvornom frazom (tj. zamenice imeničkom frazom koju zamenjuje) potrebna je da bi se iz samih reči dobio maksimum informacije sadržane u tekstu. Operator PRONR preslikava par (leksički podatak, skup leksičkih podataka) u jedinstveni niz leksičkih podataka. Prvi element domena ima sintaksnu vrstu "zamenica", drugi element domena

predstavlja specifični kontekst, a niz leksičkih podataka -slika- je reprezentacija izraza na koji se zamenica odnosi.

Mada potpuno lingvističko rešenje ni za jedan od navedena dva složena operatora ne postoji, delimični i korisni pristupi zasnovani na kontekstu izgledaju prihvatljivi. U glavi 3 biće izložene realizacije ova dva operatora primenom ekspertnih sistema [24] (operator RAMB za razrešavanje leksičke višeznačnosti i operator PRONR za odredjivanje referenata zamenica). Oba ekspertna sistema sadrže elemente modela verovatnoće kao sredstva za rešavanje problema nepouzdanog znanja u sistemu. Tehnika ekspertnih sistema dobro se uklapa u implementaciju oba operatora, kako zbog same prirode operatora, tako i zbog fleksibilnosti sistema zasnovanih na pravilima. Operatori tretiraju niz različitih slučajeva od kojih svaki može da koristi tačno onoliko koliko mu je potrebno od sistema zasnovanog na pravilima, a skup pravila je jednostavno ažurirati bez uticaja na proceduralni deo sistema. Mogućnost uključenja modela verovatnoće posebno dobro reflektuje potrebe specifičnih problema koji se rešavaju primenom ekspertnih sistema. Prikaz sistema za razrešavanje višeznačnosti i referenata zamenica nad leksičkim tipom podataka, na primeru engleskog jezika, objavljen je [49], tj. pripremljen je u [50].

Operatori RAMB i PRONR predstavljaju razumnu meru preprocesiranja tekstuelne baze podataka za aplikacije koje informaciju o tekstu dobijaju sa leksičkog nivoa. Takve aplikacije odgovaraju operatorima sledećeg nivoa hijerarhije operatora nad leksičkim tipom. Jedna od tih aplikacija je izdvajanje informacije iz teksta kao odgovora na upit, tj. realizacija operatora ovog nivoa, kome je domen skup upita nad tekstom a kodomen skup činjenica (fakata) u tekstu. Moj pristup u tezi polazi od zahteva da se ne ograničava domen tekstova, da se količina procesiranja

teksta do momenta postavljanja upita minimizira i da se, osim leksičke reprezentacije u bazi podataka, ne izgradjuje nikakva posebna reprezentacija znanja iz teksta. U glavi 4 biće izložen jednostavni eksperimentalni sistem za realizaciju ovog operatora primenom metoda relacionih baza. Sistem prikazuje tekstove kao virtuelnu relacionu bazu podataka koja odgovara zadatoj shemi. Entitet, tj. odnos predstavljen jednom relacijom u relacionoj shemi odgovara rečenici iz teksta, tj. njenom delu koji je relevantan za atribute tog entiteta (odnosa). Shema definiše prostor svih upita koji se mogu postaviti, i ona je fiktivna reprezentacija znanja iz teksta. Odgovor se dobija iz jednog ili više tekstova u vreme izvršavanja. Obrada upita intenzivno koristi karakteristike reči ugradjene u celobrojnu kodiranu reprezentaciju (vrednosti operatora prvog nivoa), kao i rezultate operatora drugog nivoa. Odgovori se unose u virtuelnu relacionu bazu podataka koja sada postaje hibridna (stvarno/virtuelna) jer sadrži neke podatke, ali za podatke koje još ne sadrži, predstavlja samo prozor na tekst. Mogućnost izdvajanja informacije iz teksta primenom virtuelne relacione sheme i odgovarajući eksperimentalni sistem izloženi su u [45, 49].

Ako su svi atributi jednorelacionog upita povezani jednom rečenicom, odgovor na taj upit dobija se iz jedne rečenice i naziva se direktnim. Moguć je slučaj da direktan odgovor nije sadržan u tekstu, ili da osim direktnog postoji i odgovor koji sa kvalifikacijom (podacima) upita nije sadržan u istoj rečenici. Kvalifikacija upita i ciljni atribut (tj. odgovor) mogu biti sadržani kao početak, odnosno kraj lanca informacija od kojih su svake dve susedne, kao kvalifikacija i ciljni atribut posebnog podupita, povezani po jednom rečenicom. Ovaj slučaj odgovara virtuelnoj relacionoj shemi (bazi podataka) sa null-vrednostima.

Korišćenjem teorije funkcionalnih i višeznačnih zavisnosti biće izložena teorija izvodjenja implicitne informacije primenom koncepta relacione baze podataka sa null-vrednostima, tj. biće dokazano kada se i kako upit može dekomponovati u niz podupita, tj. odgovor na polazni upit komponovati iz niza direktnih odgovora na podupite. Izvodjenje tražene informacije iz ovog lanca naziva se faktuelno izvodjenje i ovaj metod objavljen je u [48].

Još jedan skup leksičkih operatora ovog, trećeg nivoa, odgovara tradicionalnim operacijama nad tekstovima, kao što su automatsko indeksiranje, izdvajanje ključnih reči, apstraktiranje, pretraživanje. Predstavljanje teksta leksičkim tipom podataka u relacionoj bazi podataka omogućuje jednostavnu i efikasnu primenu raznih metoda za realizaciju ovih operatora. Eksperimenti sa ovim operacijama biće prikazani u glavi 5, a objavljeni su u [44, 46].

1.1. SRODNI REZULTATI

Apstraktni tip podataka - ATP [21, 38] primenjuje se u kontekstu relacionih baza podataka na dva načina: jedan je tretiranje relacije kao apstraktnog tipa podataka [56], drugi je upotreba ATP kao mehanizma za proširenje SUBP kompleksnim tipovima podataka u domenima atributa relacije [42, 67]. Razvijen u kontekstu programskih jezika, koncept ATP predstavlja strukturu podataka čija je implementacija skrivena pred spoljašnjim procedurama, zajedno sa skupom operatora nad tom strukturom.

U svojstvu domena atributa relacije, ATP predstavlja mehanizam za obogaćenje procesora podataka u SUBP proceduralnim znanjem pridruženim kolonama relacije. ATP obezbedjuje proces registracije za posebne tipove podataka koje korisnik definiše, kao i kolekciju funkcija koje omogućuju korisniku da definiše operatore nad uvedenim tipom podataka. Složeni tipovi podataka

koji se definišu u okviru ATP su, npr. vremenski, tekstuelni ili geometrijski podaci [65, 67].

Proširenje relacionih baza podataka direktnom podrškom leksičkog tipa podataka kao domena atributa, predloženo je u [74]. Ovde se pod leksičkim tipom podrazumeva (bilo kakva) reprezentacija teksta bazirana na rečima kao leksičkim i semantičkim jedinicama. Za razliku od ovog, leksičkog pristupa, u [27, 39, 66] izložen je pristup podršci tekstuelnih podataka slobodne segmentacije (bazirane na niskama simbola ili redovima). S obzirom da se oba pristupa bave direktnom podrškom teksta u SUBP, i da je leksička organizacija specifični, semantički obogaćeni oblik slobodne segmentacije, koncepti definisani za slobodno segmentirani tekst u [39, 66], -npr. operatori nad niskama simbola, uredjene relacije, promenljiva dužina podatka, mogu se koristiti i na leksički organizovanom tekstu. Osim ovih koncepata, nad leksički organizovanim tekstom izgradjen je niz leksičkih i leksički zasnovanih operatora (od raznih varijanti automatskog indeksiranja do morfološke, sintaksne i semantičke analize i mašinskog razumevanja teksta), u čijoj realizaciji se primenjuju udružene metode baza podataka, računarske lingvistike i veštačke inteligencije. Prirodni razlog tome je da je automatska obrada prirodnojezičkog teksta i njegovog sadržaja problematika u kojoj se preklapaju istraživanja u ovim oblastima. U preostalom delu ove glave biće pomenuti neki od tih operatora i metoda koji su srodni metodima koji se u ovom radu koriste pri izgradnji hijerarhije operatora nad leksičkim tipom podataka, kao i samim operatorima. Ovi operatori i metodi biće prikazani u delovima rada na koje se odnose.

U kontekstu sistema za procesiranje prirodnojezičkog teksta, od komunikacije sa delovima softvera kao što su baze podataka, do

pretraživanja i mašinskog "razumevanja" teksta, razmatra se specifična struktura rečnika kao i morfološki aspekti jezika (analize i sinteze). Rečnici različite organizacije i sadržaja, za različite jezike, prikazani su sa raznih aspekata (lingvističkog, reprezentacije znanja) u nizu radova, npr. [13, 26, 29, 41, 57, 62]. Simmons-ov koncept semantičkog svojstva reči kao jednomesnog predikata [62] neposredno je uticao na uključenje semantičkog atributa u relaciju rečnik u ovom radu. Mehanizmi za rešavanje problema morfološke analize i/ili sinteze predloženi su, npr. u [9, 22, 29, 34, 41, 70]. Posebno, morfologijom jezika bave se svi sistemi za prirodnojezičku komunikaciju sa bazama podataka [47]. Količina informacije koja se dodeljuje leksičkim ulazima u rečnik kao i količina procesiranja kojim se interpretiraju morfološka pravila zavisi od bogatstva oblika reči u jeziku kao bitne karakteristike jezika.

Problem jezičke višeznačnosti pojavljuje se u više oblika. U [14, 31, 36, 52], npr. razmatra se višeznačnost odnosa pojedinih delova rečenice (na koji se izraz odnosi modifikator ili relativna rečenica), i sugerišu se rešenja putem dijaloga, uvodjenjem tipova, odnosno raznih heuristika u pogledu položaja delova rečenice ili funkcionalnih reči. U [4] daje se pregled metoda za rešavanje problema leksičke višeznačnosti (višeznačnosti same reči), koji se u ovoj tezi rešava za ovu problematiku novom metodom - primenom ekspertnog sistema.

Pitanje nalaženja antecedensa (referenta) zamenice je u opštem slučaju nerešeno [60]. U literaturi su saopštena delimična rešenja za posebne slučajeve (ličnih, odnosnih zamenica), za fragmente prirodnog jezika, najčešće razmatranog u kontekstu komunikacije sa bazom podataka. Rešenja se zasnivaju, npr. na slaganju oblika zamenice i referenta, kao kod Simmons-a u [60],

gramatici koja generiše reprezentaciju domena teksta [18], predefinisanim rečeničnim obrascima i korespondiranju zamenice u pojedinoj rubrici i objekta iz iste rubruke iz prethodne rečenice [26], odnosno poslednjeg referisanog objekta iz klase kojom je rubrika okarakterisana [71].

Mehanizmi ekspertnih sistema, teorija koja ih podržava i razne primene u nauci, tehnici i poslovanju, opisani su u [24]. Organizacijom i konceptima sistema zasnovanih na pravilima, koji su gotovo sinoniman pojam pojmu ekspertnih sistema, posvećena je kolekcija radova u [73]. Zajednička karakteristika ovih sistema, koja ih razlikuje od konvencionalnih programa je manipulisanje znanjem umesto podacima [37, 72].

Tehnike vezane za obradu nepotpunog ili nepouzdanog znanja u ovim sistemima opisane su, npr. u [17, 58].

Problem pronalaženja odgovora na upit nad prirodnojezičkim tekstom kao tekstuelnom bazom podataka je dualan problemu prirodnojezičke komunikacije sa formatizovanom bazom podataka. Mada je u dosadašnjim istraživanjima mnogo veća pažnja posvećena ovom drugom problemu (npr. sistemi REQUEST [51], PARNAX [13], KODAS [22], TORUS [41], LADDER [26]), sve veći broj autora bavi se problemom izdvajanja fakata iz teksta (M. Lebowitz [36], W.Frey [18], E. Hajičova, Z Kirschner, P.Sgall, J.Panevova [22, 57], Grishman, Hirshman [20], R. Simmons [60]), a nedavno je objavljen i komercijalni proizvod ove vrste [6].

Kontekst u kom se najčešće rešava ovaj problem je širi i pripada oblasti mašinskog razumevanja teksta. Pritom se najčešće ograničava domen tekstova nad kojima se postavljaju pitanja (npr. tehnički patenti [36], medicinski izveštaji [20]), a glavni cilj je izgradnja celovite reprezentacije znanja sadržanog u tekstu (npr. semantičke mreže [22]). Brojna saopštena istraživanja imaju

za cilj upravo reprezentaciju znanja sadržanog u tekstu, pri čemu se daljom nadgradnjom može dobiti sistem (operator) za izdvajanje informacije iz teksta (R. Simmons u [60, 62], D. McDonald, C. Riesbeck, R. Duda, P. Thorndyke u [73]).

U slučaju kada je tekst predstavljen nekom od semantičkih reprezentacija sadržaja u celini [18, 22, 36], operator dobijanja odgovora na pitanje je uniforman bez obzira da li se odgovor u tekstu sadrži eksplicitno (u jednom iskazu) ili implicitno. Nasuprot tome, ako je reprezentacija teksta (kao, npr. leksička) zadržala strukturu izvorne reprezentacije teksta (niz rečenica), nalaženje odgovora koji je implicitno sadržan u tekstu predstavlja vid izvodjenja informacije iz niza posrednih informacija. U fiktivnoj reprezentaciji relevantnog znanja (iz teksta) relacionom shemom, implicitna informacija odgovara shemi sa null-vrednostima [12, 43]. Teorija funkcionalnih i višeznačnih zavisnosti (i posebno za ovakvu shemu) na kojoj se zasniva procedura izvodjenja, izložena je u [1, 2, 16, 43, 69]).

O metodama za automatsko izvodjenje tradicionalnih operacija nad tekstom, kao što su pretraživanje, indeksiranje, izdvajanje ključnih reči, apstraktiranje, postoje tomovi literature (npr. pregledni članak [64] o automatskom indeksiranju, [8, 63] o apstraktiranju, [5, 22, 54, 57], o pretraživanju). Moje istraživanje u vezi sa ovim operatorima u ovom radu nije vezano za same metode već za mogućnost primene tih metoda u izgradnji odgovarajućih operatora nad leksički organizovanim tekstom u bazi podataka.

2. UPRAVLJANJE TEKSTOM KAO PODACIMA U RELACIONOJ BAZI PODATAKA

Pošto se u delu 2.1 izlože ukratko elementi relacione baze podataka, u preostalom delu ove glave i delimično u sledećoj biće razmotreni specifični problemi koji se javljaju pri definisanju i implementaciji leksičkog tipa podataka kao sredstva za predstavljanje teksta u relacionoj bazi podataka, kao i samo predstavljanje teksta leksičkim tipom. Ti problemi su:

* efikasno čuvanje reči u relacionoj bazi podataka (definisanje leksičkog podatka),

* implementacija osnovnog skupa leksičkih operatora (operatora prvog nivoa),

* razrešavanje višeznačnosti reči predstavljenih istom niskom simbola -leksički operator drugog nivoa (zbog svoje karakterističnе metodologije rešenje je izloženo u sledećoj glavi).

2.1. O RELACIONOM MODELU BAZA PODATAKA

Relacioni model za formatirane baze podataka ([2, 10, 16]) uveden je krajem šezdesetih godina, pre svega kao sredstvo kojim se korisnik oslobadja detalja fizičke reprezentacije podataka u memoriji. Mada su u medjuvremenu i mrežni tj. hijerarhijski model dostigli nivo apstrakcije udoban za korišćenje (upitni jezici visokog nivoa), relacioni model se, zbog svoje konceptualne jednostavnosti (svi objekti predstavljaju se tabelama), mogućnosti formalnog i strogog zasnivanja i raznovrsnih domena i subjekata primene, izdvaja sve jasnije kao standard u tehnologiji baza podataka (danas gotovo svi proizvodi baza podataka na tržištu bar tvrde da su relacioni [16]).

Modeli baza podataka koriste se za predstavljanje pitanja

(upita) o predstavljenim objektima i vezama medju njima (jedinstveni naziv za objekte i veze biće "entitet" [2]). Relacioni model se sastoji od tri glavna dela: strukturnog, manipulativnog i integritetnog dela [16].

(i) STRUKTURNI DEO RELACIONOG MODELA

Strukturni deo suštinski se sastoji isključivo od n-arnih relacija, zajedno sa odgovarajućim domenima (ova "isključivost" bitno razlikuje relacioni model od mrežnog i hijerarhijskog koji operišu bar sa dve vrste objekata i time unose složenost kako u koncept tako i u neposrednu manipulaciju podacima). Relacijom se predstavljaju svi entiteti - kako objekti tako i veze medju njima. Sledeći pojmovi karakterišu strukturni deo relacionog modela [1, 2]:

Relacija R je podskup Dekartovog proizvoda skupova D_1, D_2, ..., D_n, tj. $R \subseteq D_1 x D_2 x ... x D_n$. Struktura relacije opisuje se sa $R(A_1, A_2, ..., A_n)$, gde je A_i atribut relacije R sa domenom D_i, tj. preslikavanje A_i: R -> D_i, za i=1,2,...,n. Tip entiteta E predstavljen relacijom R karakteriše se istim atributima i domenima kao ta relacija ($E(A_1, A_2, ..., A_n)$, A_i: E ->D_i; za $e \in E$, $A_i(e) \in D_i$ naziva se vrednost atributa A_i entiteta e). Funkcijom $(A_1, A_2, ..., A_n)$: E -> $D_1 x D_2 x ... x D_n$, ostvaruje se predstavljanje tipa entiteta n-torkom vrednosti njegovih atributa. Skup entiteta tipa E (relacija R) predstavlja se tabelom sa redovima koji odgovaraju elementima tog skupa i kolonoma koje odgovaraju atributima.

Relaciona baza podataka je skup relacija koje se menjaju u vremenu (ova vremenska zavisnost razlikuje relacije relacionog modela baza podataka od matematičkog pojma relacije, i odražava potrebu za ažurnim stanjem baze - neke se n-torke brišu, neke

unose, nekima menjaju vrednosti pojedinih atributa). Opis strukture relacija iz tog skupa naziva se relaciona shema.

(ii) MANIPULATIVNI DEO RELACIONOG MODELA

Manipulativni deo relacionog modela obezbedjuje skup algebarskih operatora (ili njihovih ekvivalenata u relacionom računu [11]), kojima se može opisati proizvoljan izraz sa relacijom kao vrednošću, i zatim primeniti na razne oblike manipulisanja podacima, od pretraživanja podataka do istraživanja u oblasti projektovanja baza, optimizacije upita, definisanja pogleda, zaštite i integriteta [16].

Komunikacija sa bazom podataka može se ostvariti na raznim nivoima i različitim sredstvima (konvencionalnim upitima na upitnim jezicima, npr. QUEL-u [25, 28], izrazima relacione algebre ili formulama relacionog računa [10, 11], grafičkim jezikom tipa QBE [77], formulama algebre logike [19], predikatima PROLOG-a [9], prirodno-jezičkim upitima [47]).

Mada zasnovani na relacionoj algebri ili relacionom računu, relacioni upitni jezici imaju ekspresivnije mogućnosti za definisanje radnji nad tabelama relacione baze. Primer relacionog upitnog jezika zasnovanog na relacionom računu, koji će ilustrovati osobine upitnih jezika i na kom su radjeni primeri u ovoj tezi, je QUEL (QUEry Language - "upitni jezik" - jezik relacionog sistema za upravljanje bazama podataka INGRES [25, 28]). Ovaj jezik (što je najčešće slučaj) ima dva podjezika - jezik za definisanje i jezik za manipulisanje podacima. Samo će ovaj drugi biti ukratko prikazan.

Glavne komande podjezika za manipulisanje podacima su:

RETRIEVE (pretražiti),

REPLACE (zameniti),

DELETE (izbrisati),

APPEND (dodati).

QUEL zahteva da, gdegod to ima smisla, upiti budu formulisani u terminima "domenskih promenljivih" (specifična relacija je domen specifične domenske promenljive, tj. domenska promenljiva uzima kao vrednosti elemente - n-torke - specifične relacije).

Iskaz za definisanje domenske promenljive je

RANGE OF dom_promenljiva IS relacija.

Jednostavni primeri komandi za pretraživanje, zamenu, brisanje i dodavanje nad relacijom S sa atributima ATR_1, ATR_2, ATR_3:

RANGE OF e IS S

RETRIEVE (e.ATR_1, e.ATR_3) WHERE e.ATR_2=konst

pretražiti vrednosti atributa ATR_1 i ATR_3 n-torki relacije S, koje a atributu ATR_2 imaju vrednost "konst"; RETRIEVE i lista traženih rednosti (u zagradi) su obavezni delovi ove naredbe; ukoliko ostoji WHERE u naredbi, obavezna je i lista uslova (iza WHERE)).

RANGE OF e IS S

REPLACE e (ATR_1 = e.ATR_1+$konst_1$) WHERE e.ATR_3 <= $konst_2$

u n-torci relacije S, čija je vrednost atributa ATR_3 manja ili ednaka $konst_2$, zameniti vrednost atributa ATR_1 tako da bude za $onst_1$ veća od stare vrednosti).

APPEND TO S (ATR_1=$konst_1$, ATR_2=$konst_2$, ATR_3=$konst_3$)

relaciji S dodaje se jedna n-torka sa konstantama $konst_1$, $konst_2$, $onst_3$ kao vrednostima atributa ATR_1, ATR_2, ATR_3, redom).

RANGE OF e IS S

DELETE e WHERE e.ATR_1=$konst_1$ AND e.ATR_2><$konst_2$

izbrisati sve n-torke relacije S čija je vrednost atributa ATR_1 ednaka konstanti $konst_1$ i vrednost atributa ATR_2 različita od onstante $konst_2$).

iii) INTEGRITETNI DEO RELACIONOG MODELA

Pre nego što se opišu, bez navodjenja svojstava, pravila integriteta relacionog modela, izložiću pojmove logičkih zavisnosti [1] sa aspekta primene na pravila integriteta (ne sa aspekta logičkog projektovanja).

Medju podskupovima X, Y atributa relacije R postoji funkcionalna zavisnost (u oznaci X -> Y) ako i samo ako je projekcija relacije R[XY] funkcija R[X] -> R[Y] u svakom trenutku, tj. kadgod su xy i xy' elementi relacije R[XY], tada je y=y'.

Podskup X skupa atributa relacije R je ključ relacije R ako i samo ako važi:

(i) za svaki atribut A_i relacije R postoji funkcionalna zavisnost X -> A_i;

(ii) ni jedan pravi podskup od X nema tu osobinu.

Svaka relacija ima ključ (bar svi atributi relacije), ali on ne mora biti jednoznačan. Jedan od ključeva bira se za jedinstveni identifikator entiteta i naziva se primarni ključ.

Prvo opšte pravilo integriteta ("integritet entiteta") odnosi se upravo na primarni ključ i kaže da ni jedan od atributa primarnog ključa ne sme uzeti nedefinisanu (ili nepoznatu) vrednost.

Jedno svojstvo funkcionalnih zavisnosti koje će se koristiti u glavi 4. je da egzistencija funkcionalne zavisnosti X -> Y medju podskupovima atributa relacije R(X, Y, Z) garantuje da se zamenom relacije njenim projekcijama R[X,Y], R[X,Z] ne gubi informacija, tj. da se relacija R može restaurisati primenom operacije spajanja nad projekcijama R[X,Y], R[X,Z] (R(X,Y,Z) = R[X,Y] * R[X,Z]).

Ako relacija R predstavlja vezu izmedju dva objekta predstavljena relacijama P i S, onda relacija R uključuje kao

svoje atribute, pored ostalih, i atribute primarnih ključeva relacija P i S. Primarni ključevi relacija P i S u relaciji R nazivaju se strani ključevi ("foreign keys").

Drugo opšte pravilo integriteta ("integritet referisanja") odnosi se na strani ključ i kaže da, ako relacija R uključuje strani ključ K koji je primarni ključ relacije P, onda svaka vrednost od K u R mora biti ili (a) jednaka vrednosti atributa K neke n-torke iz P ili (b) u potpunosti nepoznata. Smisao ovog pravila je da; ako se neka n-torka r relacije R referiše (odnosi) na neku n-torku p relacije P (jer R odražava vezu objekata predstavljenih relacijama P, S), onda n-torka p mora da postoji. Stoga odgovarajuća vrednost stranog ključa u R mora biti prisutna kao vrednost primarnog ključa u P, ili n-torka r ne zna na koju se n-torku p odnosi pa je vrednost stranog ključa na r nepoznata.

Drugi (opštiji) oblik zavisnosti medju atributima jedne relacije (ili relacione sheme) je višeznačna zavisnost (biće primenjena u glavi 4).

Neka je R(X, Y, Z) relacija sa m+n+r atributa, gde su X, Y, Z medjusobno disjunktni skupovi atributa sa m, n, r atributa, redom, i neka je x m-torka konkretnih vrednosti atributa iz X neke m+n+r-torke relacije R (slično za y,z). Neka je $Y_{xz}=\{y:(x,y,z)\in R\}$. U relaciji R važi višeznačna zavisnost (VZ) X->->Y ako i samo ako Y_{xz} zavisi samo od x, tj. ako i samo ako je $Y_{xz} = Y_{xz'}$ za svako x, z, z' za koje su Y_{xz} i $Y_{xz'}$ oba neprazna.

2.2. LEKSICKI TIP PODATAKA

Prirodni način za čuvanje teksta u relacionoj bazi podataka je reprezentacija teksta relacijom:

naziv_teksta(redbr, reč),

gde "redbr" označava redosled pojavljivanja a "reč" može biti reč,

interpunkcijski znak ili specijalni simbol kao, npr. "novi paragraf".

Prvo pitanje koje se postavlja je kakvim podatkom predstaviti reč. Dužina reći predstavljenih niskama simbola bitno varira od reći do reći. Stoga je reprezentacija reći niskama simbola u polju fiksne dužine vrlo neefikasna. Jedno rešenje ovog problema je celobrojno kodiranje reći u reprezentaciju fiksne dužine, koje pruža mogućnost značajne kompresije. Na primer, ceo broj dužine 4 bajta dovoljan je da predstavi rečnik od $2^{32} \sim 4 * 10^9$ reči.

Postoje i drugi podjednako ubedljivi razlozi za kodiranje. U reprezentaciji reći niskom simbola sadržano je vrlo malo leksičke informacije. Činjenica da reč "ljudi" ima "čovek" kao svoj koren očigledno ne može biti izvedena iz same niske "lj-u-d-i". Ako je jedan od ciljeva izgradnje leksičkog tipa i implementacija osnovnih leksičkih operatora, reći moraju biti predstavljene na način koji obezbedjuje eksplicitno pojavljivanje vrednosti tih operatora u izabranoj reprezentaciji. Kodirana reprezentacija reći trebalo bi da bude, u suštini, kompozicija vrednosti skupa svih dopuštenih osnovnih operatora nad tom reći.

Još jedan razlog u prilog kodiranju je odstranjenje višeznačnosti. Ista niska simbola često ima nekoliko značenja, tj. predstavlja nekoliko različitih reči, ili, preciznije, različitih "leksičkih jedinica".

Iz ovih razloga, sa gledišta aplikacija koje koriste značenje teksta, kodirana reprezentacija reći nameće se kao pravi tip podataka za čuvanje i manipulisanje tekstom u bazi podataka.

Sledeće pitanje je na koji način realizovati ovaj leksički tip podataka, tj. kako izvršiti kodiranje reči. Mada bi shema automatskog kodiranja, kojoj su ulaz samo niske simbola,

zadovoljila potrebe kompresije, druga dva cilja nije moguće ostvariti na ovaj način, s obzirom da je potrebna dodatna informacija. Kao izvor leksičkih informacija pri predstavljanju teksta leksičkim tipom podataka koristi se rečnik, a kao sredstvo za razrešavanje višeznačnosti pojedinih reči, ekspertni sistem.

Količina leksičke informacije koju treba pribaviti u cilju realizacije leksičkih podataka zavisi od leksičkih operatora koji treba da budu podržani. S druge strane, proceduralna definicija leksičkih operatora zavisi od strukture i organizacije pribavljene leksičke informacije. Stoga je redosled koraka pri izgradnji leksičkog tipa sledeći: definisanje leksičkog tipa podataka (leksičkog podatka i podržanih operatora), definisanje sredstava za snabdevanje teksta leksičkom informacijom u cilju njegovog predstavljanja leksičkim podacima, i implementacija leksičkih operatora.

Definicija 1. Leksička jedinica je slika reči pri kodiranju; Skup leksičkih podataka je skup leksičkih jedinica zajedno sa nekim predefinisanim vrednostima; Leksički tip podataka je par (X, L), gde je X skup leksičkih podataka a L skup svih podržanih (primitivnih, osnovnih) operatora.

Skup operatora L sastoji se od četiri tipa operatora: leksičkih operatora kao, npr. nalaženje korena, prefiksa, završetka ili semantičkog svojstva leksičke jedinice, izgradnja specifičnih leksičkih oblika kao što su množina za imenice ili prošlo vreme za glagol, konkatenacija ili brisanje jedne leksičke jedinice sa/iz druge; sintaksičkih operatora kao, npr. nalaženje vrste reči i oblika date leksičke jedinice (npr. vreme, lice, broj glagola, stepen, padež, broj, rod datog prideva, vrsta, red, broj, padež date zamenice); metričkih operatora kao što je dužina date leksičke jedinice u simbolima; istinosnih operatora kao što je jednakost ili poredak leksičkih jedinica zasnovan na težinama korena, prefiksa, završetaka, formi i semantičkih svojstava reči.

Primeri ovih operatora su sledeći:

koren(ljudi) = čovek;

završetak(pevanje) = nje;

broj(ljudi) = množina;

lexform(oblik$_1$, čovek) = ljudi, za oblik$_1$ = (množ, 1.padež);

lexform(oblik$_2$, ljudi) = null, za oblik$_2$ = (sad.vreme, 1.lice, jedn.);

konkat(pevati, nje) = pevanje;

konkat(naj, ši) = null.

Element iz X je oblika (id, deskr), gde je id ceo broj koji jednoznačno identifikuje element (leksičku jedinicu), a deskr je celobrojni deskriptor koji obuhvata dodatnu sintaksno-semantičku informaciju. Kodirana reprezentacija (id, deskr) je kompozicija vrednosti operatora definisanih nad datom rečju, tj.

id = (vr.koda(koren)*a +vr.koda(prefiks))*b +vr.koda(završetak)

deskr = vr.koda(vrsta_oblik)*c + vr.koda(svojstvo)

(vr.koda označava vrednost koda).

Vrednosti kodova za prefikse, završetke i semantička svojstva čitaju se iz tabela, a koren se kodira na osnovu interpolacije gustine reči u rečniku; početne vrednosti kodova za korene koji počinju specifičnim slovom odredjeni su na bazi ukupnog broja kodnih vrednosti koje su na raspolaganju i proporcionalno broju strana koje zauzimaju reči sa tim početnim slovom u običnom rečniku. "Vrsta_oblik" označava vrstu reči i njen specifični oblik (npr. "imenica, ženski rod, 1. padež", "glagol, infinitiv", itd.). Vrednost koda vrste_oblika reči je broj pridružen vrsti_obliku te reči u tabeli koja sadrži ulaz za svaki mogući oblik svake vrste reči. (Vrednost koda za semantičko svojstvo biće objašnjena kasnije). Brojevi a, b, c su zavisni od

jezika, tj. od broja elemenata u svakom od skupova prefiksa, završetaka, semantičkih svojstava.

U sledećim tačkama precizno se definiše leksički tip podataka.

(i) SKUPOVI PODATAKA

LEKS (skup leksičkih podataka) je unija sledećih skupova parova celih brojeva (id, deskr):

- kodirane potpune leksičke jedinice - kodirane leksičke jedinice iz rečnika, koje su slike reči,
- skupovi parova (id, deskr), koji sadrže vrednosti koda za sve elemente iz relacija podataka PREFIKS, ZAVRŠETAK, SEM_SVOJSTVO, redom, u odgovarajućim delovima u id, deskr, a nule u svim ostalim delovima kodirane reprezentacije i
- null;

SINT (skup sintaksičkih podataka) je unija sledećih skupova:
- VR (skup vrsta reči),
-IFORM, GFORM, PFORM, ZFORM, BFORM (skupovi svih oblika koji odgovaraju imenicama, glagolima, pridev-prilozima, zamenicama i brojevima, redom), tj.

VR = {pravilna_imenica, pravilni_glagol, pravilni_pridev, pravilni_prilog, nepravilna_imenica, nepravilni_glagol, pomoćni_glagol, nepravilni_pridev, nepravilni_prilog, zamenica, veznik, prefiks, predlog};

IFORM = {1.padež, 2.padež, 3.padež, 4.padež, 6.padež, null[1]} x {jedn, množ, null} x {m, ž, s, null};

("x" označava Dekartov proizvod skupova)

GFORM = {sad, proš, bud, null} x {1.l, 2.l, 3.l, null} x {jedn, množ, null} x {m, ž, s, null} U {infinitiv, pril.vr.sad, pril.vr.pro} ;

PFORM = {pozitiv, komp, superl,null} x {1.p, 2.p, 3.p, 4.p, 6.p, null} x {jedn, množ, null} x {m, ž, s, null};

ZFORM = {lična, prisvojna, pokazna, odnosna, null} x {1.p, 2.p, 3.p, 4.p, 6.p, null} x {jedn, množ, null} x {1.l, 2.l, 3.l, null} x {m, ž, s, null} x {jedn, množ, null}

BFORM = {ord, redni, null} x {1.p, 2.p, 3.p, 4.p, 6.p, null} x {jedn, množ, null} x {m, ž, s, null}

Q - skup brojeva;

---

[1] "null" je specijalni simbol koji, kao element n-torke čiji je par jedan element različit od null, ima značenje "skupljanja" n-torke po polju koje sadrži null, tj. projektovanje n-torke na ostala polja; ako n-torka sadrži samo null-elemente, onda u kodomenu preslikavanja ima značenje neprimenljivog svojstva. Prvi slučaj omogućuje primenu ovih istih skupova na razne jezike.

TF - skup istinosnih vrednosti {T, F}

(ii) KONSTANTE, PROMENLJIVE

Konstante:

$l_i \in$ LEKS;

$s_i \in$ SINT;

$q_i \in Q$;

T, F $\in$ TF.

Promenljive:

$L_i \in$ LEKS;

$S_i \in$ SINT;

$Q_i \in Q$;

$TR_i \in$ TF.

(iii) OPERATORI

Leksički operatori:

LEKS* -> LEKS ili LEKS* x SINT -> LEKS*;

Sintaksički operatori: LEKS* -> SINT;

Metrički operatori: LEKS * -> Q;

Istinosni operatori: LEKS* -> TF.

Specifični operatori nad leksičkim tipom podataka:

unarni:

leksički:

koren(L_i) ($\in$ LEKS);

prefiks(L_i) ($\in$ LEKS);

završetak(L_i) ($\in$ LEKS);

svojstvo(L_i) ($\in$ LEKS);

sintaksički:

vrsta(L_i) ($\in$ VR);

padež(L_i) ($\in$ IFORM U PFORM U ZFORM U BFORM);

rod(L_1) ($\in$ IFORM $\cup$ GFORM $\cup$ PFORM $\cup$ ZFORM $\cup$ BFORM);

vreme(L_1) ($\in$ GFORM);

lice(L_1) ($\in$ GFORM $\cup$ ZFORM);

broj(L_1) ($\in$ IFORM $\cup$ GFORM $\cup$ PFORM $\cup$ ZFORM $\cup$ BFORM);

stepen(L_1) ($\in$ PFORM);

vrsta(L_1) ($\in$ ZFORM $\cup$ BFORM);

metrički:

dužina(L_1) ($\in$ Q);

binarni:

leksički:

lexform(L_1, S_1) ($\in$LEKS);

konkat(L_1, L_2) ($\in$LEKS*);

isključenje(L_1, L_2) ($\in$LEKS*);

istinosni:

jednakost(L_1, L_2) ($\in$TF);

manje_jed(L_1, L_2) ($\in$TF);

veće_jed(L_1, L_2) ($\in$TF).

iv) LEKSIČKI I LOGIČKI IZRAZI

Leksički izraz je niz konstanti i promenljivih iz skupa LEKS skupova koji ga podržavaju, napisan naizmenično sa operatorima oji rezultuju vrednošću iz skupa LEKS, tj.

(i) leksička konstanta tj. leksička promenljiva (l_1, L_1) je eksički izraz;

(ii) ako je S sintaksna konstanta ili promenljiva (s_1 ili $_1$), i LI_1, LI_2 - dva leksička izraza, tada je $l\_op_1(LI_1)$, $\_op_2(LI_1, LI_2)$, $l\_op_2(LI_1, S)$ - leksički izraz, gde su l_op_1, $\_op_2$, unarni tj. binarni (respektivno) leksički operatori;

(iii) leksički izraz se dobija samo primenom pravila (i), ii).

Leksički predikat je oblika ist_op(LI_1, LI_2), gde su LI_1, LI_2 bilo koja dva leksička izraza, a ist_op je bilo koji binarni istinosni operator prethodno definisan.

Logički izraz (i prema tome kvalifikacija u strukturi upita) je proširen tako da prihvata leksičke predikate kao argumente logičkih operacija (not, and, or).

(v) PROCEDURE ZA IZRACUNAVANJE (EVALUACIJU) OPERATORA

Operatori nad leksičkim tipom podataka definisani su procedurama koje kao argumente (ulazne i izlazne) imaju kodirane leksičke jedinice (parove celih brojeva) i elemente sintaksičkog skupa. Procedure su napisane u C-jeziku [30], ili EQUEL-jeziku (upitni jezik QUEL umetnut u C-jezik [28]). Na primer, operator za nalaženje korena reči implementira se sledećom procedurom:

koren

```
koren(L)
int L[2];
{
   L[0] = (L[0] / 10**4)* 10**4;
   L[1] = 0;
}
```

.3. PREDSTAVLJANJE TEKSTA LEKSICKIM TIPOM PODATAKA

Osnovni cilj u predstavljanju teksta u relacionoj bazi odataka je učitati tekst u prirodnom obliku i automatski ga onvertovati u relaciju:

tekst(redbr, leks)

de "redbr" predstavlja redosled pojavljivanja a "leks" leksički odatak (leksička jedinica) je ili slika reči pri kodiranju ili pecijalni simbol. Proces kodiranja (a) redukuje reč na eprezentaciju fiksne dužine, (b) čini eksplicitnim leksička vojstva potrebna da bi se omogućila podrška željenih operatora i :) razrešava višeznačnost koja može biti prisutna u

reprezentaciji reči niskom simbola. Automatsko predstavljanje teksta leksičkim tipom podataka postiže se korišćenjem: tekstuelnog skenera, rečnika, skupa morfoloških pravila i ekspertnog sistema za implementaciju operatora razrešavanja višeznačnosti (operatora RAMB).

2.3.1. TEKSTUELNI SKENER

Tekstuelni skener uzima, kao ulaz, izvorni tekst, analizira njegovu strukturu i proizvodi, kao izlaz, datoteku koja sadrži "niske" (nizove simbola koji imaju značenje) i obe vrste informacija o svakoj nisci: informaciju o kontekstu - o paragrafu, rečenici, podrečenici u kojoj je reč - i grafičku informaciju - o redu i poziciji u redu u kom je reč.

Tekst skener se može pisati na raznim nivoima opštosti, te i složenosti. Takodje postoji izvesna specifičnost skenera za specifični jezik (npr. jedna razlika izmedju skenera za engleski i srpskohrvatski jezik bila bi u interpretaciji broja za kojim sledi tačka - u englskom jeziku to je gotovo uvek broj na kraju rečenice, dok je u srpskohrvatskom jeziku to redni broj a kraj rečenice ispituje se onda drugim sredstvima. Primer ulaznog teksta i rezultata rada jednostavnog skenera je sledeći:

Ulaz: "U poslednjoj deceniji, značaj relacionog modela podataka široko je priznat."

Izlaz (u relaciji "primer" sa atributima "rečen#" - broj rečenice, "preč#" - broj podrečenice, "reč#" - broj reči u podrečenici, "duž" - dužina reči, "reč" - sama reč):

rečen#	preč#	reč#	duž	reč
1	1	1	1	U
1	1	2	10	poslednjoj
1	1	3	8	deceniji
1	1	4	0	,
1	2	1	6	značaj
1	2	2	10	relacionog
1	2	3	6	modela
1	2	4	8	podataka
1	2	5	6	široko

1	2	6	2	je
1	2	7	7	priznat
1	2	8	0	.

Za potrebe ovog rada napisan je i skener za srpskohrvatski jezik na programskom jeziku C, čiji je rezultat rada nad datim ulaznim tekstom prikazan u <u>dodatku</u> 1.

Još opštiji skener mogao bi da uzima, kao ulaz, osim izvornog teksta, i sledeće elemente:

- moguće tipove niski definisane regularnim izrazima, zajedno sa dopunskim uslovima koje treba testirati (npr. niska je reč iz naslova ako je reč i ako je indikator naslova = 1),
- aktivnosti koje implicira svaka pojedinačna nadjena niska (postavljanje uslova),
- redosled ispitivanja pojedinih tipova niski.

Pravila odredjivanja brojeva paragrafa, rečenice, odrečenice, reči, mogu se uključiti u aktivnosti implicirane specifičnim tipovima niski, tako da bi sačinjavala deo ulaza a ne deo programa - skenera. Tada bi skener bio proširenje konačnog automata koji prepoznaje niške i proizvodi odgovarajuće aktivnosti. Skener korišćen u ovom radu bio bi specijalni slučaj ovog opšteg skenera, koji, osim teksta, ima kao ulaz i sledeći opis tipova niski, uslova, aktivnosti, paragrafa, rečenica i odrečenica (p# označava broj paragrafa, r#- broj reda, rr - broj reči unutar reda):

```
Indikatori uslova: indn /* indikator naslova */
                   indf /* indikator fusnote */
                   indb /* indikator bibliografije */
                   /* svi se postavljaju na nulu */
```

tip_niske	uslov	niska	aktivnost
reč	-	slovo|(slovo cifra)*	-
broj	-	znak neozn_broj	-
interp_znak	-	.|?|!|.)|?)|!)	rečen#=rečen#+1, preč#=1,reč#=0.
		,|;|:|,)|;)|:)	preč#=preč#+1, reč#=0;
uslov	indn=1	reč|broj	-
fusnota	indf=1	reč|broj	-

bibliogr.	indb=1	reč|broj	
komanda	rr=1	.pp|.lp	indn=0, p#=p#+1,
			prec#=prec#=1,
			rec#=0;
	rr=1	(.sh|.uh)cifra	indn=1;
	rr=1	.(f	indf=1;
	rr=1	.)f	indf=0;
	rr=1	.++B	indb=1;
eozn_broj	–	cifra +| cifra * .cifra +	–
nak	–	+|-|e	–
lovo	–	A|B|...|Z|a|b|...|z	–
ifra	–	1|2|...|9	–

.3.2. REČNIK

Za potrebe implementiranja definisanih osnovnih leksičkih operatora pa dakle i izabranog načina predstavljanja teksta, projektovan je rečnik tipa relacije sledeće strukture:

ečnik(reč, vrsta, oblik, koren, prefiks, završetak, svojstvo, id, eskr).

"Reč" označava nisku simbola koja predstavlja reč. Rečnik sadrži, o "reči", osnovne oblike pravilnih reči (npr. 1. padež jednine avilnih imenica), i sve oblike nepravilnih reči od kojih su ugi oblici te reči izvodivi (npr. za imenicu "čovek", 1. padež dnine i 1. padež množine, ili, u engleskom jeziku sva tri oblika pravilnih glagola). "Vrsta" označava sintaksnu klasifikaciju či (npr. "imenica"), "oblik" je specifični oblik date vrste či. "Koren", "prefiks" i "završetak" se odredjuju intuitivno, i dino pravilo pri odredjivanju korena je da je on uvek, sam za

sebe, reć.

Semantičko svojstvo je oznaka koja odražava semantiku reči ili specifične upotrebe reči (npr. "akcija" - ACT za glagol "raditi", "mesto" - LOC za prilog "tamo", "vreme" - TIM za prilog "onda", "kvalitet" - QU za imenicu "vrednost", oba TIM i LOC za predlog "do". Skup semantičkih svojstava koja se ovde upotrebljavaju je inspirisan Simmons-ovim radom o semantičkim mrežama [62], pritom proširen hijerarhijskom strukturom. Na primer, semantičko svojstvo "vreme" ima subordinirana semantička svojstva "sadašnjost" - PR, "prošlost" - PST i "budućnost" - FUT. Skup koji se u ovom radu koristi sadrži oko 50 semantičkih svojstava.

Struktura koda - id, deskr, opisana je u delu 2.2. kao struktura leksičke jedinice koja se za datu reč dobija iz rečnika.

Skupovi prefiksa, završetaka i semantičkih svojstava smešteni su u odgovarajućim relacijama sa atributima (reč, r.koda). "Reč" ovde označava prefiks, završetak, tj. semantičko svojstvo. Vrednosti kodova za završetke osnovnih oblika rastu sa nekim korakom k zavisnim od jezika, tako da razne vrednosti unutar koraka mogu da kodiraju razne oblike izvedene iz osnovnog.

U procesu kodiranja reči u rečniku učestvuje i relacija lesc_end koja sadrži vrednosti kodova svih oblika svih vrsta reči učestvuju u računanju deskriptora deskr), kao i vrednosti kodova epravilnih završetaka pojedinih oblika nepravilnih vrsta reči učestvuju u identifikatoru id). Vrednosti kodova završetaka snovnih oblika počinju od neke konstante d, zavisne od jezika, ri čemu prvih d-1 vrednosti odgovaraju završecima nepravilnih blika (iz desc_end). Relacija desc_end ima atribute vrsta, blik, deskr, završetak.

Pored osnovnog rečnika, posebno se gradi rečnik vlastitih

imena i fraza oblika

vlast_fraza(fraza, br_reči, vrsta, oblik, svojstvo, id, deskr)

sa jasnim značenjem pojedinih polja. Identifikator id za vlastite fraze računa se na sledeći način:

id= -(vr.koda("koren")*1000 + br_reči*100 +vr.koda(završetak),

gde je "koren" u slučaju fraze od jedne reči isto što i u rečniku, a u slučaju fraze od više reči, reč izgradjena od početnih slova reči.

Kodiranje rečnika odvija se na sledeći način. Datoteka sa sadržajem rečnika unosi se u sistem ručno (bez id, deskr - polja), sa "?" u "završetak" - polju u slučaju nepravilnog završetka. Glavni program čita rečničku datoteku i za svaki red poziva procedure za računanje id i deskr. Prva procedura, posle izdvajanja pojedinih polja, poziva proceduru za računanje težine korena, i čitanjem vrednosti kodova za odgovarajući prefiks i završetak, gradi identifikator id. Druga procedura čita vrednost koda za vrstu i oblik date reči i vrednost koda za odgovarajuće svojstvo, iz relacija desc_end i feat, redom, i gradi deskriptor deskr. Ona takodje modifikuje identifikator id u slučaju nepravilnog završetka ("?" - vrednost u polju "završetak"). Glazni red, dopunjen vrednošću koda (id, deskr), upisuje se u izlaznu datoteku, koja se na kraju kopira u relaciju "rečnik". Slična je procedura za kodiranje vlastitih fraza.

Izgradjen je eksperimentalni rečnik engleskog jezika sa oko 1000 ulaza (u relaciji "dictrel") i rečnik vlastitih fraza sa oko 100 ulaza (u relaciji "prop_phr") (za srpskohrvatski jezik, rečnik sa oko 700 ulaza u relaciji "dictrels" i rečnik vlastitih fraza sa oko 40 ulaza). Ovi rečnici uključuju reči iz nekoliko eksperimentalnih tekstova tipa biografija. Delovi relacija vezanih za rečnik nalaze se u dodatku 2. Program je napisan na EQUEL

jeziku i sastoji se od oko 500 linija.

Predstavljanjem rečnika relacionim modelom baza podataka, ovaj model se pojavljuje kao model baze znanja (sadržanog u rečniku), čime se za manipulaciju znanjem koriste svi mehanizmi baza podataka. Pored ovog specifičnog načina predstavljanja rečnika u relacionoj bazi podataka, glavna karakteristika pristupa rečniku u ovom radu je kodiranje.

Osim ovog načina predstavljanja znanja iz rečnika, u literaturi se za predstavljanje rečnika koriste i drugi oblici reprezentacije znanja. Pri predstavljanju rečnika semantičkom mrežom korišćenjem proceduralne logike, Simmons [62] predstavlja rečnik kao sastavni deo gramatike, a iskazi koji odgovaraju rečničkim ulazima su tipa (ART THE) ("the" je član), (ADJ GIANT) ("giant" je pridev), (V ROSE(RISE, TNS PAST)) ("rose"je glagol u prošlom vremenu sa infinitivom "rise"), (FEAT BEHIND LOC) (semantičko svojstvo reči "behind" je mesto). U sistemu LADDER za prirodno-jezičku komunikaciju sa distribuiranom bazom podataka o brodovima [26], autora sa Stenfordskog istraživačkog instituta, rečnik je LISP-funkcijama grupisan po skupovima koji odgovaraju pojedinim metasimbolima sistema (npr. metasimbolu "atribut" odgovaraju reči: klasa, komandant, nacija, dužina, tip, itd, metasimbolu "ime_broda" - reči "nautilus", "kennedy", itd). Još jedan sistem za razumevanje prirodnog jezika u kontekstu prirodno-jezičke komunikacije sa bazom podataka je TORUS [41] autora sa Univerziteta u Torontu. U slučaju ovog sistema, deo rečnika se čuva u semantičkoj mreži, pri čemu su rečnički ulazi podredjeni apštijim konceptima, a deo u relacionoj bazi - kao vrednosti atributa pojedinih entiteta. Dva različita pristupa u vezi sa analizom teksta u finskom jeziku sa izuzetnom raznovrsnošću oblika reči, prikazana su u [29] - gde se mašinski rečnik sastoji od

standardnih elemenata običnog rečnika, i u [34], gde se svaki ulaz u rečnik snabdeva informacijama o početnom delu reči, karakterističnim nastavcima za razne oblike, fonološkim karakteristikama i obrascima promene. U sličnom kontekstu, (komunikacije na italijanskom jeziku sa bazom podataka), sistem PARNAX [13] koristi dvodelni rečnik: leksički deo rečnika koji sadrži korene reči i modele promena, i semantički deo rečnika. U okviru projekta razumevanja prirodnog jezika autora sa Univerziteta u Pragu [22, 57], u rečnik se, osim osnovnih reči, uključuju i relacije sinonimije i homonimije, podredjenosti i nadredjenosti, kao i semantička svojstva reči - u specifičnom kontekstu tehničkih tekstova. Osim osnovnog rečnika, koriste se i konkordance kao veza rečničkog ulaza i njegovog pojavljivanja u tekstu.

2.3.3. MORFOLOŠKA PRAVILA

S obzirom da različiti oblici pravilnih reči nisu prisutni u rečniku, potrebna su morfološka pravila za analizu (tj. delimičnu sintezu) tih oblika, kako bi se tekst predstavio leksičkim tipom tj. ponovo vratio u izvorni oblik.

Primer takvog pravila u engleskom jeziku je sledeći:

- ako se reč u tekstu završava na "ies" a u rečniku postoji odgovarajuća imenica sa završetkom "y" umesto "ies", onda je reč iz teksta - imenica iz rečnika u množini.

Primer odgovarajućeg pravila u srpskohrvatskom jeziku je sledeći:

- ako se reč u tekstu završava na "ma" a u rečniku postoji odgovarajuća imenica bez tog završetka, onda je reč iz teksta - imenica iz rečnika u 3. padežu množine.

Morfološka pravila smeštena su u relaciji

završ.reči / završ.osn._reči / vrsta_reči / oblik_reči/ vrsta_osn._reči / obl.osn_reči / priraštaj

Završetak reči i osnovne reči (najčešće reči iz rečnika ali moguće i nekog drugog oblika) su grupe slova koje treba obrisati na kraju reči tj. dodati na tako dobijenu reč, redom, da bi se dobila reč iz koje se vrši izvodjenje reči koja se predstavlja (smer morfološke analize). U smeru morfološke sinteze, "završetak osnovne reči" treba obrisati iz oblika iz kog se vrši izvodjenje (najčešće reči u rečniku), i zatim dodati "završetak reči" da bi se dobio odgovarajući oblik reči (npr. na kraju reči "knjiga" ne treba brisati ništa, a treba dodati nastavak "ma" da bi se dobio 3. padež množine te imenice tj. reč "knjigama").

Vrsta reči, oblik reči, vrsta osnovne reči i oblik osnovne reči su sintaksna vrsta i oblik reči iz teksta tj. odgovarajuće osnovne reči, redom (npr. u primeru na srpskohrvatskom jeziku vrsta i oblik reči, tj. osnovne reči, bili bi "imenica, 3. padež množine ženskog roda", "imenica, 1. padež jednine ženskog roda").

Priraštaj izražava način na koji se izračunava leksička jedinica koja odgovara reči iz teksta, na osnovu leksičke jedinice koja predstavlja odgovarajuću osnovnu reč.

Relacija navedene strukture u relacionoj bazi podataka može se koristiti kao uniformna shema reprezentacije morfološkog znanja nezavisno od konkretnog jezika. Sadržaj i obim te relacije, medjutim, bitno će zavisiti od jezika. Tako, npr, zbog bogatstva oblika srpskohrvatskog jezika, broj pravila je mnogo veći, a pojava izvodjenja oblika iz oblika koji nije osnovni je česta (npr. "ljudima" se dobija iz 1. padeža množine ove nepravilne imenice, kao što se i "čvorovima" dobija iz 1. padeža množine pravilne imenice). Stoga u relaciji sa morfološkim pravilima za srpskohrvatski jezik "oblik osnovne reči", tj. oblik iz kog se

vrši izvodjenje oblika koji se predstavlja nije uvek oblik koji je prisutan u rečniku. Da bi se dobila odgovarajuća reč koja je ulaz u rečnik (osnovni oblik potreban za prepoznavanje ostalih), potrebno je rekurzivno primeniti morfološki analizator (npr. pravilo koje za oblik "čvorovi", briše nastavak "ovi").

Priraštaj vrednosti koda obezbedjuje jednoznačnost reprezentacije raznih nastavaka pravilnih reči (npr. nastavci "ovi" i "evi" za 1. padež množine imenica nose različite priraštaje, kao i ostali oblici koji se od njih grade). To omogućuje dekodiranje teksta odnosno konverziju teksta iz oblika predstavljenog leksičkim tipom podataka u izvorni oblik.

U eksperimentalnom sistemu za engleski jezik ima oko 70, a za srpskohrvatsi jezik oko 250 morfoloških pravila. Broj pravila, s obzirom na bogatstvo oblika, za naš jezik u suštini je i veći, jer su, radi ekonomičnosti zapisa, mnoga pravila zapisana rekurzivno, tj. umesto završetka, vrste i oblika osnovne reči, stoje završetak, vrsta i oblik nekog oblika različitog od osnovnog, koji se, pak, od osnovnog gradi nekim drugim pravilom. Npr. pravilo

ma/ - / im / mp1 / im / mp3 / +2

ma značenje: ako se u analiziranom obliku (npr. "pevačima"), risanjem nastavka "ma" i bez dodavanja ičega ("-"), dobije menica u 1. padežu množine muškog roda ("im, mp1" - "pevač"), nda je analizirani oblik - imenica u 3. padežu množine muškog oda ("im, mp3"), a vrednost koda je za 2 veća ("+2") od vrednosti oda odgovarajuće imenice u 1. padežu množine. Sada ostaje da se tvrdi, na osnovu drugog pravila, da li je dobijeni oblik "pevači") - imenica u 1. padežu množine muškog roda. To se može tvrditi pravilom

i/ - / im / ms1 / im / mp1 / 5

koje, slično prethodnom, kaže da je analizirani oblik – imenica u 1. padežu množine muškog roda, ako se brisanjem nastavka "i" dobije imenica u 1. padežu jednine muškog roda ("pevač"). Vrednost leksičke jedinice imenice u 1. padežu množine izračunava se dodavanjem broja 5 na leksičku jedinicu odgovarajuće imenice u 1. padežu jednine (iz rečnika).

Problemom morfološke analize autori se najčešće bave u širem kontekstu razumevanja prirodnog jezika. Osim morfološkog analizatora, predstavljenog skupom pravila u relacionoj bazi podataka i interpretatorom tih pravila, kao u ovom radu, drugi pristupi su prikazani u literaturi, uglavnom citiranoj pri razmatranju rečnika u tački 2.4.1. U [29], morfološka analiza za finski jezik je slična kao u ovom radu, definisana heurističkim pravilima (predstavljenim konačnim automatima) i poredjenjem sa rečnikom. Drugi rad o morfološkoj analizi i sintezi finskog jezika [34] primenjuje paralelno morfološka pravila implementirana konačnim automatima. U [22], u kontekstu sistema za pretraživanje teksta, koristi se morfološki generator svih oblika reči koje korisnik saopštava na ulazu (zajedno sa karakteristikama reči kao što su vrsta, rod, živo/neživo) i zatim se u rečniku konkordanci traže sva pojavljivanja unetih i generisanih reči i izraza. Slična vrsta morfološkog generatora za srpskohrvatski jezik opisana je u [70]. U sistemu TORUS [41], morfološki deo analize prirodno-jezičkog upita nad bazom podataka rešen je algoritamski. U IBM istraživačkom centru iskorišćena je lingvistička teorija pravila izgradnje reči za morfološki sistem zasnovan na pravilima [7]. Ovde se radi o generisanju raznih reči izvedenih iz iste osnovne reči (npr. primeniti, primena, primenljiv) a ne o raznim oblicima jedne reči (primeniti, primenim, primeniš, itd). Struktura morfoloških pravila je ipak slična onoj koja se koristi

ovom radu.

.3.4. IMPLEMENTACIJA

Procedura predstavljanja izvornog oblika teksta leksičkim ipom je sledeća: tekst se prvo skenira (izdvajaju se reći) a atim se reči zamenjuju leksičkim jedinicama i registruju rečenica o rečenica. Tekuća reč se prvo poredi sa rečnikom i izdvajaju se ečnički ulazi (reči sa svim atributima) koji zadovoljavaju oredjenje (iste slovne reprezentacije, npr. za reč "vek" rečnički laz "vek, im, m.rod-1.p.jedn, vek, -, -, TIM, id, deskr). Zatim e primenjuje interpretator morfoloških pravila koji u ovom lučaju radi kao morfološki analizator i izdvajaju se svi oblici eči slovno identični sa rečju koja se predstavlja leksičkom edinicom (npr. "vek, im, m.rod-4.p.jedn"). Ovaj proces odgovara vršavanju inverznog operatora lexform pri čemu se rezultat nverznog operatora poredi sa rečničkim ulazom. Ako je reč išeznačna - ima više interpretacija, kao u našem primeru, načava se tipom višeznačnosti (sintaksna/semantička); ako je dnoznačna, zamenjuje se svojom leksičkom jedinicom (id, deskr). atim se poziva procedura za razrešavanje višeznačnosti u celoj ečenici, koja aktivira procedure ekspertnog sistema za svaku reč rečenici označenu kao višeznačnu (ovaj deo sistema biće detaljno isan u sledećoj glavi). Leksikalizovani sadržaj se zatim edstavlja relacijom.

Interpretator morfoloških pravila kao deo sistema za edstavljanje teksta leksičkim tipom implementiran je za engleski zik. Neznatne izmene u ovom delu sistema su potrebne da bi dio i za srpskohrvatski jezik, i ogledaju se u potrebi za pštavanjem u odnosu na rekurzivnu primenljivost pravila.

Program za predstavljanje teksta leksičkim tipom

uključujući i procedure za razrešavanje višeznačnosti) iznosi oko

000 EQUEL linija.

- KONTEKSNO-ZAVISNA INFORMACIJA U LEKSIČKOM TIPU PODATAKA: PRIMENA EKSPERTNIH SISTEMA

Kada se tekst, kao skup reći u kontekstu, predstavlja ksičkim tipom podataka, potrebno je razrešiti leksičku šeznačnost reči. Na primer, slovna reprezentacija reči "smeša" edstavlja bar dva različita oblika i vrste reči. Pri procesu prezentacije teksta leksičkim tipom, ova niska simbola trebalo da se preslika u jednu od dve različite leksičke jedinice u visnosti od nameravanog značenja, koje se u većini slučajeva će izvesti iz konteksta.

Drugi konteksno-zavisni problem je odredjivanje referenata menica. Da bi se iz leksikalizovanog teksta dobilo što je moguće še informacije, na osnovu samih leksičkih jedinica, potrebno je sičke slike zamenica asocirati sa leksičkim jedinicama ili pama leksičkih jedinica na koje se odnose tj. koje zamenjuju. a potpuno lingvističko rešenje problema odredjivanja referenata enica ne postoji, delimični i korisni pristupi zasnovani na tekstu izgledaju intuitivno mogući.

U ovoj glavi biće izložena dva ekspertna sistema kojima se lementiraju sledeći operatori drugog nivoa hijerarhije sičkih operatora:

RAMB - za datu nisku simbola vraća jedinstvenu leksičku inicu (Resolving AMBiguity - razrešavanje višeznačnosti); rator se primenjuje na reči u kontekstu u vreme njihovog stavljanja leksičkim tipom. Formalno, ovaj operator slikava par skupova leksičkih jedinica (X_1, X_2) u jednu sičku jedinicu l. Skup X_1 je skup leksičkih jedinica koje imaju identifikator id (koji predstavlja nisku simbola), ali ićite deskriptore deskr; skup X_2 je skup leksičkih jedinica sa

iznim id, koje predstavljaju razne niske simbola specifičnog onteksta; leksička jedinica l je jedinstveni element skupa X_1.

PRONR - za datu leksičku jedinicu koja je slika zamenice, aća njen "original", tj. leksičku jedinicu ili niz leksičkih dinica koje predstavljaju reč ili izraz na koji se zamenica nosi (PRONoun Referencing - odredjivanje referenata zamenica).

Operatori su implementirani udruživanjem skupa jezičkih avila sa mehanizmom izvodjenja u okviru ekspertnih sistema. avila se interpretiraju sukcesivnom primenom primitivnih snovnih) operatora leksičkog tipa podataka nad leksičkim dinicama.

Pravila koja se koriste u ovim sistemima su zdravorazumske uristike i izgradjena su za testiranje pristupa (ne sa namerom obezbede realno primenljivi sistem). U tom smislu su ovi spertni sistemi više sistemi zasnovani na pravilima; u "pravom" spertnom sistemu lingvistički eksperti bi zadavali pravila, što doprinelo kvalitetu rezultata.

Pristup sistema zasnovanih na pravilima se uklapa veoma ro u implementaciju oba operatora iz više razloga. Prvo, eratori razmatraju niz različitih slučajeva. Mada je problem alize/odlučivanja uvek relativno jednostavan, on je različit za ki od slučajeva. Iz tog razloga, u svakom pojedinom slučaju menjuje se sasvim malo sintaksne (semantičke) analize, mada je pnost sintaksne i semantičke analize koja se može zahtevati i a se može primeniti dosta obimna. Ekspertni sistem je izuzetno odan u takvoj situaciji. Algoritamski pristup bi sadržao toliko go "if-then" iskaza, da bi predstavljao analogon sistemu novanom na pravilima samo bez njegove efikasnosti.

Ostale prednosti ekspertni sistemi duguju svojoj velikoj ksibilnosti. Dodavanje ili brisanje pravila vrši se lako, bez

ekta na proceduralni deo sistema. Redosled izvršavanja pravila najčešće nevažan. Opšti mehanizam izvodjenja ne mora se programirati za različite aplikacije. Ta svojstva omogućuju ko modifikovanje sistema s porastom iskustva i pojavom novih likacija.

U narednim delovima biće opisana svojstva i funkcije spertnih sistema, a zatim će biti izložena implementacija eratora RAMB, PRONR primenom ekspertnih sistema, i dokazana neka ojstva ovih operatora.

1. O EKSPERTNIM SISTEMIMA

Oblast ekspertnih sistema istražuje metode i tehnike nstruisanja sistema čovek - mašina, sa specijalizovanom spertizom za rešavanje problema (znanjem o specifičnom domenu, :umevanjem problema u domenu, i veštinom u rešavanju nekih od ı problema) [24].

Postoji više razloga za naglašavanje znanja nasuprot :odima formalnog rezonovanja [35]. Jedan je da većina teških i :eresantnih problema nema algoritamsko rešenje, jer proističe iz ženog konteksta (društvenog ili fizičkog), koji u opštem ıčaju ne dozvoljava precizan opis i strogu analizu. Ekspertni stemi se bitno razlikuju kako od konvencionalnih programa stema za obradu podataka) tako i od sistema razvijenih u raznim nama veštačke inteligencije. Od prvih, npr. simboličkom rezentacijom, simboličkim izvodjenjem, heurističkim traživanjem [72], od drugih, npr. sposobnošću da teške zadatke vršavaju na nivou eksperta, naglašavanjem domenski specifičnih ategija za rešavanje problema.

Definicija ekspertnog sistema, prema Feigenbaum-u [24] isti sedam polunezavisnih dimenzija:

* Ekspertiza – pravila visokog nivoa, izbegavanje slepog traženja, efikasnost koja dolazi iz višegodišnjeg iskustva na datom zadatku;

* Rezonovanje manipulisanjem simbola, jer se "znanje" sastoji većim delom u simboličkom predstavljanju činjenica o svetu;

* Kombinovanje osnovnih principa domena sa opštim metodima rezonovanja da bi se dobila istovremeno fleksibilnost i efikasnost;

* Težina ili složenost domena – problem treba da je dovoljno komplikovan da bi zahtevao eksperta;

* Preformulisanje opisa u formu pogodnu za primenu ekspertnih pravila;

* Rezonovanje sistema o samom sebi, posebno za objašnjavanje zaključaka i razloga za primenu odabranih pravila;

* Tip zadatka koji se rešava ekspertnim sistemom.

Većina ekspertnih sistema pripada jednom od nekoliko tipova: interpretativni (izvode opis situacije iz opaženih elemenata – npr. razumevanje govora, analiza slike, odgonetanje hemijske strukture, interpretacija signala), sistemi predvidjanja (izvode verovatni zaključak iz date situacije – npr. predvidjanje vremena, demografska predvidjanja, predvidjanje saobraćaja, vojna prognoza), dijagnostički (izvode loše funkcionisanje sistema iz opaženih elemenata –npr. medicinska, elektronska, mehanička, softverska dijagnoza), sistemi projektovanja (razvijaju konfiguracije objekata koje zadovoljavaju ograničenja problema – npr. postavljanje elektronskih kola), sistemi planiranja (projektuju akcije – npr. automatsko programiranje, problem planiranja robota, projekta, putanje, eksperimenta, vojske), monitorski (porede zapažanja o ponašanju sistema sa svojstvima

koja izgledaju presudno za uspešnu realizaciju plana - npr. nadgledanje nuklearnih elektrana, vazdušnog saobraćaja), sistemi za analizu i otklanjanje grešaka (prepisuju "lekove" za loše funkcionisanje - npr. elementi ovih sistema postoje kao tekst-editori), sistemi popravke (razvijaju i izvršavaju plan za sprovodjenje "lečenja" za neki dijagnosticirani problem - npr. u održavanju letilica, računara), obrazovni (dijagnosticiraju, analiziraju studentsko ponašanje), kontrolni (vladaju ukupnim ponašanjem sistema - npr. kontrola leta, upravljanje poslovanjem, vodjenje bitke).

Idealni ekspertni sistem sadrži sledeće komponente:

1) baza znanja koja registruje pravila, činjenice i informacije o tekućem problemu;

2) rasporedjivač pravila kontroliše redosled izvršavanja akcije i primene pravila;

3) interpretator pravila proverava relevantnost uslova pravila i izvodi promene prepisane pravilom;

4) tabla - prostor za beleženje medju-hipoteza i odluka kojima se dalje manipuliše - pristupačna svim ostalim komponentama sistema;

5) jezički procesor za problemski-orjentisanu komunikaciju izmedju korisnika i ekspertnog sistema;

6) kontrolor saglasnosti - modifikuje prethodne zaključke kadgod se menjaju podaci koji su osnova za te zaključke;

7) "obrazlagač" koji daje razloge i objašnjava ponašanje sistema.

Medju raznovrsnim implementacijama osnovne ideje ekspertnih sistema - inteligentnog rešavanja problema, izdvaja se nekoliko principa arhitekture kao metode projektovanja strukture ekspertnih sistema. Pojedini ekspertni sistemi se medjusobno razlikuju (tj.

nalikuju) karakteristikama u pogledu složenosti prostora rešenja, kvaliteta podataka i znanja, deljivosti problema u potprobleme, nezavisnosti podproblema, mogućnosti nalaženja delimičnog rešenja, promenljivosti podataka u vremenu. Ove karakteristike i odredjuju poželjnu arhitekturu u svakom pojedinačnom slučaju. Neki od principa koji odgovaraju ovim karakteristikama i koji pojedine arhitekture razlikuju su konkretan/ apstraktan prostor rešenja, potpuno/ delimično pretraživanje prostora rešenja, monotono/ približno (jednostrano/ višeizvorno) rasudjivanje, homogenost/ heterogenost modela, olančavanje unazad/ unapred (od zaključka do premisa i odgovarajućih podataka, tj. u suprotnom smeru). Tako, npr. najjednostavniju organizaciju sistema zahteva ograničena klasa problema sa malim prostorom rešenja (npr. malim brojem mogućih dijagnoza), pouzdanim, fiksnim u vremenu podacima i pouzdanim znanjem - zadovoljavajuća organizacija uključuje potpuno pretraživanje prostora rešenja, monotono i jednostrano rasudjivanje (bez primene modela verovatnoće) i s jednim izvorom "dokaznog materijala"). Problemi sa složenijim karakteristikama primenjuju kompleksnije principe projektovanja.

Izgradnja ekspertnog sistema prolazi kroz razne faze - prikupljanja znanja od eksperta, identifikacije (odredjivanja karakteristika problema), konceptualizacije (nalaženje koncepata kojima se znanje predstavlja, karakteristika toka informacije i ograničenja), formalizacije (projektovanja struktura za organizaciju znanja), implementacije (formulisanje pravila koja sadrže znanje i kontrolne strategije) i testiranja (ocena rada prototipnog programa i potrebne korekcije).Neke od ovih faza (npr. formalizacija, implementacija) zahtevaju posebna sredstva za izvodjenje (orudja za izgradnju ekspertnog sistema). Ova orudja spadaju u jednu od nekoliko grupa: programski jezici opšte namene

(npr. lisp, programski jezik "C"), skeletni sistemi (npr. emycin), jezici za reprezentaciju - opšte namene (npr. rosie).

Pri korišćenju programskih jezika opšte namene, podjezik pravila je, npr. oblika:

<pravilo> ::= (if {<antecedens>}+ then {<konsekvens>}+)

<antecedens> ::= <asocijativna_trojka>

<konsekvens> ::= <asocijativna_trojka>

<asocijativna_trojka> ::=(<objekat>,<atribut>, <vrednost>),

a mehanizam izvodjenja se piše kao procedura u tom programskom jeziku, koja proverava istinitost antecedensa i unosi konsekvens u bazu podataka, tj. izvršava odredjenu akciju. Ekspertni sistem za bilo koji domen može se izgraditi u programskom jeziku opšte namene, ali je potrebno obaviti celokupno programiranje.

Skeletni sistemi pružaju projektantu ekspertnog sistema mogućnost korišćenja unapred definisane forme pravila i procedure izvodjenja. Formu pravila treba onda ispuniti znanjem (sadržajem) iz specifične oblasti. Primeri skeletnih sistema koji služe za konstruisanje dijagnostičkih sistema su emycin, kas, expert, i njihova pravila mogu se ugraditi u složenu strukturu semantičkih mreža. Kontrolna strategija uključuje model približnog rasudjivanja (težine konsekvensa) koji će biti opisan u sledećoj tački kao procedura izvodjenja sistema mycin (od kojeg je emycin evoluirao uopštavanjem domena). Primena skeletnih sistema je dosta jednostavna, pod uslovom (često prejakim) da se problem potpuno uklapa u strukturu pravila i kontrolne strategije.

Programski jezici za izgradnju ekspertnih sistema opšte namene (npr. rosie, ops5, rll, hearsay-III) su manje ograničeni od skeletnih sistema pošto nisu vezani za posebni domen, ali je i njihova primena nešto teža nego primena skeletnih sistema. U jeziku rosie (Rand-korporacija) ili hearsay-III(Carnegie-Mellon

Univerzitet), npr, osim što se može isprogramirati sve što i u programskom jeziku opšte namene, baza "iskaznih" podataka (iskaza) može jednostavno biti kreirana, manipulisano njom i dostupna stilizovanim engleskim jezikom (u sistemu hearsay-III osnovu predstavlja relacioni sistem baza podataka i njegova kontrolna strategija). Posebno se održava baza deduktivnih pravila izvodjenja i ograničenja.

Bez obzira koje se sredstvo za izgradnju ekspertnog sistema koristi, ono mora da obezbedi funkcionisanje komponenti sistema. Funkcionisanje proceduralnih komponenti (npr. kontrolne strategije, interpretatora pravila) podržano je i teorijski (algoritam mehanizma izvodjenja, model verovatnoće). Izricanje pravila prepušteno je ekspertu, ali i ta komponenta (baza znanja) treba da ispunjava neke zahteve - npr. kompletnost i konzistentnost u smislu da za svaku dozvoljenu situaciju postoji pravilo u bazi koje reguliše tu situaciju, tj. da baza znanja daje jednoznačan savet [40].

3.1.1. MODELI PRIBLIŽNOG REZONOVANJA

S obzirom da pravila često operišu nad nepotpunim podacima ili podacima čija se korektnost može utvrditi samo do nekog stepena, posebno pogodno svojstvo sistema zasnovanih na pravilima je što oni omogućuju jednostavnu ugradnju nekog modela verovatnoće ili, opštije, modela neegzaktnog (nepreciznog) rasudjivanja. U nizu pristupa izvodjenja iz nesigurnih ili nekompletnih podataka, izdvajaju se, s jedne strane, matematički strogi i, s druge strane, neformalni i intuitivni modeli. Nedostatak prvih je obično nepostojanje adekvatnih statističkih uzoraka zbog čega je nužno osloniti se na subjektivne procene eksperata. Nedostatak drugih su teškoće u formalnom dokazivanju efekata tih modela.

Stoga poseban značaj dobijaju pokušaji da se dve krajnosti pomire.

Jedan od metoda koji koristi prednosti oba pristupa – formalnog i intuitivnog je "subjektivno Bajesovsko izvodjenje" [17]. U ovom pristupu pretpostavlja se da ekspert daje pravila oblika "ako E onda H", i (umesto verovatnoća dobijenih statističkim uzorkom) niz podataka (otuda "subjektivan" metod) – apriornu verovatnoću P(H) zaključka, "snagu" pravila $\lambda(>=0)$ koja, kada je velika ($\lambda>>1$), izražava činjenicu da je E dovoljan uslov za H, i "snagu suprotnog" pravila – ("ako nije E onda H") $\bar{\lambda}(>=0)$ koja, kada je mala ($\bar{\lambda}<<1$) izražava činjenicu da je E neophodan uslov za H (formalno, $\lambda = P(E|H)/P(E|\bar{H})$, $\bar{\lambda}=P(\bar{E}|H)/P(\bar{E}|\bar{H})$). Primenom Bajesovog pravila i podataka koje je dao ekspert, izračunava se uslovna verovatnoća P(H|E). Podaci koje daje ekspert moraju da zadovoljavaju neke uslove ovog formalizma. Tako, npr. s obzirom na vezu izmedju λ i $\bar{\lambda}$, iako je nužno da ekspert eksplicitno obezbedi obe vrednosti, on to treba da uradi tako da, ako tvrdi da prisustvo antecedensa E uvećava verovatnoću zaključka H ($\lambda>1$), onda treba da tvrdi i da odsustvo antecedensa E umanjuje verovatnoću od H ($\bar{\lambda}<1$). Ovo je mesto na kom Bajesov formalizam protivreči intuiciji (ekspert najčešće daje iskaz tipa "prisustvo antecedensa E uvećava verovatnoću zaključka, ali odsustvo od E nema značaja").

Drugi problem nastaje ako se pretpostavi (što je najčešće slučaj) da je i sam antecedens E tekućeg pravila – zaključak nekog drugog pravila ("ako E' onda E"), i da se to može rekurzivno ponavljati proizvoljan broj puta. Podaci o primarnim verovatnoćama zaključaka svih pravila dolaze od eksperta, pa stoga obično nisu konzistentni, tj. ne zadovoljavaju formule uslovne verovatnoće (npr. za P(H|E'), koja je linearna funkcija uslovne verovatnoće P(E|E') sa koeficijentima P(H|E) – P(H|$\bar{E}$), P(H|$\bar{E}$). Oba problema se

u subjektivnom Bajesovskom metodu prevazilaze raznim modifikacijama (interpolacijom) linearne funkcije (npr. za P(H|E)) izlomljenom linijom koja obezbedjuje ponašanje funkcije verovatnoće saglasno intuiciji, a u graničnim tačkama - konzistentnost.

Mada je Bajesov formalizam zgodno primeniti na kvantifikovanje snage pravila i zaključaka, sam pojam verovatnoće upotrebljen za opisivanje verodostojnosti realnog sveta protivreči intuiciji.

Shortliffe [58] iznosi sledeće paradokse: neka je h_1 hipoteza "svi gavrani su crni", a h_2 neka je logički ekvivalentna hipoteza "nijedna stvar koja nije crna - nije gavran". Ako se uspostavi analogija sa uslovnim verovatnoćama, dobije se da je $P(h_1|e)=P(h_2|e)$ za svako e. Medjutim, suprotno je intuiciji tvrdjenje da postojanje, npr. zelene vaze ide u prilog hipotezi h_1, dok izgleda kao da ide u prilog hipotezi h_2. Stoga se za verodostojnost hipoteza h_1 i h_2 moraju upotrebiti neke druge mere različite od verovatnoće, jer priroda kvantifikovanja ovih hipoteza ne odgovara pojmu verovatnoće. U oblasti medicine, npr. ako ekspert odredi da je težina zaključka dijagnoze H_1 pod uslovom simptoma S_1 jednaka 0.7 ($P(H_1|S_1)=0.7$), on će teško ustvrditi da je verovatnoća odsustva dijagnoze H_1 pod istim uslovom jednaka 0.3 ($P(\bar{H}_1|S_1) = 1- P(H_1|S_1) = 0.3$).

Neadekvatnost verovatnoće u analizi problema realnog sveta dovela je do niza alternativnih pristupa. Jedan od njih je i teorija potvrdjivanja (confirmation theory). Potvrdjivanje je interpretacija verovatnoće koja izražava stepen u kom dokazni iskaz podržava hipotezu (potvrdjivanje ne ukazuje da je hipoteza dokazana, već da joj zapažena pojava povećava izglede). Mera podrške se obično označava sa C[h,e] - stepen potvrdjivanja

hipoteze h baziran na uočenoj pojavi e. U teoriji potvrdjivanja paralelno sa funkcijom potvrdjivanja postoji (nezavisno) i funkcija opovrgavanja (disconfirmation) jer C[h,e] ≠ 1 - C[not.h,e], tj. "potvrdjivanje nečega u bilo koliko maloj meri nije uopšte opovrgavanje, jer u mnogim slučajevima dokaz u prilog neke hipoteze ne daje nikakvu podršku suprotnoj hipotezi" [58].

Model približnog rasudjivanja koji nudi Shortliffe [58], baziran na teoriji potvrdjivanja, predstavlja aproksimaciju uslovne verovatnoće. Za svako pravilo (ako e onda h) ekspert zadaje meru podrške C uočene pojave hipotezi h, koja je u ovom modelu."faktor izvesnosti" CF[h,e] ("Certainty Factor"). Faktor CF uključuje u sebe "meru uvećanog poverenja" u hipotezu h na osnovu pojave e (MB[h,e] - "Measure of increased Belief" i "meru uvećanog nepoverenja" u hipotezu h na osnovu pojave e (MD[h,e] - "Measure of increased Disbelief"), tj. CF[h,e] = MB[h,e] - MD[h,e]. Mere MB, MD definisane su tako da odražavaju odnose verovatnoća na sledeći način: ako uočavanje pojave e uvećava izglede za hipotezu h (tj. ne utiče na te izglede) (u terminima teorije verovatnoće, uslovna verovatnoća veća ili jednaka od apriorne, P(h|e) >= P(h)), onda je MB - mera tog uvećanja: MB= (P(h|e) - P(h)) / (1 - P(h)); ako je hipoteza h izvesna, (P(h) = 1), ova mera uvećanja je 1. Dakle, 0<=MB[h,e]<=1. Simetrično se definiše MD. Ove mere se odnose tako da je (saglasno sa intuicijom) mera uvećanja poverenja u hipotezu jednaka meri umanjenja nepoverenja u suprotnu hipotezu (MB[h,e] = MD[not.h, e]). Zato, ako se težine dodeljene zaključcima pravila "ako e onda h" interpretiraju kao CF[h,e] umesto kao uslovne verovatnoće, drugi od ranije navedenih paradoksa se rešava činjenicom da CF[h,e] ≠ 1-CF[not.h,e], tj. CF[h,e] = -CF[not.h,e]. Mada odstupa od verovatnoće, faktor CF u graničnim slučajevima odražava prirodu verovatnoće, npr. kada je

apriorna verovatnoća hipoteze mala ($P(h)\sim 0$), faktor CF je približno jednak uslovnoj verovatnoći $P(h|e)$. Intuitivno, faktor CF prevazilazi i probleme ilustrovane prvim paradoksom ako se "težina" hipoteze "svi gavrani su crni" interpretira kao faktor CF umesto verovatnoće ("ne znam verovatnoću da su svi gavrani crni, ali znam da kadgod mi se pokaže novi crni gavran, moje verovanje da su svi gavrani crni uvećava se za X").

Da bi numerička karakteristika pravila, dobijena od eksperta, mogla da se interpretira kao faktor CF i da pritom vodi razumnom ponašanju, faktori CF moraju da zadovolje neke uslove, kao npr. gornju granicu zbira faktora CF uzajamno isključivih hipoteza. Ako ima k takvih hipoteza h_i, onda je $\sum_{i=1}^{k} CF[h_i,e] <= 1$, što je bitno pri dodeljivanju faktora CF pojedinim pravilima. Ako, npr. ekspert odredi da je $CF[h_1,e]=0.7$ i $CF[h_2,e]=0.4$ a h_1 i h_2 se uzajamno isključuju, onda faktori CF nisu dobro odredjeni i moraju se ili korigovati ili "normalizovati" (da zbir ne predje 1).

Za definisanje mere postoje i pravila kombinovanja. Npr. 1) MB za dátu hipotezu približava se izvesnosti sa svakim novim elementom koji hipotezu potvrdjuje ($MB[h, e_1 \text{ i } e_2] = MB[h,e_1] + MB[h,e_2]*(1-MB[h,e_1])$); 2) MB za konjunkciju hipoteza je minimum od MB za dve hipoteze; 3) ako je uočena pojava e - hipoteza drugog pravila (ako e_1 onda e) sa pridruženim faktorom $CF[e,e_1]$, onda je pravo MB za hipotezu h - proizvod "poverenja" pravila ($MB[h,e]$) i "težine" pravila za e ($CF[e,e_1]$), i analogno za MD.

Opisani model približnog rasudjivanja razvijen je u kontekstu programa za dijagnosticiranje u medicini - za mycin sistem [58]. Izloženi model se primenjuje tako što se potpunim pretraživanjem za svaku moguću hipotezu, primenom svih pravila sa zaključkom jednakim toj hipotezi, skupljaju "pozitivni" i "negativni" dokazi (računaju kumulativno MB, MD, tj. faktor CF).

Za dokaz koji je sam hipoteza nekog pravila, primenjuju se formule za računanje faktora CF, konjunkcije hipoteza, tj. mera MB (MD) polazne hipoteze množi se izračunatim faktorom CF hipoteze - dokaza. Primenljivost modela širi se i na druge domene koji su dostigli izvesni nivo formalizacije ali ne potpuni, jer takvi domeni omogućuju definisanje pravila koja prosudjuju sa težinom $\neq 1$ (nisu trivijalna u smislu da su svi faktori CF jednaki 1). Broj premisa u pravilima mora biti ograničen (iz praktičnih razloga), a skupovi premisa raznih pravila sa istim zaključkom - nezavisni (da bi aproksimacija verovatnoće bila moguća).

3.2. RAZREŠAVANJE VIŠEZNAČNOSTI U LEKSIČKOM TIPU PODATAKA -OPERATOR RAMB

Problem leksičke višeznačnosti (sintaksne i semantičke) je jedan od osnovnih problema u analizi prirodnog jezika, široko priznat kao težak i važan [4]. Birnbaum-ova analiza u [4] pokazuje da većina pristupa rešavanju problema koristi sintaksne analizatore u razrešavanju sintaksne višeznačnosti i "izborna ograničenja" (selectional restrictions) ili mehanizme "skriptalnih leksikona" (scriptal lexicons) za razrešavanje semantičke višeznačnosti. Izborna ograničenja predstavljaju neku vrstu pravila u pogledu definisanih tipova ili semantičkih kategorija reči koje mogu da idu zajedno (stoje jedna pored druge), čime se eliminišu, uz pomoć konteksta, parazitska značenja. Skriptalni leksikoni predstavljaju tematske rečnike u kojima su semantički višeznačnim rečima pridružena jedinstvena značenja u okviru te teme.

Pristup u ovom radu je različit i koristi se ekspertnim sistemom kao mehanizmom RAMB-operatora razrešavanja i sintaksne i semantičke višeznačnosti (njihovo razrešenje je ponekad

paralelno).

Prema klasifikaciji ekspertnih sistema (Hayes-Roth, Waterman, Lenat, [24]), sistem projektovan za razrešavanje višeznačnosti je interpretativnog tipa. Osnovne komponente sistema su:

- baza znanja koja se sastoji od rečnika i jezičkih pravila za razrešavanje višeznačnosti i

- kontrolna strategija, tj. procedura izvodjenja koja primenjuje pravila.

Baza znanja, osim konteksno-nezavisne informacije sadržane u rečniku, sadrži i konteksno-zavisnu informaciju o pojedinom obliku reči i neke apriorne procene verodostojnosti te informacije. Procedura izvodjenja ima dve komponente: opštu i posebnu. Prva ima ulogu rasporedjivača koji kontroliše redosled primene pravila u datoj situaciji. Druga ima ulogu interpretatora koji primenjuje pojedinačna pravila iz baze.

Arhitektura ekspertnog sistema je izabrana na bazi znanja, podataka i prostora rešenja specifičnih za problem koji se rešava. Koristeći terminologiju Stefika [24], problem se može okarakterisati malim prostorom rešenja (svega nekoliko mogućnosti), nepouzdanošću podataka i znanja (kontekst koji se koristi za razrešavanje višeznačnosti reči može i sam biti višeznačan, a pravila koja predstavljaju znanje nisu apsolutno tačna), i fiksnim (vremenski nepromenljivim) podacima. Za takvo okruženje Stefik sugeriše organizaciju ekspertnog sistema koja primenjuje potpuno pretraživanje prostora rešenja i kombinuje dokaze iz višestrukih izvora sa modelom verovatnoće. Stoga je strategija koja se primenjuje nalik na strategiju ekspertnog sistema mycin za dijagnosticiranje u medicini [58]. Ona se odražava kako na strukturu pravila tako i na organizaciju

procedure izvodjenja.

Pravila u sistemu su oblika

(antecedens, konsekvens, težina),

gde antecedens precizira skup uslova pod kojima je pravilo primenjivo, konsekvens je zaključak a težina je mera pouzdanosti dodeljena zaključku. U većini slučajeva u engleskom jeziku višeznačnost je u vrstama reči (npr. imenica/glagol), dok je u srpskohrvatskom jeziku i u oblicima reči (npr. imenica u 1. padežu jednine ili 2. padežu množine ili glagol). Na primer, reč "points" može biti ili imenica u množini ("tačke") ili glagol u 3. licu jednine prezenta ("pokazuje"). U izrazu "set of points" ("skup tačaka") višeznačnost se može lako razrešiti primenom pravila koje izražava našu veru da je "kombinacija predlog-imenica mnogo verovatnija nego kombinacija predlog-glagol". Pravilo bi se moglo izraziti u obliku:

antecedens: ako je X imenica ili glagol, i X sledi za predlogom

konsekvens: onda je X imenica

težina: sa težinom 0.9.

Ako (id[poz], deskr[poz]) označava leksičku jedinicu koja predstavlja reč na mestu poz u tekstu, i ako je (id_i[poz], $deskr_i$[poz]) - i-ti kandidat za leksičku jedinicu koja predstavlja višeznačnu reč na mestu poz, onda se ovo pravilo interpretira kao primena sledećih primitivnih leksičkih operatora nad leksičkim jedinicama:

antecedens: (vrsta(id_1[poz], $deskr_1$[poz]) = imenica ili
vrsta(id_2[poz], $deskr_2$[poz]) = glagol) i
vrsta(id[poz-1], deskr[poz-1]) = predlog

konsekvens: (id[poz], deskr[poz]) = (id_1[poz], $deskr_1$[poz]).

Rasporedjivač je organizovan tako da vrši potpuno

pretraživanje skupa pravila primenjivih u datoj situaciji, osim ako utvrdi razrešenje višeznačnosti sa sigurnošću (sa težinom 1), kada prestaje sa pretraživanjem. U njega je takodje ugradjena kontrolna strategija olančavanja unazad, pri čemu je pretraživanje vodjeno hipotezom: od mogućeg rešenja za datu višeznačnu reč (konsekvensa) ka odgovarajućem antecedensu i ka potrebnim podacima.

Kao metod približnog rasudjivanja koristi se unekoliko modifikovana mycin-strategija. Razmatraju se samo pravila koja potvrdjuju - ne ona koja opovrgavaju zaključak (tj. MD=0 za svako pravilo). Težine, stoga, odgovaraju MB (=CF). Prikupljanje dokaza nije kumulativno, već se bira hipoteza dobijena pravilom sa najvećom težinom (jer ni uslov nezavisnosti premisa različitih pravila sa istim zaključkom nije u potpunosti zadovoljen, a oblast primene čini prihvatljivijim zaključak dobijen jednim pravilom sa, npr. težinom 0.9, nego drugi zaključak dobijen primenom dva pravila sa težinama, npr. 0.8 i 0.6). Funkcija kombinovanja hipoteza je, kao i u sistemu mycin, minimizacija, a konačna težina zaključka je, kao i tamo, proizvod težine pravila i težine premisa (ako su one same - zaključci nekih drugih pravila).

3.2.1. PROCEDURA IZVODJENJA OPERATORA RAMB

Precizna interpretacija višeznačne reči može da zavisi od reči koje su takodje višeznačne. Stoga se za vreme izvršavanja pojedinačnog testa (testa nad jednom leksičkom jedinicom) iz antecedensa pravila, procedura izvodjenja može pozvati rekurzivno. Kolekcija medjusobno zavisnih višeznačnih reči predstavlja se grafom zavisnih višeznačnosti koji se sastoji od: skupa veišeznačnih reči, odnosa "poziva - poziva se" medju njima, statusa reči čija se višeznačnost razrešava i dodatne informacije koja će

biti kasnije opisana. Čvorovi grafa predstavljaju višeznačne reči a grane predstavljaju odnos "poziva - poziva se". Reči se numerišu u redosledu kojim se unose u graf. Graf je dinamička struktura koja se širi pri dodavanju višeznačnih reči i skuplja pri njihovom razrešavanju. U sledećoj tački biće opisani testovi dozvoljeni u sistemu, kako oni utiču na graf zavisne višeznačnosti, i struktura grafa kojom može da rezultuje primena pravila koja obuhvataju takve testove.

(i) TESTOVI

U grafovima će se koristiti sledeći simboli:

** - jednoznačna reč

* - višeznačna reč

/ - uslovno razrešena reč (zavisno od razrešenja drugih reči)

X --> Y - razrešenje višeznačnosti reči X zahteva testiranje tipa reči Y (X poziva testiranje Y).

U sistemu su dozvoljena četiri tipa testa. Pretpostavimo da je test sastavni deo antecedensa j-tog pravila (r_j) primenjenog na razrešavanje višeznačnosti reči N_i u grafu, da test testira tip t neke reči, i da pravilo r_j dodeljuje reči N_i vrstu v_i (kao zaključak). Tada su grafičke reprezentacije i značenja moguća četiri testa (T_1 - T_4) sledeći:

T_1: $N_i:v_i$ * ——t——> ** : testira tip t jednoznačne reči (ispituje da li je jednoznačna reč tipa t);

T_2: $N_i:v_i$ * ——t——> N_k * : testira tip t višeznačne reči;

T_3: $N_{i-1}:v_{i-1}$ * <——t—— $N_i:v_i$ * : (petlja) - testira tip t prethodnog, u numeraciji grafa, čvora (reči) N_{i-1}. Reč N_{i-1} je prethodno već testirala tip t1 reči N_i (i ti

N_i u graf), kao deo antecedensa pravila koje joj pridružuje vrstu v_{i-1} kao zaključak.

T_4: $N_i:v_i$ —t→ $N_{i+1}:v_{i+1}|v_i'$: testira tip t prethodno uslovno razrešene višeznačne reči N_{i+1} (pod pretpostavkom hipotetičke vrste v_i' reči N_i – ovakav test rezultuje kreiranjem novog primerka čvora N_{i+1} ako je $v_i \neq v_i'$ ili $t \neq v_{i+1}|v_i'$).

U sadašnjem sistemu pretpostavljaju se samo testovi tipa T_1 za razrešavanje <u>semantičke</u> višeznačnosti, pa je graf zavisne višeznačnosti trivijalan za semantičke višeznačnosti i dalje neće biti diskutovan.

Pri razrešavanju <u>sintaksne</u> višeznačnosti dozvoljena su sva četiri tipa testa sa ograničenjima broja T_3, T_4 testova koji mogu biti primenjeni u jednom grafu. Svaki od testova ima, kao rezultat, odgovarajuću težinu. Težina testa T_1 je ili 1 (reč zadovoljava testiranu vrstu) ili 0 (reč ne zadovoljava testiranu vrstu). Težina testa T_2 je težina zaključka o vrsti višeznačne reči N_{i+1} dobijenoj primenom procedure RAMB. Težina testa T_3 je ili 1 (vrste višeznačnih uzajamno zavisnih reči, iz para uzajamno primenjenih pravila, slažu se, npr. "N_{i-1} je imenica ako je N_i pridev" – prethodno pravilo, "N_i je pridev ako je <u>N_{i-1} imenica</u>" – tekuće pravilo, podvučen je T_3-test), ili 0 (vrste se ne slažu, npr. "N_{i-1} je imenica ako je N_i pridev" – prethodno pravilo, "N_i je pridev ako je <u>N_{i-1} prilog</u>" – tekuće pravilo, T_3-test podvučen). Težina T_4-testa je težina zaključka o uslovnoj vrsti višeznačne reči N_{i+1}, ako vrsta uslovno razrešene reči zadovoljava test, npr: "ako je N_i imenica, primenom svih pravila zaključuje se da je N_{i+1} pridev" – uslovno razrešenje, "N_i je imenica <u>ako je N_{i+1} pridev</u> – tekuće pravilo, podvučen je T_4-test; težina T_4-testa je težina zaključka o vrsti višeznačne reči N_{i+1}, dobijenoj ponovnom primenom operatora RAMB, a pod pretpostavkom vrste za N_i koju joj dodeljuje tekuće pravilo, inače.

(ii) ALGORITAM ZA RAMB

U ovoj tački biće dat opšti rekurzivni algoritam proceduralnog dela RAMB-operatora, tj. njegove kontrolne strategije. Algoritam je napisan u "C"-olikom jeziku dopunjenom izvesnom grafičkom notacijom, i eksplicitno pokazuje kako različiti testovi u različitim kontekstima, kao i procedura izvodjenja uopšte, utiču na graf zavisne višeznačnosti. Algoritam se primenjuje na svaku reč u rečenici, čija je višeznačnost tipa "vrsta reči".

Pretpostavke o pravilima i višeznačnostima, pod kojima algoritam radi su sledeće:

- antecedens (uslovi) pravila sastoje se samo od testova tipa T_1-T_4;

- ne postoje dve petlje na raznim čvorovima u grafu (ne postoje dva različita para uzajamno zavisnih višeznačnih reči u grafu - razlog za ovu pretpostavku je tehničke prirode - složenosti algoritma);

- ne postoje dva izvesna protivrečna pravila (oba sa težinom 1, različitim zaključcima kao kandidatima za višeznačnu reč, i testovima na raznim čvorovima ili istim testovima na istim čvorovoma).

Procedura izvodjenja operatora RAMB koja se poziva za reč N_1, primenjuje svako pravilo primenjivo na tu reč, u opadajućem redosledu težina. Svako pravilo dodeljuje par (vrsta, težina) reči N_1, i to primenom skupa testova tipa T_1 - T_4, od kojih svaki može rezultovati širenjem grafa zavisne višeznačnosti. Svaki od testova testira vrstu jedne reči iz uslova u antecedensu pravila, i kao rezultat ima težinu opisanu u prethodnoj tački. Težina antecedensa pravila, koji uključuje proizvoljan broj testova T_1-

T_4, odredjuje se kao minimalna težina primenjenih testova. Ukupna težina zaključka o vrsti koja se tekućim pravilom dodeljuje reči odredjuje se kao proizvod težine uslova i težine pravila.

Pošto se na razrešavanje višeznačnosti reči N_i primene sva primenjiva pravila, reči N_i se ili dodeljuje vrsta sa najvećom težinom, ili se razrešenje reči odlaže a reč se briše iz grafa. U slučaju da je vrsta dodeljena, može biti da je to definitivno dodeljivanje i u tom slučaju operator RAMB daje rezultat, ili to može biti uslovno razrešenje (uključeni T_3, T_4 - testovi), kada definitivno dodeljivanje vrste tek treba da se izvrši, a izvršavanje operatora RAMB se nastavlja.

```
      RAMB(reči (čvor Ni),vrstai,ti-težina zaključka)
/*na početku vrstai =' ',težinai (ti)=0 za sve 1<=i<=imax; i=1*/
      {
      if( reči nije uslovno razrešena) uzteż=0;
/*uzteż -  težina potencijalne uzajamne razrešenosti                */
      tteżina=0;
/*tteżina - težina dodeljena tekućim aktivnim pravilom              */
      for(svako primenjivo pravilo sa odgovarajućim tipom i
            težinom (rj: tipj,wj))
      {
        if(wj > ti)
        {
          uneti tip tipj uz čvor Ni  (vi=tipj);
/* tekuća vrsta koja se testira je vi  */
          težinai,j(uslovi)=1;
/* inicijalizacija ukupne težine uslova testiranih primenom     */
/*pravila rj na čvor Ni                                          */
          for(svaki test koji pravilo rj zahteva)
          {
            if(T1-test)
            {
              if(vrsta jednoznačne reči ne zadovoljava test)
                težinai,j(uslovi)=0;
            }
            else if(T2-test)
            {
              i=i+1;
              RAMB(reči ,vrstai ,ti );
              if(vrstai+1 zadovoljava test t)
                težinai,j(uslovi)=min(težinai,j(uslovi),ti+1);
              else težina i,j(uslovi)=0;
            }
            else if(T3-test)
            {
              if(test tipa T3 je već primenjen na čvoru
                 različitom od tekućeg - u grafu već postoji
```

```
   petlja, težina_ij(uslovi)=0;
else
{
  uneti u graf granu "poziva-poziva se" (N_i, N_i-1)
    obeleženu vrstom "t" testa;
  if(v_i-1 ne zadovoljava test t ili v_i ne
     zadovoljava prethodno primenjen test na N_i
     (t1 u oznaci testa T_3)
  {
    težina_ij(uslovi)=0;
      uztež=max(uztež,w_j);
    }
  }
}
else if(T_4 test)
{
  if(uslovno dodeljena vrsta vrsta_i+1|v_i' čvora n_i+1
     zadovoljava test t i v_i=v_i')
   težina_ij(uslovi)=min(težina_ij(uslovi),t_i+1);
  else
  {
   dodati grafu granu (N_i,N_i+1') obeleženu testom t;
   nova_vrsta_i+1=' ';
   nova_t_i+1=0;
   RAMB(novi_primerak reči+1 (sa čvorom N_i+1'),
        nova_ vrsta_i+1, nova_t_i+1);
   if(nova_vrsta v_i+1 zadovoljava test)
      težina_ij(uslovi)=min(težina_ij(uslovi),nova_t_i+1);
    else težina_ij(uslovi)=0;
   }
  }
 }
ttežina=težina_ij(uslovi) * w_j;
if(ttežina > t_i)
{
  vrsta_i=v_i;
  t_i=ttežina;
  if(u tekućem pravilu samo su testovi tipa T_1, T_2
     primenjeni)
  {
    izvršiti sledeće zamene u grafu:
    {
      N_i:v_i        N_i+1:v_i+1|v_i'                    N_i:v_i
      * -------> */*                                     *
        <-------                  =>
    }
    {
      if(t_i >=uztež)
      {
        uztež=0;
        N_i-1:v_i-1  t        N_i:v_i    => N_i-1:v_i-1              N_i:v_i
        * ----------------> *                * ----------------> *
          <------t1-------
      }
    }
  }
else if(u tekućem pravilu primenjeni su testovi tipa
        T_1,T_2,T_3)
{
  if(t_i >=uztež) uztež=0;
```

N_{i-1}:v_{i-1} t N_i:v_i => N_{i-1}:v_{i-1} t N_i:v_i

* <---------- * * <---------- *

<----t2----

-----t1--->

-----t1--->

}

else

{

if(novi primerak od N_{i+1} (N_{i+1}') kreiran i uslovno razrešen)

{

N_{i+1} = N_{i+1}';

$vrsta_{i+1}$ = nova_$vrsta_{i+1}$;

t_{i+1} = nova_t_{i+1};

izbrisati čvor N_{i+1}' i granu (N_i ,N_{i+1}');

}

}

else if(u tekućem pravilu primenjen test T_4 na čvor N_{i+1}' (novi primerak uslovno razrešene višeznačne reči))

izbrisati čvor N_{i+1}' i granu (N_i ,N_{i+1}');

}

}

if(t_i == 0)

{

ne dodeljuje se vrsta i-toj reči;

i=i-1;

}

else if(podgraf (N_{i-1}, N_i) je oblika N_{i-1} * --t--> * N_i (tj. oblika N_i *))

{

zameniti ga sa N_{i-1} * -----> ** (tj. **);

zameniti N_i odgovarajućom leksičkom jedinicom za $vrstu_i$ (rezultat RAMB-a);

izbrisati čvor N_i i granu (N_{i-1}, N_i) iz grafa, ako postoji;

if(čvor N_{i-1} uslovno razrešen) i=i-2;

else i=i-1;

}

else if(podgraf (N_{i-1}, N_i) je oblika N_{i-1} * <--t1-- * N_i)

{

zameniti ga sa N_{i-1} * <------ */* N_i:$vrsta_i$|v_{i-1};

------>

čekati sa razrešenjem dok se N_{i-1} ne razreši;

}

else if(podgraf (N_i, N_{i+1}) je oblika

N_i * <------ */* N_{i+1}:$vrsta_{i+1}$|$vrsta_i$)

------>

{

čvor N_i i N_{i+1} su razrešeni;

zameniti reči odgovarajućim leksičkim jedinicama za vrste i, i+1 (rezultat RAMB-a);

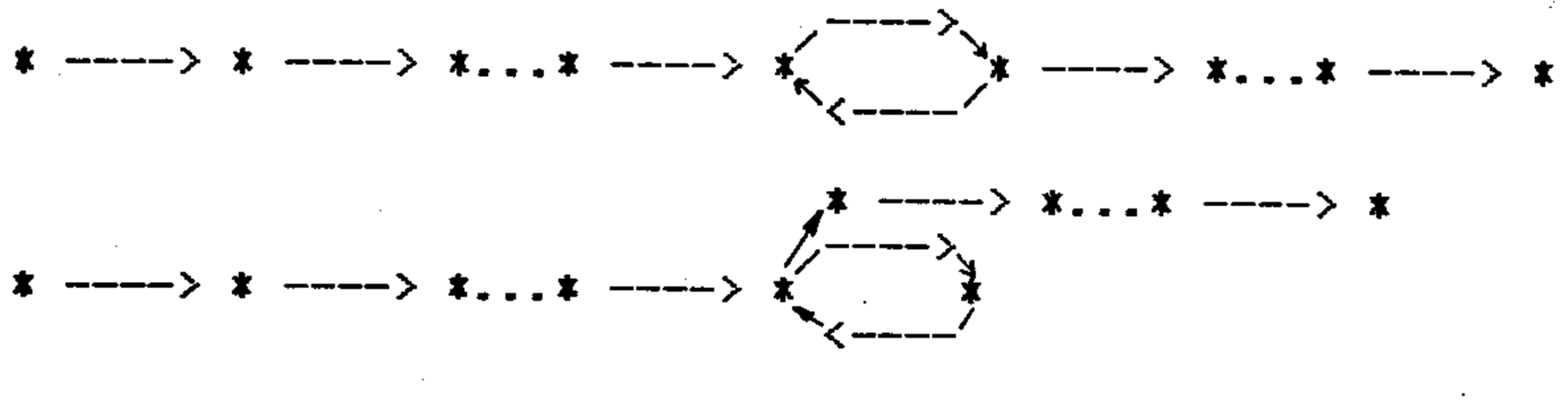
izbrisati oba čvora iz grafa;

i=i-2;

}

}

Rezultujući izgled grafa zavisne višeznačnosti je sledeći:



3.2.2. SVOJSTVA OPERATORA RAMB

Sledeće dve teoreme opisuju rešenja dobijena primenom procedure izvodjenja RAMB-operatora, tj. rad algoritma RAMB.

Teorema 1: Neka je $N=(N_1,N_2,...N_n)$ maksimalni niz zavisno višeznačnih reči u redosledu u kom se zahteva njihovo razrešavanje primenom procedure RAMB, i $S=(S_1,S_2,...,S_n)$ niz skupova odgovarajućih vrednosti. Neka je, dalje, $P=\{r_j : (v_j, w_j)\}$ skup svih pravila u sistemu (pravila pridružuju vrstu v_j sa težinom w_j), i $c=(c_1:t_1, c_2:t_2,...,c_n:t_n)$ rešenje niza N dobijeno primenom operatora RAMB, tj. $N_i=c_i$ $(\in S_i)$ sa težinom $t_i > 0$. Tada za svako $1<=i<=n$, važi da je rezultat operatora RAMB nad rečju N_i, pri fiksiranim vrednostima $(c_1, ..., c_{i-1}, c_{i+1},...,c_n)$ i pripadnim težinama $(t_1,...,t_{i-1},t_{i+1},...,t_n)$, odredjen sa maksimalnom težinom koju dozvoljavaju postojeća pravila, tj.

$$t_i = \max_{x\in S_i} \max_j \{ w_j * \text{težina}_{ij}(\text{uslovi}) \mid v_j = x \ \&$$

$$\text{težina}_{ij}(\text{uslovi}) = \min_k \{1, t_k \mid c_k \text{ se testira testom } T_2 \text{ ili } T_4 \text{ pravila } r_j\}$$

Dokaz: Uočimo, prvo, da ni u jednom trenutku izvršavanja algoritma RAMB ne moraju svi čvorovi N_1, N_2,..., N_n da budu prisutni u grafu, jer je graf dinamička struktura u koju čvorovi ulaze i iz koje izlaze u toku izvršavanja algoritma. Iz istog razloga ni indeksi čvorova ne moraju da se poklapaju sa onima u grafu. Pri izvršavanju operatora RAMB nad čvorom (rečju) N_i, relevantni podgraf je, prema algoritmu RAMB, jednog od sledeća dva oblika:

1) $N_{i1}:v_{i1}$ * -------> * $N_i:v_i$ (ili $N_i:v_i$); u ovom slučaju razrešava se višeznačnost samo čvora N_i. Svi čvorovi čiju vrstu testiraju pravila za razrešavanje čvora N_i ili su jednoznačni ili im je višeznačnost već razrešena, tj. vrste c_k dodeljene sa težinama t_k. Na početku se čvoru N_i dodeljuje težina $t_i=0$, a zatim se posle svakog primenjenog pravila, težina t_i i vrsta čvora c_i zamenjuju težinom i vrstom dobijenom primenom tog pravila, ukoliko je dobijena težina veća od težine čvora. S druge strane, težina vrste x čvora N_i (t_i) dobijena primenom j-tog pravila, prema algoritmu, jednaka je proizvodu težine pravila (w_j) i težine uslova koji se testiraju primenom pravila r_j na čvor N_i (težina$_{ij}$(uslovi)). Stoga je

$$t_i=\max_{x\in S_i}\ \max_{j}\ \{w_j * \text{težina}_{ij}(\text{uslovi}) \mid v_j=x\}.$$

S obzirom na tekući slučaj, pravilo r_j koje maksimizira t_i ne primenjuje testove T_3, T_4. Zbog toga je, s obzirom na akciju pri T_1, T_2-testu iz RAMB-algoritma,

$$\text{težina}_{ij}(\text{uslovi})=\min_{k}\ \{1, t_k \mid c_k \text{ - vrsta jednoznačne ili višeznačne reči koja se testira pravilom } r_j\}$$

U ovom slučaju teorema je dokazana.

2) $N_i:v_i$ * -------> */* $N_j:v_j|v_i$; u ovom slučaju razrešava se višeznačnost oba čvora N_i, N_j. Kao i u prethodnom slučaju, svi čvorovi čiju vrstu testiraju pravila za razrešavanje višeznačnosti čvorova N_i, N_j, - osim ova dva, ili su jednoznačni ili im je višeznačnost već razrešena, tj. vrste c_k dodeljene sa težinama t_k. Isti postupak dodeljivanja težina primenjuje se kao u slučaju 1). Za čvor N_i, pravilo koje maksimizira t_i ne primenjuje test T_3 (jer primenjuje test T_4 a najviše jedna petlja je dozvoljena u grafu u bilo kom trenutku). S obzirom na akcije pri T_1, T_2, T_4-testu iz RAMB-algoritma, važi ista ocena za težina$_{ij}$(uslovi), pa i ukupne vrednosti t_i. Za čvor N_j, pravilo koje maksimizira težinu t_j

primenjuje test T_s i prema akciji za T_s-test iz RAMB-procedure, težina tog testa je 1. Ostali testovi proizvode akcije koje se na težinu uslova odražavaju kao i u prethodnom slučaju.

Time je teorema dokazana.

U pravilima koja maksimiziraju težine čvorova podrazumeva se da testirani objekti (reči) zadovoljavaju testove (sa različitim težinama), jer je u protivnom težina testa, pa i celog pravila, 0, pa to pravilo ne dodeljuje vrstu čvoru. Kako je to pravilo maksimizirajuće, čvoru se i ne dodeljuje vrsta, odnosno RAMB i ne daje rešenje.

Razmatranja koja slede odnose se na "optimalno rešenje" skupa zavisno višeznačnih reči bez obzira na operator koji bi takvo rešenje proizveo. Primeri pokazuju da postojanje takvog rešenja zavisi od skupa pravila, a dokazuje se da, ako takvo rešenje postoji, operator (algoritam) RAMB ga nalazi.

<u>Definicija 2.</u> Neka su nizovi N, S i skup P kao u teoremi 1. Neka je $c=(c_1,c_2,\ldots,c_n)$ "rešenje" niza N, tj. $c_i \in S_i$, $i=1,2,\ldots,n$. Za $x \in S_i$,

$$\text{težina}(N_i:x \mid (c_1,\ldots,c_{i-1},c_{i+1},\ldots,c_n)) \overset{\text{def}}{=}$$

$$\max_j \{w_j \mid v_j=x \text{ i } (c_1,\ldots,c_{i-1},c_{i+1},\ldots,c_n \text{ i vrste jednoznačnih reči zadovoljavaju uslov iz } r_j)\}.$$

Tada je:

rešenje C "optimum" $\overset{\text{def}}{\iff}$ $\text{težina}(N_i:c_i \mid (c_1,\ldots,c_{i-1},c_{i+1},\ldots,c_n))=1$, za svako $i=1,2,\ldots,n$.

Definicija 2 kaže, drugim rečima, da je C optimum ako i samo ako se, fiksirajući vrste svih reči osim jedne na vrednosti iz C i birajući zatim "najbolju" vrstu za preostalu reč (zaključak pravila sa težinom 1), dobija C.

Sledeći primeri pokazuju da optimum može i ne mora da postoji (ako postoji mora biti jedinstven s obzirom na poslednju

od pretpostavki pod kojima radi algoritam RAMB), i da to zavisi od skupa pravila i ulaznih podataka (višeznačnosti). Nalaženje optimuma koji postoji je na proceduri izvodjenja odgovarajućeg operatora.

Primer 1.

Ulaz (rečenica sa višeznačnim rečima):

"It was a profound advance" ("To je bio istinski napredak");

$N=(N_1=profound, N_2=advance)$;

$S=(S_1=\{pridev, imenica\}, S_2=\{imenica, glagol\})$;

Skup pravila P:

P_1: Ako je reč imenica ili pridev i ako za njom sledi imenica, onda je reč pridev sa težinom 1.0;

P_2: Ako je reč imenica ili pridev i ako za njom sledi glagol, onda je reč imenica sa težinom 0.8;

P_3: Ako je reč imenica ili glagol i ako joj prethodi pridev, onda je reč imenica sa težinom 1.0.

Skup rešenja:

(profound,advance): $(c_1,c_2)=$ 1.(pridev,imenica)
2.(pridev,glagol)
3.(imenica,imenica)
4.(imenica,glagol)

1. težina(c_1 | c_2)=1.0
   težina(c_2 | c_1)=1.0 } optimum jedinstven
2. težina(c_1 | c_2)=0.0
   težina(c_2 | c_1)=0.0
3. težina(c_1 | c_2)=0.0
   težina(c_2 | c_1)=0.0
4. težina(c_1 | c_2)=0.8
   težina(c_2 | c_1)=0.0

Dinamika grafa:

profound:pr(1.0) advance:im(1.0)
* -------imenica----------> *
<------pridev-------------

profound:pr(1.0) advance:im(1.0) | profound:pr
* ----------------------------> */*
<----------------------------

profound:pridev,t_1=1.0 ** ** advance:imenica,t_2=1.0

RAMB nalazi optimum.

Primer 2.

Ulaz, N, S, skup rešenja: kao u primeru 1.

Skup pravila:

P_1': Ako je reč imenica ili pridev i ako za njom sledi imenica, onda je reč pridev sa težinom 0.85;

P_2': Ako je reč imenica ili pridev i ako za njom sledi glagol, onda je reč imenica sa težinom 1.0;

P_3' kao P_3 u primeru 1;

P_4': Ako je reč imenica ili glagol i ako nema drugog glagola (osim pomoćnog), onda je reč glagol sa težinom 0.8.

1. težina(c_1 | c_2)=0.85
   težina(c_2 | c_1)=1.0
2. težina(c_1 | c_2)=0.0
   težina(c_2 | c_1)=0.8
3. težina(c_1 | c_2)=0.0 Optimum ne postoji.
   težina(c_2 | c_1)=0.0
4. težina(c_1 | c_2)=1.0
   težina(c_2 | c_1)=0.8

Dinamika grafa:

```
profound:im(1.0)              advance:gl(0.8)
* --------glagol---------> *
  <-------pridev---------
uzteź=1.0
profound:im(0.8)              advance:gl(0.8)|profound:im
* --------glagol---------> */*
  <---imenica------------
profound:pr(0.85)             advance:im(1.0)
* --------imenica-------> *
  <-------pridev---------
profound:pridev,t1=0.85 **    ** advance:imenica,t2=1.0
```

RAMB nalazi rešenje koje maksimizira onaj čvor u paru uzajamno zavisno višeznačnih reči koji prvi ulazi u graf.

<u>Teorema 2.</u> Neka je N niz zavisno višeznačnih reči iz teoreme1. Ako postoji, u smislu definicije 2, optimalno rešenje niza N (ono je i jedinstveno s obzirom na poslednju od pretpostavki pod kojima radi algoritam), onda ga algoritam RAMB pronalazi.

<u>Dokaz:</u> Neka je $(i_1,i_2,...,i_n)$ permutacija niza $(1,2,...,n)$ takva da čvorovi iz N napuštaju graf zavisne višeznačnosti (zbog razrešenosti) u poretku $(N_{i1},N_{i2},...,N_{in})$ – N_{i1} prvi napušta graf – prvi je razrešen, N_{i2} za njim, itd, $N_{in}=N_1$ poslednji napušta graf; od dva čvora koji istovremeno napuštaju graf,

pretpostavljamo da prvi izlazi iz grafa onaj koji je kasnije ušao.

Neka je rešenje dobijeno algoritmom RAMB $(c_1, c_2, \ldots, c_n)$, tj. važi

$$t_1 = \max_{x \in S_1} \max_j \{w_j * \text{težina}_{1,j}(\text{uslovi}) \mid v_j = x\}.$$

Dokaz teoreme izvodi se indukcijom po i_j. Koraci u dokazu su:

1) Dokažimo da $c_{1,1}$ zadovoljava uslov optimalnosti; 2) Pretpostavimo da vrednosti prvih i_j čvorova koji izlaze iz grafa $(c_{1,1}, c_{1,2}, \ldots, c_{1,j})$ zadovoljavaju uslov optimalnosti; 3) dokažimo indukcioni korak, tj. da c_1 (vrednost čvora N_1) zadovoljava uslov optimalnosti.

1) Kako je $N_{1,1}$ prvi čvor koji napušta graf, pravila koja se koriste za razrešavanje višeznačnosti čvora $N_{1,1}$, pa i pravilo $r_{j,1}$ koje maksimizira težinu vrste $c_{1,1}$, primenjuju samo testove tipa T_1 ili/i T_3.

a) Ako se primenjuju samo testovi tipa T_1, onda je

$$\text{težina}_{1,1,j,1}(\text{uslovi}) = 1, \text{pa je}$$

$$t_{1,1} = \max_{x \in S_1} \max_j \{w_j \mid v_j = x \text{ i vrste jednozn. reči zadovoljavaju testove}\}$$

$$= w_{j,1}.$$

Kako postoji optimum, postoji $c_{1,1}'$:

$$\text{težina}(N_{1,1} : c_{1,1}') = \max_j \{w_j \mid v_j = c_{1,1}'\} = w_{j,1}' = 1.$$

Zbog neprotivurečnosti pravila $r_{j,1}$, $r_{j,1}'$, (poslednja od pretpostavki o algoritmu RAMB), $c_{1,1} = c_{1,1}'$, tj. $c_{1,1}$ zadovoljava uslov optimalnosti;

b) Neka pravilo $r_{j,1}$ koje maksimizira težinu vrste $c_{1,1}$ čvora $N_{1,1}$, primenjuje i T_3-test nad čvorom $N_{1,2}$. Kako čvor $N_{1,2}$ drugi napušta graf (posle čvora $N_{1,1}$), pravilo $r_{j,2}$ koje maksimizira vrstu $c_{1,2}$ čvora $N_{1,2}$ (kao i ostala pravila koja se koriste za razrešavanje višeznačnosti čvora $N_{1,2}$), ne primenjuje testove tipa T_3 (jer već postoji petlja $(N_{1,2}, N_{1,1})$), i primenjuje jedan test T_2 (ili T_4) nad čvorom $N_{1,1}$ i proizvoljan broj T_3-testova (ako bi

primenjivao samo T_1-testove, čvor N_{12} napustio bi graf pre čvora N_{11} što je suprotno pretpostavci). Kako pravila za maksimiziranje težina vrste c_{12} čvora N_{12} i c_{11} čvora N_{11} vrše uzajamno razrešavanje višeznačnih čvorova (N_{12}, N_{11}), za svaku vrstu c_{12}* koju pravila u opadajućem poretku težina dodeljuju čvoru N_{12} i koja primenjuju test T_2 (T_4) na N_{11}, primenjuju se pravila, u opadajućem poretku težina, na čvor N_{11} koja mu dodeljuju vrstu c_{11}*, pod uslovom vrste c_{12}* čvora N_{12}. Pritom nema drugih testova osim T_1-testova i T_3-testa nad N_{12}. Stoga je, prema algoritmu RAMB, $težina_{i_1 j_1}(uslovi)=1$, pa je

(1) $t_{11} = w_{j1} = \max_{x \in S_{11}} \max_j \{w_j \mid v_j = x$ i vrste jednozn. reči i c_{12} zadovoljavaju testove$\}$,

(2) $t_{12} = w_{j2} * t_{11} = \max_{x \in S_{12}} \max_j \{w_j * t_{11} \mid v_j = x$ i c_{11} zadovoljava test$\}$

Kako postoji optimum, postoje c_{11}', c_{12}', tako da važi:

(3) $težina(N_{11} : c_{11}') = \max_j \{w_j \mid v_j = c_{11}'$ i c_{12}' zadovoljava test$\}$ =

$w_{j1'} = 1$,

(4) $težina(N_{12} : c_{12}') = \max_j \{w_j \mid v_j = c_{12}'$ i c_{11}' zadovoljava test$\}$ =

$w_{j2'} = 1$.

Za $c_{12}* = c_{12}'$, iz (1) i (3) sledi da je $t_{11} >= 1$ (tj. $t_{11} = w_{j1} = 1$), pa iz $w_{j1'}=1$ i neprotivurečnosti pravila r_{j1}, $r_{j1'}$, sledi $c_{11}=c_{11}'$. Takodje, za $c_{11}* = c_{11}'$, iz (2) i (4) sledi da je $t_{12} >= 1$ (tj. $t_{12} = w_{j2} = 1$), pa iz $w_{j2'}=1$ i neprotivurečnosti pravila r_{j2}, $r_{j2'}$, sledi $c_{12}=c_{12}'$.

Dakle, (c_{11}, c_{12}) zadovoljava uslov optimalnosti.

2) Pretpostavimo da vrste $c_{11}, c_{12}, \ldots, c_{1j}$ prvih j čvorova koji napuštaju graf, zadovoljavaju uslov optimalnosti, tj. $c_{11}=c_{11}'$, $c_{12}=c_{12}'$, ..., $c_{1j}=c_{1j}'$. To znači da su, pošto N_{1j}

napusti graf, čvorovi N_{i1}, N_{i2},..., N_{ij} jednoznačno razrešeni (imaju jednoznačno dodeljenu vrstu sa težinom 1).

3) Dokažimo da vrsta c_{ij+1} čvora N_{ij+1} koji j+1-vi napušta graf, zadovoljava uslov optimalnosti. Razlikujemo tri slučaja:

a) Pravilo koje maksimizira težinu vrste c_{ij+1} čvora N_{ij+1} primenjuje samo T_1-testove. U ovom slučaju dokaz ide kao u a) tačke 1).

b) Pravilo koje maksimizira težinu vrste c_{ij+1} čvora N_{ij+1} primenjuje, osim T_1-testova, i T_2-testove sa težinom 1 (nad razrešenim čvorovima iz $\{N_{i1}, N_{i2},..., N_{ij}\}$) i T_3-test nad čvorom N_{ij+2}. U ovom slučaju dokazuje se tvrdjenje za par čvorova (N_{ij+1}, N_{ij+2}) kao u slučaju b) tačke 1).

c) Pravilo koje maksimizira težinu vrste c_{ij+1} čvora N_{ij+1} primenjuje, osim T_1-testova i T_2-testova sa težinom 1, i T_2 ili T_4 - test nad čvorom N_{ij}. Tada je dokaz tvrdjenja o optimalnosti vrste c_{ij+1} čvora N_{ij+1} izveden pri dokazivanju tvrdjenja o optimalnosti vrste c_{ij} čvora N_{ij}.

Time je teorema dokazana.

3.2.3. IMPLEMENTACIJA I EKSPERIMENTALNI REZULTATI

U sistemu za razrešavanje višeznačnosti reči postoje dva skupa pravila: za razrešavanje sintaksne višeznačnosti i za razrešavanje semantičke višeznačnosti.

Prvi skup pravila je u relaciji oblika

br_pravila / reč / uslovi / vrsta.

Svaka četvorka relacije odgovara jednom pravilu. "Reč" je reč na koju se pravilo primenjuje (retko prisutna), "uslovi" su lista uslova koji se testiraju u odgovarajućem kontekstu (antecedens pravila) a "vrsta" je predložena vrsta za reč za čije razrešavanje se pravilo primenjuje (konsekvens pravila). Uslovi su u relaciji

oblika

br_uslova/ kontekst/ fraza/ oblik/ pozicija/ prisutnost.

Interpretacija šestorke ove relacije je sledeća: testira se prisutnost (ili odsutnost) fraze specifičnog oblika na poziciji u kontekstu.

Različite sintaksno višeznačne reči mogu imati različite skupove vrsta. U zavisnosti od pripadnog skupa vrsta, dati kontekst može rezultovati različitim zaključkom i/ili različitom težinom. Na primer, za višeznačnu reč koja može biti imenica ili pridev (primer: "gold" - zlato, zlatan), jedno pravilo može da zaključi da je reč pridev sa težinom 0.9 (od 1.0) ako je propraćena imeničkom frazom; za višeznačnu reč koja može biti pridev, prilog, veza ili predlog (primer: "after" - posle, iza, zatim), drugo pravilo može da zaključi, pod istim uslovom, da je reč predlog sa težinom 0.95; za višeznačnu reč, pak, koja može biti lična ili prisvojna zamenica (primer: "her" - njoj, njen), treće pravilo može da zaključi, pod istim uslovom, da je reč prisvojna zamenica sa težinom 1. Stoga, uz relaciju koja sadrži pravila, ide i skup relacija oblika

br_pravila / težina,

od kojih se svaka odnosi na jedan mogući skup vrsta (npr, na skup {imenica, pridev}, skup {pridev, prilog, veza, predlog}, skup {lična zamenica, prisvojna zamenica}, itd). Svaki par iz jedne relacije dodeljuje težinu zaključku pravila sa odgovarajućim brojem, u slučaju da data reč može imati vrednost iz skupa koji odgovara toj relaciji.

Drugi skup pravila je u relaciji sličnog oblika kao relacija sa sintaksnim pravilima, osim što su težine pridružene pravilima u samoj relaciji. Odgovarajući uslovi su u relaciji koja još sadrži i atribut "svojstvo" koje fraza u kontekstu treba da zadovoljava.

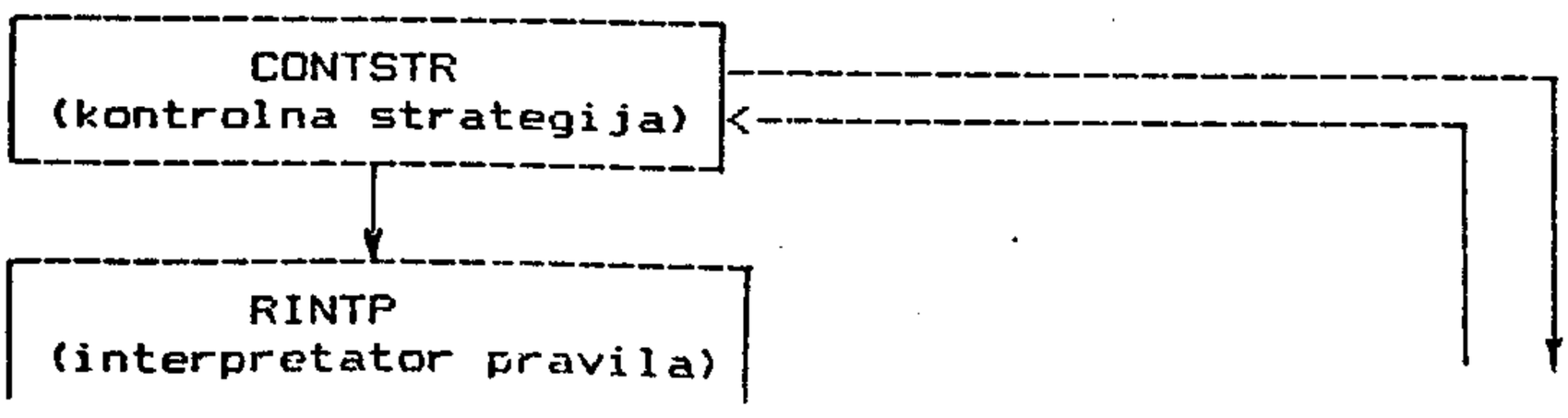
U implementiranom eksperimentalnom sistemu (za engleski jezik) ima oko 100 pravila za razrešavanje sintaksne višeznačnosti i oko 40 pravila za razrešavanje semantičke višeznačnosti. Odgovarajući skupovi uslova broje oko 80 tj. 20 uslova, redom.

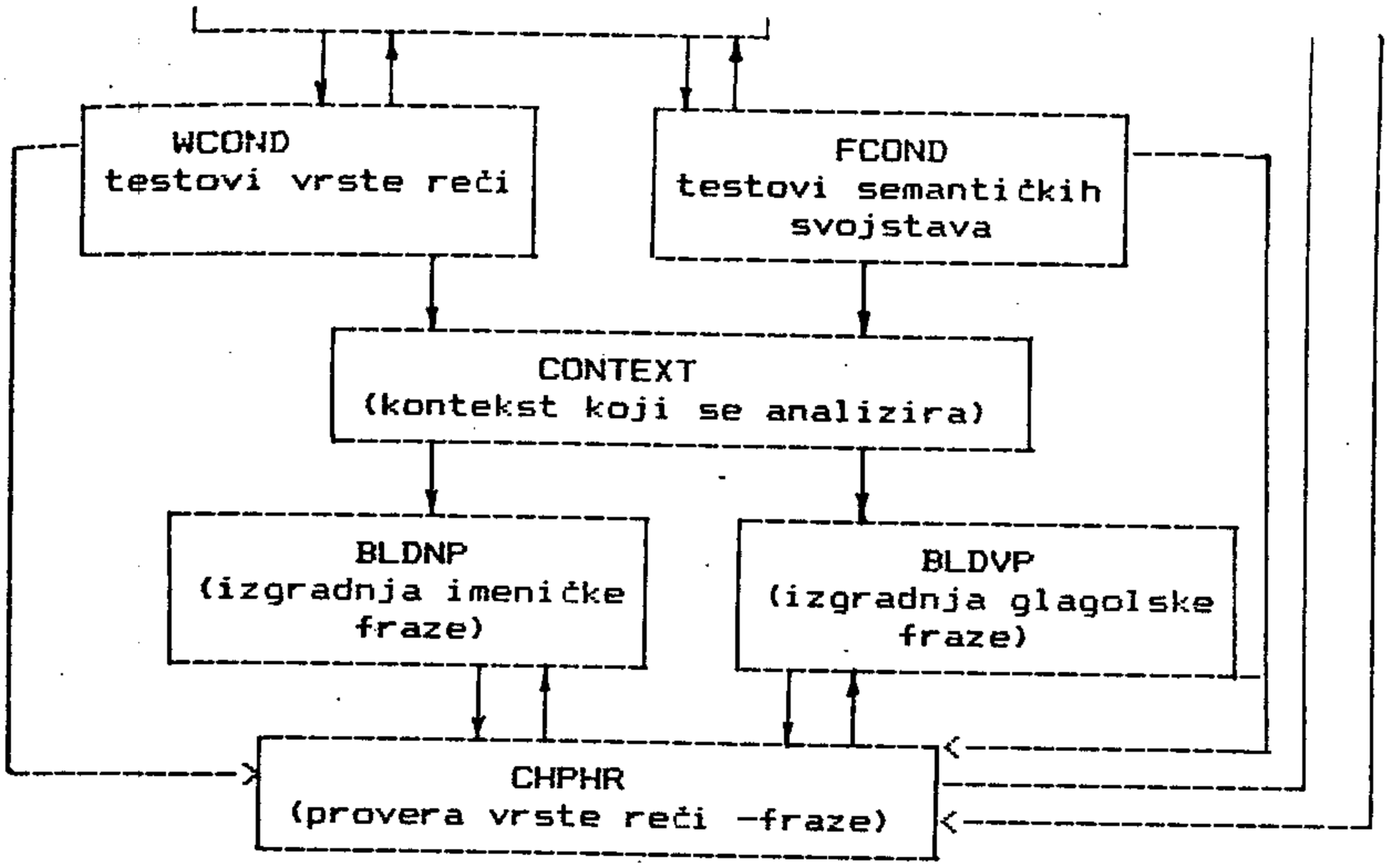
U eksperimentalnom sistemu za srpskohrvatski jezik razmatra se samo sintaksna višeznačnost i baza znanja sadrži oko 70 pravila sa oko 40 uslova.

Proceduralni deo operatora RAMB, implementiran za engleski jezik u EQUEL-u, sastoji se od glavne procedure (rasporedjivača) koja za svaku višeznačnu reč, u zavisnosti od vrste višeznačnosti, primenjuje pravila tj. poziva interpretator pravila. Interpretator se sastoji od procedura za interpretaciju konteksta, interpretaciju pravila za sintaksnu i semantičku višeznačnost, koje, sa svoje strane, pozivaju procedure za testiranje (tj. izgradnju) specifičnih fraza (imeničkih, glagolskih) i vrsta (imenica, prideva, itd). Rasporedjivač (kontrolna strategija) može da se poziva rekurzivno.

U sistemu se koriste i tri table za registrovanje medjurezultata. Prva sadrži informaciju o statusu svih reči iz rečenice koja se predstavlja leksičkim tipom, druga sadrži domen operatora RAMB, tj. kompletne ulaze iz rečnika za sve kandidate višeznačne reči, a treća sadrži informaciju o statusu višeznačnih reči nad kojima se trenutno primenjuje operator RAMB (reprezentacija grafa višeznačnosti).

Grafik toka informacije kroz sistem kojim se implementira operator RAMB je sledeći:





Delovi relacija koje sadrže bazu znanja operatora RAMB (pravila i uslove) nalaze se u dodatku 3.

Opisani eksperimentalni sistem (kako proceduralni deo tako i baza znanja za engleski jezik) izgradjen je na osnovu potreba za automatskom "leksikalizacijom" (predstavljanjem leksičkim tipom) teksta biografije Alberta Ajnštajna iz Enciklopedije Britanike. Zatim je eksperiment ponovljen, bez modifikacije sistema, na druga dva teksta tipa biografije, i dobijeni su sledeći rezultat: u prvom tekstu dužine 4100 reči, od oko 250 višeznačnih reči, 86% razrešeno je korektno. Za drugi tekst dužine 1200 reči, procenat uspeha je 81% , a u trećem tekstu dužine 2700 reči, taj procenat je 75%.

Eksperiment nad srpskohrvatskim tekstom dužine 1500 reči dao je uspešni rezultat nad 95% višeznačnih reči, na osnovu pravila koja sadrže zdravorazumske heuristike. Najčešći primeri sintaksne višeznačnosti su oblici iste reči, npr. "iz oblasti" (imenica ženskog roda u 2. padežu jednine, 3. padežu jednine, 2. padežu množine ili 4. padežu množine), mada su česti i primeri raznih

(mogućih) vrsta reći, npr. imenica/glagol: "prolazi", "primeni", "prekida", itd. U prvom slučaju često pomažu predlozi, a u opštem slučaju širi kontekst. Primer neuspelog razrešavanja u srpskohrvatskom jeziku je izraz "Radi ostvarivanja ove i drugih svojih zamisli...", gde zamenica "ove" može biti u 2. padežu jednine ženskog roda, 1. padežu množine ženskog roda, 4. padežu množine ženskog roda i 2. padežu množine srednjeg roda. S obzirom na predlog "radi" (ova je reć takodje bila višeznačna), dolaze u obzir samo 2. padeži množine, i tu se mogućnosti analize izražene pravilima iscrpljuju. U slučaju da za zamenicom "ove" sledi neposredno reć "zamisli", primenom postojećih pravila operator RAMB dao bi rezultat. Takodje se u izrazu "u laboratorijama fakulteta" ne može odrediti da li se "fakulteta" odnosi na jedninu ili množinu, a u slučaju "a onda prelazi" - da li je "prelazi" imenica ili glagol.

Izvor grešaka operatora RAMB je uglavnom u ograničenoj i vrlo jednostavnoj analizi konteksta, kao i u činjenici da pravila nije formulisao lingvistički ekspert. Takodje je, za razrešavanje semanatičkih višeznačnosti, potrebno pojačati sistem konteksno-zavisnom semantičkom komponentom.

Primer teksta na engleskom jeziku i njegove reprezentacije leksičkim tipom podataka nalazi se u dodatku 3.

Pregled postojećih metoda za rešavanje problema leksičke višeznačnosti (višeznačnosti samih reći) izložen je u radu Birnbauma [4]. Problem se razlaže na dva medjusobno zavisna podproblema: razrešavanje sintaksne i razrešavanje semantičke višeznačnosti. Najčešći pristupi rešavanju prvog (koji se naziva i višeznačnost dela govora - "part_of_speech") uključuju metode ATN sintaksnih analizatora [76], modele koji su proizašli iz njih, kao i metod ograničenog "posmatranja unapred" ("look ahead"), uz

primenu, takodje ograničene količine, semantičke informacije (ovaj drugi pristup je u osnovi primenjen u ovoj tezi). Pristupi razrešavanju semantičke višeznačnosti (rešavanje drugog podproblema) svrstavaju se globalno u dve grupe: oni koji primenjuju "selekciona ograničenja" ("selectional restrictions"), i oni koji primenjuju skriptalne leksikone ("scriptal lexicons"). Selekciona ograničenja su semantički uslovi koje treba da zadovoljavaju dva jezička izraza da bi mogla da stoje jedan uz drugi (npr. radnja jedenja zahteva da akter bude živo biće, pa se tako za objekat koji može imati semantičko svojstvo "živo biće" i "predmet", može zaključiti da u konkretnom pojavljivanju ima semantičko svojstvo "živo biće" ako je objekat akter radnje jedenja). Skriptalni leksikon je rečnik pridružen specifičnom kontekstu (oblasti), tako da u opštem slučaju semantički višeznačne reči postaju semantički jednoznačne u tom kontekstu. S obzirom na ograničenja koja postoje u oba pomenuta pristupa, Birnbaum predlaže koncept integralnog pristupa jezičkoj analizi, u kom značajnu ulogu imaju memorija (u kojoj je reprezentacija celokupnog znanja iz teksta) i metodi izvodjenja. Sugerišu se primena pravila izvodjenja i opšteg mehanizma izvodjenja i, u tom kontekstu, mogućnost adekvatne primene parametara kao što su težina pravila i nivoi aktiviranja pravila. Pristup semantičkoj analizi u ovoj tezi spada u prvu grupu - semantička svojstva su vid selekcionih ograničenja, a koncept ekspertnog sistema primenjen na rešavanje obuhvatnog problema leksičke višeznačnosti uključuje sve elemente pomenute sugestije (pravila, mehanizam izvodjenja i primena težina kao aktivacionog mehanizma za pravila).

3.3. ODREDJIVANJE REFERENATA ZAMENICA - OPERATOR PRONR

Drugi složeni operator izgradjen na bazi leksičkih operatora je PRONR-operator koji zamenicama pridružuje njihove "originale" (fraze koje zamenjuju). Ovaj operator je važan u većini primena koje, osim forme, uključuju i sadržaj teksta. Moj cilj je ograničen na lične, prisvojne (3. lice), pokazne i odnosne zamenice. Pristup problemu je opet zasnovan na pravilima. Ekspertni sistem konstruisan za implementaciju ovog operatora se, kao i u slučaju ekspertnog sistema za implementaciju operatora RAMB, sastoji od baze znanja i mehanizma izvodjenja. Ove komponente će biti opisane kasnije.

Pre nego što se definišu baza pravila i procedura izvodjenja, uočavaju se neke činjenice o odnosu zamenica i njihovih originala.

(a) zamenica i imenica u originalu moraju da pripadaju istoj leksičkoj klasi ekvivalencije; leksička klasa ekvivalencije definiše se relacijom ekvivalencije "slaganje" definisanom nad leksičkim jedinicama na sledeći način:

slaganje(l_1, l_2) <=>

{vrsta_reči(l_1), vrsta_reči(l_2) ∈ {imenica, zamenica};

broj(l_1) = broj(l_2);

rod(l_1) = rod(l_2);

svojstvo(l_1) = svojstvo(l_2);}

Na primer, zamenica "on" slaže se sa imeničkom frazom koja sadrži imenicu sa karakteristikama "muški rod, jednina, HUMAN", npr. "čoveku". Ako je (id, deskr) leksička jedinica za takvu imenicu, onda treba da je

vrsta_reči(id, deskr) = imenica;

rod(id, deskr) = muški;

broj(id, deskr) = jednina;

svojstvo(id, deskr) = HUMAN.

(b) zamenica može stajati bilo ispred bilo iza imeničke fraze - originala;

(c) zamenica se retko pojavljuje ispred svog originala (u mom eksperimentu to se dogodilo u 3% slučajeva);

(d) ako se zamenica pojavljuje ispred originala, onda su obe u istoj rečenici ali u raznim podrečenicama (razdeljenim zarezom);

(e) ako zamenica sledi za originalom, verovatno je da je original najbliža toj zamenici imenička fraza koja joj (zamenici) prethodi, i koja je u istoj leksičkoj klasi ekvivalencije kao zamenica (u mom eksperimentu to se dogodilo u 97% slučajeva).

Neke od ovih činjenica izražavaju se pravilima u bazi znanja (npr. (a), (b), (d)), a o drugima vodi računa procedura izvodjenja (npr. (c), (e)).Kao i kod operatora RAMB, pravila su oblika

(antecedens, konsekvens, težina).

Rečima, pravila kažu:

- ako se neki <u>objekat</u> sa <u>specifičnim svojstvom</u> i u istoj leksičkoj klasi ekvivalencije kao zamenica o kojoj je reč, pojavljuje u specifičnom <u>kontekstu</u>, (antecedens)
- onda je taj objekat original odgovarajuće zamenice (konsekvens)
- sa izgledima X (težina).

<u>Objekat</u> može biti ili imenička fraza ili druga zamenica; <u>kontekst</u> je hijerarhijski definisan u terminima rečenica, podrečenica, imeničkih fraza i specifičnih vrsta reči, sa eksplicitnom orjentacijom (levo, desno od zamenice), i sa zahtevom da je bliži zamenici, u toj orjentaciji, nego prethodni objekat dodeljen zamenici da se na njega <u>odnosi</u> (najširi kontekst koji se razmatra je tekuća i prethodna rečenica); <u>svojstvo</u> su specifični uslovi koje kontekst i/ili objekat moraju da zadovoljavaju (npr. imenička fraza koja sadrži imenicu - original ne treba da sadrži

zamenicu iz iste leksičke klase ekvivalencije kojoj pripada i zamenica kojoj se odredjuje original, kao u frazi "When he graduated, professor X, his father..." – "Kada je on diplomirao, profesor X, njegov otac...", gde se "he" – "on" ne odnosi na "professor X" – "profesor X").

Iz oblika pravila sledi da "zamenica se odnosi na objekat", kako je precizirano pravilom, ima značenje "levo-odnosi" ili "desno-odnosi". Štaviše, prema pravilu, zamenica može da se odnosi na drugu zamenicu a ne na original, pa stoga svako pravilo definiše dvočlanu relaciju "neposredno (se) levo-odnosi" ili "neposredno (se) desno-odnosi" koja se može razlikovati od relacije "odnosi se na original". Situacija se može grafički predstaviti na sledeći način:

* – zamenica kojoj se odredjuje original;

\+ – bilo koja druga zamenica iz iste leksičke klase ekvivalencije;

** – imenička fraza – original;

--> – (zamenica se) neposredno desno-odnosi (na objekat);

<-- – (zamenica se) neposredno levo-odnosi (na objekat);

/-/ – (zamenica se) samoodnosi (odnosi se na samu sebe);

```
      /-/              /-/   /-/   /-/             /-/
**<-- + <-- ... <-- + <-- * --> + --> ... --> + --> **.
```

Pošto je od interesa odredjivanje imeničke fraze – originala zamenice, tj. relacija "odnosi se (na original)", ta relacija se definiše aksiomatski na sledeći način:

Definicija 3:

(i) relacija "samoodnosi (se)" je relacija "desno (se) odnosi";

(ii) relacija "neposredno (se) desno-odnosi" je relacija "desno (se)-odnosi";

(iii) relacija "desno (se)-odnosi" je tranzitivna;

(iv) relacija "desno (se) odnosi" čiji je jedan element original je relacija "završno desno (se) odnosi";

(i') relacija "samoodnosi (se)" je relacija "levo (se) odnosi";

(ii') relacija "neposredno (se) levo-odnosi" je relacija "levo (se)-odnosi);

(iii') relacija "levo (se)-odnosi" je tranzitivna;

(iv') relacija "levo (se)-odnosi" čiji je jedan element original je relacija "završno levo (se) odnosi";

(I) relacija "završno levo (se) odnosi" je relacija "odnosi se" (jedan element joj je original);

(II) relacija "završno desno (se) odnosi" je relacija "odnosi se" (jedan element joj je original).

Sledeće teoreme dokazaće da relacija "desno (se)-odnosi" tj. "levo (se)-odnosi" definiše jedinstveni desni tj. levi lanac sa najviše jednim originalom u svakom (na kraju lanca). Drugim rečima, relacija "završno (se) odnosi (levo tj. desno)" ima najviše jedan element po lancu, za zamenicu o kojoj je reč.

Teorema 3: Relacija "desno (se)-odnosi" (tj. "levo (se)-odnosi") je relacija parcijalnog uredjenja.

Dokaz: Aksiome (i), (iii), obezbedjuju refleksivnost i tranzitivnost relacije "desno (se)-odnosi". Antisimetričnost ove relacije sledi iz činjenice da pravila definišu relaciju "neposredno (se) desno-odnosi" kao "strogo sledi u poretku reči". Tako, "A se neposredno desno-odnosi na B" (i stoga "A se desno-odnosi na B") implicira da B sledi za A. Ako "A se desno-odnosi na B" i "B se desno-odnosi na A", onda B sledi za A i A sledi za B u poretku reči, pa su obe relacije - relacije "samoodnosi (se)" (A=B). Dakle, relacija "desno (se)-odnosi" je relacija uredjenja. Analogno za relaciju "levo (se)-odnosi".

Teorema 4: Relacija "neposredno (se) desno-odnosi" (tj. "neposredno (se) levo-odnosi") je jedinstvena (tj. za zamenicu o kojoj je reč sadrži najviše jedan element).

Dokaz: Neka je P zamenica o kojoj je reč (kojoj se odredjuje original) i, za neki objekat O, P-->O (tj. (P,O)) neka je element relacije "neposredno (se) desno-odnosi". Ako je objekat O jedinstven, onda je odgovarajuća relacija "neposredno (se) desno-odnosi" jedinstvena. Iz načina na koji su pravila i procedura izvodjenja konstruisani, sledi da na svakom nivou težine $\underline{i}$ koji je prisutan u skupu pravila, podskup pravila koji odgovara težini w_i izabraće (jedinstvenu) frazu O_i najbliżu zamenici P zdesna, iz iste leksičke klase ekvivalencije kao P. Od svih objekat O_i, za objekat O biće izabran onaj koji je nadjen na najvišem nivou težine w_i. Dakle, relacija "neposredno (se) desno-odnosi" je jedinstvena. Analogno za relaciju "neposredno (se) levo-odnosi".

Posledica: Relacija "desno (se)-odnosi" ("levo (se)-odnosi") definiše jedinstveni lanac sa najviše jednim originalom.

Procedura izvodjenja funkcioniše na sledeći način: za svaku zamenicu u rečenici, primenjuje sva pravila (u opadajućem poretku težina) primenjiva na tu zamenicu, a čije su težine veće od težine sa kojom se zamenica već "neposredno levo-odnosi" ("neposredno desno-odnosi") na neki objekat. Pošto se i poslednjoj zamenici u rečenici odredi objekat na koji se neposredno desno odnosi, i/ili objekat na koji se neposredno levo odnosi, konačni original za svaku zamenicu nalazi se (ako je moguće) ispitivanjem odgovarajućih lanaca i njihovih "završnih referenata" (objekata koji učestvuju u odgovarajućim relacijama "odnosi se". Ovu funkciju obavlja sledeće pravilo diskriminacije:

if lanac relacije "levo (se) odnosi" završava se u originalu

(imeničkoj frazi a ne zamenici)

then zamenica se odnosi na taj original;

else if lanac relacije "desno (se) odnosi" završava se u originalu

then zamenica se odnosi na taj original;

else zamenica se ne odnosi ni na koji original.

Ovo pravilo favorizuje "leve" originale (olančavanje unazad), što je u saglasnosti sa svojstvom (c) navedenim na početku ovog dela. Rezultat ekspertnog sistema je invertovani indeks zamenica i njihovih originala (tabela odnosa). Svaki ulaz u indeks sadrži par (leksička jedinica koja predstavlja original, pozicija zamenice (u tekstu) koja se odnosi na taj original). Skup svih zamenica koje se odnose na jedan original u tabeli odnosa, zajedno sa originalom, čini semantičku klasu ekvivalencije. Semantičke klase ekvivalencije su podskupovi leksičkih klasa ekvivalencije kojima pripadaju odgovarajući originali.

3.3.1. IMPLEMENTACIJA I EKSPERIMENTALNI REZULTATI

U bazi znanja operatora PRONR postoje tri osnovne relacije. Jedna se sastoji od pravila kako odrediti fraze - kandidate za original zadate zamenice. Ta relacija je oblika broj/kontekst_uslovi/pozicija/objekat/obj_uslovi/tip/smer/težina a svaka n-torka relacije ima sledeću interpretaciju: pravilo broj dodeljuje objekat (imenicu, drugu zamenicu, vlastito ime) razmatranoj zamenici kao kandidat za original, sa težinom težina, u smeru smer (levo, desno), ako su ispunjeni sledeći uslovi: zamenica je vrste tip, objekat je na poziciji pozicija u kontekstu definisanom hijerarhijom uslova kontekst uslovi, objekat je iz iste leksičke klase ekvivalencije kao zamenica, objekat zadovoljava uslove obj uslovi i objekat je bliži zamenici koja se razmatra, s te strane, nego prethodno dodeljeni kandidat.

Relacija koja sadrži kontekst-uslove je oblika

broj/objekat/redni_broj_pojavljivanja

a trojka relacije se interpretira na sledeći način: uslov broj broj je odredjivanje (ako postoji) objekta tipa objekat (rečenica, imenička fraza, itd.) koji se pojavljuje redni broj pojavljivanja - put, brojeći od posmatrane zamenice. Npr. (2, rečenica, -1) znači: kontekstni uslov 2 je prethodna rečenica (ako postoji).

Relacija koja sadrži objekat-uslove je oblika

identifikator/opis/prisutnost

gde je identifikator (A-Z) ime uslova, opis je opis odgovarajućeg uslova, a prisutnost govori da li se testira prisustvo ili odsustvo uslova. Primer objekat-uslova je (F, relcl, -), gde "relcl" označava relativnu rečenicu ("relative clause"), a uslov testira nepripadanje objekta na koji se primenjuje ovaj objekat-uslov relativnoj rečenici. Interpretacija objekat-uslova definisana je interpretator-procedurama.

Proceduralni deo operatora PRONR, napisan u EQUEL-jeziku, sastoji se od glavnog programa koji, za svaku zamenicu u tekućoj rečenici, poziva proceduru za kontrolu upravljanja, a ova, pozivom procedure za interpretaciju pravila i testiranje uslova, odredjuje najbliži levi i desni kandidat za original. Glavna procedura zatim ispituje levi i desni lanac za svaku zamenicu, i proglašava za original imeničku fraza na kraju levog (tj. desnog) lanca. Na kraju se original sa pozicijom odgovarajuće zamenice dodaje invertovanom indeksu. Ceo program za odredjivanje referenata zamenica ima oko 3000 linija.

Relacije sa pravilima, kontekst-uslovima, objekat-uslovima za engleski i srpskohrvatski jezik, i primer teksta sa razrešenim zamenicama i invertovanim indeksom, nalaze se u dodatku 4.

Operator PRONR primenjen je na ista četiri teksta kao i operator RAMB. Procenat uspešno odredjenih originala zamenica u

ta četiri teksta je, redom, 84%, 91%, 84%, 88%. Većina pogrešno odredjenih originala je za neutralne zamenice, kao što je "it", u frazi : "...new theory and acclaim for its creator..." ("nova teorija i priznanje za njenog tvorca") ili "It was then only the influence..." (To je bio samo uticaj..."). Novi problem koji se otvara u slučaju srpskohrvatskog jezika i koji se ovde ne rešava je slučaj izostavljene zamenice (npr. "radi" umesto "on radi").

Glavni nedostatak "ispravnih" originala uočen je u slučaju kada, osim vlastitog imena (kao "Ajnštajn"), koje odgovara zamenici (kao "on"), postoji i imenička fraza (kao "vrlo zadovoljan čovek") koja takodje odgovara zamenici i bliža je zamenici nego vlastito ime.

Osim radova na temu odredjivanja referenata zamenica, pomenutih u uvodnoj glavi, izuzetno informativan sa lingvističkog aspekta je rad C. Sidner [59]. Osim što daje pregled postojećih istraživačkih pravaca (opštih heuristika, sintaksnih i semantičkih ograničenja kojima se eliminišu neki od mogućih referenata, metode izvodjenja iz reprezentacije znanja, analize odnosa objekata u kontekstu), u ovom radu se definišu lingvistički zasnovana pravila (o fokusu i potencijalnim fokusima i jezičkoj ulozi zamenice u rečenici), koja se, uz adekvatnu lingvističku interpretaciju, mogu uključiti u bazu znanja ekspertnog sistema za implementaciju operatora PRONR, tipa onog u ovoj tezi.

Problem izbora originala zamenice, od većeg broja kandidata, pod imenom "referencijalnog konflikta", razmatra se u širem kontekstu modeliranja jezičke delatnosti u [33]. Predlažu se i neka sredstva za otklanjanje ovog konflikta, koja predstavljaju formalne i semantičke karakteristike najbližeg konteksta ("unutarpozicioni faktori"), i koja bi se takodje mogla ugraditi u bazu znanja postojećeg operatora PRONR. Primer jednog takvog

faktora je nemogućnost kandidovanja jedne anafore jednog glagola za original druge anafore istog glagola (zamenice) - npr. za glagol "dati" anafora "kome" ne može biti original zamenice - anafore "ko" ili "šta". Primer drugog faktora je neslaganje kandidata za original sa postojećim znanjem o njemu (npr. semantičkim svojstvom). Npr. u rečenicama "Petar i Marko će posetiti Beograd i Zagreb. Udaljenost medju njima je 400km.", "njima" bi se odnosilo na "Beograd i Zagreb" a ne na "Petar i Marko", jer se udaljenost meri medju objektima sa semantičkim svojstvom LOC a ne HUM.

4. PRIMENE TEKSTUELNIH BAZA PODATAKA

Pod tekstuelnom bazom podataka ovde se podrazumeva tekst smešten u relacionoj bazi podataka i predstavljen leksičkim tipom podataka. Ovaj pristup organizaciji teksta omogućuje jednostavno i precizno izvodjenje niza operacija nad tekstom kao nizom leksičkih podataka. Jedna vrsta ovih operacija kao operatora sledećeg (trećeg) nivoa nad leksičkim tipom podataka, su klasične operacije kao što su automatsko indeksiranje, odredjivanje ključnih reči i fraza, apstraktiranje, pretraživanje, editovanje (eksperimenti sa ovim operacijama koje su sve iz oblasti pretraživanja informacija, opisani su u sledećoj glavi).

Druga vrsta operatora ovog nivoa su operacije koje ne pripadaju oblasti pretraživanja informacija (npr. testiranje stilske homogenosti) kao i visoko-semantičke operacije zasnovane na razumevanju teksta, kao što je izdvajanje precizne informacije iz teksta. U ovoj glavi detaljnije će biti prikazan koncept i eksperimentalni rezultati operatora izdvajanja informacija iz teksta, uz primenu osnovnog relacionog modela i modela sa null-vrednostima.

4.1. IZDVAJANJE INFORMACIJE SADRŽANE U TEKSTUELNOJ BAZI PODATAKA - RELACIONI MODEL BAZE ZNANJA

Izdvajanje precizne informacije iz teksta sastoji se iz postavljanja pitanja o činjenici iz teksta (npr. kada je rodjena osoba po imenu "X") i nalaženja odgovora (npr. 1879.). Ova aktivnost može se predstaviti operatorom nad parom skupova leksičkog tipa kao domenom - (tekst, skup upita), i skupom leksičkog tipa kao kodomenom (skup fakata iz teksta).

Problem se najčešće rešava u okviru šireg problema

razumevanja povezanog teksta. Pritom se prvo tekst analizira sa sintaksne i semantičke osnove i znanje sadržano u tekstu (činjenice, njihove veze), predstavlja nekom od pogodnih shema za reprezantaciju znanja (razni modeli memorije [4, 9, 36], semantičke mreže - proširene, kvantifikovane [9, 15, 22, 57, 60, 61, 62], teorija reprezentacije konteksta [18]). Kroz istu proceduru prolazi i upit, ako je na prirodnom jeziku, (ako nije - formalizovani upit već ima zadatu reprezentaciju), a zatim se dve reprezentacije (teksta i upita) sravnjuju i nalazi odgovor na postavljeno pitanje.

Moj pristup izdvajanju činjenica iz teksta polazi od samog problema (ne od problema razumevanja teksta uopšte), a za shemu reprezentacije ograničenog znanja iz teksta (znanja na nivou rečenice) uzima se relacioni model tj. relaciona shema. Pristup se zasniva na vidjenju teksta kao virtuelne relacione baze podataka koja odgovara zadatoj shemi. Shema (relacije, atributi, domeni), predstavlja korisnički pogled na tekst jer odražava aspekte teksta za koje je korisnik zainteresovan (npr. iz skupa biografija korisnik je zainteresovan za datume, mesta rodjenja, školovanje, zaposlenje i institucije školovanja i zaposlenja - i nizašta drugo). Stoga shema koja odražava aspekt familije tekstova za koje je korisnik zainteresovan, sadrži odgovarajuće atribute grupisane u relacije, i ni jedne druge. Shema takodje definiše, à priori, skup svih upita koji se mogu postaviti. Odgovor na upit nalazi se iz jednog ili više tekstova u vreme izvršavanja upita. Jedino preprocesiranje teksta je njegovo predstavljanje leksičkim tipom koje se vrši u vreme unosa teksta u relacionu bazu podataka i izvršavanje operatora PRONR. Preslikavanje para (tekst, upit) u odgovor na upit intenzivno koristi sintaksna i semantička svojstva reči ugradjena u leksičke podatke koji te reči predstavljaju.

U vreme definisanja relacione sheme kao pogleda na tekst, odgovarajuća relaciona baza podataka je čisto virtuelna – ne sadrži nikakve podatke (podaci su sadržani samo u leksičkoj reprezentaciji teksta). Nalaženjem odgovora na pojedine upite nad virtuelnom bazom podataka, ti odgovori se, kao podaci, unose u kreiranu bazu podataka, čime ona postaje hibridna – stvarno/virtuelna (sadrži neke podatke, a za one koje ne sadrži, predstavlja pogled na odgovarajuće podatke u tekstu). Entitet tj. odnos predstavljen jednom relacijom u relacionoj shemi odgovara rečenici iz teksta, tj. njenom delu koji je relevantan za atribute tog entiteta (odnosa). Stoga je ovako projektovana relaciona shema – reprezentacija nepovezanog teksta (teksta na nivou rečenice). Kako se kao pojedina n-torka relacije posmatra relevantni (za atribute n-torke) sadržaj pojedine rečenice, a pojedina rečenica može sadržati podatke koji su relevantni za neke (ali ne sve) atribute n-torke, relacioni model koji se koristi za predstavljanje znanja iz teksta uključuje null-vrednosti (vrednosti svih atributa n-torke predstavljene rečenicom koja ne sadrži podatke relevantne za te atribute). Primenom "faktuelnog" izvodjenja, tj. operacija projektovanja i spajanja koje ne gube informaciju (koje su bazirane na funkcionalnim i višeznačnim zavisnostima medju atributima relacione sheme), relacioni model postaje reprezentacija povezanog znanja iz teksta: iz njega se mogu dovesti u vezu vrednosti atributa (podaci) koje nisu sadržane u jednoj rečenici.

Upiti koji se mogu postaviti nad tekstom su upiti nad (u početku fiktivnom) relacionom bazom podataka. Pretpostavlja se A) da se jedan upit odnosi na jedan entitet/odnos, tj. da sadrži jednu domensku promenljivu (inače se može dekomponovati u takav upit), B) da su upiti formulisani u bilo kom jeziku (nadalje će

biti pretpostavljeno da su upiti zadati u upitnom jeziku, npr. QUEL-u).

Glavne komponente sistema za izdvajanje informacije iz teksta su definisanje fiktivne relacione baze podataka (entiteta, odnosa, relacija, domena, atributa), i definisanje operatora izdvajanja informacije (preslikavanja upita nad tom relacionom bazom u mehanizme pretraživanja teksta i algoritma pretraživanja tj. nalaženja odgovora na upit).

Da bi se realizovala prva komponenta operatora izdvajanja informacije (preslikavanje upita u mehanizme pretraživanja), relaciona shema se snabdeva izvesnom jezički-orjentisanom semantikom, tj. definišu se preslikavanja skupova, relacija, domena, atributa, konstanti u jezičke kategorije teksta (leksičke jedinice ili skupove leksičkih jedinica) sa specifičnim vrednostima leksičkih operatora - specifičnom vrstom, specifičnim korenom, semantičkim svojstvom i/ili specifičnim predlogom ispred/iza, itd:

$$
\begin{array}{lll}
\{\text{relacije-entiteti}\} & --f_1--> & \left\{\begin{array}{l}(\ldots,\text{leks}[\text{poz}],\ldots):\\ \text{koren}(\text{id}[\text{poz}]) = \text{konst}_1 \text{ i/ili}\\ \text{vrsta}(\text{deskr}[\text{poz}]) = \text{konst}_2 \text{ i/ili}\\ \text{svojstvo}(\text{deskr}[\text{poz}])=\text{konst}_3 \text{ i/ili}\\ \text{vrsta}(\text{deskr}[\text{poz}-1])=\text{predlog i/ili}\\ \text{vrsta}(\text{deskr}[\text{poz}+1])=\text{predlog i/ili}\\ \qquad(\ldots)\end{array}\right.\\
\{\text{relacije-odnosi}\} & --f_1--> & \\
\{\text{domeni}\} & --f_2--> & \\
\{\text{atributi}\} & --f_3--> &
\end{array}
$$

$$\{\text{konstante}\} \quad --f_4(=I)--> \{\text{konstante}\}.$$

Jednorelacioni upit nad relacijom R relacione baze podataka je oblika

$$\text{upit}^* \begin{cases}\text{range of e is R}\\ \text{retrieve (e.}\mathbb{C}\text{) where e.}\mathbb{A} = a\end{cases}$$

($\mathbb{C} = \{C_1, C_2, \ldots, C_k\}$, $\mathbb{A} = \{A_1, A_2, \ldots, A_m\}$ - skupovi atributa

relacije R, $\mathbb{a}$ = ($a_1, a_2, \ldots, a_m$) - m-torka konstanti, a oznaka e.$\mathbb{C}$ znači e.C_1, e.C_2, ..., e.C_k, tj. e.$\mathbb{A}$ = $\mathbb{a}$ označava e.A_1=a_1, e.A_2=a_2, ..., e.A_m=a_m).

Upit* karakteriše se relacijom R, skupovima atributa $\mathbb{A}$, $\mathbb{C}$ i konstantom $\mathbb{a}$ (operacija je uvek ista - restrikcija i projekcija, s obzirom na pretpostavku A)), tj. upit*(R,$\mathbb{A}$,$\mathbb{a}$,$\mathbb{C}$)[2].

Preslikavanje upita* u mehanizme pretraživanja po tekstu (upit nad tekstom, upitt*) je sledećeg oblika:

upit* ----f=f_1xf_2xf_3xf_4----> upitt* (= f(upit*)) (1),

upitt* = (naći sve skupove leksičkih jedinica koji zadovoljavaju f_3($\mathbb{C}$), f_2(domen($\mathbb{C}$)) i koji su u rečenici koja sadrži f_1(R) i za svaki atribut A' iz $\mathbb{A}$, konstantu f_4(a' iz $\mathbb{a}$) ili f_3(A') ili f_2(domen(A')).

U vezi sa drugom komponentom operatora izdvajanja informacije, algoritmom pretraživanja tj. nalaženjem odgovora na upit, uočimo sledeće: u slučaju stvarne relacije R, upitu* odgovara skup vrednosti atributa $\mathbb{C}$ relacije R, definisan preslikavanjem odg:

upit* --odg--> odg(upit*) (2).

Ali, kako je relacija R fiktivna i predstavlja pogled na tekst, pod odg(upit*) podrazumevaće se skup svih odgovora na upit* prisutnih u tekstu (bez obzira na način kako do njih doći).

S druge strane, upitt* nad tekstom (=f(upit(*)), ima efektivno izračunljiv skup odgovora, definisan preslikavanjem

---

[2] U opštem slučaju upit koji ima više domenskih promenljivih karakteriše se operacijama relacione algebre i atributima i konstantama nad kojima operišu. Takav upit ima oblik: upit(R, (operacija, atributi, konstante, R1)[n]), gde je "operacija" {restrikcija, projekcija, spajanje}, atributi i konstante su objekti nad kojima "operacija" deluje, Ri je dodatna relacija u slučaju višedomenskih promenljivih. Upit* je specijalni slučaj oblika upit*(R, restr, $\mathbb{A}$, $\mathbb{a}$, proj, $\mathbb{C}$).

odgt:

upitt* ----odgt----> odgt(upitt*) (3) (ovo preslikavanje bitno koristi operator PRONR za identifikaciju parova (zamenica, original)).

U idealnom slučaju (preslikavanje f idealno preslikava elemente relacione sheme u elemente jezika i svi odgovori na upit nalaze se sa podacima upita u po jednoj rečenici), bilo bi odg = odgt * f. Nešto slabiji, mada još uvek idealan slučaj bio bi odg' = odgt * f, gde je odg'(upit*) - skup odgovora na upit* koji su eksplicitno prisutni, u istoj rečenici, sa podacima upita ($\mathcal{A}$, $\mathcal{a}$). Poželjno je obezbediti bar inkluziju odgt * f $\subseteq$ odg', s obzirom da preslikavanje odg' daje sve odgovore polaznog upita (ne samo efektivno izračunljive), sadržane u istoj rečenici sa podacima. Dva svojstva preslikavanja f_1 - f_4 opisuju stepen ostvarenosti ove inkluzije, odnosno obrnute inkluzije (odg' $\subseteq$ odgt * f):

- razdvojna moć (preciznost) preslikavanja f_1 - f_4 - mera s kojom termini i svojstva termina - slika relacija, atributa, domena, odgovaraju relacijama, atributima, domenima, i izdvajaju se od termina u tekstu koji to nisu. Intuitivno, ako je preciznost od f_1-f_4 jednaka 100%, onda je odgt*f$\subseteq$odg'. Jedan potreban uslov da je preciznost = 100% je da se dva razna atributa preslikavaju u različite skupove leksičkih jedinica.

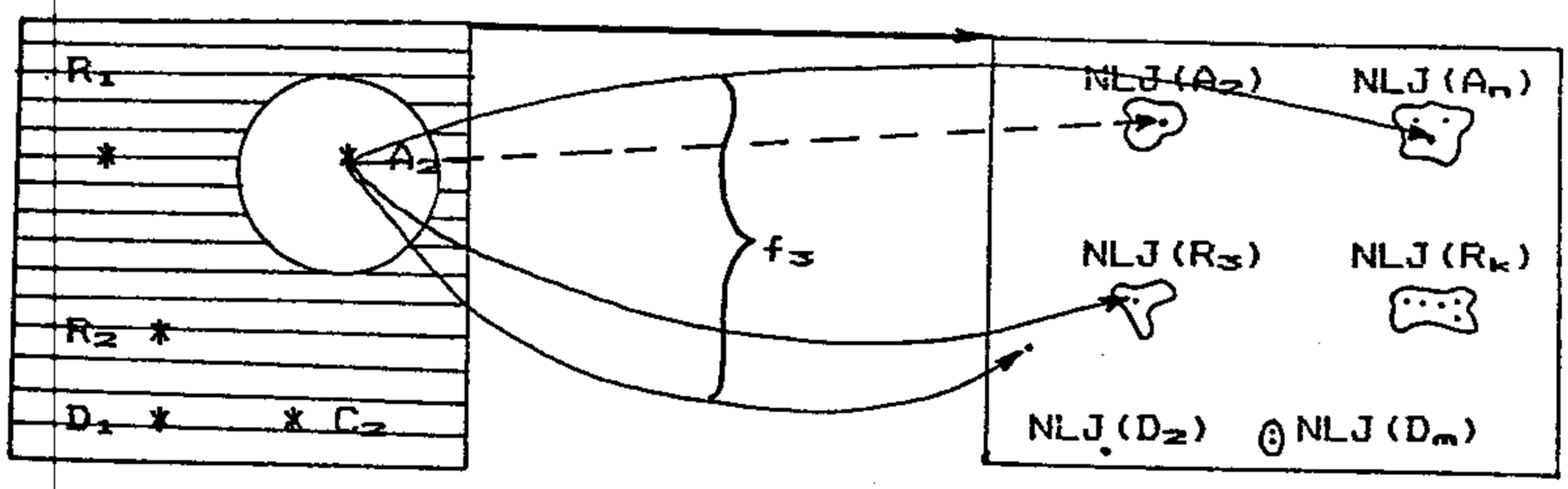
- potpunost preslikavanja f_1 - f_4 - mera s kojom termini i svojstva termina - slika relacija, domena i atributa, "pokrivaju" termine koji u tekstu odgovaraju semantički tom atributu. Intuitivno, ako je potpunost =100%, odg' $\subseteq$ odgt * f.

U realnosti, treba maksimizirati verovatnoće p=P(odgt*f $\subseteq$odg') i p'=P(odg' $\subseteq$ odgt*f) (tj. preciznost i potpunost preslikavanja odgt*f). Pritom, $p=p_{*}*p_{odgt}$, $p'=p'_{*}*p'_{odgt}$, gde su p_{*}, p_{odgt}, p'_{*}, p'_{odgt} - preciznosti i potpunosti preslikavanja f,

odgt, redom (verovatnoća sa kojom nizovi leksičkih jedinica koji odgovaraju upitu - sadrže odgovor, verovatnoća sa kojom je nadjeni odgovor iz takvog niza leksičkih jedinica tačan, verovatnoća sa kojom se niz leksičkih jedinica koji sadrži odgovor - nadje kao odgovarajući upitu, i verovatnoća sa kojom se tačan odgovor iz takvog niza leksičkih jedinica i nadje). Kako je $f=f_1 x f_2 x f_3 x f_4$, to su i $p_f=p_{f1}*p_{f2}*p_{f3}*p_{f4}$, $p'_f=p'_{f1}*p'_{f2}*p'_{f3}*p'_{f4}$, gde su p_{f1}, p_{f2}, p_{f3}, p_{f4} i p'_{f1}, p'_{f2}, p'_{f3}, p'_{f4} - preciznosti i potpunosti preslikavanja f_1 - f_4. Ako su R,A,D,C - oznake za proizvoljnu relaciju, atribut, domen i konstantu iz sheme, i NLJ(x)- niz leksičkih jedinica koji "odgovara" objektu x ($x \in \{R,D,A,C\}$) prema kriterijumu eksperta ili iskustvu sa velikim brojem tekstova, onda je:

$p_{f1}=P(r \in NLJ(R) \mid r \in f_1(R))$,

$p'_{f1}=P(r \in f_1(R) \mid r \in NLJ(R))$, i slično za p_{f2}-p_{f4}, tj. p'_{f2}-p'_{f4} (slika 1 ilustruje mogućnosti za preslikavanje f_3 (slično za f_1, f_2, f_4); samo isprekidana crta označava uspešno preslikavanje atributa A_2).

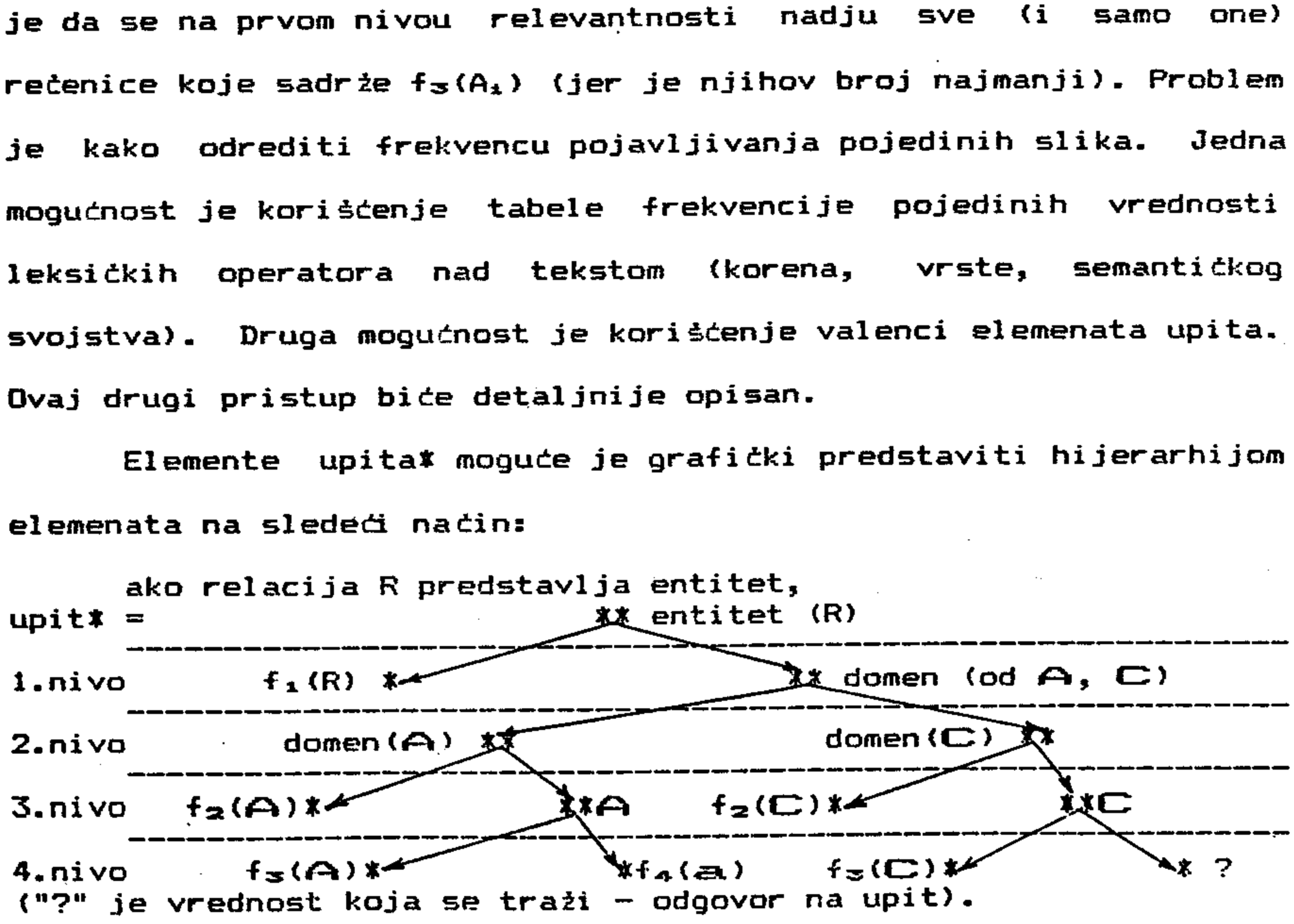


Slika 1. Mogućnosti za preslikavanja f_1-f_4 na primeru f_3; samo isprekidana crta predstavlja uspešno preslikavanje; u iscrtanom delu relacione sheme f nije definisana, tj. nizovi leksičkih jedinica u tekstu, u koje se ovaj deo relacione sheme preslikava preslikavanjem f, prazni su - nisu nadjeni u tekstu.

Kako su karakteristike preslikavanja f_1 – f_4 čisto iskustvene i pripadaju kako lingvističkoj problematici tako i semantici modela, to je maksimiziranje verovatnoća p, p', osnovni cilj projektovanja sistema i pojedinaćne sheme (npr. eksperimentalnog sistema u delu 4.1.1.).

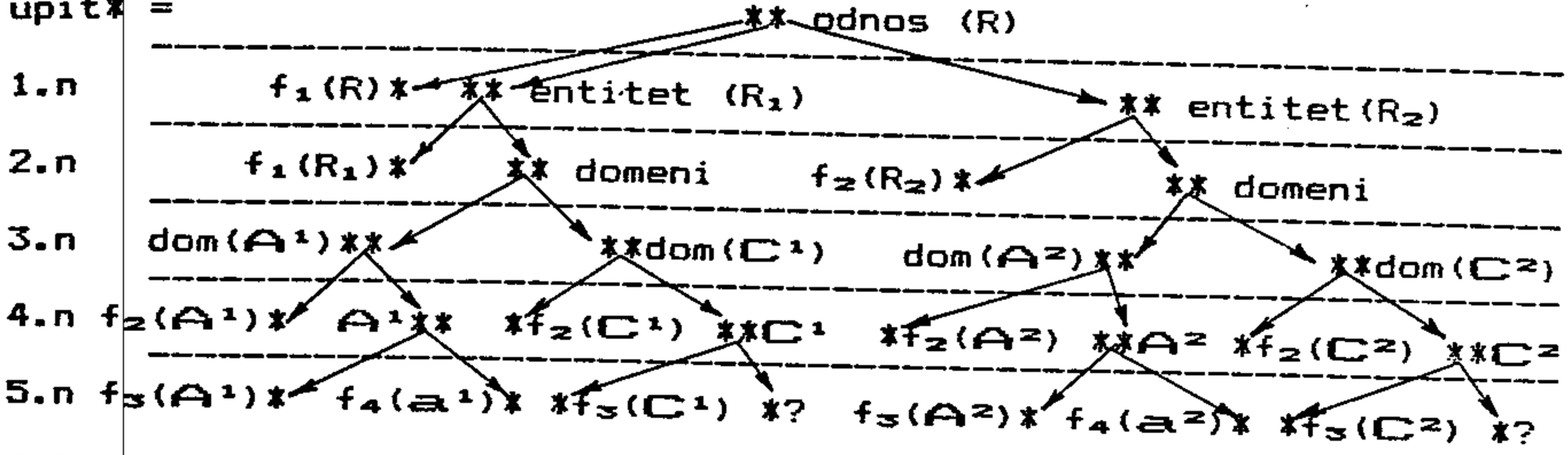
Nalaženje odgovora na upit nad tekstom (preslikavanje odgt upita nad tekstom u odgovor na taj upit) sastoji se u nalaženju relevantnih rećenica (onih koje sadrže slike elemenata upita) i izdvajanju odgovora iz slike ciljnog atributa iz tih rećenica. Radi što efikasnijeg pronalaženja skupa relevantnih rećenica, poželjno je da se on gradi iterativno, po nivoima "relevantnosti", prema hijerarhiji "diskriminativne moći" slika pojedinih elemenata upita (npr, ako je od svih slika relacije, domena, atributa, konstanti – iz upita*, (f_1(R),f_2(domen(A)), f_2(domen(C)), f_3(A), f_3(C), f_4(a)), broj pojavljivanja slike $f_3(A_1)$ najmanji, poželjno je da se na prvom nivou relevantnosti nadju sve (i samo one) rećenice koje sadrže $f_3(A_1)$ (jer je njihov broj najmanji). Problem je kako odrediti frekvencu pojavljivanja pojedinih slika. Jedna mogućnost je korišćenje tabele frekvencije pojedinih vrednosti leksićkih operatora nad tekstom (korena, vrste, semantićkog svojstva). Druga mogućnost je korišćenje valenci elemenata upita. Ovaj drugi pristup biće detaljnije opisan.

Elemente upita* moguće je grafički predstaviti hijerarhijom elemenata na sledeći način:



("?" je vrednost koja se traži – odgovor na upit).

ako relacija R predstavlja odnos entiteta predstavljenih relacijama R_1, R_2,



(elementi sa gornjim indeksom 1 su oni elementi upita* koji pripadaju entitetu u relaciji R1; sa gornjim indeksom 2 -pripadaju entitetu u relaciji R2).

Intuitivno, valenca jednog elementa upita (entiteta, domena, atributa, specifične vrednosti) je broj nadredjenih elemenata koji mu odgovaraju. Valenca se može definisati na sledeći način:

- odnosi imaju valencu 1;

- valenca entiteta je broj odnosa u kojima entitet učestvuje (u shemi);

- valenca domena je broj entiteta + broj odnosa u kojima se domen pojavljuje;

- valenca atributa je broj entiteta + broj odnosa u kojima se domen pojavljuje;

- valenca konstante je valenca odgovarajućeg domena.

Kako se u tekstu jedan entitet (atribut, domen) može pojaviti više puta, diskriminativna moć pojedinog elementa je njegova <u>popravljena valenca</u> koja uzima u obzir broj pojavljivanja elementa i jednaka je proizvodu procenjenog broja pojavljivanja elementa i njegove valence (za konstantu jednaka je samoj valenci).

Svaki terminalni čvor u grafičkoj reprezentaciji upita definiše nivo relevantnosti rečenice koja taj čvor sadrži. Kriterijum za hijerarhiju slika elemenata koji se traže i koji

odredjuju nivo relevantnosti rečenice može sada biti najmanja popravljena valenca (slike elemenata traže se u rastućem poretku popravljenih valenci). Pritom je kriterijum dobar u meri u kojoj su ispunjene sledeće pretpostavke o relacionoj shemi:

a) shema je kompletna u smislu da odgovara rečenicama teksta i da rečenice odgovaraju shemi;

b) tekst je uniforman u smislu da ako je u njemu prisutno više od jednog pojavljivanja datog entiteta ili odnosa, onda su prisutni u tekstu i svi atributi tih pojavljivanja entiteta tj. odnosa (ako se, npr. u tekstu biografije pominju dva autora, onda su za oba prisutni i atributi kao što su godina rodjenja, mesto rodjenja itd.);

c) broj pojavljivanja specifičnog elementa u tekstu može se proceniti.

Kako popravljena valenca može biti nepoznata tj. odredjena sa nedovoljnom sigurnošću, algoritam odredjivanja hijerarhije relevantnih rečenica koristi i druge kriterijume (u opadajućem poretku značaja - kriterijum nižeg značaja primenjuje se samo ako je kriterijum višeg značaja neprimenljiv):

1) najmanja popravljena valenca prvo;

2) konstanta prvo;

3) poredak traženja slika elemenata odozdo na gore (u hijerarhiji elemenata upita).

<u>Algoritam odredjivanja relevantnih rečenica:</u>

```
      naći skup tekstova koji sadrže bilo koju konstantu iz upita
kao svoju ključnu reč;
      for( svaki tekst)
      {
        relreč (relevantne rečenice) = {sve rečenice};
        while(postoji neobradjeni terminalni čvor '*' u upitu)
        {
          izabrati skup A čvorova sa najmanjom (=) popravljenom
valencom;
          izabrati podskup A1 od A koji sadrži konstante;
          while(A1 nije prazan  ili A nije prazan)
          {
```

```
            if(A1 prazan) A1=A;
            izabrati sliku elementa iz A1 na najnižem nivou;
            izbrisati sve rečenice iz relreč koje ne sadrže sliku;
            izbrisati odgovarajući čvor iz A1 i A, i sve ostale
čvorove sa istom vrednošću;
          }
        }
      }
```

Pošto je skup relevantnih rečenica najvišeg nivoa nadjen, u svakoj od njih traži se leksička jedinica (skup leksičkih jedinica) sa karakteristikama $f_3(C)$ (i/ili f_2(domen(C))). Te leksičke jedinice, zajedno sa leksičkim jedinicama koje odgovaraju slikama atributa i konstanti iz uslova upita, unose se kao n-torke u relacionu bazu podataka koja sada postaje hibridna: stvarno/virtuelna. Odgovor na svaki sledeći upit traži se prvo u relacionoj bazi podataka, a tek ako nije prisutan, primenjuju se preslikavanja f, odgt. Vrednost atributa koja u takvoj n-torci relacije nije nadjena (ili upitom nije tražena), postaje null (nepoznata ili nedefinisana). Pre nego što se opiše eksperimentalni sistem koji odgovara jednoj realizaciji operatora izdvajanja informacije iz teksta (u sledećoj tački), u ovom delu ću još prikazati rezultate srodne ovom operatoru.

Osim pristupa reprezentaciji znanja iz teksta relacionim modelom ograničenog i nepovezanog znanja, u cilju izdvajanja informacije iz teksta, drugi pristupi ovom problemu zastupljeni su u literaturi.

Semantičke mreže kao reprezentacija povezanog znanja sadržanog u tekstu prikazane su u nizu radova. U ranije citiranom radu Simmonsa i Chestera [62], prikazan je sistem aksioma klauzalne logike (predikatske logike specifične forme, sa proceduralnom interpretacijom) za dvostranu transformaciju teksta u semantičku mrežu i obratno. U semantičkoj mreži čvorovi odgovaraju rečima a lukovi su obeleženi odnosom u kom se nalaze čvorovi koje luk povezuje. Gramatika kojom se jezik opisije je

konačno klauzalna gramatika (definite clause grammar) koja predstavlja proceduralnu interpretaciju klauzalne logike. U ovoj gramatici posebnu grupu pravila čine tvrdjenja o pojedinačnim rečima i njihovim svojstvima (odgovaraju rečničkim ulazima, semantičkim svojstvima reči, semantičkim svojstvima predloga u specifičnom kontekstu), dok su pravila strukturne gramatike tipa antecedens --> konsekvens, gde se, da bi se dokazao (ili opovrgao) konsekvens, poziva dokazivanje elemenata antecedensa. Pri analizi novih tekstova, ako se ukaže potreba za novim pravilom - ono se dodaje bazi. Isti autori izgradjuju koncept semantičke mreže sa promenljivim, istinosnim funkcijama (semantičkim predikatima) i kvantifikovanim iskazima. Iz semantičke mreže odgovor na pitanje dobija se specifičnim algoritmom izvodjenja [60, 61]. Semantička mreža koja predstavlja kontekst tretira se kao uzajamno povezani skup istinosnih iskaza, pitanje se uzima kao hipoteza, a algoritam izvodjenja odredjuje da li je pitanje tačno (sa vraćenim vrednostima promenljive u pitanju), netačno ili neodredjeno s obzirom na semantičku mrežu. Osim jednostavnog sravnjivanja pitanja sa semantičkom mrežom, pristup uključuje pravila izvodjenja tipa

((X mesto Z) (Y deo_od Z) (X mesto Y)). Sistem je implementiran u LISP-u.

Razni pristupi obradi teksta i razumevanju prirodnog jezika izloženi su u [73]. P.Thorndyke izlaže obradu znanja iz teksta vodjenu obrascem i zasnovanu na hijerarhijskoj memoriji. Npr. jedna priča mogla bi biti organizovana prema sledećim obrascima: naziv priče (postavka (lokacija), tema(dogadjaj, cilj), zaplet(epizoda$_1$, epizoda$_2$), razrešavanje(stanje)) (svaki nivo unutrašnjih zagrada označava odgovarajući nivo u hijerarhiji). Osnovna teza je da koncepti na višem nivou hijerarhije odgovaraju

značajnijim informacijama tj. onima koje čovek, u svojoj memoriji, "bolje" pamti. Sistemi za razumevanje prirodnog jezika zasnovani na pravilima (za razliku od algoritama) prezentirani su u radovima Riesbecka i, Schanka i Wilenskog. Riesbeck predstavlja tekst reprezentacijom značenja konceptualne zavisnosti, rečnik - skupom pravila, a posebnu pažnju posvećuje aktivacionom mehanizmu pravila (kojih može biti i na hiljade) i sugeriše "očekivanje" kao ograničenje postavljeno prethodno izvršenim pravilom, koje treba da zadovolji tekuće pravilo iz rečnika da bi bilo primenjeno. Schank i Wilensky bave se integrisanjem znanja sa nivoa rečenice u znanja na nivou teksta. Jedan metod za ovo integrisanje je upotreba skripta - obrazaca za domene sa dobro strukturiranim znanjem. Kako skript nije uvek na raspolaganju (za nove domene, npr.) autori predlažu upotrebu znanja o ciljevima koje imaju likovi u priči, da bi se razumelo značenje priče, tj. ponašanje likova. Ciljevi su globalno podeljeni u nekoliko grupa, a svaki cilj iz odredjene grupe aktivira izvesni skup pravila.

Prema opisnim principima projektovan je veći broj sistema za "razumevanje" teksta tj. za nalaženje činjenice u tekstu.

Prototipni inteligentni informacioni sistem, RESEARCHER, [36], prihvata tekst na prirodnom jeziku (apstrakte patenata), "razume" tekst u smislu povezivanja i prepoznavanja činjenica iz teksta sa postojećim činjenicama i obrascima u memoriji, dodaje sadržaju memorije informaciju sadržanu u tekstu, generališući ukupno znanje, i odgovara na pitanja na osnovu sadržaja memorije. Osim što sadrži opštu semantičku informaciju o rečima i konceptima, memorija sadrži i opis konstrukcije objekata o kojima apstrakti patenata govore, i odnosa njihovih delova. Algoritamski je ostvareno i razrešavanje višeznačnosti tipa odnosa pojedinih delova rečenica i to identifikacijom mesta gde višeznačnost može

da se pojavi tj. gde od memorije da se zahteva razrešavanje. Jednoznačna interpretacija se dobija na osnovu primera mogućih konfiguracija već prisutnih u memoriji.

Sistem za prevodjenje teksta u bazu podataka nad kojom se zatim postavljaju upiti izložen je u [20]. Ovde su tekstovi - medicinski izveštaji i oni se smeštaju u relacionu bazu podataka, jedan izveštaj (rečenica) po n-torci. Format relacije odredjuje se kroz lingvističke pravilnosti u tekstu i zatim verifikuje od strane eksperta. Kolone relacije imenuju se odgovarajućom ulogom reči u rečenici; uloga se odredjuje pomoću rečnika i analizom teksta, tako da reči koje se pojavljuju u sličnom kontekstu nose sličnu informaciju - s obzirom da postoji veza izmedju distribucije reči i njenog značenja. Pošto se izvrši ova konverzija teksta u relacije, nad relacijom se mogu postavljati upiti bilo upitnim bilo prirodnim jezikom. Pristup je baziran na sintaksnoj obradi, a ovakvo procesiranje medicinske informacije moguće je jer je jezik ograničen i dopušta održavanje eksplicitne liste sinonima, sintaksne konstrukcije jednostavne, domen daje specijalizovano značenje rečima, višeznačnost je svedena na minimum.

Još jedan sistem za prevodjenje rečenica - iz podskupa nemačkog jezika u bazu podataka - opisan je u [18]. Dobijena baza podataka služi kao osnova za sistem odgovaranja na upite. Kao semantička komponenta (prelazna faza izmedju teksta i njegove reprezentacije bazom podataka) pojavljuju se "strukturne reprezentacije domena teksta", a ove se zatim prevode u bazu PROLOG-iskaza, ili se definišu pravila izvodjenja direktno iz ove semantičke komponente.

Na Univerzitetu u Pragu, u okviru projekta računarske lingvistike, razvijen je sistem TIBAQ [22, 57], koji izdvaja

informacije iz tehničkih tekstova, obogaćuje ih primenom pravila izvodjenja, smešta ih i pretražuje na zahtev (upit) formulisan u prirodnom jeziku. TIBAQ se sastoji od lingvističke analize ulaznog teksta i upita, skupa pravila izvodjenja koja operišu nad izlazom iz lingvističkog analizatora, traženja odgovarajućeg odgovora i sinteze nadjenog odgovora. Lingvistička analiza ulaznog teksta i upita izvodi se u dve etape - morfološkoj i sintaksno-semantičkoj, što kao rezultat daje jednoznačnu "tektogramatičku" reprezentaciju svake rečenice (jedan oblik drveta zavisnosti), koje se povezuju u neku vrstu semantičke mreže. Poredjenjem reprezentacije upita i teksta nalaze se relevantni iskazi u tekstu na osnovu kojih se nalazi odgovor. Na skup relevantnih iskaza primenjuju se, zatim, pravila izvodjenja koja generišu nove relevantne iskaze (npr. pravilo brisanja priloške fraze: iz "Moguće je održavati X bez upotrebe Y" sledi "Moguće je održavati X"). Ukupni skup relevantnih iskaza poredi se sa reprezentacijom upita, i iskaz čija reprezentacija najbliže odgovara reprezentaciji upita, biraju se za odgovor. Konačni prirodno-jezički odgovor se zatim sintetizuje posebnom procedurom.

Američka korporacija "Quantum Development" objavila je 1985. godine komercijalni proizvod za "upravljanje znanjem" [6]. Sistem za upravljanje znanjem (implementiran za VAX i IBM PC) u stanju je da smesti i da razume veliki broj dokumenata (prema izveštaju više od 250 miliona) unutar svake teme za upravljanje znanjem. Za sistem se kaže da brzo indeksira i pretražuje informaciju i sintetiše prirodno-jezički odgovor na prirodno-jezički upit. Pristup problemu zasnovan je na sistemima pravila koja izvode "značenje" iz teksta i zatim ga simbolički struktuiraju i manipulišu njime.

4.1.1. EKSPERIMENTALNI SISTEM

Komponente sistema za izdvajanje informacije iz teksta, opisanog u ovom radu, su: 1) opis baze podataka, tj. definisanje virtuelne relacione sheme, entiteta, odnosa, relacija, domena i atributa u njima; definisanje preslikavanja f_1 – f_4 elemenata relacione sheme u elemente teksta; 2) realizacija preslikavanja upita nad relacionom bazom podataka u upit nad tekstom (f:upit --> upitt); 3) realizacija preslikavanja odgt (upit nad tekstom (upitt) --> odgovor na upit), tj.

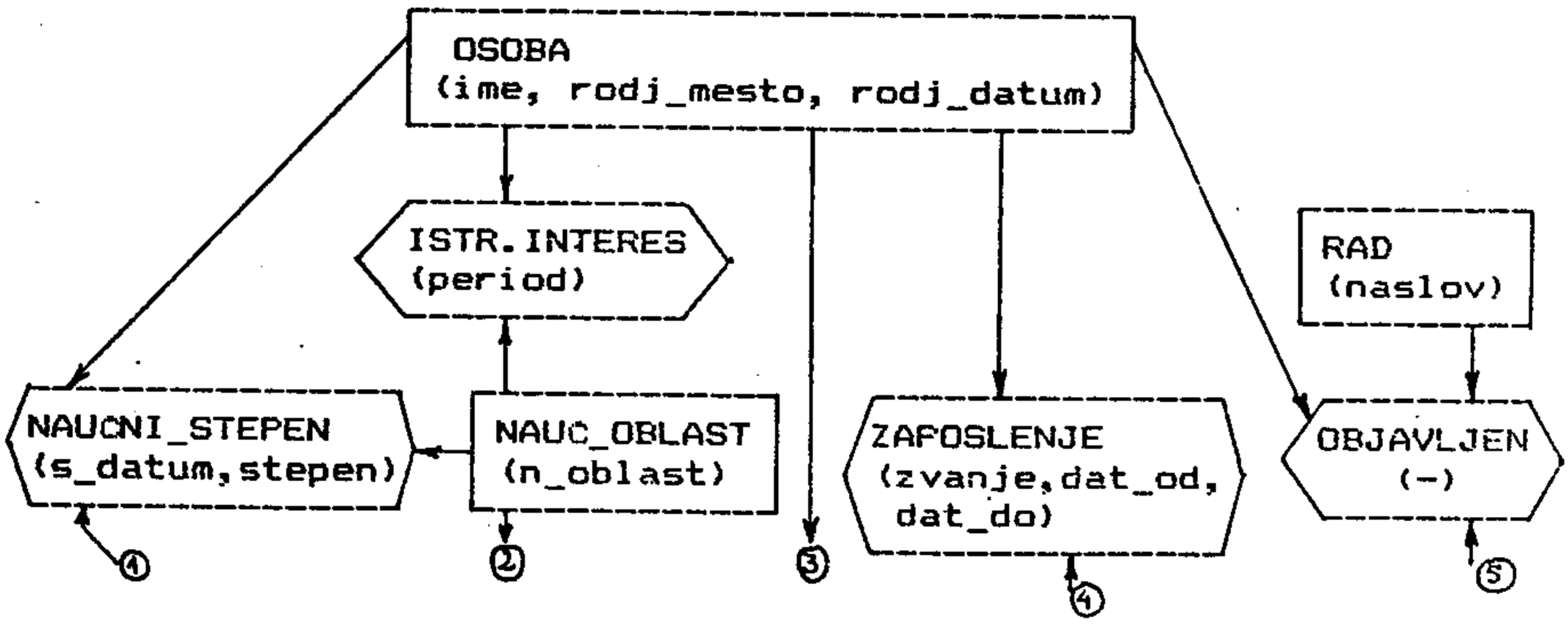
3a) nalaženje skupa relevantnih rećenica,

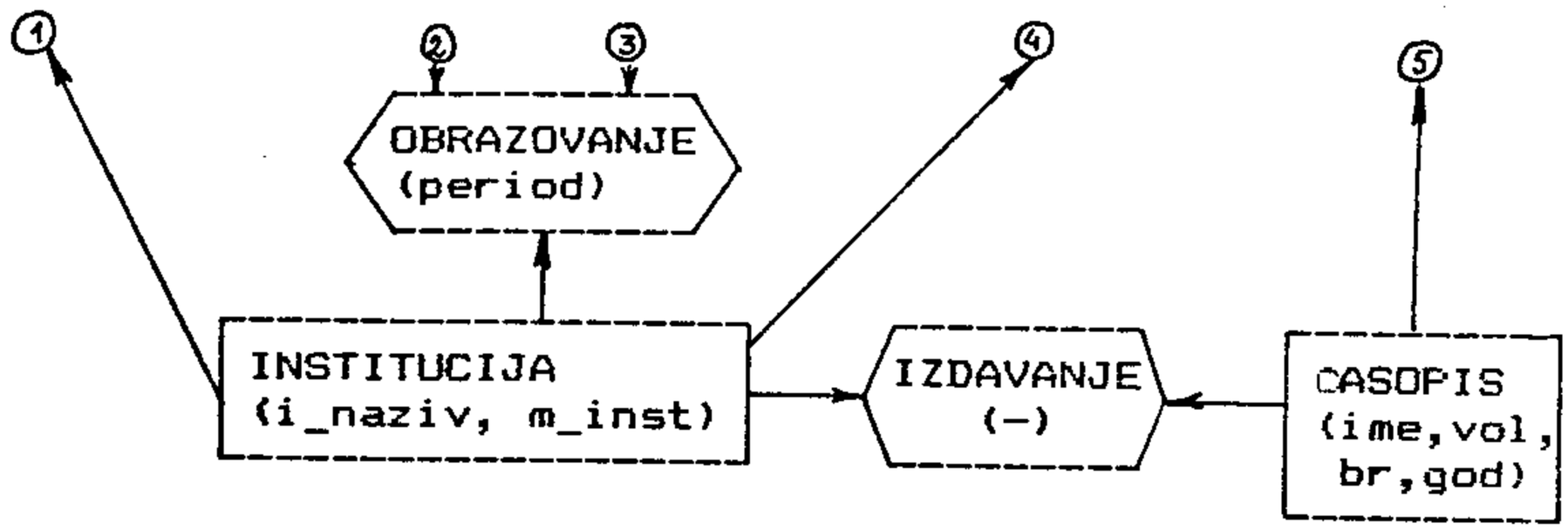
3b) nalaženje odgovora iz njih,

3c) unošenje odgovora u relacionu bazu podataka.

Konstruisani eksperimentalni sistem za izdvajanje informacije iz tekstuelne baze podataka odnosi se na skup biografija naučnika-istraživača, predstavljenih leksičkim podacima u relacionoj bazi podataka, sa razrešenim višeznačnostima i odredjenim originalima zamenica (primenjenim operatorima RAMB i PRONR).

Neka je virtuelna relaciona shema koja odgovara tekstuelnoj bazi podataka predstavljena sledećim dijagramom entiteta-odnosa [2] (entiteti su u pravougaonicima, odnosi u šestougaonicima a odgovarajući atributi su u zagradama):





Odgovarajuće relacije sa pripadnim atributima neka su:

OSOBA (ime, rodj_mesto, rodj_datum);

NAUC_OBLAST (n_oblast);

INSTITUCIJA (i_naziv, m_inst);

OBRAZOVANJE (ime, o_oblast, i_naziv, period);

ISTR_INTERES (ime, i_oblast, period);

NAUCNI_STEPEN (ime, i_naziv, s_datum, stepen, s_oblast);

ZAPOSLENJE (ime, i_naziv, zvanje, dat_od, dat_do);

RAD (naslov);

CASOPIS (ime, vol, br, god);

OBJAVLJEN (ime, naslov, ć_ime);

IZDAVANJE (naziv, ć_ime).

Prvu komponentu sistema (opis relacione baze podataka) moguće je ostvariti interaktivno, dijalogom sledećeg tipa sa sistemom (podvučeni tekst je pretpostavljeni odgovor sistema):

ime baze podataka: ISTRAŽIVACI.

relacije u bazi podataka: OSOBA, NAUC_OBLAST, INSTITUCIJA,
OBRAZOVANJE, ISTR_INTERES,
NAUCNI_STEPEN, ZAPOSLENJE, RAD,
CASOPIS, OBJAVLJEN, IZDAVANJE.

parovi: (atribut,domen)

OSOBA: ime: os_ime;
rodj_mesto: mesto;
rodj_datum: datum.

NAUC_OBLAST: n_oblast: oblast.

INSTITUCIJA: i_naziv: inst_ime;

m_inst: mesto.

OBRAZOVANJE: ime: os_ime;

o_oblast: oblast;

i_naziv: inst_ime;

period: (datum,datum) U

{leks: svojstvo(leks)=PRT - prezent}.

ISTR INTERES: ime: os_ime;

i_oblast: oblast;

period: OBRAZOVANJE.period.

NAUCNI STEPEN: ime: os_ime;

i_naziv: inst_ime;

s_datum: datum;

stepen: stepen;

s_oblast: oblast.

ZAPOSLENJE: ime: os_ime;

i_naziv: inst_ime;

zvanje: zvanje;

dat_od: datum;

dat_do: datum.

RAD: naslov: r_ime.

CASOPIS: ime: č_ime;

vol: r_broj;

br: a_broj;

god: datum.

OBJAVLJEN: ime: os_ime;

naslov: r_ime;

č_ime: č_ime.

IZDAVANJE: naziv: inst_ime;

č_ime: č_ime.

Sledeća potrebna informacija je način na koji su preslikavanja f_1 -f_4 definisana.

Za preslikavanje f_1 (relacija) bitno je još i da li relacija predstavlja entitet ili odnos, pa se dijalog odvija na sledeći način:

```
RELACIJE: entitet ili odnos?
        OSOBA: entitet;
            deskriptor? .
        NAUC_OBLAST: entitet;
            deskriptor? [koren="nauka"];
                        [koren="tražiti", pref="is"];
                        [koren="učiti"].
        INSTITUCIJA: entitet;
            deskriptor? .
        OBRAZOVANJE: odnos;
            uključeni entiteti? OSOBA, NAUC_OBLAST, INSTITUCIJA;
            deskriptor? .
        ISTR_INTERES: odnos;
            uključeni entiteti? OSOBA, NAUC_OBLAST;
            deskriptor? [koren="tražiti", pref="is"];
                        [koren="baviti"].
        NAUCNI_STEPEN: odnos;
            uključeni entiteti? OSOBA, INSTITUCIJA, NAUC_OBLAST;
            deskriptor? .
        ZAPOSLENJE: odnos;
            uključeni entiteti? OSOBA, INSTITUCIJA;
            deskriptor? [koren="rad"];
                        [koren="posao"].
        RAD: entitet;
            deskriptor? [koren="rad"];
```

[koren="članak"].

CASOPIS: entitet;

deskriptor? .

OBJAVLJEN: odnos;

uključeni entiteti? OSOBA, RAD, CASOPIS;

deskriptor? [koren="objaviti"];

[koren="štampa"].

IZDAVANJE: odnos;

uključeni entiteti? INSTITUCIJA, CASOPIS;

deskriptor? [koren="izdati"];

[koren="štampa"].

Preslikavanje f_2 (domena):

OPIS NOVIH DOMENA:

os ime: definisan skupom? ne;

vrsta reči? vlastito ime;

sem.svojstvo? HUM;

prethodi reč? ;

sledi reč? ;

procedure za definisanje operatora? .

mesto: definisan skupom? ne;

vrsta reči? vlastito ime;

sem. svojstvo? LOC.

datum: definisan skupom? ne;

vrsta reči? komponovana: [dan][mesec]godina;

dan: definisan skupom? ne;

vrsta reči? broj;

interval? 1,31.

mesec: definisan skupom? da:

{ januar, februar, mart, april, maj, jun, jul, avgust, septembar, oktobar, novembar, decembar};

godina: definisan skupom? ne;

vrsta reči? broj;

interval? 0,2000;

prethodi reč? ne;

sledi reč? [koren="godina"];

sem. svojstvo? TIM;

procedure za definisanje operatora?

pre(x,y);

posle(x,y);

istovremeno(x,y).

oblast:definisan skupom? ne;

vrsta reči? [pridlimenica;

sem. svojstvo? FLD.

inst ime: definisan skupom? ne;

vrsta reči? vlastito ime;

sem. svojstvo? ASM.

stepen: definisan skupom? ne;

vrsta reči? ;

sem. svojstvo? HUM;

koren? "magistar";

"doktor";

"diploma".

zvanje: definisan skupom? ne;

vrsta reči? [pridlimenica;

sem. svojstvo? HUM.

r ime: definisan skupom? ne;

vrsta reči? ;

sem. svojstvo? ;

prethodi reč? '"';

sledi reč? '"'.

č ime: definisan skupom? ne;

vrsta reči? vlastito ime;

sem. svojstvo? ASM.

r broj: definisan skupom? ne;

vrsta reči? rimski broj;

interval? [I - .

a broj: definisan skupom? ne;

vrsta reči? broj;

interval? [1 - .

Preslikavanje f_s (atributa):

Atributi: dopunski opis?

ime? .

rodj mesto? koren="roditi".

rodj datum? koren="roditi".

n oblast? .

i naziv? .

m inst? .

i oblast? .

period? da;

prethodi reč? [od];

sledi reč? [do].

s datum? .

stepen? .

zvanje? s_oblast.

dat od? da;

koren? [početi];

[posao, pref=za];

sem. svojstvo? PAC (pozitivna akcija);

prethodi reč? [od];

sledi reč? .

dat_do? da;

vrsta_reči? [glagol];

koren? ;

sem. svojstvo? NAC (negativna akcija);

prethodi_reč? [do];

sledi_reč? .

naslov? .

vol? da;

koren? "vol".

br? .

god? da;

koren? "godina".

č_ime? .

Kao što je već rečeno, prethodno opisanoj shemi ne odgovara ni jedna fizička relacija. Umesto toga, kolekcija tekstova je smeštena u relaciji tekstuelne baze podataka (nešto pojednostavljenog) oblika:

leks_tekst (rčn#, id, deskr),

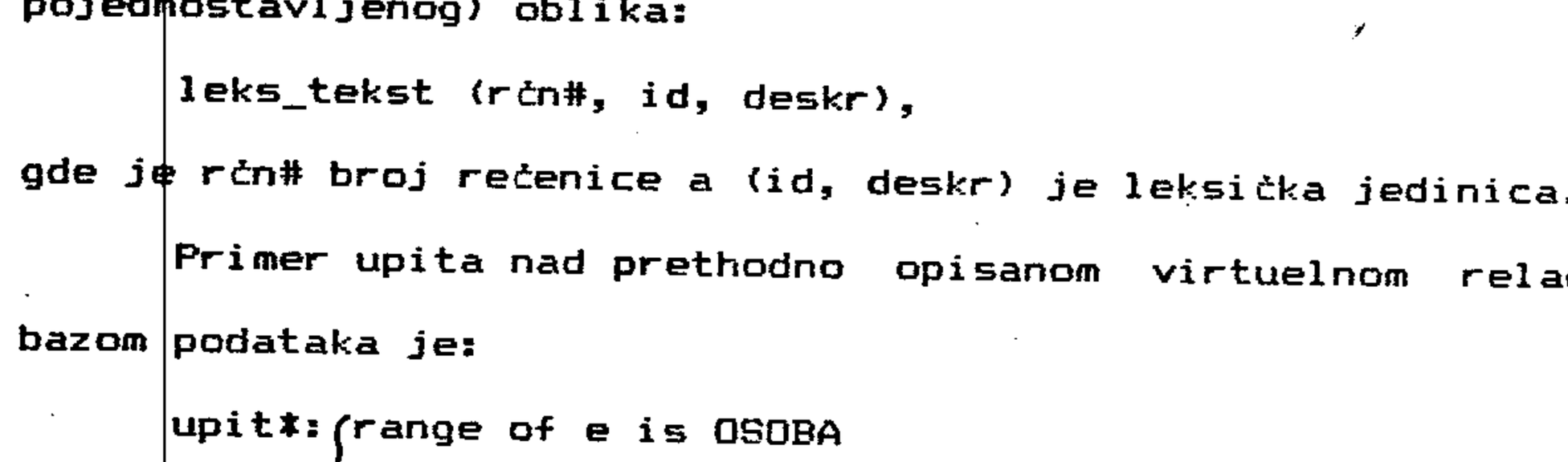
gde je rčn# broj rečenice a (id, deskr) je leksička jedinica.

Primer upita nad prethodno opisanom virtuelnom relacionom bazom podataka je:

upit*: range of e is OSOBA
retrieve (e.rodj_mesto) where e.ime="Nikola Tesla"

(gde je rodjen Nikola Tesla?)

Grafička predstava ovog upita je sledeća:



①	②	③	④
f_3(ime) *	f_4(Nikola Tesla) *	* f_5(rodj_mesto)	* ?
=vl.ime, HUM.	=Nikola Tesla.	=koren="roditi", LOC.	=vl.ime, LOC.

tj. upitt* = f(upit*)= Naći leksičku jedinicu leks sa vrsta(leks)=vlastito_ime, feat(leks)=LOC (f_2(domen(C))), u rečenici u kojoj se nalaze: reč sa korenom "roditi" (f_3(C)), konstanta "Nikola Tesla"(f_4(a)), i vlastito ime sa svojstvom HUM (f_2(A)).

Ako se broj pojavljivanja entiteta OSOBA, domena os_ime, mesto, atributa ime i rodj_mesto procenjuje na 1, popravljena valenca jednaka je valenci tih elemenata upita, tj.

pop.valenca(OSOBA) = 6, pop.valenca(os_ime) = 6,

pop_valenca(mesto) = 2, pop_valenca(ime) = 6,

pop_valenca(rodj_mesto) = 1,

pop_valenca(Nikola Tesla) = 6,

i odredjivanje skupa relevantnih rečenica odvijalo bi se po sledećim nivoima: naći sve rečenice koje sadrže reč sa korenom "roditi" (tj. leks: koren(leks)=1549) i vlastito ime sa semantičkim svojstvom LOC (tj. leks: id(leks)<0, feat(leks)=32), što bi se nad stvarnom relacijom leks_tekst zapisalo kao upit:

```
range of e is leks_tekst
range of u is leks_tekst
retrieve(e.rćn#) where e.rćn#=u.rćn# and
                       koren(e.id,e.deskr)=1549 and u.id<0
                       and feat(u.id,u.deskr)=32.
```

Već na ovom nivou bila bi pronadjena samo jedna rečenica (sa brojem 1):

reč	rćn#	id	deskr
Tesla	1	-1914100000	100026
,	1	,	
Nikola	1	-1229100000	100026
,	1	,	
naučnik	1	1109001501	100026
i	1	825000000	900028
pronalazač	1	1438000401	100026
iz	1	818000000	820032
oblasti	1	1285000001	101022
elektrofizike	1	504000001	101022
i	1	825000000	900028
elektrotehnike	1	1869000000	101022

,	1	,	
rodjen	1	1549001201	300066
je	1	163000109	250882
u	1	2008000000	847032
Smiljanu	1	-1777100002	100032
kraj	1	973000000	820032
Gospića	1	-711100001	100032
:			
:			
.			

Upitom

```
range of e is leks_tekst
retrieve(e.id, e.deskr) where e.rčn#= 1 and
                              feat(e.id, e.deskr)=32 and e.id<0,
```

iz ove jedine rečenice dobile bi se dve leksičke jedinice (id, deskr) sa vrstom=vlastito ime, svojstvom=LOC = (-1777100000, 100032), (-711100000, 100032), koje, prema rečniku vlastitih fraza, odgovaraju vlastitim imenicama Smiljan i Gospić. Da bi se od ove dve reči izabrala prava, potrebna je analiza koja izlazi iz opsega ovog rada (npr. kriterijum udaljenosti [31]). Druga mogućnost je smestiti ceo izraz (Smiljan kraj Gospića) kao vrednost atributa rodj_mesto, n-torke (Nikola Tesla, rodj_mesto, rodj_datum) u relaciju OSOBA. Vrednost atributa rodj_datum za ovu n-torku za sada ostaje null.

S obzirom da biografija kao tip teksta ne zadovoljava sve pretpostavke pod kojima redosled rastućih popravljenih valenci obezbedjuje efikasno nalaženje relevantnih rečenica (npr. shema nije kompletna u smislu da sve rečenice teksta odgovaraju shemi), odredjivanje nivoa relevantnosti rečenica moguće je zaobići jednostavnim prevodjenjem upita* u upit nad stvarnom relacijom leks_tekst u kojoj je leksikalizovani tekst, a na osnovu preslikavanja $f_1 - f_4$ elemenata tog upita, na sledeći način:

```
range of e is leks_tekst
range of u is leks_tekst
range of v is leks_tekst
retrieve(e.id) where e.id<0 and feat(e.id, e.deskr)=32 and
          e.rčn#=u.rčn# and koren(u.id, u.deskr)=1549
          and  u.rčn#=v.rčn# and v.id= -1914100000
```

Sada sam RSUBP vodi brigu o tome kojim će redosledom testirati

uslove kvalifikacije upita, i vraća, kao odgovor, dva identifikatora -1777100000, -711100000, koji odgovaraju, redom, vlastitim imenima Smiljan, Gospić.

4.2. IZVODJENJE U TEKSTUELNOJ BAZI PODATAKA

- RELACIONI MODEL BAZE ZNANJA SA NULL-VREDNOSTIMA

U relacionom modelu baze znanja sadržanog u tekstu, opisanom u prethodnom delu, kao pojedina n-torka relacije posmatra se relevantni (za atribute n-torke) sadržaj pojedine rečenice. Kako takva rečenica ne mora da sadrži podatke relevantne za sve atribute, odgovarajuća n-torka sadrži null-vrednosti kao vrednosti atributa za koje u rečenici nema relevantnih podataka. Relacioni model koji se koristi za predstavljanje znanja iz teksta je, stoga, prošireni relacioni model sa dve vrste null-vrednosti ([12], [43], [69]). Sledeći primer će da ilustruje primenu ovog modela na predstavljanje znanja iz teksta:

Neka je nad skupom biografija definisana virtuelna relaciona shema kao u 4.1.1, i neka je nad relacijom NAUCNI_STEPEN postavljen sledeći upit:

upit*1: {range of e is NAUCNI_STEPEN
{retrieve (e.s_oblast) where e.i_naziv="PMF"

(Iz kojih se naučnih oblasti daju stepeni na Prirodno-matematičkom fakultetu? - A={i_naziv}, C={s_oblast}, a={PMF}).

Pretpostavimo da se sledeće rečenice nalaze u biografiji iz datog skupa:

a) PMF daje doktorate iz hemije.

b) XX je magistrirao iz računarstva.

c) XX je magistrirao na PMF.

Rečenicama a)-c) odgovara sledeći sadržaj relacije NAUCNI_STEPEN (sadržaj nije stvarno prisutan u relaciji pre obrade

upita$*_1$):

NAUCNI_STEPEN	ime	stepen	i_naziv	s_datum	s_oblast
	ϱ	dr	PMF	ϱ	hemija
	XX	mr	ω	ω	računarstvo
	XX	mr	PMF	ω	ω

("ϱ" ima značenje "neprimenljivo svojstvo", a "ω" – nepoznata vrednost). Jedan odgovor – "hemija" može se direktno naći iz jedne rečenice. Drugi odgovor – "računarstvo", nije sadržan u jednoj rečenici sa podatkom "PMF", ali se, s obzirom na relaciju NAUCNI_STEPEN i odnose medju njenim atributima (ime i stepen jednoznačno odredjuju instituciju i_naziv i oblast s_oblast), može izvesti izjednačavanjem vrednosti atributa i_naziv i s-oblast u n-torkama sa istom vrednošću atributa ("ime, stepen").

U delu koji sledi prvo će biti ukratko opisan relacioni model baza podataka sa dve vrste null-vrednosti i teorija zavisnosti medju atributima u tom modelu (4.2.1.) – koja omogućuje izvodjenje ilustrovano prethodnim primerom, a zatim će biti izložena sama metoda izvodjenja informacije (faktuelno izvodjenje u 4.2.2.).

4.2.1. RELACIONI MODEL BAZA PODATAKA SA DVE VRSTE NULL-VREDNOSTI

Najčešće i semantički najopravdanije null-vrednosti kojima se proširuje osnovni relacioni model [10] su nepoznata vrednost (u oznaci "ω" [12], [69]), i neprimenljivo svojstvo (u oznaci "ϱ") [43] (u primeru iz tačke 4.2., iskaz "PMF daje doktorat iz hemije", svojstvo "ime" može biti neprimenljivo ("ϱ"), jer se "daje" može interpretirati kao "može da daje" a ne kao "dao je

osobi X"; s druge strane, ako je osoba XX magistrirala iz računarstva, onda je to bilo <u>nekog</u> (ali nepoznatog -ω, u navedenom kontekstu), datuma, i u <u>nekoj</u> (ali nepoznatoj) instituciji. Dakle, null-vrednosti "ω" i "$\wp$" su semantički različite).

U relacionom modelu baze podataka sa dve vrste null-vrednosti, relacija jednakosti n-torki, pripadnosti n-torke relaciji i inkluzija relacija definišu se u terminima trovalentne logike: svaka od ovih relacija može biti tačna (T), "možda-tačna" (istinosno ω) ili netačna (F). "Možda tačnost" se odnosi na slučaj kada neki elementi n-torke imaju nepoznatu vrednost (ω), pa se stoga ne može utvrditi niti negirati jednakost (pripadnost, inkluzija). Tako, npr. ako se sa τ obeleži istinosna vrednost iskaza, a sa $\doteq$ -trovalentna relacija jednakosti iskaza, onda za dve n-torke t,s, važi sledeće:

$\tau(t \doteq s)=T$ <=> t,s su definisani na istom skupu atributa, i parovi vrednosti t,s na odgovarajućim atributima jednaki su i $\neq\omega$;

$\tau(t \doteq s)=\omega$ <=> $\tau(t \doteq s)\neq T$ i za svaki atribut A_i važi: a) t i s imaju jednake vrednosti na A_i, ili b) t (s) ima vrednost ω na A_i a s (t) ima vrednost $\neq\wp$ na A_i.

Za prošireni relacioni model sa dve vrste null-vrednosti definišu se i proširene operacije relacione algebre ("tačne" i "možda" operacije) i proširene zavisnosti [43, 69]. Tako je, npr. rezultat tačne operacije prirodnog spajanja relacija R(X,Y) i S(X,Z),

$R *_T S = \{t:$ t[X,Y] =r za neku n-torku $r \in R$ i
t[X,Z] =s za neku n-torku $s \in S$ i
$\tau(r[X] \doteq s[X])=T\}$.

Rezultat "možda" operacije prirodnog spajanja zasniva se na relaciji "biti informativniji" za dve "možda jednake" n-torke r,s

[43]: r⊒s<=>za svaki atribut A_i od r,s važi: r=s ili r=ρ ili s=ω na A_i. Npr. za r=(2,3,3) i s=(2,ω,3) važi r⊒s. Pritom $\sup_{\sqsupseteq}$(r,s) =(2,3,3) ali za t=(ω,3,3), takodje je $\sup_{\sqsupseteq}$(s,t)=(2,3,3).

Sada je rezultat "možda"-operacije prirodnog spajanja relacija R, S,

$R *_{\omega} S$ = {t: t[Y]=r[Y] za neku n-torku r∈R i

t[Z]=s[Z] za neku n-torku s∈S i

$\tau(r[X] \dot{=} s[X])=\omega$ i $t[X]=\sup_{\sqsupseteq}\{r[X], s[X]\}$}.

Funkcionalna i višeznačna zavisnost (FZ, VZ, redom), u relacionom modelu sa dve vrste null-vrednosti može se definisati na sledeći način [43] (definicija funkcionalne zavisnosti odgovara "slaboj funkcionalnoj zavisnosti" iz [69]).

U relaciji R proširenog relacionog modela, sa tri disjunktna skupa atributa X,Y,Z važi funkcionalna zavisnost X->Y akko važi:

(1) $(\forall r,s \in R)(\tau(r[X] \dot{=} s[X])=T \Rightarrow \tau(r[Y] \dot{=} s[Y]) \in \{T, \omega\})$. Npr. u relaciji NAUCNI_STEPEN virtuelne relacione sheme iz dela 4.2., logički važi FZ ime->{i_naziv, s_oblast}, a iskazi a) -c) koji odgovaraju toj relaciji (kao u delu 4.2.) poštuju ove zavisnosti (prema definiciji):

r=(XX, mr, ω, ω, računarstvo), s=(XX, mr, PMF, ω, ω),

$\tau(r[X]\ (\equiv(XX, mr)) \dot{=} s[X]\ (\equiv(XX, mr)))=T \Rightarrow$

$\tau(r[Y]\ (\equiv(\omega, računarstvo)) \dot{=} s[Y]\ (\equiv(PMF,\omega)))=\omega$.

U relaciji R(X,Y,Z) proširenog modela, sa tri disjunktna skupa atributa X,Y,Z, važi višeznačna zavisnost (VZ) X->->Y akko je (2) $\tau(Y_{xz} \cup Y_{xz'}(-R) \dot{=} Y_{xz} \cdot \cup Y_{xz'} \cdot (-R)) \in \{T,\omega\}$, za svaki element x iz "tačnog" podskupa od R[X], i z, z' iz R[Z] takve da su Y_{xz}, $Y_{xz'}$ neprazni. $Y_{xz}(-R)$ označava skup onih y iz R[Y] koji sa x,z nisu u relaciji R, ali sa x,z čine n-torke koje su "možda"-jednake sa nekim n-torkama iz R, tj. $Y_{xz}(-R)$={y: y∈R[Y] i (x,y,z)∉R i $\tau(t \dot{=} (x,y,z)) \in \{T,\omega\}$ za neku n-torku t iz R).

Posmatrajmo primer relacije R sa atributima:

STUD - ime studenta,

ISPIT - ispit polagan u prethodnom ispitnom roku,

OCENA - ocena na tom ispitu,

PRED - predmet koji sluša u tekućem semestru.

R(STUD	ISPIT	OCENA	PRED)
stud1	ispit1	7	ω
stud1	ispit1	7	predm2
stud1	ispit2	6	predm1
stud1	ispit2	6	ω
stud2	ispit2	ω	predm2
stud2	ispit2	5	predm3

U relaciji R logički važi VZ STUD ->-> {ISPIT, OCENA} i VZ STUD ->-> {PRED}, a ove zavisnosti važe i prema definiciji višeznačnih zavisnosti u proširenom modelu:[3]

za x=stud1, z=ω, z'=predm2, Y_{xz}={(ispit1, 7), (ispit2,6)},

$Y_{xz}(-R)=\emptyset$, $Y_{xz\cdot}$={(ispit1,7)}, $Y_{xz\cdot}(-R)$={(ispit2,6)},

$\tau(Y_{xz} \cup Y_{xz}(-R) \doteq Y_{xz\cdot} \cup Y_{xz\cdot}(-R))=T$;

za x=stud2, z=predm2, z'=predm3, Y_{xz}={(ispit2, ω)}, $Y_{xz}(-R)=\emptyset$,

$Y_{xz\cdot}$={(ispit2,5)}, $Y_{xz\cdot}(-R)=\emptyset$, $\tau(Y_{xz} \cup Y_{xz}(-R) \doteq Y_{xz\cdot} \cup Y_{xz\cdot}(-R))=\omega$.

Za funkcionalnu i višeznačnu zavisnost u proširenom relacionom modelu važe svojstva analogna svojstvima ovih zavisnosti u osnovnom relacionom modelu:

I) ako FZ X->Y važi u relaciji R proširenog modela, onda u

---

[3] Opštija definicija VZ za slučaj da i domen X dozvoljava null-vrednosti je važenje sledećih dveju inkluzija umesto jednakosti (2): $Y_{xz} \subseteq Y_{xz\cdot} \cup Y_{xz\cdot}(-R)$, $Y_{xz\cdot} \subseteq Y_{xz} \cup Y_{xz}(-R)$. Neka je relacija R' dobijena iz relacije R bez null-vrednosti (u kojoj važi VZ B ->-> A prema definiciji za model bez null-vrednosti), zamenom specifičnih vrednosti vrednostima "ω". Intuitivno, trebalo bi da i u relaciji R' važi VZ B->->A (prema definiciji za uopšteni relacioni model), ali u njoj važe samo navedene inkluzije - a ne jednakost.

R(A	B	C)
1	2	1
1	2	2
2	2	1
2	2	2
3	1	2
3	1	3

R'(A	B	C)
1	2	1
1	2	2
2	2	1
2	2	2
3	ω	2
3	ω	3

Za x=2, z=1, z'=2, Y_{xz}={1,2},
$Y_{xz}(-R)=\emptyset$, $Y_{xz\cdot}$={1,2},
$Y_{xz\cdot}(-R)$={3}, ne važi jednakost
$\tau(Y_{xz} \cup Y_{xz}(-R) \doteq Y_{xz\cdot} \cup Y_{xz\cdot}(-R)) \in \{T,\omega\}$,
a važe gore navedene inkluzije.

relaciji R važi i VZ X->->Y;

II) VZ X->->Y važi u relaciji R(X,Y,Z) proširenog modela akko $\mathcal{T}$(R[X,Y] *$_T$R[X,Z] $\doteq$ R)$\in${T,ω};

III) VZ X->->Y važi u relaciji R(X,Y,Z) proširenog modela akko VZ X->->Z važi u toj relaciji.

Dokazi svojstava I) - III) izvode se kao dokazi analognih tvrdjenja u [43].

Važenje FZ i VZ u proširenom relacionom modelu obezbedjuje razlaganje upita nad relacionom shemom baze znanja sa null-vrednostima na podupite koji u atributima kvalifikacije i ciljnim atributima ne sadrže null-vrednosti. Ovo razlaganje upita naziva se <u>faktuelno izvodjenje</u> i biće opisano u sledećoj tački.

Osim faktuelnog, može se razmatrati i drugi oblik izvodjenja, nazvan <u>deduktivno</u> izvodjenje. Do odgovora se u ovom slučaju dolazi primenom nekih logičkih, semantičkih ili jezičkih pravila na neke informacije sadržane u tekstu. Pritom se u upitu prepoznaju samo poznanice (podaci koje zadajemo); nepoznanica (informacija koje čine odgovor) i posrednica (posrednih informacija preko kojih se odgovor dobija), eksplicitno nema, već se nepoznanice nalaze u implicitnom obliku tj. sadržane su u nekoj informaciji u tekstu. Primeri pravila kojima se vrši ekspliciranje takve implicitne informacije su logička: modus ponens, isključenje trećeg, generalizacija, semantička: tranzitivnost jednakosti, jezička: leksički oblici kvantifikatora - neki, svaki, itd.

Ovu vrstu izvodjenja ilustruje sledeći primer: neka se, osim rečenica a)-c) iz tačke 4.2, u istom kontekstu nalaze i rečenice:

d) Na PMF u Beogradu i Prirodoslovno-matematičkom fakultetu u Zagrebu izučavaju se iste discipline.

e) Na Prirodoslovno-matematičkom fakultetu u Zagrebu vrlo su

razvijene matematičke nauke.

Iz rečenica d), e), odgovor na upit*$_1$ iz tačke 4.2. trebalo bi dopuniti elementom (matematičke nauke) (dobijenim primenom tranzitivnosti jednakosti i interpretacijom reči "isti" relacijom jednakosti).

Problemom deduktivnog izvodjenja u relacionom sistemu baza podataka INGRES [25] bavi se Wong [75]. Predlaže se sintaksno homogeno i semantički konzistentno proširenje upitnog jezika INGRES-a i samog sistema, koje uključuje iskaze (definicije, pravila, za razliku od materijalizacije relacije), i primenu pravila izvodjenja nad tim iskazima. Proširenje omogućuje uniformno procesiranje podataka iz materijalizovanih relacija i relacija - pogleda (definisanih iskazom - pravilom), i generalizaciju integritetnih uslova.

Još jedan izvor ideja i realizacija deduktivnog izvodjenja je kolekcija radova [19]. Ovde ću spomenuti pet rezultata iz ove kolekcije. Kellogg i Travis opisuju deduktivno proširenje upravljanja podacima - sistem DADM, koji uključuje deduktivni procesor, projektovan za korišćenje memorije pravila i tvrdjenja, kao posrednika izmedju jezičkog procesora i konvencionalnog sistema za upravljanje podacima. Memorija pravila posebno treba da uključi teoriju logičke dedukcije i procedure dokazivanja u cilju podrške mehaničkom izvodjenju iz velike baze podataka. DADM je napisan u interlispu. U sistemu je prisutna relaciona baza podataka i niz premisa (pravila) tipa if podatak$_1$ then podatak$_2$, od kojih se u svakoj prilici pravi plan izvodjenja. Ulaz u sistem je logički formalizam tipa: dokazati tvrdjenje if podatak then podatak', tj. upit koji izražava željeni zaključak (podatak'). Plan izvodjenja uključuje i uslove koji moraju biti zadovoljivi pretraživanjem relacione baze podataka (relevantni podaci iz

relacione baze podataka). U sistemu postoje tri grupe relacija: bazne (materijalizacije relacija koje se pretražuju), proceduralne (koje se izračunavaju) i virtuelne - tvrdjenja koja povezuju druge relacije.

Grant i Minker se bave problemom optimizacije u deduktivnim relacionim sistemima baza podataka. Upit nad takvom bazom podataka, u kojoj se nove relacije mogu izvesti iz relacija u konvencionalnoj bazi podataka i odredjenih aksioma, može se transformisati u upit isključivo nad relacijama konvencionalne baze podataka. Npr. u relacionoj bazi podataka mogu da budu smešteni podaci o kursevima koji se drže na fakultetu: kursevi(naziv, predavač, godina, sala). Ako je broj kurseva veliki, uz pretpostavku izraženu tvrdjenjem da se svi kursevi sa prve godine drže u sali A, ekonomičnije je, za kurseve sa prve godine, upamtiti to tvrdjenje kao aksiomu nego dodati atribut "sala" koji će za sve kurseve sa prve godine imati istu vrednost (aksioma je: ($kurs_1$, sala A) <-- kursevi($kurs_1$,$pred_1$,god=1)). Upit: U kojoj sali predavač "a" drži predavanje? - postavljen je nad eksplicitnom relacijom o kursevima i implicitnom relacijom o kursevima sa prve godine i sali A. Dokazivač teorema zamenjuje implicitnu relaciju eksplicitnom relacijom sa desne strane aksiome, a upit postaje konjunkcija uslova nad isključivo eksplicitnim relacijama i može se rastaviti na dva upita. Problem kojim se autori bave je problem globalne optimizacije upita nad konvencionalnom bazom podataka, dobijenih transformacijom upita nad deduktivnom bazom podataka.

Reiter se bavi projektovanjem sistema pretraživanja koji kombinuje tehnike upita relacione baze podataka sa deduktivnom komponentom, za slučaj relacione baze sa velikim brojem podataka u bazi (velikim eksplicitnim delom) i malim brojem aksioma (pravila,

tj. implicitnih podataka). Dokazivač teorema se koristi kao kompilator za implicitni deo baze podataka, i on se primenjuje prvi na zadati upit, proizvodeći skup upita koji treba da budu izvršeni nad eksplicitnim delom baze podataka. Unija odgovora na te upite je skup odgovora polaznog upita. Pritom se adresiraju problemi kao što su neodredjeni odgovor (npr. problem se eliminiše ako je implicitni deo baze podataka Hornovski i ako je upit pozitivan), komunikacija sa sistemom upravljanja bazom podataka, tretiranje jednakosti u definiciji aksioma, definisanje pojma baza podataka (baza podataka je bilo koja teorija prvog reda), dobijanje svih odgovora na upit, odnos univerzalnog kvantifikatora u upitu i aksiome zatvorenosti domena.

Minker opisuje eksperimentalni relacioni sistem baza podataka zasnovan na logici, MRPPS. U ovakvoj bazi podataka, predstavljenoj skupom logičkih formula, izvodjenje novih fakata iz starih (dokazivanje teorema) je prirodna nadgradnja sistema. Karakteristike sistema su više-sortna logika u kojoj su domeni kvantifikatora ograničeni na razne domene, fakti i opšti aksiomi se čuvaju zajedno - u semantičkoj mreži, dva mehanizma izvodjenja se koriste (za Horn-iskaze i ne-Horn-iskaze). MRPPS je sistem procedura dokaza, koji se sastoji od mehanizma izvodjenja i strategije pretraživanja. Prva komponenta se koristi za izvodjenje novih iskaza iz data dva iskaza. Druga komponenta ukazuje na dva iskaza koja treba izabrati da bi se izveo novi iskaz. Upit se sastoji od konjunkcije iskaza i sravnjuje se sa modelom znanja da bi mu se utvrdila dobra formiranost. MRPPS polazi od negacije upita i traži kontradikciju. Odgovor na upit može se dobiti u simboličkom obliku, u prirodnom jeziku ili kao glasovni izlaz.

Sistem DEDUCE (Chang) koristi pristup sličan Minkerovom i Reiterovom. DEDUCE je deduktivni upitni jezik za relacione baze

podataka. U njemu se mogu izraziti upiti, aksiome, uslovi integriteta, heuristike, prioriteti. Aksiomama se definišu virtuelne relacije. Upit nad virtuelnom relacijom se transformiše, korišćenjem pristupa pravila transformacija, u upit nad baznim relacijama. Odgovor se prvo traži iz definicionog dela sistema (npr. uslova integriteta), a ako se ne može dobiti u ograničenom vremenu, dobijeni upit se predaje sistemu za upravljanje bazom podataka, na procesiranje. Kao kod Reitera, i u sistemu DEDUCE aksiome su razdvojene od fakata. Aksiome se koriste da transformišu upit, a fakti da "izračunaju" upit. U sistemu se razmatra i problem optimizacije, posebno s obzirom na analizu neophodnih spajanja.

4.2.2. FAKTUELNO IZVODJENJE

U delu 4.2. naveden je primer odgovora na upit nad tekstom, koji je (odgovor) eksplicitno sadržan u tekstu, ali se do njega dolazi posredno, preko drugih informacija sadržanih eksplicitno i neposredno u tekstu. Podaci koje zadajemo (poznanice), informacije koje čine odgovor (nepoznanice) i te posredne informacije preko kojih se odgovor dobija (posrednice), semantički su povezani u tekstu. Ta semantička veza ima svoj analogon u formalnoj shemi i to su funkcionalne i višeznačne zavisnosti u modelu sa dve vrste null-vrednosti medju atributima pojedinih relacija u shemi, preko kojih se upit formuliše, a koji odgovaraju poznanicama, nepoznanicama i posrednicama. Ovo izvodjenje zvaću <u>faktuelno izvodjenje</u>.

Postupak koji će biti primenjen da bi se iz teksta dobio odgovor na upit, kada je odgovor eksplicitno ali posredno sadržan u tekstu (u odgovarajućoj relaciji virtuelne sheme, poznanice se nalaze u n-torci sa null-vrednostima na atributima nepoznanica, a

vrednosti nepoznanica nalaze se u drugoj n-torci sa null-vrednostima na atributima poznanica) je sledeći: upit postavljen nad atributima virtuelne baze podataka zameniće se nizom upita nad atributima koji su funkcionalno ili višeznačno zavisni; zatim će se na svaki od medjuupita tražiti neposredni odgovor iz teksta.

U daljem tekstu prvo ću izložiti algoritam dobijanja odgovora na upit uključujući i dekompoziciju upita, a zatim niz teorema koje će voditi dokazu da je odgovor na upit, dobijen algoritmom, precizan (tačan) onoliko koliko su precizna preslikavanja formalnog u tekstuelni upit i tekstuelnog upita u odgovor iz teksta, i potpun u meri u kojoj je odgovor u tekstu sadržan eksplicitno i prethodno pomenuta preslikavanja potpuna.

(i) ALGORITAM IZVODJENJA

Neka je relaciona shema R* definisana skupovima {R} - relacija, {A} - atributa, {D} - domena, {c} - konstanti, i funkcionalnim i višeznačnim zavisnostima, i neka je nad relacijom R(A, C, OST) (A, C, OST - disjunktni skupovi atributa) te sheme zadan upit

upit*: { range of e is R
       { retrieve (e.C) where e.A=a.

Tada sledeći algoritam daje odgovore na upit* koji su sadržani u činjenicama iz teksta (posredno ili neposredno) i za koje nije potrebno primeniti bilo kakva pravila, tj. odgovore odg'(upit*) (nalaženje posredno sadržane informacije odgovara svodjenju upita nad n-torkama sa null-vrednostima na upite nad drugim n-torkama bez null-vrednosti relevantnih atributa).

Algoritam A

1. preslikati upit* u leksičke kategorije teksta - u upitt* (preslikavanje f);

2. Naći odgovor(e) ako postoji(e) na osnovu upita upitt* (preslikavanje odgt);

3. ako je neki odgovor nadjen korakom 2, tada

3a. ako u relaciji R postoji FZ A --> C tada kraj;

4. za niz par po par disjunktnih skupova atributa B_1, B_2, ..., B_k relacije R, koji ne uzimaju null-vrednosti, i za koje u R[A, B_1, B_2, ..., B_k, C] važe višeznačne zavisnosti B_1 ->->B_2, B_2 ->-> B_3, ..., B_k ->-> C, za najmanje k>0 za koje korak 4. nije već uradjen, uraditi:

4a. zameniti upit* upitom upit**

upit**: range of e_1 is R
range of e_2 is R
⋮
range of e_{k+1} is R
retrieve (e.C) where $e_{k+1}.B_k=e_k.B_k$
and $e_k.B_{k-1}=e_{k-1}.B_{k-1}$
and ... and $e_2.B_1=e_1.B_1$
and $e_1.A=a$.

4b. Dekomponovati upit** u k+1 upita $upit**_1$, ekvivalentnih upitu upit**:

$upit**_1$: range of e_1 is R
retrieve ($e_1.B_1$) where $e_1.A=a$
($=c_1$)

$upit**_2$: range of e_2 is R
retrieve ($e_2.B_2$) where $e_2.B_1=c_1$
($=c_2$)

$upit**_{k+1}$: range of e_{k+1} is R
retrieve ($e_{k+1}.C$) where $e_{k+1}.B_k=c_k$

4c. Preslikati svaki od upita u 4b. u upit nad tekstom (f) i tražiti, redom, odgovor na svaki - ako postoji; odgovor na poslednji upit je traženi odgovor na polazni upit.

4d. Ponoviti korak 4. ako je moguće, inače kraj.

(ii) Teorema A (o Algoritmu A): Neka su p_f, p_{f1}-p_{f4}, p_{odgt} - preciznosti preslikavanja f, f_1-f_4, odgt - redom. Tada za svaki odgovor na upit*, dobijen algoritmom A, važi: P(odgovore odg(upit*)) = $(p_{f1}*p_{f2}*p_{f3}*p_{f4}*p_{odgt})^{k+1}$, gde je k+1 -broj

promenljivih upita upit** na koji je dobijen odgovor, tj. svaki odgovor dobijen algoritmom A je tačan do na preciznost preslikavanja odgt*f.

Napomena1: Može se pretpostaviti da je $p_{f1}=p_{f2}=p_{f3}$, jer su sva preslikavanja rezultat jedinstvenog ljudskog (tj. ekspertnog) kriterijuma. Za slučaj da su svi p_{f1} (=p) =1, svi odgovori dobijeni algoritmom A su tačni odgovori na upit*. Jedan potreban uslov za preciznost p=1 je da se svaka dva različita atributa virtuelne baze podataka preslikavaju u razne skupove leksičkih kategorija.

Dokaz teoreme A koristi rezultate sledeće dve teoreme:

(iii) Teorema 5: (o ekvivalentnom razlaganju upita nad skupovima atributa jedne relacije)

Neka su nad relacijom R(A, C, OST) (A, C, OST - kao u algoritmu A), za proizvoljni skup atributa $B \subseteq OST$ koji ne uzimaju null-vrednosti, i proizvoljan skup konstanti $a \subseteq D(A)$, postavljena sledeća dva upita:

```
upit*:  range of e is R
        retrieve (e.C) where e.A=a

upit**: range of e is R
        range of u is R
        retrieve (u.C) where u.B=e.B and e.A=a.
```

Važe sledeća tvrdjenja:

1) $odg(upit*) \subseteq odg(upit**)$, gde su odg(upit*), odg(upit**) - odgovori na upit*, upit**, redom;

2) dovoljan uslov za $odg(upit*) \supseteq odg(upit**)$ (tj. odg(upit*) = odg(upit**)) je važenje VZ B ->-> C u R[A,B,C];

3) ako FZ A -> C važi u R, tada je odgovor na upit* jedinstven.

Dokaz: Koristeći semantiku QUEL-a [75], upit*, upit** u modelu sa dve vrste null-vrednosti, mogu se interpretirati na sledeći način:

upit*: (a) restrikuj R prema izrazu $\mathcal{T}(e.A \doteq a) \in \{T, \omega\}$;

(b) projektuj na C;

(c) eliminiši duplikate.

upit**: (a') napravi Dekartov proizvod RxR;

(b') restrikuj prema istinosnoj funkciji (u.B=e.B i τ(e.A$\doteq$a) $\in$ {T, ω});

(c') projektuj na u.C;

(d') eliminiši duplikate.

Jezikom relacione algebre, upit* i upit** mogu se sada zapisati kao sledeći izrazi:

upit*: $R[A=_{T,\omega} a][C]$;

upit**: $(R[A=_{T,\omega} a][B] * R)[C]$ $(\equiv(R[A=_{T,\omega} a][B] * R[B,C])[C])$.

Kako atributi iz B ne uzimaju null-vrednosti,

1) iz $R[A,B,C] \subseteq R[A,B] * R[B,C]$ sledi:

$R[A,B,C][A=_{T,\omega} a] \subseteq (R[A,B] * R[B,C])[A=_{T,\omega} a]$

$= R[A=_{T,\omega} a][A,B] * R[B,C]$ (zamenom mesta restrikcije i spajanja i restrikcije i projekcije). Odatle,

$R[A,B,C][A=_{T,\omega} a][C] \subseteq (R[A=_{T,\omega} a][A,B] * R[B,C])[C]$

$= (R[A=_{T,\omega} a][B] * R)[C]$.

Kako je $R[A,B,C][A=_{T,\omega} a][C]=R[A=_{T,\omega} a][C]$, sledi da odg(upit*) $\subseteq$ odg(upit**).

2) Ako u R[A,B,C] važi VZ B ->->C, tada važi $\tau(R[A,B] *_T R[B,C] \doteq R[A,B,C]) \in \{T,\omega\}$ (teorema o karakterizaciji VZ), a s obzirom da atributi iz B ne uzimaju null-vrednosti, i jednakost R[A,B] * R[B,C] = R[A,B,C] umesto svih inkluzija u 1), te odg(upit*)=odg(upit**);

3) sledi trivijalno iz definicije FZ.

<u>Posledica 1</u>: Dovoljan uslov za važenje jednakosti odg(upit*) = odg(upit**) u tački 2) teoreme 5 je važenje VZ B ->-> C u celoj relaciji R, a posebno FZ B --> C, jer FZ B --> C povlači VZ B->->C u R, što povlači (jer B ne uzima null-vrednosti) R[B,C]*R[A,OST]=R a odatle (R[B,C] * R[A,OST])[A,B,C] (=R[B,C] * R[A,B]) = R[A,B,C].

Ovo je posebno značajno jer se odnos atributa u shemi obično zadaje zavisnostima u relaciji a ne u delovima relacije.

Napomena 2: Važenje VZ B ->-> C u R[A,B,C] (u teoremi 5) nije potreban uslov za važenje odg(upit*)=odg(upit**). Štaviše, aparat zavisnosti (FZ i VZ) izgleda nemoćan po tom pitanju. Utvrdjivanje potrebnog uslova obezbedilo bi iznalaženje svih mogućnosti ekvivalentne dekompozicije upita upit*.

(iv) Teorema 6: ("horizontalno" uopštenje teoreme 5 - uopštenje po broju posrednih skupova atributa)

Neka su nad relacijom $R(A,B_1,B_2,\ldots,B_k,C,OST)$ (A, B_1, ..., B_k,C,OST - kao u algoritmu A), za proizvoljni skup konstanti $a \subseteq D(A)$ postavljena sledeća dva upita:

```
upit*:  range of e is R
        retrieve (e.C) where e.A=a

upit**: range of e1 is R
        range of e2 is R
           :
        range of ek is R
        range of ek+1 is R
        retrieve (ek+1.C) ,where ek+1.Bk=ek.Bk and
                                 ek.Bk-1=ek-1.Bk-1 and ... and
                                 e2.B1=e1.B1 and e1.A=a.
```

Važe sledeća tvrdjenja:

1),3) - analogno odgovarajućim tačkama teoreme 5;

2) Dovoljan uslov za odg(upit*)⊇odg(upit**) (tj. odg(upit*) = odg(upit**)) je važenje niza VZ-i: $B_1 \to\!\!\to B_2$, $B_2 \to\!\!\to B_3$,..., $B_{k-1} \to\!\!\to B_k$, $B_k \to\!\!\to C$, u $R[A,B_1,B_2,\ldots,B_k,C]$.

Dokaz: 1) analogno tački 1) teoreme 5, osim što za upit** u semantici QUEL-a, u tački (a') stoji Dekartov proizvod od k+1 relacija R, a u (b') restrikcija po celoj istinosnoj formuli iz upit**; 2) ova tačka se dokazuje indukcijom:

Slučaj k=1 identičan je tački 2) teoreme 5.

Neka važi induktivna pretpostavka za k, tj. važi tvrdjenje kao u tački 2) ove teoreme.

Tvrdjenje se dokazuje za slučaj k+1, tj. za upit

upit**:' range of e_1 is R
range of e_2 is R
⋮
range of e_{k+2} is R
retrieve (e_{k+2}.C) where $e_{k+2}.B_{k+1} = e_{k+1}.B_{k+1}$ and
$e_{k+1}.B_k = e_k.B_k$ and ... and
$e_1.A = a$,

dovoljan uslov za važenje jednakosti odg(upit*) = odg(upit**') je važenje niza VZ $B_1 \to\to B_2$, $B_2 \to\to B_3$, ..., $B_k \to\to B_{k+1}$, $B_{k+1} \to\to C$, u $R[A, B_1, B_2, \ldots, B_k, B_{k+1}, C]$.

Semantika upita**' je sledeća:

upit**': (a) napraviti Dekartov proizvod RxRx...xR k+2 relacije R;

(b) restrikovati po istinosnoj formuli $e_{k+2}.B_{k+1}=e_{k+1}.B_{k+1}$ and $e_{k+1}.B_k=e_k.B_k$ and ... and $\tau(e_1.A \doteq a) \in \{T, \omega\}$;

(c) projektovati na e_{k+2}.C;

(d) eliminisati duplikate,

što je ekvivalentno semantici:

(a') napraviti Dekartov proizvod RxRx...xR k+1 relacije R;

(b') restrikovati po istinosnoj formuli $e_{k+1}.B_k=e_k.B_k$ and $e_k.B_{k-1}=e_{k-1}.B_{k-1}$ and ... and $\tau(e_1.A \doteq a) \in \{T, \omega\}$;

(c') projektovati na $e_{k+1}.B_{k+1}$;

(d') pomnožiti (Dekartovski) sa R;

(e') restrikovati po formuli $e_{k+2}.B_{k+1}=e_{k+1}.B_{k+1}$;

(f') projektovati na e_{k+2}.C;

(g') eliminisati duplikate,

što odgovara sledećim upitima:

(tačke (a')-(c')) upit+: range of e_1 is R
range of e_2 is R
⋮
range of e_{k+1} is R
retrieve ($e_{k+1}.B_{k+1}$) where $e_{k+1}.B_k=e_k.B_k$ and
$e_k.B_{k-1}=e_{k-1}.B_{k-1}$ and ... and
$e_1.A=a$,
(čiji je odgovor odg(upit+) = $\{c_1, c_2, \ldots, c_r\}$),

(tačke (d')-(g'))
upit*$_i$' (i=1,2,...,r): range of u is R
retrieve (u.C) where $u.B_{k+1} = c_i$.

Upit+ ekvivalentan je upitu upit+':

upit+': range of e is R
retrieve (e.B_{k+1}) where e.A=a

(zbog induktivne pretpostavke, a zbog važenja VZ-i $B_1 \to\!\!\to B_2$, ..., $B_k \to\!\!\to B_{k+1}$ u R[A,B_1,B_2,...,B_{k+1},C] pa i u R[A,B_1,B_2,...,B_{k+1}]). Dakle, za upit*' = upit+' U {upit*$_i$'}$_{i=1,2,...,r}$, važi

odg(upit**') = odg(upit*').

Obrnutim redosledom transformacije semantike upita*',

upit*' <=> (a") restrikovati R po istinosnoj formuli
τ(e.A≐a) ∈ {T, ω};
(b") projektovati na e.B_{k+1};
(c") pomnožiti (Dekartovski) sa R;
(d") restrikovati po istinosnoj formuli e.B_{k+1}=u.B_{k+1};
(e") projektovati na u.C;
(f") eliminisati duplikate

<=> (a"') napraviti Dekartov proizvod RxR;
(b"') restrikovati po formuli e.B_{k+1}=u.B_{k+1} and
τ(e.A≐a) ∈ {T, ω};
(c"') projektovati na u.C;
(d"') eliminisati duplikate

<=> range of e is R
upit*": range of u is R
retrieve (u.C) where u.B_{k+1}=e.B_{k+1} and e.A=a,

dobije se upit*" takav da je odg(upit*") = odg(upit*').

Zbog važenja VZ $B_{k+1} \to\!\!\to$ C u R[A,B_1,B_2,...,B_{k+1},C], pa i u R[A,B_{k+1},C], važi i upit*" <=> upit* (prema prvom koraku indukcije k=1), tj. odg(upit*") = odg(upit*). Iz tranzitivnosti jednakosti sledi da je odg(upit**') = odg(upit*) što je trebalo dokazati.

Dokaz teoreme A: Algoritam A sadrži dva koraka u kojima je moguće naći odgovor: koraci 3a. i 4d.

U slučaju koraka 3a, odgovor na upit je dobijen kao odgt(f(upit*)) (algoritamski koraci 1, 2). Kako je (pod realnom pretpostavkom da se sa verovatnoćom =0, direktnim traženjem odgovora iz teksta dobije odgovor koji je indirektno sadržan u tekstu) P(odgt(f(upit*)) ⊆ odg(upit*)) =

P(odgt(f(upit*)) ⊆ odg'(upit*))=p (=p_{+1}*p_{+2}*p_{+3}*p_{+4}*p_{odgt}),

to je tvrdjenje dokazano za k=1.

U slučaju koraka 4d, odgovor je dobijen kao odgovor na upit

upit**:
```
range of e1 is R
range of e2 is R
        :
        :
range of ek+1 is R
retrieve (ek+1.C) where ek+1.Bk=ek.Bk and
    ek.Bk-1=ek-1.Bk-1 and ... and e1.A=a,
```

koji je ekvivalentan (zbog semantike QUEL-a, kao u dokazu teoreme

6) nizu od k+1 upita

upit**1:
```
range of e1 is R
retrieve (e1.B1) where e1.A=a
          └─c1─┘
```

upit**2:
```
range of e2 is R
retrieve (e2.B2) where e2.B1=c1
          └─c2─┘
    :
    :
```

upit**k+1:
```
range of ek+1 is R
retrieve (ek+1.C) where ek+1.Bk=ck.
```

Važi odg(upit*) = odg(upit**) (iz tačaka 1),2) teorema 5,6).

Stoga, $P(\text{odgovor} \in \text{odg(upit*)}) = P(\text{odgt(f(upit**))} \subseteq \text{odg(upit*)})$

$= P(\text{odgt(f(upit**))} \subseteq \text{odg(upit**)}) =$

$\prod_{i=1,k+1} P(\text{odgt(f(upit**}_i\text{))} \subseteq \text{odg(upit**}_i\text{))}$

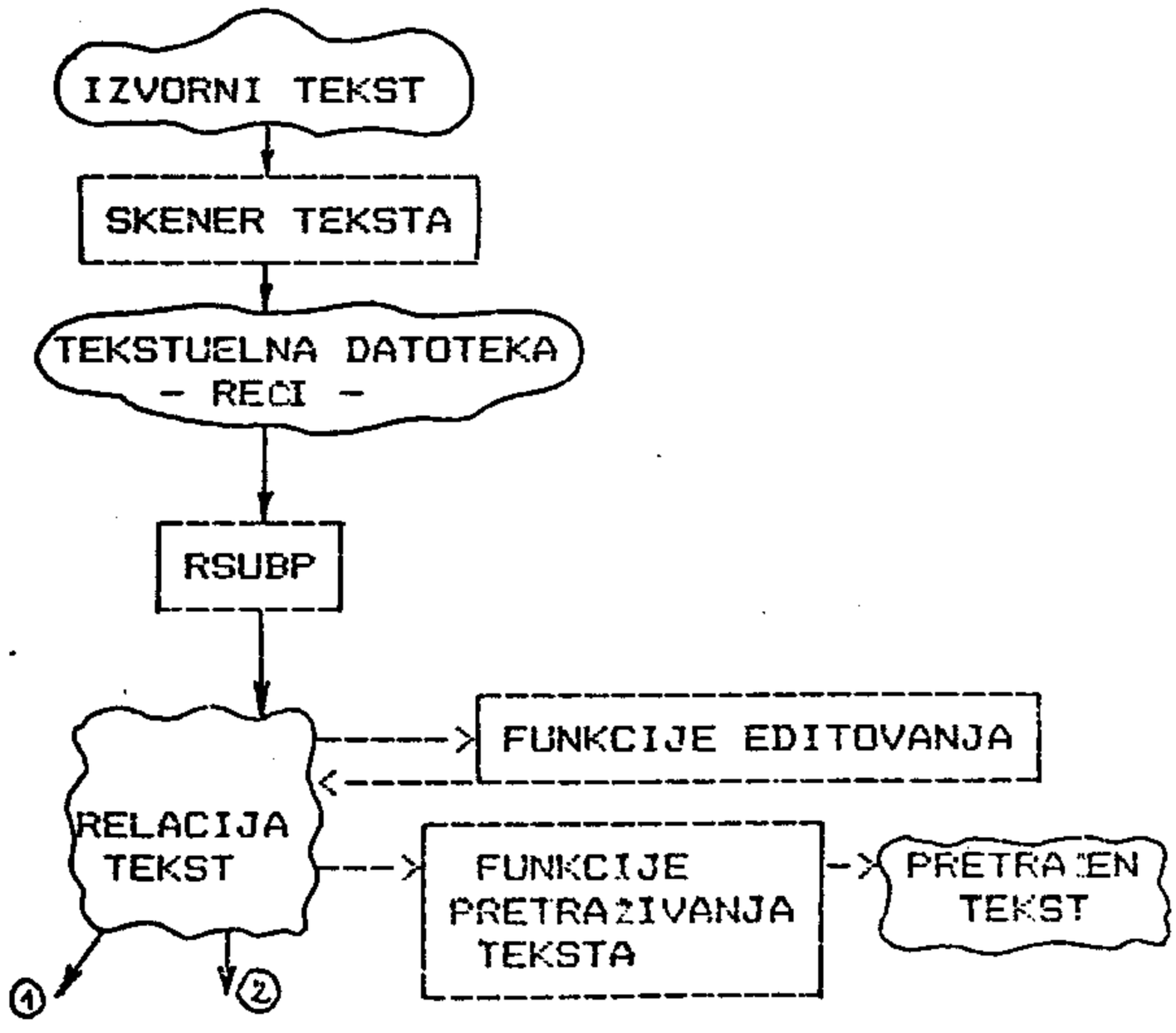
(jer su dogadjaji čije se verovatnoće odredjuju nezavisni)

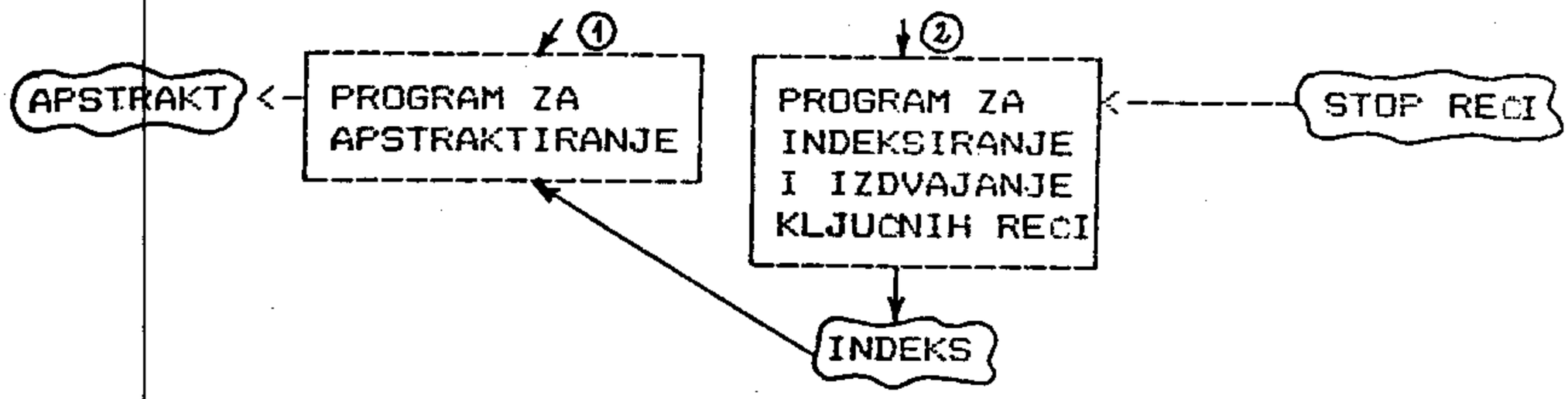
$= \prod_{i=1,k+1} P(\text{odgt(f(upit**}_i\text{))} \subseteq \text{odg'(upit**}_i\text{))}$ (kao u slučaju koraka 3a)

$= p^{k+1}$ $(= (p_{f1} * p_{f2} * p_{f3} * p_{f4} * p_{odgt})^{k+1})$, što je i trebalo dokazati.

5. KLASIČNE OPERACIJE NAD TEKSTUELNOM BAZOM PODATAKA

U ovoj glavi biće prikazani eksperimenti sa leksičkim operatorima trećeg nivoa tipa klasičnih operacija pretraživanja informacija nad tekstuelnom bazom podataka, i posebno sa automatskim indeksiranjem, izdvajanjem ključnih reči, apstraktiranjem, pretraživanjem, i editovanjem. Cilj ovih eksperimenata nije evaluacija velikog broja metoda razvijenih poslednjih decenija za automatsko izvršavanje ovih operacija, već testiranje pogodnosti relacionog sistema (posebno INGRES-a) i posebno leksičkog pristupa tretiranju teksta, za implementaciju ovih metoda. Kako sve metode koriste za svoju implementaciju ista svojstva teksta (reči) i relacionog sistema, u ovim eksperimentima su testirane najjednostavnije od tih metoda kao primeri koliko se efikasno relacioni sistem može upotrebiti za izvodjenje cele klase metoda. Svi programi napisani su u EQUEL-u. Globalni dijagram toka obrade teksta u ovim eksperimanetima predstavljen je na sledećoj slici:





5.1. LEKSICKA OBRADA I EKSPERIMENTI

Prvu grupu implementiranih operacija čine:

- automatsko indeksiranje
- izdvajanje ključnih reći
- automatsko apstraktiranje
- pretraživanje.

Ove operacije su u literaturi najčešće susretane operacije obrade teksta, i one su sve bazirane isključivo na rečima kao jedinicama teksta, tj. na osnovnim leksičkim operatorima, te je leksički pristup tretiranju teksta posebno pogodan za njihovu implementaciju.

Automatsko indeksiranje se sastoji od izdvajanja iz teksta (ili dodeljivanja tekstu) značajnih reći, bilo kako one definisane, u automatskom procesu.

Izdvajanje ključnih reći sastoji se u izvodjenju najrelevantnijih termina iz teksta, tj. termina koji mogu da predstavljaju deskriptore teksta (u zavisnosti od značenja pojma relevantnosti).

Automatsko apstraktiranje je automatski proces izdvajanja najznačajnijih rečenica iz datog teksta, gde je značajnost rečenice funkcija značajnosti reči u njoj..

Pretraživanje ima više značenja. Bibliografsko pretraživanje je proces u kom, na dati zahtev koji opisuje temu, sistem odgovara listom referenci (moguće uredjenom po nivou značajnosti). Pretraživanje teksta je pretraživanje specifičnih

rečenica iz datog teksta, koji zadovoljavaju postavljeni zahtev.

Postoji mnoštvo tehnika i metoda za automatsko indeksiranje, izdvajanje ključnih reči, apstraktiranje i pretraživanje, zasnovanih na načinu na koji su definisani pojmovi značajnosti reči, relevantnosti termina, najvećoj značajnosti rečenica, značajnosti teksta [55, 63, 64]. Istraživanja u ovoj oblasti odvijaju se poslednjih tridest godina. Glavni cilj tih istraživanja (s obzirom da indeksiranje i apstraktiranje utiču na pretraživanje), je maksimiziranje mera <u>odziva</u> i <u>preciznosti</u> pretraženih dokumenata. Te mere su definisane na sledeći način:

$$\text{odziv} = \frac{\text{broj pretraženih relevantnih dokumenata}}{\text{broj relevantnih dokumenata u kolekciji}}$$

$$\text{preciznost} = \frac{\text{broj pretraženih relevantnih dokumenata}}{\text{broj pretraženih dokumenata}}$$

Ukratko ću prikazati neke od tehnika i pristupa indeksiranju, apstraktiranju i pretraživanju.

Za <u>automatsko indeksiranje</u> postoje dve suštinski različite metode [64]:

- izvedeni indeksi
- dodeljeni indeksi

Izvedeni indeksi su reči iz teksta. Oni mogu da uključe sve reči iz teksta osim funkcionalnih ("stop") reči (članovi, predlozi, pomoćni glagoli, zamenice), ili reči iz naslova, ili citiranih referenci (pri čemu je svaka propraćena listom izvornih dokumenata koji je citiraju).

Izvedeni indeksi mogu biti i modifikovani u smislu da ne budu sve ne-funkcionalne reči iz teksta (ili naslova) - indeksne reči. U modifikovanom izvedenom indeksiranju postoje dva pristupa: sintaksno i semantičko indeksiranje. Kao sintaksni indeksi obično se biraju neke sintaksno karakteristične grupe reči - npr.

imenice, ili kombinacije pridev-imenica [3]. Semantički pristup čini pokušaj da odredi značajnost termina na osnovu značaja tog termina za dati tekst. Mera zanačaja termina za tekst je pretežno bazirana na statističkoj informaciji kao što je frekvenca termina u datom tekstu, relativna frekvenca termina u odnosu na celu kolekciju, frekvenca termina u odnosu na veličinu teksta [63], položaj prvog pojavljivanja termina u tekstu, frekvenca termina u relevantnim tekstovima [55], itd. Postoji i posebni metod - klasifikacija- posvećen proširenju indeksa terminima koji imaju tendenciju da se pojavljuju zajedno sa indeksnim terminima u datoj kolekciji tekstova [63].

Na osnovu pomenutih kriterijuma, termini mogu biti uključeni u indeks sa različitim težinama. U pretraživanju, težine indeksnih termina koji se pojavljuju u zahtevu koriste se za odredjivanje ukupne težine (značajnosti) pretraženog teksta.

Dodeljeni indeksi su reči koje se dodeljuju tekstu na sledeći način: prvo se analizira kolekcija dokumenata da bi se dobili indeksni termini za pojedine kategorije tekstova (statističkim metodima kao što je matrica korelacije [5]), zatim se novi tekst dodaje nekoj od tih kategorija, ako sadrži više od nekog broja njenih indeksnih termina, i tada svi indeksni termini te kategorije postaju indeksni termini tog novog teksta.

I izvedenim i dodeljenim indeksima mogu biti dodeljene težine na osnovu kriterijuma pomenutih za izvedene indekse.

U automatskom apstraktiranju, iz datog teksta se izdvajaju rečenice koje sadrže najveću koncentraciju značajnih reči, gde je značajnost reči njena tezina, kojigod od pomenutih kriterijuma za dodeljivanje težina bio izabran [8].

Pretraživanje teksta je definisano zadatim zahtevom; mogu biti izabrani samo oni paragrafi (rečenice) koji sadrže zadate

termine, ili paragrafima (rečenicama) mogu biti dodeljene težine na osnovu prethodnih kriterijuma i zadatog zahteva.

U bibliografskom pretraživanju, tekstovi se mogu izabrati buleanskim pretraživanjem - poklapanjem termina iz zahteva sa indeksima tekstova, ili dodeljivanjem težina tekstovima na osnovu težina indeksnih termina koji se pojavljuju u zahtevu ili u zahtevu proširenom terminima koji se pojavljuju zajedno, u datoj kolekciji tekstova, sa terminima iz zahteva.

Bibliografsko pretraživanje može se proširiti na kompleksno pretraživanje u kom se ne biraju samo tekstovi pretraženi prethodnim metodama, već i tekstovi koji sadrže izvesnu značajnu količinu indeksnih termina prethodno pretraženih tekstova [54], ili na interaktivno on-line pretraživanje u kom se prvo primenjuju neki od prethodno pomenutih metoda, a zatim se konsultuje korisnik; ako nije zadovoljan, primenjuje se neka druga shema dodeljivanja težina, i izdaje se lista pretraženih dokumenata, ako korisnik nije zadovoljan ni jednom od njih, traži se preformulacija zahteva (npr. korišćenjem dodeljenih indeksa, ako postoje, različitim kategorijama tekstova) [63].

U eksperimentu za primenu INGRES-a u leksičkoj obradi, prvi korak je program za skeniranje, opisan u glavi 2. Drugi korak je implementacija jednostavne tehnike automatskog indeksiranja, koja kao indeksne termine prihvata sve ne-funkcionalne reči, zanemarujući razliku izmedju malih i velikih slova. Relacija indeksnih termina takodje sadrži apsolutne frekvence tih termina. Neki različiti indeksni termini jednaki su u svojoj osnovi i značenju (jedan može biti u jednini, drugi u množini), i to može bitno da utiče na frekvencu pojedinih reči. Zato je kreirana druga relacija koja sadrži skraćene (do na pet slova) indeksne termine i njihove frekvence, i koja se koristi za dalju obradu (izdvajanje

ključnih reči, apstraktiranje, pretraživanje). S obzirom na postojanje eksplicitnih karakteristika reči ugradjenih u njenu leksičku reprezentaciju, kao frekvenca reči može se uzeti frekvenca pojavljivanja svih reči sa pripadnim korenom.

Treći korak je izdvajanje ključnih reči i fraza iz liste indeks termina. Primena frekvenci pojavljivanja reči, kao mere značajnosti, izuzetno je jednostavna u INGRES-u. U izvedenom eksperimentu, za ključne reči datog teksta biraju se termini u nekom intervalu apsolutne frekvence (može da ga zadaje i sam korisnik, a u eksperimentu je to otvoreni interval [10,-). Kao primer izdvajanja ključnih reči u INGRES-u, unet je tekst o semantičkom proširenju upitnog jezika QUEL, u relaciju:

exquel(rčn#, prčn#, rč#, duž, lex),

sa 2702 n-torke. QUEL-komanda

```
range of e is exquel
retrieve into kljreči(x.lex, frek=count(x.rčn# by x.lex))
   where count(x.rčn# by x.lex)>=10 and x.duž > 4
```

rezultuje sledećom relacijom:

kljreči relation

lex		duž	frek
geometric	(geometrijski)	9	29
point	(tačka)	5	22
example	(primer)	7	21
employee	(radnik)	8	20
operators	(operatori)	9	19
operations	(operacije)	10	17
relational	(relacioni)	10	16
group	(grupa)	5	14
types	(tipovi)	5	14
column	(kolona)	6	13
polygon	(poligon)	7	13
relation	(relacija)	8	13
salary	(zarada)	6	13
query	(upit)	5	12
language	(jezik)	8	11
between	(izmedju)	7	10
domains	(domeni)	7	10
geography	(geografija)	9	10
nation	(nacija)	6	10
objects	(objekti)	7	10

Mada ova lista uključuje neke reči koje nisu adekvatne ključne reči, ona takodje sadrži većinu kandidata za ključne reči. Cilj eksperimenta i nije da testira adekvatnost frekvence kao mere značajnosti ključnih reči, već da testira pretpostavku da INGRES dopušta da se takva shema jednostavno primeni.

Sledeći korak u eksperimentu je sačinjavanje apstrakta. Program za apstraktiranje izdvaja rečenice iz teksta koje sadrže najznačajnije ključne reči. Primenjena shema izdvajanja ključnih reči je suviše gruba da bi se upotrebila za generisanje apstrakta. Ipak, osnovna ideja može se ilustrovati pretraživanjem onih rečenica koje sadrže specifičnu ključnu frazu. Na primer, jednostavni EQUEL-program upotrebljen za izdvajanje rečenica koje sadrže frazu "relational operators" (relacioni operatori), daje sledeći "apstrakt" (ekstrakt):

"The way in which computation and aggregation are used allows the aritmetical operators on numerical domains to enrich the semantics of quel far beyond that provided by relational operators alone."

("Način na koji su izračunavanje i agregacija upotrebljeni dozvoljava aritmetičke operatore na numeričkom domenu da bi se obogatila semantika quel-a daleko iznad one koju obezbedjuju sami relacioni operatori").

Kako u vreme izvodjenja eksperimenta nije bila na raspolaganju kolekcija tekstova u mašinskom obliku, sa bibliografskim pretraživanjem nije ni eksperimentisano. Neke operacije pretraživanja teksta se, ipak, mogu lako implementirati u QUEL-u.

Sledeća dva primera ilustruju ove operacije i njihovu QUEL-formulaciju:

Primer 1: Naći sve rečenice koje sadrže frazu "null vrednosti"

```
range of e is text
range of s is text
range of t is text
retrieve (e.all) where e.rčn#=s.rčn# and s.rčn#=t.rčn# and
```

t.rć#=s.rć#+1 and s.lex="null" and t.lex="vrednosti"

Primer 2: Naći sve rečenice koje sadrže reč koja počinje sa "vred"

```
range of e is text
range of s is text
retrieve (e.all) where e.rćn#=s.rćn# and s.lex="vred*".
```

Zaključak do kog sam došla izvodeći eksperimente leksičke obrade je da je relacioni sistem i leksički pristup tretiranju teksta izuzetno pogodan za leksičku obradu teksta. Posebno korisne za leksičku obradu teksta su primarne i sekundarne strukture INGRES-a (primarni i sekundarni indeksi), kao i svi osnovni leksički operatori.

5.2. EDITOVANJE I EKSPERIMENTI

Implementacija efikasnih operacija editovanja leksikalizovanog teksta nije bio primarni cilj u izboru pristupa za podršku teksta u bazi podataka. Kako taj zahtev može da se postavi nad istom strukturom, izvršeni su eksperimenti sa gotovo svim operacijama standardnog editora. Zaključak je da, mada je moguće vršiti sve operacije i nad leksički organizovanim tekstom, ovaj pristup nije posebno pogodan za editovanje; potrebno je vršiti dosta preuredjivanja u tekstuelnoj relaciji, mada se gotovo sve može odložiti za kraj editor-sesije.

Pri editovanju je moguće adresirati bilo broj reda i broj reći unutar reda (ilustrovano primerima 3 i 4), bilo brojeve - deskriptore leksičke jedinice -broj paragrafa, rečenice, podrečenice i same reći unutar podrečenice (primer 5).

Primer 3: Zameniti prvo pojavljivanje reći "vrednosti" rečju "vrednost" u redovima 5 do 40.

```
range of e is text
replace e(lex="vrednost") where e.red#>=5 and e.red#<=40 and
```

```
e.rećred#=min(e.rećred# by e.red# where e.lex="vrednosti")
```

Primer 4: Premestiti redove 33 do 35 iza reda 47.

```
range of e is text
replace e(red#=e.red#/100 +4700-32) where e.red#>=3300 and
                                          e.red#<=3500.
```

(prethodno - u okviru skenera, brojevi svih redova su pomnoženi sa 100, što je potrebno očuvati i posle izvršenja ove operacije. Takodje je potrebno preurediti i brojeve paragrafa, rečenica, podrečenica, reči unutar podrečenica, narušene ovom operacijom. Ovo preuredjenje moguće je izvršiti posle operacije editovanja).

Primer 5: Premestiti treću rečenicu prvog paragrafa u treći paragraf posle druge rečenice.

```
range of e is text
replace e(par#=300, rćn#=201) where e.par#=100 and
                                    e.rćn#=300
```

(slićno preuredjenje kao u prethodnom primeru treba izvršiti nakon ove editor-operacije).

Primeri operacija editovanja izvršeni su na svim postojećim INGRES-strukturama. Uredjene relacije (i to multidimenzionalne) [39] izgledaju superiorne pre svega u pogledu formulacije upita.

6. ZAKLJUČAK

Predmet rada "Baze podataka i ekspertni sistemi u upravljanju tekstom" je proširenje relacionih sistema za upravljanje bazama podataka na upravljanje tekstom, i primena savremene tehnologije baza podataka i ekspertnih sistema na rešavanje problema automatske obrade teksta.

Osnovni cilj oko kojega su skoncentrisane hipoteze, istraživanja i rezultati rada je organizacija tekstuelnih podataka u relacionim bazama podataka na način koji je prostorno i vremenski efikasan i koji omogućuje realizaciju aplikacija kao što je složena semantička operacija automatskog izdvajanja informacije sadržane u tekstu.

Glavne hipoteze koje se u radu dokazuju ili su podržane eksperimentima su sledeće:

1) Definisanje leksičkog tipa podataka i predstavljanje teksta leksičkim tipom u bazama podataka omogućuje efikasno i potpuno čuvanje informacije sadržane u tekstu. Leksički tip podataka je celobrojna reprezentacija reči zajedno sa primitivnim leksičkim operacijama nad njima.

2) Leksički tip podataka je osnov za realizaciju hijerarhije leksičkih (jezičkih) operatora, počevši od primitivnih leksičkih operatora nad pojedinim leksičkim podacima (odrediti koren, prefiks, vrstu, oblik reči, izgradjiti zadati oblik zadate reči, itd), preko operatora drugog nivoa nad skupovima leksičkih podataka, kao što su operatori razrešavanja leksičke višeznačnosti ili odredjivanja referenata zamenica, do složenih operatora nad tekstovima kao skupovima leksičkih podataka, kao što su automatsko indeksiranje, apstraktiranje, pretraživanje teksta, i najzad, visoko semantički operator automatskog izdvajanja informacija iz

teksta.

3) Dok su rezultati primitivnih leksičkih operatora izvedivi iz samih leksičkih podataka na koje se primenjuju, za realizaciju operatora sledećih nivoa neophodni su složeniji koncepti. Tako, efikasan način za realizaciju operatora drugog nivoa (razrešavanje višeznačnosti i odredjivanje referenata zamenica) predstavljaju ekspertni sistemi, dok se za realizaciju složenog operatora izdvajanja informacija iz tekstą uspešno primenjuju koncepti sheme relacione baze podataka kao filtra nad sadržajem teksta, i teorija logičkog projektovanja relacione baze podataka.

Rad je podeljen u pet glava. U prvoj - uvodnoj - glavi prikazan je predmet rada i mesto unutar oblasti kojoj pripada, kao i motivacija za pojedine hipoteze i vrsta dobijenih rezultata. Takodje su navedeni publikovani rezultati kao i srodni rezultati po temama koje se u radu obradjuju.

U drugoj glavi, "Upravljanje tekstom kao podacima u relacionoj bazi podataka", prvo je izložen koncept relacionog modela (strukturni, manipulativni i integritetni deo), zatim je definisan leksički tip podataka u relacionoj bazi podataka kao uredjeni par (X, L), gde je X - skup leksičkih podataka odredjene strukture, a L - skup primitivnih operatora nad leksičkim podacima (definicija 1 u delu 2.2.). Leksički podatak se definiše kao kompozicija vrednosti leksičkih operatora nad tim podatkom, te ovaj vid kodiranja reči predstavlja način podrške unapred zadatog skupa leksičkih operatora. U poslednjem delu druge glave (2.3.) izložen je postupak "leksikalizacije" teksta tj. predstavljanja teksta leksičkim tipom podataka. Za realizaciju ove aktivnosti koriste se tekstuelni skener (2.3.1.), rečnik specifične strukture (sa elementima kao što su koren, prefiks, završetak, semantičko svojstvo) (2.3.2.), skup morfoloških pravila (2.3.3.) i operator

razrešavanja leksičke višeznačnosti (definisan i implementiran u trećoj glavi). Rečnik specifične strukture kao i skup morfoloških pravila predstavljaju se relacionim modelom baza podataka kao modelom baze znanja.

Direktna podrška leksičkog tipa podataka kao domena atributa, od strane relacionih sistema za upravljanje bazama podataka, predložio je E. Wong u radu [74]. Pod leksičkim tipom podataka tu se podrazumeva (bilo kakva) reprezentacija teksta sa rečju kao osnovnom leksičkom i semantičkom jedinicom. Alternativni pristupi tekstuelnim bazama podataka (slobodno segmentirani tekst sa proizvoljnom niskom simbola ili redom kao osnovnom jedinicom) predloženi su u IBM-ovoj laboratoriji [27], tj. u radovima M. Stonebraker-a i saradnika [39, 66].

Koncepti rečnika i morfološkog znanja (njihova specifična struktura, organizacija i sadržaj), koji se u ovom radu primenjuju pri predstavljanju teksta leksičkim tipom podataka, razmatraju se u kontekstu gotovo svih sistema za procesiranje prirodno - jezičkog teksta. Za reprezentaciju znanja sadržanog u rečniku specifične strukture i skupu morfoloških pravila, osim relacionog modela baze znanja , kao u ovom radu, u literaturi se koriste i druge strukture i drugi oblici reprezentacije znanja za razne jezike - semantičke mreže u sprezi sa proceduralnom logikom (R. Simmons u [62]), ili u sprezi sa relacionom bazom podataka i algoritamski rešenom morfologijom (J. Mylopoulos u sistemu TORUS na Univerzitetu u Torontu [41]; grupišuće LISP-funkcije (G. Hendrix u sistemu LADDER na Stenfordskom istraživačkom institutu [26]); konačni automati kao reprezentacija heurističkih morfoloških pravila nad standardnim rečničkim ulazima (H. Jappinen, u sistemu za komunikaciju sa bazama podataka na finskom jeziku [29]), tj. nad rečnikom bogatije strukture, snabdevenim raznim informacijama

o morfološkim i fonološkim karakteristikama reči (K. Koskenniemi u morfološkom analizatoru za finski jezik [34]); dvodelni, leksičko - semantički rečnik sa uključenim modelima promena (R. Camino u sistemu PARNAX za italijanski jezik [13]); rečnik obogaćen relacijama sinonimije, homonimije, podredjenosti i nadredjenosti uz algoritamsko rešenje morfološkog generisanja oblika reči iz rečnika [22, 57]). Slična vrsta morfološkog generatora za srpskohrvatski jezik opisana je u radu D. Vitasa [70]. Na uključenje atributa semantičkog svojstva reči u strukturu rečnika u ovom radu, posebno je uticao odgovarajući koncept semantičkog svojstva R. Simmons-a [62].

U trećoj glavi, "Konteksno-zavisna informacija u leksičkom tipu podataka: primena ekspertnih sistema", izlažu se koncepti i realizacije leksičkih operatora drugog nivoa, primenom ekspertnih sistema - operatora razrešavanja višeznačnosti (RAMB) i operatora odredjivanja referenata zamenica (PRONR). Posebno se, saglasnošću problema i mehanizma ekspertnih sistema, motiviše primena ekspertnih sistema na realizaciju oba operatora analizom ograničenog konteksta. U delu (3.1.) prvo su izloženi osnovni pojmovi o ekspertnim sistemima - definicija, tipovi, komponente, principi arhitekture, faze i sredstva izgradnje, modeli približnog rezonovanja. U delu (3.2.) opisuju se struktura i funkcija leksičkog operatora za razrešavanje višeznačnosti - RAMB, tj. arhitektura ekspertnog sistema za realizaciju tog operatora - baza pravila oblika (antecedens, konsekvens, težina) i odgovarajuća kontrolna strategija. Posebna pažnja u delu (3.2.) posvećuje se proceduralnom delu operatora RAMB (tačka 3.2.1.), tj. rekurzivnom algoritmu primene pravila iz baze znanja, kojim se izvode jednoznačne leksičke reprezentacije niza zavisno-višeznačnih reči (višeznačnih reči u čijem razrešavanju učestvuju druge višeznačne

reči). U tački (3.2.2.) dokazuju se svojstva operatora RAMB tj. karakteristike rešenja dobijenih algoritmom RAMB. Teorema 1 utvrdjuje korektnost svakog pojedinačnog elementa niza rešenja $(c_1, c_2, \ldots, c_k)$, dobijenog operatorom RAMB sa odgovarajućim težinama $(t_1, t_2, \ldots, t_k)$, tj. tvrdi da je rezultat operatora RAMB nad rečju N_i, pri fiksiranim vrednostima $(c_1, \ldots, c_{i-1}, c_{i+1}, \ldots, c_k)$ i pripadnim težinama $(t_1, \ldots, t_{i-1}, t_{i+1}, \ldots, t_k)$, odredjen sa maksimalnom težinom koju dozvoljavaju postojeća pravila. Definicija 2 definiše optimalno rešenje $(c_1, c_2, \ldots, c_k)$ niza zavisno-višeznačnih reči kao rešenje čiji se svaki element c_i, pri fiksiranim vrednostima $(c_1, \ldots, c_{i-1}, c_{i+1}, \ldots, c_k)$ i postojećim pravilima, dobija sa težinom 1. Pod uslovom da takvo rešenje postoji (daju se primeri kada ono postoji i kada ne postoji), teorema 2 tvrdi da ga algoritam RAMB pronalazi.

U delu (3.3.) opisuju se struktura i funkcija operatora odredjivanja referenata zamenica - PRONR, tj. arhitektura odgovarajućeg ekspertnog sistema. Pravilima oblika (antecedens, konsekvens, težina), definiše se relacija "neposredno se levo (desno) odnosi" - zamenica na objekat, tj. pravilima se odredjuju leksičke jedinice - objekti (leksičke slike imeničkih fraza ili zamenica iz iste leksičke klase ekvivalencije), najbliži zamenici, na odgovarajućem nivou težine, s leve tj. desne strane. Procedura izvodjenja, na osnovu ovih "neposrednih referenata", odredjuje jedinstveni niz leksičkih jedinica - sliku imeničke fraze, originala - rezultat operatora PRONR. Rad procedure izvodjenja može se opisati definicijom 3 i teoremama 3 i 4 tj. pravilom diskriminacije. Definicijom 3 definiše se relacija "desno (levo) se odnosi" kao tranzitivno pokrivanje relacija definisanih pravilima, i relacija "završno desno (levo) se odnosi" kao odgovarajuća relacija u kojoj je objekat - imenička fraza.

Teorema 3 tvrdi da je relacija "desno (levo) se odnosi" - relacija parcijalnog uredjenja, a teorema 4 da je relacija definisana pravilima - jedinstvena, što zajedno daje posledicu da prethodno definisana relacija parcijalnog uredjenja definiše jedinstveni levi tj. desni lanac referenata. U slučaju da je na kraju takvog lanca - imenička fraza, ona je i jedan (od najviše dva) kandidat za original. Diskriminaciono pravilo sada favorizuje "levi" original (saglasno sa uočenom frekventnošću pojavljivanja zamenice i originala u poretku (original, zamenica)).

Na kraju delova (3.2.), (3.3.), u tačkama (3.2.3.) tj. (3.3.1.) prikazani su neki detalji implementacije i eksperimentalni rezultati operatora RAMB tj. PRONR. Oba operatora testirana su nad bazama pravila za engleski i srpskohrvatski jezik. Pravila se izražavaju u terminima primitivnih leksičkih operatora nad zadatim kontekstom, i mada predstavljaju zdravorazumske heuristike a ne rezultat rada lingviste, eksperimentalni rezultati su veoma ohrabrujući.

Mada ni za jedan od problema koji se rešavaju operatorima RAMB, PRONR, ne postoji celovito rešenje, u literaturi su objavljene metode kojima se ti problemi delimično ali korisno rešavaju. L. Birnbaum daje, u radu [4], pregled metoda za rešavanje problema leksičke višeznačnosti (ATN-sintaksni analizatori, ograničeno posmatranje unapred, selekciona ograničenja, skriptalni leksikoni) i kao rešenje sugeriše integralni pristup jezičkoj analizi, uz korišćenje elemenata sistema zasnovanog na pravilima. Ovi elementi, kao i, u osnovi, metodi ograničenog posmatranja unapred i selekcionih ograničenja, koriste se i u ovoj tezi. Pri odredjivanju referenata zamenica, kriterijumi koji se koriste su slaganje oblika zamenice i referenta (R. Simmons, [60]), slaganje u reprezentaciji domena

teksta generisanoj gramatikom (W. Frey, [18]), korespondencija zamenice i objekta iz odgovarajuće rubrike predefinisanog rečeničnog obrasca (G. Hendrix, [26, 71]). Neki elementi Simmons-ovog rešenja koriste se i u ovom radu. Sa lingvističkog stanovišta, za ovaj problem značajni su radovi C. Sidner [59] i A. Kibrik [33]. U prvom se daje pregled istraživačkih pravaca u oblasti i definišu lingvistički zasnovana pravila o jezičkoj ulozi zamenice u rečenici, koja bi se mogla, uz adekvatnu formulaciju, uključiti u bazu pravila operatora PRONR. U drugom se prati jedan od pomenutih istraživačkih pravaca - definišu se sintaksna i semantička ograničenja koima se eliminišu neki od potencijalnih referenata; ova ograničenja bi takodje mogla da nadju svoju interpretaciju u okviru sistema PRONR.

Glava 4, "Primene tekstuelnih baza podataka", odnosi se na realizaciju leksičkog operatora trećeg nivoa - izdvajanja informacije iz teksta. Ovaj operator predstavlja najvažniju, s aspekta ovog rada, aplikaciju tekstuelnih baza podataka. Njime se preslikava par skupova leksičkog tipa (tekst, skup upita), u skup leksičkog tipa - skup fakata iz teksta. U realizaciji operatora učestvuje teorija logičkog modeliranja relacionih baza podataka. U delu (4.1.) definiše se sam operator i opisuju njegove dve glavne komponente - virtuelna relaciona baza podataka koja odgovara relevantnim aspektima teksta (i predstavlja relacioni model znanja iz teksta), i samo izdvajanje informacija kao odgovora na upit nad virtuelnom relacionom bazom (preslikavanje upita u mehanizme pretraživanja teksta i algoritam nalaženja odgovora). Definišu se i karakteristike komponenti koje utiču na preciznost i potpunost rezultata operatora - direktnog odgovora na upit, sadržanog u jednoj rečenici sa kvalifikacijom upita, kao i na efikasnost dobijanja rezultata. U tački (4.1.1.) prikazan je eksperimentalni

sistem koji se odnosi na specifični skup tekstova - biografija.

Deo (4.2.) odnosi se na realizaciju onog dela operatora izdvajanja informacije iz teksta, kad entitet iz virtuelne relacione baze ne odgovara jednoj rečenici teksta, tj. kada je odgovarajuća virtuelna relaciona baza - relaciona baza sa null-vrednostima. Za dobijanje informacije u tom slučaju koristi se teorija logičkih zavisnosti u relacionom modelu sa null-vrednostima i na njoj zasnovana dekompozicija upita, a sam postupak naziva se faktuelno izvodjenje. Pošto je u tački (4.2.1.) opisan relacioni model baza podataka sa dve vrste null-vrednosti, u tački (4.2.2.) dat je algoritam faktuelnog izvodjenja i dokazano da su svi odgovori izvedeni tim algoritmom - tačni, do na preciznost (definisanu u 4.1.) preslikavanja virtuelne relacione sheme u jezičke kategorije teksta, i ovih kategorija u odgovor na upit, dakle, do na preciznost direktnog odgovora (teoreme 5,6,A).

Osim ovog, virtuelno-relacionog modela, za reprezentaciju znanja iz teksta i dobijanje odgovora na upit upotrebljavaju se razni drugi modeli, kao, npr. semantičke mreže (R. Simmons, D.Chester [62]), hijerarhijska memorija (P. Thorndyke [73]), reprezentacija značenja konceptualne zavisnosti (Riesbeck [73]), skript (R. Wilensky, R. Schank [73]) (svi opisani detaljnije u delu 4.1.). U delu 4.1. navedeni su i sistemi za "razumevanje" teksta tj. za izdvajanje informacije iz teksta, projektovani prema opisanim modelima (RESEARCHER [36], sistem medicinskih izveštaja [20], TIBAQ [22,57], itd.).

Osim faktuelnog, drugi vid izvodjenja iz znanja, bez obzira na reprezentaciju, je deduktivno izvodjenje. Ovim izvodjenjem bave se, npr. Wong u [75], Kellog i Travis (sistem DADM), Grant i Minker (sistem MRPPS), Reiter, Chang (sistem DEDUCE), u [19] (detaljni opis u tački 4.2.1.).

Najzad, u glavi 5, "Klasične operacije nad tekstuelnom bazom podataka", prikazane su teorijske osnove i eksperimenti sa leksičkim operatorima trećeg nivoa koji odgovaraju klasičnim operacijama nad tekstom – automatskom indeksiranju, apstraktiranju, pretraživanju (5.1.) i editovanju (5.2.). Cilj eksperimentisanja je ispitivanje mogućnosti i efikasnosti realizacije ovih operatora nad leksički organizovanim tekstom, a ne istraživanja u samim oblastima klasičnih operacija. Eksperimenti pokazuju veliku primerenost leksički organizovanog teksta – leksičkim operacijama indeksiranja i pretraživanja.

Dalja istraživanja koja su u fokusu interesovanja autora odnosiće se, pre svega, na kompleksnije aspekte operatora izdvajanja informacija iz teksta, kao najzanimljivije aplikacije tekstuelnih baza podataka, kako sa teorijskog tako i sa praktičnog stanovišta. Ova istraživanja uključiće definisanje preciznijeg metoda za dobijanje direktnog odgovora na upit nad tekstuelnom bazom (dalje mogućnosti primene relacionog modela baze znanja i alternativnih pristupa) i metoda deduktivnog izvodjenja za dobijanje odgovora implicitno sadržanog u tekstu, kao i elemente implementacije ovog operatora. U domenu implementacije, obratiće se pažnja i implementiranju konstruisanih operatora drugog nivoa (razrešavanja višeznačnosti i odredjivanja referenata zamenica) za srpskohrvatski jezik, a posebno verifikaciji odgovarajućih baza znanja od strane eksperata.

LITERATURA:

[1] Alagić,S. Relacione baze podataka, "Svjetlost", Sarajevo, 1984;

[2] Alagić,S. Relational Database Technology, Texts and Monographs in Comp. Sci., Springer - Verlag, 1986;

[3] Baxedale,P.B. An Empirical Model for Computer Indexing, in Machine Indexing, 1962, pp.207-218;

[4] Birnbaum,L. Lexical Ambiguity as a Touchstone for Theories of Language Analysis, in Proc. of the 9th Int. Joint Conf. on AI, California, 1985, pp. 815-820;

[5] Borko,H, Bernick,M.D. Toward the Establishment of a Computer Based Classification System for ScientificDocumentation, Rept. No. TM-1763, System Development Corp, Santa Monica, Cal, Feb. 1964,p.47;

[6] Brownstein,M. Managing Information Intelligently, Hardcopy, Nov.1985, pp.139-141;

[7] Byrd,R. Word Formation in Natural Language Processing Systems, in Proc. of the 8th Int. Joint Conf. on AI, West Germany, (1983), vol.2, pp. 704-706;

[8] Carrol,J.M. Content Analysis as a Word-Processing Option, in Proc. of the 4th Int. Conf. on Information Storage and Retrieval, California, 1981, ACMSIGIR vol.XVI, no.1, 1981, pp.126-131;

[9] Ceri,S. et al. Interfacing Relational Databases and Prolog Efficiently, in Proc. of the 1st Int. Conf. on Expert Database Systems, S. Carolina, (1986), pp.141-153;

[10] Codd,E.F. A Relational Model of Data for Large Shared Data Banks, CACM 13(6) 1970, pp.377-387;

[11] Codd,E.F. Relational Completness of Data Base Sublanguages,

Courant Comp.Sc. Symp. 7, Data Base Systems, N.Y.City, 1977, pp.65-98;

[12] Codd,E.F. Extending the relational Model to Capture More Meaning, ACMTODS, Vol.4, No.4, 1979, pp.397-434;

[13] Comino,R. et al. Understanding Natural Language through Parallel Processing of Syntactic and Semantic Knowledge: An Application to Data Base Query, in Proc. of the 8th Int. Joint Conf. on AI, West Germany, 1983, pp.663-667;

[14] Dahl,V. Quantification in a Three-Valued Logic for Natural Language Question-Answering Systems, in Proc. of the 6th Int. Joint Conf. on AI, (1979), pp.182-187;

[15] Danlos,L. Some Issues in Generation from a Semantic Representation, in Proc. of the 8th Int. Joint Conf. on AI, West Germany, (1983), vol.2, pp.606-609;

[16] Date,J.C. An Introduction to Data Base Systems, Fourth Edition, Addison-Wesley Publ. Comp., 1986;

[17] Duda,R.O. et al. Subjective Bayesian Methods for Rule-Based Inference Systems, in Proc. of the AFIPS 76 National Comp. Conf, vol.45, pp.1075-1082;

[18] Frey,W. et al. Automatic Construction of a Knowledge Base by Analysing Texts in Natural Language, in Proc. of the 8th Int. Joint Conf. on AI, West Germany, (1983), vol.2, pp.727-729;

[19] Gaillaire,H., Minker,J. (eds.) Logic and Data Bases, Plenum Press, 1978;

[20] Grishman,R., Hirschman,L. Question Answering from Natural Language Medical Data Bases, AI 7(1978), pp.33-44;

[21] Guttag,J. Abstract Data Types and the Development of Data Structures, CACM, June 1977.

[22] Hajičova,E. et al. Computer Applications of Linguistics in Prague, Informatica 1/1982, pp.59-66;

[23] Harris,L.R. User-oriented Data Base Query with the ROBOT Natural Language Query System, in Proc. of the 3rd Int. Conf. on VLDB, Japan, 1977, pp]

[24] Hayes-Roth,F. et al.(eds.) Building Expert Systems, Addison-Wesley Publ. Comp., 1983;

[25] Held,G.D. et al. INGRES - A relational Data Base System, in Proc. of the National Comp. Conf., 1975, pp.409-416;

[26] Hendrix,G.G. et al. Developing a Natural Language Interface to Complex Data, ACMTODS, 3(2), June 1978, pp.105-147;

[27] IBM Systems and Product Guide, Seventh Edition, 1985;

[28] INGRES-VERSION 7 Reference Manual, ERL, Univ.of California, Berkeley, Memo.no. UCB/ERL M81/61, 1981;

[29] Jappinen,H. et al..Portable Database Interface for Finnish, SIGART Newsletter No.86, Oct.1983, pp.42-43;

[30] Karnigham,B.W., Ritchie,D.M. The C Programing Language, Prentice-Hall, 1978;

[31] Kaplan,S,J. Designing a Portable Natural Language Database Query System, ACMTODS 9(1), Mar.1984, pp.1-19;

[32] King,J. (ed.) Special Issues on AI and Database Research, ACM SIGART Newsletter No.86, Oct.1983, pp.32-72;

[33] Kibrik,A.E., Narin'jani,A.S. (red.) Modelirovanie jazykovoj dejatel'nosti v intelektual'nyh sistemah, Moskva, "Nauka", 1987;

[34] Koskenniemi,K. Two-Level Model for Morphological Analysis, in Proc. of the 9th Int. Joint Conf. on AI, California, (1985), pp.683-685;

[35] Kowalski,R. Logic for Problem Solving, AI series 7, North Holland, 1979;

[36] Lebowitz,M. RESEARCHER: An Experimental Intelligent Information System, in Proc. of the 9th Int. Joint Conf. on

AI, California, (1985), pp.858-862;

[37] Liebowitz,J. Useful Approach for Evaluating Expert Systems, Expert Systems, 3(2), 1986, pp. 86-96;

[38] Liskov,B., Zilles,S. Programing with Abstract Data Types, ACM SIGPLAN Notices, 1974;

[39] Lynn,N. Implementing Ordered Relations in the Relational Database System INGRES, Masters Report, EECS Dept., Univ.of California, Berkeley, 1982;

[40] Marek,W. Completness and Consistency in Knowledge Base Systems, in Proc. of the 1st Int. Conf. on Expert Database Systems, S. Carolina, (1986), pp.75-82;

[41] Mylopoulos,J.,Borgida,A. TORUS - A Natural Language Understanding System for Data Management, in Proc. of the 4th Int. Joint Conf. on AI, SUSR, 1975, pp.414-421;

[42] Ong,J. The Design and Implementation of Abstract Data Types in the Relational Database System INGRES, Masters Report, EECS Dept., Univ.of California, Berkeley, 1980;

[43] Pavlović,G. Jedan pristup relacionom modelu baza podataka sa dve vrste nula vrednosti, Magistarski rad, Univerzitet u Beogradu, 1981;

[44] Pavlović,G. Using a Relational Data Base System to Store Text, ERL, Univ.of California, Berkeley, Memo.no. UCB/ERL, M83/44, 1983;

[45] Pavlović-Lažetić,G. Lexical Organization and Factual Queries on Texts in Relational Databases, Kolokvijum "Gramatike u rečnicima", Beograd, 1983;

[46] Pavlović-Lažetić,G. Automatska obrada teksta u relacionim bazama podataka, Informatika, 17(1983)4, pp.181-186;

[47] Pavlović-Lažetić,G. Aspects of Natural Language Communication with Databases, u Zborniku sa 7. medjunarodnog simpozijuma

"Kompjuter na sveučilištu", Cavtat, 1985, pp.305.1-305.8;

[48] Pavlović-Lažetić,G. Factual Inference in Knowledge from text-Based Relational Database Systems, u Zborniku sa 7. medjunarodnog simpozijuma "Kompjuter na sveučilištu", Cavtat, 1985, pp.306.1-306.8;

[49] Pavlović-Lažetić,G., Wong,E. Managing Text as Data, in Proc. of the 12th Int. Conf. on VLDB, 1986, pp.111-116;

[50] Pavlović-Lažetić,G., Wong,E. Resolving Word Ambiguity and Pronoun Reference: Rule-Based Inference in Textual Databases, u pripremi;

[51] Plath,W.J. REQUEST: A Natural Language Question-Answering System, IBM Journal of Research and Development, 20(4), 1976, pp.326-335;

[52] Popović,Lj. Gramatički sistem i sintaksička homonimija, u Zborniku 3. naučnog skupa Računalniška obdelava jezikovih podatkov, Bled, 1985, pp.225-238;

[53] Proceedings of the Workshop on Data Abstraction, Databases and Conceptual Modelling, Colorado, 1980;

[54] Robertson,S.E. Term Frequency and Term Value, in Proc. of the 4th Int. Conf. on Information Storage and Retrieval, California, 1981, ACM SIGIR vol XVI, No.1, 1981, pp.22-29;

[55] Salton,G, Wu.H. The Measurement of Term Importance in Automatic Indexing, in Journal of the American Society for Information Science, May 1981, pp.175-186;

[56] Schmidt,J. Type Concepts for Database Definition, in Proc. Int. Conf. on Databases, Israel, 1978;

[57] Sgall,P. Natural Language Understanding and the perspective of Question Answering, in Proc. of COLING 82, Horecky(ed.), North Holland, 1982, pp.357-364;

[58] Shortliffe,E.H., Buchanan,B.G. A Model of Inexact Reasoning

in Medicine, Mathematical Biosciences 23, 1975, pp.351-379;

[59] Sidner,C. Focusing for Interpretation of Pronouns, American J. of Comp. Ling. 7(4), 1981, pp.217-231;

[60] Simmons,R.F. Rule-Based Computations on English, in Waterman,D.A., Hayes-Roth,F.(eds.):Pattern-Directed Inference Systems, Academic Press, 1978, pp.455-468;

[61] Simmons.R.F, Chester,D. Inferencing in Quantified Semantic Networks, in Proc. of the 5th Int. Joint Conf. on AI, 1979, pp.267-273;

[62] Simmons,R., Chester,D. Relating Sentences and Semantic Networks with Procedural Logic, CACM, 25(8), 1982, 527-547;

[63] Spark,J.K. Automatic Indexing, Journal of Documentation, 30, 1974, pp.393-432;

[64] Stevens, M.E. Automatic Indexing: A State-of-the-Art Report, Monograph, 91, Nat. Bureau of Standards, Washingtcn D.C (1965-1970);

[65] Stonebraker,M. Applications of Artificial Intelligence Techniques to Database Systems, ERL, Univ.of California, Berkeley, Memo.no. UCB/ERL M82/31;

[66] Stonebraker,M. et al. Document Processing in a Relational Data Base System, ERL, Univ.of California, Berkeley, Memo.no. UCB/ERL M82/32, 1982;

[67] Stonebraker,M. et al. Application of Abstract Data Types and Abstract Indices to CAD Data Bases, ERL, Univ.of California, Berkeley, Memo.no. UCB/ERL M83/3;

[68] UNIX USER's Reference Guide, 4.2 BSD, Univ.of California, Berkeley, 1984;

[69] Vassiliou, J. Functional Dependencies and Incomplete Information, A Panache of DBMS Ideas III (D. Tsichritzis, ed.) Tech. Report CSRG - 111, 1980, pp. 93-118;

[70] Vitas,D. Generisanje imenićkih oblika u srpskohrvatskom jeziku, Informatica 4(3), Ljubljana, 1980;

[71] Waltz,D. Natural Language Access to a Laroe Data Base: An Engineering Approach, in Proc. of the 4th Int. Joint Conf. on AI, Tbilisi, SUSR, 1975, pp.868-872;

[72] Waterman,D.A. How Do Expert Systems Differ from Conventional Programs?, Expert Systems, 3(1), 1986, pp.16-19;

[73] Waterman,D.A., Hayes-Roth,F. (eds.) Pattern-Directed Inference Systems, Academic Press, 1978;

[74] Wong,E. EXQUEL: A Semantic Extension to QUEL, ERL, Univ.of California, Berkeley, Memo.no. UCB/ERL M82/44, 1982;

[75] Wong,E. Deductive Inference in Relational Database Systems, Proposal for NSF Grant, 1983;

[76] Woods,W.A. Transition Network Grammar for Natural Language Analysis, CACM, 13(10), 1970, pp.591-606;

[77] Zloof,M.M. Query By Example, in Proc. AFIPS Nat. Comp. Conf., vol.44, 1975, pp.431-438 .

DODATAK 1: REZULTATI SKENIRANJA TEKSTA

IZVORNI TEKST:

.pp
TESLA, Nikola, naucynik i pronalazacy iz oblasti
elektrofizike i elektrotehnike,
rodjen je u Smiljanu, kraj Gospicxa, 10. jula 1856, a umro u Njujorku,
7. januara 1943. Sa izvesnim prekidima u sykolovanju zbog svog
krhkog zdravlja, Tesla je zavrsyio 1866. osnovnu sykolu u rodnom
nestu, a gimnaziju sa maturom 1875. u Karlovcu.

SKENIRANI TEKST:

),0,0,0,100,10;3,c,.pp
.00,100,100,100,200,10,5,w,TESLA
.00,100,100,200,200,20,0,p,\,
.00,100,200,100,200,30,6,w,Nikola
.00,100,200,200,200,40,0,p,\,
.00,100,300,100,200,50,8,w,naucynik
.00,100,300,200,200,60,1,w,i.
.00,100,300,300,200,70,11,w,pronalazacy
.00,100,300,400,200,80,2,w,iz
00,100,300,500,200,90,7,w,oblasti
00,100,300,600,300,10,13,w,elektrofizike
00,100,300,700,300,20,1,w,i
00,100,300,800,300,30,14,w,elektrotehnike
00,100,300,900,300,40,0,p,\,
00,100,400,100,400,10,6,w,rodjen
00,100,400,200,400,20,2,w,je
00,100,400,300,400,30,1,w,u
00,100,400,400,400,40,8,w,Smiljanu
00,100,400,500,400,50,0,p,\,
00,100,500,100,400,60,4,w,kraj
00,100,500,200,400,70,8,w,Gospicxa
00,100,500,300,400,80,0,p,\,
00,100,600,100,400,90,2,n,10
00,100,600,200,400,100,0,p,\.
00,100,600,300,400,110,4,w,jula
00,100,600,400,400,120,4,n,1856
00,100,600,500,400,130,0,p,\,
00,100,700,100,400,140,1,w,a
00,100,700,200,400,150,4,w,umro
00,100,700,300,400,160,1,w,u
00,100,700,400,400,170,8,w,Njujorku
00,100,700,500,400,180,0,p,\,
00,100,800,100,500,10,1,n,7
00,100,800,200,500,20,0,p,\.
00,100,800,300,500,30,7,w,januara
00,100,800,400,500,40,4,n,1943
00,100,800,500,500,50,0,p,\.
00,200,100,100,500,60,2,w,Sa
00,200,100,200,500,70,8,w,izvesnim
00,200,100,300,500,80,9,w,prekidima

IZVORNI TEKST:

pp
lbert Einstein was born in Ulm,
ermany,
n March 14, 1879.
he following year his family moved to Munich,
here Hermann Einstein,
is father,
nd Jakob Einstein, his uncle,
et up a small electrical plant and engineering works.
pp
t the behest of his mother,
instein also studied music,
nd, though he played exclusively for relaxation,
e became an accomplished violinist.

SKENIRANI TEKST:

```
,0,0,0,100,10,3,c,.pp
00,100,100,100,200,10,6,w,Albert
00,100,100,200,200,20,8,w,Einstein
00,100,100,300,200,30,3,w,was
00,100,100,400,200,40,4,w,born
00,100,100,500,200,50,2,w,in
00,100,100,600,200,60,3,w,Ulm
00,100,100,700,200,70,0,p,\.
00,100,200,100,300,10,7,w,Germany
00,100,200,200,300,20,0,p,\.
00,100,300,100,400,10,2,w,on
00,100,300,200,400,20,5,w,March
00,100,300,300,400,30,2,n,14
00,100,300,400,400,40,0,p,\.
00,100,400,100,400,50,4,n,1879
00,100,400,200,400,60,0,p,\.
00,200,100,100,500,10,3,w,The
00,200,100,200,500,20,9,w,following
00,200,100,300,500,30,4,w,year
00,200,100,400,500,40,3,w,his
00,200,100,500,500,50,6,w,family
00,200,100,600,500,60,5,w,moved
00,200,100,700,500,70,2,w,to
00,200,100,800,500,80,6,w,Munich
00,200,100,900,500,90,0,p,\.
00,200,200,100,600,10,5,w,where
00,200,200,200,600,20,7,w,Hermann
00,200,200,300,600,30,8,w,Einstein
00,200,200,400,600,40,0,p,\.
0,200,300,100,700,10,3,w,his
0,200,300,200,700,20,6,w,father
0,200,300,300,700,30,0,p,\.
0,200,400,100,800,10,3,w,and
0,200,400,200,800,20,5,w,Jakob
0,200,400,300,800,30,8,w,Einstein
0,200,400,400,800,40,0,p,\.
0,200,500,100,800,50,3,w,his
0,200,500,200,800,60,5,w,uncle
```

DODATAK 2: OSNOVNE RELACIJE I REZULTATI PREDSTAVLJANJA REČNIKA LEKSIČKIM TIPOM

RELACIJE SA ZAVRŠECIMA, PREFIKSIMA I SEMANTIČKIM SVOJSTVIMA I NJIHOVIM KODIRANIM VREDNOSTIMA ZA ENGLESKI JEZIK:

ending table

id	wd
21	age
25	al
29	an
33	ar
37	ble
41	ce
45	cy
49	de
53	ed
197	irreg
57	ee
61	en
65	er
69	ful
73	fulness
81	ic
85	ics
89	ing
93	ion
77	hood
97	ism
101	ist
105	ity
109	ive
113	ize
117	le
121	less
125	ment
129	ness
133	nt
137	ny
141	ology
145	or
149	our
153	ous
157	red
161	rn
165	ry
169	ship
173	some
177	ss
181	te
185	th
189	um
193	ure

pref table

id	wd
1	a
4	be
7	co
10	demi
12	dis
15	en
17	ex
20	ideo
22	im
24	in
26	inter
30	ir
0	mis
33	poly
35	post
37	pro
40	re
0	semi
43	sub
46	trans
49	un
32	mis
42	semi

feat table

id	wd
2	ABT
4	ACT
6	ANM
8	ASM
10	BEH
12	CMP
14	CTR
16	DST
18	EMT
20	EVT
22	FLD
24	FUT
26	HUM
28	INC
30	INS
32	LOC
34	MAC
36	MSR
38	NAC
40	NEM
42	NMA
44	NST
46	OBJ
48	ORD
50	PAB
52	PAC
54	PAR
56	PEM
58	PMA
60	POS
62	PRT
64	PSF
66	PST
68	PTT
70	QL
72	QU
74	RSN
76	SBT
78	SHW
80	SRC
82	ST
84	TIM
86	VEL

RELACIJA SA OBLICIMA RECI I NJIHOVIM KODIRANIM VREDNOSTIMA ZA

ENGLESKI JEZIK: desc_end table

wcl	form	descr	ends	offs
ad	c	11	0	1
ad	der	13	0	3
na		180	0	0
ad	p	10	0	0
ad	s	12	0	2
aj	c	21	0	1
aj	p	20	0	0
aj	posesp	24	0	2
aj	posess	23	0	1
aj	s	22	0	2
ar	def	36	0	0
ar	und	35	0	0
av	f_s_pr	171	8	0
av	ft_s_pt	175	11	0
av	ger	172	0	2
av	inf	170	0	0
av	oth_pr	174	10	0
av	oth_pt	176	12	0
av	pp	177	13	0
av	t_s_pr	173	9	0
cj		40	0	0
cm		60	0	0
ep	-1p	121	20	0
ep	-1s	120	20	0
ep	-2-	122	20	0
ep	-3p	126	20	0
ep	f3s	124	20	0
ep	m3s	123	20	0
ep	n3s	125	20	0
hp	--p	111	19	0
hp	--s	110	0	0
id	c	16	14	0
id	p	15	0	0
id	s	17	15	0
ij	c	31	14	0
ij	p	30	0	0
ij	s	32	15	0
in	pn	151	5	0
in	sn	150	0	0
ip	---1	133	0	0
ip	---4	134	16	0
ip	---s	135	17	0
iv	ger	167	0	0
iv	inf	165	0	0
iv	pp	169	7	0
iv	pr3	166	0	1
iv	pt	168	6	0
nb	ord	51	0	1
nb	reg	50	0	0
pp		70	0	0
qp	---	112	0	0
rn	ger	142	0	2
rn	gerp	143	0	3
rn	pn	141	0	1
rn	sn	140	0	0

2.3

RELACIJE SA ZAVRŠECIMA, PREFIKSIMA I SEMANTIČKIM SVOJSTVIMA I NJIHOVIM KODIRANIM VREDNOSTIMA ZA SRPSKOHRVATSKI JEZIK:

endings table

id	wd
301	a
401	acy
501	ak
601	an
701	anstvo
801	ar
901	ca
1001	cxa
1101	cxi
1201	en
1301	ev
1401	ija
1501	ik
1601	in
1701	iv
1801	ka
1901	ki
2001	lje
2101	n-
2201	n-ti
2301	na
2401	ni
2501	nja
2601	nje
2701	nji
2801	o
2901	om
3001	or
3101	orski
3201	ost
3301	ovan
3401	s-
3501	s-ti
3601	sytvo
3701	tet
3801	u
3901	ut
4001	vanje

prefs table

id	wd
2	auto
4	do
5	nedo
8	inter
11	iz
12	neiz
15	na
16	nena
19	naiz
20	nenaiz
23	naj
24	najne
25	ne
28	o
29	neo
32	ob
33	neob
36	od
37	neod
40	po
41	nepo
44	poli
45	nepoli
48	pri
49	nepri
52	pro
55	proto
58	re
59	nere
62	sa
63	nesa
66	su
67	nesu
70	tele
73	trans
74	netrans
77	u
78	neu
81	ultra
84	za
85	neza

feats table

id	wd
2	ABT
4	ACT
6	ANM
8	ASM
10	BEH
12	CMP
14	CTR
16	DST
18	EMT
20	EVT
22	FLD
24	FUT
26	HUM
28	INC
30	INS
31	INT
32	LOC
34	MAC
35	MOD
36	MSR
38	NAC
40	NEM
42	NMA
44	NST
46	OBJ
48	ORD
50	PAB
52	PAC
54	PAR
56	PEM
57	PHN
58	PMA
60	POS
62	PRT
64	PSF
66	PST
68	PTT
70	QL
72	QU
74	RSN
76	SBT
78	SHW
80	SRC
82	ST
84	TIM

RELACIJA SA OBLICIMA REČI I NJIHOVIM KODIRANIM VREDNOSTIMA ZA SRPSKOHRVATSKI JEZIK:

esc_ends table

wcl	form	descr	ends
im	ms1	1000	0
im	ms2	1001	0
im	ms3	1002	0
im	ms4	1003	0
im	ms6	1004	0
pi	mp1	1005	0
im	mp2	1006	0
im	mp3	1007	0
im	mp4	1008	0
im	mp6	1009	0
im	zs1	1010	0
im	zs2	1011	0
im	zs3	1012	0
im	zs4	1013	0
im	zs6	1014	0
im	zp1	1015	0
im	zp2	1016	0
im	zp3	1017	0
im	zp4	1018	0
im	zp6	1019	0
im	ss1	1020	0
im	ss2	1021	0
im	ss3	1022	0
im	ss4	1023	0
im	ss6	1024	0
im	sp1	1025	0
im	sp2	1026	0
im	sp3	1027	0
im	sp4	1028	0
im	sp6	1029	0
ɔg	inf	2000	0
ɔg	s-s1	2001	0
ɔg	s-s2	2002	0
ɔg	s-s3	2003	0
ɔg	s-p1	2004	0
ɔg	s-p2	2005	0
ɔg	s-p3	2006	0
ɔg	pms-	2007	0
ɔg	pmp-	2008	0
ɔg	pzs-	2009	0
ɔg	pzp-	2010	0
ɔg	pss-	2011	0
ɔg	psp-	2012	0
ɔg	b-s1	2013	0
ɔg	b-s2	2014	0
ɔg	b--3	2015	0
ɔg	b-p1	2016	0
ɔg	b-p2	2017	0
ɔg	ps	2018	0
ɔg	pp	2019	0
ıg	s-s1	2001	101
ıg	s-s2	2002	102

RELACIJA RECNIK ZA ENGLESKI JEZIK:

wd	cl	form	root	pref	ending	feat
I	rp		I			HUM
me	rp		I		?	HUM
my	sp		I		?	HUM
myself	ep		I		?	HUM
a	ar		a			OBJ
abdominal	aj		abdomen		al	PAR
abhor	rv		abhor			NMA
ability	rn	n	able		ity	PST
able	aj		able			PST
unable	aj		able	un		NST
abolish	rv		abolish			NMA
about	pp		about			OBJ
abroad	ad		abroad			LOC
abrupt	aj		abrupt			QL
absolute	aj		absolute			ST
academic	aj		academy		ic	SRC
academy	rn	n	academy			ASM
accept	rv		accept			PAC
accomplish	rv		accomplish			PAC
according	ad		accord		ing	PMA
accountabl	aj		account		ble	PST
acquiescen	aj		acquiesce		nt	PST
acquire	rv		acquire			POS
act	rv		act			ACT
action	rn	n	act		ion	ACT
active	aj		act		ive	BEH
reaction	rn	n	act	re	ion	ACT
transactio	rn	n	act	trans	ion	ACT
add	rv		add			PAC
addition	rv		add		ion	PAC
admit	rv		admit			PMA
advance	rn	n	advance			PAC
advance	rv		advance			PAC
advent	rn	n	advent			EVT
affair	rn	n	affair			ACT
after	ad		after			FUT
after	aj		after			FUT
after	cj		after			FUT
after	pp		after			FUT
against	pp		against			DST
age	rn	n	age			TIM
age	rv		age			TIM
agree	rv		agree			PMA
air	rn	n	air			SET
algorithm	rn	n	algorithm			OBJ
all	qp		all			QU
almost	ad		almost			MSR
along	ad		along			LOC
also	ad		also			INC
although	ad		although			CTR
always	ad		always			TIM
amaze	rv		amaze			ST
amicable	aj		amity		ble	BEH
among	pp		among			INC

id	desc
898530CC0	8026
898530C16	8126
898530C17	9326
898530C20	12026
117760CCC	3546
8780C25	2054
9360C00	16042
9880C97	14066
988CCC0	2066
9884900	2044
10350CCC	16042
10400C00	7046
10800C0C	1032
10830CC0	2070
10950C00	2082
1234CC73	2060
12340CCC	14008
12630C00	16052
12690CC0	16052
12700C81	1058
12710037	2066
14745025	2066
14750CC0	16060
15220C00	16004
15220C85	14004
15225C01	2010
15224085	14004
15224685	14004
16960C00	16052
1696CC85	16052
18160C00	16058
19420CC0	14052
1942CCC0	16052
19440C00	14020
25210CC0	14004
27260CC0	1024
27260CC0	2024
27260CC0	4024
27260CC0	7024
28590CC0	7016
2927CC00	14084
29270C00	16084
31030CCC	16058
3920e0CC0	14076
49760CC0	14046
50560CC0	11272
5C630CC0	1036
5C920CC0	1032
51500C00	1028
51610C00	1014
52010CCC	1084
53C00CC0	16082
54140037	2010
54970C00	7028

RELACIJA REČNIK VLASTITIH FRAZA ZA ENGLESKI JEZIK:

ds	wno	cl	fea
Transaction on Database Systems	5	rn	ASM
yne Computing Ltd.	3	rn	ASM
e Georges Lemaitre	3	rn	HUM
demie de Medicine	3	rn	ASM
demie des Sciences	3	rn	ASM
lf Hitler	2	rn	HUM
ert	1	rn	HUM
ert Einstein	2	rn	HUM
rica	1	rn	LOC
rican	1	aj	SRC
terdam	1	rn	LOC
Appeal to Reason	4	rn	OBJ
re-Louis Debierne	4	rn	HUM
ria Leverkuhn	2	rn	HUM
alen	1	rn	OBJ
alen der Physik	3	rn	OBJ
il	1	rn	TIM
b	1	rn	HUM
ociation for Computing Machinery	4	rn	ASM
ociation of Professional Engineers of Ontarioo	6	rn	ASM
ust	1	rn	TIM
tria	1	rn	LOC
gian	1	aj	SRC
gium	1	rn	LOC
keley	1	rn	LOC
lin	1	rn	LOC
n	1	rn	LOC
nstein	1	rn	HUM
n	2	rn	HUM
tain	1	rn	LOC
nia	1	rn	HUM
denbrook	1	rn	HUM
ifornia	1	rn	LOC
ifornia Institute of Technology	4	rn	ASM
ɔridge	1	rn	LOC
ada	1	rn	LOC
adian Information Processing Society	4	rn	ASM
ıth	1	rn	LOC
ar Koch	2	rn	HUM
torp	1	rn	HUM
lon	1	rn	LOC
im Weizmann	2	rn	HUM
ɔlotte	1	rn	HUM
ɔhov	1	rn	HUM
nittee of Intellectual Cooperation	4	rn	ASM
ɔuter Corporation of America	4	rn	ASM
ɔuter Graphics Group	3	rn	ASM
ɔord	1	rn	LOC
fidence Man	2	rn	OBJ
ɛnhagen	1	rn	LOC
ıell University	2	rn	ASM
ıcil of the League of Nations	6	rn	ASM
ie	1	rn	HUM
ie Foundation	2	rn	ASM
ɛ	1	rn	HUM

m	id	desc
	-8331508	14008
	-1391308	14008
	-3013326	14426
	-1811308	14008
	-1898308	14008
	-3245226	14426
	-4899126	14426
	-2028226	14426
	-5354132	14032
ess	-5355180	2380
	-5558132	14032
	-705446	14046
	-2043426	14426
	-4867226	14426
	-5882146	14046
	-1854346	14046
	-6755184	14084
	-7316126	14026
	-2484408	14008
	-6318608	14008
	-8629184	14084
	-8803132	14032
ess	-13928180	2380
	-13929132	14032
	-14032132	14032
	-14033132	14032
	-14034132	14032
	-14035126	14426
	-24604226	14426
	-20036132	14032
	-20134126	14626
	-21362126	14426
	-25025132	14032
	-27404408	14008
	-25033132	14032
	-25043132	14032
	-27414408	14008
	-25072132	14032
	-27825226	14426
	-25103126	14426
	-26334132	14032
	-31340226	14426
	-26964126	14626
	-27003126	14426
	-29092408	14008
	-25639408	14008
	-26727308	14008
	-29145132	14032
	-28411246	14046
	-29166132	14032
	-30754208	14008
	-29211608	14008
	-30946126	14426
	-26361208	14008
	-33473126	14426

RELACIJA RECNIK ZA SRPSKOHRVATSKI JEZIK:

	wcl	form	root	pref	ending	feat
	vz		a			CTR
	vz		a			INC
demija	pi	zs1	akademija			ASM
ivan	pp	ms1op	akcija		an	PST
uelan	pp	ms1op	aktuelnost		an	PRT
uelnost	pi	zs1	aktuelnost			PRT
	vz		ali			CTR
ernator	pi	ms1	alternacija		or	OBJ
ena	pi	zs1	antena			OBJ
enski	pp	ms1np	antena		ki	OBJ
rat	pi	ms1	aparat			OBJ
omobil	pi	ms1	automobil			OBJ
on	pi	ms1	avion			OBJ
anje	pi	ss1	baciti		nje	NAC
iti	pg	inf	baciti		s-	NAC
iti	pg	inf	baviti		n-	ST
ljenje	pi	ss1	baviti		nje	ST
ezyen	pp	ms1op	belezyiti		en	ST
ezyiti	pg	inf	belezyiti		n-	MAC
	pd	g	bez			CTR
lioteka	pi	zs1	biblioteka			ASM
obiografski	pp	ms1np	biografija	auto	ki	OBJ
bran	pp	ms1op	birati	iz	an	ST
brati	pg	inf	birati	iz	s-	PMA
a	mg	psp--	biti		?	ST
a	mg	pzs--	biti		?	ST
e	mg	pzp--	biti		?	ST
i	mg	pmp--	biti		?	ST
o	mg	pss--	biti		?	ST
	mg	pms--	biti		?	ST
i	mg	inf	biti		?	ST
i	pg	inf	biti			NAC
	mg	s-s3k	biti		?	ST
am	mg	s-s1d	biti		?	ST
i	mg	s-s2d	biti		?	ST
no	mg	s-p1d	biti		?	ST
ce	mg	s-p2d	biti		?	ST
ce	mg	s-s3d	biti		?	ST
i	mg	s-p3d	biti		?	ST
e	mg	s-s3n	biti		?	NST
am	mg	s-s1n	biti		?	ST
i	mg	s-s2n	biti		?	ST
no	mg	s-p1n	biti		?	ST
ce	mg	s-p2n	biti		?	ST
i	mg	s-p3n	biti		?	NST
	mg	s-s1k	biti		?	ST
	mg	s-s2k	biti		?	ST
	mg	s-p1k	biti		?	ST
	mg	s-p2k	biti		?	ST
	mg	s-p3k	biti		?	ST
u	pr	p	blizu			LOC
jacxen	pp	ms1op	bogat	o	en	PST
jatiti	pg	inf	bogat	o	s-ti	PST
ba	pi	zs1	borba			ACT
l	pi	ms1	brod			OBJ

	descr
11700000	900014
11700000	900028
46000000	101008
47000601	300066
48000601	300062
48000000	101062
50000000	900014
51003001	100046
61000000	101046
61001901	300146
68000000	100046
90000000	100046
92000000	100046
118002601	102038
118003401	200038
120002101	200082
120002601	102082
146001201	300082
146002101	200034
147000000	820014
159000000	101008
161021901	300146
162110601	300082
162113401	200058
163000125	252482
163000122	252182
163000123	252282
163000121	252082
163000124	252382
163000120	251982
163000101	250082
163000000	200038
163000109	250882
163000102	250182
163000105	250482
163000111	251082
163000114	251382
163000108	250782
163000117	251682
163000110	250944
163000104	250382
163000107	250682
163000113	251282
163000116	251582
163000119	251844
163000103	250282
163000106	250582
163000112	251182
163000115	251482
163000118	251782
172000000	400032
191281201	300066
191283501	200066
192000000	101004
203000000	100046

RELACIJA RECNIK VLASTITIH FRAZA ZA SRPSKOHRVATSKI JEZIK:

ds	nwo	root	ending	cl	feat	form
ere	1			pi	HUM	ms1
grad	1			pi	LOC	ms1
impesyta	1			pi	LOC	zs1
kago	1			pi	LOC	ms1
son	1			pi	HUM	ms1
ctrical World	2			pi	OBJ	ms1
leska	1			pi	LOC	zs1
opa	1			pi	LOC	zs1
adej	1			pi	HUM	ms1
ncuska	1			pi	LOC	zs1
picx	1			pi	LOC	ms1
c	1			pi	LOC	ms1
z	1			pi	HUM	ms1
lovac	1			pi	LOC	ms1
orado	1			pi	LOC	ms1
g Island	2			pi	LOC	ms1
svel	1			pi	HUM	ms1
ibor	1			pi	LOC	ms1
agara	1			pi	LOC	zs1
agarin	1	Nijagara	in	pp	LOC	ms1op
ola	1			pi	HUM	ms1
ola Tesla	2			pi	HUM	ms1
jork	1			pi	LOC	ms1
iz	1			pi	LOC	ms1
g	1			pi	LOC	ms1
ljan	1			pi	LOC	ms1
orado Springs	2			pi	LOC	ms1
azbur	1			pi	LOC	ms1
la	1			pi	HUM	ms1
lin	1	Tesla	in	pp	HUM	ms1op
Electrical Experiment	3			pi	OBJ	ms1
t	1			pi	HUM	ms1
tinghouse	1			pi	ASM	ms1
reb	1			pi	LOC	ms1

	descr
-611000000	100026
-150100000	100032
-221100000	101032
-321100000	100032
-481100000	100026
-563200000	100046
-526100000	101032
-562100000	101032
-584100000	100026
-627100000	101032
-711100000	100032
-718100000	100032
-770100000	100026
-968100000	100032
-978100000	100032
1019200000	100032
1092100000	100026
1093100000	100032
1228100000	101032
1228101601	300032
1229100000	100026
1262200000	100026
1286100000	100032
1462100000	100032
1470100000	100032
1777100000	100032
-299200000	100032
1824100000	100032
1914100000	100026
1914101601	300026
1913300000	100046
2049100000	100026
2082100000	100008
2145100000	100032

DODATAK 3: OSNOVNE RELACIJE I REZULTATI PREDSTAVLJANJA TEKSTA LEKSICKIM TIPOM

MORFOLOŠKA PRAVILA ZA ENGLESKI JEZIK:

ul table

ings	suppl	prewc	postwc	descr	offset
		rn	rn	pl	1
		rn	rn	pl	1
	y	rn	rn	pl	1
	y	rn	rn	pl	1
		rv	rv	pr3sing	1
		rv	rv	pr3sing	1
	y	rv	rv	pr3sing	1
	e	rv	rv	cont	2
		rv	rv	cont	2
s+ing		rv	rv	cont	2
	e	rv	rn	ger	2
		rv	rn	ger	2
s+ing		rv	rn	ger	2
		rv	rv	ps	3
		rv	rv	ps	3
	y	rv	rv	ps	3
s+ed		rv	rv	ps	3
		aj	ad		3
	y	aj	ad		3
		ad	ad	cmp	1
		aj	aj	cmp	1
		aj	aj	sup	2
		ad	ad	sup	2
		aj	aj	cmp	1
		ad	ad	cmp	1
		aj	aj	sup	2
		ad	ad	sup	2
	y	aj	aj	cmp	1
	y	ad	ad	cmp	1
t	y	aj	aj	sup	2
t	y	ad	ad	sup	2
		rn	aj	ps/sg	1
		rn	aj	ps/pl	2
		rn	aj	ps	1
	e	rn	aj	ps	1
	fe	rn	rn	pl	1
	y	nb	rn	pl	2
	e	aj	ad		3
y		aj	ad		3
		rn	aj	ps	1
s		rv	rn	ger/pl	3
s	e	rv	rn	ger/pl	3
		iv	rn	ger	2
s+ing		iv	rn	ger	2
s+ings		rv	rn	ger/pl	2
		in	aj	ps/sg	1
	e	iv	rn	ger	2
s+ing	e	iv	rn	ger	2
		iv	iv	ger	2
	e	iv	iv	ger	2
s+ing		iv	iv	ger	2

USLOVI VRSTE REČI ZA ENGLESKI JEZIK:

	envir	phrase	form	positi	feat	presen
1	sentence	ar		-1		+
2	sequence	nphr		1		+
3	sentence	vb		0		-
4	sentence	av		-1		+
5	sentence	vb		1		+
6	sentence	nphr		0		-
7	vphr	anyt		1		+
8	vphr	anyt		-1		+
9	sentence	dv		2		+
10	nphr	anyt		1		+
11	nphr	anyt		-1		+
12	sequence	vphr		-1		+
24	sequence	no		-1		+
14	sequence	vphr		0		+
15	sequence	nphr		-1		+
16	sequence	nphr		1		-
17	sequence	nphr		2		-
20	sequence	vphr		1		+
21	sequence	dj		-1		+
22	sentence	dv		-1		+
23	sentence	dv		1		+
34	sequence	prph		-1		+
35	sequence	rp		-1		+
26	sentence	ar		1		+
27	sequence	"be"		-2		+
36	sequence	ger		-1		+
29	sentence	ar		2		+
30	sentence	dj		2		+
39	sequence	no		2		+
32	sequence	pp		-1		+
33	sequence	nb		-1		+
41	sequence	"-"		1		+
42	sequence	sp		-1		+
72	sequence	dv		-2		+
45	sequence	"that"		1		+
46	sequence	"and"		-1		+
47	sequence	no		-2		+
49	sequence	dj		-2		+
51	sequence	"and"		1		+
52	sequence	no		2		+
54	sequence	dj		2		+
56	sequence	"be"		-1		+
57	sequence	anyt		1		+
58	sequence	dj		1		+
60	sequence	"be"		1		+
61	sequence	pp		1		+
62	sequence	"can"		-1		+
63	sequence	vb		-2		+
65	sequence	"do"		-1		+
66	sequence	"do"		-2		+
67	sequence	"not"		-1		+
69	sequence	"cannot"		-1		+
70	sequence	"can"		-2		+
71	sequence	"which"		-1		+
24	sequence	no		-1		+

PRAVILA VRSTE RECI ZA ENGLESKI JEZIK:

word	cond	type
18	33	no
20	34,24	vb
21	35	vb
22	36	no
70	12,2	dj
29	2	pp
1	1,2	dj
50	27,67,5	dv
56	56,5	dv
2	1	no
3	21,2	dj
4	21	no
5	22	dj
7	23	vb
8	26	vb
9	58	vb
10	45,29	vb
12	45,30	vb
14	32,2	dj
16	4	vb
17	33,2	dj
25	7	vb
26	8	vb
31	20	dv
34	2,3	vb
35	5,6	no
39	42,2	dj
40	42	no
53	62	vb
58	66,67	vb
59	69	vb
60	67,70	vb
19	4	dj
45	56	dj
46	58	dv
32	12	dv
11	45,29	dj
13	45,30	dj
23	10	no
24	11	no
33	2	dj
36	2,9	vb
38	39,41	no
41	46,47	no
42	46,49	dj
43	51,52	no
44	51,54	dj
47	60	no
54	46,63	vb
55	58	dj
6	22	vb
15	32	no
28	17	pn
27	14,15	no
30	2,15	cj

nv table

rulno	wgh
2	1.000
4	1.000
6	1.000
7	1.000
18	1.000
20	1.000
21	1.000
22	1.000
40	1.000
47	1.000
53	1.000
58	1.000
59	1.000
60	1.000
86	0.950
103	1.000
8	0.950
10	0.950
12	0.950
15	0.950
34	0.950
35	0.950
38	0.950
104	1.000
25	0.900
26	0.900
52	0.900
61	0.900
23	0.800
24	0.800
36	0.800
41	0.800
43	0.800
54	0.800
27	0.700
57	0.700
74	0.700
62	0.600
99	0.600

USLOVI SEMANTICKIH SVOJSTAVA:

	envir	phrase	form	pos	feat	pres
1	sequence			1	TIM	+
2	sequence			1	LOC	+
3	sequence	nphr		1	TIM	+
4	sequence	nphr		1	LOC	+
5	sequence	rv		-1		+
6	sequence	iv		-1		+
7	sequence	nphr		-1		+
8	sequence	anyt		-1		-
9	sequence	rv	past	-1		+
10	sequence	iv	part	-1		+
11	sequence			+	INS	+
12	sequence			+	OBJ	+
13	sequence	"perform"				+
14	sequence	"play"				+
15	sequence			-1	TIM	+
16	sequence			-1	LOC	+
17	sentence	rv	past	-2		+
18	sentence	iv	part	-2		+
19	sentence	dv		-1		+

PRAVILA SEMANTICKIH SVOJSTAVA ZA ENGLESKI JEZIK:

	word	cond	certainty	type
12	by	9	1.000	SRC
13	by	10	1.000	SRC
16	drama	13	1.000	FLD
17	drama	14	1.000	FLD
1		1	0.950	TIM
2		2	0.950	LOC
5		3	0.950	TIM
6		4	0.950	LOC
22	tragedy	13	1.000	FLD
23	tragedy	14	1.000	FLD
25		1	0.950	FUT
26		3	0.950	FUT
27		1	0.950	PTT
28		3	0.950	PTT
29		1	0.950	PRT
30		3	0.950	PRT
31		15	0.950	TIM
32		16	0.950	LOC
33		15	0.950	FUT
34		15	0.950	PTT
35		15	0.950	PRT
9	on	5	0.700	LOC
10	on	6	0.700	LOC
14	by	11	0.700	INS
18	drama	def	0.700	NEM
19	for	7	0.700	OBJ
21	from	def	0.700	SRC
3		2	0.700	DST
4		2	0.700	SRC
7		4	0.700	DST
8		4	0.700	SRC
24	tragedy	def	0.700	NEM
11	on	def	0.600	OBJ
15	by	12	0.600	INS
20	for	def	0.600	RSN
36	by	17,19	1.000	SRC
37	by	18,19	1.000	SRC

MORFOLOKA PRAVILA ZA SRPSKOHRVATSKI JEZIK:

lexrules table

završ. izved. reči	završ. osnove	vrsta osnove	oblik osnove	vrsta izved. reči	oblik izved. reči	gl.prom	offset
e	a	im	zs1	im	zs2	-	1
i	a	im	zs1	im	zs3	-	2
u	a	im	zs1	im	zs4	-	3
om	a	im	zs1	im	zs6	-	4
e	a	im	zs1	im	zp1	-	1
a	a	im	zs1	im	zp2	-	0
ama	a	im	zs1	im	zp3	-	7
e	a	im	zs1	im	zp4	-	1
idp_cons+i	np_cons+a	im	zs1	im	zs3	p1-a	10
i	a	im	zs1	im	zp2	-	2
a+cons+a	-a+cons+a	im	zs1	im	zp2	a	22
sti	st	im	zs1	im	zs2	-	1
sti	st	im	zs1	im	zs3	-	1
šću	st	im	zs1	im	zs6	č-š	4
sti	st	im	zs1	im	zp1	-	1
sti	st	im	zs1	im	zp4	-	1
ti	t	im	zs1	im	zs2	-	99
ti	t	im	zs1	im	zs3	-	99
ću	t	im	zs1	im	zs6	-	12
ju	-	im	zs1	im	zs6	-	20
ju	-	im	zs1	im	zs6	-	28
i	-	im	zs1	im	zs6	-	36
li	ao	im	zs1	im	zs2	a-o	1
li	ao	im	zs1	im	zs3	a-o	1
šlju	ao	im	zs1	im	zs6	a-o-š	4
-	-	im	zs1	im	zs4	-	0
-	-	im	zp1	im	zp2	-	+0
ma	-	im	zp1	im	zp3	-	+2
e	i	im	zp1	im	zp4	-	+3
e	a	im	ms1	im	ms2	-	1
i	a	im	ms1	im	ms3	-	2
u	a	im	ms1	im	ms4	-	3
om	a	im	ms1	im	ms6	-	4
idp_cons+i	np_cons+a	im	ms1	im	ms3	p1-a	10
a	-	im	ms1	im	ms2	-	1
u	a	im	ms2	im	ms3	-	+1
-	-	im	ms1	im	ms4	-	0
-	-	im	ms2	im	ms4	-	+0
om	a	im	ms2	im	ms6	-	+3
i	-	im	ms1	im	mp1	-	5
a	-	im	ms1	im	mp2	-	1

JSLOVI VRSTE REČI ZA SRPSKOHRVATSKI JEZIK:

wconds table

no	envir	phrase	form	respect	pos	feat	presence
1	seq	pd	g	snphr	-1	-	+
2	seq	pd	a	snphr	-1	-	+
3	seq	pd	i	snphr	-1	-	+
4	seq	pd	e	snphr	-1	-	+
5	seq	snphr	-	-	+1	-	+
6	seq	snphr	+p2	-	+1	-	+
7	seq	snphr	-	-	+3	-	+
8	seq	o/u z.	----++p	-	+1	-	+
9	seq	o/u z.	----++s	-	+1	-	+
10	seq	imenica	-	snphr	-1	-	+
11	seq	glagol	--s-	snphr	-1	-	+
12	seq	glagol	--p-	snphr	-1	-	+
13	seq	glagol	-	-	-1	-	+
14	seq	-	-	-	+0	INS	+
15	seq	pril/vz_phr	-	-	-1	-	+
16	seq	snphr	+++	-	-2	-	+
17	seq	pd	-	-	-1	-	+
18	seq	imenica	-	-	-1	-	+
19	seq	lič.zam.	-	-	-1	-	+
20	seq	pridev	-	-	-1	-	+
21	seq	pok.zam.	-	-	-1	-	+
22	seq	"se"	-	-	+1	-	+
23	seq	"se"	-	-	-1	-	+
24	seq	pridev	+++++	-	-2	-	+
25	seq	imenica	-	-	+1	-	+
26	seq	pridev	-	-	+1	-	+
27	seq	glagol	-	-	+1	-	+
28	seq	pd	-	-	+1	-	+
29	seq	vz	-	-	+1	-	+
30	seq	pridev	-	-	+2	-	+
31	seq	imenica	zs-	-	-1	-	+
32	seq	imenica	zp-	-	-1	-	+
33	seq	imenica	ms-	-	-1	-	+
34	seq	imenica	mp-	-	-1	-	+
35	seq	imenica	ss-	-	-1	-	+
36	seq	imenica	sp-	-	-1	-	+

PRAVILA VRSTE REČI ZA SRPSKOHRVATSKI JEZIK:

wclrules table

no	word	conds	tset	type	wgh
1	–	1,6	imenica	+p2	1.0
2	–	1,5,7	imenica	+p2	1.0
3	–	1,5,-7	imenica	+s2	1.0
4	–	1,8	imenica	+p2	1.0
5	–	1,9	imenica	+s2	1.0
6	–	2,5,7	imenica	+p4	1.0
7	–	2,5,-7	imenica	+s4	1.0
8	–	2,6	imenica	+s4	1.0
9	–	2	imenica	++4	1.0
10	–	2,8	imenica	+p4	1.0
11	–	2,9	imenica	+s4	1.0
12	–	4	imenica	++3	1.0
13	–	4,5,7	imenica	+p3	1.0
14	–	4,5,-7	imenica	+s3	1.0
15	–	4,6	imenica	+p3	1.0
16	–	4,8	imenica	+p3	1.0
17	–	4,9	imenica	+s3	1.0
18	–	3	imenica	++6	1.0
19	–	15,16	imenica	+++	0.8
20	–	1	imenica	zs2	0.6
21	–	10	im:zs2,zp1,zp4	zs2	1.0
22	–	1	im:zs2,zp1,zp4	zs2	1.0
23	–	2	im:zs2,zp1,zp4	zp4	1.0
24	–	13	im:zs2,zp1,zp4	zp4	0.9
25	–	-13,8	im:zs2,zp1,zp4	zp1	0.9
26	–	-13,9	im:zs2,zp1,zp4	zs2	0.9
27	–	-13	im:zs2,zp1,zp4	zp1	0.8
28	–	3	im:+p3,+p6	+p6	1.0
29	–	-3,-4,14	im:+p3,+p6	+p6	0.9
30	–	-3,-4	im:+p3,+p6	+p3	0.8
31	–	4	im:+p3,+p6	+p3	1.0
32	–	2	im:+s1,+s4	+s4	1.0
33	–	13	im:+s1,+s4	+s4	0.9
34	–	-2,-13	im:+s1,+s4	+s1	.85
35	–	def	im:ms2,mp2	ms2	0.8
36	–	17	imenica/glagol	im	1.0
37	–	18	imenica/glagol	gl	0.85
38	–	19	imenica/glagol	gl	0.85
39	–	20	imenica/glagol	im	1.0
40	–	21	imenica/glagol	im	1.0
41	–	18	im:++2/glagol	im	0.9
42	–	22	imenica/glagol	gl	1.0
43	–	23	imenica/glagol	gl	1.0
44	–	3	pridev	++ón+	1.0
45	–	10	pridev	++2n+	0.9
46	–	1	pridev	++3n+	1.0

TEKST SA OZNAČENIM VIŠEZNAČNIM REČIMA:

```
0,0,100,10,3,c,.pp                              ,0,0,
100,100,100,200,10,6,w,Albert,-2028226,14426,
100,100,200,200,20,8,w,Einstein,-2028226,14426,
100,100,300,200,30,3,w,was,142120011,17566,
100,100,400,200,40,4,w,born,159300007,16982,
100,100,500,200,50,2,w,in,861740000,7032,
100,100,600,200,60,3,w,Ulm,-201621132,14032,
100,100,700,200,70,0,p,\,,0,0,
100,200,100,300,10,7,w,Germany,-66852132,14032,
100,200,200,300,20,0,p,\,,0,0,
100,300,100,400,10,2,w,on,0,0,f
100,300,200,400,20,5,w,March,-109347184,14084,
100,300,300,400,30,2,n,14,140000,5000,
100,300,400,400,40,0,p,\,,0,0,
100,400,100,400,50,4,n,1879,18790000,5000,
100,400,200,400,60,0,p,\.,0,0,
200,100,100,500,10,3,w,The,1916340000,3646,
200,100,200,500,20,9,w,following,0,0,w
200,100,300,500,30,4,w,year,2139730000,14084,
200,100,400,500,40,3,w,his,766130017,9926,
200,100,500,500,50,6,w,family,583930000,14026,
200,100,600,500,60,5,w,moved,1147660003,16304,
200,100,700,500,70,2,w,to,1951000000,7016,
200,100,800,500,80,6,w,Munich,-116956132,14032,
200,100,900,500,90,0,p,\,,0,0,
200,200,100,600,10,5,w,where,0,0,w
200,200,200,600,20,7,w,Hermann,-76348226,14426,
200,200,300,600,30,8,w,Einstein,-76348226,14426,
200,200,400,600,40,0,p,\,,0,0,
200,300,100,700,10,3,w,his,766130017,9926,
200,300,200,700,20,6,w,father,584530000,14426,
200,300,300,700,30,0,p,\,,0,0,
200,400,100,800,10,3,w,and,57510000,4028,
200,400,200,800,20,5,w,Jakob,-90857226,14426,
200,400,300,800,30,8,w,Einstein,-90857226,14426,
200,400,400,800,40,0,p,\,,0,0,
200,500,100,800,50,3,w,his,766130017,9926,
200,500,200,800,60,5,w,uncle,2017790000,14426,
200,500,300,800,70,0,p,\,,0,0,
200,600,100,900,10,3,w,set,1722870000,16504,
200,600,200,900,20,2,w,up,2020640000,1016,
200,600,300,900,30,1,w,a,117760000,3546,
200,600,400,900,40,5,w,small,1774410000,2036,
200,600,500,900,50,10,w,electrical,511500025,2076,
200,600,600,900,60,5,w,plant,0,0,b
200,600,700,900,70,3,w,and,57510000,4028,
200,600,800,900,80,11,w,engineering,0,0,w
200,600,900,900,90,5,w,works,0,0,w
200,600,1000,900,100,0,p,\.,0,0,
0,0,1000,10,3,c,.pp                              ,0,0,
100,100,100,1100,10,2,w,At,85320000,7032,
100,100,200,1100,20,3,w,the,1916340000,3646,
100,100,300,1100,30,6,w,behest,138600000,14034,
100,100,400,1100,40,2,w,of,1330800000,7054,
100,100,500,1100,50,3,w,his,766130017,9926,
100,100,600,1100,60,6,w,mother,1147400000,14626,
```

TEKST SA RAZREĐENIM VIŠEZNACNIM RECIMA:

```
0,0,100,10,3,c,.pp                              ,0,0,
100,100,100,200,10,6,w,Albert,-2028226,14426,
100,100,200,200,20,8,w,Einstein,-2028226,14426,
100,100,300,200,30,3,w,was,142120011,17566,
100,100,400,200,40,4,w,born,159300007,16982,
100,100,500,200,50,2,w,in,861740000,7032,
100,100,600,200,60,3,w,Ulm,-201621132,14032,
100,100,700,200,70,0,p,\.,0,0,
100,200,100,300,10,7,w,Germany,-66852132,14032,
100,200,200,300,20,0,p,\.,0,0,
100,300,100,400,10,2,w,on,1380670000,7084,
100,300,200,400,20,5,w,March,-109347184,14084,
100,300,300,400,30,2,n,14,140000,5000,
100,300,400,400,40,0,p,\.,0,0,
100,400,100,400,50,4,n,1879,18790000,5000,
100,400,200,400,60,0,p,\.,0,0,
200,100,100,500,10,3,w,The,1916340000,3646,
200,100,200,500,20,9,w,following,617500002,14248,
200,100,300,500,30,4,w,year,2139730000,14084,
200,100,400,500,40,3,w,his,766130017,9926,
200,100,500,500,50,6,w,family,583930000,14026,
200,100,600,500,60,5,w,moved,1147660003,16304,
200,100,700,500,70,2,w,to,1951000000,7016,
200,100,800,500,80,6,w,Munich,-116956132,14032,
200,100,900,500,90,0,p,\.,0,0,
200,200,100,600,10,5,w,where,2082960000,13032,
200,200,200,600,20,7,w,Hermann,-76348226,14426,
200,200,300,600,30,8,w,Einstein,-76348226,14426,
200,200,400,600,40,0,p,\.,0,0,
200,300,100,700,10,3,w,his,766130017,9926,
200,300,200,700,20,6,w,father,584530000,14426,
200,300,300,700,30,0,p,\.,0,0,
200,400,100,800,10,3,w,and,57510000,4028,
200,400,200,800,20,5,w,Jakob,-90857226,14426,
200,400,300,800,30,8,w,Einstein,-90857226,14426,
200,400,400,800,40,0,p,\.,0,0,
200,500,100,800,50,3,w,his,766130017,9926,
200,500,200,800,60,5,w,uncle,2017790000,14426,
200,500,300,800,70,0,p,\.,0,0,
200,600,100,900,10,3,w,set,1722870000,16504,
200,600,200,900,20,2,w,up,2020640000,1016,
200,600,300,900,30,1,w,a,117760000,3546,
200,600,400,900,40,5,w,small,1774410000,2036,
200,600,500,900,50,10,w,electrical,511500025,2076,
200,600,600,900,60,5,w,plant,1466920000,14046,
200,600,700,900,70,3,w,and,57510000,4028,
200,600,800,900,80,11,w,engineering,520540067,14204,
200,600,900,900,90,5,w,works,2103120001,14104,
200,600,1000,900,100,0,p,\.,0,0,
0,0,1000,10,3,c,.pp                             ,0,0,
100,100,100,1100,10,2,w,At,85320000,7032,
100,100,200,1100,20,3,w,the,1916340000,3646,
100,100,300,1100,30,6,w,behest,138600000,14034,
100,100,400,1100,40,2,w,of,1330800000,7054,
100,100,500,1100,50,3,w,his,766130017,9926,
100,100,600.1100,60,6,w,mother,1147400000,14626,
```

DODATAK 4: OSNOVNE RELACIJE I REZULTATI ODREDJIVANJA REFERENATA ZAMENICA

RELACIJE SA PRAVILIMA I USLOVIMA ZA ENGLESKI JEZIK:

pnrules table

no	ccond	pos	object	ocond	t	dir
1	1	+	pn	A	rse	+
3	1,3A,F		noun	A,B	rse	+
4	1	-	pn	A,F	rse	-
5	2A		pn	A	rse	-
9	1	-	noun	A,B,F,G	r	-
10	1	-	noun	A,B,F,G	se	-
11	2A		noun	A,B,F	rse	-
12	1,5A,E		pn	-	t	-
14	1	-	noun	A,E	t	-
15	4	0	noun	-	h	+
17	1	-	pradj		rs	-
18	1,3A,F		pradj		rs	+
16	1	-	noun	A,C,E	t	-
19	2A		pradj		rs	-

contcond table

no	object	offset
1	sentence	0
2	sentence	-
3	sequence	+
4	sequence	0
5	cnp	-
6	cnp	
7	cprph	-
8	cprph	
9	cnp	0

objcond table

ind	charac	pres
A	quot	-
B	cnp_pn_source	-
C	prepphr	-
D	same_seq	-
E	paren	-
F	relcl	-
G	srcrel	-

TEKST SA REFERENTIMA ZAMENICA PRIDRUŽENIM ZAMENICAMA:

```
,0,0,100,10,3,c,.pp                                    ,0,0,0,0,0,0
,100,100,100,200,10,6,w,Albert,-2028226,14426,0,0,0,0
,100,100,200,200,20,8,w,Einstein,-2028226,14426,0,0,0,0
,100,100,300,200,30,3,w,was,142120011,17566,0,0,0,0
,100,100,400,200,40,4,w,born,159300007,16982,0,0,0,0
,100,100,500,200,50,2,w,in,861740000,7032,0,0,0,0
,100,100,600,200,60,3,w,Ulm,-201621132,14032,0,0,0,0
,100,100,700,200,70,0,p,\,,0,0,0,0,0,0
,100,200,100,300,10,7,w,Germany,-66852132,14032,0,0,0,0
,100,200,200,300,20,0,p,\,,0,0,0,0,0,0
,100,300,100,400,10,2,w,on,1380670000,7084,0,0,0,0
,100,300,200,400,20,5,w,March,-109347184,14084,0,0,0,0
,100,300,300,400,30,2,n,14,140000,5000,0,0,0,0
,100,300,400,400,40,0,p,\,,0,0,0,0,0,0
,100,400,100,400,50,4,n,1879,18790000,5000,0,0,0,0
,100,400,200,400,60,0,p,\.,0,0,0,0,0,0
,200,100,100,500,10,3,w,The,1916340000,3646,0,0,0,0
,200,100,200,500,20,9,w,following,617500002,14248,0,0,0,0
,200,100,300,500,30,4,w,year,2139730000,14084,0,0,0,0
,200,100,400,500,40,3,w,his,766130017,9926,0,0,-18,-17

ɔert Einstein

,200,100,500,500,50,6,w,family,583930000,14026,0,0,0,0
,200,100,600,500,60,5,w,moved,1147660003,16304,0,0,0,0
,200,100,700,500,70,2,w,to,1951000000,7016,0,0,0,0
,200,100,800,500,80,6,w,Munich,-116956132,14032,0,0,0,0
,200,100,900,500,90,0,p,\,,0,0,0,0,0,0
,200,200,100,600,10,5,w,where,2082960000,13032,0,0,-2,-2

nich

,200,200,200,600,20,7,w,Hermann,-76348226,14426,0,0,0,0
,200,200,300,600,30,8,w,Einstein,-76348226,14426,0,0,0,0
,200,200,400,600,40,0,p,\,,0,0,0,0,0,0
,200,300,100,700,10,3,w,his,766130017,9926,0,0,-28,-27

bert Einstein

,200,300,200,700,20,6,w,father,584530000,14426,0,0,0,0
,200,300,300,700,30,0,p,\,,0,0,0,0,0,0
,200,400,100,800,10,3,w,and,57510000,4028,0,0,0,0
,200,400,200,800,20,5,w,Jakob,-90857226,14426,0,0,0,0
,200,400,300,800,30,8,w,Einstein,-90857226,14426,0,0,0,0
,200,400,400,800,40,0,p,\,,0,0,0,0,0,0
,200,500,100,800,50,3,w,his,766130017,9926,0,0,-35,-34

bert Einstein

,200,500,200,800,60,5,w,uncle,2017790000,14426,0,0,0,0
,200,500,300,800,70,0,p,\,,0,0,0,0,0,0
,200,600,100,900,10,3,w,set,1722870000,16504,0,0,0,0
,200,600,200,900,20,2,w,up,2020640000,1016,0,0,0,0
,200,600,300,900,30,1,w,a,117760000,3546,0,0,0,0
,200,600,400,900,40,5,w,small,1774410000,2036,0,0,0,0
,200,600,500,900,50,10,w,electrical,511500025,2076,0,0,0,0
,200,600,600,900,60,5,w,plant,1466920000,14046,0,0,0,0
,200,600,700,900,70,3,w,and,57510000,4028,0,0,0,0
,200,600,800,900,80,11,w,engineering,520540067,14204,0,0,0,0
```

INVERTOVANI INDEKS:

:8226,Albert Einstein ,1020104
:956132,Munich ,1020201
:8226,Albert Einstein ,1020301
:8226,Albert Einstein ,1020501
:8226,Albert Einstein ,2010105
:991126,Einstein ,2010402
:991126,Einstein ,2010501

RELACIJE PRAVILA I OBJEKAT-USLOVA ZA SRPSKOHRVATSKI JEZIK:

pnrules table

no	ccond	pos	object	form	ocond	type	dir	wgh
1	1,6F	+	1	--1	A	up	+	0.9
2	1,6F	-	imen.	--1	A	up	-	0.95
3	1	-	1	--1	A	up	-	0.95
4	4,9	+	imen.	+++	-	k	+	1.0
5	4	+	imen.	+-+	-	k	+	0.8
6	1,6F	+	imen.	--1	A,I	up	+	0.9
7	1	-	vl.pridev	---	A	up	-	0.7
8	1	+	vl.pridev	---	A	up	+	0.7
9	1	+	imen.fr		A	up	+	0.9
10	1	-	imen.fr		A	up	-	0.9
11	1	-	imen.	+++	A,F,G	l	-	0.95
12	1	-	imen.	++-	A,E,H	l	-	0.9
13	1,3A		imen.	++-	A,J	l	+	0.8
14	2A		pr.pridev	++-	-	l	-	0.9
15	1	-	pr.pridev	++-	-	l	-	0.95
16	2A		imen.	++-	-	l	-	0.9
17	1	-	imen.	++-	A,E,H	p	-	0.95
18	1	-	pr.pridev	+++----	-	p	-	0.95
19	2A		pr.pridev	+++----	-	p	-	0.95
20	1	-	kompl.im.fr	+++	K	uo	-	0.95

(p= prisvojna, k= pokazna, l= lična, uo= upitno-odnosna,
up= univerzalna prisvojna zamenica tipa "svoj")

objcond table

ind	charac	pres	comment
A	navod	-	navodnici
B	im-zam.fr	-	zamenica i original u istoj im.frazi
C	predl.fr.	-	predloška fraza
D	sekv	-	ista sekvenca
E	zagr	-	zagrade
F	rel.reč.	-	relativna rečenica
G	sl.rel.reč.	-	složena relativna rečenica
H	predl.im.fr.	-	imen.fr. sa predloškom u kojoj je zamen.
I	sl.im.fr.	-	složena imenička fraza
J	nenabr.	-	nenabrajajući zarez
K	sem.sl.	-	semantičko slaganje