

# SUSTAV ZA INTEGRACIJU RELACIJSKE BAZE PODATAKA I JEDNOSTAVNIH REČENICA PRIRODNOG JEZIKA PRIMJENOM KONCEPTUALNOG OKVIRA "NODE OF KNOWLEDGE"

---

**Ašenbrener Katić, Martina**

**Doctoral thesis / Disertacija**

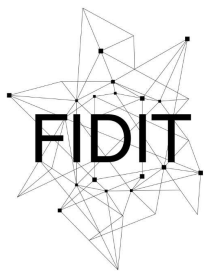
**2017**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **University of Rijeka / Sveučilište u Rijeci**

*Permanent link / Trajna poveznica:* <https://urn.nsk.hr/urn:nbn:hr:195:390832>

*Rights / Prava:* [Attribution-NonCommercial-ShareAlike 4.0 International/Imenovanje-Nekomercijalno-Dijeli pod istim uvjetima 4.0 međunarodna](#)

*Download date / Datum preuzimanja:* **2024-07-25**



Sveučilište u Rijeci  
**Fakultet informatike  
i digitalnih tehnologija**

*Repository / Repozitorij:*

[Repository of the University of Rijeka, Faculty of Informatics and Digital Technologies - INFORI Repository](#)



SVEUČILIŠTE U RIJECI  
ODJEL ZA INFORMATIKU

Martina Ašenbrener Katić

**SUSTAV ZA INTEGRACIJU  
RELACIJSKE BAZE PODATAKA I  
JEDNOSTAVNIH REČENICA  
PRIRODNOG JEZIKA PRIMJENOM  
KONCEPTUALNOG OKVIRA "NODE OF  
KNOWLEDGE"**

DOKTORSKI RAD

Rijeka, 2017.



SVEUČILIŠTE U RIJECI  
ODJEL ZA INFORMATIKU

Martina Ašenbrener Katić

**SUSTAV ZA INTEGRACIJU  
RELACIJSKE BAZE PODATAKA I  
JEDNOSTAVNIH REČENICA  
PRIRODNOG JEZIKA PRIMJENOM  
KONCEPTUALNOG OKVIRA "NODE OF  
KNOWLEDGE"**

DOKTORSKI RAD

Mentor: prof. dr. sc. Mile Pavlić

Komentor: prof. dr. sc. Zdravko Dovedan Han

Rijeka, 2017.

UNIVERSITY OF RIJEKA  
DEPARTMENT OF INFORMATICS

Martina Ašenbrener Katić

**A SYSTEM FOR INTEGRATION OF  
RELATIONAL DATABASE AND SIMPLE  
SENTENCES IN NATURAL LANGUAGE  
USING THE "NODE OF KNOWLEDGE"  
CONCEPTUAL FRAMEWORK**

DOCTORAL THESIS

Rijeka, 2017.

Mentor rada: prof. dr. sc. Mile Pavlić

Komentor rada: prof. dr. sc. Zdravko Dovedan Han

Doktorski rad obranjen je dana 04. travnja 2017. na Odjelu za informatiku Sveučilišta u Rijeci, pred povjerenstvom u sastavu:

1. Prof. dr. sc. Mario Radovan
2. Izv. prof. dr. sc. Patrizia Pošćić
3. Doc. dr. sc. Alen Jakupović

## **Zahvala**

Najljepše zahvaljujem mentoru, prof. dr. sc. Mili Pavliću te komentoru, prof. dr. sc. Zdravku Dovedan Hanu na vodstvu, savjetima te pruženoj svesrdnoj stručnoj i osobnoj podršci tijekom izrade ovog doktorskog rada.

Također iskreno zahvaljujem svim kolegama s Odjela za informatiku Sveučilišta u Rijeci koji su svojim savjetima i podrškom pridonijeli kvaliteti ovog rada.

Zahvaljujem gospodinu Veljku Klemencu iz tvrtke Prim-Soft te Sanjinu Maržiću iz tvrtke Ris d.o.o. na pomoći pri izradi prototipa sustava uključenih u ovo istraživanje.

Zahvaljujem i svim članovima moje obitelji na pruženom strpljenju i osobnoj podršci tijekom mog rada i napredovanja.

## Sažetak

Postojeći informacijski sustavi i njihove pripadne baze podataka ne mogu prihvatiti znanja zapisana u rečenicama prirodnog jezika. Na osnovi konceptualnog okvira "Node of Knowledge (NOK)", analizira se i modelira novi sustav temeljen na NOK metodi koji će u svojoj bazi podataka (BPNOK) čuvati tekstualno znanje iz rečenica prirodnog jezika bez gubitka semantike. Osim toga, sustav temeljen na NOK metodi može prihvatiti i pitanja dana upitnim rečenicama prirodnog jezika te na temelju znanja pohranjenog u BPNOK dati odgovor na postavljeno pitanje. Cilj ovog rada je obogatiti postojeće informacijske sustave i njihove baze podataka sustavom za prihvati i predstavljanje znanja zapisanih u tekstovima s jednostavnim rečenicama prirodnog jezika, odnosno sustavom temeljenom na konceptualnom okviru "Node of Knowledge". Integracija dvaju sustava omogućuje dodavanje novih znanja vezanih za podatke u bazi podataka postojećeg informacijskog sustava bez izmjene postojeće sheme baze podataka. Rezultat tog proširenja je integralna semantički bogatija baza podataka.

U radu je definiran konceptualni okvir "Node of Knowledge". Definirana su pravila transformacije izjavnih i upitnih rečenica prirodnog jezika u FNOK, odnosno QFNOK zapis. Također, definirani su modeli sustava temeljenog na NOK metodi, algoritmi za transformaciju FNOK i QFNOK zapisa u relacijsku bazu podataka te algoritam za kreiranje odgovora. Razvijen je model integrirane baze podataka, odnosno baze podataka postojećeg informacijskog sustava koji je integriran sa sustavom koji omogućuje unos znanja iskazanog rečenicama prirodnog jezika. Sada je moguće definirati pitanja prirodnog jezika koja obuhvaćaju znanje iz oba sustava. Upiti su vezani za obje relacijske baze podataka. Osim toga, omogućeno je i dobivanje odgovora na postavljeno pitanje.

**Ključne riječi:** informacijski sustav, relacijska baza podataka, predstavljanje znanja, metoda entiteta i veza, metoda "Node of Knowledge", pravila transformacije



## **Abstract**

The existing information systems and their associated databases cannot accept the knowledge written in sentences in natural language. Based on the conceptual framework Node of Knowledge (NOK), a new system based on the NOK method is analysed and modelled. It will save textual knowledge from natural language sentences into its database (DBNOK) without loss of semantics. In addition, a system based on NOK method can also accept questions in the form of interrogative sentences of natural language, and based on the knowledge stored in DBNOK, provide answers to the questions. The aim of this paper is to enrich existing information systems and their databases with a system for the reception and representation of knowledge written in texts using simple sentences of natural language; i.e. with a system that is based on the conceptual framework of "Node of Knowledge". The integration of two systems allows the addition of new knowledge related to the data in the database of the existing information system without changing the existing database schema. The result of this expansion is an integral, semantically richer database.

The paper defines the conceptual framework "Node of Knowledge". Rules for transformation of declarative and interrogative sentences in natural language into FNOK and QFNOK records are defined. Also, models of the system based on NOK method, algorithms for transformation of FNOK and QFNOK records into a relational database and an algorithm for creating answers are defined. A model of an integrated database, i.e. a database of the existing information system that is integrated with a system which allows entering knowledge expressed in sentences in natural language is developed. It is now possible to define questions in natural language which include knowledge of both systems. Queries are connected to both databases. In addition, it is possible to obtain answers to the set questions.

**Keywords:** information systems, relational database, knowledge representation, entity and relationship method, method "Node of Knowledge", rules of transformation

# Sadržaj

1	Uvod .....	1
1.1	Motivacija.....	2
1.2	Ciljevi, hipoteze i znanstveni doprinosi .....	3
1.3	Struktura rada .....	5
2	Pregled područja .....	7
2.1	Umjetna inteligencija.....	7
2.2	Znanje .....	10
2.3	Predstavljanje znanja .....	14
2.3.1	Semantičke mreže.....	16
2.3.2	Produkcijski sustavi .....	17
2.3.3	Okviri.....	18
2.3.4	Objektno orijentirani pristup .....	18
2.3.5	Deskriptivna logika.....	19
2.3.6	Ontologije .....	19
2.4	Grafičke metode za predstavljanje znanja .....	19
2.4.1	Basic Conceptual Graphs .....	20
2.4.2	Multi-layered extended semantic networks .....	20
2.4.3	Hierarchical Semantic Form .....	20
2.4.4	Resource Description Framework.....	21
2.4.5	Nodes of Knowledge (NOK).....	21
2.5	Obrada prirodnog jezika (engl. <i>natural language processing (NLP)</i> ) .....	21
2.6	Sustavi pitanja i odgovora (engl. <i>question – answering systems (QA systems)</i> ).....	24
2.7	Jednostavne rečenice i pitanja .....	26
3	Metoda "Node of Knowledge" .....	30
3.1	Koncepti NOK metode.....	32
3.2	Jednosmjerna i dvosmjerna NOK metoda .....	37
3.2.1	Dvosmjerna NOK metoda.....	37
3.2.2	Jednosmjerna NOK metoda .....	39
3.2.2.1	FNOK formalizam .....	40
3.3	Konceptualni okvir "Node of Knowledge" .....	41
3.3.1	Formalizacija teksta .....	42
3.4	Preslikavanje između relacijske baze podataka i NOK metode .....	44
3.4.1	Preslikavanje metode entiteta i veza u NOK metodu .....	44
3.4.2	Preslikavanje NOK metode u metodu entiteta i veza .....	50
4	Pravila prevođenja rečenica prirodnog jezika u FNOK.....	56
4.1	Pravila za prevođenje rečenica u FNOK prema službi riječi u rečenici (odnosno elementima rečeničnog ustrojstva) .....	57
4.2	Pravila prevođenja rečenice u FNOK prema vrsti riječi u rečenici.....	59
4.2.1	Osnovna pravila prevođenja .....	60
4.2.2	Glagoli .....	64

4.2.2.1	Glagol na početku zapisa.....	64
4.2.2.2	Glagol ispred glagola .....	65
4.2.2.3	Hijerarhija glagolskih vremena.....	65
4.2.3	Imenice.....	69
4.2.3.1	Odnos imenice prema glagolu .....	69
4.2.3.2	Imenski predikat.....	70
4.2.3.3	Apozicija i apozicijski skup .....	71
4.2.3.4	Padeži imenica.....	72
4.2.3.5	Article (Član) - A, an, the .....	72
4.2.4	Pridjevi .....	73
4.2.4.1	Pridjev kao dio imenskog predikata.....	74
4.2.4.2	Pridjev koji opisuje imenicu .....	74
4.2.4.3	Komparacija pridjeva.....	75
4.2.5	Zamjenice .....	75
4.2.5.1	Imenične zamjenice .....	77
4.2.5.2	Pridjevne zamjenice .....	78
4.2.6	Brojevi.....	80
4.2.6.1	Pravilo broja .....	80
4.2.6.2	Mjerne jedinice .....	80
4.2.7	Prilozi.....	83
4.2.7.1	Prilozi vezani za glagole .....	85
4.2.7.2	Prilozi vezani za druge vrste riječi (osim glagola).....	85
4.2.8	Prijedlozi.....	86
4.2.9	Napomena .....	87
4.3	Pravila prevođenja upitnih rečenica prirodnog jezika u QFNOK .....	87
4.3.1	Da/Ne pitanje (engl. polar question).....	88
4.3.2	Upitne rečenice koje započinju wh-pitanjima.....	89
4.3.2.1	Traženi odgovor na nižim razinama hijerarhije .....	90
4.3.2.2	Traženi odgovor je na najvišoj razini hijerarhije .....	93
5	Meta modeli NOK metode.....	94
5.1	Priprema teksta za transformaciju.....	95
5.2	Transformacija teksta u skladu s NOK metodom .....	105
5.2.1	Pojednostavljanje meta modela NOK-a za dvosmjernu vezu .....	107
5.2.2	Meta model NOK-a za jednosmjernu vezu.....	108
5.2.3	Transformacija dvosmjernog DNOK-a u NOK bazu podataka teksta (NOKBPT) .....	112
5.3	Dokaz hipoteze H1.....	119
6	Sustav temeljen na NOK metodi .....	120
6.1	Meta model FNOK zapisa .....	122
6.1.1	Povezivanje čvorova FNOK zapisa s rječnikom.....	125
6.1.2	Rastavljanje FNOK rečenice na skup pitanje-odgovor.....	127
6.1.3	Relacijska shema baze podataka .....	129

6.1.4	Algoritam transformacije FNOK zapisa u BPNOK .....	129
6.2	Meta model QFNOK zapisa .....	135
6.2.1	Relacijska shema baze podataka .....	137
6.2.2	Algoritam transformacije QFNOK zapisa u BPNOK .....	137
6.3	Kreiranje odgovora .....	140
7	Primjena sustava temeljenog na NOK metodi .....	148
7.1	Primjena sustava temeljenog na NOK metodi .....	148
7.2	Analiza rezultata.....	153
7.2.1	Rezultati za hrvatski jezik.....	155
7.2.2	Rezultati testiranja sustava rečenicama engleskog jezika .....	160
7.3	Usporedba dvaju sustava temeljenih na NOK metodi .....	167
7.3.1	Sustav temeljen na gramatici (engl. Grammar-based system (GBS)) .....	167
7.3.2	Sustav temeljen na NOK metodi .....	168
7.3.3	Rezultati usporedbe dvaju sustava .....	170
7.4	Dokaz hipoteze H2.....	175
8	Proširenje postojeće baze podataka odabranog informacijskog sustava bazom podataka sustava temeljenog na NOK-u.....	180
8.1	Proširenje baze postojećeg informacijskog sustava.....	180
8.1.1	Propitivanje BPNOK baze podataka.....	187
8.2	Propitivanje integrirane baze podataka.....	191
8.2.1	Odgovori integriranog sustava.....	194
8.3	Analiza rezultata.....	199
8.4	Dokaz hipoteze H3.....	201
9.	Zaključak .....	203
	Literatura.....	208
	Popis slika.....	215
	Popis tablica.....	217
	Popis privitaka.....	221
	Privitak 1. SQL skripte za sustav BPNOK .....	222
	Privitak 2. Tablice.....	239
	Privitak 3. Formalizacija rečenica i pitanja .....	289
	Privitak 4. SQL upiti nad integriranom bazom podataka.....	294
	Životopis .....	314

# 1 Uvod

Informacijski sustav sastavni je dio svakoga poslovnog sustava, a razvoj informacijskog sustava kontinuiran je i zahtjevan posao [91]. U današnje vrijeme, poslovne organizacije bez obzira na veličinu i bogatstvo, ne mogu bez informacijskog sustava zasnovanog na informacijsko-komunikacijskoj tehnologiji. Informacija je ključan resurs za uspješno odvijanje poslovnih procesa [86].

Područje razvoja informacijskih sustava kontinuirano se razvija. Intenzivno se razvijaju organizacijski sustavi, informacijski sustavi i informacijska tehnologija. Do danas je razvijeno više od 1000 metodologija za razvoj informacijskih sustava [87]. Jedna od njih je i specijalizirana Metodologija za razvoj informacijskog sustava (skraćeno MIRIS) koja se sastoji od skupa metoda i uputa čiji je ukupni cilj projektirati i izgraditi informacijski sustav (IS), odnosno izgraditi model informacijskog sustava koji se sastoji od modela procesa, modela podataka i modela resursa. MIRIS propisuje faze razvoja i aktivnosti pojedine faze do potrebne razine detalja. U fazi izvedbenog projekta metodologije MIRIS, metodom entiteta i veza (EV) modeliraju se podaci te se model podataka, odnosno dobiveni dijagram entiteta i veza (DEV) prevodi u relacijsku shemu baze podataka. Na temelju relacijske sheme baze podataka u odabranom softveru za razvoj programskog proizvoda stvara se aplikacija. Jednom izgrađenu aplikaciju složeno je mijenjati, stoga kod projektiranja modela podataka treba predvidjeti buduće zahtjeve. Da bi se u aplikaciji promijenio jedan atribut, potrebno je mijenjati shemu baze podataka, a zatim i samu aplikaciju.

Nad podacima u relacijskoj bazi podataka moguće je postavljati SQL upite koji će, pretražujući po relacijama, odnosno tablicama u bazi, prikupiti podatke i dati odgovor na zadani SQL upit. Međutim, ako postavimo upit prirodnim jezikom, relacijska baza podataka ne može nam dati odgovor. Isto tako, ako podacima u relacijskoj bazi podataka želimo dodati nova znanja koja će se s vremena na vrijeme izmjenjivati, javlja se problem izmjene relacijske baze podataka. Ako pak novo znanje unosimo kao rečenice prirodnog jezika, dolazimo do problema jer rečenice prirodnog jezika nije moguće upisati u relacijsku bazu podataka (osim kao tekstualno polje, međutim nad njime nije moguće vršiti propitivanje kao nad ostalim podacima upotrebom SQL upitnog jezika). U relacijsku bazu podataka mogu se upisivati samo podaci i to prema određenim

kriterijima. To znači da je rečenicu potrebno rastaviti na manje dijelove (frazе ili riječi) koje se mogu upisati u bazu podataka. Pritom treba paziti da se ne izgubi semantika između riječi ili fraza.

Danas postoje mnogi dodatni sustavi i alati poslovne inteligencije [112], [31], na primjer, OLAP ili *data warehousing* alati koji mogu poboljšati zaključivanje nad podacima u informacijskom sustavu. Međutim, još uvijek postoji problem sa semantikom prirodnog jezika. Jezik je jedan od univerzalnih kognitivnih alata našeg uma, komunikacijski alat koji odražava život. Semantički aspekt riječi - njeno pravo značenje – postaje jedan od najfascinantiјih pitanja na području računalne lingvistike i umjetne inteligencije [64].

Jedno od rješenja problema unosa tekstualnog znanja u relacijsku bazu podataka daje metoda "Node of Knowledge" (NOK metoda) koja se zasniva na istraživanju i analizi prirodnog jezika. Analiziraju se rečenice, riječi i njihovo značenje te redoslјed povezivanja riječi u složenije oblike. Drugim riječima, baze podataka ne mogu čuvati rečenice prirodnog jezika pa želimo povezati relacijsku bazu podataka s bazom podataka novog sustava temeljenog na NOK metodi. Samim time pokazala se potreba za sustavom koji bi omogućio pohranjivanje tekstualnog znanja u baze podataka. Isto tako, postoji potreba za postavljanjem pitanja i traženjem odgovora u tekstualnim zapisima.

## **1.1 Motivacija**

S obzirom da tekstualno znanje iskazano rečenicama prirodnog jezika želimo pohraniti u relacijsku bazu podataka, dolazimo do problema kako rečenice prirodnog jezika upisati u relacijsku bazu podataka na način da je omogućeno semantičko propitivanje. Ako rečenicu upisujemo u tekstualno polje, tada nad tim rečenicama nije moguće vršiti propitivanje, kao nad ostalim podacima u bazi. Ovi problemi vrlo su važni u domeni razvoja informacijskih sustava i baza podataka. Također, problem je usko povezan s poljem obrade prirodnog jezika. Međutim, opći sustavi koji mogu pretvarati rečenice prirodnog jezika u relacijske baze podataka bez gubitka semantike ne postoje.

Drugi problem koji se javlja je kako propitivati postojeće informacijske sustave upitima definiranim na prirodnom jeziku te kako nova znanja vezana za podatke u bazi podataka postojećeg informacijskog sustava upisati u postojeću bazu podataka bez izmjene sheme baze podataka postojećeg sustava.

U radu će biti definiran model sustava koji će omogućiti pohranu tekstualnih znanja u relacijskim bazama podataka na način da se dobiju smisleni odgovori na upite vezane uz pohranjeno tekstualno znanje. Drugim riječima, motivacija za istraživanje jest naći sustav koji će omogućiti pohranu znanja iz rečenica prirodnog jezika u relacijsku bazu podataka bez gubitka semantike. U radu je analiziran model takvog sustava temeljen na NOK metodi. Postojeći informacijski sustavi ne mogu se širiti rečenicama prirodnog jezika te je motivacija istraživanja analiza mogućnosti proširenja postojećih (gotovih) informacijskih sustava dijelom koji pohranjuje rečenice jezika. Proširenja koja integralna baza podataka treba omogućiti jesu propitivanja kroz obje baze podataka pitanjima danim upitnim rečenicama prirodnog jezika. Rezultat integracije dvaju sustava je semantički bogatija baza podataka.

## **1.2 Ciljevi, hipoteze i znanstveni doprinosi**

Cilj ovog rada je obogatiti postojeće informacijske sustave i njihove baze podataka sustavom za prihvata i predstavljanje znanja zapisanih u tekstovima s jednostavnim rečenicama prirodnog jezika. Naglasak istraživanja je na povezivanju postojeće relacijske baze podataka poslovnog informacijskog sustava s tekstovima napisanim jednostavnim rečenicama prirodnog jezika. Ta integracija omogućuje dodavanje novih znanja vezanih za podatke u bazi podataka postojećeg informacijskog sustava (bez izmjene postojeće sheme baze podataka). Za postizanje ovoga cilja definirana su pravila koja omogućuju transformaciju rečenica prirodnog jezika u formalizirani zapis koji je moguće upisati u relacijsku bazu podataka bez gubitka semantike. Razvijeni su modeli podataka novog sustava za prihvata tekstualnog znanja iskazanog rečenicama prirodnog jezika u relacijskoj bazi podataka. Razvijen je model integrirane baze podataka, odnosno baze podataka postojećeg informacijskog sustava koji je integriran sa sustavom koji omogućuje unos znanja iskazanog rečenicama prirodnog jezika. Rezultat tog proširenja je integralna semantički bogatija baza podataka.

Opseg istraživanja je ograničen na jednostavne rečenice hrvatskog i engleskog jezika. Pod definicijom "jednostavne rečenice hrvatskog jezika" podrazumijevamo izjavne rečenice koje imaju samo jedan predikat u sadašnjem glagolskom vremenu u potvrdnom obliku te izrečen subjekt u nominativu. Pod "jednostavnom rečenicom engleskog jezika"

podrazumijevamo izjavne rečenice s jednim glagolom u Simple present tense ili Present continuous tense, subjektom, objektom te priložnim oznakama.

U radu su postavljene tri istraživačke hipoteze:

**H1: Primjenom metode entiteta i veza može se projektirati meta model metode "Node of Knowledge".**

Hipoteza će se smatrati potvrđenom ako se metodom entiteta i veza napravi meta model metode "Node of Knowledge (NOK metode)". Potvrđenost ove hipoteze opisana je u poglavlju 5.3.

**H2: Semantika jednostavnih rečenica prirodnog hrvatskog jezika (izjavnih rečenica u nominativu te u sadašnjem glagolskom vremenu) ostaje očuvana u relacijskoj bazi podataka NOK-a (BPNOK).**

Hipoteza će se smatrati potvrđenom ako se pokaže da se upisivanjem rečenica prirodnog jezika (koje su NOK metodom i njenim pravilima transformirane u odgovarajući FNOK zapis prikladan za upis u relacijsku bazu podataka) neće izgubiti semantika u rečenici. Očuvanost semantike dokazuje se propitivanjem relacijske baze podataka pitanjima prirodnog jezika (također transformiranih u odgovarajući QFNOK zapis koristeći NOK metodu), odnosno dobivanjem odgovora na postavljeno pitanje na temelju podataka u relacijskoj bazi podataka. Potvrđenost ove hipoteze opisana je u poglavlju 7.4.

**H3: Rezultat proširenja relacijske baze podataka odabranog informacijskog sustava jednostavnim rečenicama zapisanim u BPNOK je integralna semantički bogatija baza podataka informacijskog sustava.**

Hipoteza će se smatrati potvrđenom ako se na temelju logičkih i implementacijskih modela postojećeg informacijskog sustava i modela sustava temeljenog na NOK metodi razvije integralni sustav nad kojim će se moći vršiti propitivanje SQL upitima koji će dati ispravne odgovore. Upiti će biti sastavljeni na način da koriste podatke iz obje baze podataka. Ukoliko dobijemo odgovor na upite pokazat ćemo da je rezultat proširenja relacijske baze podataka odabranog informacijskog sustava jednostavnim rečenicama zapisanim u BPNOK, integralna semantički bogatija baza podataka informacijskog sustava. Dodatno, ako postavljanjem upita nad integralnom bazom podataka dobijemo



rezultat koji koristi tablice u postojećoj bazi podataka, a taj rezultat ne bismo mogli dobiti samo postavljanjem upita nad postojećom bazom podataka također možemo zaključiti da je rezultat proširenja relacijske baze podataka odabranog informacijskog sustava jednostavnim rečenicama zapisanim u BPNOK integralna semantički bogatija baza podataka informacijskog sustava. Potvrđenost ove hipoteze opisana je u poglavlju 8.4.

Prilikom izrade ovog rada postignuti su sljedeći znanstveni doprinosi:

- Definiranje meta modela sustava za prihvatanje i predstavljanje tekstualnog znanja iskazanog jednostavnim rečenicama prirodnog jezika
- Definiranje pravila prevođenja koncepata NOK metode u koncepte metode entiteta i veza i obrnuto
- Definiranje pravila transformacije rečenica prirodnog jezika u FNOK zapis i definiranje pravila transformacije pitanja prirodnog jezika u QFNOK zapis
- Definiranje i razvoj modela sustava temeljenog na konceptualnom okviru "Node of knowledge"
- Oblikovanje semantički bogatijeg sustava proširenjem postojećeg informacijskog sustava i njegove baze podataka novim znanjima

### **1.3 Struktura rada**

U prvom poglavlju opisan je uvod u temu ovog doktorskog istraživanja. Iskazana je motivacija za opisano istraživanje, naveden je cilj istraživanja, hipoteze te znanstveni doprinos istraživanja.

U drugom poglavlju opisan je pregled područja istraživanja. Definirani su osnovni pojmovi važni za ovo istraživanje. Naveden je pregled područja koji pokriva istraživanje u području razvoja baza podataka i informacijskih sustava s aspektima povezanim s poljem umjetne inteligencije, a posebice obrade prirodnog jezika. Osim toga, uključen je pregled područja povezan s eksternim izvorima znanja. Želi se naglasiti da je ovaj pristup drugačiji od svih ostalih pokušaja da se integriraju elementi poslovne inteligencije u informacijske sustave.

U trećem poglavlju opisana je metoda "Node of Knowledge" (NOK metoda) koja je temelj za razvoj novog sustava. Definiran je konceptualni okvir "Node of Knowledge" te pravila prevođenja metode entiteta i veza u NOK metodu i obrnuto.

U četvrtom poglavlju definirana su pravila transformacije rečenica prirodnog jezika u FNOK zapis i pravila transformacije pitanja prirodnog jezika u QFNOK zapis. Svako pravilo popraćeno je primjerima.

U petom poglavlju opisani su model procesa sustava transformacije teksta u bazu podataka, model hijerarhije tekstualnog znanja povezan s modelom rječnika te meta modeli NOK metode za dvosmjernu i jednosmjernu vezu. Na temelju meta modela NOK-a na nekoliko rečenica prirodnog jezika popunjene su tablice u relacijskoj bazi podataka te je demonstrirano postavljanje pitanja i dobivanje odgovora SQL upitima. Za jednosmjernu NOK metodu korišteno je znanje iskazano rečenicama prirodnog hrvatskog jezika, a za dvosmjernu NOK metodu je korišten engleski jezik.

U šestom poglavlju opisan je algoritam za transformaciju FNOK zapisa u relacijsku bazu podataka, algoritam za transformaciju QFNOK zapisa u relacijsku bazu podataka te algoritam za kreiranje odgovora.

U sedmom poglavlju prikazani su rezultati primjene algoritma transformacije na skupu rečenica prirodnog jezika te skupu pitanja prirodnog jezika u sustavu temeljenom na NOK metodi. Izvršena je transformacija rečenica i pitanja u tablice relacijske baze podataka. Prikazani su rezultati algoritma generiranja odgovora na postavljena pitanja (na temelju polaznih rečenica).

U osmom poglavlju opisano je proširenje baze podataka postojećeg informacijskog sustava bazom podataka sustava temeljenog na NOK-u. Prikazano je propitivanje ove integrirane baze podataka pitanjima koja kombiniraju podatke, odnosno znanje iz oba sustava.

U devetom poglavlju nalazi se zaključak te planovi za budući rad.

## 2 Pregled područja

Iako je područje umjetne inteligencije (UI) relativno mlado područje, naslijedilo je mnoge ideje, stavove i tehnike iz drugih znanosti (humanističkih, prirodnih i kognitivnih znanosti). U današnje vrijeme, umjetna inteligencija spaja teorije zaključivanja i učenja (iz filozofije), formalnu logiku, teoriju vjerojatnosti, odlučivanja i računanje (iz matematike), mogućnost istraživanja ljudskog uma i znanstveni jezik u kojem se izražavaju nastale teorije (iz psihologije), teoriju strukture i značenje jezika (iz lingvistike) te alat s kojim pomoću UI može stvoriti stvarnost preuzet iz računalnih znanosti.

Jedan od središnjih problema na području umjetne inteligencije jest razvoj dovoljno precizne i djelotvorne notacije za prikaz znanja u inteligentnom sustavu. Tijekom vremena razvili su se brojni pristupi za predstavljanje znanja. U ovisnosti o zahtjevima, različite aplikacije trebaju različite formalizme i jezike koji se razlikuju po svojoj složenosti i stupnju formalizma.

Odabir prikladnog rječnika i stupnja formalizma, naizmjenice, ovisi o osnovnim konceptima inteligentnog ponašanja [22].

U nastavku su opisani osnovni pojmovi vezani za područje umjetne inteligencije i predstavljanje znanja te objašnjeni osnovni pristupi za sheme za prikaz znanja s naglaskom na grafičke metode. Ukratko je dan i pregled područja obrade prirodnog jezika te sustava pitanja i odgovora. Postavljena je definicija jednostavne rečenice prirodnog hrvatskog jezika te definicija jednostavne rečenice prirodnog engleskog jezika.

### 2.1 Umjetna inteligencija

Da bismo mogli govoriti o umjetnoj inteligenciji, potrebno je definirati što je inteligencija. Inteligencija dolazi od lat. *intelligere* – razabirati, shvaćati, razumijevati. Ne postoji suglasnost oko definicije inteligencije. Većina definicija uključuje koncepte kao što su apstraktno rasuđivanje, razumijevanje, samosvijest, komunikacija, učenje, planiranje i rješavanje problema [121].

Neke od definicija inteligencije su:

- svojstvo uspješnog snalaženja jedinke u novim situacijama (R. Pintner),
- sposobnost učenja prilagodbe na okolinu (Colvin),
- opća sposobnost apstraktnog zaključivanja pri rješavanju problema (Terman),
- urođena opća kognitivna sposobnost (Burt),
- svrsishodno i prilagodljivo ponašanje u danim okolnostima [36].
- Inteligencija se manifestira u odnosu na neki posebni društveni i kulturni kontekst (Weizenbaum).

Naziv "Umjetna inteligencija" (engl. *artificial intelligence*) definirao je J. McCarthy 1956. godine na konferenciji u Dartmouthu. Umjetna inteligencija je inteligentno ponašanje strojeva sa sposobnošću: zaključivanja, prikupljanja i upotrebe znanja, korištenja znanja i izmjenjivanja zamisli te postavljanja problema [106]. U [106] su dane definicije umjetne inteligencije organizirane u četiri kategorije, ovisno o tome stavlja li se naglasak na misaoni proces (razmišljanje) ili na ponašanje te obavlja li se inteligencija u terminima ljudske izvedbe ili racionalnosti. S obzirom na to, definicije se dijele na: "razmišljati ljudski", "razmišljati racionalno", "ponašati se ljudski", "ponašati se racionalno" (tablica 2.1).

**Tablica 2.1. Definicije umjetne inteligencije organizirane u četiri kategorije**

<p style="text-align: center;"><b><i>Razmišljati ljudski</i></b></p> <ul style="list-style-type: none"> <li>- Uzbudljivi novi pokušaj da se omogući razmišljanje računalima ... strojevi s umovima, u punom i doslovnom smislu. (Haugeland, 1985.)</li> <li>- Automatizacija aktivnosti koje asociramo s ljudskim razmišljanjem, poput donošenja odluka, rješavanja problema, učenja ... (Bellman, 1978.)</li> </ul>	<p style="text-align: center;"><b><i>Razmišljati racionalno</i></b></p> <ul style="list-style-type: none"> <li>- Proučavanje mentalnih svojstava kroz uporabu računalnih modela. (Charniak i McDermott, 1985.)</li> <li>- Proučavanje postupaka koji mogućim čine percipiranje, rasuđivanje i reagiranje. (Winston, 1992.)</li> </ul>
<p style="text-align: center;"><b><i>Ponašati se ljudski</i></b></p> <ul style="list-style-type: none"> <li>- Proces stvaranja strojeva koji obavljaju funkcije koje zahtijevaju inteligenciju koju imaju ljudi (Kurzwil, 1990.)</li> <li>- Proučavanje kako učiniti da računala rade stvari u kojima su, trenutno, ljudi bolji (Rich i Knight, 1991.)</li> </ul>	<p style="text-align: center;"><b><i>Ponašati se racionalno</i></b></p> <ul style="list-style-type: none"> <li>- Polje rada koje želi objasniti i emulirati inteligentno ponašanje u smislu računalnih procesa (Schalko, 1990.)</li> <li>- Grana računalnih znanosti koja se bavi automatizacijom inteligentnog ponašanja (Luger i Stubbleeld, 1993.)</li> </ul>

D. W. Patterson (1990.) definira "Umjetna inteligencija je grana računarske znanosti koja se bavi proučavanjem i oblikovanjem računarskih sustava koji pokazuju neki oblik inteligencije. Takvi sustavi mogu učiti, mogu donositi zaključke o svijetu koji ih okružuje, oni razumiju prirodni jezik te mogu spoznati i tumačiti složene vizualne scene te obavljati druge vrste vještina za koje se zahtijeva čovjekov tip inteligencije."

Prema [105], [62] inteligentni sustav je svaki sustav koji pokazuje sljedeća svojstva:

- *Pokazuje prilagodljivo ciljno usmjereno ponašanje* - Željeni cilj ili ciljeve predočiti podciljevima i koristiti znanje o operacijama i postupcima koji prevode željeni cilj u slijed akcija. Ako neki od podciljeva nije ostvariv, sustav traži alternativni put prema konačnom cilju sustava.
- *Uči na temelju iskustva* - Sustav ima algoritme za automatsku modifikaciju strukture i funkcija na temelju iskustva koja stječe u radu - učenje podrazumijeva da sustav može prikupljati, prikazivati i upotrebljavati znanje.
- *Koristi velike količine znanja* - Količina znanja pohranjena u sustavu mora biti slična količini znanja koju posjeduje čovjek da bi riješio takav problem.
- *Pokazuje svojstva svjesnosti* - Sustav ima sposobnost objašnjavanja svojeg ponašanja, nadgledanja i dijagnoze stanja te oporavka u slučaju pogreške.
- *Komunicira s čovjekom prirodnim jezikom i govorom* - Sustav mora komunicirati s čovjekom i drugim inteligentnim sustavima na prijateljski način - zato upotrebljava prirodni jezik i govor. Takva komunikacija podrazumijeva i baratanje dvosmislenostima i gramatički neispravnim rečenicama.
- *Tolerira pogreške i nejasnoće u komunikaciji*
- *Odgovara u stvarnom vremenu*

Osim navedenih svojstava, inteligentni sustav ima i sljedeće funkcije [15]:

- prikupljanje i obrada podataka,
- interakcija s vanjskim svijetom (radnom okolinom),
- komunikacija s čovjekom i/ili drugim inteligentnim sustavima,
- prikupljanje znanja (učenje),
- rukovanje znanjem,
- obrada znanja i zaključivanje,
- planiranje
- i drugi.

Čovjek razvija inteligentne sustave koji mogu računati, govoriti, slati poruke, pamtili podatke, ponovo ih koristiti, raditi koristan rad i drugo. Umjetna inteligencija je općenito bilo kakvo dodavanje bilo kojeg inteligentnog svojstva u sustav [89].

Prema *Association of Computing Machinery* (ACM), područja umjetne inteligencije su:

- Opće područje (kognitivno modeliranje, filozofske osnove),
- Ekspertni sustavi i primjene,
- Automatsko programiranje,
- Dedukcija i dokazivanje teorema,
- Formalizmi i metode prikaza znanja,
- Strojno učenje,
- Razumijevanje i obrada prirodnih i umjetnih jezika,
- Rješavanje problema, metode upravljanja i metode pretraživanja prostora stanja,
- Robotika,
- Računalni vid, raspoznavanje uzoraka i analiza scena,
- Raspodijeljena umjetna inteligencija.

Područje umjetne inteligencije surađuje s mnogim drugim znanostima, primjerice s humanističkim znanostima (lingvistika, filozofija, psihologija), prirodnim znanostima (matematika, biologija), kognitivnom znanostima (interdisciplinarno znanstveno istraživanje uma (računarstvo, neuroznanost, psihologija, lingvistika, antropologija)) i tako dalje.

## 2.2 Znanje

O pitanju "Što je znanje?" raspravljali su filozofi još u antičkoj Grčkoj, ali još uvijek nije precizno određeno što ono jest [13]. Znanje (engl. *knowledge*) je u [43] opisano kao:

- I) poznavanje čega ili spoznaja o čemu (činjenica, stvari);
- II) a. teoretsko ili praktično poznavanje predmeta, jezika i sl. b. ukupnost poznavanja, sustavnost spoznaja tih predmeta; znanost.

U [15] znanje je definirano kao *Spoznaja + Logika*. Spoznaja uključuje nepropozicijsko razumijevanje (npr. kao što je percepcija, pamćenje i refleks), ali i propozicijsko razumijevanje te razumijevanje sudova o nepropozicijskom razumijevanju.

Logika se bavi problematikom valjanosti zaključivanja i konzistentnosti diskursa. Pravilo zaključivanja logički je valjano ako i samo ako uvijek (nužno) čuva istinitost svojih premisa. Logika traži i pronalazi svoje temeljne principe u strukturnim značajkama ljudskog jezika, razmišljanja i razumijevanja. Logički valjano rasuđivanje ne mora nužno dovesti do istinitog zaključka: to ovisi o tome jesu li premise istinite [101].

U [86] je znanje opisano kao skup informacija koji je ljudima razumljiv i koristan za bilo koje ljudsko djelovanje. Autori u [21] znanje opisuju kao fluidnu mješavinu oblikovanoga iskustva, vrijednosti, povezanih informacija i ekspertnih mišljenja koja osigurava okvir za ocjenjivanje i uključivanje novih iskustava i informacija. Znanje potječe i primijenjeno je u umovima znalaca. U organizacijama, znanje je često pohranjeno ne samo u dokumentima ili spremištima (bazama znanja), već i u organizacijskim rutinama, procesima, praksama i normama. Znanje se može prikazati kroz nekoliko aspekata znanja (podatak, informacija, znanje, mudrost), njegove svrhe i konteksta upotrebe [44].

U [51] podatak je definiran kao bilo koji predmet mišljenja koji može prenijeti informaciju; formalizirani znakovni prikaz činjenica, pojmova i naredaba pogodan za priopćavanje, interpretiranje te analognu i digitalnu obradu.

Informacija je u [51] definirana kao:

- osobina podatka kojim se smanjuje neodređenost u sustavu;
- ono što podatak prenosi, ali i osobina podatka (ono što podatak nosi i što je za njega vezano);
- neodređeni pojam čiji je karakter u bliskoj vezi s načinom na koji se informacija prenosi (npr. pismo, govor, slika i sl.);
- informacija se može predočiti kao značenje što se pridodaje podacima pomoću konvencija koje se koriste za njihovo tumačenje.

Mudrost je pojam kojim se označuje posebna kvaliteta inteligencije, znanja i djelovanja, ... istoznačnica za pamet, premda ova označava sposobnost za mudrost, ali ne još i samu mudrost. U starih Grka mudrost (*sofía*) je odlika filozofa koji teži istinskom ... [99].

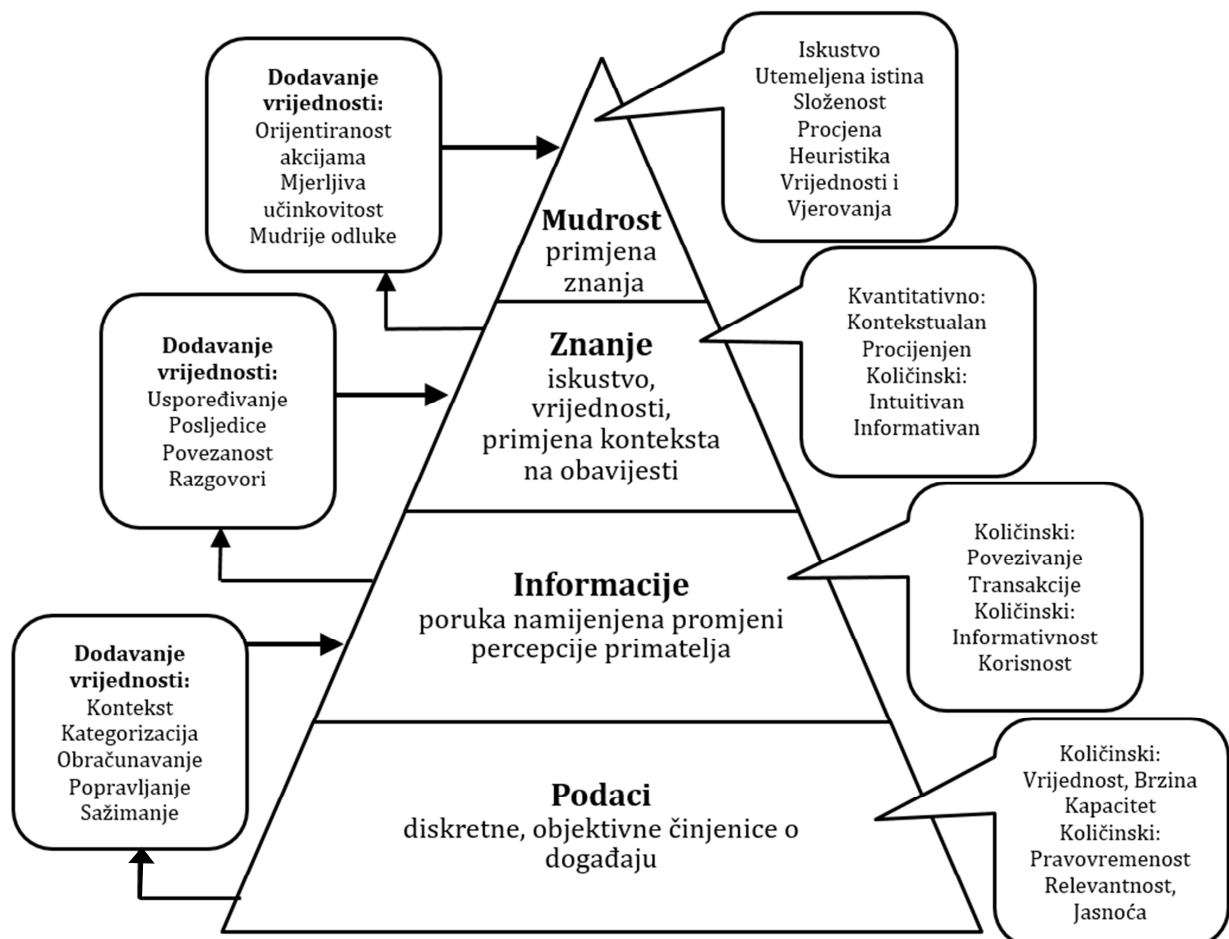
Modeli prikaza znanja (podatak – informacija – znanje – mudrost (engl. *data – information – knowledge – wisdom*, skraćeno *DIKW*) ne uključuju uvijek sve četiri komponente. Ranije verzije nisu uključivale podatke, novije verzije izostavljaju mudrost, a neki modeli uključuju neke dodatne elemente. Podatak, informacija i znanje tri su osnovna koncepta u kontekstu informacijskih znanosti. Akademska i profesionalna IS literatura pruža raznolika mišljenja o svakom konceptu. Očigledno, priroda relacije između njih je sporna kao i njihovo značenje [21]. Povezanost podataka, informacija, znanja i mudrosti moguće je hijerarhijski prikazati piramidom (slika 2.1) [9], [60]. Bazu

piramide tvore podaci, nakon toga hijerarhijski slijede informacija, zatim znanje i na vrhu mudrost [9].

Iako slika 2.1 prikazuje model znanja predstavljen hijerarhijom ili piramidom, taj se model često može prikazati i kao [122]:

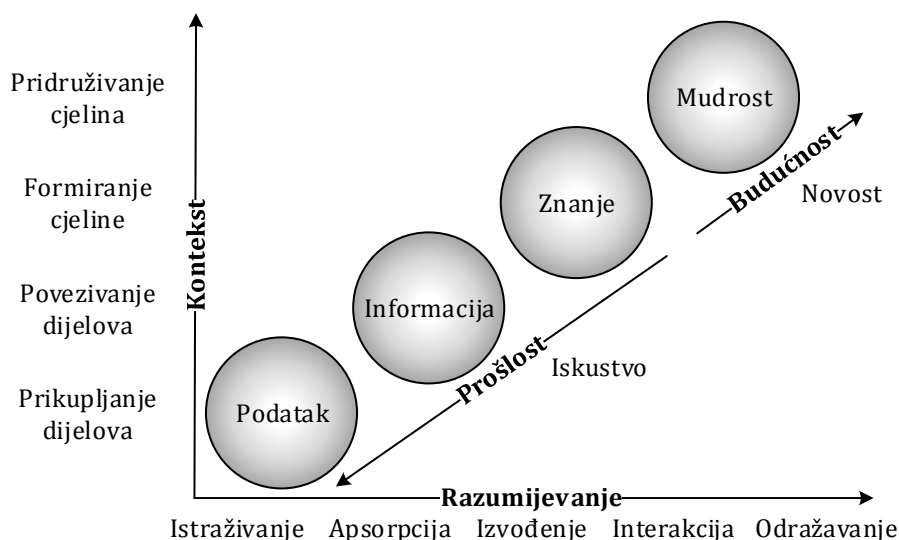
- Lanac (Lievesley, Denise; Data Information Knowledge Chain; 2006),
- Okvir (Chisholm, James; Warmnan, Greg; Experimental Learning in Change Management; 2007),
- Kontinuum (Choo, Chun Wei; Detlor, Brian; Turnbull, Don; Web Work: Information Seeking and Knowledge Work on the World Wide Web; 2006).

Slika 2.2 prikazuje jedan od pogleda na DIKW hijerarhiju.



Slika 2.1. Hijerarhijski prikaz znanja [9], [60]





Slika 2.2. Jedan pogled na DIKW hijerarhiju (Clark, 2004.)

Znanje proizlazi iz uma prilikom rada. To je ukupnost svega što je bilo spoznato, otkriveno ili izvedeno zaključivanjem. Znanje je neophodno za dobra odlučivanja, tj. za prepoznavanje i razumijevanje uzročno/posljedičnih odnosa koji utječu na poslovanje organizacije, a time i na sposobnost predviđanja njezine budućnosti [120]. U [33], [47] navodi se sedam glavnih značajki znanja:

1. Znanje ima praktičnu svrhu - znanje pomaže u rješavanju praktičnih problema.
2. Znanje može biti vezano uz osobu, a i ne mora - znanje dolazi u dva oblika - osobno (znanje koje posjeduje neka osoba) i predstavljeno (npr. kroz tekst, sliku, zvuk i bilo što drugo što može nositi znanje).
3. Znanje ima normativnu strukturu - znanje se sastoji od prihvaćenih izjava.
4. Znanje je interno povezano - interna struktura znanja se sastoji od dijelova koji su međusobno povezani.
5. Znanje je izvana povezano - znanje je povezano s drugim znanjem.
6. Znanje je dinamično - znanje se mijenja (ne samo da raste, već se odbacuje, prepoznaje, koristi, zapisuje, prenosi, dijeli, preformulira, itd.).
7. Znanje ima institucionalni kontekst - institucije se brinu o generiranju, formiranju i distribuciji znanja.

Prikaz znanja treba olakšati njegovu praktičnu primjenu, distribuciju među ljudima (ali i računalima te ljudima i računalima), treba moći prikazati izjave iz kojih se znanje sastoji, njihovu međusobnu povezanost, povezanost jednoga znanja s drugim, treba omogućiti mijenjanje znanja te olakšati institucijama generiranje, formiranje i distribuciju znanja.

Znanje se može prikazati na bilo koji način koji može nositi znanje - u biti, način kojim se znanje prikazuje predstavlja nositelja znanja. Jedan od nositelja znanja je tekst [47].

### 2.3 Predstavljanje znanja

Predstavljanja znanja (prikaz znanja, reprezentacija znanja, engl. *knowledge representation*, KR) počelo se razvijati u okviru područja umjetne inteligencije sedamdesetih godina prošlog stoljeća. To je područje odigralo ogromnu ulogu u razvoju umjetne inteligencije i ostalo je kao jedno od najjačih potpodručja umjetne inteligencije [74]. Bavi se pronalaženjem načina na koji se informacije (i znanje) mogu formalno opisati. Formalni opis podrazumijeva prikaz u nekom jednoznačnom jeziku ili notaciji koja ima dobro formiranu sintaksu i semantiku [13], [39], [106].

Sintaksa specificira skup pravila za kombiniranje simbola i formiranje izraza reprezentacijskog jezika. Na osnovu definiranih pravila moguće je jednoznačno odrediti je li izraz dobro formiran, odnosno može li mu se pridružiti značenje.

Semantika specificira kako konstruirani izrazi trebaju biti interpretirani, odnosno kako izvesti značenje iz forme. Značenje se prvo pridružuje atomarnim objektima jezika, a potom se proširuje na složenije jezične strukture.

Gledano iz perspektive područja umjetne inteligencije, neki od problema koji se pojavljuju u predstavljanju znanja su [53]:

- Kako ljudi predstavljaju znanje?
- Koja je priroda znanja?
- Treba li shema za prikaz znanja baratati nekom specifičnom domenom, ili bi trebala biti opće namjene?
- Koliko je ekspresivna shema za prikaz znanja?
- Treba li shema za prikaz znanja biti deklarativna ili proceduralna?
- Jedan od središnjih problema na području umjetne inteligencije jest razvoj dovoljno precizne i djelotvorne notacije za prikaz znanja u inteligentnom sustavu.

Rezultat istraživanja na području predstavljanja znanja je niz reprezentacijskih jezika, odnosno programskih jezika koji su orijentirani k organizaciji opisa objekata i ideja. U okviru reprezentacijskih jezika znanje se može predstaviti kroz skup sintaktičkih i semantičkih pravila koja omogućuju opisivanje objekata.

Razvijene su različite tehnike reprezentacije znanja, kao i različiti jezici. Za predstavljanje znanja razvijeni su različiti pristupi, formalizmi, metode i jezici koji variraju po složenosti (jednostavniji do složenih) te po semantici (od manje semantički bogatih do vrlo ekspresivnih) [48].

Scheme za prikaz znanja (engl. *knowledge representation scheme*) moraju zadovoljiti sljedeće zahtjeve [15]:

- Primjerenost prikaza (engl. *representational adequacy*): sposobnost prikaza svih vrsta znanja koja su potrebna za određeno područje,
- Primjerenost zaključivanja (engl. *inferential adequacy*): sposobnost baratanja prikazanim strukturama na takav način da se izvode nove strukture koje odgovaraju novom znanju dobivenom zaključivanjem na temelju starog znanja,
- Djelotvornost zaključivanja (engl. *inferential efficiency*): mogućnost ugradnje dodatne informacije u strukturu znanja koja se može koristiti za usmjeravanje pažnje mehanizma zaključivanja prema obećavajućim smjerovima,
- Djelotvornost učenja (engl. *acquisitional efficiency*): sposobnost lakog prikupljanja znanja korištenjem automatizama (generiranje novog znanja).

Znanje se simbolički može prikazati na mnogo načina [18]. Scheme za prikaz znanja možemo podijeliti u 4 osnovna pristupa [15], [52]: mrežne scheme, logičke scheme, proceduralne scheme i teorije okvira. Tablica 2.2 prikazuje, ovisno o osnovnim pristupima, kategorizaciju shema za prikaz znanja te neke tehnike i programske jezike [15].

**Tablica 2.2. Podjela shema, tehnika i jezika za prikaz znanja prema osnovnim pristupima**

<b>Mrežne scheme</b>	<b>Logičke scheme</b>	<b>Proceduralne scheme</b>	<b>Strukturne scheme (Okviri)</b>
Semantičke mreže	Propozicijska logika	Produkcijski sustav	Sustav okvira
SCHOLAR	Predikatna logika	PLANNER	FRL
Dijeljene semantičke mreže	Logičko programiranje	SHRDLU	KRL
Strukturni opisi	Nemonotona logika		KL-ONE
Propozicijske mreže NETL	NeizrAZita logika		
KRP			

Logičke scheme prikazuju znanje koristeći matematičke ili ortografske simbole, pravila zaključivanja i temelje se na precizno definiranoj sintaksi i semantici [97]. Koriste se ako je ishodišna točka istinitost izjava o stanjima. U logičke scheme ubrajaju se propozicijska

logika, predikatna logika te drugi logički formalizmi (nemonotona logika, fuzzy logika, logika višeg reda i slično). Realiziraju se logičkim programskim jezicima: Prolog, HiLog, F-logika i slično. Logičke sheme ponekad nisu najbolji odabir za prikaz znanja jer nisu toliko prirodne i zaključivanje ponekad nije učinkovito. Ponekad su bolji jezici koji su više specijalizirani [13], [39], [106].

Kod proceduralnih shema, znanje je prikazano kao skup instrukcija za rješavanje problema. To omogućuje jednostavnu modifikaciju baze znanja i odvajanje baze znanja od mehanizama zaključivanja. Za proceduralne sheme vežemo produkcijske sustave, programski jezik PLANNER, jezik CLIPS.

Mrežne sheme koriste grafove za prikaz znanja. Čvorovi grafa označavaju objekte ili koncepte u domeni, a lukovi definiraju vezu između objekata, njihove atribute i vrijednosti atributa. U mrežne sheme pripadaju semantičke mreže i propozicijske mreže.

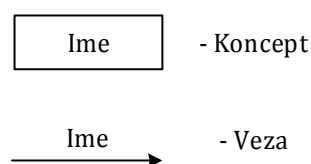
Teorije okvira proširuju mrežne prikaze označavanjem svakog čvora u grafu kao složenu podatkovnu strukturu. Ovdje se ubrajaju: okviri, jezici okvira (KL-ONE) i slično.

Osim četiri osnovna pristupa, postoje i drugi, složeniji pristupi i tehnike za prikaz znanja. Neki od njih su: objektno orijentirani pristup, deskriptivna logika, taksonomije, ontologije i drugi.

U nastavku će biti objašnjene samo neke od shema za prikaz znanja.

### 2.3.1 Semantičke mreže

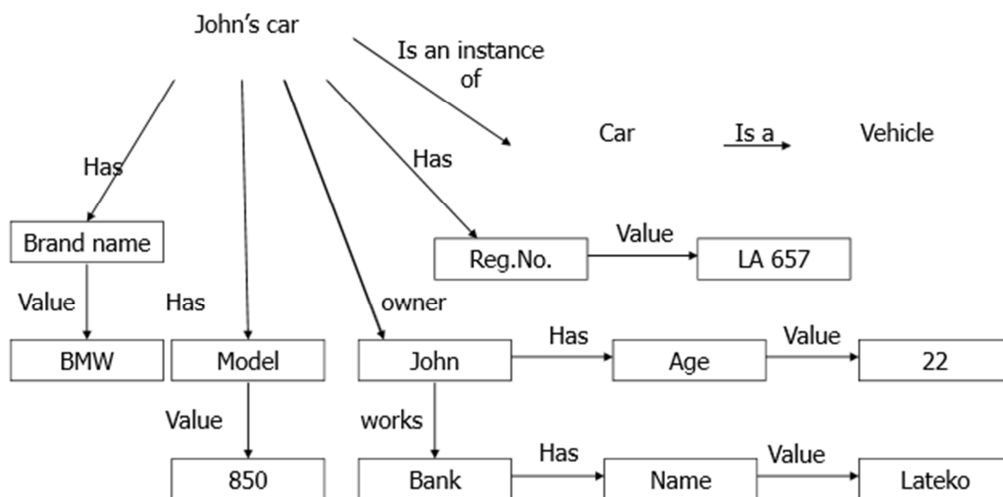
Semantičke mreže (engl. *semantic networks*) (autor: Quillian, 1968.) je shema za prikaz znanja koja prikazuje znanje kao graf. Čvorovi označavaju objekte ili koncepte, njihova svojstva i odgovarajuće vrijednosti. Lukovi označavaju relacije između čvorova. Čvor i luk uglavnom su imenovani (lukovi imaju težinu). Čvorovi mogu predstavljati koncepte, objekte, događaje, značajke, vrijeme i tako dalje [52], [32]. Slika 2.3 prikazuje osnovne koncepte semantičkih mreža.



Slika 2.3. Osnovni koncepti semantičkih mreža

Sve vrste formalizama semantičkih mreža koncentriraju se na izražavanje taksonomne strukture kategorija objekata i relacija između njih [35].

U okviru prikaza znanja ključni koncept predstavlja relacija. Kod prikaza znanja primjenom semantičkih mreža, relacija između objekata prikazuje se kao oznaka brida koji spaja ta dva objekta (vrha grafa). Dvije relacije koje se ističu kao specifične relacije semantičkih mreža jesu 'is-a' relacija i 'a-kind-of' relacija. Relacija *ISA* predstavlja koncept 'is-a' ('biti-istanca-od'). Koristi se za povezivanje pojedinačnih objekata u klasu. ISA relacijom iskazuje se pripadnost instance klasi. Relacija *AKO* uspostavlja međusobne odnose između klasa, 'a-kind-of'. Može se promatrati kao relacija na višoj razini od ISA relacije. Prikazuje hijerarhijsku strukturu klasa. *ISA* i *AKO* relacije definiraju hijerarhijsku strukturu te pripadnost objekta (instance) klasi [68], [52]. Slika 2.4 prikazuje primjer semantičke mreže [52].



Slika 2.4. Primjer semantičke mreže [52]

### 2.3.2 Produkcijski sustavi

Produkcijski sustavi (engl. *production systems, rule-based systems*) (autor: E. Post, 1943.) najčešći su način prikaza ljudskog znanja u obliku *ako-onda* pravila (engl. *if-then rules*) [95]. Različita su područja primjene, primjerice u medicini, bankarstvu, projektiranju računarskih sustava, geologiji,... Primjeri produkcijskih sustava su DENDRAL [23], MYCIN [76], R1/XCON [124]. Prednost produkcijskih sustava jest mogućnost prikazivanja pravila koja koristi da bi se došlo do zaključaka. Nažalost, oni mogu efikasno rješavati probleme samo ako su osigurana opsežna, specifična znanja visoke kvalitete koja se odnose na uski problem u domeni [50].

### 2.3.3 Okviri

Okviri (engl. *frames, schemata*) (autor: M. Minski, 1975.) uvedeni su kako bi se riješio nedostatak semantičkih mreža, tj. njima je uvedena mogućnost različite interpretacije istog čvora. Na primjer, čvor "pas" može predstavljati klasu svih pasa ili nekog određenog psa. Kod okvira je jasna razlika između klase i instance klase. Okviri omogućuju detaljniji prikaz od semantičkih mreža. Okviri se sastoje od atributa ("*slots*") za koje se određuju ili računaju vrijednosti ("*filters*"). Vrijednosti atributa mogu biti skalarne vrijednosti ili reference na druge okvire ili procedure. Okviri koji se smatraju podređenim okvirima ('*subframes*') nasljeđuju svojstva (vrijednosti atributa, restrikcije na vrijednosti atributa) od okvira koji se smatraju nadređenim okvirima ('*superframes*') i to prema pravilima hijerarhije definiranim u okviru strategije nasljeđivanja [95]. Jezici okvira prvenstveno se koriste za prepoznavanje i opisivanje klasa objekata. Okvir podrazumijeva neka zadana svojstva u koja se objekt mora uklopiti. Slog se može smatrati čvorom u grafu koji u jednom slotu ima ime objekta kojeg opisuje, a u ostalim slotovima pohranjuju se vrijednosti atributa. Svojstva (atributi) se mogu naslijediti od okvira na višim razinama ili mogu biti ispunjeni nekom novom vrijednošću atributa (time se prikazuju iznimke). Tablica 2.3 prikazuje primjer okvira.

Tablica 2.3. Primjer okvira [68]

Naziv	Mačka
Je (ISA)	Sisavac Životinja
Svojstva	Ime Omiljeno_mjesto_boravka Težina Boja_krzna Boja_oči Starost
Defaults (unaprijed postavljene vrijednosti)	Omiljeno_mjesto_boravka Košara kraj peći Težina 3 kg Boja_oči zelena

### 2.3.4 Objektno orijentirani pristup

U razvoju jezika za prikaz znanja aktualan je objektno orijentirani pristup [95]. Klasični modeli za prikaz znanja (na primjer predikatna logika) povezuju se s objektno orijentiranim pristupom. Neke strukture za prikaz znanja (npr. semantičke mreže, sustavi okvira) sadrže u sebi elemente objektno orijentirane paradigme. Neki objektno

orijentirani jezici koji se koriste na području AI su: Comon LISP Object Systems (CLOS), SmallTalk, F-logika, C++ [68].

### **2.3.5 Deskriptivna logika**

Deskriptivna logika (engl. *description logic*, DL) podrazumijeva familiju jezika za prikaz znanja. Razvijene su kako bi se prikazalo značenje riječi [77]. Objedinjuje ideje sustava okvira, semantičkih mreža, jezika za reprezentaciju znanja (kao što je KL-ONE), objektno orijentirane reprezentacije, semantičkih modela i sustava tipova. Teorija deskriptivne logike sastoji se od izjava o konceptima, pojedinostima i njihovim relacijama. U logici prvog reda pojedinosti odgovaraju konstantama, a koncepti odgovaraju unarnim predikatima [35].

### **2.3.6 Ontologije**

Ontologije su početkom devedesetih postale popularna istraživačka tema. U novije vrijeme notacija ontologija počela se jako rasprostranjivati na različitim poljima. Razlog tome je zajedničko i općenito razumijevanje nekih domena s kojima može biti komunicirano preko ljudi i računala [114]. Ontologija (engl. *ontology*) u računarskim znanostima podrazumijeva način predstavljanja znanja kojim se oblikuje promatrana domena imenovanjem koncepata u domeni i formiranjem relacija među njima [98]. Danas su ontologije široko prihvaćene u uporabi u mnogim konkretnim primjenama i predstavljaju svojevrsan standard kod izrade baze znanja u sustavima za potporu pri odlučivanju. Ontologijama se opisuju entiteti iz određene domene, njihova svojstva i veze. Ontologije omogućuju podjelu znanja i zajedničko razumijevanje ljudi, ali i aplikacijskih sustava unutar neke određene domene [32].

## **2.4 Grafičke metode za predstavljanje znanja**

Predstavljanje znanja i zaključivanje bavi se problemom kako znanje simbolički prikazati te njime manipulirati na automatizirani način zaključivanjem programa [13] i uključuje strojno interpretiranu reprezentaciju [35]. Neke od grafičkih metoda za prikaz znanja su:

- Basic Conceptual Graphs (BCG),
- Multi-layered extended semantic networks (MULTINET),

- Hierarchical Semantic Form (HSF),
- Resource Description Framework (RDF) i
- Nodes of Knowledge (NOK).

U nastavku će biti ukratko opisana svaka od tih metoda. Sve navedene metode mogu grafički prikazati znanje. Takav graf se sastoji od čvorova i veza. Čvorovima se prikazuju koncepti, a veza među čvorovima koristi se za predstavljanje odnosa među konceptima [18], [40], [113] i [45].

#### **2.4.1 Basic Conceptual Graphs**

Čvorovi u *Basic Conceptual Graphs* (BCG) dijele se na konceptualne i relacijske koji su međusobno povezani vezama. Svaki čvor ima svoj naziv i grafički se prikazuje pravokutnikom (ako se radi o konceptualnom čvoru), odnosno elipsom (ako se radi o relacijskom čvoru). Relacijski čvor povezuje konceptualne čvorove i određuje smjer čitanja dijagrama. Veza među njima posjeduje broj prema kojemu se zna redoslijed konceptualnih čvorova u nekoj relaciji, a koji je važan prilikom čitanja znanja [48], [18].

#### **2.4.2 Multi-layered extended semantic networks**

U metodi *Multi-layered extended semantic networks* (MULTINET), svaki čvor pripada nekoj klasi iz predefinirane klasifikacije čvorova koja se u metodi naziva konceptualna ontologija (ukupno se sastoji iz 29 klasa). Svaki čvor posjeduje 7 predefiniranih atributa, čije vrijednosti ga smještaju u semantički prostor. Među čvorovima se uspostavljaju veze primjenom neke od 89 predefiniranih vrsta veza. 16 od njih pripada u tzv. kognitivne uloge i to su one veze koje opisuju odnos između glavnih sudionika u prikazanoj situaciji [40]. Metoda MULTINET je izrazito složena zbog velikog broja elemenata koji se trebaju definirati za svaki čvor [48].

#### **2.4.3 Hierarchical Semantic Form**

*Hierarchical Semantic Form* (HSF) sastoji se od dva osnovna koncepta: grupe (engl. *group*) i veze (engl. *link*). Grafički su predstavljeni kružnicama (prazna za grupe, puna za veze) koje se međusobno povezuju strelicama (kako bi se ukazao smjer čitanja). Koncept grupa se koristi u označavanju pojedinoga znaka, grupe znakova, riječi, semantičkih kategorija i složenih uzoraka. Koristi se kako bi se prikazao slijed (engl. *sequence*)



označenih pojmova na različitoj razini apstrakcije (grupa prikazuje vezu - link na prvi element slijeda). Ista grupa se može pojaviti na različitim razinama apstrakcije [48], [113]. Metoda HSF pojedinom slijedu pojmova daje semantiku tako što ih veže na koncept grupa koji u sebi čuva semantički opis (npr. "je dio" ili "dio dana" i sl.). Metoda HSF posjeduje algoritam SOUL koji joj daje sposobnost učenja novih uzoraka [113].

#### **2.4.4 Resource Description Framework**

*Resource Description Framework (RDF)* je metoda koja prikazuje odnos među web resursima primjenjujući imenovano svojstvo i njegovu vrijednost. Kao web resurs ovdje se pojavljuju razni podaci, dokumenti, slike i slično. Imenovano svojstvo se odnosi na promatrani resurs. Ime svojstva se definira preko *Uniform resource identifier* (URI) i zbog toga se samo svojstvo može promatrati kao novi resurs. Vrijednost svojstva može biti drugi resurs ili neki podatak (npr. string, broj, datum i slično). RDF metoda posjeduje više načina prikaza, a jedan od njih je RDF graf koji se sastoji od tri osnovna koncepta: čvora koji predstavlja resurs, čvora koji predstavlja podatak i linije sa strelicom koja povezuje čvorove. Elipsa je grafički simbol za resurs, dok je pravokutnik grafički simbol za podatak. Linija sa strelicom predstavlja svojstvo i na njoj se upisuje naziv svojstva. Strelica ukazuje na vrijednost svojstva. Budući da resursi imaju svojstva, uvijek se na strani linije bez strelice nalazi elipsa (čvor za resurs), dok se na strani linije sa strelicom nalazi elipsa ili pravokutnik (dakle čvor za resurs ili podatak). RDF metodom se mogu prikazati izjave sa strukturom subjekt, objekt i predikat [48], [45].

#### **2.4.5 Nodes of Knowledge (NOK)**

Metoda za modeliranje čvorova (engl. *Node of Knowledge method* (skraćeno *NOK*)) je nova metoda prikaza znanja na kojoj se zasniva ovo istraživanje.

NOK će detaljnije biti objašnjen u poglavlju 3.

### **2.5 Obrada prirodnog jezika (engl. *natural language processing (NLP)*)**

Obradom teksta prirodnog jezika i izvlačenjem znanja iz teksta bavi se grana Natural Language Processing (NLP). U ovom poglavlju dan je pregled područja koji pokriva istraživanja u području razvoja baza podataka i informacijskih sustava s aspektima povezanim na polje umjetne inteligencije, a posebice obrade prirodnog jezika. Dodatno,

uključen je pregled područja povezan s eksternim izvorima znanja, kao što su rječnici i WordNet. Treba naglasiti da je ovaj pristup drugačiji od svih ostalih pokušaja da se integriraju elementi poslovne inteligencije u informacijske sustave.

Na području obrade prirodnog jezika razvijene su brojne metode i alati za obradu teksta. NLP se bavi problemom izvlačenja znanja iz teksta i postoje brojni znanstveni članci posvećeni tom polju. Već se u ranijim istraživanjima pokazala potreba za povezivanjem baza podataka s područjem obrade prirodnog jezika. Tako, na primjer, postoje istraživanja vezana uz postavljanje upita u prirodnom jeziku [54], [81]. Ti sustavi rade tako da prevode zahtjev za podacima iz upita u SQL tako da relacijska baza podataka može dati traženi odgovor. Primjer ekspertnog sustava koji prima upite korisnika iskazane prirodnim jezikom opisan je u [46]. Međutim, problem koji još uvijek nije riješen je zapisivanje rečenica prirodnog jezika u relacijsku bazu podataka na način da se očuva semantika rečenica.

Stalno se nastoji uvesti elemente umjetne inteligencije u domenu informacijskih sustava i baza podataka. Puno truda uloženo je u transformaciju izraza prirodnog jezika (pitanja) u SQL upite koristeći inteligentno sučelje [54], [81], [78], [58], [38]. Cilj takvih sustava je razumijevanje izraza prirodnog jezika. U [78] je opisano inteligentno sučelje koje koristi tehnike semantičke podudarnosti za pretvaranje upita prirodnog jezika u SQL korištenjem rječnika i skupa pravila. Sva ostala istraživanja na tom polju ograničena su na problem razumijevanja izraza prirodnog jezika u obliku pitanja i njegovo prevođenje u SQL upite. Fokus ovog istraživanja je razvoj okvira (frameworka) koji može prevesti bilo koje (općenito) tekstom izraženo znanje u podatke pohranjene u relacijskoj bazi podataka.

Na polju relacijskog rudarenja podacima (engl. *relational data mining (RDM)*) mogu se pronaći istraživanja koja se bave navođenjem modela ili uzoraka iz višestruko-povezanih podataka [94]. Ovo je potpuno drugačije istraživanje od istraživanja prezentiranog u ovom radu.

U velikom broju radova postoje opisi sustava za automatsku transformaciju prirodnog jezika u formalne specifikacije [30], [55], [19]. Ponekad se koriste i formalizmi konceptualnih grafova kao korak koji prethodi transformaciji u formalni zapis [30]. NOK metoda također koristi grafički prikaz tekstualnog znanja, međutim, njen cilj je

općenitiji, formalizirati bilo koje tekstualno znanje i prevesti ga u relacijski model baze podataka.

U [65] i [108], autori rješavaju problem automatskog integriranja nestrukturiranih tekstova u multi-relacijske baze podataka. Međutim, oni koriste statističke modele za rješavanje ovog problema. U [108] prikazan je sustav koji koristi state-of-the-art statističke modele za ekstrakciju strukture i podudaranje. U [65] koriste se statistički modeli za ko-referentnu rezoluciju i ekstrakciju informacija u bazama podataka. U usporedbi s tim, NOK metoda uopće ne koristi statističke pristupe i ovdje je predložen potpuno drugačiji pristup.

Eksterna znanja, obično iskazana ontologijama domena (engl. *domain ontology*), treba povezati sa sustavima temeljnim na znanju. Velik rast repozitorija tekstualnih podataka i ontologija dogodio se u području medicine i biologije. Na taj način većina informacija je ugrađena u nestrukturiran ili polu-strukturiran tekst. Razvijeni su različiti sustavi za dubinsku analizu teksta koji koristeći NLP tehnike i *co-occurrence-based* analize, otkrivaju moguće odnose među podacima [1], [125]. Navedena istraživanja imaju svoju primjenu u uskoj stručnoj domeni medicine i biologije, za razliku od općeg pristupa koji donosi NOK metoda. Ovdje se predlaže okvir koji može integrirati bilo koji jednojezični rječnik s relacijskom bazom podataka. Suprotan pristup tome je ekstrakcija ontologija iz relacijske baze podataka kao što je opisano u [107]. Autori predlažu pristup ekstrakciji ontologije na vrhu relacijske baze podataka ugrađujući koncept hijerarhije kao "pozadinsko" znanje.

Rječnici su dragocjeni izvor znanja, posebno semantičkog znanja (semantičkih veza kao sinonimi, hiponimi, hiperonimi, antonimi i slično). Postojeća istraživanja rijetko koriste rječnik kao dodatni izvor znanja. Češće se koristi WordNet koji se može smatrati rječnikom ili ontologijom. WordNet je velika leksička baza podataka u kojima se imenice, glagoli, pridjevi i prilozi grupiraju u skupove kognitivnih sinonima (*synsets*), gdje svaki izražava poseban koncept [29], [72]. *Synsetsi* su međusobno povezani pomoću konceptualno-semantičkih i leksičkih odnosa. Istraživanja u području umjetne inteligencije i NLP često koriste WordNet. WordNet se koristi kao vanjski izvor znanja, međutim, ima nekoliko nedostataka. WordNet nije definiran za sve jezike, a ponekad nisu uključene ni sve riječi pojedinog jezika. Primjerice, za hrvatski jezik ne pokriva sve riječi

nego uključuje samo mali podskup [102]. Jednojezični rječnik čini se kao dobra zamjena za WordNet, a uključuje još više znanja o entitetima.

U ovom se radu predlaže integracija rječnika kao vanjskog izvora znanja s relacijskom bazom podataka. U [69], [70] prikazana je formalizacija mogućih semantičkih odnosa izraženih u rječniku. Za potrebe razvijenog sustava, model rječnika je prikazan modelom entiteta i veza. Iako postoje sustavi koji koriste WordNet kao vanjski izvor znanja, htjeli smo u sustavu imati mogućnost integracije bilo kojeg rječnika (na bilo kojem jeziku) s relacijskom bazom podataka. U [59] autori predlažu pristup stvaranju sučelja prirodnog jezika temeljenog na WordNet-u za relacijske baze podataka koja omogućuje krajnjim korisnicima pristupanje i propitivanje informacija u bazi podataka korištenjem pitanja prirodnog jezika. Njihov pristup integrira WordNet kao osnovni leksikon te ontologiju kao bazu znanja semantičkog interpretera.

## **2.6 Sustavi pitanja i odgovora (engl. *question – answering systems (QA systems)*)**

S obzirom da podsustav "Generiranja odgovora na pitanja", sustava temeljenog na konceptualnom okviru NOK možemo smatrati QA sustavom, ukratko ćemo prikazati pregled ovog područja. Primjere QA sustava usporedit ćemo s QA sustavom opisanim u ovom radu u poglavlju 6.

Sustavi pitanja i odgovora (engl. *question – answering systems (QA systems)*) pronalaze točne i sažete odgovore na pitanja postavljena prirodnim jezikom [41]. QA sustavi se odnose na pretraživanje informacija i obradu prirodnog jezika u okviru izgradnje sustava za automatsko odgovaranje na pitanja postavljena prirodnim jezikom. Za pronalaženje odgovora mogu koristiti različite strukturirane baze podataka, zbirke dokumenata na lokalnoj razini, interne dokumente poduzeća, mreže novinskih izvješća, Internet i slično [117].

Istraživanja o QA sustavima započela su 1960-ih godina [73], [24], a intenzivnija istraživanja započela su ranih 1970.-ih godina [126]. U osamdesetima je interes slabio. Jedan od razloga za to je i činjenica da istraživanja u polju umjetne inteligencije nisu osigurala odgovarajuću tehnologiju potrebnu za razvoj QA sustava [126]. Novija istraživanja u tom polju, s različitim uspjehom evaluacije odgovora, objavljuju se od

1999. godine u sklopu Text Retrieval Conference (TREC) čime su stvorene snažne interesne zajednice i ubrzan je napredak u istraživanju QA sustava [41].

U posljednjih 50 godina istraživanja o QA sustavima, razvijeni su brojni sustavi (s različitom razinom uspješnosti). Također, razvijeni su i različiti načini prikaza tekstualnog znanja. Neki implementirani QA sustavi su rezultat velikih projekata (velik broj sudionika, velika baza znanja, dugoročni razvoj sustava, velike investicije) [90]. Primjeri QA sustava su:

- KRAKEN System [67], [83]. Temelji se na CYC projektu i koristi CycL jezik i CYC ontologiju s oko 190,000 koncepata povezanih u hijerarhiju u preko 2,200,000 navoda na temelju kojih sustav djeluje [57]. Jezik CycL je deklarativan jezik temeljen na logici prvog reda sa značajkama višeg reda. Ne prikazuje internu semantiku rečeničnih dijelova već je semantički sadržaj proširen hijerarhijskim sustavom znanja. Na taj način, sadržaj rečenice interpretiran je iz šire perspektive znanja [90]. Nasuprot tome, u NOK-u je semantička povezanost rečeničnih dijelova očuvana.
- OMPS – Open Mind Common Sense [111]. Sustav dizajniran za stjecanje i izgradnju zdravorazumskog znanja. Više od 15 000 suradnika surađuje na izgradnji ovog sustava. Sakupljeno je više od milijun činjenica na engleskom jeziku. Razvijene su različite aplikacije nad bazom znanja. Sustav indukcijom određenih rečenica razvija različite uzorke koji interpretiraju rečenice i koji se koriste kao nova znanja. OPMS koristi prirodni jezik kako bi se omogućilo da što više ljudi unosi ljudsko znanje u bazu znanja. Razvijen je rječnik, a uključeno je i kontekstualno znanje i hijerarhijski odnosi između koncepata [90].
- NOK ne koristi uzorke. Identifikacija semantičke veze između rečeničnih dijelova koristi se za interpretaciju. U osnovi, korištenje uzoraka predstavlja generalizaciju rečenica i potencijalno može rezultirati gubitkom semantike sadržane u rečenicama [90].
- Trips system [3], [2]. To je pomoćnik suradničkom planiranju u kriznim situacijama. Interakcija između korisnika i sustava odvija se kroz dijalog na prirodnom jeziku. Korisnici uz postavljanje pitanja mogu sustavu davati i naredbe. U sebi pohranjuje plan koji se odnosi na informacije, strukture hijerarhijskih zadataka, eksplicitne prikaze mogućih rješenja te privremene baze

znanja (reprezentacija riječi u vremenu u odnosu na različita rješenja). NOK se trenutno fokusira na zajedničku interakciju između korisnika i sustava, ali u formi postavljanja pitanja i davanja odgovora [90].

Osnovno pitanje na koje istraživanja QA sustava pokušavaju dati odgovor je kako točno odgovoriti na korisnikovo pitanje postavljeno prirodnim jezikom. Prikaz tekstualnog znanja ima važnu ulogu u odgovaranju na ovo pitanje. Neka rješenja ovog problema proizlaze iz Text-Meaning-Representation Theory (sustav DIONISUS) [80] i Conceptual Dependency Theory (C-diagram) [109], [63]. U radu [90] ove su teorije uspoređene s NOK-om.

## 2.7 Jednostavne rečenice i pitanja

Sintaksa (grč. *syntaxis* – red, slaganje, uređivanje, razmještaj, namještanje) označava dio gramatike u kojem se proučavaju odnosi među jedinicama u nizu, odnosno odnosi među riječima i njihovim oblicima, odnosi među spojevima riječi, odnosi među rečenicama (surečenicama ili klauzulama) u složenoj rečenici te odnosi među rečenicama u tekstu [110]. Najmanja jedinica kojom sintaksa barata jest riječ, koja sama ili s drugim riječima tvori rečenicu. Rečenica je u sintaksi temeljna jezična jedinica [8].

S obzirom na priopćajnu svrhu koju želimo ostvariti razlikujemo [115]:

- izjavne rečenice – rečenice kojima se izjavljuje da nešto jest ili nije (oznaka na kraju izjavne rečenice je točka),
- upitne rečenice – rečenice kojima se nešto pita (oznaka na kraju upitne rečenice je upitnik),
- usklične rečenice – rečenice kojima se izražava osobita naglašenost i osjećajnost sadržaja (oznaka na kraju izjavne rečenice je uskličnik) i
- poticajne rečenice – rečenice kojima se izriče poticaj (nazivaju se još i zapovjedne ili željne rečenice). One izriču savjet, zapovijed, zabranu, opomenu, upozorenje, želju, tj. poticanje da se što dogodi ili ne dogodi. Oznaka na kraju poticajne rečenice je uskličnik.

Upitne rečenice tvore se pomoću upitnih zamjenica (npr. *tko, što*), upitnih priloga (npr. *zašto, otkud*) i upitnih čestica (npr. *li, zar, da*). Upitnost rečenica u govoru se izražava intonacijom, a u pismu se na kraj upitne rečenice stavlja znak upitnik [115].

Pitanje je govorni čin kojim se nešto pita, dok je upitna rečenica sintaksna konstrukcija s upitnim obilježjem koje je jako i mora biti glasovno ostvareno pomoću posebne upitne intonacije [96].

Pitanja su u hrvatskome jeziku slabo istražena. Neke se vrste pitanja u hrvatskim gramatikama uopće ne spominju, a pojedine suvremene gramatike ne sadrže osnovnu podjelu pitanja [66].

Pitanja možemo podijeliti u nekoliko osnovnih vrsta [71]:

- jestno-niječna pitanja:
  - o kod kojih je važna cijela propozicija. Odgovor na takvo pitanje je potvrda ili nijekanje propozicije. Razlikujemo nulta i čestična pitanja. U nulta pitanja pripadaju pitanja oblika bez inverzije (npr. *Petra piše pjesme?*) te oblik s inverzijom (*Ideš ti s nama u kino?*). Nulta pitanja najčešće se koriste u razgovornom jeziku. U književnom jeziku koriste se pitanja s česticom *li* (npr. *Jesu li dobro odigrali utakmicu?*).
- dopunska pitanja:
  - o Kod njih je upitni dio na kraju strukture, a očekuje se potvrđan odgovor. Najčešće dopune takvih pitanja su čestice *jel da* (npr. *On ti je najdraži, jel da?*) i *zar ne* (npr. *To sam ti ja rekla, zar ne?*). Osim toga, moguće su i drugačije vrste dopunskih pitanja [71], [66]:
    - rečenični niz koji se sastoji od dvije nezavisne rečenice od kojih je druga – ona upitna – eliptična struktura, odnosno riječ je o dvije nezavisno složene rečenice spojene bez veznika, npr. *Natoči mi čašu soka, hoćeš li?*
    - zavisno složena rečenica u kojoj je zavisna rečenica tematizirana i pomaknuta na početak, odnosno inverzija, tj. da je prvi dio zavisna surečenica, a drugi dio glavna, npr. *Kad je sinoć došla, zar nećeš reći?*
    - jednorečenična struktura kojoj je upitni dio iza surečenice, npr. *Ti ideš u kino, je li?*
- posebna ili zamjenična pitanja:

- Kod njih se ne pita je li cijela rečenica istinita ili ne, već se samo pita za jedan njezin dio. Zovu se još i pronominalna pitanja, a u novije vrijeme u svjetskoj lingvistici i wh-pitanja [71]. Dva su njihova obilježja:
  - uvedene su posebnim upitnim riječima – jednom ili više njih, najčešće zamjениčkoga podrijetla,
  - surečenica koja slijedi iza te riječi nije potpuna, već sadrži jedno ili više praznih mjesta.
- Odgovor je skup izraza koji ubačeni na to prazno mjesto čine rečenicu istinitom.
- alternativna pitanja:
  - nabrajaju moguće odgovore, a odgovor je jedan od ponuđenih odgovora, npr. *Jesi li ti onaj koji ima doći ili da drugoga čekamo?, Jeste li u Rijeku došli vlakom ili autobusom?* i slično. Na njih nije moguće odgovoriti potvrdno ili niječno, niti su to otvorena pitanja koja treba nadopuniti.
- ječna pitanja:
  - traži se obavijest o nejasnim dijelovima sugovornikova razgovora. Upitna zamjenica nalazi se na mjestu tipičnom za sintaktičku ulogu koju predstavlja i takva pitanja imaju osobitu jezičnu intonaciju.

U engleskom jeziku razlikujemo tri osnovne vrste pitanja [10], [100], [49]:

- da – ne pitanja (engl. *polar questions*) – odgovor na takvo pitanje je *da* ili *ne* (*yes* ili *no*),
- wh – pitanja (engl. *wh-questions*) – pitanja koja koriste wh-riječi (na primjer, *who*, *whom*, *whose*, *what*, *which*, *why*, *when*, *where*, *how* i tako dalje) te
- alternativna pitanja (engl. *alternative questions*) – pitanje se sastoji od dva pitanja povezana s *ili* (engl. *or*).

U engleskom jeziku najčešće se koriste da-ne pitanja (70%), zatim wh – pitanja (27%) te alternativna pitanja (3%) [14], [47].

U ovom istraživanju, tekstualno znanje pohranjeno u bazi podataka sastoji se od jednostavnih rečenica hrvatskog jezika i jednostavnih rečenica engleskog jezika.

Pod **jednostavnim rečenicama hrvatskog jezika** podrazumijevamo izjavne rečenice koje imaju samo jedan predikat u sadašnjem glagolskom vremenu u potvrdnom obliku



te izrečen subjekt u nominativu. Takve rečenice nemaju veznika, usklika, čestica, negacije, upitnih i odnosnih zamjenica.

Pod **jednostavnim rečenicama engleskog jezika** podrazumijevamo izjavne rečenice s jednim glagolom u Simple present tense ili Present continuous tense, subjektom, objektom i/ili priložnim oznakama.

Znanje u bazi podataka sustava temeljenog na NOK metodi propituje se upitnim rečenicama, odnosno pitanjima na odgovarajućem hrvatskom ili engleskom jeziku. Za oba jezika, postavljena su pitanja tipa wh-pitanje (*wh-questions*, zamjenična pitanja) te da-ne pitanja (jestno-niječna pitanja, polarna pitanja).

### 3 Metoda "Node of Knowledge"

Čvorovi znanja (engl. *Node of Knowledge*, skraćeno NOK) nova je metoda za prikaz znanja. Grupa autora razvila je NOK metodu sa svrhom istraživanja i prikazivanja znanja sadržanog u prirodnom jeziku [92]. Cilj metode je omogućiti strukturiranje mreže znanja sadržane u tekstualno zapisanom obliku u bilo kojem ljudskom prirodnom jeziku [93]. Metoda se koristi za modeliranje znanja iskazanog pisanim ljudskim jezikom, odnosno znanja koje se sastoji iz skupa rečenica koje mogu, ali i ne moraju biti međusobno povezane. Ova općenitost omogućava modeliranje različitih vrsta znanja: znanja iz rječnika i enciklopedija, znanja iz postojećih baza podataka, znanja iz poslovnih procesa, znanja iz poslovnih dokumenata, znanja iz pisanih tekstova i slično [92]. NOK metoda može modelirati znanje na različitim razinama apstrakcije, od činjeničnoga znanja do znanja koja predstavljaju pravila. NOK omogućuje projektiranje meta baza podataka za alternativno čuvanje znanja na način različit od jezika, govora, pisma, odnosno misli ljudskoga uma [88].

NOK metoda grafički prikazuje znanje u obliku Dijagrama "Node of Knowledge" (DNOK) [6], [104]. Dijagram "Node of Knowledge", skraćeno DNOK, kao i ostale grafičke metode za prikaz znanja, prikazuje znanje u obliku grafa koji se sastoji od čvorova i veza. Čvorovima se prikazuju koncepti, a veze među čvorovima koriste se za predstavljanje odnosa među konceptima. Uspoređujući NOK metodu s drugim grafičkim metodama (Basic Conceptual Graphs (BCG) [18], Multi-layered extended semantic networks, (MULTINET) [40], Hierarchical Semantic Form (HSF) [113], Resource Description Framework (RDF) [45], može se zaključiti da je NOK metoda jednostavnija od ostalih metoda (ima manji broj elemenata), ekspresivnija (omogućuje prikaz znanja na različitim razinama apstrakcije) te jednostavnija za čitanje (u NOK metodi moguće je znanja čitati počevši od bilo kojega čvora, ali uz primjenu uloga veze) [48]. Navedene karakteristike NOK metode omogućuju njezinu jednostavnu implementaciju te time i njezinu široku primjenjivost, a to je jedan od osnovnih razloga zašto je odabrana NOK metoda.

Glavne odlike NOK metode [48]:

- jednostavnost,
- bogatstvo prikaza različitih vrsta ljudskoga znanja,

- mogućnost automatskog otkrivanja novih znanja iz postojećih,
- mogućnost vođenoga unosa novih znanja te
- mogućnost jednostavne izrade korisničkih upita nad bazom znanja.

Metoda se zasniva na istraživanju i analizi prirodnog jezika. Analiziraju se rečenice, riječi i njihovo značenje te redoslijed povezivanja riječi u složenije oblike. NOK polazi od rečenice i u njoj pronalazi eksplicitan i implicitan sadržaj te oba unosi u mrežu znanja. U mrežu znanja se znanje zapisuje tako da svaki čitatelj, bez obzira ima li pozadinsko predznanje ili ne, može tumačiti značenje mreže znanja te da iz istog dijagrama mreže znanja (DNOK) može izvući isto znanje.

Osnovno pravilo postavljeno u metodi NOK glasi: "Svako verbalizirano znanje je sastavljeno od dijelova, ima strukturu i sadržaj te se kao takvo može predstaviti u obliku mreže znanja". Dijelovi te mreže znanja su čvorovi međusobno povezani vezama. Svaki čvor je jedinica znanja, koja se dalje može dijeliti u dijelove na nižoj razini apstrakcije ili imati atribute koji tom čvoru dodjeljuju podatke [88].

Prema NOK metodi za predstavljanje znanja sve složeno ljudsko znanje predstavlja mrežu povezanih pojmova (DNOK). Pojmove zovemo *čvorovima*. Značenje pojedinog pojma (čvora) daje ljudska inteligencija. Značenje odnosa među čvorovima drži se u strukturi DNOK-a konceptima *veze* među čvorovima. Složeno ljudsko znanje može se prikazati dijagramom različitim od ljudskog jezika na način da se isti pojam pojavljuje samo jednom na DNOK-u, a da je povezan vezama neograničen broj puta s različitim drugim pojmovima (čvorovima).

NOK metoda se sastoji od:

- DNOK (Diagram Node of Knowledge) – formalizam za grafičku reprezentaciju (grafički prikaz u obliku dijagrama NOK-a),
- FNOK (Formalized Node of Knowledge) – formalizam za prikaz znanja u tekstualnom obliku (formalizirani tekstualni zapis) te
- QFNOK (Question Formalized Node of Knowledge) – formalizam za prikaz pitanja u tekstualnom obliku (formalizirani tekstualni zapis pitanja)

Postoje dvije varijante NOK metode: jednosmjerna i dvosmjerna NOK metoda. Osnovna razlika između tih dviju metoda je u tome, ima li veza koja povezuje dva čvora jednu ili dvije uloge. O tome će više riječi biti u poglavlju 3.2.

### 3.1 Koncepti NOK metode

Prema [92], NOK metoda se sastoji od koncepata: čvor, veza, procesni čvor, kontekstni čvor, kontekstna veza i uloga veze. Osim tih osnovnih koncepata za prikaz verbaliziranog znanja postoje i podatkovni čvor te atributski procesni čvor koji služe za prikaz baze podataka. Koncepti NOK metode prikazani su u tablici 3.1.

Tablica 3.1. Osnovni koncepti NOK metode

Koncept	Simbol
Čvor	
Procesni čvor	
Veza	
Kontekstna veza	
Atributski procesni čvor	
Podatkovni čvor	

**Čvor (node)** je kapljica znanja (pojam, entitet, riječ) različita od bilo kojeg drugog znanja u modelu. To je najmanja jedinica znanja koja se ne dijeli u dijelove. Svi pojmovi koji imaju vlastito značenje su u čvorovima. Čvorovi mogu predstavljati konkretne imenovane ljude, stvari, događaje, radnje, ideje, ali i koncepte na višoj razini apstrakcije (osoba, stol, sportski događaj, učenje, emocije, aktivnosti i slično). Veće jedinice znanja predstavljene su skupom povezanih čvorova. Čvorovi nisu skupovi (relacije, tablice, klase) sličnih entiteta. Čvor nije klasifikacija entiteta već jedan pojedinačni entitet. Svaki novi pojam je novi čvor. Čvor sadrži jedan i samo jedan pojam [92].

Kako bi se prikazale različite vrste koncepata iz nekog znanja koriste se različite vrste čvorova:

- obični čvor (statički čvor, entitet, koncept, pojam),
- kontekstni čvor (apstraktni, grupa, vrsta, klasa, okvir, tip),
- podatkovni čvor (mjesto gdje se čuvaju podaci) i
- procesni čvor (vezni, odnosni, dinamički, funkcijski, akcijski, uvjetovani).

U literaturi koja opisuje NOK metodu, se za obični čvor koriste još i nazivi *specijalizirani čvor*, *opisni čvor*, *opisani čvor* ili *čvor*. U nastavku ćemo obični čvor označavati samo s *čvor*. Čvor se predstavlja pravokutnikom unutar kojeg je naveden naziv čvora.

**Kontekstni čvor** je čvor na višoj razini znanja u odnosu na specijalizirani čvor s niže razine (npr. Osoba – Iva).

**Procesni čvor** je čvor koji svojim vezama povezuje čvorove i zajedno s njima predstavlja složenije prikaze znanja u agregirana znanja. Namijenjen je za prikazivanje onih znanja koja ne možemo prikazati običnim čvorovima jer predstavljaju: odnose među čvorovima, aktivnosti, veze više čvorova, radnje, zbivanja i slično. Razlikujemo: Atributski procesni čvor, Specijalizirani procesni čvor te Kontekstni procesni čvor. Kontekstni procesni čvor je čvor na višoj razini u odnosu na specijalizirani procesni čvor. U literaturi koja opisuje NOK metodu, za procesni čvor koristi se još i naziv *specijalizirani procesni čvor*, ali u nastavku ćemo specijalizirani procesni čvor označavati samo s *procesni čvor*. Procesni čvor se označava ovalom, elipsom ili krugom te se unutar njega upisuje naziv procesnog čvora. Posebni procesni čvorovi uvedeni su kako bi se modelirale točke agregacije unutar neke rečenice, dok veza među procesnim čvorovima omogućava povezivanje više rečenica [88].

**Veza** povezuje najviše dva čvora u mreži. Predstavlja se ravnom ili zakrivljenom crtom. Veza nema naziv veze, ali može imati naziv uloga. Naziv **uloge** je podatak koji pripada čvoru i propituje ulogu povezivanja tog čvora s drugim čvorom [92]. **Uloga** je točka dodira veze i čvora. U svakoj vezi postoje dvije uloge. Najčešće, uloga je wh-pitanje. Trostruke ili višestruke veze, odnosno veze koje povezuju tri ili više čvora su zabranjene. Različite veze među čvorovima se koriste kako bi se omogućilo grupiranje koncepata u složenije semantičke konstrukcije [88].

Razlikujemo obične, kontekstne, procesne i podatkovne veze, ovisno o tome koje čvorove povezuju.

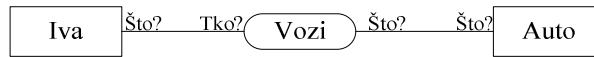
**Kontekstna veza** je specijalna veza između kontekstnog čvora (čvora na višoj razini apstrakcije, opći, nadređeni, klasa, generički, nadpojam) i specijaliziranog čvora (opisanog, konkretnog, čvora na nižoj razini apstrakcije, pojavnog, pripadnog). Kontekstna veza za pojedini specijalizirani čvor odgovara na pitanja: što je čvor, koje je vrste čvor, kojeg je tipa, koje je klase, u koju grupu pripada i slično [92]. Jedan čvor može

biti kontekstni čvor za neograničen broj opisnih čvorova. Jedan čvor može imati neograničen broj nadređenih kontekstnih čvorova [88]. Kontekstna veza prikazuje se linijom sa strelicom koja pokazuje na čvor na nižoj razini.

Koncepti atribut i podatak uvode se kako bi baza znanja mogla, kao i baza podataka, čuvati podatke, a ne samo njihove strukture u NOK metodi [88]. Za podatak se uvodi čvor najniže razine hijerarhije, odnosno **podatkovni čvor**, a za atribut se uvodi **atributski procesni čvor** koji povezuje neki čvor i njemu pripadni podatak.

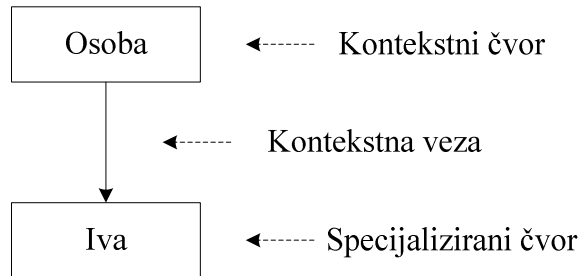
Svaki čvor ima bar jedan atribut (naziv čvora, jedinstveni identifikator čvora). Podatak se s čvorom povezuje linijom na kojoj je upisano ime atributa. Linija veze predstavlja atribut. Atribut na jednom kraju povezuje čvor, a na drugom podatak. Time je podatak protumačen i postaje informacija u mreži znanja [88]. Podatkovni čvor je običan čvor koji ima oznaku D (skraćeno od Data - podatak). Atributski čvor je procesni čvor koji ima ime sastavljeno iz dva dijela odvojena znakom ";". Prvi dio je naziv atributskog procesnog čvora i povezan je s nazivom uloga veze (pitanjima). Drugi dio je naziv atributa i ima oznaku "A:", nakon koje slijedi ime atributa. Na ovo mjesto (naziv atributa) će se prenositi imena iz relacijske sheme baze podataka i na taj način povezivati podaci u bazama podataka i dati njihovo tumačenje kroz DNOK. Metoda NOK će omogućiti konverziju baze podataka u bazu znanja i višestruku razinu opisa značenja podataka, što sada baze podataka nisu u stanju. Opis značenja podataka izvodi se u zasebnom programskom dijelu. Radi pojednostavljivanja DNOK-a dozvoljeno je kraće zapisivanje atributa i podataka, odnosno atribut je linija koja spaja podatak s čvorom. Piše se samo ime atributa. Model je malo manje čitljiv jer nema pitanja i imena atributskog procesnog čvora.

Prikažimo koncepte NOK metode na jednostavnom primjeru. Slika 3.1 prikazuje rečenicu *Iva vozi auto.* u DNOK-u. DNOK se sastoji od dva obična čvora *Iva* i *Auto* te procesnog čvora *Vozi*. Oba čvora povezana su vezama s procesnim čvorom. Svaka od veza ima dva pitanja koja opisuju uloge veze i ima važnu ulogu prilikom propitivanja znanja. Primjerice, čvor *Iva* i procesni čvor *Vozi* povezuje veza s dvije uloge *Što?* i *Tko?*. Ako pitamo *Što Iva?* odgovor je *Vozi*. Ako pak pitamo *Tko vozi?* dobivamo odgovor *Iva*. Slično je i s vezom procesnog čvora *Vozi* i čvora *Auto*.



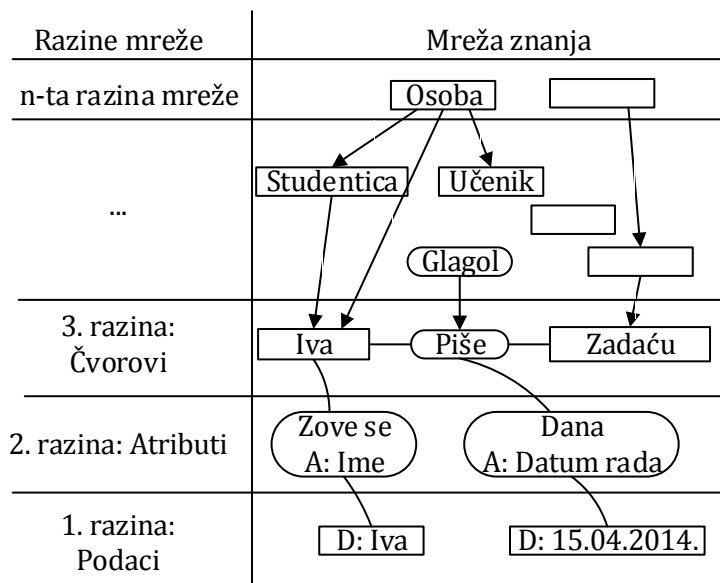
Slika 3.1. DNOK za rečenicu *Iva vozi auto*.

Slika 3.2 prikazuje primjer kontekstnog čvora i kontekstne veze. Kontekstni čvor *Osoba* je kontekstnom vezom povezana sa specijaliziranim čvorom *Iva*.



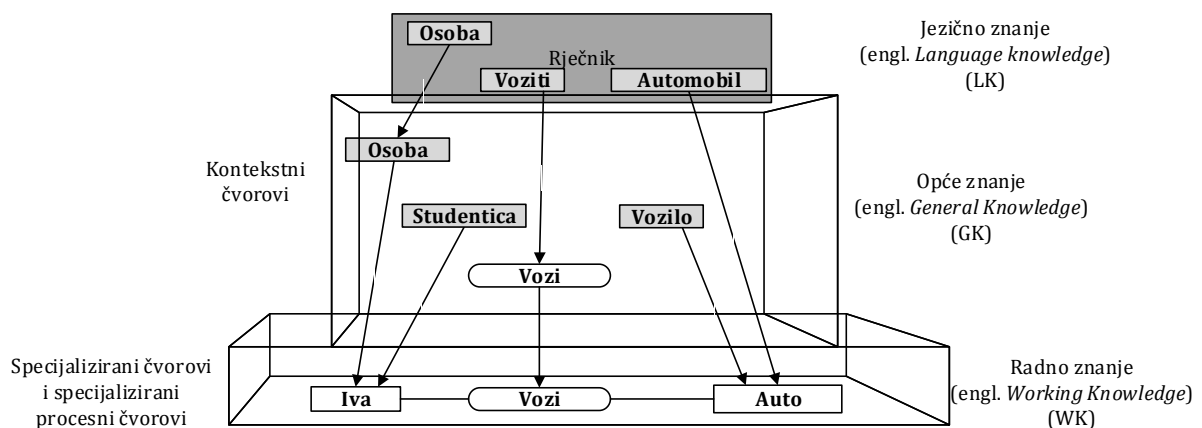
Slika 3.2. Kontekstna veza između kontekstnog i specijaliziranog čvora

Ako se povezuju dva čvora različitih razina onda se kontekstna veza prikazuje linijom sa strelicom. Strelica pokazuje na čvor na nižoj razini. Na slici 3.3 prikazana je hijerarhija između čvorova s isticanjem čvorova koji pripadaju pojedinoj razini [93], [92]. Na prvoj i drugoj razini prikazani su atributi i podaci. Oni se obično ne crtaju na DNOK-u. Modeliranje znanja, odnosno prikaz znanja DNOK-om obično kreće od 3. razine na kojoj se nalaze specijalizirani čvorovi i specijalizirani procesni čvorovi. Na višim razinama hijerarhije su kontekstni čvorovi. Primjerice, na trećoj razini je čvor *Iva* koji je povezan s čvorovima *Studentica* i *Osoba* na višim razinama hijerarhije te su čvoru *Iva* dodijeljena dodatna znanja.



Slika 3.3. Različite razine mreže znanja

Tekstualno znanje iskazano prirodnim jezikom se NOK metodom organizira u tzv. kocku znanja ili mrežu znanja (engl. *knowledge network*, KN) koja se sastoji od tri različite razine znanja [103] prikazane na slici 3.4. Na najvišoj razini je *Jezično znanje* (engl. *language knowledge*, LK), zatim *Opće znanje* (engl. *general knowledge*, GK) te *Radno znanje* (engl. *working knowledge*) na najnižoj razini. Radno znanje je povezano s gornjim razinama i predstavlja konkretno znanje. U radno znanje ubrajaju se rečenice prirodnog jezika. Na razini višoj od radnog znanja je opće znanje gdje se nalaze znanja kao dani u tjednu, obiteljski odnosi (muž – žena – dijete – otac – sin ...) i mnogi drugi apstraktni pojmovi i njihovi odnosi. Na najvišoj razini, jezično znanje, nalaze se riječi iz rječnika i pravila jezika. Rječnici su važni za razvoj sustava za razumijevanje značenja rečenica. Na osnovu semantike riječi sadržane u rječnicima započinje proces tumačenja značenja rečenica [103]. Potrebno je za svaku riječ (koja ima: kontekstno značenje, složenice, izvedenice, sinonime, homonime i dr.) napraviti poseban model znanja samo te riječi. Skup svih ovih modela predstavlja model rječnika prirodnog jezika, odnosno jezično znanje [89]. Moguće je međusobno povezivanje čvorova s bilo koje razine, a tada čvor s više razine nazivamo kontekstni čvor u odnosu na povezani čvor s niže razine. U okviru iste razine, moguće su kontekstne veze. Riječi u rečenici (konkretnoj, radnoj, pojavnoj rečenici) postaju čvorovi u DNOK-u koji se povezuju s čvorovima (riječima) iz razine općeg znanja (nijednim, jednim ili više njih). Na taj način se tumači značenje riječi kroz kontekst riječi. Svaka riječ u konkretnoj rečenici ima vezu prema jezičnom znanju. Na slici 3.4 u radnom znanju, DNOK-om je prikazana rečenica *Iva vozi auto*. Prema mreži znanja vidimo da je Iva osoba i studentica koja vozi auto, a auto se, između ostalog, ubraja pod vozila, a u rječniku je povezan s čvorom automobil.



Slika 3.4. Tri razine znanja



## 3.2 Jednosmjerna i dvosmjerna NOK metoda

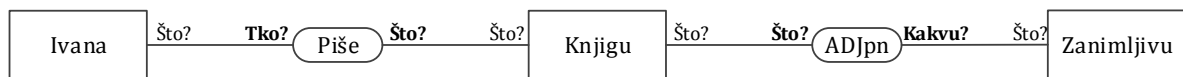
Postoje dvije varijante NOK metode – jednosmjerna i dvosmjerna NOK metoda. U nastavku će biti opisane obje varijante.

### 3.2.1 Dvosmjerna NOK metoda

Kod dvosmjerne (dvostrane) NOK metode, veza ima dvije uloge te predstavlja dva preslikavanja (dvije veze, od prvog čvora k drugom i obrnuto).

Kod dvosmjernog NOK-a vrijedi pravilo da dva obična čvora ne mogu biti povezana samo vezom s dvije uloge već između njih mora postojati procesni čvor. Ukoliko iz rečenice nije jasno koji procesni čvor povezuje dva obična čvora, uvode se procesni čvorovi koji u DNOK-u nemaju posebnu semantiku kao riječi. Ti procesni čvorovi služe kao poveznica između čvorova koji su usko povezani i omogućuju očuvanje semantike u rečenicama. Razlikujemo sljedeće procesne čvorove: *ADJpn*, *PREpn*, *NUMBpn* i *ARTpn*, odnosno pridjevski procesni čvor (engl. *adjectival process node*), prijedložni procesni čvor (engl. *prepositional process node*), brojčani procesni čvor (engl. *numeric process node*) te članski procesni čvor (engl. *article process node*). *ARTpn* koristi se u DNOK-u samo za rečenice na prirodnom engleskom jeziku (povezuje čvor koji sadrži jedan od članova *a*, *an* ili *the* s riječi (odnosno čvorom) kojoj pripadaju, primjerice imenici), dok se ostali čvorovi mogu koristiti i kod hrvatskog i kod engleskog jezika. *ADJpn* koristi su kod povezivanja pridjeva s njemu hijerarhijski nadređenim čvorom, *PREpn* povezuje prijedlog s njemu hijerarhijski nadređenim čvorom, *NUMBpn* koristi se za povezivanje broja s njemu nadređenim čvorom. Hijerarhijska ovisnost kod pojedinih vrsta riječi opisana je u poglavlju 4.2. Primjeri upotrebe tih procesnih čvorova pokazani u poglavlju 5.2.3. Osnovni koncepti koji se koriste u ovoj verziji NOK metode su: čvor (oznaka: pravokutnik), procesni čvor (oznaka: elipsa) te veza (oznaka: ravna crta) s dvjema ulogama koje se navode na vezi između dva čvora, svaka uloga bliže onom čvoru koji postavlja pitanje.

Dvosmjernu NOK metodu pokazat ćemo na primjeru. Slika 3.5 prikazuje dvosmjernan DNOK za rečenicu *Ivana piše zanimljivu knjigu*.



Slika 3.5. Dvosmjernan DNOK za rečenicu *Ivana piše zanimljivu knjigu*.

DNOK za tu rečenicu sastoji se od tri obična čvora: *Ivana*, *Zanimljivu* i *Knjigu* te procesnog čvora *Piše*. Procesni čvor *ADJpn* na DNOK-u koristimo kako bismo povezali dva obična čvora čija se veza u rečenici podrazumijeva, ali nije eksplicitno navedena. Čvor *Ivana* i čvor *Knjigu* međusobno su povezani procesnim čvorom *Piše*. Osim toga, čvor *Knjigu* povezan je s čvorom *Zanimljivu* s pridjevskim procesnim čvorom *ADJpn*. Čvor *Zanimljivu* detaljnije opisuje čvor *Knjigu*, što nam sugerira procesni čvor *ADJpn*. Na svakoj vezi među čvorovima navedena su dva pitanja koja definiraju ulogu i ona su vrlo važna kod čuvanja i propitivanja znanja. Uloge su najčešće wh-pitanja.

Na primjer, čvor *Ivana* i procesni čvor *Piše* povezani su vezom sa dvije uloge *Što?* i *Tko?*. Ako pitamo *Što Ivana?* odgovor je *Piše*. Ako pitamo *Tko piše?* odgovor je *Ivana*. Sljedeća dva čvora koja su međusobno povezana su *Piše* i *Knjigu* s ulogama *Što?* i *Što?*. Ako pitamo *Što piše?* odgovor je *Knjigu*. Ako pitamo *Što knjigu?* odgovor je *Piše*.

Čvor *Knjigu* je također povezan pridjevskim procesnim čvorom *ADJpn* vezom s dvjema ulogama *Što?* i *Što?*. Ako pitamo *Što knjigu?*, odgovor je *ADJpn*. Ako pitamo *Što ADJpn?* odgovor je *Knjigu*. Isti taj procesni čvor *ADJpn* je s druge strane povezan vezom s dvije uloge *Kakvu?* i *Što?* s čvorom *Zanimljivu*. Ako pitamo *Što zanimljivu?*, odgovor je *ADJpn*. Ako pitamo *Kakvu ADJpn?* odgovor je *Zanimljivu*. U ovom slučaju pridjevski procesni čvor, kao što je ranije rečeno, služi za povezivanje dvaju običnih čvorova pa iz te veze možemo zaključiti da čvor *ADJpn* pridjevski procesni čvor služi za detaljnije objašnjenje jednog običnog čvora, točnije čvora *Knjigu* detaljnije određujemo čvorom *Zanimljivu*.

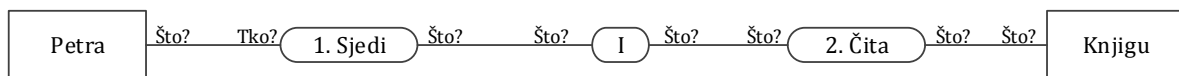
DNOK se čita tako da odaberemo procesni čvor npr. *Piše* i pitamo se pitanja iz uloga i to: *Piše Tko?* Odgovor je čvor na drugom kraju veze, dakle odgovor je: *Ivana*.

Isto vrijedi i suprotno. Pitamo se *Ivana Što?* Odgovor je *Piše*. Međutim, ovdje se može javiti *Problem jednosmjernih veza*. Primjerice, neki procesni čvorovi povezuju dva čvora tako da je jedan čvor koji vrši radnju (subjekt), a drugi trpi radnju (objekt) te veza među njima ima smisla samo u jednom smjeru. U prethodnom primjeru, *Ivana piše knjigu*. na pitanje *Što?* kod trpnog čvora *Knjigu* odgovor kao ime procesnog čvora *piše* nema smisla, jer to možemo tumačiti i kao da *Knjiga piše Ivanu*. što nije ispravno. Pravo značenje te

mreže je *Knjiga je pisana od Ivane*. A onda se ovdje radi o novoj mreži te je moguće da neka uloga nema definirano pitanje te je potrebno proširivanje mreže.

U većini slučajeva uloga sa strane čvora prema procesnom čvoru je *Što?* (ako ona postoji).

Slika 3.6 prikazuje primjer procesne veza koja se pojavljuje kada su dva procesna čvora međusobno povezana. Primjerice, u rečenici: *Petra sjedi i čita knjigu*. *Sjedi* i *čita* su procesni čvorovi, s ulogama *tko?* i *što?* te *što?* i *što?*. Između dva procesna čvora je veznik *I*. Opis je analogan prethodnom opisu.



Slika 3.6. Dvosmjernan DNOK za rečenicu *Petra sjedi i čita knjigu*

### 3.2.2 Jednosmjerna NOK metoda

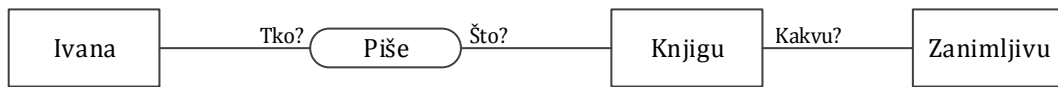
Kod dvosmjernog NOK-a vidjeli smo da je pitanje sa strane čvora prema procesnom čvoru uvijek *Što?* (ili *What?* u engleskom jeziku), a ponekad ta uloga čak ni ne mora imati smisla i može biti neizrečena, stoga tu ulogu zanemarujemo u ovoj varijanti NOK-a.

Jednosmjerna (jednostrana) NOK metoda je ona varijanta NOK metode u kojoj veza među čvorovima ima samo jednu ulogu. Ona povezuje dva čvora tako da kreće od jednog čvora k drugome. Veza između dva čvora ostvaruje se postavljanjem pitanja vezanog za prvi čvor, a na postavljeno pitanje kao odgovor dobivamo drugi čvor (koji je na drugoj strani veze). Obrnuto, pitanje koje postavljamo drugom čvoru da bismo kao odgovor dobili prvi čvor zanemarujemo. Za ovu varijantu NOK metode razvijen je FNOK zapis (formalizirani zapis rečenica).

Osnovni koncepti koji se koriste u ovoj verziji NOK metode su: čvor (oznaka: pravokutnik), procesni čvor (oznaka: elipsa) te veza (oznaka: ravna crta) s jednom ulogom koja se navodi na vezi između dva čvora, bliže onom čvoru koji postavlja pitanje. U ovoj verziji dozvoljeno je povezati dva obična čvora vezom te između njih ne mora biti procesni čvor.

Korištenjem koncepata NOK metode nastaje DNOK. Glagol u rečenici prirodnog jezika odgovara procesnom čvoru, a ostale riječi odgovaraju čvorovima. Jednosmjernu NOK

metodu pokazat ćemo na primjeru. Slika 3.7 prikazuje jednosmjernan DNOK za rečenicu *Ivana piše zanimljivu knjigu*.



Slika 3.7. Jednosmjernan DNOK za rečenicu *Ivana piše zanimljivu knjigu*.

DNOK za tu rečenicu sastoji se od 3 obična čvora: *Ivana*, *Zanimljivu* i *Knjigu* te procesnog čvora *Piše*. Čvor *Ivana* i čvor *Knjigu* međusobno su povezani procesnim čvorom *Piše*. Osim toga, čvor *Knjigu* povezan je s čvorom *Zanimljivu* vezom s ulogom *Kakvu?*. Čvor *Zanimljivu* detaljnije opisuje čvor *Knjigu*, što nam sugerira veza između njih. Na svakoj vezi među čvorovima navedeno je jedno pitanje koja definira ulogu i ona su vrlo važna kod čuvanja i propitivanja znanja. Uloge su najčešće wh-pitanja.

Na primjer, čvor *Ivana* i procesni čvor *Piše* povezani su vezom s ulogom *Tko?*. Ako pitamo *Tko piše?* odgovor je *Ivana*. Obrnutu ulogu nemamo istaknutu. Sljedeća dva čvora koja su međusobno povezana su *Piše* i *Knjigu*. Povezani su vezom s ulogom *Što?* na strani procesnog čvora. Ako pitamo *Što piše?* odgovor je *Knjigu*. Dodatno, čvor *Knjigu* je također povezan vezom s ulogom *Kakvu?* s čvorom *Zanimljivu*. Ako pitamo *Kakvu knjigu?*, odgovor je *Zanimljivu*.

DNOK se čita tako da odaberemo procesni čvor npr. *Piše* i pitamo se pitanja iz uloga i to : *Piše Tko?* Odgovor je čvor na drugom kraju veze, dakle odgovor je: *Ivana*.

Usporedbom jednosmjerne i dvosmjerne NOK metode zaključujemo da je dvosmjerna NOK metoda semantički bogatija od jednosmjerne (jednostrane) NOK metode. Ujedno, ona je i složenija te ima mnogo više čvorova i uloga. Samim time je teža za implementaciju.

### 3.2.2.1 FNOK formalizam

Osim grafičkog prikaza tekstualnog znanja (DNOK-a), razvijen je i FNOK (Formalized Node of Knowledge) formalizam za prikaz znanja u tekstualnom obliku. FNOK-om se rečenice sa semantičkom vezom među čvorovima, koristeći wh-pitanja, mogu prikazati tekstualno.

FNOK je detaljno opisan u poglavlju 4.2.

FNOK zapis za rečenicu *Ivana piše zanimljivu knjigu*. prikazanu DNOK-om na slici 3.7 glasi: *piše ("tko?" Ivana, "što?" knjigu ("kakvu?" zanimljivu))*.

### 3.3 Konceptualni okvir "Node of Knowledge"

Od samog početka razvoja u 2011. godini, u sklopu NOK metode razvijeno je nekoliko formalizama, a sama metoda je u [90] prikazana kao dio Konceptualnog okvira za razvoj sustava temeljenog na znanju "*Node of Knowledge*" (engl. *Conceptual framework for knowledge-based system development "Node of Knowledge"*), odnosno skraćeno Konceptualni okvir NOK metode.

Konceptualni okvir "Node of Knowledge (NOK)" je skup metoda i pravila te odgovarajućih alata za analizu i prikaz semantike sadržane u rečenicama prirodnog jezika. Ovaj konceptualni okvir NOK obuhvaća:

- NOK metodu,
- DNOK formalizam,
- FNOK formalizam,
- QFNOK formalizam,
- Pravila prevođenja rečenice u DNOK,
- Pravila prevođenja rečenice u FNOK zapise,
- Pravila prevođenja pitanja u QFNOK zapise,
- Softver za konverziju rečenica u FNOK zapise,
- Softver za konverziju upitnih rečenica (pitanja) u QFNOK zapise,
- Softver za punjenje FNOK zapisa u relacijsku bazu podataka i
- Softver za odgovaranje na pitanja.

DNOK formalizam opisan je u [93], [92], [104]. Softver za konverziju rečenica u FNOK zapise i Softver za konverziju upitnih rečenica (pitanja) u QFNOK zapise, odnosno detaljan opis algoritma transformacije i odgovarajućeg programskog proizvoda u Pythonu za odabrane izjavne i upitne rečenice dan je u radovima o NOK metodi [47], [90]. U tim radovima autori nisu riješili transformaciju teksta u baze podataka u kojima bi se čuvale rečenice, što je znanstveni doprinos ovoga rada.

Ostali dijelovi konceptualnog okvira "Node of Knowledge" (NOK) bit će opisani u sljedećim poglavljima.

### 3.3.1 Formalizacija teksta

Formalizacija teksta, koja se odvija Softverom za konverziju rečenica u FNOK zapise te Softverom za konverziju upitnih rečenica (pitanja) u QFNOK zapise, sastoji se od nekoliko dijelova. To su: sintaksna analiza rečenica/pitanja, formalizacija rečenica/pitanja, phrase structure grammar (PSG) i leksikon. PSG preko koje se generira jezik formaliziranog zapisa rečenica, izvedena je indukcijom. Izvođenje gramatike se temelji na pretpostavljenoj jednostavnoj PSG koja se koristila u sintaksoj analizi neformalnog zapisa rečenice. U osnovni, predložena metoda vrši prevođenje rečenica neformalnog jezika u rečenice formalnog jezika kojeg generira izvedena PSG [47].

Postoje četiri klase gramatika, tipa 0, 1, 2 i 3 koje generiraju jezike istoga tipa. Ovdje se koristi beskontekstna (engl. *context free*) gramatika koju čine [26]:

$N$  konačan skup neterminalnih simbola ili varijabli,

$V$  konačan skup terminalnih simbola (rječnik) uz uvjet da je  $V \cap N = \emptyset$

$P$  konačan skup parova nizova:  $\{(\alpha, \beta): \alpha = \alpha_1 \gamma \alpha_2; \alpha \in N, \alpha_1, \alpha_2, \beta \in (N \cup V)^*, \gamma \in N\}$

$S$  poseban znak iz  $N$ ,  $S \in N$ , nazvan početni simbol.

Beskontekstne gramatike primijenjene na prirodne jezike mogu biti korisne samo u prikazu osnovne strukture rečenica, bez značenja (semantike). U prirodnom jeziku rečenica predstavlja grupu sintaktički povezanih riječi. Sintaksna povezanost riječi u rečenici govori o načinu na koji su riječi organizirane kako bi pokazale povezanost unutar rečenice [20]. Način slaganja riječi se može izraziti gramatikom, nizom pravila kojima se riječi slažu u fraze, koje se dalje slažu u složenije fraze, sve dok se ne dođe do rečenice [27]. Jedna od najpoznatijih generativnih gramatika (engl. *generative grammar*) je *phrase structure grammar* (PSG) koju je prvi iznio Noam Chomsky (1957) [25].

U svrhu razvoja metode formalizacije znanja iskazanoga tekstom koja bi izjave transformirala u njihov formalizirani oblik uspostavljene su dvije gramatike te jedan morfološko sintaktičko-semantički leksikon. Prva faza u formalizaciji tekstom iskazanoga znanja je faza parsiranja - provjera sintaksne ispravnosti rečenice i utvrđivanje njezinih sastavnih dijelova. Uspostavlja se jednostavan PSG gdje će se koristiti morfosintaktički leksikon engleskog jezika MULTEXT-East. Morfosintaktički leksikon engleskoga jezika MULTEXT-East proširuje se semantičkim dijelom kojim će se za svaki oblik riječi identificirati i wh-pitanje na koje riječ može primiti odgovor,

odnosno na koje riječ može dati odgovor. Na taj način izgrađuje se morfološko sintaktičko-semantički leksikon. Nakon uspostave PSG za engleski jezik te morfološko sintaktičko-semantičkog leksikona, istom notacijom razvija se PSG za formalizaciju rečenica što predstavlja drugu fazu u formalizaciji tekstem iskazanoga znanja - fazu semantičke interpretacije [47].

PSG promatra rečenicu kao spoj dviju fraza - *noun phrase* i *verb phrase*. Svaka od ovih fraza se dalje može sastojati iz drugih fraza (*noun phrase, adverb phrase, adjective phrase, preposition phrase*) i/ili vrsta riječi (*noun, verb, adjective, adverb, determiner, prepositions, wh-words* itd.) - općenito se svaka fraza sastoji iz drugih fraza i/ili vrsta riječi. U PSG, vrste riječi nemaju daljnje razlaganje, već se zamjenjuju odgovarajućom riječi iz leksikona. Leksikon je lista svih riječi jezika i njihovih lingvističkih svojstava. Može se reći da se PSG sastoji iz leksikona i pravila kojima se odgovarajuće riječi iz leksikona odabiru i slažu u fraze do konačne rečenice [14], [123], [20], [47].

U praksi, nakon izvođenja gramatike nekog jezika i njezine verifikacije, dalje će njezina uporaba biti u provjeri je li dani niz rečenica jezika. Tada se problem svodi na nalaženje niza izvođenja koji bi rezultirao tim nizom (rečenicom). Takav postupak naziva se sintaksna analiza ili parsiranje (engl. *parsing*). Ustrojba postupka sintaksne analize na računalu (program u izabranom jeziku za programiranje) naziva se parser [27], ako je jezik definiran gramatikom, odnosno, prepoznavać, ako je jezik definiran generatorom (automatom odgovarajućeg tipa) [27].

Općenito, transformacija znanja iskazanoga tekstem u formalizirani prikaz znanja izvodi u tri faze [47]: faza parsiranja, faza semantičke interpretacije i faza kontekstualne interpretacije. Fazom parsiranja se vrši sintaksna analiza rečenica iz kojih se sastoji tekst. Analiza se temelji na PSG gramatici, a njezin zadatak je utvrditi sintaksnu ispravnost rečenice te njezine osnovne dijelove (kod PSG dijelovi rečenice su fraze, vrste riječi i leksemi). Nakon faze parsiranja (utvrđivanja sastavnih dijelova rečenice), slijedi faza semantičke interpretacije rečenice. Semantička interpretacija rečenice se temelji na sastavnim dijelovima rečenice utvrđenim u fazi parsiranja i ova faza treba rezultirati formalnom reprezentacijom rečenice. Nakon formalne reprezentacije rečenice, ona se nastoji uklopiti u od prije formalizirano znanje (usvojeno znanje). To znači da se utvrđeni sastavni dijelovi rečenice koji su sada formalno zapisani, nastoje povezati s dijelovima drugih formaliziranih rečenica čime se širi ukupno formalizirano znanje te se

nova formalizirana rečenica stavlja u neki kontekst. Ovo se sve događa u fazi kontekstualne interpretacije [61].

### **3.4 Preslikavanje između relacijske baze podataka i NOK metode**

Zbog izgradnje mreže znanja te proširivanja baze podataka novim znanjem, važno je NOK metodu preslikati u relacijsku bazu podataka, odnosno u logičku shemu dobivenu metodom entiteta i veza te na kraju u relacijsku bazu podataka. U nastavku su opisana preslikavanja metode entiteta i veza u NOK metodu te obrnuto, preslikavanje NOK metode u metodu entiteta i veza. Metoda entiteta i veza, odnosno njen dijagram entiteta i veza (DEV) prevodi se u relacijski model po definiranim pravilima, odnosno stvara se shema baze podataka [87].

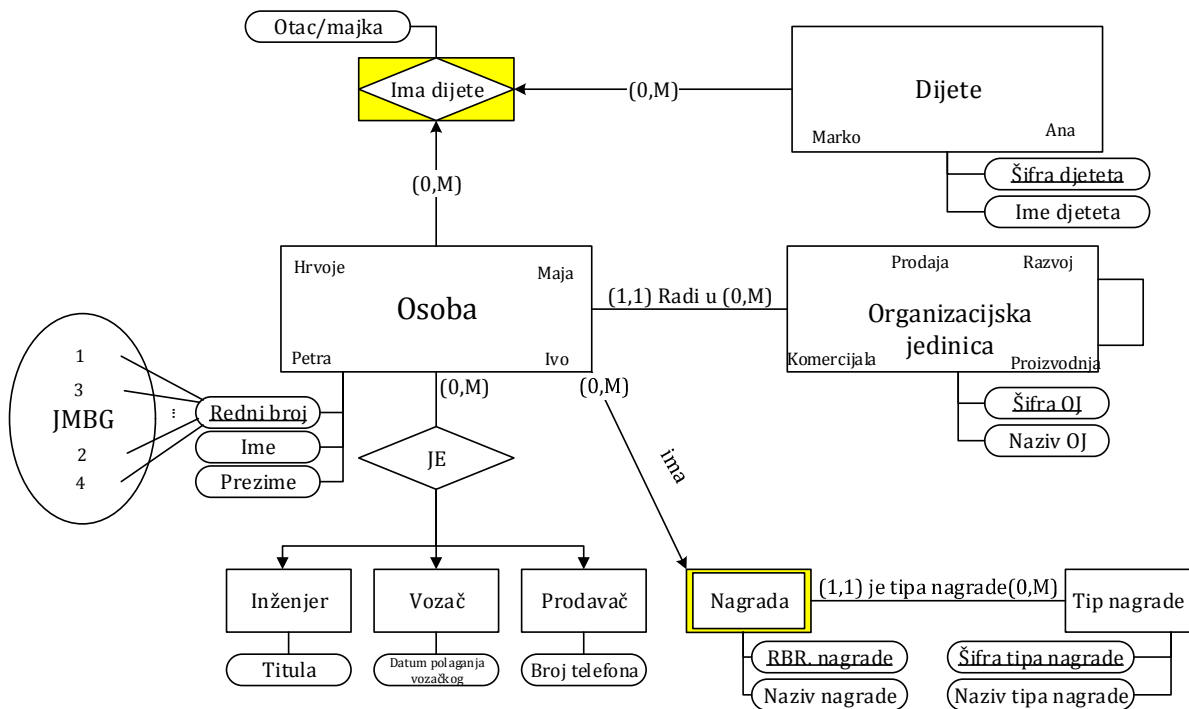
#### **3.4.1 Preslikavanje metode entiteta i veza u NOK metodu**

Postavlja se pitanje je li moguće prikazati strukturu i sadržaj baze podataka nekog informacijskog sustava koristeći NOK metodu. Zadatak je ekvivalentan pronalaženju pravila preslikavanja (transformacija), odnosno definiranju tablice preslikavanja metode entiteta i veza u NOK metodu. To uključuje proučavanje metode entiteta i veza s odgovarajućom bazom podataka i NOK metode općenito, ali i koristeći konkretne primjere. U svrhu provjere navedenog zadatka, postavlja se hipoteza: Moguće je koncepte metode entiteta i veza preslikati u koncepte NOK metode. Cilj je razvoj sustava temeljenog na NOK metodi, odnosno razvoj softvera za upis znanja sadržanog u DNOK-u. Računalni sustavi uglavnom imaju bazu podataka. Baza podataka sadrži mrežu znanja u nekom relacijskom sustavu za upravljanje bazom podataka. Zbog toga je važno koncepte metode entiteta i veza preslikati u odgovarajuće koncepte NOK metode.

Postavlja se pitanje: kako povezati postojeći informacijski sustav s njegovom mrežom znanja. Pokazat ćemo kako projekt informacijskog sustava (posebice njegov dio koji se odnosi na bazu podataka) povezati s mrežom znanja koristeći NOK metodu. Svaki projekt informacijskog sustava ima model procesa i model podataka. Model podataka se obično prikazuje nekom verzijom metode entiteta i veze, a sama baza relacijskom metodom. S ciljem provjere postavljene hipoteze potrebno je za svaki koncept metode entiteta i veza pronaći odgovarajući koncept NOK metode. U procesu zaključivanja korišteni su različiti primjeri, počevši od jednostavnijih prema sve složenijima. U nastavku je prikazan jedan od primjera.

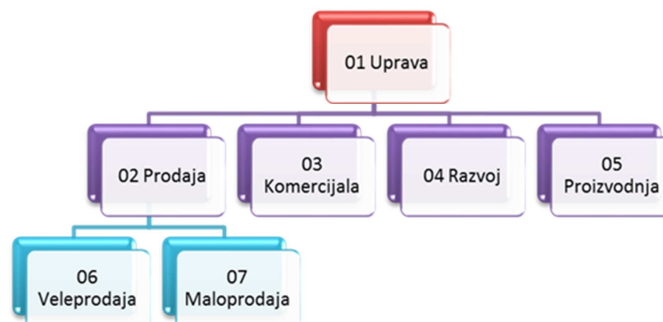


Slika 3.8 prikazuje dijagram entiteta i veza (DEV) koji daje pregled osoba zaposlenih u organizacijskoj jedinici (OJ). Model obuhvaća i podatke o djeci zaposlenika, kao i podatke o primljenim nagradama.



Slika 3.8. DEV Osoba u OJ

Pod tip entiteta *Osoba* pripadaju sve pojedinačne osobe, na primjer, Hrvoje, Petra, Maja, Ivo i tako dalje. Osoba radi u Organizacijskoj jedinici. Pod tip entiteta *Organizacijska jedinica (OJ)* pripadaju sve pojedinačne organizacijske jedinice (na primjer, Prodaja, Razvoj, Komercijala, Proizvodnja...). Tip entiteta *Organizacijska jedinica (OJ)* organiziran je hijerarhijski. Slika 3.9 prikazuje primjer hijerarhijske organizacije Organizacijske jedinice (OJ). Jedna osoba iz tipa entiteta *Osoba* radi u jednoj i samo jednoj OJ. U jednoj OJ iz tipa entiteta *OJ* radi niti jedna (na primjer, jer je OJ tek osnovana) ili mnogo entiteta iz tipa entiteta *Osoba*.



Slika 3.9. Hijerarhijska organizacija Organizacijske jedinice (OJ)

Tip entiteta *Dijete* sadrži podatke o svakom pojedinačnom djetetu. Jedna osoba iz tipa entiteta *Osoba* može imati nijedno ili mnogo djece. Jedno dijete iz tipa entiteta *Dijete* može imati nijednog ili više roditelja (najčešće je to 2, ali u nekim slučajevima može biti i više, primjerice kod posvojene djece imamo biološke roditelje i roditelje koji su to dijete posvojili). Zbog veze M : M po gornjoj granici, između tipa entiteta *Osoba* i tipa entiteta *Dijete* javlja se agregirani tip entiteta (agregacija) *Ima dijete*.

Slabi tip entiteta *Nagrada*, ovisan je o tipu entiteta *Osoba* i sastoji se od svih nagrada koje je dobila neka osoba. Jedna osoba iz tipa entiteta *Osoba* dobila je nijednu ili mnogo nagrada, dok jednu nagradu iz tipa entiteta *Nagrada* mogla je dobiti jedna i samo jedna osoba. Svaka nagrada je nekog tipa. Jedna nagrada iz slabog tipa entiteta *Nagrada* je jednog i samo jednog tipa iz tipa entiteta *Tip nagrade*, dok jednog tipa nagrade iz tipa entiteta *Tip nagrade* može biti nijedna ili mnogo nagrada iz tipa entiteta *Nagrada*.

Tipovi entiteta *Inženjer*, *Vozač* i *Prodavač* podtipovi su tipa entiteta *Osoba*.

DEV sa slike 3.8 preveden u relacijski model daje sljedeće relacije:

**Osoba** (Redni broj, Ime, Prezime, Šifra OJ)  
**Inženjer** (Redni broj, titula)  
**Vozač** (Redni broj, datum polaganja vozačkog)  
**Prodavač** (Redni broj, broj telefona)  
**Organizacijska jedinica**(Šifra OJ, Naziv OJ, *Šifra\_OJ\_nad*)  
**Nagrada** (Redni broj, Rbr. nagrade, Naziv nagrade, *Šifra tipa nagrade*)  
**Tip nagrade** (Šifra tipa nagrade, Naziv tipa nagrade)  
**Dijete** (*Šifra djeteta*, Ime djeteta)  
**Ima dijete** (Redni broj, *Šifra djeteta*, otac/majka)

Popunjene tablice nastale na temelju relacijske sheme prikazane su u tablicama 3.2 do 3.10.

Tablica 3.2. Osoba

Šifra osobe	Ime	Prezime	Šifra OJ
1	Maja	Marić	05
2	Petra	Perić	01
3	Ivo	Ivić	04
4	Danijel	Horvat	03

Tablica 3.3. Inženjer

Redni broj	Titula
1	dipl. ing
2	mag .oecc.

Tablica 3.4. Vozač

Redni broj	Datum polaganja vozačkog
3	17.04.1980.

Tablica 3.5. Prodavač

Redni broj	Broj telefona
4	01 5774124

Tablica 3.6. Dijete

Šifra djeteta	Ime
D001	Marko
D002	Ana

Tablica 3.7. Ima dijete

Redni broj	Šifra djeteta	Otac/majka
3	D001	Otac
1	D002	Majka
2	D001	Majka

Tablica 3.8. Tip nagrade

Šifra tipa nagrade	Naziv tipa nagrade
T1	radnik godine
T2	radnik mjeseca

Tablica 3.9. Nagrada

Redni broj	RBR. nagrade	Naziv nagrade	Šifra tipa nagrade
1	1	Radnik godine 2011	T1
2	1	Radnik mjeseca lipnja	T2
2	2	Radnik godine 2012	T1

Tablica 3.10. Organizacijska jedinica

Šifra OJ	Naziv OJ	Šifra_OJ_nad
01	Uprava	
02	Prodaja	01
03	Komercijala	01
04	Proizvodnja	01
05	Razvoj	
06	Veleprodaja	02
07	Maloprodaja	02

Čovjek svojim znanjem i poznavanjem metode entiteta i veza te svojim zaključivanjem, može iz DEV-a prikazanog na slici 3.8 te korištenjem podataka iz baze podataka iščitati različite rečenice. Korištenjem samo DEV-a sa slike 3.8, možemo iščitati neke opće rečenice koje povezujemo s General knowledge (GK) u NOK-u. Ako uz DEV koristimo i podatke iz baze podataka dolazimo do rečenica koje možemo povezati s Working knowledge (WK) u NOK metodi.

Neke od rečenica koje se mogu iščitati iz DEV-a su:

- Osoba ima dijete.
- Osoba radi u OJ.
- U OJ radi osoba.
- Inženjer, vozač i prodavač su osobe.

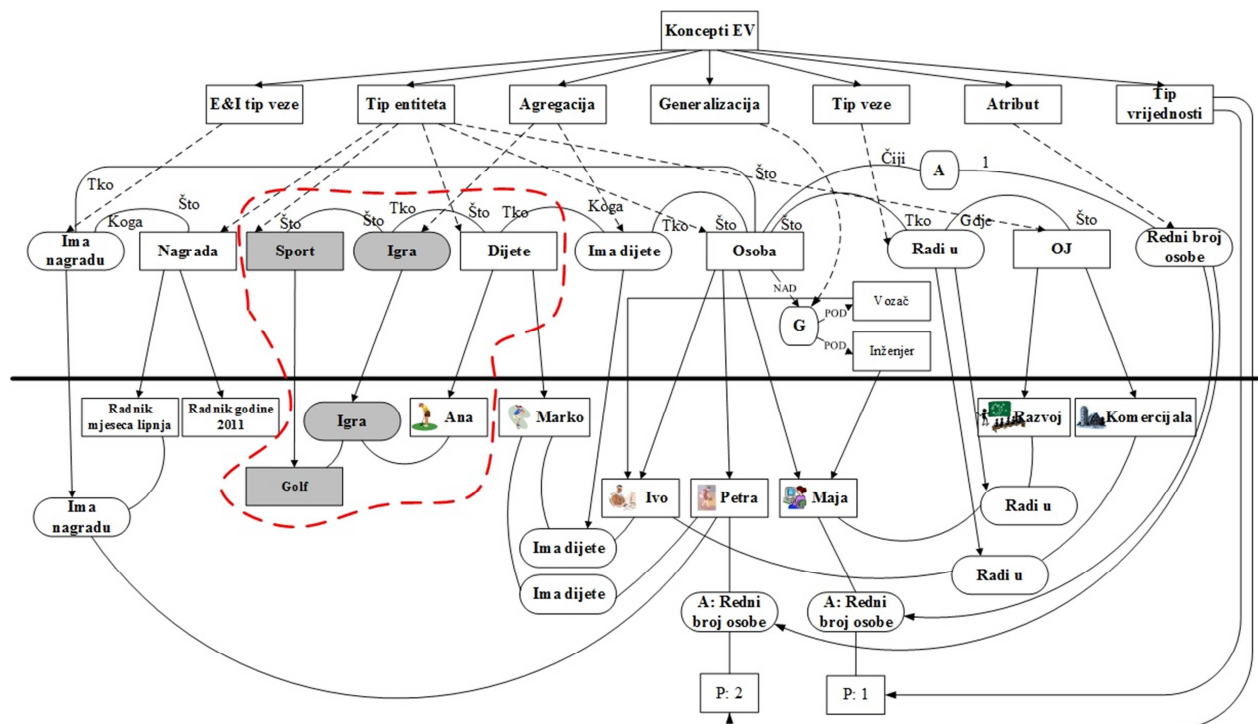
- Osoba ima nagradu.
- Nagrada je tipa nagrade.

Neke od rečenica koje se mogu iščitati iz strukture i podataka u bazi podataka:

- Maja radi u razvoju.
- Ivo radi u komercijali.
- Ivo i Petra imaju dijete Marka.
- Petra ima nagradu "radnik mjeseca lipnja".
- Maja je inženjer.

Slika 3.10 prikazuje podatke iz baze podataka Osoba u formi DNOK-a. Na slici su prikazane dvije razine znanja General knowledge (GK) (iznad crte) te Working knowledge (WK) (ispod crte). Iznad crte prikazana je shema u DNOK-u, odnosno prikazano je znanje koje se može pročitati iz dijagrama entiteta i veza. Ispod crte je mreža znanja, odnosno znanje dobiveno na temelju podataka u bazi. Dakle, DNOK sadrži shemu baze podataka i samu bazu podataka, odnosno prikazano je koji koncepti na DNOK-u odgovaraju kojim konceptima na DEV-u.

Isprekidanom linijom prikazana je veza između NOK-a i DEV-a.



Slika 3.10. Rečenice prikazane DNOK-om

Objasnit ćemo samo dio slike. Iz DEV-a je moguće "izvući" rečenicu: *Osoba radi u OJ*. Kada nju prikazujemo u NOK-u, na DNOK-u uočavamo procesni čvor *Radi u* te čvorove

*Osoba* i *Organizacijska jedinica*. Procesni čvor *Radi u* je s ulogama *Što* i *Tko* povezan s čvorom *Osoba* te s ulogama *Gdje* i *Što* povezan s čvorom *OJ*. Konkretni primjer na WK razini koristeći podatke u bazi bio bi *Ivo radi u komercijali*. *Ivo* i *Komercijala* su čvorovi, *Radi u* je procesni čvor. Uloge u vezi iste su kao i na GK razini.

Osim toga, na WK razini koristeći podatke imamo rečenice:

- Maja radi u razvoju.
- Ivo radi u komercijali.
- Ivo i Petra imaju dijete Marka.
- Petra ima nagradu "radnik mjeseca lipnja".
- Maja je inženjer.

Čvorovi *Osoba* i *OJ* pripadaju pod koncept *Tip entiteta* na DEV-u, a *Radi u* pripada pod *Tip veze*.

NOK ima jednostavnu mogućnost proširenja svoje mreže novim znanjem. Ako smo došli do novih znanja, NOK ćemo jednostavno proširiti novom rečenicom. Na primjer, novo znanje daje rečenica *Ana igra golf*. pa jednostavno možemo proširiti mrežu znanja (kao što je prikazano na slici 3.10) novim čvorovima. Izmjena modela entiteta i veza ili sheme baze podataka nije jednostavna.

Na temelju raznih primjera i primjenom grafičkih metoda, metoda analize i sinteze te apstrakcije, postaje jasno kako preslikati koncepte dijagrama entiteta i veza u koncepte NOK metode. Korištenjem apstrakcije na više razina, preslikavanje je poopćeno, a koncepti su grupirani u određene skupine te se mogu zaključiti pravila preslikavanja metode entiteta i veza u NOK metodu.

Tablica 3.11 prikazuje preslikavanje metode entiteta i veza u NOK metodu. Svaki koncept metode entiteta i veza može se preslikati u neki koncept NOK metode.

**Tablica 3.11. Preslikavanje metode entiteta i veza u NOK metodu**

<b>Metoda entiteta i veza</b>	<b>NOK metoda</b>
Tip entiteta	Kontekstni čvor
Entitet	Čvor
Podatak	Podatkovni čvor
Atribut	Atributski procesni čvor
Veza	Specijalizirani procesni čvor
Tip veze	Specijalizirani procesni čvor; Kontekstni procesni čvor
Agregacija	Specijalizirani procesni čvor, Agregacijski procesni čvor

Vrijednost	Podatkovni čvor
Egzistencijalni i identifikacijski (E&I) tip veze	Egzistencijalni i identifikacijski (E&I) procesni čvor (E& IPČ)
Generalizacija <S> / Specijalizacija	Generalizacijski procesni čvor
Prethođenje <P>	Prethođenje procesni čvor

Rezultati koji su prikazani u tablici 3.11 te slici 3.10 koja prikazuje rečenice dobivene iz dijagrama entiteta i veza te pripadne baze podataka potvrđuju hipotezu. Zaključujemo da je moguće preslikati koncepte metode entiteta i veza u koncepte NOK metode.

Iz ovog primjera možemo zaključiti koji je potencijal NOK metode. Sadržaji baze podataka mogu se pretvoriti u tekstualni oblik. Moguće je i obrnuto. Pored toga moguće je mrežu znanja obogatiti novim rečenicama kojih u postojećim bazama podataka nema.

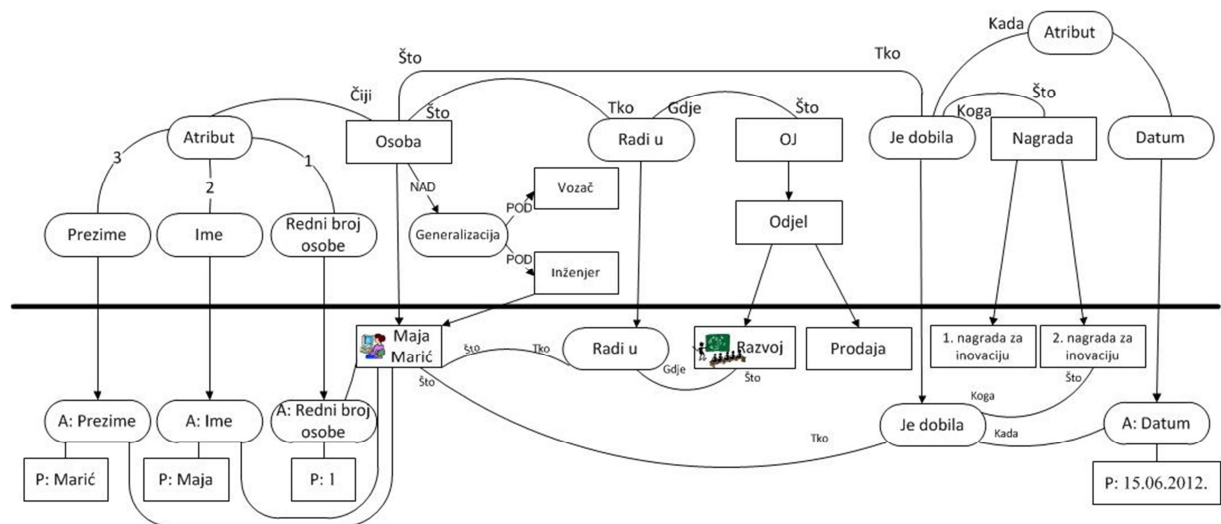
### 3.4.2 Preslikavanje NOK metode u metodu entiteta i veza

Preslikavanje metode NOK u metodu entiteta i veza prema metodologiji MIRIS prikazat ćemo kroz primjer. Za zadanu rečenicu napraviti ćemo DNOK koji ćemo zatim prevesti u DEV i relacijski model.

Rečenica glasi:

*Inženjerka Maja Marić iz odjela Razvoj dobila je 15.06.2012. "2. nagradu za inovaciju".*

DNOK je prikazan na slici 3.11.



Slika 3.11. Rečenica u DNOK-u

Prikazane su dvije razine znanja Opće znanje ili General knowledge (GK) (iznad crte) te Radno znanje ili Working knowledge (WK) (ispod crte). Iznad crte prikazana je shema u DNOK-u, a ispod baza podataka u DNOK-u.

Na slici 3.11 iznad crte se nalaze procesni čvorovi (*Radi u, Je dobila*), kontekstni i obični čvorovi (*Osoba, Vozač, Inženjer, OJ, Odjel i Nagrada*) te podatkovni čvorovi (*Redni broj osobe, Prezime, Ime, Datum*).

Čvor *Osoba* nadređen je čvorovima *Vozač* i *Inženjer*. Čvor *Osoba* ima podatkovne čvorove *Redni broj osobe, Prezime, Ime*. Čvor *Osoba* povezan je s procesnim čvorovima *Radi u* i *Je dobila*. Pitanja koja možemo postaviti uz čvor *Osoba* i čvor *Radi u* su *Što* uz *Osoba* (na primjer, *Što Osoba?* - odgovor je *Radi u*) te *Tko* uz procesni čvor *Radi u* (na primjer, *Tko Radi u?* - odgovor je *Osoba*).

Procesni čvor *Radi u* povezan je s čvorom *OJ*. Pitanja koja postavljamo su *Gdje* i *Što*. Konkretno, *Gdje radi u?*, odgovor je *OJ* te *Što OJ?*, odgovor *Radi u*. Time smo prikazali u DNOK-u dio znanja "Osoba radi u OJ". Kontekstni čvor *OJ* nadređen je čvoru *Odjel*.

Druga veza čvora *Osoba* je veza s procesnim čvorom *Je dobila*. Uloge na tu vezu su: *Što* i *Tko*. *Što Osoba?* – odgovor: *Je dobila* te *Tko je dobila?* – odgovor *Osoba*.

Procesni čvor *Je dobila* povezan je s čvorom *Nagrada* s ulogama veze *Koga* i *Što* te s podatkovnim čvorom *Datum* koji je odgovor na pitanje *Kada*.

Sada imamo općenito znanje da *Osoba* radi u nekoj *OJ* (nekom *Odjelu*) te da je nekog datuma dobila neku nagradu. Ako želimo znati o kojoj osobi se ovdje radi te kakvu nagradu je osvojila nekog datuma, moramo preći u radno znanje gdje ćemo saznati detalje.

Kontekstni čvor *Osoba* iz GK kontekstnom vezom povezan je s čvorom *Maja Marić* (s atributima: Redni broj osobe: *1*, Ime: *Maja*, Prezime: *Marić*) u WK pa možemo zaključiti da je *Maja Marić* konkretna osoba sa svojim atributima. Također, *Maja Marić* je kontekstnom vezom povezana s čvorom *Inženjer* koji je podređen kontekstnom čvoru *Osoba*. Nadalje, čvorovi *Razvoj* i *Prodaja* kontekstnom vezom povezani su s čvorom *Odjel*. Čvor *Maja Marić* povezan je s procesnim čvorom *Radi u* koji ima svoj nadređeni *Radi u* u GK. Pitanja koja ovdje postavljamo su: *Što* i *Tko* (*Što Maja Marić?* – odgovor *Radi u* te *Tko radi u?* – odgovor: *Maja Marić*). Drugi procesni čvor s kojim je čvor *Maja Marić* povezan je *Je dobila*. I taj procesni čvor u WK ima svog nadređenog u GK. Čvorovi *1. nagrada za inovaciju* i *2. nagrada za inovaciju* u WK povezani su kontekstnim vezama s kontekstnim čvorom *Nagrada* u WK. Čvor *Maja Marić* povezan je s procesnim čvorom *Je dobila* s ulogama *Što* i *Tko* analogno pitanjima u GK. Nadalje, procesni čvor *Je dobila*

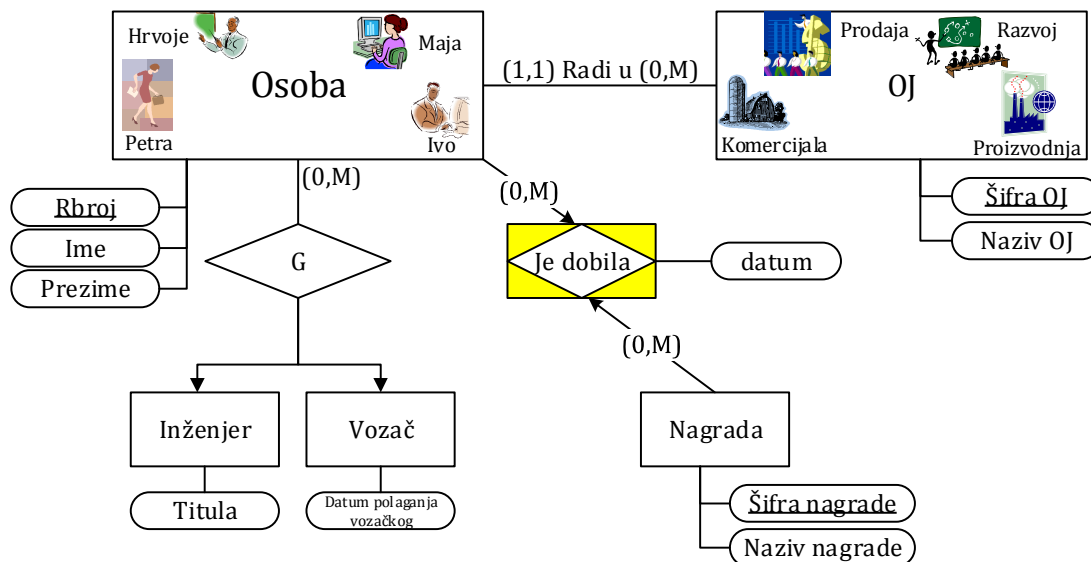
povezan je s konkretnom nagradom, odnosno čvorom 2. *nagrada za inovaciju s ulogama Koga i Što*. Kako u GK procesni čvor *Je dobila* ima podatkovni čvor *Datum*, tako je i u WK procesni čvor *Je dobila* povezan s konkretnim podatkom, odnosno podatkovnim čvorom 15.06.2012.

DNOK sadrži shemu baze podataka i same podatke. Kod metode entiteta i veza imamo samo shemu baze podataka, bez pojavljivanja. Podaci se nalaze u bazi podataka.

Slika 3.12 prikazuje rečenicu modeliranu metodom entiteta i veza. Ovakav prikaz ne prati jezik. Nedostaju imenice, glagoli, pridjevi, prilozi i slično.

Pod tip entiteta *Osoba* ubrajaju se sve pojedinačne osobe, npr. Hrvoje, Petra, Maja, Ivo itd. *Osoba* radi u Organizacijskoj jedinici. Pod tip entiteta *Organizacijska jedinica (OJ)* pripadaju sve pojedinačne OJ (npr. Prodaja, Razvoj, Komercijala, Proizvodnja...). Jedna osoba iz tipa entiteta *Osoba* radi u jednoj i samo jednoj *OJ*. U jednoj *OJ* iz tipa entiteta *OJ* radi niti jedna (npr. jer je *OJ* tek osnovana) ili mnogo entiteta iz tipa entiteta *Osoba*.

Tipovi entiteta *Inženjer* i *Vozač* podtipovi su tipa entiteta *Osoba* (koji je nadtip). Podtipovi i nadtipovi razlikuju se u atributima. U nadtip se dodaju atributi zajednički svim entitetima, a u podtip samo oni atributi koji pripadaju jednoj grupi entiteta [87].



Slika 3.12. Rečenica u DEV

Tip entiteta *Nagrada* sastoji se od svih nagrada koje se dodjeljuju u toku neke godine. Jedna osoba iz tipa entiteta *Osoba* dobila je nijednu ili mnogo nagrada, dok je jednu nagradu iz tipa entiteta *Nagrada* mogla dobiti nijedna ili mnogo osoba. Zbog M : M po gornjoj granici uvodimo agregacijski tip entiteta (agregaciju) *Je dobila*.



Prevedemo li DEV sa slike 3.12 u relacijski model dobivamo sljedeće relacije:

**OJ** (Šifra OJ, Naziv OJ)  
**Osoba** (Rbroj, Ime, Prezime, *Šifra OJ*)  
**Inženjer** (Rbroj, titula)  
**Vozač** (Rbroj, datum polaganja vozačkog)  
**Nagrada** (Šifra nagrade, Naziv nagrade)  
**Je dobila** (Rbroj, Šifra nagrade, datum)

Na temelju sheme relacijskog modela možemo napraviti relacijsku bazu podataka tj. tablice popunjene podacima (tablice 3.12 do 3.17).

U tablicama je prikazana kombinacija shema baze podataka (zaglavlje svake tablice) te konkretni podaci. Na temelju toga, možemo povući paralelu između GK i WK u NOK-u te ovih tablica. Općem znanju sa slike 3.11 (dio iznad crte) odgovara zaglavlje u tablicama a sami podaci odgovaraju WK (dio ispod crte) na istoj slici. Kada pogledamo same tablice i njihov sadržaj, vidimo da u njima nisu sadržane veze, odnosno, ne vidimo razlog zašto su dvije tablice povezane (nedostaju nam glagoli). Na primjer, pogledamo li tablice *OJ* i *Osoba*. Teško nam je zaključiti zašto su one povezane, odnosno iz njih ne možemo pročitati da *Osoba radi u OJ*. (iako čovjek glagol *radi u* podrazumijeva u mislima). Relacijski model je semantički siromašniji od modela entiteta i veza.

**Tablica 3.12. OJ**

Šifra OJ	Naziv OJ
01	Uprava
02	Prodaja
03	Komercijala
04	Proizvodnja
05	Razvoj
06	Veleprodaja
07	Maloprodaja

**Tablica 3.13. Osoba**

Rbroj	Ime	Prezime	Šifra OJ
1	Maja	Marić	05
2	Petra	Perić	01

**Tablica 3.14. Inženjer**

Rbroj	Titula
1	Inženjer strojarstva

**Tablica 3.15. Vozač**

Rbroj	Datum polaganja vozačkog
2	25.03.1985.

**Tablica 3.16. Nagrada**

Šifra nagrade	Naziv nagrade
1	1. Nagrada za inovaciju
2	2. Nagrada za inovaciju
3	3. Nagrada za inovaciju

**Tablica 3.17. Je dobila**

Rbroj	Šifra nagrade	Datum
1	2	15.06.2012.
2	1	20.07.2011.

Tablica 3.18 prikazuje preslikavanje koncepata metode za modeliranje znanja (NOK) u koncepte metode entiteta i veza (EV). Svaki od koncepata metode NOK preslikan je u barem jedan koncept metode entiteta i veza. Na primjer,

- Kontekstni čvor u NOK metodi preslikava se u tip entiteta, tip entiteta nadtip ili tip jakog entiteta u metodi entiteta i veza.
- Specijalizirani čvor preslikava se u entitet, tip entiteta, tip entiteta podtip ili tip slabog entiteta, ovisno o čemu se radi.
- Specijalizirani procesni čvor može se preslikati u vezu, tip veze, agregaciju, vrijednost ili ulogu.
- i slično.

U NOK metodi postoje koncepti koji se nisu preslikali niti u jedan koncept metode entiteta i veza. Na primjer: Uloga (pitanje) iz NOK metode, nije se preslikala ni u jedan koncept metode entiteta i veza. NOK metoda je semantički bogatija od metode entiteta i veza. Metoda entiteta i veza i relacijska metoda, odnosno njihovi modeli i baza podataka zajedno sadrže znanje koje je podskup od znanja sadržanog u modelu DNOK-a.

**Tablica 3.18. Preslikavanje NOK metode u metodu entiteta i veza**

NOK metoda	Metoda entiteta i veza
Kontekstni čvor	Tip entiteta, Tip entiteta nadtip, Tip jakog entiteta
Specijalizirani čvor (obični čvor)	Entitet, Tip entiteta, Tip entiteta podtip, Tip slabog entiteta
Specijalizirani procesni čvor	Veza, Tip veze, Agregirani tip entiteta (agregacija), Uloga
Kontekstni procesni čvor	Tip veze
Podatkovni čvor	Vrijednost
Kontekstni podatkovni čvor	Tip vrijednosti
Atributski procesni čvor	Atribut tipa entiteta
Egzistencijalni procesni čvor	Egzistencijalni tip veze, Identifikacijski tip veze, Egzistencijalni & identifikacijski tip veze, Tip veze prethođenja
Kontekstna veza	Generalizacijski tip veze, Hijerarhijski tip veze
Atribut procesnog čvora	Atribut agregiranog tipa entiteta
Generalizacijski procesni čvor	Generalizacija
Uloga (pitanje)	Nema odgovarajući koncept
Veza	Nema odgovarajući koncept

Rezultati koji su prikazani u tablici 3.18 te slici 3.12 koja prikazuje rečenice prirodnog jezika u obliku dijagrama entiteta i veza te pripadne baze podataka potvrđuju da je moguće direktno preslikati koncepte NOK metode u koncepte metode entiteta i veza, uz gubljenje informacija.

Ideja koja se razrađuje u ovom radu ne predlaže direktno preslikavanje koncepata jedne metode u drugu, već oblikovanje sustava za kontroliranu transformaciju rečenica zapisanih u FNOK obliku u relacijsku bazu podataka. Drugi razlog zašto direktna transformacija NOK metode u relacijsku bazu podataka nije moguća je što se jedan koncept iz NOK-a može preslikati u više koncepata, odnosno općenito veza između metoda je tipa  $M : M$ . Zato je ideja istraživanja naći sustav transformacije FNOK zapisa u relacijsku bazu podataka bez gubitka semantike.

## 4 Pravila prevođenja rečenica prirodnog jezika u FNOK

Tekst možemo opisati kao skup znakova, simbola ili rečenica koji su međusobno organizirani u neki napisani odlomak.

Prilikom prevođenja nekog teksta, odnosno niza rečenica (koje mogu, ali i ne moraju međusobno biti sadržajno povezane) u formalizirani NOK zapis (FNOK) potrebno je primjenjivati pravila prevođenja. Uvedimo osnovne definicije.

**Riječ** je najmanja samostalna jezična jedinica koja ima samostalno značenje [115].

**Rečenica** je niz riječi ili jedna riječ kojom prenosimo najmanju cjelovitu obavijest. Riječi se u rečenicu uvrštavaju po pravilima slaganja (definira se odnos među riječima u rečenici) i po pravilima nizanja (definira se red riječi u rečenici). Predikat, kao rečenična jezgra, ima veliku ulogu u slaganju riječi u rečenicu jer otvara mjesto ostalim članovima rečeničnog ustrojstva [115]:

- riječi u nominativu kojom se kazuje na što se odnosi obavijest izrečena predikatom, tj. subjektu,
- riječi kojom se izriče predmet radnje, tj. objektu,
- riječima koje izriču različite okolnosti (mjesto, vrijeme, način, uzrok i dr.), tj. priložnim ili adverbnim oznakama.

U pravopisnom smislu rečenica započinje velikim slovom, a završava točkom, upitnikom ili uskličnikom. Kod usmenog govora bitna je i intonacija kojom se označava početak i kraj rečenice.

**Odlomak** ili **paragraf** je novi stavak teksta koji ima samostalno značenje [84], [85].

Riječi u rečenici možemo promatrati na dva načina [115], [110]:

- Prema službi riječi u rečenici (odnosno elementima rečeničnog ustrojstva) razlikujemo:
  - o predikat (temeljni rečenični dio koji otvara mjesto ostalim rečeničnim dijelovima),
  - o subjekt (vršitelj predikatne radnje),
  - o objekt (predmet glagolske radnje),
  - o priložna oznaka (okolnost glagolske radnje),
  - o apozicija (dopuna imenici) i

- atribut (dopuna imenici).
- Prema vrsti riječi u rečenici:
  - Promjenjive
    - imenice
    - zamjenice
    - brojevi
    - pridjevi
    - glagoli
  - Nepromjenjive
    - prilozi
    - prijedlozi
    - veznici
    - usklici
    - čestice

#### **4.1 Pravila za prevođenje rečenica u FNOK prema službi riječi u rečenici (odnosno elementima rečeničnog ustrojstva)**

Kao što je ranije rečeno, rečenica se u hrvatskom jeziku sastoji od niza riječi i svaka riječ ili grupa riječi ima određenu funkciju u rečenici (predikat, subjekt, objekt, priložna oznaka, apozicija i atribut).

U engleskom jeziku se rečenica također sastoji od niza riječi, a svaka riječ ili skupina tih riječi ima određenu funkciju u rečenici. Funkcije mogu biti: subjekt, predikat, objekt (direktni i indirektni), atribut (subjekta ili objekta) i priložna oznaka (za predikat, atribut ili priložnu oznaku). Da bi se rečenica mogla formirati, ona treba imati barem subjekt i predikat. Predikat realizira glagol. Kod prepoznavanja subjekta i predikata u rečenici (a time i ostalih dijelova), najjednostavnije je pronaći glagol koji tvori izjavu ili kazuje što će se dogoditi [14], [123], [10], [100], [37], [47].

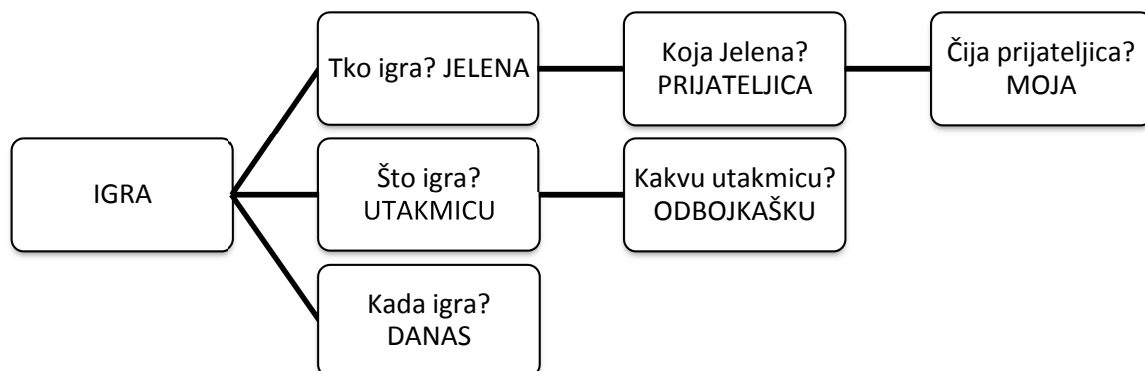
Sintaksa uključuje pravila kako strukturirati kompleksne rečenice grupirajući riječi u fraze koje se mogu nalaziti na različitim mjestima u većim sintaktičkim strukturama. Prema tome, za rečenicu možemo reći da se sastoji od fraza povezanih u strukturu stabla te da povezuje fraze jedne s drugima na različitim razinama [7]. Rečenicu možemo analizirati i na način kako je sastavljena. Najčešće je rečenica sastavljena kao

kombinacija dviju fraza (imenske fraze (engl. *noun phrase*) i glagolske fraze (engl. *verb phrase*)) koje se mogu sastojati od drugih fraza (imenske fraze (engl. *noun phrase*), priložne fraze (engl. *adverb phrase*), pridjevske fraze (engl. *adjective phrase*), prijedložne fraze (engl. *prepositional phrase*)) ili drugih vrsta riječi (imenice (engl. *noun*), glagoli (engl. *verb*), pridjevi (engl. *adjective*), prilozi (engl. *adverb*), odrednice (engl. *determiner*), prijedlozi (engl. *prepositions*), wh-riječi (engl. *wh-words*) i slično).

Na primjer, neka se rečenica može sastojati od: imenske fraze (engl. *noun phrase*), glagola (engl. *verb*) i prijedložne fraza (engl. *prepositional phrase*) koja se dalje dijeli na prijedlog (engl. *preposition*) i imensku frazu sastavljenu od člana (engl. *article*) i imenice (engl. *noun*) [118], [119].

Slika 4.1 prikazuje primjer povezivanja riječi u rečenicu tako da jedna riječ otvara mjesto drugoj te pitanja koja postavljamo da bismo otvorili novo mjesto drugoj riječi. U primjeru je korištena rečenica:

*Moja prijateljica Jelena danas igra odbojkašku utakmicu.*



**Slika 4.1. Primjer kako jedna riječ otvara mjesto drugoj**

Kao početna riječ od koje smo krenuli analizirati rečenicu uzet je glagol u prezentu, odnosno predikat *igra*. Postavljanjem dodatnih pitanja na glagol, dolazimo do subjekta, priložnih oznaka i time proširujemo rečenicu. Postavljanjem dodatnih pitanja na subjekt dolazimo do atributa i apozicija. Na predikat možemo postaviti pitanja: *Tko?*, *Što?*, *Kada?*, odnosno *Tko igra?*, *Što igra?*, *Kada igra?* na koja su odgovori redom *Jelena*, *utakmicu*, *danas*. Dodatno, mogu se postaviti pitanja koja pobliže označuju subjekt *Jelena* i objekt *utakmicu*. Na pitanje *Koja Jelena?* odgovor je apozicija *prijateljica*, a na pitanje *Kakvu utakmicu?* odgovor je *odbojkašku*. Na apoziciju *prijateljica* moguće je postaviti pitanje *Čija?*, odnosno *Čija prijateljica?* na koje je odgovor atribut *moja*.

Dijagram sa slike 4.1 možemo zapisati u tekstualnom obliku tvoreći hijerarhiju (znak "(" služi za spuštanje na nižu razinu hijerarhije, a ")" vraćanje na razinu iznad):

*igra (tko? Jelena (koja? prijateljica (čija? moja)), što? utakmicu (kakvu? odbojkašku), kada? danas)*

Pri tome pitanja *tko?*, *koja?*, *čija?*, *kakvu?*, *kada?* postavljamo da bismo otvorili novo mjesto drugoj riječi. Međutim, u bazama podataka i rječnicima nemamo podatke o službi riječi u rečenici. Problem je osmisliti algoritam koji će izvoditi računalni program s ciljem točnog određivanja službe riječi u rečenici. Primjerice kako odrediti je li neka imenica u službi subjekta, objekta, apozicije i slično. Stoga od ovog pristupa prevođenja rečenica u FNOK odustajemo.

## **4.2 Pravila prevođenja rečenice u FNOK prema vrsti riječi u rečenici**

Drugi pristup zasniva se na vrstama riječi. Uz pomoć rječnika moguće je točnije odrediti vrstu riječi svake riječi.

Pravila prevođenja se primjenjuju na sve riječi u rečenici. Najprije se odredi vrsta svake riječi u rečenici. Zatim se, u ovisnosti o vrsti riječi s kojom ju povezujemo, primjenjuje i konkretno pravilo. Cjelokupni algoritam prikazan je u 8 koraka.

Algoritam za prevođenje rečenice u FNOK je sljedeći:

1. Iz odlomka (paragrafa) uzeti rečenicu.
2. Pronaći prvu riječ u rečenici.
3. Odrediti vrstu riječi kojoj ta riječ pripada (koristeći pravila prevođenja u FNOK (PPFNOK))
4. Pronaći drugu riječ u rečenici (i kasnije sve ostale riječi do kraja rečenice) te za svaku od njih odrediti kojoj vrsti riječi pripadaju.
5. Kad imamo sve riječi iz rečenice i znamo kojoj vrsti riječi svaka od njih pripada, tražimo riječi koje imaju homonime te se iz konteksta rečenice određuje vrsta riječi te o čemu se radi.
6. Od svih riječi u rečenici biramo onu koja je po vrsti riječi glagol. Glagol stavljamo na prvo mjesto i otvaramo zagradu.

7. Nakon toga se traži veza među riječima, odnosno traži se veza između nađenog glagola i riječi koje su s njime povezane (koje pripadaju tom glagolu). To su sljedeće vrste riječi: imenice, zamjenice, priloga i drugi glagoli ili glagolski oblici.
8. Ako rečenica ima više glagola, imenica i priloga, potrebno je odrediti točnu pripadnost. Za sada se zadržavamo na jednostavnim rečenicama koje imaju jedan glagol tako da njemu pripadaju navedene vrste riječi.

Napomena 1: Iako je u [47] opisan sustav koji, koristeći *Phrase structure grammar* u sintaktičkoj analizi neformalnog zapisa rečenice, pretvara rečenice u njihov formalizirani zapis, korak 5. se za sada izvodi ručno, bez programa. Naknadno će se za taj postupak proširiti postojeći program koji će oponašati ljudski postupak.

U nastavku će biti definirana pravila prevođenja prema vrsti riječi u hrvatskom i engleskom jeziku u FNOK. Nakon kratkog opisa (definicije) pojedine vrste riječi bit će definirano pravilo te dani primjeri gdje se to pravilo primjenjuje. Primjeri će biti dani i na hrvatskom i engleskom jeziku, tamo gdje je to moguće primijeniti.

#### **4.2.1 Osnovna pravila prevođenja**

Svaka riječ u rečenici prirodnog jezika odgovara jednom čvoru (engl. *node*) u FNOK zapisu. Glagol u rečenici prirodnog jezika odgovara procesnom čvoru u FNOK-u.

Procesni čvor je na najvišoj razini hijerarhije. Svi ostali čvorovi su u hijerarhiji ispod procesnog čvora što označavamo zagradama s desne strane. Kad se procesni čvor (a to je najčešće glagol u rečenici) pronađe, stavlja se na prvo mjesto i otvara zagrada "(" . Nakon navođenja svih ostalih čvorova koje su hijerarhijski podređeni procesnom čvoru ili običnom čvoru, zagrada se zatvara ")" što označava vraćanje na višu razinu. Ispred svakog čvora u hijerarhiji dolazi upitna riječ (odnosno pitanje) kojom je čvor na nižoj razini hijerarhije povezan s čvorom na višoj razini, odnosno čvor na nižoj razini hijerarhije je odgovor na postavljeno pitanje.

Pitanje označavamo navodnicima i znakom upitnik ("?"). Neka od pitanja koja možemo postaviti su: "tko?" / "who?", "što?" / "what?", "kada?" / "when?", "kako?" / "how?", "gdje?" / "where?", "kakav?" / "what?", "koliko?" / "how much?", "how many?" i slično ovisno o vrsti riječi. Za glagole možemo postaviti različita pitanja primjerice "tko?" / "who?", "što?" / "what?", "kada?" / "when?", "kako?" / "how?", "gdje?" / "where?",



"kakav?" / "what?" i slično, koja će otvarati mjesta ostalim vrstama riječi hijerarhijski ovisnima o glagolu. Pridjevi primjerice odgovaraju na pitanje "kakav?" / "what?", "čiji?" / "whose?", "koji?" / "which?", "od čega?" / "from what?". Brojevi odgovaraju na pitanje "koliko?" / "how much?", "how many?" te "koji?" / "which?". Kod pitanja (uloga) u engleskom jeziku se najčešće radi o wh-pitanjima (engl. *wh-questions*), tako da ćemo u nastavku pod *wh-pitanje* podrazumijevati i pitanja na hrvatskom jeziku.

Iznimno, dvije (ili više riječi) koje su usko povezane i za koje je rastavljanje na više međusobno povezanih čvorova nepotrebno jer predstavljaju semantički kohezivnu cjelinu, u NOK-u tvore jedan čvor. Te riječi pišemo međusobno povezane podvlakom ("\_") ili razmakom (" "), npr. *how\_much*, *how\_many*, *how\_often*, *koliko\_često*, *Velika\_Britanija*, *Great\_Britain* ili *how much*, *how many*, *how often*, *koliko često*, *Velika Britanija*, *Great Britain* i slično.

### **Def. Odnos dviju riječi**

Neka su *X* i *Y* dvije riječi u rečenici te neka između njih postoji sintaksno-semantička veza (prema pravilima gramatike te se riječi moraju nalaziti u određenom redosljediu i taj redosljed otkriva dodatnu semantiku). Ako se veza između riječi *X* i *Y* može identificirati preko nekog wh-pitanja *Z?*, tada se formalno odnos tih dviju riječi zapisuje ovako  $X("Z?" Y)$ . Drugim riječima, moguće je za riječ (u rečenici), odnosno čvor u FNOK-u *X* postaviti takvo wh-pitanje *Z?* da kao odgovor dobijemo riječ, odnosno čvor *Y*.

### **Općeniti zapis FNOK-a:**

**process node ("role 1?" node 1, "role 2?" node 2 ("role 3?" node 3 ("role 4?" node 4, ...), ...), "role n?" node n)**

gdje su *role 1?* do *role n?* wh-pitanja, *node 1* do *node n* čvorovi, a *process node* je procesni čvor

Svaki čvor može imati svoju hijerarhiju.

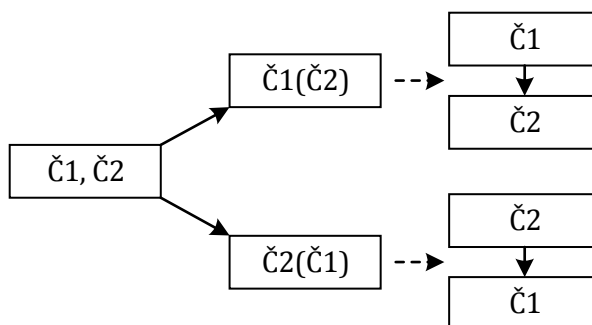
### **Def. Hijerarhijska ovisnost**

Neka su  $\check{C}1$  i  $\check{C}2$  dva čvora. Hijerarhijska ovisnost ovih dvaju čvorova može imati 2 slučaja:

1.  $\check{C}1$  je hijerarhijski nadređen  $\check{C}2$  (odnosno  $\check{C}2$  je hijerarhijski podređen  $\check{C}1$ ) što možemo kraće pisati  $\check{C}1(\check{C}2)$ , ili

2. Č2 je hijerarhijski nadređen Č1 (odnosno Č1 je hijerarhijski podređen Č2) što možemo kraće pisati Č2(Č1).

Grafički prikaz hijerarhijske ovisnosti prikazan je na slici 4.2.



Slika 4.2. Hijerarhijska ovisnost

### **Def. Redoslijed čvorova**

Na istoj razini hijerarhije zadržava se onaj redoslijed navođenja čvorova u kojem se nalaze riječi u rečenici. Čvorovi i njihova pripadajuća pitanja odvajaju se zarezom.

U hijerarhiji prednost ima procesni čvor (najčešće je to glagol, ali može biti i druga vrsta riječi, primjerice: ne (čestica), je (pomoćni glagol biti) i slično), nakon toga slijede imenice, zatim pridjevi pa brojevi te na kraju prijedlozi. Prilozi su na istoj razini hijerarhije kao i imenice. Ukoliko je više čvorova (npr. imenica i/ili priloga) na istoj razini hijerarhije redoslijed čvorova je proizvoljan. Međutim, ako je moguće, trebalo bi zadržati redoslijed čvorova kakav je i redoslijed riječi u rečenici.

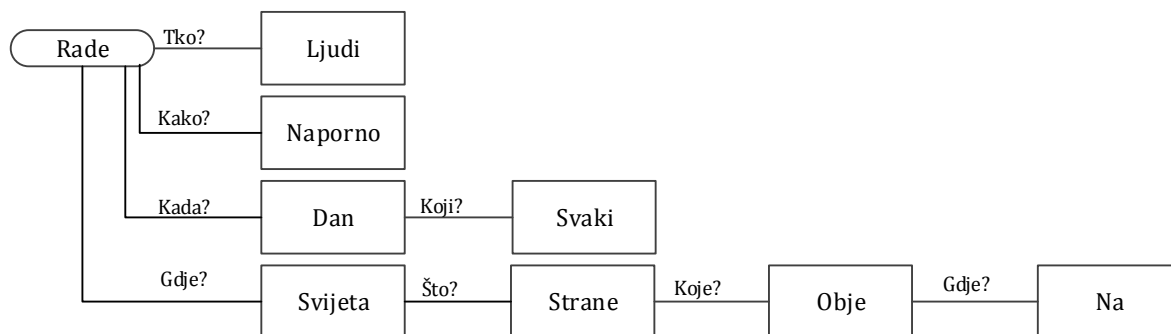
Na primjer, za rečenicu na hrvatskom jeziku *Na obje strane svijeta ljudi naporno rade svaki dan*. FNOK prikazan po razinama hijerarhije je:

- rade
  - o (
  - o "tko?" ljudi,
  - o "kako?" naporno,
  - o "kada?" dan
    - (
    - "koji?" svaki
    - ),
  - o "gdje?" svijeta
    - (
    - "što?" strane
      - (
      - "koje?" obje
        - (
        - "gdje?" na

Isti zapis kraće možemo pisati u jednom retku kao:

*rade ("tko?" ljudi, "kako?" naporno, "kada?" dan ("koji?" svaki), "gdje?" svijeta ("što?" strane ("koje?" obje ("gdje?" na )))*

Slika 4.3 prikazuje DNOK za tu rečenicu.



**Slika 4.3. DNOK za rečenicu *Na obje strane svijeta ljudi naporno rade svaki dan.***

Ako prevedemo gore navedenu rečenicu na engleski jezik, rečenicu glasi: *On either side of the world people work hard every day.*

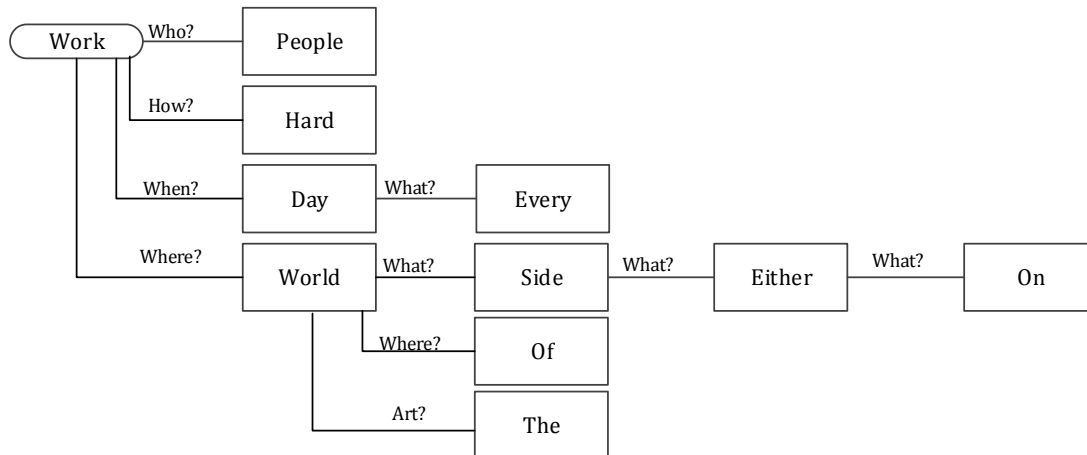
Analogno, FNOK za tu rečenicu je:

- work
  - o (
  - o "who?" people,
  - o "how?" hard,
  - o "when?" day
    - (
    - "what?" every
    - ),
  - o "where?" world
    - (
    - "what?" side
      - (
      - "what?" either
        - (
        - "what?" on
        - )
      - ),
      - "where?" of,
      - "art?" the
    - )
  - o )

Zapis iste rečenice u jednom redu je:

*work ("who?" people, "how?" hard, "when?" day ("what?" every), "where?" world ("what?" side ("what?" either ("what?" on)), "where?" of, "art?" the))*

DNOK za tu rečenicu prikazan je na slici 4.4.



Slika 4.4. DNOK za rečenicu *On either side of the world people work hard every day.*

U nastavku će biti opisana pravila za pretvaranje rečenica u FNOK u ovisnosti o vrsti riječi.

## 4.2.2 Glagoli

Glagol je promjenjiva vrsta riječi kojom se izriče radnja, stanje ili zbivanje. U hrvatskom jeziku, glagoli se mijenjaju po vremenima, licima i načinima. Razlikujemo svršeni i nesvršeni vid (na primjer, sjesti (svršeni) i sjediti (nesvršeni vid)). Pomoćni glagoli *biti* i *htjeti* služe za tvorbu složenih glagolskih oblika te tvorbu prošlog i budućeg glagolskog vremena [115], [110].

### 4.2.2.1 Glagol na početku zapisa

Glagol (koji inače označava procesni čvor) u FNOK zapisu stavljamo na najvišu razinu hijerarhije. Sve ostale vrste riječi su hijerarhijski ovisne o njemu, što možemo pisati kao:

$G, X \rightarrow G(X)$ , gdje je  $G$  oznaka za glagol,  $X$  oznaka za bilo koju vrstu riječi.

Preciznije, ako u prethodni zapis uključimo pitanje (ulogu), možemo pisati:

$G, X \rightarrow G(\text{"role?" } X)$ , gdje je  $G$  glagol,  $X$  bilo koja vrsta riječi, *role?* označava wh-pitanje.

Na primjer, u rečenici *Vedran vozi auto*. postoji jedan glagol *vozi* koji je ujedno i procesni čvor. Njega stavljamo na početak FNOK zapisa i tražimo riječi koje su povezane s tim glagolom.

- Vedran *vozi* auto. →<sup>1</sup> Vozi (...).
- Vedran *drives* a car. → Drives (...)

#### 4.2.2.2 *Glagol ispred glagola*

Glagol je uvijek ispod glagola kada se radnja navodi jedna za drugom onim redoslijedom kojim se glagoli pojavljuju u rečenici.

Na primjer:

- **Daj** mi nešto za **piti**. → **Daj** ("što?" **piti** ("što?" za), "kome?" mi, "što?" nešto)
- **Give** me something **to drink**. → **give** ("what?" **drink** ("what?" **to**), "whom?" me, "what?" something)

#### 4.2.2.3 *Hijerarhija glagolskih vremena*

Kod određenih glagolskih vremena se u rečenicama uz glavni glagol nalazi i jedan ili više pomoćnih glagola. Postavlja se pitanje kako glavni glagol i pomoćne glagole koje tvore određeno glagolsko vrijeme prevesti u FNOK.

U hrvatskom jeziku hijerarhiju uočavamo kod:

- glagolskih vremena:
  - o perfekt (prošlo vrijeme)
    - tvorba: glagolski pridjev radni + nenaglašeni prezent pomoćnoga glagola biti
    - na primjer: sam pisao, si pisao, je pisao, smo pisali, ste pisali, su pisali
  - o pluskvamperfekt (pretprošlo vrijeme)
    - tvorba: perfekt ili imperfekt glagola biti + glagolski pridjev radni
    - na primjer: bijah (bjeh) pisao, bijaše (bješe) pisao i slično ili bio sam pisao, bio si pisao i slično.
  - o futur prvi (buduće vrijeme)

---

<sup>1</sup> Simbol "→" označava preslikavanje rečenice u FNOK

- tvorba: infinitiv + nenaglašeni prezent pomoćnoga glagola htjeti
- na primjer: ću pisati, ćeš pisati, pisat ću, pisat ćeš, ću ići, ići ćeš i tako dalje.
- futur drugi (prebuduće vrijeme)
  - tvorba: svršeni prezent pomoćnoga glagola biti + glagolski pridjev radni
  - na primjer: budem pisao, budeš pisao, budete pisali i tako dalje.
- glagolskih načina:
  - kondicional prvi ili sadašnji (pogodbeni način)
    - tvorba: nenaglašeni aorist pomoćnoga glagola biti + glagolski pridjev radni
    - na primjer: bih trebao, bi trebao, bismo trebali i tako dalje.
  - kondicional drugi ili prošli (pogodbeni način)
    - tvorba: kondicional prvi pomoćnog glagola biti + glagolski pridjev radni
    - na primjer: bih bio trebao, bi bio trebao, bismo bili trebali i tako dalje.

Hijerarhija kod tvorbe glagolskih vremena u engleskom jeziku nije posebno tekstualno opisana, ali je prikazana na slici 4.5. Ovisno o pojedinom glagolskom vremenu, nemaju sva glagolska vremena sve razine hijerarhije.

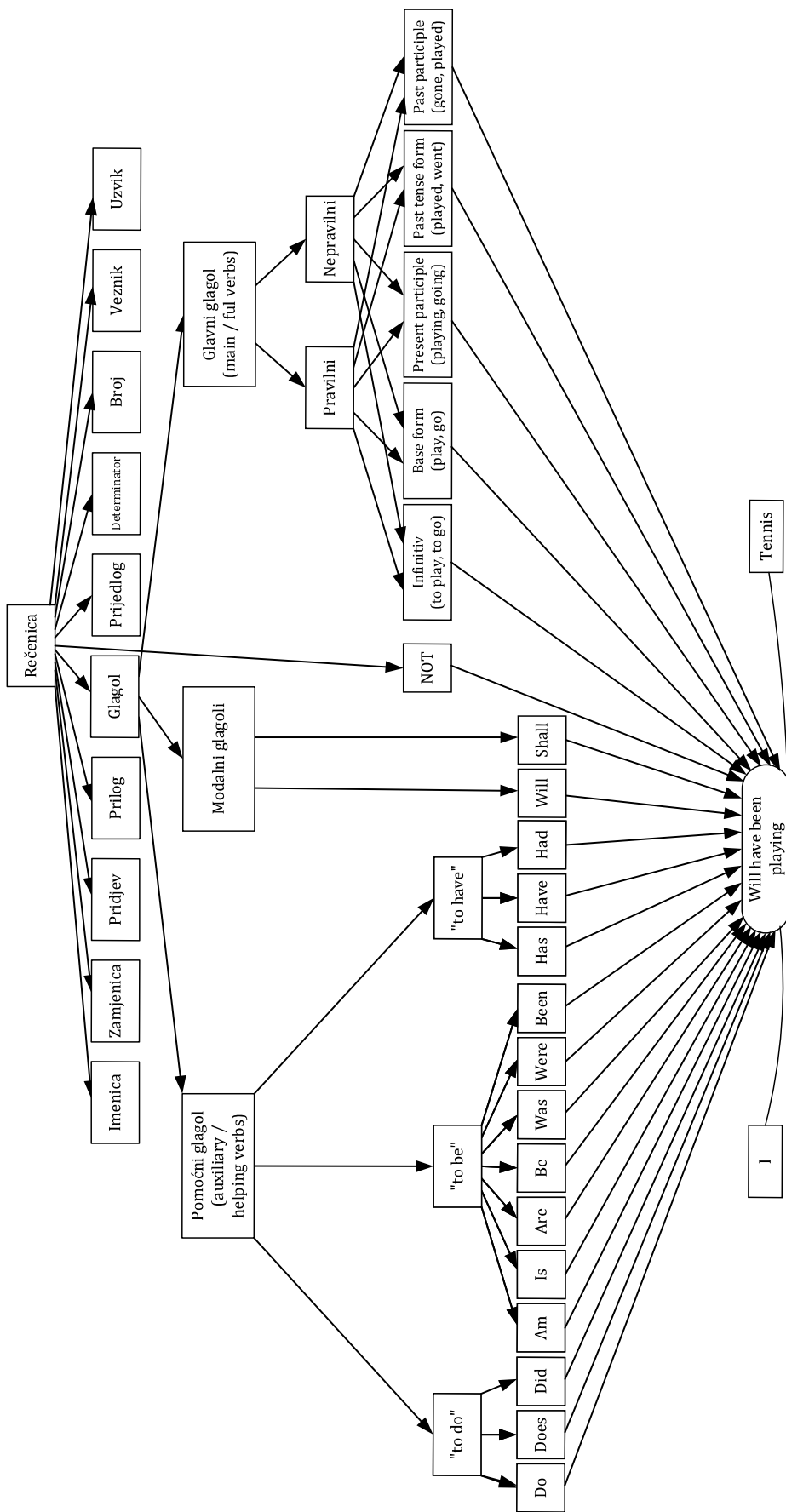
Kod tvorbe FNOK-a imamo dvije mogućnosti:

- sve riječi koje tvore jedno glagolsko vrijeme predstaviti jednim procesnim čvorom ili
- rastaviti riječi (čvorove) u nekoliko razina hijerarhije određenim redosljedom.

S obzirom da u FNOK-u želimo zadržati rečeničnu strukturu i semantički sadržaj, postavlja se sljedeće pravilo.

*Ukoliko se glagolsko vrijeme sastoji od više dijelova, oni su u hijerarhiji jedan ispod drugoga. Glavni glagol stavljamo na najnižu razinu hijerarhije, a svi pomoćni glagoli ili glagolski oblici su iznad njega (hijerarhijski ovisni tvore hijerarhiju).*

Dobro rješenje bilo bi i potpuno suprotni redosljed jer sadrži istu semantiku. Tu se može odabrati ono rješenje koje je bolje za implementaciju.



Slika 4.5. Tvorba glagolskih vremena u engleskom jeziku i njihova hijerarhija

Hijerarhija u FNOK-u za glagolska vremena i glagolske načine u hrvatskom jeziku je sljedeća:

- na najvišoj razini je bilo koji oblik pomoćnog glagola biti ili htjeti koji je nadređen glagolskom pridjevu radnom ili glagolu u infinitivu.

Prikažimo ovo pravilo na nekoliko primjera. Tablica 4.1 sastoji se od 5 stupaca: RBR (Redni broj), THJ (Rečenice na hrvatskom jeziku), FNOK HJ (FNOK zapis rečenice na hrvatskom jeziku), TENG (Rečenice na engleskom jeziku) i FNOK ENG (FNOK zapis rečenice na engleskom jeziku). Ova tablica je izvadak iz tablice 19 i tablice 20 iz Privitka 2. Redni brojevi rečenica u ovoj tablici odgovaraju rednim brojevima rečenica u cjelokupnim tablicama.

Tablica 4.1. Primjeri za modalne glagole te različita glagolska vremena u hrvatskom i engleskom jeziku

RBR	THJ	FNOK HJ	TENG	FNOK ENG
39	Ti <b>možeš kupiti</b> bilo što.	<b>možeš ("što?" kupiti</b> ("što?" bilo što <sup>2</sup> , "tko?" ti))	You <b>can buy</b> anything.	<b>can ("what?" buy</b> ("who?" you, "what?" anything))
85	Ti <b>možeš dotaknuti</b> sestrinu bilježnicu.	<b>možeš ("što?" dotaknuti</b> ("tko?" ti, "što?" bilježnicu ("čiju?" sestrinu)))	You <b>can touch</b> sister's notebook.	<b>can ("what?" touch</b> ("who?" you, "what?" notebook ("whose?" sister's)))
86	Mi <b>možemo biti</b> prijatelji.	<b>možemo ("što?" biti</b> ("tko?" mi, "što?" prijatelji))	We <b>could be</b> friends.	<b>could ("what?" be</b> ("who?" we, "what?" friends))
87	Ja <b>ću raditi</b> sutra	<b>ću ("što?" raditi</b> ("tko?" ja, "kada?" sutra))	I <b>will work</b> tomorrow.	<b>will ("what?" work</b> ("who?" I, "when?" tomorrow))
90	Naš brod <b>je prešao</b> Atlantik u 7 dana.	<b>je ("što?" prešao</b> ("što?" brod ("čiji?" naš), "što?" Atlantik, "koje vrijeme?" dana ("koje vrijeme?" u, "koliko?" 7)))	Our ship <b>crossed</b> the Atlantic in 7 days.	<b>crossed ("what?" ship</b> ("whose?" our), "what?" Atlantic ("art?" the), "what_time?" days ("art?" in, "how_many?" 7))
100	Marija <b>ne izgleda</b> umorno.	<b>ne ("što?" izgleda</b> ("tko?" Marija, "kako?" umorno))	Mary <b>doesn't<sup>3</sup> look</b> tired.	<b>doesn't ("what?" look</b> ("who?" Mary, "how?" tired))
101			I <b>will have been working</b> for ten hours.	<b>will ("what?" have</b> ("what?" been ("what?" working ("who?" I, "how many?" hours ("how many?" for, "how many?" ten))))))

<sup>2</sup> Pojam "bilo što" predstavlja jedan čvor sastavljen od dvije riječi

<sup>3</sup> negacija "does not" ili skraćeno "doesn't" predstavlja jedan čvor



U tablici 4.1 pod rednim brojevima 39., 85., 86. dani su primjeri za modalne glagole u engleskom jeziku, dok je pod rednim brojevima 87., 90, 100 i 101. dan primjer za različita glagolska vremena (budućnost, prošlost i zanijekani oblik). Upotreba FNOK-a kod ostalih glagolskih vremena i modalnih glagola istovjetna je ovima.

Na primjer, *Future perfect continuous tense* tvori se kao:

*subjekt + pomoćni glagol "will" + pomoćni glagol "have" + past particip od "be" + present particip glavnog glagola.*

Hijerarhija u FNOK-u zadržava redoslijed tvorbe koja se pojavljuje u rečenici (vidi RBR 101. u tablici 4.1), stoga je hijerarhija u FNOK-u za *Future perfect continuous tense*:

- pomoćni glagol "will"
  - o pomoćni glagol "have"
    - past particip od "be"
      - present particip glavnog glagola
        - subjekt.

### **4.2.3 Imenice**

U hrvatskom jeziku imenice su riječi koje imenuju bića stvari i pojave. Gramatička obilježja imenica su rod (muški, ženski, srednji), broj (jednina i množina) te padež (nominativ, genitiv, dativ, akuzativ, vokativ, lokativ i instrumental) [115].

U engleskom jeziku imenice su riječi kojima imenujemo bića, predmete, pojave i osjećaje. One imaju rod (muški, ženski, srednji i zajednički), broj (jednina i množina), padeže i članove (*a, an, the*) [11], [12].

Najčešće se koristi u rečenici kao subjekt ili objekt glagolu ili kao objekt prijedloga [79].

#### **4.2.3.1 Odnos imenice prema glagolu**

Za glagol smo odredili da je uvijek na početku FNOK zapisa rečenice. Postavlja se pitanje odnosa imenice prema glagolu u FNOK zapisu rečenice. Za imenice hijerarhijski ovisne o glagolu definirano je sljedeće pravilo:

Imenice su u hijerarhiji ispod glagola. Navodimo ih nakon otvorene zagrada i navođenja pitanja na koje je ta imenica odgovor u ovisnosti o glagolu.

Imenice hijerarhijski ovisne o glagolu (odnosno procesnom čvoru) možemo kraće pisati kao:

$$G, I \rightarrow G(I), \text{ gdje je } G \text{ glagol, } I \text{ imenica}$$

preciznije,

$$G, I \rightarrow G(\text{"role?" } I), \text{ gdje je } G \text{ glagol, } I \text{ imenica, } \text{role?} \text{ označava wh-pitanje.}$$

Primjer:

- Vedran **vozi** auto. → **Vozi** ("tko?" Vedran, "što?" auto).
- Vedran **drives** a car. → **Drives** ("who?" Vedran, "what?" car ("art?" a))

#### 4.2.3.2 *Imenski predikat*

Imenski predikat je predikat koji se sastoji od spone, tj. pomoćnoga glagola biti i neke imenske riječi [115]. Za imenski predikat definirano je sljedeće pravilo.

Imenica kao dio imenskog predikata hijerarhijski je ovisna o pomoćnom glagolu te se u hijerarhiji nalazi ispod njega, ali u razini onih riječi koje su u službi subjekta.

Imenica kao dio imenskog predikata je hijerarhijski ovisna o pomoćnom glagolu (odnosno procesnom čvoru) s kojim tvori imenski predikat. To možemo kraće pisati kao:

$$PomGl, I_p \rightarrow PomGl(I_p), \text{ gdje je } PomGl \text{ pomoćni glagol, } I_p \text{ imenica kao dio imenskog predikata}$$

preciznije,

$$PomGl, I_p \rightarrow PomGl(\text{"role?" } I_p), \text{ gdje je } PomGl \text{ pomoćni glagol, } I_p \text{ imenica kao dio imenskog predikata, } \text{role?} \text{ označava wh-pitanje.}$$

Ako prethodni zapis proširimo imenicom, zamjenicom i slično koje su u službi subjekta imamo:

$$PomGl, S, I_p \rightarrow PomGl(S, I_p), \text{ gdje je } PomGl \text{ pomoćni glagol, } S \text{ riječ u službi subjekta, } I_p \text{ imenica kao dio imenskog predikata}$$

preciznije,

$PomGl, S, I_p \rightarrow PomGl ("role1?" S, "role2?" I_p)$ , gdje je  $PomGl$  pomoćni glagol,  $S$  riječ u službi subjekta,  $I_p$  imenica kao dio imenskog predikata,  $role1?$  i  $role2?$  označava wh-pitanja.

Primjer:

- Hrvatska **je država**.  $\rightarrow$  je ("tko?" Hrvatska, "što?" država)
- Petar **je usamljen čovjek**.  $\rightarrow$  je ("tko?" Petar, "što?" čovjek ("kakav?" usamljen))
- To **je kameni stol**.  $\rightarrow$  je ("što?" to, "što?" stol ("kakav?" kameni))

U navedenim rečenicama imenski predikati su *je država, je čovjek, je stol*.

#### 4.2.3.3 Apozicija i apozicijski skup

Ukoliko se u rečenici nalaze dvije ili više imenica od kojih je jedna hijerarhijski ovisna o glagolu, a ostale se odnose na imenicu ovisnu o glagolu (služe kao dopuna imenici (apozicija) i s njom se najčešće slažu u rodu, broju i padežu), tada se i imenice hijerarhijski organiziraju. Navodi se ona imenica koja je na višoj razini hijerarhije (koja je vezana za glagol), a nakon nje sve ostale imenice (i ostale vrste riječi, odnosno čvorovi) koje su s njom u vezi. Naravno, uz svaku imenicu potrebno je napisati i pripadno pitanje (ulogu) kojom ona pobliže označuju hijerarhijski nadređenu imenicu. Odnos imenice u odnosu na drugu imenicu definiran je sljedećim pravilom:

Apoziciju i apozicijski skup koji sačinjava jedna ili više apozicija sa svojim atributima u FNOK pretvaramo kao imenicu (skraćeno  $I_{ap}$ ) hijerarhijski ovisnu o drugoj imenici (odnosno čvoru  $I$ ) što možemo kraće pisati:

$I, I_{ap} \rightarrow I(I_{ap})$ , gdje je  $I$  imenica,  $I_{ap}$  dopuna imenici (apozicija)

preciznije,

$I, I_{ap} \rightarrow I("role?" I_{ap})$ , gdje je  $I$  imenica,  $I_{ap}$  dopuna imenici (apozicija),  $role?$  wh-pitanje.

Primjer:

- Moj **prijatelj Vedran** vozi auto.  $\rightarrow$  vozi ("tko?" Vedran ("koji?" prijatelj ("čiji?" moj)), "što?" auto)
- My **friend Sean** drives a car.  $\rightarrow$  drives ("who?" Sean ("which?" friend ("whose?" my)), "what?" car ("art?" a))

#### 4.2.3.4 Padeži imenica

U hrvatskom jeziku kod imenica u lokativu ili instrumentalu uz imenicu mogu se pojaviti prijedlozi (na, o, po, prema, pri, u, s, sa).

U engleskom jeziku sama imenica je nepromjenjiva po padežima, a padeži se tvore uz pomoć prijedloga koji se stavljaju ispred imenice (*of, to, about, with*).

Pravilo za prijedloge koje koristimo u tvorbi padeža imenica definirano je u nastavku.

Prijedloge koje koristimo u tvorbi padeža imenica pretvaramo u FNOK tako da su oni u hijerarhiji ovisni o imenici ispod koje se i stavljaju. U pitanje na koju je imenica odgovor ne uključujemo prijedlog već samu imenicu povezujemo s tim prijedlogom (prijedlog je na nižoj razini hijerarhije od imenice). Za pitanje (ulogu) prepisujemo pitanje (ulogu) koje dovodi do imenice.

To možemo zapisati kao slijedeće preslikavanje:

$Adj, I \rightarrow I (Adj)$ , gdje je *I* imenica, *Adj* prijedlog

preciznije,

$X, Adj, I \rightarrow X(\text{"role?" } I (\text{"role?" } Adj))$ , gdje je *X* bilo koja vrsta riječi, *I* imenica, *Adj* prijedlog (engl. *adjective*), *role?* je wh-pitanje.

Primjer:

- Knjiga je **na** **stolu**. → je ("što?" knjiga, "**gdje?"** **stolu** ("**gdje?"** **na**))
- Studenti razgovaraju **o** **rješenju**. → razgovaraju ("tko?" studenti, "**čemu?"** **rješenju** ("**čemu?"** **o**))
- I talk **with** my **brother**. → talk ("who?" I, "**who?"** **brother** ("**who?"** **with**, "whose?" my))
- Noa gave this book **to** a **friend**. → gave ("who?" Noa, "what?" book ("which?" this), "**who?"** **friend** ("**who?"** **to**, "art?" a))
- Students talk **about** the **solution**. → talk ("who?" students, "**what?"** **solution** ("**what?"** **about**, "art?" the))
- I need a lot **of** **money**. → need ("who?" I, "**what?"** **money** ("how much?" lot ("art?" a), "**what?"** **of**))

#### 4.2.3.5 Article (Član) - A, an, the

U engleskom jeziku razlikujemo neodređeni (*a, an*) i određeni član (*the*).

---

<sup>4</sup> primjer uloge sastavljene od dvije riječi

Neodređeni član *a* upotrebljava se ispred imenica u jednini te u određenim izrazima (*a few, a little, a lot, such a, quite a, half a*). Ako imenica počinje samoglasnikom, tada se umjesto *a* upotrebljava neodređeni član *an*.

Određeni član *the* se stavlja ispred imenice kada se misli na neki određeni ili poznati predmet ili kada se ta imenica spominje drugi put u nekom kontekstu [116], [11], [34].

U FNOK se član prevodi kao hijerarhijski ovisan o imenici. Za pitanje (ulogu) se stavlja oznaka "*art?*". Možemo pisati:

*Article, I* → *I(Article)*, gdje je *I* imenica, a *Article* član/article

preciznije,

*Article, I* → *I ("art?" Article)*,

gdje je *I* imenica, *Article* član/article, "*art?*" oznaka koja označava da se radi o članu.

Na primjer:

- I see **a** teacher. → see ("who?" I, "who?" **teacher ("art?" a)**)
- **The** train to New York is fast. → is ("what?" **train ("art?" the**, "where?" New York ("where?" to)), "what?" fast)
- Paul has **an** office in London. → has ("who?" Paul, "what?" **office ("art?" an)**, "where?" London ("where?" in))
- Tom is **a** good boy. → is ("who?" Tom, "what?" **boy ("what?" good ("art?" a))**)

#### 4.2.4 Pridjevi

Pridjevi su vrsta riječi koje pobliže označuju imenice, izražavajući osobine bića, stvari i pojava. Sklanjaju se po padežima te imaju rod i broj. Opisni pridjevi se mogu i stupnjevati ili komparirati. U hrvatskom jeziku razlikujemo [115], [110]:

- opisni ili kvalitativni pridjevi- označuju kakvo je što, npr. kakav? plav,
- posvojni ili posesivni pridjevi – označuju čije je što, npr. čiji? sestrin,
- gradivni ili materijalni pridjevi – označuju od čega je što?, npr. od čega? drven.

Pridjevi imaju dva oblika [115]:

- Neodređeni oblik - predmet se kvalificira, utvrđuje se kakav je. Na primjer, Ja volim bilo koji crven cvijet na livadi.
  - o crven (kakav?) – pridjev crven kazuje općenito, neodređeno
  - o U predikatnoj službi pridjev mora biti u neodređenom obliku.

- Određeni oblik – predmet se identificira, utvrđuje se koji je. Na primjer, *Ja volim taj crveni cvijet u tvojoj ruci.*
  - o crveni (koji?) – pridjev crveni određuje potpunu točnost

U engleskom jeziku pridjevi su riječi koje se dodaju imenici kako bi se pobliže označila svojstva tih imenica. Za razliku od pridjeva u hrvatskom jeziku, u engleskom jeziku pridjevi su nepromjenjiva vrsta riječi te se ne sklanjaju po padežima i nemaju rod i broj. Postoje tri vrste pridjeva: opisni (Kakvo je što?), posvojni (Čije je što?) te gradivni (Od čega je što?). Opisni pridjevi se kompariraju [12], [11].

#### 4.2.4.1 Pridjev kao dio imenskog predikata

Ako se pridjev koristi uz pomoćni glagol biti, odnosno pridjev je dio imenskog predikata, tada je pridjev ispod pomoćnog glagola. Za pridjeve hijerarhijski ovisne o pomoćnom glagolu (odnosno procesnom čvoru) možemo kraće pisati:

*PomGl, Pr → PomGl(Pr)*, gdje je *PomGl* pomoćni glagol, a *Pr* pridjev kao dio imenskog predikata

preciznije,

*PomGl, Pr → PomGl("role?" Pr)*, gdje je *PomGl* pomoćni glagol, *Pr* pridjev kao dio imenskog predikata, *role?* označava wh-pitanje.

Primjeri:

- Auto **je crn.** → **je** ("što?" auto, "**kakav?" crn**)
- The car **is black.** → **is** ("what?" car ("art?" the), "**what?" black**)
- Stol **je drven.** → **je** ("što?" stol, "**kakav?" drven**)
- The table **is wooden.** → **is** ("what?" table ("art?" the), "**what?" wooden**)
- Gramatika **je komplicirana.** → **je** ("što?" gramatika, "**kakva?" komplicirana**)
- Grammar **is complicated.** → **is** ("what?" grammar, "**what?" complicated**)

#### 4.2.4.2 Pridjev koji opisuje imenicu

Ako pridjev opisuje imenicu, tada je u FNOK-u pridjev hijerarhijski ispod imenice koju opisuje. Za pridjeve hijerarhijski ovisne o imenici (odnosno čvoru) možemo kraće pisati:

*I, Pr → I(Pr)*, gdje je *I* imenica, *Pr* pridjev

preciznije,

*I, Pr → I("role?" Pr)*, gdje je *I* imenica, *Pr* pridjev, *role?* označava wh-pitanje.

Primjer:

- Vedran vozi **crveni auto**. → vozi ("tko?" Vedran, "što?" auto ("koji?" crveni))
- Petar vozi **bratov auto**. → vozi ("tko?" Petar, "što?" auto ("čiji?" bratov))
- Josip crta **drvenim bojicama**. → crta ("tko?" Josip, "čime?" bojicama ("kakvim?" drvenim))
- Joseph draws with **wooden crayons**. → draws ("who?" Joseph, "what?" crayons ("what?" with, "what?" wooden))
- He is a **lonely man**. → is ("who?" he, "what?" man ("art?" a, "what?" lonely))
- **Studentova olovka**... → ...**olovka** ("čija?" studentova)
- **Student's pencil**... → ...**pencil** ("whose?" student's)

#### 4.2.4.3 Komparacija pridjeva

U oba jezika se opisni pridjevi kompariraju kroz tri stupnja ili tri stanja:

1. stupanj ili pozitiv: lijep, mlad, dobar, beautiful, young, good
2. stupanj ili komparativ: ljepši, mlađi, bolji, more beautiful, younger, better
3. stupanj ili superlativ: najljepši, najmlađi, najbolji, the most beautiful, the youngest, the best

Kod pretvaranja različitih stupnjeva pridjeva u FNOK koriste se pravila opisana u 4.2.4.1 (Pridjev kao dio imenskog predikata), 4.2.4.2 (Pridjev koji opisuje imenicu) te 4.2.3.5 (Article (Član) - A, an, the).

Primjeri:

- Poštenje je **najbolja** vrlina → je ("što?" poštenje, "što?" vrlina ("kakva?" najbolja))
- Honesty is **the best** policy. → is ("what?" honesty, "what?" policy ("what?" best ("art?" the)))
- Tom je **mlađi** od Petra. → je ("tko?" Tom, "kakav?" mlađi ("koga?" Petra ("koga?" od)))
- Tom is **younger** than Peter. → is ("who?" Tom, "what?" younger ("who?" Peter ("who?" than)))
- Josip je **najmlađi**. → je ("tko?" Josip, "kakav?" najmlađi)
- Joseph is **the youngest**. → is ("who?" Joseph, "what?" youngest ("art?" the))
- ... **the most beautiful** ... → ..."what?" beautiful ("art?" the, "how?" most)

#### 4.2.5 Zamjenice

Riječ koja zamjenjuje neku drugu riječ zove se zamjenica. U hrvatskom jeziku postoje sljedeće vrste zamjenica [115], [110]:

- osobne (lične): *ja, ti, on, ona, ono, mi, vi, oni, one, ona*
- posvojne: *moj, tvoj, njegov, njezin, naš, vaš, njihov; moja; moje; moji; moje; moja,...*

- pokazne: *ovaj, taj, onaj, ovakav, takav, onakav, ovolik, tolik, onolik,...*
  - o *To je moj auto. (That is my car.)*
  - o *Ovo je moja mačka. (This is my cat.)*
- neodređene: *netko, nešto, neki, nitko, ništa, nikakav, itko, išta, svačiji, svatko, bilo tko, ...*
- upitne i odnosne: *tko, što, koji, čiji, kakav, kolik*
- povratna: *sebe (se)*
  - o *Perem se (sebe).*
  - o *On govori o sebi.*
- povratno-posvojna: *svoj*

U engleskom jeziku postoje sljedeće vrste zamjenica [11], [12]:

- osobne (engl. *personal pronouns*): *I, you, he, she, it, we, you, they*
  - o *It is my bicycle.*
  - o *I work hard.*
- posvojne (engl. *possessive pronouns*): *my, your, his, her, its, our, your, their; mine, yours, his, hers, ours, yours, theirs*
  - o *It is **my** bicycle.*
- pokazne (engl. *demonstrative pronouns*): *this, that; these, those*
  - o ***That** is my car.*
  - o ***This** is my cat.*

Zamjenice *this* i *that* se u engleskom jeziku upotrebljavaju i kao pridjevi i tada stoje uz imenicu.

- o ***This** boy is my brother. (**Ovaj** dječak je moj brat.)*
- o ***That** car is red. (**Taj** auto je crven.)*
- neodređene (engl. *indefinite pronouns*):
  - o dijele se na one koje označavaju:
    - lica ili stvari (*somebody, someone, something, anybody, anyone, anything, nobody, no one, nothing, everybody, everyone, everything*).

Tvore se pomoću neodređenih zamjenica *some, any, no, every* i sufiksa *-body, -one, -thing*. Imaju oblik jednine i koriste se sa glagolom u jednini. Ne odnose se ni na kakvu prethodnu riječ ili sintagmu u rečenici.



- količinu: *another, any, both, each, either, few, least, less, little, a lot (of), lots (of), many, more, most, much, neither, none, one, other(s), plenty (of), several, some*

Mogu se koristiti i samostalno i ispred imenice, tj. kao pridjevi. Jedini izuzetak je zamjenica *none*, koja se koristi isključivo samostalno. Njen pridjevski ekvivalent je *no*.

- upitne (engl. *interrogative pronouns*): *who, which, what, whose, whom*
- odnosne (engl. *relative pronouns*): *who, which, what, whose, whom, that*
  - *The girl **who** lives in her house is her friend. (Djevojka **koja** živi u njevoj kući je njena prijateljica.)*
  - *It is the most interesting film **that** I saw. (To je najzanimljiviji film **koji** sam vidio.)*
- povratna (engl. *reflexive pronoun*): *myself, yourself, himself, herself, itself, ourselves, yourselves, themselves*
  - *I am washing myself. (Perem se (sebe).)*
  - *He speaks about himself. (On govori o sebi.)*
  - *You speak to yourself. (Govoriš sam sa sobom.)*

Po funkciji razlikujemo **imenične zamjenice** (zamjenjuju imenicu, u rečenici su samostalne te odgovaraju na pitanje *tko?, što?*) i **pridjevne zamjenice** (zamjenjuju pridjeve, u rečenici služe kao dodatak imenici, odgovaraju na pitanja *koji?, čiji?, kakav?, kolik?*; imaju razlikovanje po rodu: *koji, čija, kakvo, kolike* itd.) [127], [115], [110].

U FNOK se zamjenice pretvaraju ovisno o vrsti na dva načina.

#### 4.2.5.1 Iмениčne замјенице

Pravilo za imenične zamjenice glasi:

Imenične zamjenice koje u rečenici zamjenjuju imenice, hijerarhijski su ovisne o glagolu (njemu su podređene) što možemo kraće pisati kao:

$$G, Z_i \rightarrow G(Z_i), \text{ gdje je } G \text{ glagol, } Z_i \text{ imenična zamjenica}$$

preciznije,

$$G, Z_i \rightarrow G(\text{"role?" } Z_i), \text{ gdje je } G \text{ glagol, } Z_i \text{ imenična zamjenica, } \text{role?} \text{ označava wh-pitanje.}$$

Primjer:

- Osobne zamjenice
  - o **She** is a teacher. → is ("who?" **she**, "what?" teacher ("art?" a))
  - o **Ona** je nastavnica. → je ("tko?" **ona**, "što?" nastavnica)
  - o **They** stares. → stares ("who?" **they**)
  - o **I** work hard. → work ("who?" **I**, "how?" hard)
  - o **It** is my bicycle. → is ("what?" **it**, "what?" bicycle ("whose?" my))
- Pokazne zamjenice
  - o **Those** are stars. → are ("what?" stars, "**what?**" **those**)
  - o **Ono** su zvijezde. → su ("što?" zvijezde, "**koje?**" **ono**)
  - o **To** je moj auto. → je ("**što?**" **to**, "što?" auto ("čiji?" moj))
  - o **That** is my car. → is ("**what?**" **that**, "what?" car ("whose?" my))
  - o **Ovo** je tvoja mačka. → je ("**što?**" **ovo**, "što?" mačka ("čija?" tvoja))
  - o **This** is your cat. → is ("**what?**" **this**, "what?" cat ("whose?" your))
- Povratna zamjenica
  - o Ja češljam **sebe**. → češljam ("tko?" ja, "**koga?**" **sebe**)
  - o I brush **myself**. → brush ("who?" I, "**who?**" **myself**)
  - o On govori **o sebi**. → govori ("tko?" on, "**komu?**" **sebi** ("**komu?**" o))
  - o He speaks **about himself**. → speaks ("who?" he, "**who?**" **himself** ("**who?**" **about**))
  - o You speak **to yourself**. → speak ("who?" you, "**who?**" **yourself** ("**who**" to))
- Neodređene zamjenice\_koje se upotrebljavaju samostalno
  - o **Netko** je u kući. → je ("**tko?**" **netko**, "gdje?" kući ("gdje?" u))
  - o **Somebody** is in the house. → is ("**who?**" **somebody**, "where?" house ("where?" in, "art?" the))
  - o **Nitko** nije došao. → nije ("što?" došao, "**tko?**" **nitko**)
  - o **Nobody** came. → came ("**who?**" **nobody**)
  - o **Nikoga** nije bilo na plaži. → nije ("što?" bilo, "**koga?**" **nikoga**, "gdje?" plaži ("gdje?" na))
  - o There was **no-one** on the beach. → was ("where?" there, "**who?**" **no-one**, "where?" beach ("where?" on, "art?" the))
  - o **Ništa** nije ostalo. → nije ("što?" ostalo, "što?" ništa)
  - o There is **nothing** left. → is ("where?" there, "what?" left ("**what?**" **nothing**))
  - o **Svatko** se boji njega. → se ("što?" boji, "**tko?**" **svatko**, "koga?" njega)
  - o **Everybody** is afraid of him. → is ("what?" afraid, "**who?**" **everybody**, "whom?" him ("whom?" of))

#### 4.2.5.2 Pridjevne zamjenice

Pravilo za pridjevne zamjenice glasi:

Pridjevne zamjenice koje zamjenjuju pridjeve te u rečenici služe kao dodatak imenici, u hijerarhiji su podređene imenici, ovisno o imenici. To možemo kraće pisati kao:

$I, Z_p \rightarrow I(Z_p)$ , gdje je  $I$  imenica, a  $Z_p$  pridjevna zamjenica

preciznije,

$I, Z_p \rightarrow I(\text{"role?" } Z_p)$ , gdje je  $I$  imenica,  $Z_p$  pridjevna zamjenica, *role?* označava wh-pitanje.

Primjer:

- Posvojne zamjenice
  - o To je **moj** auto. → je ("što?" to, "što?" auto ("čiji?" **moj**))
  - o That is **my** car. → is ("what?" that, "what?" car ("whose?" **my**))
  - o Ovo je **tvoja** mačka. → je ("što?" ovo, "što?" mačka ("čija?" **tvoja**))
  - o This is **your** cat. → is ("what?" this, "what?" cat ("whose?" **your**))
  - o To su **moji** prijatelji. → su ("što?" to, "što?" prijatelji ("čiji?" **moji**))
  - o These are **my** friends. → are ("what?" these, "what?" friends ("whose?" **my**))
  
- Pokazne zamjenice koje zamjenjuju pridjeve
  - o **Ovaj** dječak je Josipov brat. → je ("tko?" dječak ("**koji?" ovaj**), "što?" brat ("čiji?" Josipov))
  - o **This** boy is Joseph's brother. → is ("who?" boy ("**which?" this**), "what?" brother ("whose?" Joseph's))
  - o **Taj** auto je crven. → je ("što?" auto ("**koji?" taj**), "kakav?" crven)
  - o **That** car is red. → is ("what?" car ("which?" **that**), "what?" red)
  
- Neodređene zamjenice koje zamjenjuju pridjeve
  - o Ja trebam **mnogo** novca. → trebam ("tko?" ja, "što?" novaca ("**koliko?" mnogo**))
  - o I need **a lot** of money. → need ("who?" I, "what?" money ("**how much?" lot** ("**art?" a**), "what?" of))
  - o Maja ima **nekoliko** zanimljivih knjiga. → ima ("tko?" Maja, "što?" knjiga ("**koliko?" nekoliko**, "kakvih?" zanimljivih))
  - o Maya has **some** interesting books. → has ("who?" Maya, "what?" books ("**how many?" some**, "what?" interesting))
  - o **Every** day is the same. → is ("what?" day ("**what?" every**), "what?" same ("art?" the))
  - o **Svaki** dan je isti. → je ("što?" dan ("**koji?" svaki**), "kakav?" isti)
  - o **Svi** su moji prijatelji dobri dečki. → su ("tko?" prijatelji ("**koji?" svi**, "čiji?" moji), "što?" dečki ("kakvi?" dobri))
  - o **All** my friends are good boys. → are ("who?" friends ("**what?" all**, "whose?" my), "what?" boys ("what?" good))

U ova pravila nisu uključene upitne zamjenice (engl. *interrogative pronouns*) i odnosne zamjenice (engl. *relative pronouns*), zato što upotrebom upitnih zamjenica dolazimo do upitnih rečenica, a upotrebom odnosnih zamjenica dolazimo do zavisno-složenih rečenica koje imaju više predikata te ih ne podrazumijevamo pod jednostavnim rečenicama. To je preostalo za daljnju analizu i doradu pravila.

## 4.2.6 Brojevi

Brojevi su riječi kojima se izriče točno koliko čega ima ili koje je što po redu [115]. Postoje dvije vrste brojeva, glavni i redni. **Glavni** brojevi (u engl. *cardinal numbers* (npr. jedan, dva, tri, one, two, three)) nam govore koliko je nečega (količinu), a **redni** brojevi (u engl. *ordinal numbers* (npr. prvi, drugi, treći, first, second, third)) nam govore redni broj nečega (poredak).

### 4.2.6.1 Pravilo broja

Brojevi u FNOK-u pripadaju imenicama, odnosno u hijerarhiji su podređene imenici, ovise o imenici. To možemo kraće pisati kao:

$$B, I \rightarrow I(B), \text{ gdje je } I \text{ imenica, } B \text{ broj}$$

preciznije,

$$B, I \rightarrow I(\text{"role?" } B), \text{ gdje je } I \text{ imenica, } B \text{ broj, } \text{role?} \text{ označava wh-pitanje.}$$

Primjer:

- Tom ima **dva** brata. → ima ("tko?" Tom, "što?" brata (**"koliko?" dva**))
- Tom has **two** brothers. → has ("who?" Tom, "what?" brothers (**"how many?" two**))
- Marko je bio **petnaesta** osoba... → je ("što?" bio ("tko?" Marko, "što?" osoba (**"koja?" petnaesta**)),...)
- Mark was **the fifteenth** person... → was ("who?" Mark, "what?" person (**"what?" fifteenth ("art?" the)**)..)
- U hotelu su **sedamdeset i dva** studenta. → su ("tko?" studenta (**"koliko?" sedamdeset i dva**), "gdje?" hotelu ("gdje?" u))
- There are **seventy-two** students in the hotel. → are ("where?" there, "who?" students (**"how many?" seventy-two**), "where?" hotel ("where?" in, "art?" the))

Uz brojeve se vrlo često pojavljuju mjerne jedinice.

### 4.2.6.2 Mjerne jedinice

Mjerne jedinice su odabrane, dogovorene i objavljene poznate vrijednosti mjernih (fizikalnih) veličina s kojima se pri mjerenju uspoređuju sve druge istovrsne veličine. Vrijednosti, nazivi, znakovi i uporaba mjernih jedinica određeni su međunarodnim dogovorima, na kojima se temelje norme i mjeriteljski zakoni pojedinih zemalja [42]. Neki od primjera mjernih jedinica su kilogram (kg), metar (m), sekunda (s) i slično. Odnos između broja i mjerne jedinice najčešće možemo opisati pitanjima *čega?* ili *što?*

(npr. *18 što?* ili *18 čega?* - *kilograma, metara, sati, godina* i slično). Ponekad će se tražiti znanje koje će zahtijevati odnos između broja i mjerne jedinice. Na primjer, ako tražimo znanje za *Čega 18?* kao odgovor ćemo očekivati 18 godina, ali i 18 ljudi, osoba, krumpira... a rjeđe 18 kg, metara, sati... Stoga uvodimo pravilo mjerne jedinice.

Mjerne jedinice su na višoj razini hijerarhije od broja, odnosno vrijednosti mjerne jedinice. Za pitanje (ulogu) uz mjernu jedinicu prepisujemo ono pitanje uz broj. Pišemo:

$JMJ, B \rightarrow MJ(B)$ , gdje je  $JMJ$  mjerna jedinica, a  $B$  broj

preciznije,

$JMJ, B \rightarrow \text{"role1?" } MJ(\text{"role2?" } B)$ , gdje je  $JMJ$  mjerna jedinica,  $B$  broj, a  $role1?$  i  $role2?$  označavaju wh-pitanje.

Kod glavnih brojeva,  $role1=role2$  dok se kod rednih brojeva  $role1?$  i  $role2?$  mogu razlikovati. U slučaju kada se  $role1?$  i  $role2?$  razlikuju, za  $role1?$  definira se ona uloga (pitanje) koja opisuje cjelinu (broj i mjernu jedinicu) npr. *koliko?, kada?, how much?, how many?, when?* i slično, a za  $role2?$  ono wh-pitanje koje opisuje broj (*koji?, koja? koje? koliko, which?* i slično).

Primjer:

- ...200 metara... → ...*"koliko?" metara ("koliko?" 200)*...
- ...20 kg... → ...*"koliko?" kg ("koliko?" 20)*...
- ...1999 godine... → ...*"kada?" godine ("koje?" 1999)*...
- ...6 sati... → ...*"koliko?" sati ("koliko?" 6)*
- ...six o'clock... → ...*"what time?" o'clock ("what time?" six)*
- ... 10 pounds... → ...*"how many?" pounds ("how many?" 10)*

U rečenici *Hotel je udaljen 200 m.* Ako se postavi pitanje *Koliko je hotel udaljen?* kao odgovor se dobiva *200 m.* Ako se postavi pitanje *Koliko metara je hotel udaljen?* kao odgovor se dobiva *200.* U ovom primjeru se za oba čvora *200* i *m* kao ulogu (pitanje) definira isto wh-pitanje *koliko?*

Iznimno, u nekim slučajevima se  $role1?$  i  $role2?$  razlikuju. Primjerice kod godina (npr. u rečenici *Trgovina je otvorena 2016. godine.*) ako kod postavljanja pitanja (odnosno upitne rečenice) želimo dobiti odgovor *2016. godine*, tada postavljamo pitanje *Kada...? (Kada je otvorena trgovina?)*. Ako želimo dobiti odgovor *2016.*, tada postavljamo pitanje *Koje godine...? (Koje godine je trgovina otvorena?)*. Prema tome, pitanje (uloga) uz čvor *godina* je *kada?*, a uz broj *2016.* je *koje?*

#### 4.2.6.2.1 Pravilo svojstva

Dodatno, prethodno pravilo se može proširiti.

Ako je dano svojstvo (osobina, pojava, karakteristika, atribut entiteta i sl.) koje se mjeri brojem i ima mjernu jedinicu, tada je hijerarhija FNOK oblika:

$B, SV, JMJ \rightarrow SV (JMJ (B))$ , gdje je  $SV$  svojstvo (najčešće imenica ili pridjev),  $JMJ$  mjerna jedinica, a  $B$  broj

preciznije,

$B, SV, JMJ \rightarrow SV ("role1?" JMJ ("role2?" B))$ , gdje je  $SV$  svojstvo (najčešće imenica ili pridjev),  $JMJ$  mjerna jedinica,  $B$  broj, a  $role1?$  i  $role2?$  označavaju wh-pitanja koja su najčešće jednaka na obje razine.

Primjer:

- Marko je **visok 185 cm**.  $\rightarrow$  je ("tko?" Marko, "što?" visok ("koliko?" cm ("koliko?" 185)))
- Tomislav kupuje **10 kila krumpira** u trgovini.  $\rightarrow$  kupuje ("tko?" Tomislav, "što?" krumpira ("koliko?" kila ("koliko?" 10)), "gdje?" trgovini ("gdje?" u))
- Thomas buys **10 pounds of potatoes** in the store.  $\rightarrow$  buys ("who?" Thomas, "what?" potatoes ("how many?" pounds ("how many?" 10), "what?" of), "where?" store ("where?" in, "art?" the))

#### 4.2.6.2.2 Pravilo domene

Ako imamo vrijednost podatka iz domene (skupa, klase, pojedinačnog prema općem) i ime domene kao dva čvora međusobno semantički ovisna, tada prvo navodimo vrijednost podatka iz domene, a zatim ime domene, odnosno vrijednost podatka iz domene nadređena je domeni. Pišemo:

$pD, D \rightarrow pD (D)$ , gdje je  $pD$  vrijednost podatka iz domene, a  $D$  domena

preciznije,

$pD, D \rightarrow pD ("role?" D)$ , gdje je  $pD$  vrijednost podatka iz domene,  $D$  domena,  $role?$  označava wh-pitanje.

Primjer:

- Hrvoje Horvat posjeduje vozačku dozvolu **B kategorije**  $\rightarrow$  posjeduje ("tko?" Hrvoje Horvat, "što?" dozvolu ("kakvu?" vozačku ("koje?" kategorije ("koje?" B))))
- Hrvoje Horvat has a **B category** driving licence.  $\rightarrow$  has ("who?" Hrvoje Horvat, "what?" licence ("art?" a, "what?" driving ("which?" category ("which?" B))))

Napomena: U ovom se primjeru slovo *B* tretira po principu broja, a *kategorije*, odnosno *category* je domena.

#### 4.2.7 Prilozi

Prilozi su nepromjenjiva vrsta riječi koje se najčešće dodaju glagolima za izricanje okolnosti radnje (mjesto, vrijeme, način, uzrok, svrha i količina). Prilozi se mogu dodati i pridjevima (npr. prilično dobar), priložima (npr. malo prije) i imenicama (npr. mnogo hrastića) [115].

U hrvatskom jeziku razlikujemo [115], [110]:

- Vremenski prilozi (kada?, otkada?, dokada?):
  - o **kada?** (*danās, večeras, noćas, jučer, sinoć, preksinoć, danas, sutra, preksutra, ljetos, proljetos, jesenas, zimus, proljeti, ljeti, jeseni, zimi, lani, preklani, obdan, obnoć, odmah, smjesta, sada, tada, onda, ikada, bilo kada, nikada, nekada, ponekad, katkad, uvijek, svagda, često, rijetko, rano, kasno, prije, poslije, potom, nedavno, skoro, uskoro, napokon, ...*)
  - o **otkada?** (*odasad, otad, oduvijek, odavna, odmalena ...*)
  - o **dokada?** (*dosad, dotad, dogodne, dovečer ...*)
- Mjesni prilozi (gdje?, kamo?, kuda?, odakle?, dokle?, otkud?, dokud?):
  - o **gdje?** (*ovdje, tu, ondje, negdje, igdje, nigdje, ponegdje, gore, dolje, unutra, vani, sprijeda, straga, ...*)
  - o **kamo?** (*ovamo, onamo, tamo, nekamo, nikamo, ikamo, naprijed, natrag ...*)
  - o **kuda?** (*ovuda, onuda, tuda, nikuda, nekuda, ikuda, kojekuda ...*)
  - o **otkuda? odakle?** (*odavde, otud, odatle, odonud, niotkuda, odozgo, odozdo, odostraga, izdaleka, izvana, izbliza ...*)
  - o **dokle? dokud?** (*donekle, doвле, dotle, donle*)
- Načinski prilozi (kako?, kojim načinom?):
  - o **kako?** (*ovako, tako, onako, nikako, nekako, ikako, kojekako, svakojako, kriomice, potajice, poimence, iznenada, odjednom, jedva, svejedno* i tako dalje.
  - o Načinski prilozi nastaju i od pridjeva:
    - Od pozitivna, komparativa ili superlativa opisnih pridjeva: *blago, blaže, najblaže, brzo, daleko, dalje, jako, lako, lijepo, teško, teže,*

*najteže, tiho, veselo, zlo, žestoko* i tako dalje (na primjer, *Dijete se igra (kako?) veselo.*)

- Od odnosnih pridjeva na –ski: *bratski, gospodarski, hrvatski, ljudski, slavenski, školski, vojnički* i tako dalje (na primjer, *Govorimo (kako?) hrvatski.*)
- Uzročni prilozi (zašto?, zbog čega?):
  - **zašto?, zbog čega?** (*zato, stoga*)
- Količinski prilozi (koliko?, koliko puta?):
  - **koliko?** (*ovoliko, toliko, onoliko, nekoliko, malo, premalo, više, previše, prekoviše, najviše, ponajviše, manje, najmanje, ponajmanje, dosta, odveć, opet, još, sasvim, potpuno, previše*)
  - **koliko puta?** (*jedanput, dvaput, triput, stoput*)
- Posljedični prilozi (s kojom posljedicom?, s kojim ishodom?):
  - **S kojim ishodom?** (*uzalud, uzaman, utaman*)
- Postoji i niz priloga koji ne odgovaraju na gornja pitanja, dodaju se ne samo glagolima već i drugim vrstama riječi pa i cijeloj rečenici te izriču:
  - Pouzdanost, istinitost onoga na što se prilog odnosi: *naravno, pouzdano, neosporno, neprijeporno, nesumnjivo*
  - Veću ili manju pouzdanost, vjerojatnost onoga na što se odnosi: *sigurno, zasigurno, jamačno, vjerojatno*
  - Sumnju u ono na što se prilog odnosi: *navodno, tobože, naizgled*
  - Neočekivanost onoga na što se prilog odnosi: *ipak*
  - Ograničenost onoga na što se prilog odnosi: *jedan, jedino*
  - ...

U engleskom jeziku, prilozi su riječi koje se najčešće prilažu glagolima i pobliže određuju okolnosti glagolske radnje (mjesto, vrijeme, način, količinu i uzrok). Prilog se nalazi iza glagola kojeg modificira [12].

Priloge dijelimo na:

1. priloge mjesta: *here, there, down,...*
2. priloge vremena: *yesterday, today, now, tonight, often, always...*
3. priloge načina: *softly, slowly, quickly, carefully, ...*
4. priloge količine: *enough, quite, very, too much, too little, ...*



Ponekad prilozima mogu modificirati pridjeve te ih tako jačati ili slabiti. Također, neke vrste priloga mogu modificirati druge priloge mijenjajući njihov stupanj ili preciznost. Priloge u FNOK pretvaramo na dva načina.

#### 4.2.7.1 Prilozi vezani za glagole

Ako su prilozima vezani za glagole, onda su u FNOK-u oni hijerarhijski ovisni o glagolima, odnosno u hijerarhiji su podređeni glagolu. To možemo kraće pisati kao:

*Prilog, G → G (Prilog)*, gdje je *G* glagol, *Prilog* prilog

preciznije,

*Prilog, G → G ("role?" Prilog)*, gdje je *G* glagol, *Prilog* prilog, *role?* označava wh-pitanje.

Primjer:

- Medvjedi jedu **pohlepno**. → jedu ("tko?" medvjedi, "**kako?" pohlepno**)
- The bears eat **greedily**. → eat ("who?" bears ("art?" the), "**how?" greedily**)
- Marko **danas brzo** vozi auto. → vozi ("tko?" Marko, "što?" auto, "**kada?" danas, "kako?" brzo**)
- Mark drives a car **quickly today**. → drives ("who?" Mark, "what?" car ("art?" a), "**when?" today, "how?" quickly**)

#### 4.2.7.2 Prilozi vezani za druge vrste riječi (osim glagola)

Ako su prilozima vezani za pridjeve ili druge priloge, tada ih u FNOK prevodimo kao hijerarhijski ovisne o pridjevima, imenicama ili prilozima koje modificiraju. To možemo kraće pisati kao:

*Prilog, X → X (Prilog)*, gdje je *Prilog* prilog, *X* pridjev, imenica ili prilog kojeg *Prilog* modificira

preciznije,

*Prilog, X → X ("role?" Prilog)*, gdje je *Prilog* prilog, *X* pridjev, imenica ili prilog kojeg *Prilog* modificira, *role?* označava wh-pitanje.

Primjer:

- Ti izgledaš **jako** dobro. → izgledaš ("tko?" ti, "kako?" dobro ("**kako?" jako**))
- You look **absolutely** fabulous! → look ("who?" you, "how?" fabulous ("**how?" absolutely**))
- Marija svira violinu **izuzetno** dobro. → ("tko?" Marija, "što?" violinu, "kako?" dobro ("**kako?" izuzetno**))
- Mary plays the violin **extremely** well. → plays ("who?" Mary, "what?" violin ("art?" the), "how?" well ("**how?" extremely**))

- Marko danas **vrlo** brzo vozi auto. → vozi ("tko?" Marko, "što?" auto, "kada?" danas, "kako?" brzo (**"koliko?" vrlo**))
- Mark drives a car **very** quickly today. → drives ("who?" Mark, "what?" car ("art?" a), "when?" today, "how?" quickly (**"how?" very**))

#### 4.2.8 Prijedlozi

Prijedlozi su nepromjenjiva vrsta riječi koja služi da bi se pokazali odnosi među bićima, stvarima i pojavama te utječu na padež riječi uz koji stoje. Najčešće su udruženi s imenicama i imeničnima zamjenicama, ali mogu se odnositi i na osamostaljene riječi (pridjeve, pridjevne zamjenice i broj) [115], [110]. Prijedlozi ne mogu stajati samostalno.

U engleskom jeziku prijedlozi (engl. *prepositions*) su vrsta riječi koja pokazuje odnose između predmeta, bića i pojava. Ne mogu stajati samostalno te su uvijek udruženi s imenicama [11]. Prijedlozi se grupiraju s drugim vrstama riječi u prijedložne fraze (engl. *prepositional phrases*) koje su najčešće sličnog oblika: prijedlog praćen odrednicama (engl. *determiners*) te sa jednim ili dva pridjeva praćenim imenicama ili zamjenicama. Cijela ta fraza ima modificirajuću ulogu, ponaša se kao pridjev ili prilog locirajući nešto u vremenu i prostoru, modificira imenicu ili kazuje kada, gdje ili pod kojim uvjetima se nešto dogodilo [34].

Za pretvorbu prijedloga u FNOK postavlja se sljedeće pravilo.

U FNOK-u su prijedlozi hijerarhijski ispod imenica, odnosno uz onu vrstu riječi uz koju su udruženi (imenica, imenična zamjenica ili na osamostaljene riječi (pridjeve, pridjevne zamjenice i broj)). Svaki prijedlog pripada onoj imenici na koju se odnosi. Prvo se navodi imenica, a nakon nje prijedlog. Pitanje (uloga) kod prijedloga se prepisuje od pripadne mu imenice hijerarhijski nadređene. Pišemo:

*Prij, I* → *I (Prij)*, gdje je *I* imenica, a *Prij* prijedlog

preciznije,

*X, Prij, I* → *X("role1?" I ("role1?" Prij))*, gdje je *X* bilo koja vrsta riječi, *I* imenica, *Prij* prijedlog, a *role1?* je wh-pitanje.

Primjer:

- Knjiga je **na** stolu. → je ("što?" knjiga, "**gdje?"** stolu (**"gdje?" na**))
- The book is **on** the table. → is ("what?" book ("art?" the), "**where?"** table (**"where?" on**, "art?" the))

- I do my homework every day **at** five o'clock. → do ("who?" I, "what?" homework ("whose?" my), "when?" day ("what?" every), "when?" o'clock ("**when?**" five ("**when?**" **at**)))
- Ja radim svoju domaću zadaću svaki dan **u** pet sati. → radim ("tko?" ja, "što?" zadaću ("čiju?" svoju, "kakvu?" domaću), "kada?" dan ("koji?" svaki), "**koliko?**" sati ("**koliko?**" **u**, "koliko?" pet))
- Jack je **u** kući. → je ("tko?" Jack, "**gdje?**" kući ("**gdje?**" **u**))
- Jack is **in** the house. → is ("who?" Jack, "**where?**" house ("**where?**" **in**, "art?" the))

#### 4.2.9 Napomena

U gore navedenim pravilima nisu obuhvaćeni:

- veznici,
- usklici,
- čestice,
- odnosne zamjenice (opisane u poglavlju 4.2.5) i
- upitne zamjenice (opisane u poglavlju 4.2.5).

Odnosne zamjenice su po semantici sličnije veznicima. Veznici i odnosne zamjenice tvore složene rečenice koje također ne opisujemo. Za njih je potrebno naći odgovarajuća pravila kako bi se složene rečenice obuhvatile, a to je izvan opsega ovog rada. Upitne zamjenice tvore upitne rečenice koje sadrže znanje kao i izjavne rečenice te imaju jedan čvor čije je ime nepoznato, ali je poznato wh-pitanje koje mu pripada.

### 4.3 Pravila prevođenja upitnih rečenica prirodnog jezika u QFNOK

Kao što izjavne rečenice prirodnog jezika kojima iskazujemo znanje pretvaramo u FNOK zapis kako bismo znanje mogli zapisati u relacijsku bazu podataka, tako upitne rečenice prirodnog jezika kojima propitujemo znanje iskazano izjavnim rečenicama pretvaramo u formalizirani NOK zapis za pitanja iskazana upitnim rečenicama (skraćeno QFNOK zapis) kako bismo i njih mogli upisati u relacijsku bazu podataka te pronaći odgovor na postavljeno pitanje. Kod propitivanja, sustavu je važno da FNOK zapis i QFNOK zapis imaju sličnu strukturu. Kod prevođenja upitnih rečenica u QFNOK vrijede ista pravila prevođenja kao i kod prevođenja jednostavnih izjavnih rečenica u FNOK uz neka dodatna pravila. Odnosi u hijerarhiji između čvorova u QFNOK-u jednaki su onima u FNOK-u. Za razliku od FNOK-a gdje svaki čvor ima ulogu (pitanje), u QFNOK-u važna je samo ona uloga kod onog čvora koji je nepoznat.

U QFNOK-u se uvodi nekoliko dodatnih znakova:

- "X" – predstavlja poziciju odgovora u FNOK zapisu rečenice prirodnog jezika.
- "\_" – zamjenjuje ulogu (role) koju neki čvor ima u odnosu na nadređeni čvor. Ta uloga nije poznata i nije nužna za propitivanje znanja i pronalazak odgovora.
- "&" – stavlja se ispred onih čvorova nastalih iz upitnih rečenica čiji oblik može biti promjenjiv i time prilično utjecati na potpunost odgovora. Kod propitivanja se čvor ispred kojeg stoji taj znak zanemaruje. Na primjer, ispred article (*a, an, the*), u različitim oblicima glagolskih vremena i tvorbe izjavnih i upitnih oblika (na primjer, *do/does, am/is/are* i slično) i tako dalje.

Na primjer, za znanje iskazano rečenicom prirodnog jezika *Danijel vozi.*, odnosno odgovarajućeg FNOK zapisa *vozi ("tko?" Danijel)* možemo postaviti pitanje iskazano upitnom rečenicom *Tko vozi?* koje propituje znanje iskazano prethodnom rečenicom. Poštujući pravila prevođenja rečenica u FNOK te hijerarhijske ovisnosti među čvorovima, odgovarajući QFNOK zapis za upitnu rečenicu je *vozi ("tko?" X)*.

Razlikujemo dva osnovna oblika QFNOK zapisa ovisno o vrsti upitne rečenice koja je postavljena (Da/Ne pitanja i pitanja koja počinju upitnom zamjenicom (wh-pitanja)). U nastavku slijedi detaljnije objašnjenje.

#### 4.3.1 Da/Ne pitanje (engl. polar question)

QFNOK zapis za upitnu rečenicu, koja kao odgovor daje Da/Ne (Yes/No), odnosno za Da/Ne pitanje (engl. polar question), ne sadrži znak X jer su svi čvorovi poznati. Osim toga, nema ni iskazanih uloga, već na svim mjestima uloga stoje "\_". Znak "&" može se pojaviti kod čvorova *li, does, do, a, an, the* i slično koji se ili ne nalazi u FNOK zapisu ili se mogu nalaziti u nekom drugom obliku. Taj čvor se kod propitivanja zanemaruje. Kod uparivanja FNOK i QFNOK zapisa svi se čvorovi moraju podudarati (osim čvora označenog sa &). Ako se svi čvorovi podudaraju, odgovor je DA. U slučaju da se svi čvorovi ne podudaraju odgovor je NE.

**Općeniti zapis QFNOK-a za Da/Ne pitanja:**

**Process node ( \_ node 1, \_ node 2 ( \_ node 3 ( \_ node 4, ...), ...), \_ node n)**

gdje su *node 1* do *node n* čvorovi, a *process node* je procesni čvor.

Tablica 4.2 prikazuje upitne rečenice, odnosno Da/Ne pitanja nastalog na temelju izjavne rečenice. Za svaku od rečenica prikazan je FNOK i QFNOK zapis te odgovor na postavljeno pitanje. Primjeri su na hrvatskom i engleskom jeziku.

Tablica 4.2. Primjeri Da/Ne pitanja

Rečenica	FNOK	Upitna rečenica	QFNOK	Odgovor
Život je ponekad kompliciran.	je ("što?" život, "kakav?" kompliciran ("kada?" ponekad))	Je li život ponekad kompliciran?	je ( ( & li), _ život, _ kompliciran ( _ ponekad))	DA
Luka jede sendvič.	jede ("tko?" Luka, "što?" sendvič)	Jede li Luka sendvič?	jede ( ( & li), _ Luka, _ sendvič)	DA
Natjecanja su u zadnjem tjednu travnja.	su ("što?" natjecanja, "kada?" travnja ("kada?" tjednu ("kada?" u, "kojem?" zadnjem)))	Jesu li natjecanja u zadnjem tjednu travnja?	jesu ( ( & li), _ natjecanja, _ travnja ( _ tjednu ( _ u, _ zadnjem)))	DA
Natjecanja su u zadnjem tjednu travnja.	su ("što?" natjecanja, "kada?" travnja ("kada?" tjednu ("kada?" u, "kojem?" zadnjem)))	Jesu li natjecanja u zadnjem tjednu svibnja?	jesu ( ( & li), _ natjecanja, _ svibnja ( _ tjednu ( _ u, _ zadnjem)))	NE
Life is sometimes complicated.	is ("what?" life, "what?" complicated ("when?" sometimes))	Is life sometimes complicated?	is ( _ complicated ( _ life, _ sometimes))	YES
Lucas eats a sandwich.	eats ("who?" Lucas, "what?" sandwich ("art?" a))	Does Lucas eat a sandwich?	eat ( ( & does) ( _ Lucas, _ sandwich ( & a))	YES
Competitions are in the last week of April.	are ("what?" competitions, "when?" April ("when?" week ("when?" in, "what?" last ("art?" the)), "what?" of))	Are competitions in the last week of April?	are ( _ competitions, _ April ( _ of, _ week ( _ in, _ last ( _ the)))	YES
Competitions are in the last week of April.	are ("what?" competitions, "when?" April ("when?" week ("when?" in, "what?" last ("art?" the)), "what?" of))	Are competitions in the last week of May?	are ( _ competitions, _ May ( _ of, _ week ( _ in, _ last ( _ the)))	NO

#### 4.3.2 Upitne rečenice koje započinju wh-pitanjima

Kod upitnih rečenica koje započinju wh-pitanjem razlikujemo:

1. upitne rečenice kojima je odgovor jedna ili više semantički povezanih riječi (odnosno jedan ili više međusobno ovisnih čvorova) te
2. upitne rečenice kojima je odgovor glagol (odnosno procesni čvor) ili pomoćni glagol semantički povezan s još nekoliko vrsta riječi (procesni čvor i nekoliko čvorova njemu podređenih) kako bi se dobila potpuna semantika odgovora.

Sukladno tome razlikujemo dvije vrste QFNOK zapisa:

1. Traženi odgovor je jedan ili više hijerarhijski ovisnih čvorova, odnosno  $X$  je unutar zapisa na nižoj razini hijerarhije.
2. Traženi odgovor je sam procesni čvor ili procesni čvor i nekoliko njemu podređenih čvorova kako bi se dobila potpuna semantika; odnosno  $X$  je na početku zapisa, na najvišoj razini hijerarhije.

#### **4.3.2.1 Traženi odgovor na nižim razinama hijerarhije**

Kod upitnih rečenice koje započinju wh-pitanjem, a kojima je odgovor jedna ili više semantički povezanih riječi (odnosno jedan ili više međusobno ovisnih čvorova),  $X$  se nalazi unutar QFNOK zapisa. Njegova uloga je upravo wh-pitanje kojim započinje upitna rečenica. Svi čvorovi hijerarhijski se organiziraju prema pravilima prevođenja (objašnjeno u poglavlju 4.2), ali za njihove uloge stavljamo "\_" ili "&" ovisno o tome uzimamo li čvor u obzir kod traženja odgovora ili ne. Nakon uparivanja FNOK i QFNOK zapisa, kao odgovor dobiva se jedan ili više hijerarhijski ovisnih čvorova ovisno o tome koliko je općenito ili specifično pitanje postavljeno. Također, moguće je dobiti više istih ili različitih odgovora na isto pitanje, ovisno o polaznom znanju i postavljenom pitanju.

#### **Općeniti zapis QFNOK-a:**

**process node ( \_ node 1, \_ node 2 ( \_ node 3 ("wh-pitanje?"  $X$ , ...), ...), ...)**

gdje je *wh-pitanje* ono wh-pitanje preuzeto iz upitne rečenice, *node 1* do *node n* su čvorovi, *process node* je procesni čvor, a  $X$  je traženi odgovor, odnosno tražena pozicija odgovora.  $X$  se može pojaviti umjesto bilo kojeg čvora (node).

Tablice 4.3 i 4.4 prikazuju primjer znanja iskazanog pomoću nekoliko rečenica prirodnog jezika te postavljenih upitnih rečenica koje propituju iskazano znanje. Rečenice prirodnog jezika pretvorene su u FNOK, dok su upitne rečenice pretvorene u QFNOK zapise. Osim toga dani su odgovori na postavljeno pitanja te redni broj rečenice iz tablice 4.3 u kojoj je pronađen odgovor. Svaka od tablica se sastoji od A i B dijela. A dio je za hrvatski jezik, a B za engleski jezik. Tablica 4.3 je isječak iz tablice 19 i tablice 22 iz Privitka 2, a tablica 4.4 je isječak iz tablice 20 i tablice 23 iz Privitka 2.

Tablica 4.3. Primjer rečenica s pripadajućim FNOK zapisima na hrvatskom (A dio) i engleskom (B dio) jeziku

RBR	Rečenica	FNOK
<b>A</b>		
<b>Hrvatski jezik</b>		
1	Danijel vozi.	vozi ("tko?" Danijel)
2	Jakov vozi kamion.	vozi ("tko?" Jakov, "što?" kamion)
3	Tomislav vozi auto.	vozi ("tko?" Tomislav, "što?" auto)
4	Ivan vozi automobil.	vozi ("tko?" Ivan, "što?" automobil)
5	Moj prijatelj Vedran vozi auto.	vozi ("tko?" Vedran ("koji?" prijatelj ("čiji?" moj)), "što?" auto)
6	Katarina vozi crveni auto na autocesti.	vozi ("tko?" Katarina, "što?" auto ("koji?" crveni), "gdje?" autocesti ("gdje?" na))
7	Katarina vozi zeleni auto na utrci.	vozi ("tko?" Katarina, "što?" auto ("koji?" zeleni), "gdje?" utrci ("gdje?" na))
8	Petar vozi bratov auto.	vozi ("tko?" Petar, "što?" auto ("čiji?" bratov))
54	Marko danas vrlo brzo vozi auto.	vozi ("tko?" Marko, "što?" auto, "kada?" danas, "kako?" brzo ("koliko?" vrlo))
<b>B</b>		
<b>Engleski jezik</b>		
1	Daniel drives.	drives ("who?" Daniel)
2	Jacob drives a truck.	drives ("who?" Jacob, "what?" truck ("art?" a))
3	Thomas drives a car.	drives ("who?" Thomas, "what?" car ("art?" a))
4	John drives an automobile.	drives ("who?" John, "what?" automobile ("art?" an))
5	My friend Sean drives a car.	drives ("who?" Sean ("who?" friend ("whose?" my)), "what?" car ("art?" a))
6	Catherine drives a red car on the highway.	drives ("who?" Catherine, "what?" car ("art?" a, "what?" red), "where?" highway ("where?" on, "art?" the))
7	Catherine drives a green car in the race.	drives ("who?" Catherine, "what?" car ("art?" a, "what?" green), "where?" race ("art?" the, "where?" in))
8	Peter drives his brother's car.	drives ("who?" Peter, "what?" car ("whose?" brother's ("whose?" his)))
54	Mark drives a car very quickly today.	drives ("who?" Mark, "what?" car ("art?" a), "when?" today, "how?" quickly ("how?" very))

Tablica 4.4. Upitne rečenice s pripadajućim QFNOK zapisima i odgovorima temeljenim na znanju iz tablice 4.3 na hrvatskom (A dio) i engleskom (B dio) jeziku

RBR	Upitna rečenica	QFNOK	Odgovor	Rečenica u kojoj je pronađen odgovor
<b>A</b>				
<b>Hrvatski jezik</b>				
68	Tko vozi?	vozi ("tko?" X)	Danijel Jakov Tomislav Ivan Moj prijatelj Vedran Katarina Katarina Petar Marko	1 2 3 4 5 6 7 8 54
69	Tko vozi auto?	vozi ("tko?" X, _ auto)	Tomislav Ivan Moj prijatelj	3 4 5

			Vedran Katarina Katarina Petar Marko	6 7 8 54
70	Čiji auto vozi Petar?	vozi ( _ Petar, _ auto ("čiji?" X))	bratov	8
71	Kako Marko danas vozi auto?	vozi ( _ Marko, _ auto, _ danas, "kako?" X)	vrlo brzo	54
72	Kada Marko vrlo brzo vozi auto?	vozi ( _ Marko, _ auto, "kada?" X, _ brzo ( _ vrlo))	danas	54
73	Koji auto vozi Katarina?	vozi ( _ Katarina, _ auto ("koji?" X))	crveni zeleni	6 7
74	Koji auto vozi Katarina na utrci?	vozi ( _ Katarina, _ auto ("koji?" X), _ utrci ( _ na))	zeleni	7
75	Gdje Katarina vozi crveni auto?	vozi ( _ Katarina, _ auto ( _ crveni), "gdje?" X)	na autocesti	6
76	Što Jakov vozi?	vozi ( _ Jakov, "što?" X)	kamion	2
<b>B</b>	<b>Engleski jezik</b>			
68	Who drives?	drives ("who?" X)	Daniel Jacob Thomas John My friend Sean Catherine Catherine Peter Mark	1 2 3 4 5 6 7 8 54
69	Who drives a car?	drives ("who?" X, _ car ( & a))	Thomas John My friend Sean Catherine Catherine Peter Mark	3 4 5 6 7 8 54
70	Whose car does Peter drive?	drive ( (& does) ( _ Peter, _ car ("whose?" X)))	His brother's	8
71	How does Mark drive today?	drive ( (& does) ( _ Mark, _ today, "how?" X))	very quickly	54
72	When does Mark drive very quickly?	drive ( (& does) ( _ Mark, "when?" X, _ quickly ( _ very)))	today	54
73	What kind of a car does Catherine drive?	drive ( (& does) ( _ Catherine, _ car ( _ a, "what?" X ( & kind, & of))))	a red car a green car	6 7
74	What car does Catherine drive in the race?	drive ( (& does) ( _ Catherine, _ car ( "what?" X), _ race ( _ in, _ the)))	a green car	7
75	Where does Catherine drive a red car?	drive ( (& does) ( _ Catherine, _ car ( _ a, _ red), "where?" X))	on the highway	6
76	What does Jacob drive?	drive ( (& does) ( _ Jacob, "what?" X))	A truck	2



#### 4.3.2.2 Traženi odgovor je na najvišoj razini hijerarhije

Druga vrsta upitnih rečenica koje započinju wh-pitanjem su one rečenice kojima se propituje što netko ili nešto radi. Takve rečenice su obično oblika *Što ... radi?* ili *What does ... do?* U takvim je rečenicama nepoznat glagol, odnosno procesni čvor. Traženi odgovor je sam procesni čvor. Prošireni odgovor na pitanje je procesni čvor i nekoliko njemu podređenih čvorova kako bi se dobila potpuna semantika odgovora. *X* je na početku QFNOK zapisa, na najvišoj razini hijerarhije. Ispred *X*, kao ni ispred procesnog čvora, nema uloge. Svi ostali čvorovi hijerarhijski se organiziraju prema pravilima prevođenja (objašnjeno u poglavlju 4.2). Za njihove uloge stavljamo "\_" ili "&" ovisno o tome uzimamo li čvor u obzir kod traženja odgovora ili ne. Također, moguće je dobiti više odgovora na isto pitanje, ovisno o polaznom znanju i postavljenom pitanju.

##### Općeniti zapis QFNOK-a:

**X ( (& node 1), \_ node 2 ( \_ node 3), ...), ...)**

gdje su *node 1* do *node n* čvorovi, *X* je traženi odgovor, odnosno tražena pozicija odgovora.

Uloga čvora označenog sa *X* je upravo wh-pitanje kojim započinje upitna rečenica. Sve čvorove označene sa & koje ne uzimamo u obzir kod propitivanja, stavljamo unutar zagrada () i navodimo sve čvorove koje želimo isključiti. Na taj način nam QFNOK zapis sadrži sve riječi polazne rečenice bez gubljenja semantike.

Tablica 4.5 prikazuje upitne rečenice kod kojih je traženi odgovor na najvišoj razini hijerarhije, s pripadajućim QFNOK zapisima i odgovorima temeljenim na znanju iz tablice 4.3. Prikazani su odgovori na pitanje. Odgovor u kojem je samo procesni čvor je semantički ispravan odgovor. Osim njega, prikazan je i prošireni odgovor kojim bi se moglo odgovoriti na postavljeno pitanje. Tablica je ulomak iz tablice 20 i tablice 23 u Pravitku 2. Wh-pitanja *what* i *što* ne ističemo u QFNOK-u. Ove uloge su zamijenjene znakom &.

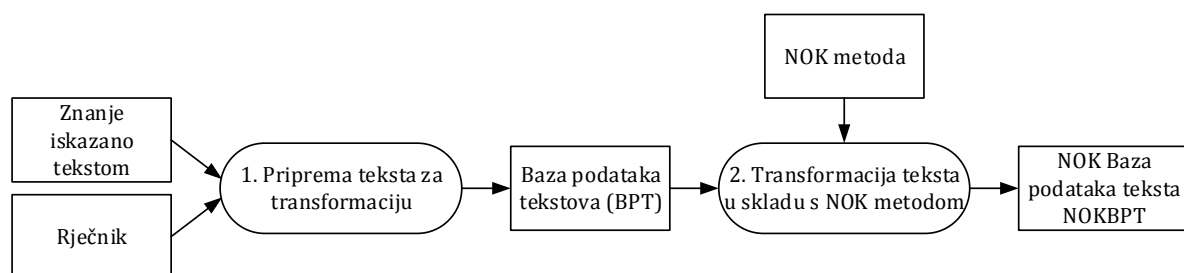
Tablica 4.5. Upitne rečenice, kod kojih je traženi odgovor na najvišoj razini hijerarhije, s pripadajućim QFNOK zapisima i odgovorima temeljenim na znanju iz tablice 4.3

RBR	Upitna rečenica	QFNOK	Odgovor	Prošireni odgovor	Rečenica u kojoj je pronađen odgovor
165	What does Peter do?	X ( (& does), (& do), _ Peter)	drives	drives his brother's car	8
165	Što Petar radi?	X ( (& radi), _ Petar)	vozi	vozi bratov auto	8

## 5 Meta modeli NOK metode

Kako bismo tekstualno znanje prebacili u relacijsku bazu podataka, potrebno je analizirati na koji je način tekstualno znanje organizirano te kako rečenice prirodnog jezika zapisati u relacijsku bazu podataka. U tu svrhu u ovom poglavlju predstavljeni su modeli podataka i modeli procesa koji opisuju sustav za transformaciju. Za prikaz modela podataka koristit će se koncepti metodologije MIRIS. MIRIS je specijalizirana metodologija za razvoj informacijskog sustava [87] koja se sastoji od skupa metoda i uputa čiji je ukupni cilj projektirati i izgraditi informacijski sustav (IS), odnosno izgraditi model informacijskog sustava. Pored elemenata metodologije MIRIS u nastavku se koriste i koncepti metode NOK [47].

Sustav za transformaciju rečenica prirodnog jezika u bazu podataka podrazumijeva dva procesa. Prvi proces predstavlja analizu teksta i pripremu teksta za transformaciju u bazu podataka tekstova. Drugi proces transformira pripremljeni tekst prema pravilima NOK metode i stvara se NOK baza podataka teksta (NOKBPT). Slika 5.1 prikazuje model procesa sustava koji omogućuje upisivanje rečenica prirodnog jezika u bazu podataka.



**Slika 5.1. Model procesa sustava transformacije teksta u bazu podataka**

Bilo kakav ulazni tekst ili knjiga (na primjer, iz .doc ili sličnih formata) može biti učitani u bazu podataka tekstova (BPT), što je uloga prvog dijela sustava. Sustav pamti polazne rečenice i njihov položaj u hijerarhijskoj strukturi teksta od rečenice preko odlomka, poglavlja do naziva teksta. Cjelokupni sustav integrira model rječnika i model hijerarhije teksta kao dodatnog izvora znanja. Integracija rječnika s modelom hijerarhije teksta prikazana je na slici 5.2. U bazi podataka riječi se izdvajaju iz rečenice i za svaku se riječ započinje proces prepoznavanja riječi, odnosno traženja koja je to riječ u rječniku. Slika 5.4 prikazuje univerzalno rješenje čuvanja tekstova neograničenog broja razina hijerarhije.

Drugi dio sustava uzima pripremljene tekstove iz BPT. Slika 5.5 prikazuje model sustava transformacije teksta u skladu s pravilima NOK metode. Rezultat transformacije je izlazna baza podataka skraćeno NOKBPT. Kako bi se riješio problem unosa znanja iskazanog tekstom u bazu podataka uz očuvanje semantike teksta, koristi se metoda NOK. U procesu analize sadržaja rječnika i tekstualnog zapisa koristi se metoda entiteta i veza (EV) za prikaz modela podataka sadržanih u rječnicima i tekstovima. Ovi modeli predstavljaju organizaciju baza podataka tekstova pripremljenih za transformaciju u relacijsku bazu podataka. Model ciljne baze podataka tekstova (NOKBPT) također je izrađen metodom entiteta i veza.

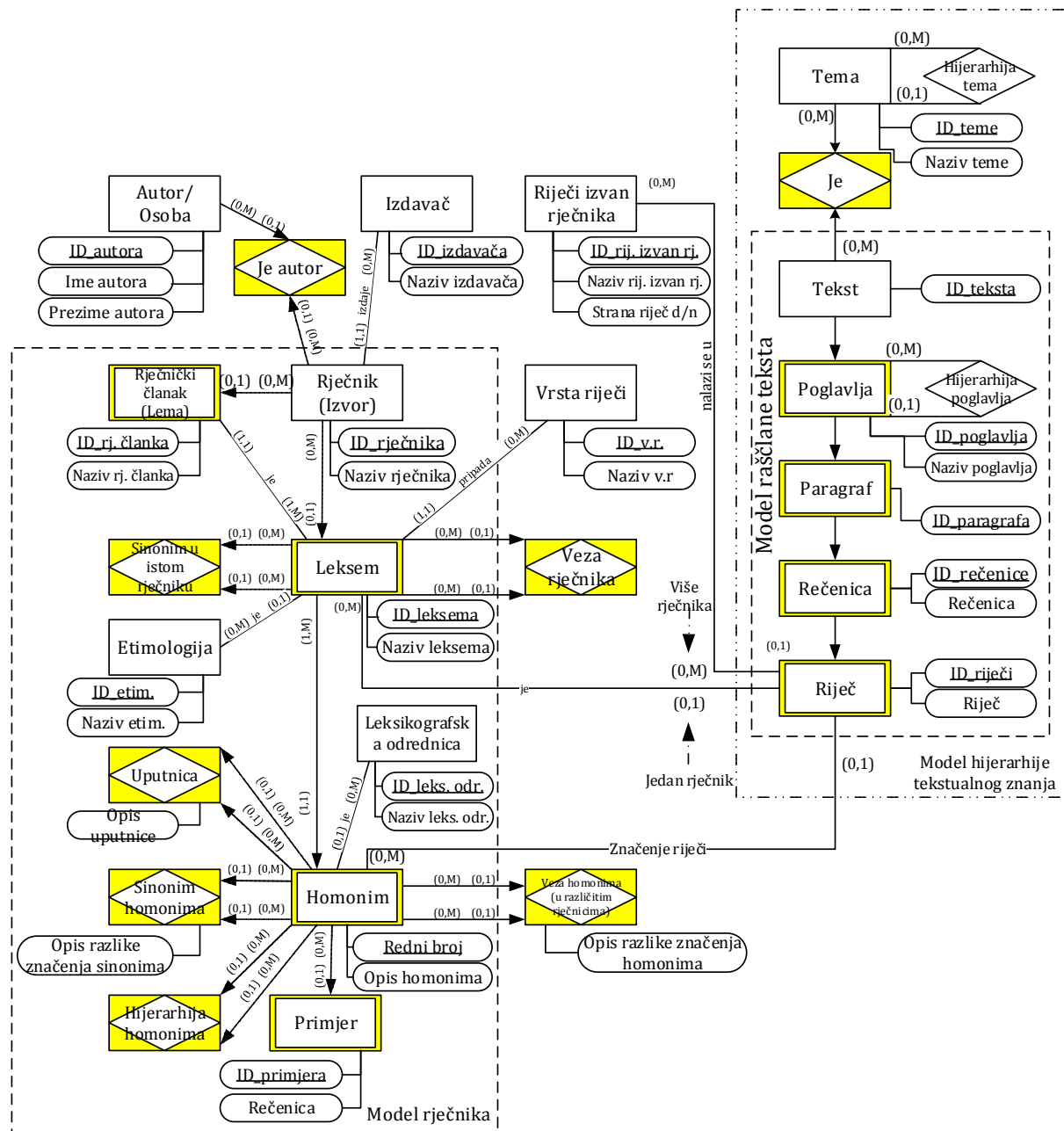
## 5.1 Priprema teksta za transformaciju

Za unos teksta (odnosno znanja iskazanog tekstom) u obliku jednostavnih rečenica prirodnog jezika koje želimo upisati u ulaznu bazu podataka teksta (BPT) potrebno je prvo pripremiti tekst za transformaciju. U ovom poglavlju opisan je model (slika 5.2) koji to omogućava. Model podataka nacrtan je metodom entiteta i veza te je dobiven dijagram EV (DEV). Pravokutnici predstavljaju tipove entiteta. Linije sa ili bez strelice predstavljaju tipove veza među tipovima entiteta. Atributi su prikazani ovalima. Hijerarhija je prikazana u rombu i povezana dvjema vezama na isti tip entiteta ili izlomljenom linijom sa strelicom na isti tip entiteta. Ovako dobiveni model može se transformirati u jednoznačnu shemu relacijske baze podataka [87].

Opišimo predloženi model detaljnije. Na desnoj strani slike 5.2 dan je analitički prikaz modela hijerarhije znanja u tekstualnom obliku. Prikazana je razrađena shema teksta, odnosno detaljnija analitika odnosa pojmova od teksta do riječi. Na lijevoj strani prikazan je dio modela podataka rječnika koji se koristi za povezivanje i opisivanje značenja riječi u tekstu.

Tip entiteta *Tema* ima dva atributa *ID\_teme* i *Naziv teme*. Tekstualno znanje, odnosno znanje iskazano rečenicama prirodnog jezika, može se klasificirati u više tema koje su hijerarhijski organizirane, odnosno teme su strukturirane na više razina. Jedna tema može imati više podtema, dok jedna podtema (na bilo kojoj razini hijerarhije) ima jednu nadređenu temu.

Tekstove možemo opisati kao mjesto gdje leži znanje (izvor znanja). To mogu biti knjige, udžbenici, priručnici, novinski članci, web stranice i drugo. Tip entiteta *Tekst* ima samo atribut *ID\_teksta*. Atribut *Naziv\_teksta* je izostavljen zato što je naziv teksta prikazan modelom kao rečenica u prvom paragrafu prvog poglavlja čija je vrijednost atributa *Naziv\_poglavlja* = *Naziv\_teksta*.



Slika 5.2. Model hijerarhije tekstualnog znanja povezan s modelom rječnika

Tekstovi su klasificirani po temama. Pod jednom temom postoji niti jedan ili mnogo tekstova, a jedan tekst se može svrstati pod nijednu ili mnogo tema. Zbog toga nastaje agregirani tip entiteta (agregacija) *Je*. Primjerice, neke od tema su Informacijski sustavi,

Programiranje, Umjetna inteligencija, Diskretna matematika i tako dalje. Pod temom Umjetna inteligencija razlikujemo mnoge izvore znanja, primjerice knjige, članke i web stranice, a navest ćemo samo neke od njih iz referenci ovog doktorskog rada pod [18], [13], [74], [39], [106] i tako dalje.

Svaki od ovih tekstova organiziran je po poglavljima koja su organizirana hijerarhijski. Jedno poglavlje sadrži više podpoglavlja, dok jedno podpoglavlje pripada niti jednom ili najviše jednom nadređenom poglavlju, ovisno na kojoj razini hijerarhije se nalazimo (na najvišoj razini hijerarhije niti jednom). Slika 5.3 prikazuje primjer hijerarhije poglavlja u tekstu, odnosno u knjizi.

<b>Introduction 1</b>
1.1 What Is AI?
1.2 The Foundations of Artificial Intelligence
1.3 The History of Artificial Intelligence
1.4 The State of the Art
1.5 Summary, Bibliographical and Historical Notes, Exercises

Slika 5.3. Prikaz poglavlja u knjizi S. Russell i P. Norvig, *Artificial Intelligence* [106]

Slabi tip entiteta *Poglavlja* koji ima dva atributa *ID\_poglavlja* i *Naziv poglavlja*, ovisan je o tipu entiteta *Tekst*. Jedno pojedinačno pojavljivanje tipa entiteta *Tekst* može sadržavati niti jedan ili mnogo pojedinačnih pojavljivanja iz slabog tipa entiteta *Poglavlja*. Obrnuto, jedno poglavlje pripada jednom i samo jednom pojedinačnom pojavljivanju tipa entiteta *Tekst*.

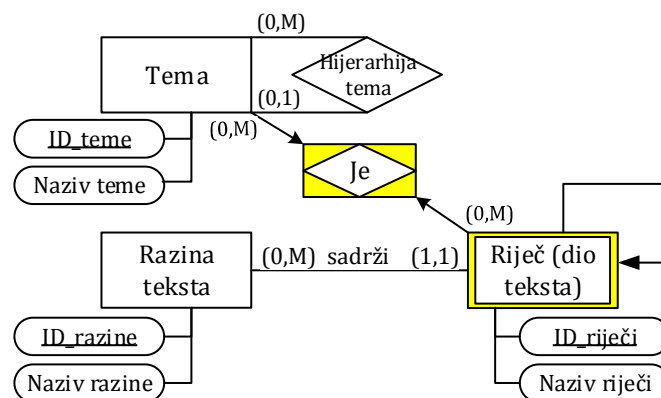
Poglavlja se dalje dijele na paragrafe, paragrafi na rečenice, a rečenice na riječi. Podpoglavlje bilo koje razine može imati paragrafe i svu nižu razinu dekompozicije teksta. Paragraf je skup rečenica poglavlja, bez naslova, odvojen u zasebnu grupu. Slabi tip entiteta *Paragraf* koji ima atribut *ID\_paragrafa*, ovisan je o slabom tipu entiteta *Poglavlja*. Jedno pojedinačno pojavljivanje slabog tipa entiteta *Poglavlja* može sadržavati niti jedan ili mnogo pojedinačnih pojavljivanja iz slabog tipa entiteta *Paragraf*. Obrnuto, jedan paragraf pripada jednom i samo jednom pojedinačnom pojavljivanju slabog tipa entiteta *Poglavlja*.

Rečenica je skup riječi u nizu koji završava točkom ili nekim drugim znakom za oznaku kraja, na primjer ".", "!", "?", ";", ":", ",". Slabi tip entiteta *Rečenica* koji ima attribute *ID\_rečenice* i *Rečenica*, ovisan je o slabom tipu entiteta *Paragraf*. Jedno pojedinačno pojavljivanje slabog tipa entiteta *Paragraf* može sadržavati niti jedan ili mnogo pojedinačnih pojavljivanja iz slabog tipa entiteta *Rečenica*. Obrnuto, jedna rečenica

pripada jednom i samo jednom pojedinačnom pojavljivanju slabog tipa entiteta *Paragraf*. U atributu *Rečenica* sadržan je tekstualni zapis polazne rečenice.

Riječ je skup znakova odvojenih razmakom. Slabi tip entiteta *Riječ* koji ima attribute *ID\_riječi* i *Riječ*, ovisan je o slabom tipu entiteta *Rečenica*. Jedno pojedinačno pojavljivanje slabog tipa entiteta *Rečenica* može sadržavati niti jedan ili mnogo pojedinačnih pojavljivanja iz slabog tipa entiteta *Riječ*. Obrnuto, jedna riječ pripada jednom i samo jednom pojedinačnom pojavljivanju slabog tipa entiteta *Riječ*. U atributu *Riječ* sadržana je jedna riječ iz polazne rečenice.

Koristeći klasifikacijsku apstrakciju, možemo više tipova entiteta različitih razina s desne strane slike 5.2 prikazati jednim meta tipom entiteta u okviru NOK metode, stoga model raščlane teksta sa slike 5.2 možemo prikazati slikom 5.4. Značenje pojma *Riječ (dio teksta)* mijenja se ovisno o razini teksta na kojoj se nalazi i ono može biti neki od pojmova sa slike 5.2 (Tekst, Poglavlje, Paragraf, Rečenica ili Riječ). Zbog toga je uveden novi tip entiteta *Razina teksta* tako da za svaku riječ znamo točnu razinu kojoj pripada. Za sliku 5.4, koja prikazuje sažeti prikaz hijerarhije znanja, možemo reći da prikazuje meta model, odnosno sintetički prikaz modela znanja, odnosno raščlane tekstualnog znanja.



Slika 5.4. DEV Raščlane teksta

Na najnižoj razini hijerarhije tekstualnog znanja nalaze se riječi bez opisa i značenja. Opis i značenje pojedine riječi čuvamo izvan teksta u rječnicima. Kako bismo shvatili značenje riječi potrebno je izgraditi model podataka rječnika i povezati ga s modelom podataka teksta. Ovdje ćemo prikazati i opisati dio rječnika koji je potreban kako bi se precizno riječ u tekstu povezala sa značenjem te riječi opisane u rječniku. Za to je potrebno analizirati strukturu opisa značenja riječi u rečenicama.

U nastavku je dan detaljniji opis rječnika s više njegovih atributa. Na lijevoj strani slike 5.2 nalazi se model rječnika napravljen analizom sadržaja postojećih rječnika s ciljem nalaženja opisa koji na najnižoj razini definira značenje riječi. *Rječnik (Izvor)*, s atributima *ID\_rječnika*, *Naziv rječnika*, sadrži popis svih mogućih (odabranih) rječnika, odnosno izvora znanja.

Tip entiteta *Izdavač* s atributima *ID\_izdavača* i *Naziv izdavača* sadrži popis izdavačkih kuća. Pojedinačno pojavljivanje tipa entiteta *Rječnik (Izvor)* izdao je jedan i samo jedan *Izdavač* iz tipa entiteta *Izdavač*. Obrnuto, jedan *izdavač* iz tipa entiteta *Izdavač* izdao je niti jedan ili mnogo rječnika (izvora) iz tipa entiteta *Rječnik (Izvor)*.

Tip entiteta *Autor/Osoba* s atributima *ID\_autora*, *Ime autora* i *Prezime autora* sadrži popis svih autora. Pojedinačno pojavljivanje tipa entiteta *Rječnik (Izvor)* može imati najmanje jednog autora, a može ih biti i više (mnogo). S druge strane, jedan autor/osoba iz tipa entiteta *Autor/Osoba* može napisati niti jedan ili mnogo rječnika (izvora) iz tipa entiteta *Rječnik (Izvor)* stoga se između ova dva tipa entiteta uvodi agregacija *Je autor*.

Slabi tip entiteta *Rječnički članak (Lema)*, ovisan o tipu entiteta *Rječnik (Izvor)*, ima attribute *ID\_rj. članka* i *Naziv rj. članka*. Svi podaci o jednoj leksičkoj jedinici (odnosno leksemu - ukupnost svih oblika i značenja neke riječi [56] nalaze se u rječničkom članku ili lemi. Struktura samog rječničkog članka razlikuje se od rječnika do rječnika. Rječnički članak minimalno mora sadržavati natuknicu (osnovnu riječ koju objašnjavamo), podatak o vrsti riječi i definiciju. Svi podaci o rječničkom članku pohranjeni su u atributu *Naziv rj. članka* i razlažu se detaljnije po tipovima entiteta u nastavku. Pojedinačno pojavljivanje tipa entiteta *Rječnik (Izvor)* sastoji se od niti jednog ili mnogo pojedinačnih pojavljivanja slabog tipa entiteta *Rječnički članak (Lema)*. Obrnuto, jedan rječnički članak (lema) iz tipa entiteta *Rječnički članak (Lema)* pripada niti jednom ili samo jednom *Rječniku (Izvoru)*.

Svaki *Rječnik (Izvor)* sastoji se od *Leksema* (natuknica, riječi, pojmova). Pojedinačno pojavljivanje tipa entiteta *Rječnik (Izvor)* sastoji se od niti jednog (u slučaju novootvorenog rječnika bez sadržaja) ili mnogo pojedinačnih pojavljivanja slabog tipa entiteta *Leksem*. Obrnuto, jedan *leksem* iz tipa entiteta *Leksem* pripada jednom i samo jednom *Rječniku (Izvoru)*. Iako rječnici sadrže velik broj istih leksema u predloženom modelu se leksemi različitih rječnika predstavljaju kako zasebni entiteti. Atributi slabog

tipa entiteta *Leksem* su *ID\_Leksema* i *Naziv leksema*. *Leksem* nema atribut za čuvanje opisa riječi, već se opis riječi čuva na detaljnijoj razini. U rječnicima dva različita leksema (različiti *ID\_Leksema*) imaju isti *Naziv leksema*, npr. "book" kao imenica i kao glagol. Svaki od tih leksema može imati više značenja i predstavlja semantički srodnu grupu homonima. Oba ta leksema su homonimi, jedan prema drugom. Osim toga oni mogu sadržavati niz odvojenih značenja pri čemu se neka od tih značenja mogu dalje grupirati i dobivaju oznake kao: 1, 2a, 2b, 3 i slično. Svi oni imaju isti naziv, ali u modelu dobivaju jedinstveni identifikator leksema.

Ovisno o nekim autorima i različitostima rječnika, postoji mogućnost da je isti *leksem* razbijen u više rječničkih članaka. Primjer za to možemo uzeti iz [4]: rok *m*; rok, v. rock; Rok, m. Drugi primjer možemo uzeti iz [28]: exact<sup>1</sup> *adj...*; exact<sup>2</sup> *v....* Zbog toga se uvodi veza između slabog tipa entiteta *Rječnički članak (Lema)* i slabog tipa entiteta *Leksem*. Jedan rječnički članak (*lema*) iz slabog tipa entiteta *Rječnički članak (Lema)* odgovara jednom i samo jednom leksemu iz slabog tipa entiteta *Leksem*. Obrnuto, jedan *leksem* iz tipa entiteta *Leksem* povezan je s jednim ili više rječničkih članaka (*lema*) iz slabog tipa entiteta *Rječnički članak (Lema)*.

Tip entiteta *Vrsta riječi*, s atributima *ID\_v.r.* i *Naziv v.r.*, sadrži popis svih vrsta riječi, na primjer, imenice, zamjenice, glagoli i slično. Popis vrsta riječi za hrvatski jezik dan je u [75]. Jedno pojedinačno pojavljivanje slabog tipa entiteta *Leksem* može pripadati jednom i samo jednom pojedinačnom pojavljivanju iz tipa entiteta *Vrsta riječi*. Obrnuto, jedno pojedinačno pojavljivanje iz *Vrste riječi* pripada niti jednom ili mnogo pojedinačnih pojavljivanja slabog tipa entiteta *Leksem*. Drugim riječima, jedan *leksem* u rječniku je samo jedne vrste riječi, dok pod nekom vrstom riječi možemo imati više leksema. Za imenice se definira rod (muški, ženski, srednji), kod glagola razlikujemo svršeni, nesvršeni i dvovidni glagolski oblik. Ostale vrste riječi označene su svojim nazivima: pridjev, prilog i slično.

Tip entiteta *Etimologija* s atributima *ID\_etim.* i *Naziv etim.* sadrži popis svih jezika te zbog veze sa slabim tipom entiteta *Leksem* omogućuje nam informaciju o podrijetlu leksema, odnosno o izvornom jeziku iz kojeg riječ dolazi. Jedno pojedinačno pojavljivanje slabog tipa entiteta *Leksem* može pripadati niti jednom i samo jednom pojedinačnom pojavljivanju iz tipa entiteta *Etimologija*. Obrnuto, jedno pojedinačno



pojavljivanje iz *Etimologija* pripada niti jednom ili mnogo pojedinačnih pojavljivanja slabog tipa entiteta *Leksem*.

Agregirani tip entiteta (agregacija) *Veza rječnika*, koja nastaje zbog (0,M) : (0,M) veze, omogućuje pregled istog leksema u različitim rječnicima, odnosno jedan leksem u prvom rječniku povezan je s odgovarajućim leksemom u drugom rječniku. Dva različita rječnika nemaju istu definiciju niti dekompoziciju leksema u homonime. Ova veza je korisna za pretraživanje istog znanja u različitim rječnicima.

Svaki leksem (riječ, pojam) ima izdvojene svoje homonime, odnosno raščlanu i opise tog homonima. Na primjer, leksem *pas* u hrvatskom jeziku znači: 1. *životinja* i 2. *pojas*; leksem *jam* u engleskom jeziku znači: 1. *A bread spread*, 2. *Difficult situation* 3. *Blockage* i tako dalje. Pojedini homonim ima barem jedan opis, a svi opisi, bez obzira na njihov broj bit će navedeni u slabom tipu entiteta *Homonim* koji ima attribute *Redni broj* i *Opis homonima*. To znači da se definicije natuknice, odnosno tekstovi iz rječnika koji opisuju riječi nalaze u atributu *Opis homonima*.

Tip entiteta *Leksikografska odrednica* s atributima *ID\_leks. odr.* i *Naziv leks. odr.* sadrži popis područja u kojem pojedine riječi imaju specifično značenje. Veza između tipa entiteta *Leksikografska odrednica* i slabog tipa entiteta *Homonim* omogućuje nam informaciju da se homonim upotrebljava u okvirima jedne ili više struka, područja ili djelatnosti. Jedno pojedinačno pojavljivanje slabog tipa entiteta *Homonim* može pripadati niti jednom i samo jednom pojedinačnom pojavljivanju iz tipa entiteta *Leksikografska odrednica*. Obrnuto, jedno pojedinačno pojavljivanje iz tipa entiteta *Leksikografska odrednica* pripada niti jednom ili mnogo pojedinačnih pojavljivanja slabog tipa entiteta *Homonim*.

Aggregacija *Uputnica* definira vezu između više homonima. Opis uputnice može biti: vidi, usporedi, jednako, oprečno.

Slabi tip entiteta *Primjer*, ovisan o slabom tipu entiteta *Homonim* omogućuje pohranu jednog ili više primjera ili cijele rečenice u kojoj se nalazi taj homonim. Primjer pomaže boljem shvaćanju značenja riječi unutar određenog konteksta.

Aggregacija *Veza homonima (u različitim rječnicima)* omogućuje povezivanje istog opisa nekog leksema (odnosno homonima) u dva različita rječnika. Ova veza precizno definira koji pojam u jednom rječniku odgovara kojem pojmu u drugom rječniku. Ako dva pojma

ne odgovaraju potpuno već među njima postoji semantička razlika, ona se može dodati u atribut *Opis razlike značenja homonima* u agregaciji *Veza homonima*.

Agregacija *Sinonim u istom rječniku*, omogućuje pretraživanje sinonima u istom rječniku, odnosno omogućuje povezivanje dvaju leksema u istom rječniku koji su sinonimi (na primjer, ljekarna – apoteka, sustav – sistem, begin – start, yearly – annually). Ako neki leksem ima više opisa, javlja se problem povezivanja tog leksema s njegovim sinonimom jer ne znamo za koji opis je taj leksem sinonim drugom leksemu. Taj problem je riješen uvođenjem agregacije *Sinonim homonima* koji omogućuje povezivanje dvaju opisa nekog homonima u istom rječniku.

Agregacijom *Sinonim homonima* obogatili smo rječnike novim znanjima. Kao i kod veze među homonimima, veza među sinonimima može povezivati dva sinonima koji imaju potpuno isto značenje. Ali moguće je da postoji i djelomično preklapanje značenja te je opis razlike značenja sadržan u *Opis razlike značenja sinonima*.

Agregacija *Hijerarhija homonima* omogućuje hijerarhiju između više homonima.

Tip entiteta *Riječi izvan rječnika* s atributima *ID\_rij. izvan rj.*, *Naziv\_rij. izvan rj.* i *Strana riječ d/n* sadrži sve riječi koje su se pojavile u nekom tekstu, a ne možemo ih povezati s *Leksemom*. Na primjer, strana riječ (primjerice na njemačkom) ili neki drugi oblik riječi primjerice hoda / hodati ili walk / walks (lematizirani rječnici). Tumačenje dodajemo ručno. U rječnik ne dodajemo morfološke oblike.

Ovaj model rječnika obuhvatio je isključivo elemente rječnika koji čuvaju značenje pojmova, bez modeliranja gramatike prirodnog jezika i niza atributa sadržanih u rječnicima. Model nam omogućuje pregled popisa riječi i njihove opise kroz različite rječnike.

Objasnimo model rječnika na dva primjera, leksemima "book" i "land" iz različitih rječnika. Ako potražimo "book" u Cambridge Dictionaries Online [16] možemo zaključiti da su u ovom rječniku za "book" kao imenicu (engl. noun) definirane tri osnovne definicije (3 osnovna homonima od kojih neki imaju dodatne homonime), a za "book" kao glagol (engl. verb) definirane dvije definicije (2 homonima). "Book" kao noun i "book" kao verb na modelu i u bazi podataka pojavljuju se kao dva različita leksema (prvi puta kao leksem "book" gdje je vrsta riječi imenica, a drugi puta kao leksem "book" gdje je vrsta riječi glagol). Oba leksema imaju svoj jedinstveni *ID\_leksema*.

U Oxford Dictionaries [82] također razlikujemo "book" u ovisnosti o vrsti riječi (imenica ili glagol) te za svaki od ovih leksema imamo nekoliko homonima od kojih su neki međusobno grupirani.

U Cambridge Dictionaries Online [17] *land* je opisano:

**noun uk /lənd/ us**

1. the surface of the Earth that is not covered by water
2. an area of ground, especially when used for a particular purpose such as farming or building
3. farms, farming, and the countryside

U Rječniku hrvatskoga jezika Vladimira Anića [4], *zemlja* ima sljedeća značenja:

**zèmlja ž N mn zemlje, G zemáljā**

- 1. a. (Zemlja) planet na kojem živimo, treći unutarnji planet Sunčeva sustava s jednim prirodnim satelitom (Mjesec) b. mjesto, prostor života i ljudske djelatnosti; svijet
- 2. površina Zemlje, kopno, suho, opr. voda
- 3. površina tla, gornji sloj Zemljine kore [nad zemljom i pod zemljom]
- 4. a. zemljište, tlo kao izvor dobara i hrane b. (i u mn) parcela, prostor koji se obrađuje, iskorištava; zemljište kao imovina, vlasništvo [privatna zemlja; zadružna zemlja; državna zemlja]
- 5. tip tla koje se obrađuje, na kojem se gradi itd. [pjeskovita zemlja; raskvašena zemlja]
- 6. pov. državno-upravna jedinica u smislu državne, administrativne i političke podjele Habsburške Monarhije i Austro-Ugarske Monarhije

Iz ovog primjera vidljivo je da različiti rječnici imaju različit broj i definicije homonima. Na slici 5.2 također je prikazano kako su međusobno povezani model rječnika i model hijerarhije tekstualnog znanja, odnosno model raščlane teksta. Jedan *Rječnik (Izvor)* sastoji se od Leksema (riječi, pojmova) koje su ovisne o rječniku (izvoru). Svaki leksem ima svoje homonime. Slabi tip entiteta *Leksem* u modelu rječnika povezan je sa slabim tipom entiteta *Riječ* u modelu hijerarhije tekstualnog znanja. Brojnosti te veze razlikuju se promatramo li ga sa stajališta jednog ili više rječnika. Radi li se o promatranju više rječnika (postojanje više homonima), imamo vezu  $(0,M) : (0,M)$  pa ta veza postaje agregacija. Jedna riječ u nekoj rečenici nekog teksta, može odgovarati više riječi/leksema u rječnicima (izvorima). Jedna riječ/leksem u jednom rječniku (izvoru) može odgovarati više riječi u rečenicama neke knjige.

Kod promatranja samo kroz jedan rječnik imamo vezu  $(0,M) : (0,1)$ . Jedna riječ u nekom tekstu opisana je jednom ili nijednom u nekom rječniku. Prednost korištenja jednog rječnika je ta što jedna riječ u tekstu ima samo jedno značenja i povezana je s jednim

homonimom. Ostale veze s drugim rječnicima ostvaruju se kroz navedene agregacije među rječnicima. Zato predlažemo ograničenje na model i korištenje jednog referentnog rječnika. *Riječ* u rečenici i *Leksem* u rječniku mogu se povezati automatski kroz program, uz uvjet da za riječ u rečenici znamo koje je vrste riječi. Ta veza nije dovoljno dobra jer znati o kojem se leksemu radi ne znači znati o kojem se značenju homonima radi u rečenici. Zato je potrebna detaljnija veza u modelu.

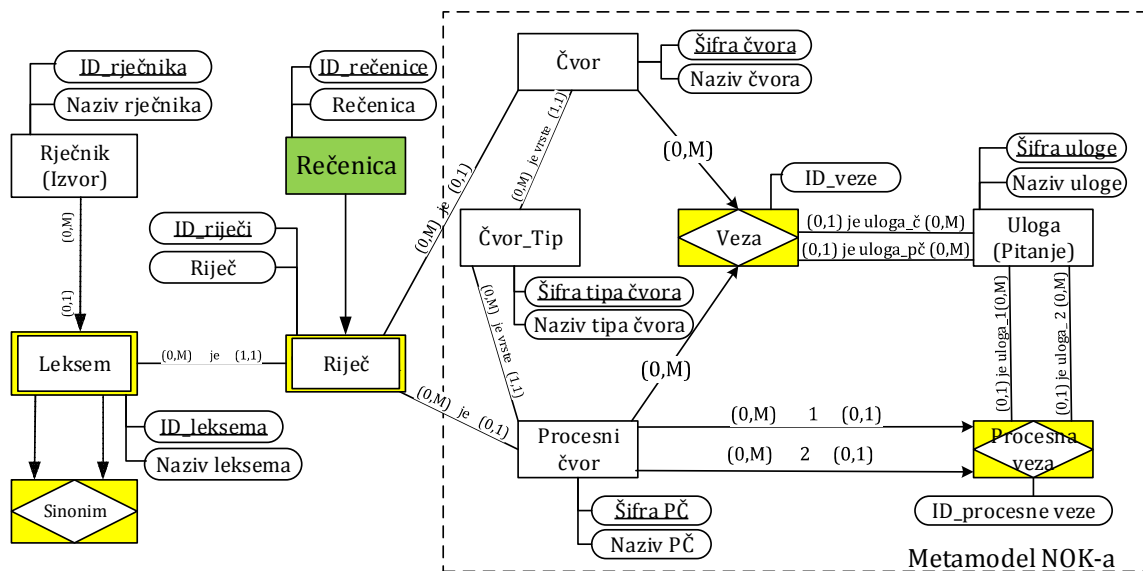
Pomoću veze *Značenje riječi* slabi tip entiteta *Riječ* iz modela raščlane teksta (koji je bez značenja) povezan je sa svojim opisom u dijelu modela rječnika sa slabim tipom entiteta *Homonim*. Svaka riječ može biti povezana s niti jednim homonimom ako je to neka nova riječ koje još nema u rječniku, ili može biti povezana samo s najviše jednim homonimom koji opisuje njeno značenje.

Time smo riječi bez značenja pridružili njeno pravo značenje kroz opis u rječniku (Homonim). Ovdje se pojavljuje problem zbog više homonima jednog leksema za istu vrstu riječi. Ako riječ ima više homonima, potrebno je pogledati kontekst u kojem se rečenica nalazi i pronaći odgovarajući kontekst za homonim. Ovo je zaseban problem koji se u ovom radu ne rješava. Pitanje je do koje mjere treba nadograditi predloženi model podataka rječnika kako bi se riješio ovaj problem. Istaknimo, kada ljudi pišu rečenicu, oni koriste riječi koji imaju točno određeno značenje, uglavnom jednog homonima. Ali kada druga osoba čita tekst, ona može zbog njenog manjeg kontekstualnog predznanja krivo razumjeti ili ne razumjeti tekst. Ovakva veza bi osigurala jedinstveno tumačenje znanja sadržanog u riječi. Postoji i namjera pisca (govornika) da koristi homonime koji imaju više značenja (primjerice u šalama), kako bi izazvao smijeh. Takav tekst i dalje ima riječi povezane samo s jednim homonimom i to onim koji nije provokativan, ali ljudski um nalazi druga značenja, koja mogu izazvati željenu emociju (smijeh, ljutnju).

Tekstovima sadržanim u ovako dobivenoj bazi podataka, koji imaju precizno definirano značenje (prikazano na DEV na slici 5.2), može se proizvoljno pristupati i transformirati ih u NOKBPT.

## 5.2 Transformacija teksta u skladu s NOK metodom

Drugi proces sustava odnosi se na transformaciju teksta primjenom NOK metode. Za potrebe tog procesa bilo je potrebno definirati model izlazne baze podataka, takozvanu NOK bazu podataka teksta (NOKBPT) u koju će se, nakon primjene NOK metode, ulazne rečenice transformirati i zapisivati. U tu svrhu definiran je meta model NOK metode prikazan na slici 5.5. NOK metoda omogućuje zapis rečenica prirodnog jezika u grafičkom obliku (DNOK-om). Analizirajući pravila NOK metode, možemo prikazati model NOK metode (meta model) koristeći metodu entiteta i veza. Meta model se odnosi na dvosmjernu NOK metodu, odnosno na vezu s dvije uloge (pitanja), opisanu u poglavlju 3.2.1.



Slika 5.5. Meta model NOK-a (model NOKBPT) povezan s dijelom modela raščlane teksta

Opišimo predloženi model detaljnije. Tip entiteta *Čvor\_Tip* s atributima *Šifra tipa čvora* i *Naziv tipa čvora* sadrži popis svih mogućih vrsta čvorova. Pojedinačno pojavljivanje tipa entiteta *Čvor\_Tip* može imati nijedno ili više pojavljivanja tipova entiteta *Čvor* i *Procesni čvor*. Pojedinačno pojavljivanje tipova entiteta *Čvor* i *Procesni čvor* pripada jednom i samo jednom entitetu *Čvor\_Tip*. Tipovi entiteta *Čvor* i *Procesni čvor* s atributima *Šifra čvora* i *Naziv čvora* te *Šifra PČ* i *Naziv PČ*, sadrže popis svih čvorova u NOK-u. Ova verzija NOK metode sastoji se od dvije vrste čvorova, običnih i procesnih.

U NOK-u među čvorovima postoje veze. Pojedinačno pojavljivanje tipa entiteta *Čvor* može biti u vezi s nijednim ili mnogo pojavljivanja tipa entiteta *Procesni čvor*. Pojedinačno pojavljivanje tipa entiteta *Procesni čvor* može biti u vezi s nijednim ili

mnogo pojavljivanja tipa entiteta Čvor. Zbog toga se za vezu među čvorovima uvodi agregirani tip entiteta *Veza* koji se sastoji od čvora i procesnog čvora. Svaka veza ima ulogu veze prema čvoru. Uloga (pitanje) omogućuje jasniji opis znanja. To su obično upitne zamjenice i sve upitne riječi koje je moguće postaviti. Svaka veza ima dvije uloge (pitanja).

Tip entiteta *Uloga (Pitanje)* s atributima *Šifra uloge*, *Naziv uloge* povezan je dvjema vezama s agregiranim tipom entiteta *Veza*. Pojedinačno pojavljivanje tipa entiteta *Uloga (pitanje)* može biti uloga u nijednom ili više pojavljivanja agregiranog tipa entiteta *Veza*. Pojedinačno pojavljivanje agregiranog tipa entiteta *Veza* ima dva kraja (na primjer, s jednim je povezan Čvor (*Je uloga\_č*), a s drugim Procesni čvor (*Je uloga\_pč*). Brojnosti uloga su (0,1) što znači da neka uloga ne mora imati naziv iz šifarnika.

U slučaju potrebe za povezivanjem dvaju tipova entiteta *Procesni čvor* dolazimo do agregiranog tipa entiteta *Procesna veza*. Isto kao i kod agregiranog tipa entiteta *Veza*, agregirani tip entiteta *Procesna veza* povezan je s tipom entiteta *Uloga (pitanje)* dvjema vezama. Pojedinačno pojavljivanje tipa entiteta *Uloga (pitanje)* može imati nijedno ili više pojavljivanja agregiranog tipa entiteta *Procesna veza*. Pojedinačno pojavljivanje agregiranog tipa entiteta *Procesna veza* pripada nijednom ili samo jednom entitetu *Uloga (pitanje)*.

Slika 5.5 prikazuje meta model NOK-a povezanog s dijelom modela hijerarhije tekstualnog znanja, a samim time i s modelom rječnika. Veza između ova dva modela (slika 5.2 i slika 5.5) dobiva se vezom između tipova entiteta *Čvor* ili *Procesni čvor* (s jedne strane) te slabog tipa entiteta *Riječ* ovisnog o tipu entiteta *Rečenica* s modela hijerarhije znanja (s druge strane).

Jedno pojavljivanje tipa entiteta *Čvor* može biti povezano s niti jednim ili samo jednim pojavljivanjem slabog tipa entiteta *Riječ*. Jedno pojavljivanje tipa entiteta *Riječ* može biti povezano niti s jednim ili više pojavljivanja tipa entiteta *Čvor*. Drugim riječima, jedan čvor bit će povezan s jednom ili niti jednom riječi iz teksta ili rječnika, dok za neku riječ iz teksta ili rječnika može postojati nekoliko čvorova s kojima su oni povezani.

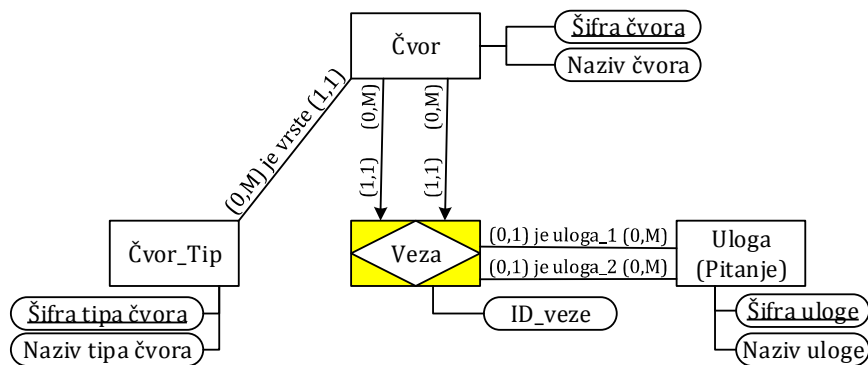
Ista situacija je s povezivanjem tipa entiteta *Procesni čvor* i slabog tipa entiteta *Riječ*.

Meta model NOK metode omogućuje transformaciju i zapis rečenica prirodnog jezika u relacijsku bazu podataka koristeći NOK metodu. Takvu bazu podataka nazivamo NOK baza podataka teksta (NOKBPT).

Jednom kad smo unijeli znanje u relacijsku bazu podataka, imamo mogućnost čuvanja znanja i propitivanja tog znanja SQL upitima. Meta model omogućuje obogaćivanje rečenica s pitanjima.

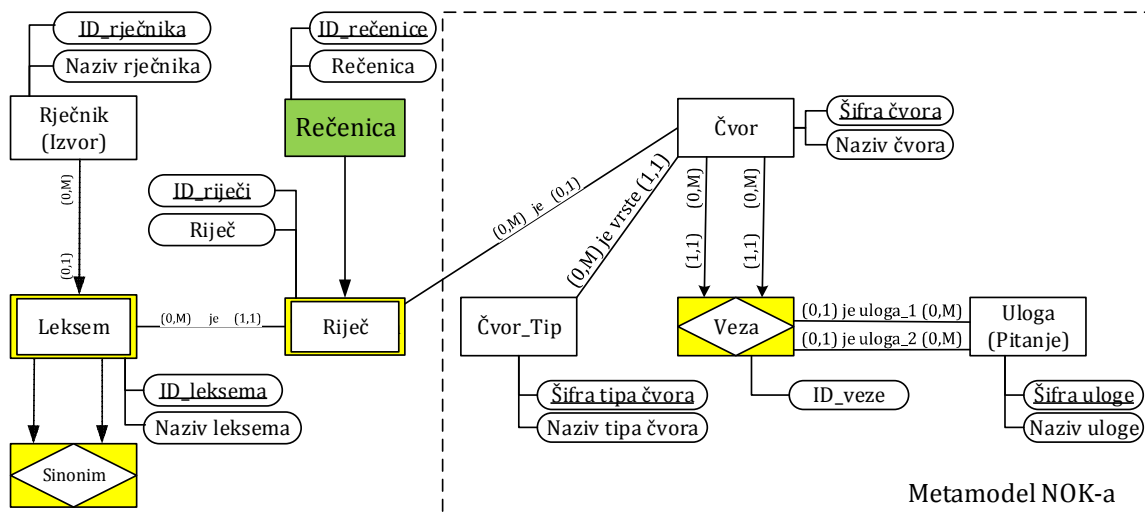
### 5.2.1 Pojednostavljanje meta modela NOK-a za dvosmjernu vezu

Zbog simetričnosti meta modela NOK-a tipove entiteta *Čvor* i *Procesni čvor* izjednačujemo i imenujemo jedinstveni tip entiteta *Čvor*. Isto napravimo s agregiranim tipovima entiteta *Veza* i *Procesna veza* te dobijemo jednu agregaciju *Veza*. Pojednostavljeni meta model prikazan je na slici 5.6.



Slika 5.6. Pojednostavljeni meta model NOK-a

Kada pojednostavljeni meta model NOK-a povežemo zajedno s dijelom modela raščlane teksta (koji je dalje povezan s modelom rječnika) dobijemo model prikazan na slici 5.7.

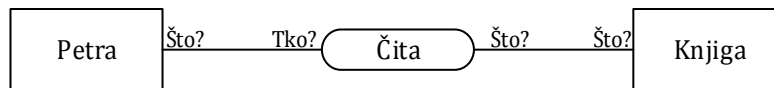


Slika 5.7. Pojednostavljeni meta model NOK-a povezan s dijelom modela raščlane teksta

Model raščlane teksta na modelu je ograničen smo samo na rečenice i riječi, a riječi su dalje povezane s leksemima u rječniku. Ostale razine hijerarhije za sada zanemarujemo.

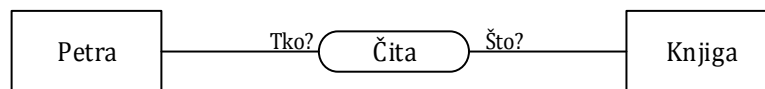
### 5.2.2 Meta model NOK-a za jednosmjernu vezu

Prikažimo rečenicu "Petra čita knjigu." DNOK-om na dva načina: pomoću dvosmjernog (dvostranog) (slika 5.8) i jednosmjernog (jednostranog) NOK-a (slika 5.9).



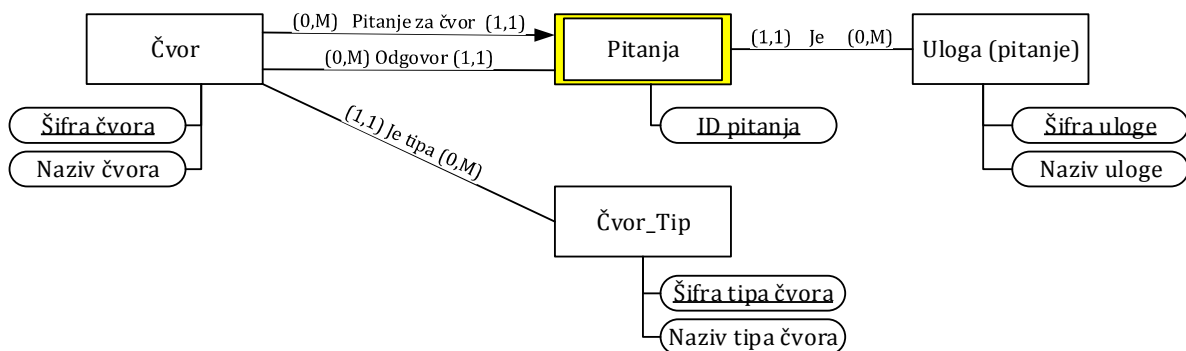
Slika 5.8. Primjer dvosmjernog NOK-a za rečenicu *Petra čita knjigu.*

Pitanje (ulogu) *Što?* sa strane čvora *Petra* i čvora *Knjiga* prema procesnom čvoru *Čita* zanemarujemo, odnosno s ove strane čvora uloga je uvijek pitanje *Što?*. To bi značilo da se druga strana uloge (pitanja) podrazumijeva te ju nećemo navoditi (*Petra Što?, Knjiga Što?*), ako znamo s kim je *Čita Tko?, Čita Što?* u vezi.



Slika 5.9. Primjer jednosmjernog NOK-a za rečenicu *Petra čita knjigu*

Nazivi čvorova (imenice i svi njeni oblici) definirani su u nominativu jednine. Nazivi procesnih čvorova (glagoli) su 3. licu jednine. Slika 5.10 prikazuje meta model NOK-a za jednosmjernu vezu. Možemo uočiti sličnosti s modelom na slici 5.6, s tim da nedostaje jedna veza za jednu ulogu, a agregacija postaje slabi tip entiteta.



Slika 5.10. Meta model NOK-a za jednosmjernu vezu

Opišimo predloženi meta model detaljnije. Tip entiteta *Čvor\_Tip* s atributima *Šifra tipa čvora* i *Naziv tipa čvora* sadrži popis svih vrsta čvorova. Pojedinačno pojavljivanje tipa entiteta *Čvor\_Tip* može imati nijedno ili više pojavljivanja tipova entiteta *Čvor*. Pojedinačno pojavljivanje tipa entiteta *Čvor* pripada jednom i samo



jednom entitetu *Čvor\_Tip*. Tip entiteta *Čvor* s atributima *Šifra čvora* i *Naziv čvora*, sadrži popis svih čvorova u NOK-u, bez obzira radi li se o običnim čvorovima ili procesnim čvorovima.

Tip entiteta *Uloga (Pitanje)* s atributima *Šifra uloge*, *Naziv uloge* sadrži popis svih upitnih zamjenica i upitnih riječi, odnosno uloga koje jedan čvor može imati u odnosu na drugi čvor.

Slabi tip entiteta *Pitanja* s atributom *ID\_pitanja* ovisan je o tipu entiteta *Čvor*. On sadrži podatke o dva međusobno povezana čvora. Odnosno, za pojedini čvor za koji se postavlja pitanje ili traži uloga, daje podatke o čvoru koji je odgovor na postavljeno pitanje te ulogu s kojom su čvorovi međusobno povezani. Pitanje je uvijek slabo od onog čvora koji postavlja pitanje za čvor. Pitanje je povezano običnom vezom s onim čvorom koji je odgovor na to pitanje.

Pretvorimo li meta model NOK-a za jednosmjernu vezu (sa slike 5.10) u relacijski model, dobivamo logičku shemu NOKBPT:

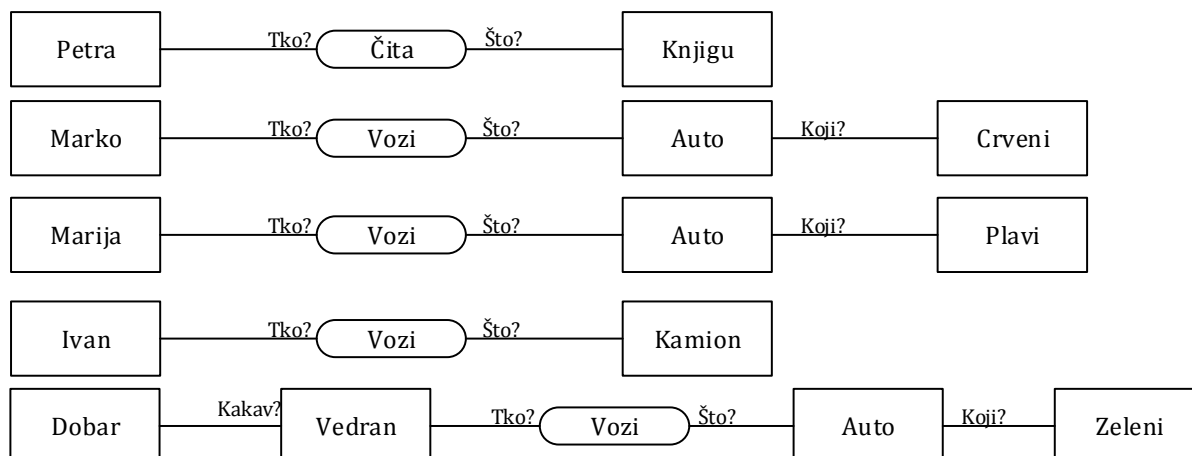
**ČVOR\_TIP** (Šifra tipa čvora, Naziv tipa čvora)  
**ČVOR** (Šifra čvora, Naziv čvora, *Šifra tipa čvora*)  
**ULOGA (PITANJE)** (Šifra uloge, Naziv uloge)  
**PITANJA** (Šifra čvora - pitanje za čvor, ID pitanja, *šifra čvora - odgovor*, *šifra uloge*)

Ove relacije prikazat ćemo tablicama i popuniti ih podacima iz rečenica prirodnog jezika za koje su napravljeni pripadni DNOK-ovi. Na primjer, rečenice prirodnog jezika i njihovi FNOK zapisi dani su u tablici 5.1.

Tablica 5.1. Rečenice prirodnog jezika i njihov FNOK zapis.

RBR	Rečenica	FNOK
1	Petra čita knjigu.	Čita ("tko?" Petra, "što?" knjigu)
2	Marko vozi crveni auto.	Vozi ("tko?" Marko, "što?" auto ("koji?" crveni))
3	Marija vozi plavi auto.	Vozi ("tko?" Marija, "što?" auto ("koji?" plavi))
4	Ivan vozi kamion.	Vozi ("tko?" Ivan, "što?" kamion)
5	Dobar Vedran vozi zeleni auto.	Vozi ("tko?" Vedran, ("kakav?" dobar), "što?" auto ("koji?" zeleni))

Pripadni DNOK-ovi za rečenice prikazani su na slici 5.11.



Slika 5.11. DNOK-ovi za rečenice prirodnog jezika

Radi lakšeg praćenja, preglednosti i jednostavnosti, u relaciju *Pitanja* dodali smo još i podatke *Naziv čvora\_pitanje za čvor*, *Naziv uloge* te *Naziv čvora\_odgovor*. Time je omogućena brža kontrola, iako je takva relacija *Pitanja* nenormalizirana i sadrži redundantne podatke. Prevođenjem logičke sheme NOKBTP u fizičku shemu NOKBPT i dodavanjem navedenih atributa dobivamo radnu shemu NOKBPT i to:

**ČVOR\_TIP** (Šifra tipa čvora, Naziv tipa čvora)  
**ČVOR** (Šifra čvora, Naziv čvora, Šifra tipa čvora)  
**ULOGA (PITANJE)** (Šifra uloge, Naziv uloge)  
**PITANJA** (ID Pitanja, Šifra uloge, Naziv uloge, Šifra čvora\_pitanje za čvor, Naziv čvora\_pitanje za čvor, Šifra čvora\_odgovor, Naziv čvora\_odgovor)

Svaki čvor koji se crta u DNOK-u, upisuje se u tablicu ČVOR.

Popunjene tablice u relacijskoj bazi podataka, na osnovi primjera iz tablice 5.1, prikazane su u tablicama 5.2 do 5.5. Svaka tablica ima u zaglavlju naveden naziv stupca i kraticu koja je uvedena zbog SQL upita.

Tablica 5.2. ČVOR

Šifra čvora	Naziv čvora	Šifra tipa čvora
S_CVR	N_CVR	S_CVR_TIP
a	Petra	O
b	Čita	P
c	Knjigu	O
d	Marko	O
e	Vozi	P
f	Crveni	O
g	Auto	O
h	Marija	O
i	Vozi	P
j	Plavi	O
k	Auto	O

l	Ivan	O
m	Vozi	P
n	Kamion	O
o	Dobar	O
p	Vedran	O
r	Vozi	P
s	Zeleni	O
t	Auto	O

Tablica 5.3. ULOGA (PITANJE)

Šifra uloge	Naziv uloge
S_ULG	N_ULG
I	Tko
II.	Što
III	Koga
IV	Kakav
V	Koji

Tablica 5.4. ČVOR\_TIP

Šifra tipa čvora	Naziv tipa čvora
S_CVR_TIP	N_CVR_TIP
O	Obični čvor
P	Procesni čvor

Tablica 5.5. PITANJA

ID Pitanja	Šifra uloge	Naziv uloge	Šifra čvora_pitanje za čvor	Naziv čvora_pitanje za čvor	Šifra čvora_odgovor	Naziv čvora_odgovor
ID_P	S_ULG	N_ULG	S_CVR_P	N_CVR_P	S_CVR_O	N_CVR_O
1.	I	<i>Tko?</i>	b	<i>Čita</i>	a	<i>Petra</i>
2.	II	<i>Što?</i>	b	<i>Čita</i>	c	<i>Knjigu</i>
3.	I	<i>Tko?</i>	e	<i>Vozi</i>	d	<i>Marko</i>
4.	II	<i>Što?</i>	e	<i>Vozi</i>	g	<i>Auto</i>
5.	V	<i>Koji?</i>	g	<i>Auto</i>	f	<i>Crveni</i>
6.	I	<i>Tko?</i>	i	<i>Vozi</i>	h	<i>Marija</i>
7.	II	<i>Što?</i>	i	<i>Vozi</i>	k	<i>Auto</i>
8.	V	<i>Koji?</i>	k	<i>Auto</i>	j	<i>Plavi</i>
9.	I	<i>Tko?</i>	m	<i>Vozi</i>	l	<i>Ivan</i>
10.	II	<i>Što?</i>	m	<i>Vozi</i>	n	<i>Kamion</i>
11.	I	<i>Tko?</i>	r	<i>Vozi</i>	p	<i>Vedran</i>
12.	II	<i>Što?</i>	r	<i>Vozi</i>	t	<i>Auto</i>
13.	V	<i>Koji?</i>	t	<i>Auto</i>	s	<i>Zeleni</i>
14.	IV	<i>Kakav?</i>	p	<i>Vedran</i>	o	<i>Dobar</i>

Sada kada su rečenice prirodnog jezika transformirane u relacijsku bazu podataka, moguće je nad njom vršiti SQL upite, odnosno propitivati uneseno znanje.

Neka moguća pitanja, odgovori i SQL upiti jesu:

1. Pitanje: Tko čita?  
Odgovor: Petra

SQL upit za 1. pitanje glasi:

```
Select N_CVR_O from PITANJA where N_CVR_P='čita' AND S_ULG=(Select S_ULG from ULOGA_PITANJE where N_ULG='Tko');
```

2. Pitanje: Tko vozi?

Odgovor: Marko, Marija, Ivan, Vedran

SQL upit za 2. pitanje glasi:

```
Select N_CVR_O from PITANJA where N_CVR_P='Vozi' AND S_ULG=(Select S_ULG from ULOGA_PITANJE where N_ULG='Tko');
```

3. Pitanje: Tko vozi auto?

Odgovor: Marko, Marija, Vedran

SQL upit za 3. pitanje glasi:

```
Select N_CVR_O from PITANJA where S_CVR_P in (Select S_CVR_P from PITANJA where N_CVR_P='Vozi' and N_CVR_O='Auto') AND S_ULG=(Select S_ULG from ULOGA_PITANJE where N_ULG='Tko');
```

4. Pitanje: Tko vozi crveni auto?

Odgovor: Marko

SQL upit za 4. pitanje glasi:

```
Select N_CVR_O from PITANJA where S_CVR_P in (Select S_CVR_P from PITANJA where N_CVR_P='Vozi' and N_CVR_O='Auto' AND S_CVR_O in (Select S_CVR_P from PITANJA where N_CVR_P='Auto' and N_CVR_O='Crveni')) AND S_ULG=(Select S_ULG from ULOGA_PITANJE where N_ULG='Tko');
```

5. Pitanje: Koji auto vozi Marija?

Odgovor: Plavi

SQL upit za 5. pitanje glasi:

```
SELECT N_CVR_O FROM PITANJA WHERE S_CVR_P = (SELECT S_CVR_O FROM PITANJA WHERE N_CVR_O='Auto' AND S_CVR_P =(SELECT S_CVR_P FROM PITANJA WHERE N_CVR_P='Vozi' AND N_CVR_O='Marija')) AND S_ULG=(SELECT S_ULG FROM ULOGA_PITANJE WHERE N_ULG='Koji');
```

U ovom poglavlju prikazan je model NOK baze podataka i sama NOK Baza podataka napunjena jednostavnim rečenicama hrvatskog jezika.

### 5.2.3 Transformacija dvosmjernog DNOK-a u NOK bazu podataka teksta (NOKBPT)

U ovom poglavlju demonstrirat ćemo kako se ulazne rečenice prirodnog engleskog jezika upisuju u relacijsku bazu podataka. Prema pravilima prevođenja dijagrama entiteta i veza (DEV) u relacijsku shemu [87], na temelju meta modela NOK metode za dvosmjernu vezu sa slike 5.5 definirana je relacijska shema baze podataka NOK metode

za čuvanje teksta (NOKBPT). U nastavku je prikazana transformacija dvosmjerne veze DNOK-a. Odabrano je šest rečenica prirodnog jezika koje su transformirane u DNOK (slika 5.12). Primjeri rečenica prirodnog jezika su navedeni u nastavku, a riječi (odnosno čvorovi) u tim rečenicama odijeljene su oznakom "|".

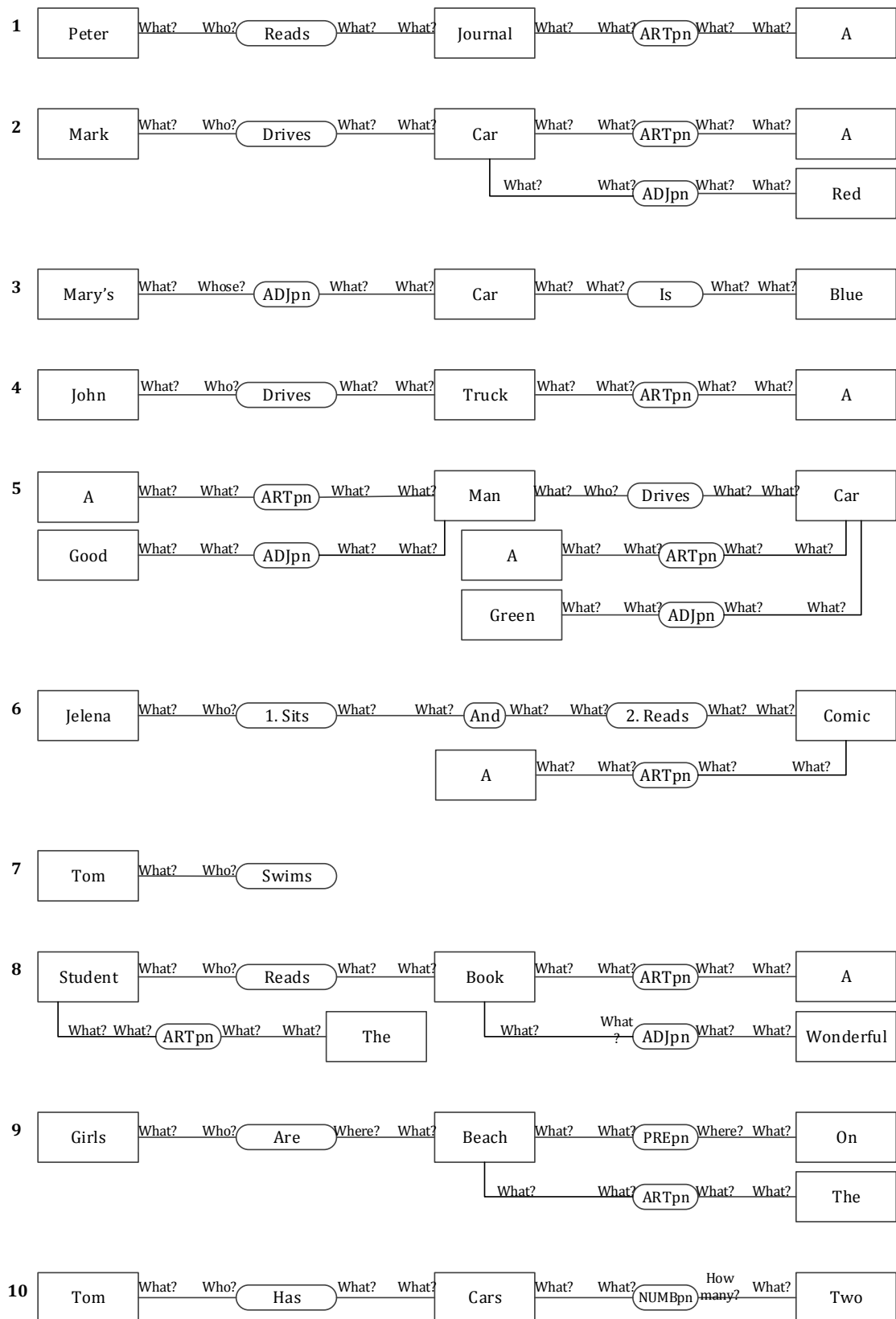
1. Peter | reads | a | journal.
2. Mark | drives | a | red | car.
3. Mary's | car | is | blue.
4. John | drives | a | truck.
5. A | good | man | drives | a | green | car.
6. Jelena | sits | and | reads | a | comic.

Osim toga odabrane su 4 rečenice prirodnog engleskog jezika od 88 iz [90] s pripadajućim FNOK zapisom (tablica 5.6). Njihov FNOK zapis je u ovim primjerima doraden i prilagođen pravilima FNOK-a definiranim u poglavlju 4. te su i te rečenice, odnosno njihovi FNOK zapisi također transformirani u DNOK. Na temelju DNOK-a popunjene su tablice u bazi.

**Tablica 5.6. Rečenice prirodnog engleskog jezika i njihov FNOK zapis**

No.	TENG	FNOK
7	Tom swims.	swims("who?" Tom)
8	The student reads a wonderful book.	reads("who?" student ("art?" the), "what?" book ("art?" a, "which?" wonderful))
9	Girls are on the beach.	are("who?" girls, "where?" beach ("where?" on, "art?" the))
10	Tom has two cars.	has("who?" Tom, "what?" cars ("how_many?" two))

Odabrani primjer teksta (sastavljen od deset rečenica prirodnog jezika) predstavlja jedno poglavlje razdijeljeno u jedan paragraf. Taj paragraf dalje dijelimo na rečenice (točnije 10 njih), a svaku rečenicu dijelimo na riječi (prema modelu hijerarhije znanja sa slike 5.2). Članove *a*, *an* i *the* u DNOK-u povezujemo s riječima (odnosno čvorovima) kojima pripadaju posebnom vrstom procesnog čvora *ART<sub>pn</sub>* (članski procesni čvor (engl. *article process node*)). Slika 5.12 prikazuje pripadne DNOK-ove za rečenice zadane prirodnim jezikom ili FNOK zapisom. U odabranim primjerima pokazane su različite vrste riječi i njihovo modeliranje NOK metodom: imenice, glagoli, pridjevi, prijedlozi, brojevi i veznici. Imenice i glagoli su dvije osnovne vrste čvorova. Pridjevi, brojevi i prijedlozi pripadaju imenicama i povezuju se posebnim procesnim čvorovima. Veznike prikazujemo kao procesne čvorove.



Slika 5.12. DNOK-ovi za rečenice prirodnog jezika i rečenice zadane FNOK-om

Procesni čvorovi *ADJpn*, *PREpn*, *NUMBpn* i *ARTpn*, odnosno pridjevski procesni čvor (*Adjectival process node*), prijedložni procesni čvor (*Prepositional process node*), brojčani

procesni čvor (*Numeric process node*) te članski procesni čvor (*Article process node*) u DNOK-u nemaju posebnu semantiku kao riječi, već predstavljaju klasu. Ubacujemo ih kako bismo povezali dva obična čvora, a osim toga služe nam i kao poveznica između čvorova koji su usko povezani i omogućuju očuvanje semantike u rečenicama. U rečenici: *6. Jelena sits and reads a comic. Sits i Reads su procesni čvorovi, s ulogama što? i tko?, kao što je prikazano na slici 5.12. Veznik And iz te rečenice na DNOK-u crtamo kao procesni čvor koji spaja dva procesna čvora.*

Relacijski model baze podataka (shema baze podataka) temeljen na meta modelu NOK-a (sa slike 5.5 bez veze na model rječnika) je:

**ČVOR\_TIP** (Šifra tipa čvora, Naziv tipa čvora)  
**ČVOR** (Šifra čvora, Naziv čvora, *Šifra tipa čvora*)  
**PROCESNI ČVOR** (Šifra PČ, Naziv PČ, *Šifra tipa čvora*)  
**ULOGA (PITANJE)** (Šifra uloge, Naziv uloge)  
**VEZA** (Šifra čvora, Šifra PČ, ID\_veze, *Šifra uloge\_č*, *Šifra uloge\_pč*)  
**PROCESNA VEZA** (Šifra PČ 1, Šifra PČ 2, ID\_procesne\_veze, *Šifra uloge\_1*, *Šifra uloge\_2*)

Koristeći DNOK-ove sa slike 5.12 možemo popuniti tablice temeljene na relacijskoj shemi baze podataka (tablice 5.7 do 5.12).

Tablica 5.7. Čvor\_tip

Šifra tipa čvora	Naziv tipa čvora
O	Regular node
P	Process node
K	Context node
ADJ	Adjectival process node
PRE	Prepositional process node
NUM	Numeric process node
ART	Article process node

Tablica 5.8. Čvor

Šifra čvora	Naziv čvora	Šifra tipa čvora
a	Peter	O
c	A	O
d	Journal	O
e	Mark	O
g	A	O
h	Red	O
i	Car	O
j	Mary's	O
k	Car	O
m	Blue	O
n	John	O
p	A	O
r	Truck	O
s	A	O

t	Good	0
u	Man	0
z	A	0
aa	Green	0
ab	Car	0
ac	Jelena	0
ag	A	0
ah	Comic	0
ai	Tom	0
ak	The	0
al	Student	0
an	A	0
ao	Wonderful	0
ap	Book	0
ar	Girls	0
at	On	0
au	The	0
av	Beach	0
az	Tom	0
bb	Two	0
bc	Cars	0

Tablica 5.9. Procesni čvor

Šifra PČ čvora	Naziv PČ	Šifra tipa čvora
b	Reads	P
b_1	ARTpn	ART
f	Drives	P
f_1	ARTpn	ART
f_2	ADJpn	ADJ
l	Is	P
l_1	ADJpn	ADJ
o	Drives	P
o_1	ARTpn	ART
v	Drives	P
v_1	ARTpn	ART
v_2	ADJpn	ADJ
v_3	ARTpn	ART
v_4	ADJpn	ADJ
ad	Sits	P
ae	And	P
af	Reads	P
af_1	ARTpn	ART
aj	Swims	P
am	Reads	P
am_1	ARTpn	ART
am_2	ARTpn	ART
am_3	ADJpn	ADJ
as	Are	P
as_1	PREpn	PRE
as_2	ARTpn	ART
ba	Has	P



ba_1	NUMBpn	NUM
------	--------	-----

Tablica 5.10. Uloga (pitanje)

Šifra uloge	Naziv uloge
I	Who
II.	What
III	Whom
IV	What kind of
V	Where
VI	How many
VII	Which
VIII	Whose

Prvih pet stupaca u tablici 5.11 prikazuje sadržaj relacijske baze podataka. Posljednja dva stupca *Naziv čvora* i *Naziv PČ*, odvojeni praznim stupcem od ostalih stupaca i precrtani, uvedeni su kako bi se olakšalo praćenje transformacije zapisa sadržaja rečenica u relacijsku bazu podataka. Isto se odnosi i na tablicu 5.12 gdje su dodani stupci *Naziv PČ\_1* i *Naziv PČ\_2*.

Tablica 5.11. Veza

ID_veze	Šifra čvora	Šifra PČ	Šifra uloge_č	Šifra uloge_pč	<i>Naziv čvora</i>	<i>Naziv PČ</i>
1.	a	b	II (What)	I. (Who)	<i>Peter</i>	<i>Reads</i>
2.	d	b	II (What)	II (What)	<i>Journal</i>	<i>Reads</i>
3.	d	b_1	II (What)	II (What)	<i>Journal</i>	<i>ARTpn</i>
4.	c	b_1	II (What)	II (What)	<i>A</i>	<i>ARTpn</i>
5.	e	f	II (What)	I. (Who)	<i>Mark</i>	<i>Drives</i>
6.	i	f	II (What)	II (What)	<i>Car</i>	<i>Drives</i>
7.	i	f_1	II (What)	II (What)	<i>Car</i>	<i>ARTpn</i>
8.	g	f_1	II (What)	II (What)	<i>A</i>	<i>ARTpn</i>
9.	i	f_2	II (What)	II (What)	<i>Car</i>	<i>ADJpn</i>
10.	h	f_2	II (What)	II (What)	<i>Red</i>	<i>ADJpn</i>
11.	j	l_1	II (What)	VIII. (Whose)	<i>Mary's</i>	<i>ADJpn</i>
12.	k	l_1	II (What)	II (What)	<i>Car</i>	<i>ADJpn</i>
13.	k	l	II (What)	II (What)	<i>Car</i>	<i>Is</i>
14.	m	l	II (What)	II (What)	<i>Blue</i>	<i>Is</i>
15.	n	o	II (What)	I. (Who)	<i>John</i>	<i>Drives</i>
16.	r	o	II (What)	II (What)	<i>Truck</i>	<i>Drives</i>
17.	r	o_1	II (What)	II (What)	<i>Truck</i>	<i>ARTpn</i>
18.	p	o_1	II (What)	II (What)	<i>A</i>	<i>ARTpn</i>
19.	s	v_1	II (What)	II (What)	<i>A</i>	<i>ARTpn</i>
20.	u	v_1	II (What)	II (What)	<i>Man</i>	<i>ARTpn</i>
21.	t	v_2	II (What)	II (What)	<i>Good</i>	<i>ADJpn</i>
22.	u	v_2	II (What)	II (What)	<i>Man</i>	<i>ADJpn</i>
23.	u	v	II (What)	I. (Who)	<i>Man</i>	<i>Drives</i>
24.	ab	v	II (What)	II (What)	<i>Car</i>	<i>Drives</i>
25.	ab	v_3	II (What)	II (What)	<i>Car</i>	<i>ARTpn</i>
26.	z	v_3	II (What)	II (What)	<i>A</i>	<i>ARTpn</i>
27.	ab	v_4	II (What)	II (What)	<i>Car</i>	<i>ADJpn</i>

28.	aa	v_4	II (What)	II (What)		<i>Green</i>	<i>ADJpn</i>
29.	ac	ad	II (What)	I. (Who)		<i>Jelena</i>	<i>Sits</i>
30.	ah	af	II (What)	II (What)		<i>Comic</i>	<i>Reads</i>
31.	ah	af_1	II (What)	II (What)		<i>Comic</i>	<i>ARTpn</i>
32.	ag	af_1	II (What)	II (What)		<i>A</i>	<i>ARTpn</i>
33.	ai	aj	II (What)	I. (Who)		<i>Tom</i>	<i>Swims</i>
34.	ak	am_1	II (What)	II (What)		<i>The</i>	<i>ARTpn</i>
35.	al	am_1	II (What)	II (What)		<i>Student</i>	<i>ARTpn</i>
36.	al	am	II (What)	I (Who)		<i>Student</i>	<i>Reads</i>
37.	ap	am	II (What)	II (What)		<i>Book</i>	<i>Reads</i>
38.	ap	am_2	II (What)	II (What)		<i>Book</i>	<i>ARTpn</i>
39.	an	am_2	II (What)	II (What)		<i>A</i>	<i>ARTpn</i>
40.	ap	am_3	II (What)	II (What)		<i>Book</i>	<i>ADJpn</i>
41.	ao	am_3	II (What)	II (What)		<i>Wonderful</i>	<i>ADJpn</i>
42.	ar	as	II (What)	I (Who)		<i>Girls</i>	<i>Are</i>
43.	av	as	II (What)	V (Where)		<i>Beach</i>	<i>Are</i>
44.	av	as_1	II (What)	II (What)		<i>Beach</i>	<i>PREpn</i>
45.	at	as_1	II (What)	V (Where)		<i>On</i>	<i>PREpn</i>
46.	av	as_2	II (What)	II (What)		<i>Beach</i>	<i>ARTpn</i>
47.	au	as_2	II (What)	II (What)		<i>The</i>	<i>ARTpn</i>
48.	az	ba	II (What)	I (Who)		<i>Tom</i>	<i>Has</i>
49.	bc	ba	II (What)	II (What)		<i>Cars</i>	<i>Has</i>
50.	bc	ba_1	II (What)	II (What)		<i>Cars</i>	<i>NUMBpn</i>
51.	az	ba_1	II (What)	VI (How many?)		<i>Two</i>	<i>NUMBpn</i>

Tablica 5.12. Procesna veza

ID_procesne _veze	Šifra PČ_1	Šifra PČ_2	Šifra uloge_1	Šifra uloge_2	Naziv PČ_1	Naziv PČ_2
1	ad	ae	II (What)	II (What)	<i>Sits</i>	<i>And</i>
2	ae	af	II (What)	II (What)	<i>And</i>	<i>Reads</i>

U tablicama 5.7 do 5.12 prikazano je kako se znanje iz jednostavnih rečenica prirodnog jezika može upisati u relacijsku bazu podataka koja će nam omogućiti propitivanje i čuvanje znanja.

Na primjer, ako pitamo *Who drives* na temelju tablica možemo dobiti odgovore: *Mark*, *John* i *Man*. Ako pitanje proširimo još jednim čvorom, dobit ćemo i precizniji odgovor. Primjerice, *Who drives a truck?* odgovor je *John*. Na pitanje *Who drives a car?* dobit ćemo odgovore: *Mark* i *Man*; dok na pitanje *Who drives a red car?* dobit ćemo odgovor: *Mark*.

Dodatno, integracija s jednojezičnim rječnikom omogućava uključivanje različitih semantičkih odnosa. Vezano za prethodni primjer, netko može pitati *Who drives an automobile?*. Informacija o automobilu nije izravno u unesena bazu podataka. Međutim, prema podacima iz jednojezičnog rječnika, postoji veza između riječi *car* i njegova sinonima *automobile*. Stoga bi odgovor na postavljeno pitanje bio *Mark* i *Man*.

Preciznost i konkretnost odgovora (jednog ili više njih) ovise o tome koliko detaljno je pitanje postavljeno te koliko čvorova u pitanju je korišteno. Ponekad će se odgovor jednostavno naći u tablicama *Veza* i *Procesna veza* jednim prolaskom po tablicama, dok će odgovor ponekad zahtijevati više prolazaka kroz tablicu. Drugim riječima, složenost SQL upita može varirati.

U nastavku će se koristiti jednosmjerni NOK (osim ako nije drugačije navedeno) te će jednosmjerni NOK biti osnova za razvoj sustava temeljenog na NOK metodi te integraciju postojećeg informacijskog sustava sa sustavom temeljenim na NOK metodi.

### 5.3 Dokaz hipoteze H1

Hipoteza H1 glasi: *Primjenom metode entiteta i veza može se projektirati meta model metode "Node of Knowledge".*

Hipotezu smatramo potvrđenom ako se metodom entiteta i veza napravi meta model metode NOK.

Modelima i meta modelima na slikama:

- *Slika 5.5. Meta model NOK-a (model NOKBPT) povezan s dijelom modela raščlane teksta* prikazuje model izlazne baze podataka, takozvanu NOK bazu podataka teksta (NOKBPT) u koju će se, nakon primjene NOK metode, ulazne rečenice transformirati i zapisivati. Meta model se odnosi na dvosmjerni NOK.
- *Slika 5.6. Pojednostavljeni meta model NOK-a i Slika 5.7. Pojednostavljeni meta model NOK-a povezan s dijelom modela raščlane teksta* također prikazuju model izlazne baze podataka, NOKBPT u koju će se, nakon primjene NOK metode, ulazne rečenice transformirati i zapisivati. Pojednostavljenje meta modela omogućila je simetričnost kod nekih tipova entiteta na slici 5.5.
- *Slika 5.10. Meta model NOK-a za jednosmjernu vezu* prikazuje meta model metode NOK koji će biti osnova za daljnji razvoj sustava temeljenog na NOK metodi.

pokazali smo da je moguće primjenom metode entiteta i veza projektirati meta model NOK metode te je time hipoteza H1 potvrđena.

## 6 Sustav temeljen na NOK metodi

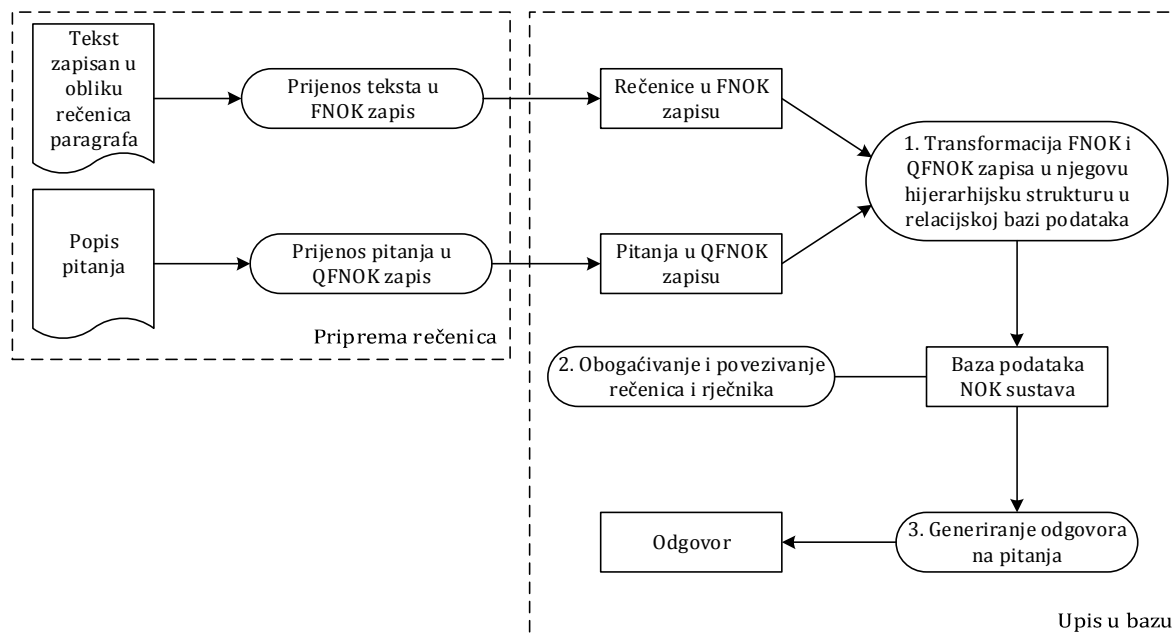
U ovom poglavlju opisano je kako koristeći NOK metodu, odnosno njen FNOK formalizam za zapis tekstualnog znanja, razviti algoritam za transformaciju FNOK zapisa u relacijsku bazu podataka. U bazu podataka tekst se ne upisuje kao cjeloviti FNOK zapisi rečenica, već se radi njihova transformacija u međusobno povezane skupove podataka. Tako dobiveni podaci upisuju se kao vrijednosti u odgovarajuće atribute, a vanjski ključevi osiguravaju vezu među n-torkama koji predstavljaju jedan FNOK zapis rečenice.

Dizajniran je model baze podataka teksta metodom entiteta i veza te je za njega prikazana shema relacijske baze podataka. Opisano je povezivanje čvorova FNOK zapisa rečenica s rječnikom. Opisana je ideja rastavljanja rečenica u skup nezavisnih uvjeta koje sadrži polazna rečenica. Predložen je sustav transformacije obogaćenih FNOK zapisa rečenica (odnosno sustav temeljen na NOK metodi) u relacijsku bazu podataka kao skup algoritama. Algoritmi su prikazani u SQL programskom jeziku.

Rezultati ovog rada bitni su za nastavak istraživanja proširenja postojećih relacijskih baza podataka tekstualnim znanjima. U nastavku je prikazan algoritam preslikavanja obogaćenih rečenica prirodnog jezika u relacijske baze podataka. Pohrana tekstova u baze podataka potrebna je jednako kao i pohrana podataka, ali je do sada bila teško ostvariva. Prednost ovog pristupa postigla se primjenom NOK metode, koja omogućuje pohranu i izvlačenje znanja. Sve je to omogućeno prethodnim formaliziranjem rečenica NOK metodom.

Prilikom razvoja sustava temeljenog na NOK metodi, koristila se NOK metoda za jednosmjernu vezu (tzv. jednosmjerna NOK metoda) za koju je i razvijen formalizirani zapis rečenica (FNOK). Jednosmjerna veza kojom su dva čvora međusobno povezana omogućuje da pronađemo odgovor na postavljeno pitanje. Drugim riječima, za jedan čvor koji postavlja pitanje, prateći vezu u modelu pronalazimo drugi čvor s kojim je prvi čvor povezan te pronalazimo odgovor na postavljeno pitanje. Suprotna veza, u ovoj verziji NOK metode se ne razmatra.

Model sustava temeljenog na NOK metodi za transformaciju FNOK zapisa rečenica prirodnog jezika i QFNOK zapisa pitanja u relacijsku bazu podataka te generiranje odgovora na pitanja prikazan je na slici 6.1.



**Slika 6.1. Model sustava temeljenog na NOK-u**

Osnovna ideja ovog pristupa je da se rečenice prirodnog jezika zapisane u FNOK zapisu i pitanja prirodnog jezika zapisana u QFNOK zapisu transformiraju u relacijsku bazu podataka. Priprema rečenica prirodnog jezika u FNOK zapis, odnosno pitanja prirodnog jezika u QFNOK zapis odvija se prema pravilima opisanim u poglavlju 4.2., odnosno 4.3.

Modeliranje sustava izvedeno je u skladu sa specijaliziranom metodologijom za razvoj informacijskog sustava (MIRIS). Model podataka relacijske baze napravljen je prema sadržaju i strukturi NOK metode. Opišimo predloženi sustav detaljnije.

Sustav za transformaciju koristi kao ulaz formalizirani zapis rečenica u FNOK-u nazvan "Rečenice u FNOK zapisu" te formalizirani zapis pitanja u QFNOK-u nazvan "Pitanja u QFNOK zapisu". Ove rečenice i pitanja nastaju primjenom procedure za prijenos teksta u FNOK, odnosno QFNOK zapis. Paragrafi tekstova zapisani u obliku rečenica postaju paragrafi FNOK rečenica, također, pitanja se grupiraju u paragrafe QFNOK zapisa.

Sustav temeljen na metodi NOK sastoji se od tri osnovna procesa. Prvi proces sustava transformira ulazne FNOK rečenice u hijerarhijsku strukturu odnosa među riječima u rečenici. Drugim se procesom transformirane rečenice i njihovi dijelovi (riječi) obogaćuju povezujući ih s rječnikom. Tako se stvara baza podataka NOK sustava. Trećim procesom se, koristeći podatke u bazi podataka NOK sustava, generira odgovor na pitanje.

## 6.1 Meta model FNOK zapisa

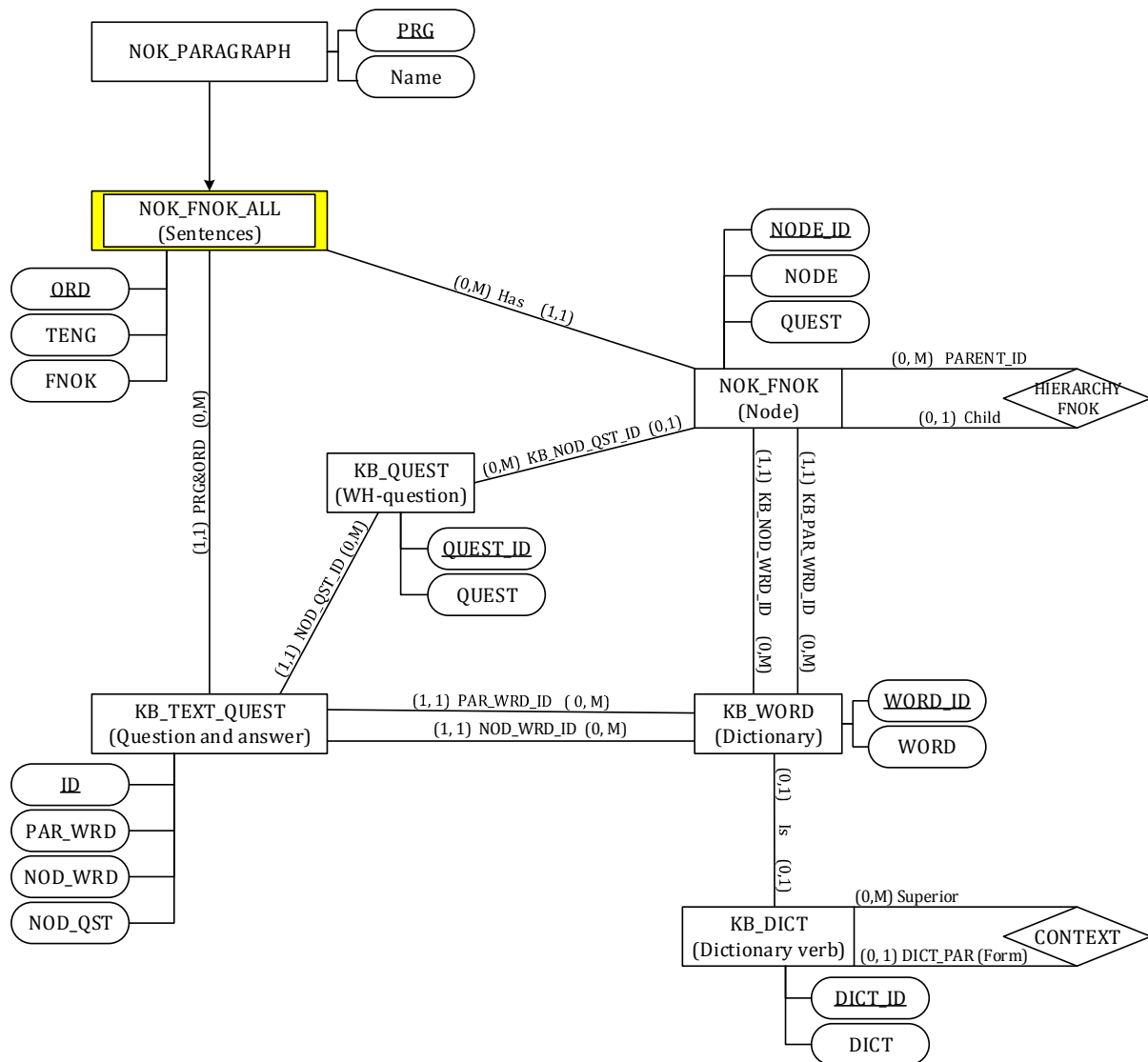
Postavlja se istraživačko pitanje "Kako FNOK zapise rečenica upisati u relacijsku bazu podataka?". Da bismo odgovorili na to pitanje, potrebno je analizirati podatkovni sadržaj FNOK zapisa i FNOK formalizam te naći odgovarajući meta model. Na osnovi meta modela podataka FNOK formalizma potrebno je definirati relacijsku shemu baze podataka koja će sadržavati FNOK zapise i algoritme transformacije početnog zapisa rečenica u povezane podatke u bazi podataka.

Za meta modeliranje podataka korištena je metoda entiteta i veza čiji je rezultat modeliranja dijagram entiteta i veza (DEV). Na osnovi DEV-a procesom prevođenja konceptata semantički bogatoga modela u relacijski model podataka definira se relacijska shema baze podataka. Na osnovi sheme baze podataka definira se baza podataka koja je dana u poglavlju 7. Na osnovi DEV-a i detaljnijom analizom procesa transformacije definiraju se potrebni algoritmi transformacije FNOK zapisa u relacijsku bazu podataka, prikazani u poglavlju 6.1.4.

FNOK formalizam omogućuje zapis rečenica prirodnog jezika u FNOK zapise obogaćene simbolima i upitnim riječima. Analizirajući pravila zapisivanja rečenica u FNOK zapis, dolazimo do modela FNOK formalizma (meta model). Meta model FNOK-a je prikazan na slici 6.2. Model podataka FNOK-a razlikuje se od modela podataka DNOK-a. Ovaj model FNOK-a u osnovi sadrži čvorove i pitanja kao i model DNOK-a. Međutim, model FNOK-a je proširen s dva dijela: dijelom za čuvanje polaznih rečenica i dijelom zajedničkog rječnika. Također, ovaj model predstavlja fizičku implementaciju u Oracle relacijsku bazu podataka sa SQL procedurama stoga je model prilagođen potrebama izgradnje programskog sustava.

Pravokutnici na meta modelu predstavljaju tipove entiteta koji će u relacijskom modelu biti prevedeni i implementirani kao tablice. Atributi su prikazani kružnim ovalima i u tablici se prikazuju kao stupci tablice. Linije sa ili bez strelice predstavljaju tipove veza među tipovima entiteta koji će predstavljati povezane tablice putem vanjskih ključeva. Zbog ovih veza među tablicama dodaju se atributi (vanjski ključevi) u shemu baze podataka te ih ima više nego na DEV-u. Vanjski ključevi su podvučeni isprekidanom linijom ili nakošeni. Ključni atributi su podvučeni punom linijom. Hijerarhija među

entitetima je prikazana simbolom romba i povezuje entitete istog tipa entiteta (nadređeni – podređeni) same sa sobom.



Slika 6.2. Meta model strukture i sadržaja FNOK zapisa

Opišimo sadržaj predloženog meta modela detaljnije. Tekstovi rečenica prirodnog jezika mogu biti grupirani u različite oblike, pri čemu se u ovom radu zadržavamo samo na grupi rečenica koje nazivamo paragrafom. Popis paragrafa sadržan je u tipu entiteta Paragraf (Paragraph). Svaki paragraf ima ključni atribut PRG i atribut kratki opis paragrafa (NAME).

Svaki paragraf ima najmanje jednu rečenicu, a može imati mnogo rečenica. Popis rečenica prikazan je slabim tipom entiteta (prikazan dvostrukim pravokutnikom) NOK\_FNOK\_ALL (Sentences). Jedna rečenica uvijek pripada jednom i samo jednom Paragrafu. Tip entiteta NOK\_FNOK\_ALL (Sentences) ima tri atributa: redni broj rečenice

(ORD), tekst rečenice na prirodnom jeziku (TENG, skraćeno od tekst na engleskom jeziku) i FNOK zapis rečenice (FNOK).

Svaki FNOK zapis NOK\_FNOK\_ALL (Sentences) sastoji se od riječi (čvorova), među kojima postoji hijerarhijski odnos. Čvorovi su prikazani tipom entiteta Čvor (oznake NOK\_FNOK (Node)). Svaki čvor pripada jednoj i samo jednoj rečenici, odnosno FNOK zapisu. Svaki čvor ima svoj identifikator (NODE\_ID) kao redni broj pojavljivanja unutar rečenice i atribut naziv čvora (NODE) što predstavlja čvor iz polaznog FNOK zapisa. Jedan zapis može imati više čvorova, odnosno niti jedan (ako još nije proveden proces transformacije zapisa u čvorove). Jedan čvor u NOK\_FNOK (Node) pripada uvijek jednoj i samo jednoj rečenici iz NOK\_FNOK\_ALL (Sentences).

Tip entiteta NOK\_FNOK (Node) nema više vlastitih atributa. Ostali atributi iz tipa entiteta NOK\_FNOK (Node) nastat će iz tipova veza s drugim tipovima entiteta u procesu transformacije DEV-a u relacijsku shemu baze podataka. Tako će se u procesu prevođenja modela podataka u relacijsku shemu ključ polazne rečenice (PRG i ORD) spustiti u svaki redak tablice iz kojeg je taj čvor nastao.

Tip entiteta NOK\_FNOK (Node) povezan je sam sa sobom povratnim tipom veze HIERARCHY FNOK prikazane rombom. FNOK zapis ima hijerarhijsku strukturu. Na početku zapisa se nalazi glagol (procesni čvor) koji je nadređen svim ostalim čvorovima. Unutar FNOK zapisa za svaki pojedini čvor koji ima hijerarhijski podređenih čvorova koristi se okrugla zagrada i u njoj popisuju svi njeni podređeni čvorovi. Na taj način moguće je postići neograničen broj razina hijerarhije. Za svaki čvor vrijedi da može imati jedan nadređen čvor (PARENT\_ID) i da može imati više podređenih čvorova (Child). Stoga povratna veza HIERARCHY FNOK ima dvije uloge i to: Parent i Child. Jedan čvor Parent može imati više podređenih čvorova (na primjer, prvi čvor) ili najmanje niti jedan podređeni (ako je zadnji bez raščlane, na primjer, pridjev ili prilog). Jedan čvor Child pripada najviše jednom roditelju i najmanje nijednom roditelju (u slučaju prvog čvora nema roditelja).

Svim čvorovima u FNOK zapisu, osim prvom procesnom čvoru u rečenici, pripada jedna upitna riječ (uloga, pitanje). Na primjer, čvoru *Marko* pripada pitanje "tko?" u rečenici *Marko vozi auto.*, odnosno FNOK zapisu *vozi ("tko?" Marko, "što?" auto)*. Stoga se u modelu dodaje atribut za "naziv postavljenog pitanja" (QUEST), kao atribut tipa entiteta



NOK\_FNOK (Node). Samo prvi čvor u rečenici ima za vrijednost ovog atributa "null vrijednost", što je označeno u stupcu QUEST simbolom podcrte "\_". Upitna riječ (QUEST) predstavlja ugrađeno pitanje dodano prilikom obogaćivanja.

### **6.1.1 Povezivanje čvorova FNOK zapisa s rječnikom**

Jedan od važnih koraka strojnog razumijevanja teksta je povezivanje riječi u rečenici s riječima u rječniku. Svaka riječ u rječniku ima identifikator na razini jezika i jedinstveno značenje. Pridruživanje identifikatora iz rječnika čvoru u FNOK zapisu važno je za dodjeljivanje značenja pojedinih čvorova. Implementacija rječnika moguća je na više načina. Jedan od načina je korištenje gotovih rječnika prirodnog jezika prilagođenih računalnoj obradi podataka. Jedan od takvih rječnika za engleski jezik je morfosintaktički leksikon engleskoga jezika MULTEXT-East [75], korišten u definiranju prevođenja rečenica engleskog jezika u FNOK zapis [47]. Klasični rječnici bogati su nizom atributa (svojstava). Za potrebe ovog istraživanja dovoljna nam je sama riječ iz rječnika bez dodatnih svojstava. U ovom istraživanju odlučili smo se za razvoj vlastitog rječnika.

Model sustava je adaptibilan te nam to omogućuje automatsko širenje rječnika riječima kojih nema u rječniku. Moguće je da, ma kako velik polazni skup riječi sadrži odabrani rječnik, naiđemo na rečenicu koja sadrži novu riječ, novi pojam, novo ime osoba ili drugih stvari. Zato je potrebno omogućiti širenje rječnika novim riječima.

Probleme rječnika prirodnoga jezika kao što su homonimi zanemarujemo. Ukoliko se pojavi taj problem on će biti razriješen ručno od strane korisnika. Ako se pojavi homonimna riječ (na primjer, auto - automobil) u FNOK zapisu koja u rječniku ima više različitih pojavljivanja i za svako značenje poseban identifikator, problem povezivanja razriješit će se ručno. Dio problema moguće je riješiti prema vrsti riječi ako su homonimi različite vrste riječi. Ako nisu, onda je potrebna kontekstna analiza što izlazi iz okvira ovoga rada.

Za rješavanje problema čuvanja podataka iz rječnika, predložena su tri tipa entiteta (tablice). Moguća su i općenitija rješenja u slučaju proširenja sustava novim zahtjevima i novim znanstvenim pitanjima kao što su: navedeni problemi homonima, sinonimi,

povezivanje dvije i više rečenica, odnos apstraktnih i konkretnih pojmova, čuvanje pitanja i odgovora na pitanje.

Rječnik (Dictionary) se sastoji od tri tipa entiteta i to: KB\_QUEST (Wh-question), KB\_WORD (Dictionary) i KB\_DICT (Dictionary verb).

Prvi tip entiteta je jednostavna tablica popisa pitanja wh-pitanja (KB\_QUEST) kojima se obogaćuje FNOK zapis rečenica. Pod wh-pitanjima podrazumijevamo onu vrstu pitanja koja koristi wh-riječi (primjerice, u engleskom jeziku su to *who, whom, whose, what, which, why, when, where, how* i tako dalje, u hrvatskom su to *tko, što, gdje, kada* i slično). Tablica KB\_QUEST ima dva atributa, identifikator (QUEST\_ID) i samu riječ pitanja (QUEST). Tablica ima mali broj *n*-torki, (nekoliko desetina pojavljivanja) i ne očekuje se njen veliki porast.

Drugi je tip entiteta Rječnik riječi (KB\_WORD (Dictionary)) i predstavlja adaptabilni rječnik svih riječi koje se pojavljuju u tekstu. Njegovi atributi jesu: identifikator riječi (WORD\_ID) i sama riječ (WORD). Alternativno je moguće ručno dodati opis značenja riječi, vrstu riječi i druge attribute, odnosno povući te podatke zajedno s riječima iz postojećih rječnika. Za potrebe ovog istraživanja to nije bilo potrebno.

Treći tip entiteta koji pripada rječniku je Rječnik glagola (KB\_DICT (Dictionary verb)). Ovaj tip entiteta za sada sadrži samo glagole i imenice. KB\_DICT (Dictionary verb) je povezan povratnom vezom CONTEXT sam sa sobom. Za razliku od hijerarhije međusobnih dijelova rečenice u tipu veze HIERARCHY FNOK, ova povratna veza modelira hijerarhiju kontekstnog odnosa semantički bogatije riječi i pojedinačnih podpojmovia koje ta riječ obuhvaća. Ako uzmemo apstraktan glagol *be* on ima više konkretnih pojavljivanja (*am, are, is, was, were, been, being*) ovisno o licu i vremenu koje imenujemo ulogom Form. Model je općenit za sve glagole i ostale promjenjive vrste riječi. Moguće je isti model kontekstualnog povezivanja riječi u rječniku primijeniti i na druge vrste promjenjivih riječi. Kod imenica, za primjerice imenicu *auto* imamo više pojavljivanja *auta, aute, auti* i slično, odnosno padeže imenice te brojnosti (jednina i množina). Predloženi model omogućuje više razina hijerarhije. Jedna riječ kao apstraktno nadređena riječ može imati najmanje nijednu do najviše mnogo konkretnih formi. Jedna konkretna riječ može biti forma najmanje ni jedne riječi (najviše u hijerarhiji) ili najviše samo jedne apstraktne riječi.

KB\_DICT (Dictionary verb) i KB\_WORD (Dictionary) su povezani tipom veze *Is*, tako da po gornjim granicama jedna riječ odgovara samo jednoj riječi preslikavanjem u oba smjera. Po donjim granicama predložene su nule (0), zato što sve riječi iz rječnika nisu glagoli ili imenice ili nemaju kontekstnu definiciju, a obrnuto je moguće ažurirati hijerarhijske odnose u sustav i prije ažuriranja rječnika.

S ovako definiranim rječnikom u tri tipa entiteta, potrebno je povezati čvorove u tipu entiteta NOK\_FNOK (Node). Poseban algoritam će proći kroz sve FNOK zapise jednog paragrafa i povezati svaki čvor u NOK\_FNOK (Node) s KB\_QUEST, tipom veze KB\_NOD\_QST\_ID. Vodeći čvor ne mora sadržavati ni jedno pitanje, a svi ostali čvorovi u rečenici sadrže najviše jedno pitanje iz KB\_QUEST. Jedno pitanje u KB\_QUEST ne mora biti niti u jednom čvoru, a može se pojaviti u mnogo čvorova. Node je povezan i s tipom entiteta KB\_WORD. Postoje dvije "identifikacijske" veze: KB\_NOD\_WRD\_ID i KB\_PAR\_WRD\_ID. Kroz te veze svaki čvor u rečenici dobiva jedinstvenu identifikaciju i svoje atomsko tumačenje opisano semantikom toga pojma prema rječniku. Identifikacijska veza KB\_NOD\_WRD\_ID dodjeljuje pojedinom čvoru u rečenici identifikator te riječi u rječniku. Identifikacijska veza KB\_PAR\_WRD\_ID dodjeljuje pojedinom čvoru u rečenici identifikator riječi njegovog roditelja.

### 6.1.2 Rastavljanje FNOK rečenice na skup pitanje-odgovor

Transformacija FNOK zapisa u relacijsku bazu podataka završava se algoritmom koji pojedine čvorove iz NOK\_FNOK (Node) i njihove rekurzivne veze HIERARCHY FNOK prepisuje u skup pitanja i odgovora u tipu entiteta KB\_TEXT\_QUEST (Question and answer). Pojedini entitet iz KB\_TEXT\_QUEST predstavlja agregaciju tri vrste entiteta i to: riječi iz KB\_WORD (Dictionary) po vezi PAR\_WRD\_ID (čvora koji traži odgovor, čvor "roditelj (parent)"), pitanja iz KB\_QUEST (Wh-question) po vezi NOD\_QST\_ID) koje čvor "roditelj" postavlja i riječi iz KB\_WORD (Dictionary) po vezi NOD\_WRD\_ID (čvora koji je odgovor na postavljeno pitanje, čvora "dijete (child)"). Prema ovome, svaka FNOK rečenica razbijena je u maksimalan skup atomskih pitanja i odgovora na ta pitanja koja ta rečenica sadrži. Ta pitanja i odgovori predstavljaju skup nezavisnih uvjeta (iskaza, *n*-torki, pravila) koji svi zajedno sadrže semantiku početne rečenice. Tip entiteta KB\_TEXT\_QUEST ima svoj ključni atribut (ID). Svaki entitet u KB\_TEXT\_QUEST povezan je s jednim i samo jednim "parent" ("roditelj") čvorom u NOK\_FNOK (Node) koji će postaviti

jedno i samo jedno pitanje iz KB\_QUEST i na koje će pitanje dobiti jedan i samo jedan odgovor iz NOK\_FNOK (Node). Ova veza nije direktno implementirana, već se ostvaruje preko KB\_WORD (Dictionary). Opišimo veze pojedinog KB\_TEXT\_QUEST (Question and answer) s okolnim tipovima entiteta detaljnije. Jedno pitanje iz KB\_QUEST može biti korišteno u mnogo entiteta u KB\_TEXT\_QUEST. Jedan entitet iz KB\_TEXT\_QUEST uvijek uzima jedno i samo jedno pitanje iz KB\_QUEST. Tip entiteta KB\_TEXT\_QUEST ima onoliko pojavljivanja koliko ima odgovora na pitanje u FNOK zapisu. Jedan entitet iz KB\_TEXT\_QUEST uvijek ima jedan i samo jedan čvor iz NOK\_FNOK (Node) u ulozi roditelja i jedan u ulozi odgovora na pitanje. Čvor u ulozi roditelja traži odgovor na postavljeno pitanje. Vodeći čvorovi na najvišoj razini hijerarhije iz NOK\_FNOK (Node) se ne preslikavaju u svoja pojavljivanja u KB\_TEXT\_QUEST jer oni nisu odgovor niti na jedno pitanje sadržano u FNOK zapisu. Ostali čvorovi se preslikavaju u najviše jedan entitet. Kako za svako pitanje znamo jedinstveni ključ (QUEST\_ID) iz KB\_QUEST, tako isto želimo imati jedinstveni identifikator (WORD\_ID) svakog čvora iz KB\_WORD. Za to koristimo veze PAR\_WRD\_ID i NOD\_WRD\_ID. Ovim postupkom redni broj čvora (NODE\_ID) u NOK\_FNOK (Node) zamjenjujemo općim identifikatorom čvora prema KB\_WORD (Dictionary). Tip veze PAR\_WRD\_ID je preslikavanje koje omogućuje da jedna riječ iz KB\_WORD ne mora biti korištena niti u jednoj rečenici, a može biti korištena u mnogo entiteta u KB\_TEXT\_QUEST. Obrnuto preslikavanje, jedno pitanje i odgovor iz KB\_TEXT\_QUEST uvijek se preslikava na jednu i samo jednu riječ u rječniku koja predstavlja čvor "roditelj" koji postavlja pitanje. Druga veza NOD\_WRD\_ID je po brojnostima ista kao i PAR\_WRD\_ID, a predstavlja povezivanje pitanja s odgovorom, odnosno s riječju u KB\_WORD (Dictionary) koja je odgovor na to pitanje. Tip entiteta KB\_TEXT\_QUEST predstavlja skup entiteta "pitanje-odgovor", pri čemu pojam pitanje nije samo upitna riječ, već povezan čvor koji postavlja to pitanje i upitna riječ. Time smo predložili model koji će hijerarhijski složene FNOK zapise transformirati u skup nezavisnih iskaza među kojima nema direktnih zavisnosti. Semantika integralne rečenice je razložena u skup iskaza, *n*-torki relacijske baze podataka, podataka nad kojima se može provoditi daljnje istraživanje i razvoj QA sustava. Ovime je postignut cilj, a to je putem FNOK formalizma transformirati rečenice u relacijsku bazu podataka.

Transformacija rečenica prirodnog jezika u FNOK zapis je vrlo važna. Bez tog postupka obogaćivanja rečenica pitanjima ne bismo mogli dovesti relacijsku shemu do uređene

trojke: tko pita, pitanje i odgovor. Samim tim ne bi bila uspostavljena veza među riječima te takva baza ne bi mogla poslužiti za odgovaranje na pitanja.

### 6.1.3 Relacijska shema baze podataka

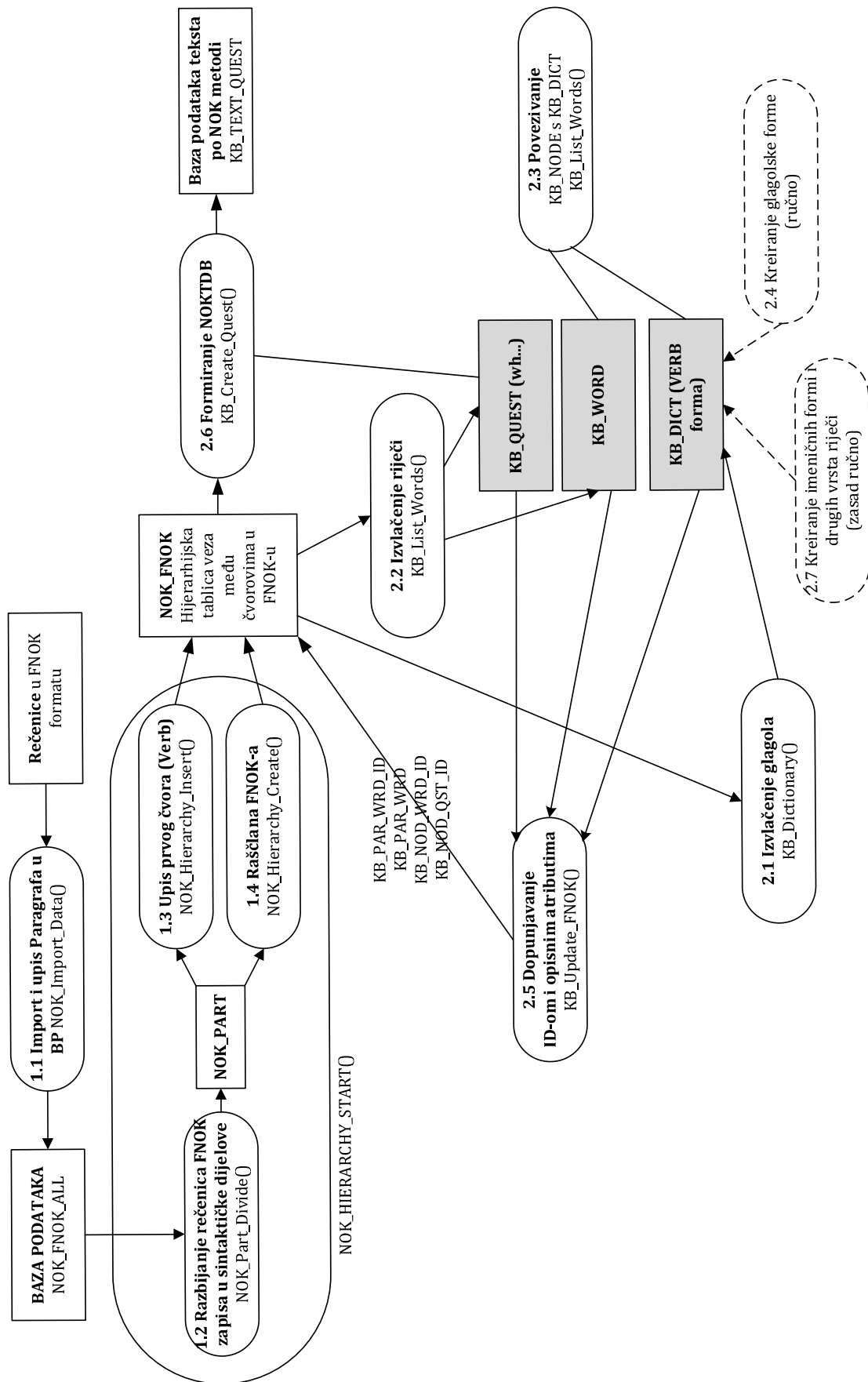
Primjenom pravila za prevođenje DEV-a sa slike 6.2 u relacijsku shemu baze podataka dobivena je relacijska shema sastavljena od sedam tablica (relacija), prikazana u nastavku. Svaki tip entiteta (pravokutnici na DEV-u) postaju tablice u shemi, ime tipa entiteta odgovara imenu tablice. Ključ tablice je atribut ili skup atributa podvučen punom linijom. Vanjski ključevi su podvučeni isprekidanom linijom i nakošeni. Podatkovni sadržaj ovih tablica je prikazan u poglavlju 7.

```
NOK_PARAGRAPH (PRG, NAME)
NOK_FNOK_ALL (PRG, ORD, TENG, FNOK)
NOK_FNOK (NODE_ID, NODE, PRG, ORD, QUEST, PARENT_ID, KB_PAR_WRD_ID,
KB_PAR_WRD, KB_NOD_WRD_ID, KB_NOD_QST_ID)
KB_QUEST (QUEST_ID, QUEST)
KB_DICT (DICT_ID, DICT, DICT_PAR)
KB_WORD (WORD_ID, WORD, DICT_ID)
KB_TEXT_QUEST (ID, PAR_WRD, NOD_WRD, NOD_QST, PRG, ORD, PAR_WRD_ID,
NOD_WRD_ID, NOD_QST_ID)
```

Relacija NOK\_FNOK je denormalizirana kako bi nam se omogućilo lakše praćenje podataka u relacijskoj bazi podataka, odnosno osim identifikatora riječi iz rječnika, dodani su i stupci NODE, QUEST i KB\_PAR\_WRD koje sadrže originalne riječi iz rečenica.

### 6.1.4 Algoritam transformacije FNOK zapisa u BPNOK

U ovom je poglavlju dan opis algoritma transformacije kojim se rečenice inicijalno zapisane u FNOK zapisu kroz niz koraka transformacije zapisuju u relacijskoj bazi podataka teksta prema NOK metodi. Na slici 6.3 detaljnije su razrađeni i opisani procesi sa slike 6.1.



Slika 6.3. Prikaz transformacije formaliziranog teksta FNOK u relacijsku bazu podataka

Inicijalni zapis teksta (rečenica) dan je u FNOK obliku. Njihov import i upis paragrafa teksta u bazu podataka na Oraclu vrši se automatski, u privremenu tablicu NOK\_FNOK\_ALL, a upisuju se rečenice jednog po jednog paragrafa. Procedura kojom prepisujemo tablice NOK\_FNOK\_IMP (dakle rečenice prirodnog jezika i njihove odgovarajuće FNOK zapise) u tablicu NOK\_FNOK\_ALL prikazana je u tablici 1 u Privitku 1.

Transformacija rečenice u hijerarhijski zapis pokreće se procedurom NOK\_Hierarchy() (tablica 2 u Privitku 1). Ona poziva funkciju NOK\_Hierarchy\_Start() i prosljeđuje joj podatak radi li se o FNOK-u ili QFNOK-u te broj paragrafa rečenica ili pitanja. Ovim se složenim procesom formira hijerarhija među riječima u rečenici. Algoritam funkcije NOK\_Hierarchy\_Start() prikazan je u tablici 3 u Privitku 1.

Procedura NOK\_Hierarchy\_Start() sastoji se od tri dijela: razbijanja rečenice u sintaktičke dijelove funkcijom NOK\_Part\_Divide() (tablica 4 u Privitku 1), unosa najprije prvog čvora funkcijom NOK\_Hierarchy\_Insert() (tablica 7 u Privitku 1) koji je glagol (VERB) te unosa svakog preostalog čvora korištenjem rekurzivne procedure NOK\_Hierarchy\_Create() (tablica 8 u Privitku 1).

Funkcija NOK\_Part\_Divide() dijeli FNOK zapis u njegove sintaktičke dijelove te ih upisuje u pomoćnu tablicu NOK\_PART. Za svaki znak (engl. *character*) iz ulaznog FNOK zapisa provjerava se je li on neki od delimitera ( ) , " & \_ (obje zagrade, zarez, navodnik , znak "&" ili podvlaka). U slučaju "&" i "-" potrebno je provjeriti sljedeći znak te ako je on " " (razmak) onda se radi o delimiteru, inače je riječ o dijelu teksta. Ako je pronađen delimiter, ispiše se prethodni sintaktički dio (riječ – text) Part i nakon toga ispiše se i sam delimiter, na kraju se isprazni Part za početak novog upisa. Ukoliko učitani znak nije delimiter, onda ga se pridoda – pripiše prethodnom sintaktičkom dijelu Part. Algoritam funkcije NOK\_Part\_Divide() dan je u tablici 4 u Privitku 1.

Za upisivanje i čitanje sintaktičkih dijelova FNOK zapisa u pomoćnu tablicu NOK\_PART, koriste se funkcije NOK\_Part\_Write(NumP, Part) i NOK\_Part\_Read(Pos), opisane u tablici 5 i tablici 6 u Privitku 1.

U okviru procesa 1.3 Upis prvog čvora, pokreće se funkcija NOK\_Hierarchy\_Insert() (tablica 7 u Privitku 1). Proces 1.3. Upis prvog čvora (Verb) pročita prvi sintaktički dio Part FNOK zapisa iz pomoćne tablice NOK\_PART uz pomoć funkcije NOK\_Part\_Read(Pos). (Sintaksa se ne provjerava, već se pretpostavlja da je prvi sintaktički dio ispravan i da je

to glagol – VERB). Pripremi se prvi čvor `NodId := 1`, `ParId := 0` i `Quest := '_'` te se upiše u tablicu `NOK_FNOK` pozivajući funkciju `NOK_Hierarchy_Insert (PrgId, OrdNo, ParId, NodId, Part, Quest)`. Vrijednosti se dodjeljuju:

`PrgId` → Paragraph ID,

`OrdNo` → Sentence Order Number

U okviru procesa 1.4 Raščlana FNOK-a pročita se (ukoliko ima još dijelova) sljedeći sintaktički dio `Part` pozivanjem funkcije `NOK_Part_Read(Pos)`. Ako ima još dijelova, sljedeći sintaktički dio `Part` trebao bi biti zagrada '(' . Ako sljedeći `Part` nije zagrada '(', prekida se obrada tog zapisa (ide na sljedeći zapis). Ako je sljedeći `Part` zagrada '(', znači da ima i drugih dijelova pa se prelazi na sljedeću razinu hijerarhije pozivajući funkciju `NOK_Hierarchy_Create(TNOK, NOKID, PrgId, OrdNo, NodId, NumP, Pos)` koja je dana u tablici 8 u Pravitku 1.

Rekurzivna funkcija `NOK_Hierarchy_Create()` nastavlja čitati slijedeće sintaktičke dijelove zapisa, stvarati čvorove i upisivati ih u hijerarhijsku tablicu `NOK_FNOK`, korištenjem već navedenih funkcija za čitanje i pisanje. Specifičnost ove rekurzivne funkcije je da kod svake otvorene zagrade '(' u sintaksi zapisa stvara jednu nižu razinu hijerarhije te da se kod svake zatvorene zagrade ')' vraća na prethodnu razinu, sve dok ne obradi sve sintaktičke dijelove. Hijerarhija čvorova `NODE` stvara se u tablici `NOK_FNOK` dodavanjem i popunjavanjem stupca `PARENT_ID` za svaki `NODE_ID`. Provedbom opisanih procedura podaci su zapisani u tablici `NOK_FNOK`. Nad podacima navedene tablice sad se provodi proces obogaćivanja i povezivanja rečenica i rječnika (proces 2). Rječnik (Dictionary) se sastoji od tri tablice: `KB_DICT`, `KB_WORD` i `KB_QUEST`.

Čvor koji je `VERB` po sintaksi je uvijek na prvoj razini hijerarhije (First PARENT). Glagoli se procesom 2.1, odnosno procedurom `KB_Dictionary()`, čitaju iz tablice `NOK_FNOK` (čitaju se sva različita pojavljivanja prvih čvorova, First PARENT = 0) i upisuju u tablicu `KB_DICT`. Ova je procedura dana u tablici 9 u Pravitku 1.

Glagolski čvorovi su posebno izdvojeni jer glagoli u rečenici često imaju različite oblike (npr. 3. lice jednine prezenta ili druge pravilne ili nepravilne glagolske forme). Trenutno se na ovaj način obrađuju glagolske forme te padeži imenica, ali ista se procedura može provesti i nad drugim vrstama riječi.



Tablica NOK\_FNOK sadrži sve riječi koje su se pojavile u ulaznim FNOK zapisima. Ova je tablica prvi upotrebljivi FNOK zapis u relacijskoj bazi podataka. Svaki se čvor iz tablice NOK\_FNOK procedurom KB\_List\_Words() (tablica 10. u Pravitku 1) prepisuje u dictionary i to u jednu od dvije tablice: KB\_QUEST ili KB\_WORD, što je označeno procesom 2.2 na slici 6.3. Svaki se čvor (svaka riječ iz rečenice) izvlači i upisuje u tablicu KB\_WORD, odnosno u tablicu KB\_QUEST ako je riječ o wh - pitanju (pitanja oblika what, who, where, tko, što, gdje i slično). S obzirom na karakteristike formalnog zapisa, sustav može automatski prepoznati u koju od navedenih tablica treba izvršiti upis: u tablicu KB\_WORD upisuju se riječi iz stupca NOK\_FNOK.NODE, a u tablicu KB\_QUEST upisuju se podaci zapisani u stupcu NOK\_FNOK.QUEST.

U okviru iste procedure dana je podrška procesu 2.3. povezivanja riječi iz KB\_WORD sa KB\_DICT. Ako procedura KB\_List\_Words() pronađe istu riječ u tablicama KB\_DICT i KB\_WORD (KB\_DICT.DICT = KB\_WORD.WORD), ona popuni identifikator čvora s identifikatorom riječi u rječniku (popunjava se KB\_WORD.DICT\_ID sa KB\_DICT.DICT\_ID).

S ciljem prepoznavanja glagola ili imenice koji nisu u svojoj osnovnoj glagolskoj formi ili u osnovnom obliku imenice, potrebno je svako pojavljivanje glagola / imenice u formi koja je različita od osnovne forme povezati s osnovnom formom. To se provodi u tablici KB\_DICT, a na slici 6.3 je to prikazano procesom 2.4. Ručno se provjerava je li VERB NODE na primjer, nepravilni oblik glagola, ili neka varijanta koja se razlikuje od glagola u infinitivu. Ako se to utvrdi, navedena riječ povezuje se sa svojim "originalom". Na primjer, riječ *is* se povezuje sa *be*, a riječ *song* sa *sing*.

Međutim, budući da je realno za očekivati da se u rečenicama pojave glagoli u raznim formama prije nego što se pojave u svojoj originalnoj formi (pa još nisu zapisani u KB\_DICT rječniku), unos originalne forme često treba napraviti ručno. Proces u okviru kojeg se u rječniku kreiraju glagolske forme označen je oznakom 2.4.

Da bi se vizualno prikazala hijerarhija rječnika, može se koristiti upit prikazan u tablici 11 u Pravitku 1.

Budući da se i imenice, kao i druge vrste riječi mogu u rečenici pojaviti u obliku koji ne odgovara originalnoj rječničkoj formi (množina s nastavkom -s i slično, padeži), bit će potrebno i njih povezati s originalnim oblicima. Na primjer, ako se u rečenici pojavi riječ *girls*, ona u rječniku treba biti povezana s jedninom *girl*.

Unos onih originalnih formi koje se još nisu pojavile u rečenicama pa tako ni u rječniku, prikazan je procesom 2.7. Isprekidana linija označava da se taj dio radi ručno.

Inicijalno punjenje tablice NOK\_FNOK transformiralo je FNOK zapis rečenica u relacijsku bazu podataka te je tablica NOK\_FNOK napunjena čvorovima. Preostaje povezati retke u toj tablici s riječima iz rječnika (tablice KB\_WORD i KB\_QUEST). S tim ciljem dopunjena je struktura tablice NOK\_FNOK atributima KB\_NOD\_WRD\_ID i KB\_NOD\_QST\_ID, odnosno veza je s navedenim tablicama iz rječnika ostvarena preko ID-a. Međutim, zbog lakšeg praćenja rezultata (ljudima bliskog), dodani su i stupci KB\_PAR\_WRD\_ID i KB\_PAR\_WRD koji opisuju nadređeni čvor. Punjenje ova četiri atributa provodi se u okviru procesa 2.5 procedurom KB\_Update\_FNOK() (tablica 12 u Priritku 1).

Prikazani sustav je sustav u kojem se događaju dva važna elementa:

- Svaka rečenica obogaćuje rječnik novim riječima (kad se riječ pojavi prvi put u nekoj rečenici teksta, ona se upisuje u rječnik jer tamo još ne postoji. Dodjeljuje joj se identifikacija (ID)).
- Rječnik obogaćuje svaku riječ u rečenici (u svakoj se rečenici (kad sve riječi rečenice postoje u rječniku) prilikom njenog zapisa u bazu podataka vezuje konkretno pojavljivanje riječi u rečenici s odgovarajućom riječi iz rječnika. Tako rječnik obogaćuje izvorni zapis riječi u rečenici).

Nakon kreiranja šifarnika pitanja KB\_QUEST i šifarnika čvorova KB\_WORD te njihovog povezivanja s tablicom NOK\_FNOK preko identifikatora pitanja (upitnih riječi) i identifikatora čvorova, može se kreirati i napuniti konačna tablica s pitanjima KB\_TEXT\_QUEST dobivena iz ulaznih početnih FNOK zapisa rečenica (NOK\_FNOK\_ALL (Sentences)). Tablica KB\_TEXT\_QUEST sadrži attribute ID, oznaku paragrafa te oznaku rečenice na temelju koje je pitanje kreirano. Osim navedenih atributa, tablica sadrži i:

PAR_WRD_ID	Parent Word Identifier
NOD_WRD_ID	Node Word Identifier
NOD_QST_ID	Node Question Identifier.

S ciljem lakšeg praćenja rezultata testiranja, a uzevši u obzir činjenicu da su identifikacijske oznake numeričke, dodane su i njihove tekstualne vrijednosti: PAR\_WRD, NOD\_WRD, NOD\_QST. Ovi tekstualni zapisi služe samo za vizualnu provjeru i ne koriste se u daljnjim algoritmima.

U okviru procesa 2.6 procedura KB\_Create\_QUEST(PrgId) puni ovu tablicu uzimajući gotove podatke pripremljene u "radnoj" tablici NOK\_FNOK (tablica 13 u Privitku 1).

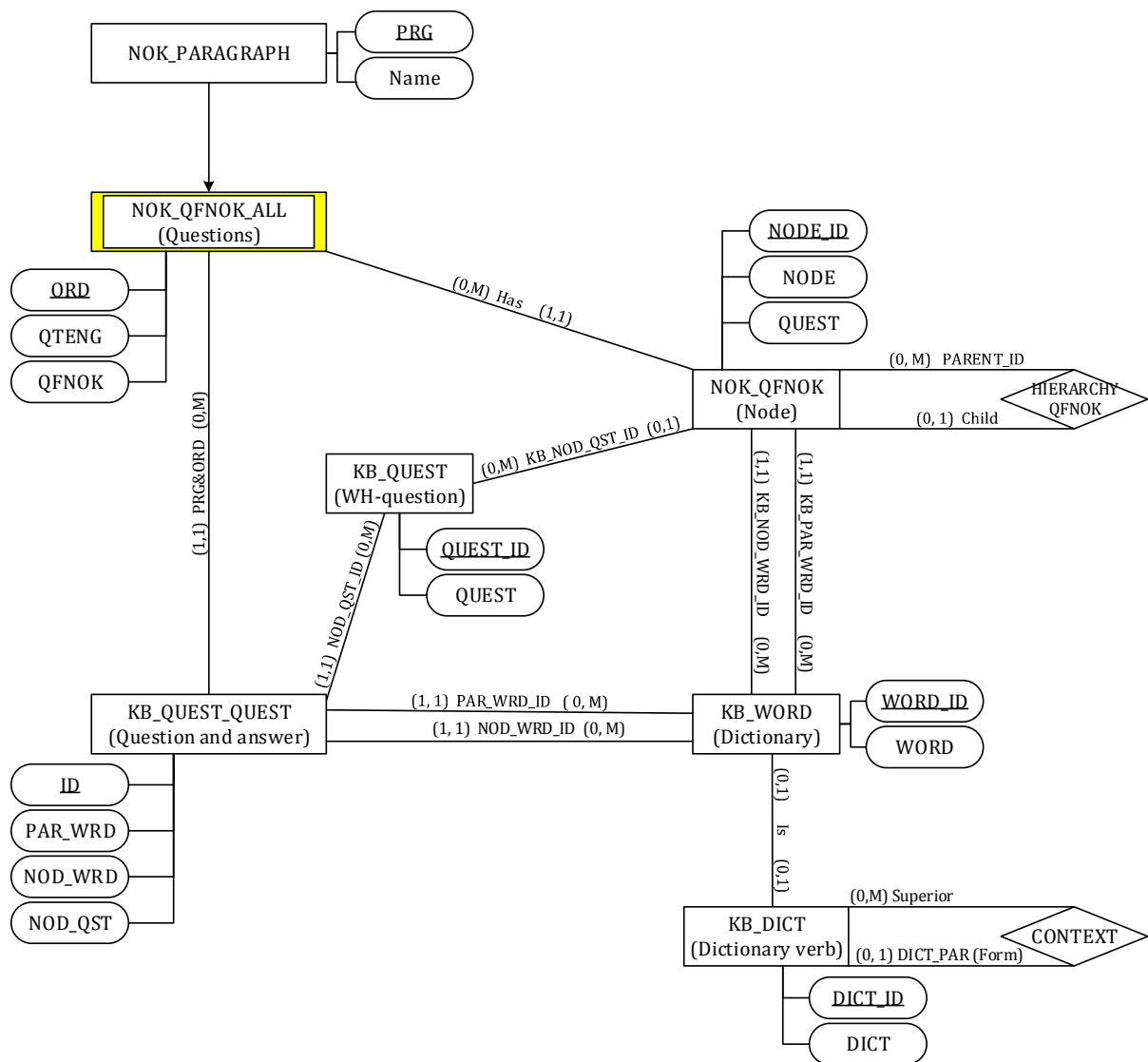
## 6.2 Meta model QFNOK zapisa

Postavlja se istraživačko pitanje "kako QFNOK zapise pitanja ili upitnih rečenica upisati u relacijsku bazu podataka?". Kako bismo odgovorili na to pitanje, potrebno je analizirati podatkovni sadržaj QFNOK zapisa i QFNOK formalizam te naći odgovarajući meta model. Na osnovi meta modela podataka QFNOK formalizma potrebno je definirati relacijsku shemu baze podataka koja će sadržavati QFNOK zapise i algoritme transformacije početnog zapisa upitnih rečenica u povezane podatke u bazi podataka.

QFNOK formalizam omogućuje zapis upitnih rečenica (pitanja) prirodnog jezika u QFNOK zapise obogaćene simbolima i upitnim riječima. Analizirajući pravila zapisivanja rečenica u QFNOK zapis, dolazimo do modela QFNOK formalizma (meta model). Meta model QFNOK-a je prikazan na slici 6.4.

Obrada pitanja, odnosno QFNOK zapisa slična (izmorfna) je obradi rečenica, odnosno pripadnih FNOK zapisa. Obrada pitanja u QFNOK zapisu odvija se paralelno s obradom rečenica u FNOK zapisu. Sve prethodno navedene procedure i funkcije za FNOK, rade i na jednu i drugu stranu istovremeno, odnosno vrijede i za QFNOK.

Opis meta modela sa slike 6.4 analogan je opisu meta modela sa slike 6.2. Prva razlika na ta dva modela je što se na slici 6.2 NOK\_PARAGRAPH sastoji od NOK\_FNOK\_ALL (Sentences), a sada se NOK\_PARAGRAPH sastoji od NOK\_QFNOK\_ALL (Questions). Tip entiteta NOK\_QFNOK\_ALL (Questions) ima tri atributa: redni broj rečenice (ORD), tekst upitne rečenice (pitanja) na prirodnom jeziku (QTENG, skraćeno od pitanje na engleskom jeziku) i QFNOK zapis pitanja (QFNOK).



Slika 6.4. Meta model strukture i sadržaja QFNOK zapisa

Svaki QFNOK zapis iz NOK\_QFNOK\_ALL (Questions) također se sastoji od riječi (čvorova), među kojima postoji hijerarhijski odnos. Čvorovi su prikazani tipom entiteta Čvor (oznake NOK\_QFNOK (Node)). Svaki čvor pripada jednom i samo jednom pitanju, odnosno QFNOK zapisu. Svaki čvor ima svoj identifikator (NODE\_ID) kao redni broj pojavljivanja unutar pitanja i atribut naziv čvora (NODE) što predstavlja čvor iz polaznog QFNOK zapisa. Jedan zapis može imati više čvorova, odnosno niti jedan (ako još nije proveden proces transformacije zapisa u čvorove). Jedan čvor u NOK\_QFNOK (Node) pripada uvijek jednom i samo jednom pitanju iz NOK\_QFNOK\_ALL (Questions).

Tip entiteta NOK\_QFNOK (Node) povezan je sam sa sobom povratnim tipom veze HIERARCHY QFNOK prikazane rombom. Opis HIERARCHY QFNOK identičan je opisu HIERARCHY FNOK. Povezivanje čvorova QFNOK zapisa s rječnikom te rastavljanje QFNOK

zapisa na skup pitanje – odgovor identično je povezivanju čvorova FNOK zapisa s rječnikom te rastavljanje FNOK zapisa na skup pitanje – odgovor opisanim u poglavljima 6.1.1 i 6.1.2. Rastavljanjem QFNOK zapisa na skup pitanje – odgovor dolazimo do tipa entiteta KB\_QUEST\_QUEST koji ima svoj ključni atribut (ID). Svaki entitet u KB\_QUEST\_QUEST povezan je s jednim i samo jednim "parent" ("roditelj") čvorom u NOK\_QFNOK (Node) koji će postaviti jedno i samo jedno pitanje iz KB\_QUEST i na koje će pitanje dobiti jedan i samo jedan odgovor iz NOK\_QFNOK (Node). Ova veza nije direktno implementirana, već se ostvaruje preko KB\_WORD (Dictionary).

### 6.2.1 Relacijska shema baze podataka

Primjenom pravila za prevođenje DEV-a sa slike 6.4 u relacijsku shemu baze podataka dobivena je relacijska shema sastavljena od sedam tablica (relacija), prikazana u nastavku.

```

NOK_PARAGRAPH (PRG, NAME)
NOK_QFNOK_ALL (PRG, ORD, QTENG, QFNOK)
NOK_QFNOK (NODE_ID, NODE, PRG, ORD, QUEST, PARENT_ID, KB_PAR_WRD_ID,
KB_PAR_WRD, KB_NOD_WRD_ID, KB_NOD_QST_ID)
KB_QUEST (QUEST_ID, QUEST)
KB_DICT (DICT_ID, DICT, DICT_PAR)
KB_WORD (WORD_ID, WORD, DICT_ID)
KB_QUEST_QUEST (ID, PAR_WRD, NOD_WRD, NOD_QST, PRG, ORD, PAR_WRD_ID,
NOD_WRD_ID, NOD_QST_ID)

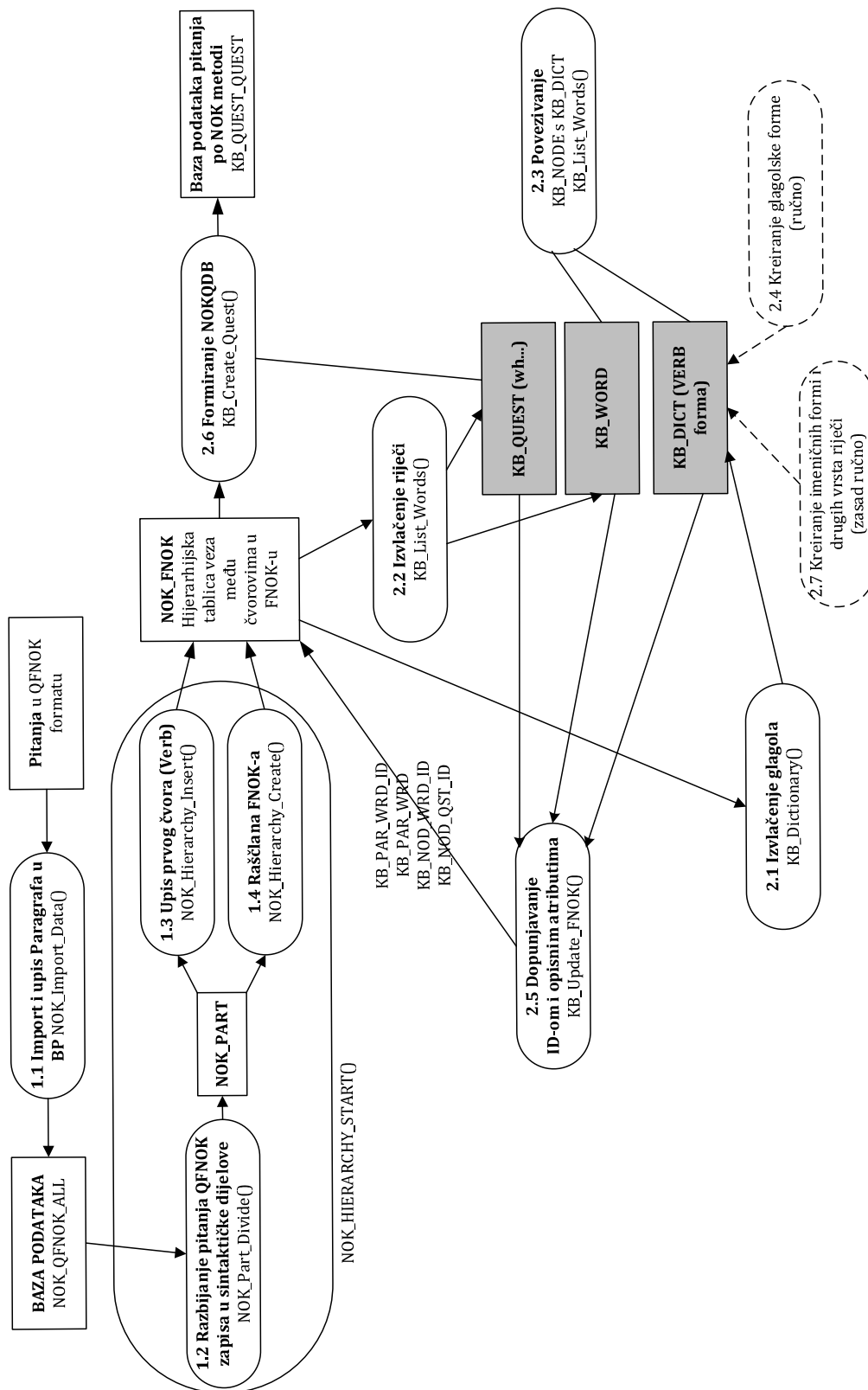
```

Relacija NOK\_QFNOK je denormalizirana relacijska shema jer je u nju dodano polje KB\_PAR\_WRD.

### 6.2.2 Algoritam transformacije QFNOK zapisa u BPNOK

Algoritam transformacija QFNOK zapisa u BPNOK analogan je algoritmu transformacija FNOK zapisa u BPNOK i prikazan je na slici 6.5.

Inicijalni zapis upitnih rečenica (pitanja) dan je u QFNOK obliku. Njihov import i upis paragrafa teksta u bazu podataka na Oraclu vrši se procedurom NOK\_Import\_Data(), u privremenu tablicu NOK\_QFNOK\_ALL, a upisuju se pitanja jednog po jednog paragrafa. Algoritam procedure NOK\_Import\_Data() je prikazan u tablici 1 u Pravitku 1.



Slika 6.5. Prikaz transformacije QFNOK zapisa u relacijsku bazu podataka

Transformacija pitanja u hijerarhijski zapis pokreće se procedurom `NOK_Hierarchy()` koja poziva funkciju `NOK_Hierarchy_Start()` i prosljeđuje joj podatak da se radi o QFNOK-u. Procedurom `NOK_Hierarchy(PrgId)` sva pitanja iz nekog paragrafa razbijemo,

rastavimo i prepisemo u hijerarhijske zapise u tablicu NOK\_QFNOK. Ove tablice sadrže sve čvorove (nodes) u hijerarhiji. Za ovu proceduru nam je potrebna pomoćna (radna) tablica NOK\_PART.

Struktura procedure:

```
proc NOK_Hierarchy(PrgId)
  func NOK_Hierarchy_Start(TNOK, PrgId)
  func NOK_Part_Divide(FNOK)
  func NOK_Part_Write(NumP, Part)
  func NOK_Part_Read(Pos)
  func NOK_Hierarchy_Insert(TNOK, PrgId, OrdNo, ParId, NodId, Part, Quest)
    - prvi čvor
  func NOK_Hierarchy_Create(TNOK, NOKID, PrgId, OrdNo, NodId, NumP, Pos)
    - ostali čvorovi
  func NOK_Part_Read(Pos)
  func NOK_Hierarchy_Insert(TNOK, PrgId, OrdNo, ParId, NodId, Part, Quest)
    - čvor
  func NOK_Hierarchy_Create(TNOK, NOKID, PrgId, OrdNo, NodId, NumP, Pos)
    - rekurzija
```

Sada u tablici NOK\_QFNOK imamo ulazne zapise razbijene u hijerarhije čvorova.

Iz tablice NOK\_QFNOK izvlačimo sve glagole (verb) u KB\_DICT procedurom KB\_Dictionary (PrgId). Tablicu KB\_DICT uređujemo na način da joj dodajemo osnovne (generičke) forme glagola i imenica te da povežemo sve pripadne forme na osnovnu formu. Pritom moramo paziti da svaka osnovna forma pokazuje na samu sebe – svatko mora imati svoga nadređenog čvora. Ova će se tablica s vremenom nadopunjavati i postajati sve bogatija. U tablicu se mogu upisati svi novi (očekivani) glagoli i imenice unaprijed, ili ih se može prepisivati iz nekog rječnika.

Nakon toga se iz tablice NOK\_QFNOK izvlače i ostale riječi (wh-pitanja; te imenice, pridjeve, članove, ...) pa ih upisujemo u tablice KB\_QUEST i KB\_WORD procedurom KB\_List\_WORDS(PrgId). Time je rječnik potpuno uređen i sastoji se od tri tablice: KB\_DICT, KB\_QUEST i KB\_WORD.

Tablicu NOK\_QFNOK nadopunjujemo riječima i identifikatorima iz rječnika pomoću procedure KB\_Update\_FNOK(PrgId). Ovom procedurom dopunjujemo stupce KB\_PAR\_WRD, KB\_PAR\_WRD\_ID, KB\_NOD\_WRD\_ID i KB\_NOD\_QST\_ID.

Tablicu KB\_QUEST\_QUEST punimo procedurom KB\_Create\_QUEST(PrgId).

Kao što tablica KB\_TEXT\_QUEST ima sve hijerarhijske zapise čvorova rečenica tako tablica KB\_QUEST\_QUEST ima sve hijerarhijske zapise čvorova pitanja. Posebnost ovih tablica je u

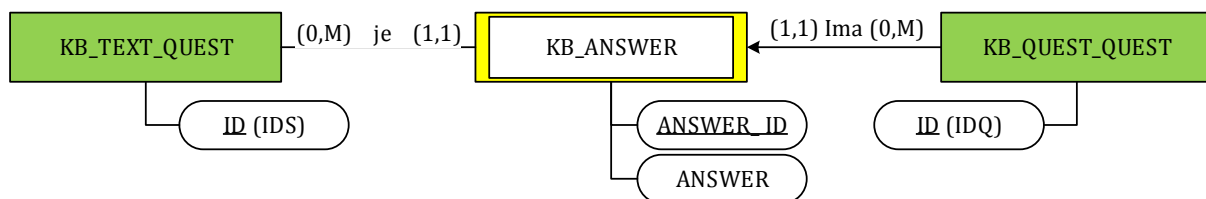
tome što osim identifikatora riječi (verb, question, word, ...) iz rječnika koji će nam trebati u svim budućim obradama, redundantno sadrže i same riječi za bolje i lakše čovjekovo razumijevanje i praćenje.

### 6.3 Kreiranje odgovora

Od trenutka kada su tablice KB\_TEXT\_QUEST i KB\_QUEST\_QUEST napunjene podacima, možemo početi s generiranjem Odgovora. Podsjetimo, tablice KB\_TEXT\_QUEST i KB\_QUEST\_QUEST sadrže sve hijerarhijske zapise čvorova početnih rečenica (FNOK zapisa), odnosno pitanja (QFNOK zapisa). Konkretno, tablica KB\_TEXT\_QUEST sadrži broj paragrafa rečenice kojem pripada (PRG), redni broj rečenice unutar tog paragrafa (ORD), identifikator (ID) nadređenog čvora (PAR\_WRD\_ID), identifikator čvora riječi, odnosno trenutnog (aktivnog) čvora (NOD\_WRD\_ID) te identifikator uloge (pitanja) kojim su nadređeni i aktualni čvor u vezi (NOD\_QST\_ID). Posljednja tri polja u toj tablici PAR\_WRD, NOD\_WRD i NOD\_QST služe nam za lakše praćenje podataka u tablici i sastoje se od nadređenog čvora, aktualnog čvora i pitanja (uloge) kojom su ta dva čvora u vezi, odnosno uz identifikatore čvorova i uloga imamo i njihove konkretne vrijednosti.

Analogan opis vrijedi i za tablicu KB\_QUEST\_QUEST koja koristi podatke iz tablice pitanja.

Meta model strukture i sadržaja FNOK zapisa prikazan na slici 6.2 i meta model strukture i sadržaja QFNOK zapisa prikazan na slici 6.4 povezani su u novi pogled na sustav koji je još i obogaćen dijelom koji prikazuje odgovaranje na pitanja. Meta model strukture i sadržaja odgovora prikazan je na slici 6.6. Struktura KB\_TEXT\_QUEST i KB\_QUEST\_QUEST razrađena je na slikama 6.2 i 6.4 te ovdje nije ponovo prikazivana.



Slika 6.6. Meta model strukture i sadržaja odgovora

Slabi tip entiteta KB\_ANSWER s atributima Answer\_ID i Answer, ovisan je o tipu entiteta KB\_QUEST\_QUEST. Za jedan hijerarhijski zapis čvorova pitanja u KB\_QUEST\_QUEST imamo najmanje nijedno, a najviše mnogo odgovora iz tipa entiteta KB\_ANSWER. Obrnuto, za jedan odgovor iz tipa entiteta KB\_ANSWER imamo najmanje jedan i najviše jedan hijerarhijski zapis čvorova pitanja u tipu entiteta KB\_QUEST\_QUEST. Za jedan odgovor iz



tipa entiteta `KB_ANSWER` imamo najmanje jedan i najviše jedan hijerarhijski zapis čvorova rečenica u tipu entiteta `KB_TEXT_QUEST`. Za jedan hijerarhijski zapis čvorova rečenica u `KB_TEXT_QUEST` imamo najmanje nijedno, a najviše mnogo odgovora iz slabog tipa entiteta `KB_ANSWER`.

Rezultat pretvaranja DEV-a sa slike 6.6 (ali uključujući i slike 6.2 i 6.4) u relacijsku shemu baze podataka BPNOK je:

```

NOK_PARAGRAPH (PRG, NAME)
KB_QUEST (QUEST_ID, QUEST)
KB_DICT (DICT_ID, DICT, DICT_PAR)
KB_WORD (WORD_ID, WORD, DICT_ID)
NOK_FNOK_ALL (PRG, ORD, TENG, FNOK)
NOK_FNOK (NODE_ID, NODE, PRG, ORD, QUEST, PARENT_ID, KB_PAR_WRD_ID,
KB_PAR_WRD, KB_NOD_WRD_ID, KB_NOD_QST_ID)
KB_TEXT_QUEST (ID, PAR_WRD, NOD_WRD, NOD_QST, PRG, ORD, PAR_WRD_ID,
NOD_WRD_ID, NOD_QST_ID)
NOK_QFNOK_ALL (PRG, ORD, QTENG, QFNOK)
NOK_QFNOK (NODE_ID, NODE, PRG, ORD, QUEST, PARENT_ID, KB_PAR_WRD_ID,
KB_PAR_WRD, KB_NOD_WRD_ID, KB_NOD_QST_ID)
KB_QUEST_QUEST (ID, PAR_WRD, NOD_WRD, NOD_QST, PRG, ORD, PAR_WRD_ID,
NOD_WRD_ID, NOD_QST_ID)
KB_ANSWER (IDQ, ANSWER_ID, ANSWER, IDS)

```

pri čemu je `IDQ=ID.KB_QUEST_QUEST` i `IDS=ID.KB_TEXT_QUEST`.

Radi lakšeg praćenja i razumijevanja rezultata u tablicu `KB_ANSWER` dodat ćemo još nekoliko polja pa relacija `KB_ANSWER` izgleda ovako:

```

KB_ANSWER (ANSWER_ID, PRGQ, ORDQ, QTENG, ANSWER, PRGS, ORDS, TENG,
QFNOK, FNOK)

```

U prikazu vidimo nazive atributa umjesto ključeva. `IDQ` (ID pitanja) zamijenili smo s `PRGQ` i `ORDQ` (broj paragrafa i broj pitanja unutar paragrafa), `IDS` (ID rečenice) zamijenili smo s `PRGS` i `ORDS` (brojem paragrafa i brojem rečenice unutar tog paragrafa). Osim toga, u relaciju je još dodano pitanje na prirodnom jeziku (`QTENG`) za koje tražimo odgovor i njegov odgovarajući formalizirani zapis (`QFNOK`) te rečenicu na prirodnom jeziku (`TENG`) u kojoj je odgovor pronađen i njen odgovarajući formalizirani zapis (`FNOK`). Time se struktura same baze podataka ne mijenja, već samo prikaz baze podataka u ovom radu.

Analizom tablice `KB_TEXT_QUEST` vidimo da su svi stupci popunjeni (vidi tablicu 7.7). Ukoliko neki podatak nedostaje, to znači da tog čvora nema.

Analizom tablice `KB_QUEST_QUEST` vidimo da uvijek nešto nedostaje, a ono što nedostaje je upravo ono što tražimo kao odgovor (vidi tablicu 7.10).

Ako u tablici KB\_QUEST\_QUEST (dio te tablice je prikazan u tablici 7.10) provjerimo postojeće (moguće) kombinacije teksta (popunjeni čvor ili pitanje označavamo s 'TEXT', čvor koji tražimo s 'X' te ulogu između dvaju čvorova koja nije definirana s '\_'):

```
Select distinct
  Case When PAR_WRD in ('X', '_') Then PAR_WRD Else 'TEXT' End PARENT
, Case When NOD_WRD in ('X', '_') Then NOD_WRD Else 'TEXT' End NODE
, Case When NOD_QST in ('X', '_') Then NOD_QST Else 'TEXT' End QUESTION
From KB_QUEST_QUEST;
```

dobijemo rezultat prikazan u tablici 6.1.

Tablica 6.1. Različite kombinacije za Parent, Node i Question u tablici KB\_QUEST\_QUEST

PARENT	NODE	QUESTION
TEXT	X	_
TEXT	TEXT	_
TEXT	X	TEXT
X	TEXT	_

Slično možemo napraviti i s identifikatorima (ključevima):

```
Select distinct
  Case When PAR_WRD_ID is null Then 'null' Else 'ID' End PARENT
, Case When NOD_WRD_ID is null Then 'null' Else 'ID' End NODE
, Case When NOD_QST_ID is null Then 'null' Else 'ID' End QUESTION
From KB_QUEST_QUEST;
```

pa dobijemo rezultat prikazan u tablici 6.2.

Tablica 6.2. Različite kombinacije za Parent, Node i Question u tablici KB\_QUEST\_QUEST za identifikatore

PARENT	NODE	QUESTION
ID	null	null
ID	ID	null
ID	null	ID
null	ID	null

Iz ovoga zaključujemo:

- Kada je identifikator pitanja NOD\_QST\_ID prazan (null), u njegovom NOD\_QST se uvijek nalazi '\_', što znači da to nikada ne tražimo.
- Kada je identifikator nadređenog čvora (parenta) PAR\_WRD\_ID prazan (null), u njegovom PAR\_WRD se uvijek nalazi 'X', što znači da **njega tražimo**. 'X' ćemo pronaći kada je NOD\_WRD\_ID isti (KB\_QUEST\_QUEST.NOD\_WRD\_ID = KB\_TEXT\_QUEST.NOD\_WRD\_ID), bez obzira na to što piše u NOD\_QST\_ID.
- Kada je identifikator nadređenog čvora PAR\_WRD\_ID popunjen, imamo različite situacije:

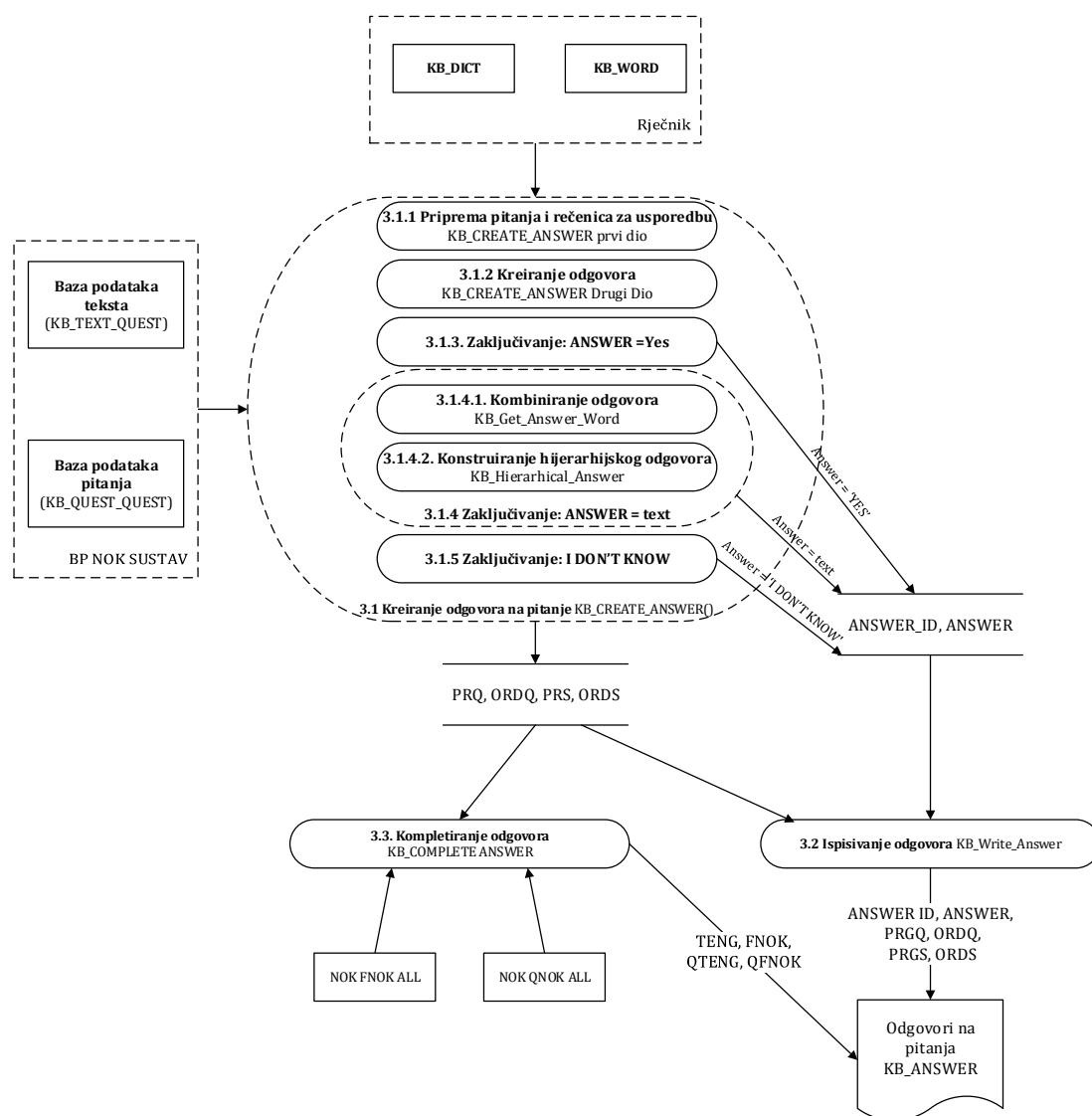
- Kada je identifikator čvora NOD\_WRD\_ID popunjen, moramo ga usporediti (naći istoga) i uzeti jer se vjerojatno radi o polarnom pitanju.
  - Kada je identifikator čvora NOD\_WRD\_ID prazan (null), moramo naći istoga i uzeti u obzir, ali moramo provjeriti da li je uloga (NOD\_QST\_ID) prazna te dodati NOD\_WRD u odgovor ili pokazuje na konkretno pitanje na koje moramo odgovoriti pa ako je u redu i to dodati u odgovor.

Ove dvije spoznaje, utječu na pojednostavljivanje algoritma za generiranje odgovora jer nam omogućuju brzo uparivanje čvorova pitanja s čvorovima rečenica.

Slika 6.7 prikazuje model procesa niže razine za proces 3. Generiranje odgovora na pitanje sa slike 6.1. Kada su tablice KB\_TEXT\_QUEST i KB\_QUEST\_QUEST (koje čine BPNOK) popunjene, možemo započeti generiranje odgovora u tablicu KB\_ANSWER. Odgovore generiramo procedurom KB\_Create\_Answer (PrgIdQ, PrgIdS). Parametar PrgIdQ je ID paragrafa pitanja (ID Paragraph of Questions) i može sadržavati broj paragrafa pitanja kojega želimo obraditi ili 0 (null) ako želimo obrađivati sve paragrafe pitanja. Parametar PrgIdS je ID paragrafa rečenica (ID Paragraph of Sentences) i može sadržavati broj paragrafa rečenica kojega želimo obraditi ili 0 (null) ako želimo obrađivati sve paragrafe rečenica.

Na taj način možemo obrađivati:

- jedan paragraf pitanja sa jednim paragrafom rečenica,
- jedan paragraf pitanja sa svim paragrafima rečenica,
- sve paragrafe pitanja sa jednim paragrafom rečenica,
- sve paragrafe pitanja sa svim paragrafima rečenica.



Slika 6.7. Model procesa niže razine za proces 3. Generiranje odgovora na pitanje.

Funkcije koje koristi procedura `KB_Create_Answer(PrgIdQ, PrgIdS)` su:

- `KB_Hierarchical_Answer(PrgS, OrdS, ParS)`: Služi za konstruiranje hijerarhijskog odgovora (Answer), ovisno od toga koliko razina hijerarhije ima čvor odgovora (tablica 16 u Privitku 1).  
Parametar `PrgS` je broj paragrafa rečenice, `OrdS` je redni broj rečenice unutar paragrafa, a `ParS` je identifikator nadređenog čvora (parenta).
- `KB_Get_Answer_Word(WrdId)`: Služi za pretvorbu identifikatora riječi u samu riječ (text) kako bi se kombinirao (konstruirao) određeni odgovor (tablica 15 u Privitku 1).
- `KB_Write_Answer (AnsId, Answer, PrgQ, OrdQ, PrgS, OrdS)`: Služi za upisivanje pojedinog odgovora Answer u konačnu tablicu `KB_ANSWER` (tablica 17 u Privitku 1).

- Parametar AnsId je identifikator zadnjeg prethodno upisanog odgovora, kako bi funkcija povećala identifikator za trenutni (aktualni) odgovor.
  - Answer je konkretan tekst odgovora.
  - PrgQ je broj paragrafa postavljenog pitanja.
  - OrdQ je redni broj postavljenog pitanja unutar paragrafa.
  - PrgS je broj paragrafa rečenice u kojoj je pronađen odgovor.
  - OrdS je redni broj rečenice u kojoj je pronađen odgovor.
- **KB\_Complete\_Answers()**: služi na kraju algoritma da bi se popunila tablica KB\_ANSWER s originalnim tekstovima rečenica i pitanja te njihovim formaliziranim zapisima (TENG, FNOK, QTENG, QFNOK), čime se omogućuje ljudima jednostavnije analiziranje točnosti (korektnosti) provedenog algoritma generiranja odgovora (tablica 18 u Pravitku 1).

Kod za proceduru KB\_Create\_Answer (PrgIdQ, PrgIdS) prikazan je u tablici 14 u Pravitku 1. U okviru procesa 3.1. Kreiranje odgovora na pitanje, procedura **KB\_Create\_Answer (PrgIdQ, PrgIdS)** za svoj rad koristi pomoćne tablice XTT\_QST i XTT\_TXT. U XTT\_QST upisuje čvorove pitanja (nodes of questions), a u XTT\_TXT upisuje čvorove rečenica (nodes of sentences). Ove tablice pune se preuzimajući podatke iz KB\_QUEST\_QUEST i KB\_TEXT\_QUEST iz baze podataka BPNOK te iz KB\_DICT i KB\_WORD iz rječnika.

Ova procedura u ovisnosti od tipa tablice KB\_QUEST\_QUEST ili KB\_TEXT\_QUEST, popuni stupce:

- ID – Unikatni identifikator čvora (node),
- PRG – Paragraf,
- ORD – Redni broj zapisa,
- PAR – Nadređeni čvor (Parent node),
- NOD – Čvor,
- QST – Ulogu (Pitanje) (wh-question).

Također, napuni i PAR\_DICT (Parent Verb) i NOD\_DICT (Node Verb) te njihove identifikatore koristeći podatke iz KB\_DICT.

Na kraju dodaje i tri pomoćna stupca:

- FLW – Flow Order koji unikatno određuje svaki pojedini zapis (rečenicu ili pitanje) kroz sve upotrebene paragrafe,

- RID – Row Identifier koji unikatno određuje svaki pojedini čvor (node) kroz sve upotrebljene paragrafe i zapise,
- OK – *flag* koji služi za označavanje čvorova (nodes) koji su uspješno upareni u usporedbi pitanja s rečenicom, kako bi se moglo ustanoviti daje li rečenica odgovor na pitanje.

Kada su pomoćne (radne) tablice napunjene, proces 3.1.2 Kreiranje odgovora koristeći drugi dio procedure `KB_Create_Answer` (`PrgIdQ`, `PrgIdS`) kreće u kreiranje (traženje) odgovora. Uzima se pitanje po pitanje (za sva pitanja iz parametra `PrgIdQ`) i za svako pitanje prođe kroz sve rečenice (sentences) (iz parametra `PrgIdS`) te najprije označi sve njihove čvorove (nodes) (`OK = 'N'`).

Procedura tada krene u prvi krug (proces 3.1.3 Zaključivanje: YES), pokušavajući upariti sve čvorove određenog pitanja sa svim čvorovima iz rečenica (Questions & Sentences) koji se preklapaju-uparuju, kako bi se mogao dati potpuni odgovor (answer) polarni 'YES' – ako ima čvorova rečenice koje se u potpunosti preklapaju s čvorovima pitanja. Za svaku kombinaciju pitanje – rečenica uzme sve njihove čvorove (nodes). Sve čvorove pitanja i čvorove rečenice koji su već upareni (`OK = 'Y'`) preskače. Ako je `ParQ` (Question Parent) popunjen i jednak `ParS` (Sentence Parent) (ili njihov generički oblik `ParDictQ = ParDictS`) i `NodQ` (Question Node) popunjen i jednak `NodS` (Sentence Node) (ili njihov generički oblik `NodDictQ = NodDictS`) označi čvor pitanja (question node) i čvor rečenice (sentence node) kao dobre (`OK = 'Y'`) te poveća broj uparenih čvorova. Ako je na kraju broj uparenih čvorova jednak broju čvorova pitanja, znači da je našao rečenicu koja u potpunosti odgovara na pitanje te je odgovor `Answer = 'YES'` i on se upisuje u tablicu `KB_ANSWER` funkcijom `KB_Write_Answer` (`AnsId`, `Answer`, `PrgQ`, `OrdQ`, `PrgS`, `OrdS`) (proces 3.2. ispisivanje odgovora). Ako je algoritam došao do odgovora 'YES', znači da je rečenica zadovoljila pitanje, ide dalje na slijedeću rečenicu pa se postupak ponavlja (ide na prvi krug). Ako čvorovi rečenice nisu zadovoljili sve čvorove pitanja, ide u drugi krug. U drugom krugu (proces 3.1.4. Zaključivanje: ANSWER) pokušava naći odgovore na čvorove istog Pitanja koji nisu bili upareni u prvom krugu, na njih treba naći Odgovore iz preostalih Čvorova istih Rečenica. Za istu (prethodnu) kombinaciju Pitanje-Rečenica (Question-Sentence) ponovno uzme sve njihove Čvorove (Nodes) i preskače one koji su već upareni (`OK='Y'`). Za neuparene Čvorove Pitanja odgovor traži na sljedeći način: Ako je `ParQ` (Question Parent) popunjen i `NodQ` (Question Node) popunjen, a nije već uparen u

prvom krugu, znači da na taj Čvor Pitanja nema odgovora niti u jednom Čvoru Rečenice i da ta Rečenica ne daje Odgovor na to Pitanje pa odustaje od te Rečenice i ide na slijedeću Rečenicu – vraća se na početak prvog kruga. Ako je ParQ (Question Parent) popunjen, a NodQ (Question Node) prazan, treba popuniti NodQ (Question Node) iz NodS (Sentence Node) tako da se najprije provjeri je li ParQ (Question Parent) jednak ParS (Sentence Parent) (ili njihov generički oblik ParDictQ = ParDictS) pa ako su ParQ i ParS isti, provjeri i QstQ (Wh-question of Question Node) koji mora biti prazan ili jednak QstS (Wh-question of Sentence Node). Kada su upareni Parent i Wh-question, oba označi s OK='Y' i poveća broj uparenih čvorova, a Odgovor (Answer) za NodQ dobije iz NodS (Sentence Node) korištenjem funkcije KB\_Get\_Answer\_Word(NodS) (proces 3.1.4.1). Takav NodS (Sentence Node) može imati svoje podređene Čvorove u hijerarhiji pa se kompletan Odgovor dobije korištenjem funkcije KB\_Hierarchical\_Answer(PrgS, OrdS, NodS) (proces 3.1.4.2) koja izvuče (vrati) cijelu hijerarhiju (tekst) tog Čvora Rečenice NodS. Ako je, za razliku od prethodnog stavka, ParQ (Question Parent) prazan, što znači da je NodQ (Question Node) popunjen, provjeri da li je NodQ jednak NodS (Sentence Node) (ili njihov generički oblik NodDictQ = NodDictS). Ako je to u redu, Odgovor (Answer) dobije iz funkcije KB\_Get\_Answer\_Word(ParS), korištenjem ParS (Sentence Parent). Označi uparene čvorove s OK='Y' i poveća broj uparenih čvorova.

Bilo kojim putem došao do ove točke, provjeri je li broj uparenih čvorova jednak broju Čvorova Pitanja pa ako je upario sve Čvorove Pitanja i složio Odgovor (Answer), ispiše ga u tablicu KB\_ANSWER funkcijom KB\_Write\_Answer (AnsId, Answer, PrgQ, OrdQ, PrgS, OrdS). Ako je složio Odgovor (Answer), znači da je Rečenica zadovoljila Pitanje i ide dalje na slijedeću Rečenicu pa se cijeli postupak ponavlja od prvog kruga.

Ako je prošao oba kruga, a nije dobio Odgovor, nije pronašao niti jednu Rečenicu koja daje Odgovor na Pitanje (proces 3.1.5. Zaključivanje: ANSWER='I DON'T KNOW'), Odgovor je Answer = 'I DO NOT KNOW' i to upiše u tablicu KB\_ANSWER funkcijom KB\_Write\_Answer ( AnsId, Answer, PrgQ, OrdQ, null, null ).

Kada je obradio jedno Pitanje vraća se na početak, uzima novo pitanje i ponavlja cijeli postupak. Na kraju generiranja Odgovora (proces 3.3. Kompletiranje odgovora), kada je u tablicu KB\_ANSWER upisao samo stupce ANSWER\_ID, ANSWER, PRGQ, ORDQ, PRGS, ORDS, pozove funkciju KB\_Complete\_Answers() koja popuni stupce originalnim, izvornim tekstovima TENG, FNOK, QTENG i QFNOK.

## 7 Primjena sustava temeljenog na NOK metodi

U ovom poglavlju prikazani su rezultati primjene algoritma transformacije na skupu rečenica prirodnog jezika te skupu pitanja prirodnog jezika u sustavu temeljenom na NOK metodi. Izvršena je transformacija rečenica i pitanja u tablice relacijske baze podataka. Osim toga prikazani su i rezultati algoritma generiranja odgovora na postavljena pitanja (na temelju polaznih rečenica).

U Pravitku 2 tablice su grupirane u tri skupine od kojih svaka skupina ima tri vrste tablica:

- tablica koja sadrži rečenice prirodnog jezika (koje predstavljaju znanje) i odgovarajućih FNOK zapisa (koji su nastali transformacijom rečenica prirodnog jezika prema odgovarajućim pravilima),
- tablica koja sadrži pitanja prirodnog jezika nastala na temelju rečenica kojima se propituje postojeće znanje te njihov odgovarajući QFNOK zapis,
- tablica koja prikazuje odgovore sustava na postavljena pitanja i samu analizu tih odgovora u usporedbi s odgovorom koje je dao čovjek.

Sustav je testiran na 103 jednostavne rečenice hrvatskog jezika i 180 pitanja temeljenih na tim jednostavnim rečenicama hrvatskog jezika prikazanih u tablicama 19 i 20 u Pravitku 2. Rezultati testiranja (odgovori na pitanja) prikazani su u tablici 21 u Pravitku 2.

Također, sustav je testiran i na 100 jednostavnih rečenica engleskog jezika te 174 pitanja temeljenih na 100 jednostavnih rečenica engleskog jezika prikazanih u tablicama 22 i 23 u Pravitku 2. Rezultati ovog testiranja (odgovori na pitanja) prikazani su u tablici 24 u Pravitku 2.

Dodatno, sustav je testiran i nad 42 rečenice i 88 pitanja na engleskom jeziku koje su prikazane u tablicama 25 i 26 u Pravitku 2. U tablici 27 u Pravitku 2 prikazani su rezultati testiranja, odnosno dobiveni odgovori na pitanja.

### 7.1 Primjena sustava temeljenog na NOK metodi

Prikažimo dio sadržaja baze podataka opisane u poglavlju 6.3 dobiven nakon transformacije rečenica i pitanja iz Pravitka 2. Prvi redak u tablici predstavlja popis



atributa. Ostali redovi predstavljaju podatke u bazi podataka. Ključni atribut je podvučen punom linijom Prva tablica u relacijskoj bazi podataka sustava temeljenog na NOK metodi je jednostavna i sadrži popis paragrafa: NOK\_PARAGRAPH (PRG, NAME). Njen sadržaj prikazan je u tablici 7.1.

Tablica 7.1. Popis paragrafa

<u>PRG</u>	<u>NAME</u>
2201	životopis Hrvoje Horvat
2202	životopis Jelena Ivić
1	Text 10-01-2016
222	životopisi
333	100 rečenica ENG
444	100 rečenica HJ
111	Katarina

Druga tablica je: NOK\_FNOK\_ALL (PRG, ORD, TENG, FNOK), a prikazana je u tablici 7.2. Ona predstavlja popis rečenica prirodnog jezika i njen FNOK zapis, po paragrafima. Prikažimo samo prvih 5 rečenica iz paragrafa *100 rečenica HJ*.

Tablica 7.2. NOK\_FNOK\_ALL

<u>PRG</u>	<u>ORD</u>	<u>TENG</u>	<u>FNOK</u>
444	1	Danijel vozi.	vozi ("tko?" Danijel)
444	2	Jakov vozi kamion.	vozi ("tko?" Jakov, "što?" kamion)
444	3	Tomislav vozi auto.	vozi ("tko?" Tomislav, "što?" auto)
444	4	Ivan vozi automobil.	vozi ("tko?" Ivan, "što?" automobil)
...	...	...	...

Tri tablice koje predstavljaju rječnik prikazane su u tablicama 7.3, 7.4 i 7.5.

Tablica 7.3. KB\_QUEST

<u>QUEST_ID</u>	<u>QUEST</u>
1	gdje
2	kada
3	kakav
4	kako
5	kakva
...	...

Tablica 7.4. KB\_DICT

<u>DICT_ID</u>	<u>DICT</u>	<u>DICT_PAR</u>
151	vidjeti	151
126	vidim	151
141	biti	141
142	kupiti	142
128	voli	152
...	...	...

Vanjski ključ je podvučen isprekidanom linijom. Vanjski ključ DICT\_PAR nastao je prevođenjem povratne veze CONTEXT sa slike 6.2.

Tablica 7.5. KB\_WORD

<u>WORD_ID</u>	WORD	<u>DICT_ID</u>
51	auto	169
52	autocesti	
53	automobil	168
54	bicikl	
55	bilježnicu	

NOK\_FNOK je tablica nastala transformacijom rečenice iz tablice NOK\_FNOK\_ALL, a prikazana je u tablici 7.6. Za svaki čvor u rečenici nastaje jedan redak tablice i taj se redak veže za početnu rečenicu i paragraf u kome se nalazi. Ova tablica je popis čvorova u rečenici i oznaka hijerarhijski nadređenog čvora (PARENT\_ID). Svaki redak tablice se veže za rječnik u kome se ta riječ nalazi s njenom jedinstvenom semantikom. Redak tablice se veže i s ključem nadređene riječi u rječniku. Na kraju se redak tablice povezuje i s ključem pitanja koje je postavljeno u polaznoj rečenici. Na ovaj način se riječima u polaznim rečenicama određuje jedinstveno značenje iz rječnika. Procesni čvorovi u zapisu nemaju nadređenog čvora te im je vrijednost PARENT\_ID = 0.

Tablica 7.6. NOK\_FNOK - popis čvorova

<u>PRG</u>	<u>ORD</u>	<u>NODE_ID</u>	NODE	QUEST	<u>PARENT_ID</u>	<u>KB_PAR_WRD_ID</u>	KB_PAR_W RD	<u>KB_NOD_WRD_ID</u>	<u>KB_NOD_QST_ID</u>
444	6	1	vozi	_	0			280	
444	6	2	Katarina	tko	1	280	vozi	27	25
444	6	3	auto	što	1	280	vozi	51	26
444	6	4	crveni	koji	3	51	auto	72	19
444	6	5	autocesti	gdje	1	280	vozi	52	1
444	6	6	na	gdje	5	52	autocesti	153	1

Konačna tablica KB\_TEXT\_QUEST transformacija početnih FNOK zapisa u relacijsku bazu podataka prikazana je u tablici 7.7. Svaka veza čvorova iz FNOK zapisa postaje jedan redak u ovoj tablici i ima svoj jedinstveni ključ ID u bazi podataka. Preko vanjskih ključeva povezan je s izvornom rečenicom kojoj pripada te s dvije riječi u rječniku i jednim pitanjem.

Tablica 7.7. KB\_TEXT\_QUEST - popis veza

<u>ID</u>	<u>PRG</u>	<u>ORD</u>	<u>PAR_WRD_ID</u>	<u>NOD_WRD_ID</u>	<u>NOD_QST_ID</u>	<u>PAR_WRD</u>	<u>NOD_WRD</u>	<u>NOD_QST</u>
8	444	5	280	46	25	vozi	Vedran	tko
9	444	5	46	204	19	Vedran	prijatelj	koji
10	444	5	204	147	32	prijatelj	moj	čiji

11	444	5	280	51	26	vozi	auto	što
12	444	6	280	27	25	vozi	Katarina	tko
13	444	6	280	51	26	vozi	auto	što
14	444	6	51	72	19	auto	crveni	koji
15	444	6	280	52	1	vozi	autocesti	gdje
16	444	6	52	153	1	autocesti	na	gdje
...								

Tri zadnja prekrižena atributa PAR\_WRD, NOD\_WRD i NOD\_QST nisu atributi ove tablice, ali smo ih dodali radi lakšeg praćenja sadržaja tablice. Jedna rečenica je razbijena u onoliko redaka koliko ona sadrži uloga. Pitanje (ulogu) postavlja čvor Parent, a odgovor na to pitanje je sam čvor Node.

Tablica NOK\_QFNOK\_ALL (PRG, ORD, QTENG, QFNOK) prikazana je u tablici 7.8. Ona predstavlja popis pitanja prirodnog jezika i njihov QFNOK zapis po paragrafima. Prikažimo samo prvih 5 pitanja.

Tablica 7.8. NOK\_QFNOK\_ALL

PRG	ORD	QTENG	QFNOK
444	1	Tko jede?	jede ("tko?" X)
444	2	Tko pohlepno jede?	jede ("tko?" X, _ pohlepno)
444	3	Tko jede čokoladu?	jede ("tko?" X, _ čokoladu)
444	4	Kako medvjedi jedu?	jedu ( _ medvjedi, "kako?" X)
...	...	...	...

NOK\_QFNOK je tablica nastala transformacijom pitanja iz tablice NOK\_QFNOK\_ALL a prikazana je u tablici 7.9. Za svaki čvor u pitanju nastaje jedan redak tablice i taj se redak veže za početno pitanje i paragraf u kome se nalazi. Ova tablica je popis čvorova u pitanjima i oznaka hijerarhijski nadređenog čvora (PARENT\_ID). Svaki redak tablice se veže za rječnik u kome se ta riječ nalazi s njenom jedinstvenom semantikom. Redak tablice se veže i s ključem nadređene riječi u rječniku. Na kraju se redak tablice povezuje i s ključem pitanja koje je postavljeno u polaznoj rečenici. Na ovaj način se riječima u polaznim pitanjima određuje jedinstveno značenje iz rječnika. Procesni čvorovi u zapisu nemaju nadređenog čvora te im je vrijednost PARENT\_ID = 0. Čvor koji tražimo označen je sa X.

Tablica 7.9. NOK\_QFNOK - popis čvorova

PRG	ORD	NODE ID	NODE	QUEST	PARENT ID	KB_PAR_WRD ID	KB_PAR_WRD	KB_NOD_WRD ID	KB_NOD_QST ID
444	1	1	jede	_	0			115	
444	1	2	X	tko	1	115	jede		25
444	2	1	jede	_	0			115	

444	2	2	X	tko	1	115	jede		25
444	2	3	pohlepno	_	1	115	jede	200	
444	3	1	jede	_	0			115	
444	3	2	X	tko	1	115	jede		25
...									

Konačna tablica KB\_QUEST\_QUEST, koja je transformacija početnih QFNOK zapisa u relacijsku bazu podataka, prikazana je u tablici 7.10. Svaka veza čvorova iz QFNOK zapisa postaje jedan redak u ovoj tablici i ima svoj jedinstveni ključ ID u bazi podataka. Preko vanjskih ključeva povezan je s izvornim pitanjem kojem pripada te s dvije riječi u rječniku i jednim pitanjem.

Tablica 7.10. KB\_QUEST\_QUEST – popis veza

ID	PRG	ORD	PAR_WRD_ID	NOD_WRD_ID	NOD_QST_ID	PAR_WRD	NOD_WRD	NOD_QST
1	444	1	115		25	jede	X	tko
2	444	2	115		25	jede	X	tko
3	444	2	115	200		jede	pohlepno	_
4	444	3	115		25	jede	X	tko
5	444	3	115	301		jede	čokoladu	_
6	444	4	116	141		jedu	medvjedi	_
...								

Tri zadnja prekrižena atributa PAR\_WRD, NOD\_WRD i NOD\_QST nisu atributi ove tablice, ali smo ih dodali radi lakšeg praćenja sadržaja tablice. Jedno pitanje je razbijeno u onoliko redaka koliko ono sadrži uloga. Pitanje (ulogu) postavlja čvor Parent, a odgovor na to pitanje je sam čvor Node.

Kada popunimo tablice KB\_TEXT\_QUEST i KB\_QUEST\_QUEST podacima (napunjene sa svim Paragrafima koje želimo i njihovim rečenicama (FNOK zapisi) i pitanjima (QFNOK zapisi), možemo popuniti tablicu odgovora (tablica 7.11). Pogledajmo strukturu:

KB\_ANSWER (ANSWER\_ID, PRGQ, ORDQ, QTENG, ANSWER, PRGS, ORDS, TENG, QFNOK, FNOK)

Tablica 7.11. KB\_ANSWER

ANSWER_ID	PRGQ	ORDQ	QTENG	ANSWER	PRGS	ORDS	TENG	QFNOK	FNOK
1	444	1	Tko jede?	medvjedi	444	53	Medvjedi pohlepno jedu.	jede ("tko?" X)	jedu ("tko?" medvjedi, "kako?" pohlepno)

2	444	1	Tko jede?	Luka	444	88	Luka jede sendvič.	jede ("tko?" X)	jede ("tko?" Luka, "što?" sendvič)
3	444	1	Tko jede?	studenti	444	89	Studenti jedu čokoladu.	jede ("tko?" X)	jedu ("tko?" studenti, "što?" čokoladu)
4	444	2	Tko pohlepno jede?	medvjedi	444	53	Medvjedi pohlepno jedu.	jede ("tko?" X, _ pohlepno)	jedu ("tko?" medvjedi, "kako?" pohlepno)
5	444	3	Tko jede čokoladu?	studenti	444	89	Studenti jedu čokoladu.	jede ("tko?" X, _ čokoladu)	jedu ("tko?" studenti, "što?" čokoladu)
6	444	4	Kako medvjedi jedu?	pohlepno	444	53	Medvjedi pohlepno jedu.	jedu ( _ medvjedi, "kako?" X)	jedu ("tko?" medvjedi, "kako?" pohlepno)
7	444	5	Jede li Luka sendvič?	YES	444	88	Luka jede sendvič.	jede ( ( & li), _ Luka, _ sendvič)	jede ("tko?" Luka, "što?" sendvič)

Radi lakšeg praćenja i snalaženja po tablici, u tablica KB\_ANSWER su dodana polja QTENG, TENG, QFNOK i FNOK. Popunjavanjem tablice 7.11 dobili smo odgovore na postavljena pitanja prirodnog jezika koji su pronađeni u rečenicama prirodnog jezika.

## 7.2 Analiza rezultata

Nakon primjene skupa pitanja na znanje iskazano rečenicama prirodnog jezika dobiveni su odgovori na postavljena pitanja. Sustav je odgovore na postavljena pitanja tražio unutar teksta iskazanog rečenicama prirodnog jezika.

Svaki odgovor se analizirao u ovisnosti o potpunosti i ispravnosti samog odgovora te usporedio s odgovorom kojeg bi dao čovjek. Odgovori su kategorizirani u nekoliko kategorija.

1. **Netočan odgovor** – Odgovor koji je sustav dao ne temelji se na pohranjenom znanju, odnosno pogrešan skup riječi je korišten u odgovoru.
  - Primjer 1: za pitanje *Što Petar radi?* sustav je, između ostalog, odgovorio *ima*. Međutim, ovaj odgovor ne možemo prihvatiti kao točan.
  - Primjer 2: sustav nije pronašao odgovor, iako je trebao i kao odgovor javlja *I do not know*.

2. **Nedostaje odgovor** – Sustav nije dao odgovor, a odgovor je bio očekivan.
  - Primjer: Na pitanje *Kakav je otac?* sustav odgovara *star* na temelju rečenice *Marijin otac je star*. Međutim, u sustavu postoji još jedna rečenica koja je mogla poslužiti u pronalasku odgovora na pitanje: *Otac je dobar čovjek*. Sustav je na polazno pitanje trebao odgovoriti i *dobar čovjek*, odnosno na ovo pitanje trebao je dati dva odgovora, a ne samo jedno.
3. **Semantički nepotpun odgovor** – Sustav djelomično (nepotpuno) odgovara na pitanje, odnosno sustav pronalazi izoliran, djelomičan odgovor, izvučen iz ispravnog znanja. Međutim, bilo bi prirodnije da je odgovor na postavljeno pitanje potpuniji.
  - Primjer 1: Za pitanje *Što je London?* na temelju znanja iz rečenice *London je glavni grad Velike Britanije.*, sustav odgovara *glavni grad*. Ovaj odgovor je točan, međutim potpuniji odgovor bio bi *glavni grad Velike Britanije*.
  - Primjer 2: Za pitanje *Kada Luka radi svoju domaću zadaću?* na temelju znanja iz rečenice *Luka radi svoju domaću zadaću svaki dan u sedam sati.*, sustav odgovara *svaki dan*. Ovaj odgovor je točan, međutim potpuniji odgovor bio bi *svaki dan u sedam sati*.
  - Primjer 3: Za pitanje (QTENG) *To whom does Julia write a letter?* na temelju znanja iz TENG *Julia writes a letter to a friend.* sustav odgovara *friend*; Ovaj odgovor je točan, međutim poželjno bi bilo da je sustav odgovorio *a friend* ili čak *to a friend* što je više u duhu jezika.
4. **Semantički ispravan, ali gramatički neispravan odgovor** – sustav daje odgovor temeljen na ispravnom znanju, ali odgovor ne odgovara pravilima gramatike.
  - Primjer 1: Na pitanje *Što je ukradeno?* sustav odgovara *njegovog oca automobil*. Potpuno točan odgovor bio bi *Automobil njegovog oca*.
  - Primjer 2: Na pitanje *Što je brzo?* sustav odgovara *za Zagreb vlak*. Potpuno točan odgovor bio bi *Vlak za Zagreb*.
  - Primjer 3: Na pitanje *Whose car drives?* sustav odgovara *my*. Potpuno točan odgovor bio bi *my car* ili *mine*.
5. **Točan odgovor** – Odgovor koji je sustav dao je točan.

Ako bi odgovore podijelili u dvije klase, mogli bismo reći da su pod 1. i 2. kategorijom netočni odgovori. Odgovori 3., 4. i 5. kategorije su točni odgovori, s time da bi algoritam za formatiranje odgovora trebalo doraditi. Dorada algoritma za formatiranje odgovora nije bio cilj ovog rada.

### 7.2.1 Rezultati za hrvatski jezik

Sustav je testiran na 103 jednostavne rečenice hrvatskog jezika (skraćeno THJ) koje su predstavljale skup ulaznog znanja. Podsjetimo, pod jednostavnim rečenicama hrvatskog jezika podrazumijevamo izjavne rečenice koje imaju samo jedan predikat te izrečen subjekt u nominativu. Uz rečenice u sadašnjem glagolskom vremenu, u skup rečenica nad kojim je sustav testiran stavili smo nekoliko rečenica u prošlom i budućem glagolskom vremenu te imperativne rečenice. Također, uvedene su i rečenice koje nemaju izrečen subjekt ili se on podrazumijeva.

Na temelju 103 THJ rečenice, postavljeno je 180 pitanja prirodnog jezika (skraćeno QHJ) kojim se propituje ulazno znanje. Sve THJ rečenice i sva QHJ pitanja prikazani su u tablici 19, odnosno tablici 20 u Privitku 2. U tablici 7.12 prikazat ćemo samo dio tablice 19 iz Privitka 2, a u tablici 7.13 dio tablice 20 iz Privitka 2. Svi THJ i QHJ (tj. rečenice i pitanja) pretvoreni su u FNOK, odnosno QFNOK zapise.

Tablica 7.12. Isječak THJ rečenica i njihovih pripadnih FNOK-ova

RBR	THJ	FNOK
53	Medvjedi pohlepno jedu.	jedu ("tko?" medvjedi, "kako?" pohlepno)
88	Luka jede sendvič.	jede ("tko?" Luka, "što?" sendvič)
89	Studenti jedu čokoladu.	jedu ("tko?" studenti, "što?" čokoladu)

Tablica 7.13. Isječak THJ pitanja i njihovih pripadnih QFNOK-ova

RBR	QHJ	QFNOK
1	Tko jede?	jede ("tko?" X)
2	Tko pohlepno jede?	jede ("tko?" X, _ pohlepno)
3	Tko jede čokoladu?	jede ("tko?" X, _ čokoladu)
4	Kako medvjedi jedu?	jedu ( _ medvjedi, "kako?" X)
5	Jede li Luka sendvič?	jede ( ( & li), _ Luka, _ sendvič)

Na temelju 103 THJ rečenice te 180 QHJ pitanja nastala su 222 odgovora jer je na neko pitanje bilo moguće odgovoriti s dva ili više odgovora. U tablici 21. prikazani su svi rezultati dobiveni sustavom, a u tablici 7.14 izdvojeni su samo neki dijelovi te tablice grupirani po kategoriji odgovora.

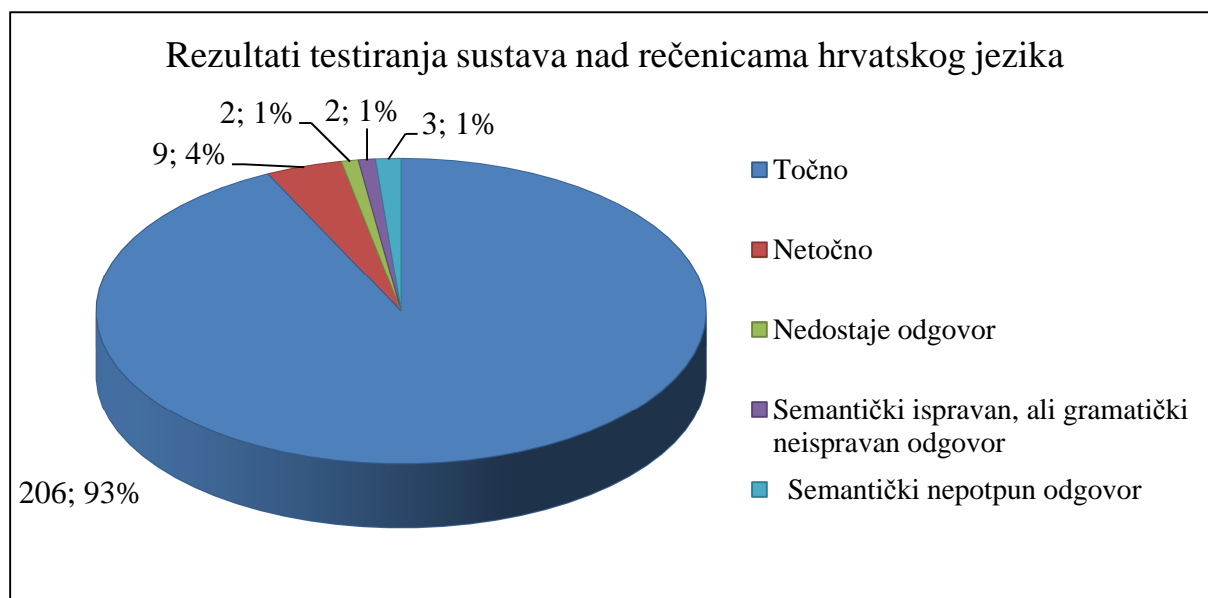
Tablica 7.14. Isječak odgovora sustava

ANS WER _ID	PR GQ	OR DQ	QHJ	Očekivani odgovor	ANSWER sustava	Analiza odgovora	PR GS	OR DS
<b>Netočni odgovori</b>								
19	444	16	Kakve knjige ima Maja?	zanimljive	I DO NOT KNOW	Netočno		
29	444	26	Jesu li natjecanja u zadnjem tjednu svibnja?	NE	I DO NOT KNOW	Netočno		
47	444	40	Kakvi su moji prijatelji?	dobri dečki	I DO NOT KNOW	Netočno		
116	444	90	Što ja trebam?	mnogo novca	I DO NOT KNOW	Netočno		
199	444	165	Što Petar radi?		je	Netočno	444	15
200	444	165	Što Petar radi?		ima	Netočno	444	62
209	444	174	Što studenti rade?		su	Netočno	444	49
212	444	174	Što studenti rade?		je	Netočno	444	95
19	444	16	Kakve knjige ima Maja?	zanimljive	I DO NOT KNOW	Netočno		
<b>Nedostaje odgovor</b>								
146	444	118	Kakav je stol?	kameni		Nedostaje odgovor		
186	444	154	Kakav je otac?	dobar čovjek		Nedostaje odgovor		
<b>Semantički nepotpun odgovor</b>								
75	444	64	Kada Luka radi svoju domaću zadaću?	svaki dan u sedam sati	svaki dan	Semantički nepotpun odgovor	444	83
138	444	112	Tko je Vedrana?	petnaesta dobitnica nagrade od 1999 godine	petnaesta dobitnica	Semantički nepotpun odgovor	444	50
177	444	145	Što je London?	glavni grad Velike Britanije	glavni grad	Semantički nepotpun odgovor	444	69
<b>Semantički ispravan, ali gramatički neispravan odgovor</b>								
136	444	110	Što je ukradeno?	Automobil njegovog oca	njegovog oca automobil	Semantički ispravan, ali gramatički neispravan odgovor	444	92
180	444	148	Što je brzo?	Vlak za Zagreb	za Zagreb vlak	Semantički ispravan, ali gramatički neispravan odgovor	444	60
<b>Točni odgovori</b>								
1	444	1	Tko jede?	Medvjedi	medvjedi	Točno	444	53
2	444	1	Tko jede?	Luka	Luka	Točno	444	88
3	444	1	Tko jede?	Studenti	studenti	Točno	444	89
4	444	2	Tko pohlepno jede?	Medvjedi	medvjedi	Točno	444	53
5	444	3	Tko jede čokoladu?	Studenti	studenti	Točno	444	89
6	444	4	Kako medvjedi jedu?	pohlepno	pohlepno	Točno	444	53



198	444	165	Što Petar radi?	vozi bratov auto	vozi	Točno	444	8
210	444	174	Što studenti rade?	razgovaraju o rješenju	razgovaraju	Točno	210	444
211	444	174	Što studenti rade?	jedu čokoladu	jedu	Točno	211	444
213	444	174	Što studenti rade?	igraju košarku	igraju	Točno	213	444
214	444	174	Što studenti rade?	pišu seminarski rad	pišu	Točno	444	98

Answer\_ID je redni broj odgovora, PRGQ i ORDQ su broj paragrafa i broj pitanja iz tablice 20 iz Privitka 2, PRGS i ORDS su broj paragrafa i broj rečenice na temelju koje je odgovor proizašao (tablica 19 u Privitku 2). QHJ je pitanje zadano u obliku prirodnog hrvatskog jezika i ovdje je ubačen radi lakše provjere rezultata. Očekivani odgovor je odgovor na pitanje prirodnog jezika kojeg je dao čovjek na temelju ulaznog znanja (jednostavnih rečenica prirodnog jezika), ANSWER sustava je odgovor koji je dao sustav, a Analiza odgovora je kategorizacija odgovora u jednu od ranije navedenih kategorija. Slika 7.1 prikazuje rezultate testiranja sustava temeljenog na NOK-u nad rečenicama i pitanjima na hrvatskom jeziku.



**Slika 7.1. Rezultati testiranja sustava nad rečenicama hrvatskog jezika**

Sustav je netočno odgovorio u 9 od 222 slučaja. Svi netočni odgovori prikazani su u tablici 7.14 u dijelu Netočni odgovori. Pod netočnim odgovorom podrazumijevamo onaj odgovor sustava koji se ne temelji na pohranjenom znanju, odnosno ako je u odgovoru

korišten pogrešan skup riječi. Također, pod netočnim odgovorima sustava podrazumijevamo i one odgovore kada odgovori nisu pronađeni iako su trebali, odnosno kada sustav kao odgovor javlja *I DO NOT KNOW*.

Sustav odgovara *I DO NOT KNOW* u sljedećim slučajevima:

- Za pitanje 16. *Kakve knjige ima Maja?* očekivani odgovor je *zanimljive*, nastao na temelju znanja u rečenici 67. *Maja ima nekoliko zanimljivih knjiga*. Usporedbom pripadnih QFNOK-ova i FNOK-ova ( *ima ( \_ Maja, \_ knjige ("kakve?" X)* ) i *ima ("tko?" Maja, "što?" knjiga ("koliko?" nekoliko, "kakvih?" zanimljivih)* ) te rezultata u tablicama KB\_DICT, KB\_WORD i KB\_QUESTION, zaključujemo da u KB\_QUESTION-u uloge (pitanja) nisu uređeni hijerarhijski, niti povezani sa KB\_DICT, što bi u ovom slučaju bilo korisno zbog povezivanja uloga *kakvih* i *kakve* te bi sustav mogao dati točan odgovor.
- Za pitanje 40. *Kakvi su moji prijatelji?* očekivani odgovor je *dobri dečki*, nastao na temelju znanja u rečenici 27. *Svi su moji prijatelji dobri dečki*. Usporedbom pripadnih FNOK i QFNOK zapisa ( *su ("tko?" prijatelji ("koji?" svi, "čiji?" moji), "što?" dečki ("kakvi?" dobri)* ) i *su ( \_ prijatelji ( \_ moji), "kakvi?" X* ) vidimo da se odgovori na pitanje *kakvi?* nalaze na različitim razinama u hijerarhiji, stoga bi trebalo omogućiti uspoređivanje čvorova na različitim razinama hijerarhije.
- Na pitanja 90. *Što ja trebam?*, QFNOK: *trebam ( \_ ja, "što?" X)* i 180. *Tko treba mnogo novaca?*, QFNOK: *treba ("tko?" X, \_ novaca ( \_ mnogo))* postavljenih za provjeru znanja u rečenici 23. *Trebam mnogo novca*. (FNOK: *trebam ("što?" novaca ("koliko?" mnogo))* ) sustav nije odgovorio *mного novca* i *ja*. Razlog zašto je sustav odgovorio *I DO NOT KNOW* je u skrivenom subjektu koji u polaznoj rečenici nije eksplicitno izražen. Samim time nije naveden u FNOK-u, a u prvom pitanju i pripadnom QFNOK zapisu *ja* je naveden. U drugom pitanju tražimo upravo ovaj subjekt koji nije istaknut. Uvođenjem kontekstnih znanja ovaj problem bi se riješio.
- Na pitanje 26. *Jesu li natjecanja u zadnjem tjednu svibnja?* sustav odgovara *I DO NOT KNOW* umjesto očekivanog odgovora *NE*. Razlog za netočan odgovor je u nedostatku kontekstnog znanja. Sustav ima znanje na temelju rečenice 81. *Natjecanja su u zadnjem tjednu travnja*. ali trebalo bi doraditi algoritam za provjeru antonima, hiponima, hiperonima i slično kojima bi se mogle provjeravati

varijacije riječi npr. travanj – svibanj, prvi – zadnji i slično te na temelju toga dati odgovor *NE*.

- Na pitanja 165. *Što Petar radi?* i 174. *Što studenti rade?* sustav između ostalog, odgovara *je, ima, odnosno su, je*. Međutim, ove odgovore ne možemo prihvatiti kao točne jer su nepotpuni i ne kazuju radnju subjekta.

Sustav u 2 od 222 slučaja nije dao nikakav odgovor, iako je odgovor bio očekivan i mogao se pronaći u polaznom znanju. Slučajevi kad nedostaje odgovor prikazani su u tablici 7.14 u dijelu Nedostaje odgovor.

- Na pitanje 118. *Kakav je otac?* sustav odgovara *star* na temelju rečenice 74. *Marijin otac je star*. Međutim, u sustavu postoji još jedna rečenica koja je mogla poslužiti u pronalasku odgovora na pitanje: 72. *Otac je dobar čovjek*. Sustav je na polazno pitanje trebao odgovoriti i *dobar čovjek*, odnosno na ovo pitanje trebao je dati dva odgovora, a ne samo jedno. Razlog zbog kojeg sustav nije dao i drugi odgovor na ovo pitanje je u različitim razinama hijerarhije na kojima se uloga *kakav?* i čvor koji je odgovor na pitanje *kakav* nalaze.
- Slična situacija je i s pitanjem 118. *Kakav je stol?* Sustav na temelju rečenice 13. *Stol je drven*. odgovara *drven*. Međutim, na temelju rečenice 11. *To je kameni stol*. sustav je trebao odgovoriti i *kameni*. Ovdje je opet razlika u razinama hijerarhije.

Sustav je dao semantički nepotpun odgovor u 3 od 222 slučaja. Svi takvi odgovori prikazani su u tablici 7.14 u dijelu Semantički nepotpun odgovor. Kod takve vrste odgovora, sustav djelomično (nepotpuno) odgovara na pitanje. On pronalazi izoliran, djelomičan odgovor koji je izvučen iz ispravnog znanja. Međutim, bilo bi prirodnije da je odgovor na postavljeno pitanje potpuniji, opširniji.

- Na pitanje 64. *Kada Luka radi svoju domaću zadaću?* na temelju znanja iz rečenice 83. *Luka radi svoju domaću zadaću svaki dan u sedam sati.*, sustav odgovara *svaki dan*. Ovaj odgovor je točan, međutim potpuniji odgovor bio bi *svaki dan u sedam sati*.
- Na pitanje 112. *Tko je Vedrana?* na temelju znanja iz rečenice 50. *Vedrana je petnaesta dobitnica nagrade od 1999 godine.* sustav je odgovorio *petnaesta dobitnica*. Ovaj odgovor je točan, međutim potpuniji odgovor bio bi *petnaesta dobitnica nagrade od 1999 godine*.

- Na pitanje 145. *Što je London?* na temelju znanja iz rečenice 69. *London je glavni grad Velike Britanije.*, sustav odgovara *glavni grad*. Ovaj odgovor je točan, međutim potpuniji odgovor bio bi *glavni grad Velike Britanije*.

Sustav je dao semantički ispravan, ali gramatički neispravan odgovor u 2 od 222 slučaja. Svi takvi odgovori prikazani su u tablici 7.14 u dijelu Semantički ispravan, ali gramatički neispravan odgovor. Kod takvih odgovora, sustav daje odgovor temeljen na ispravnom znanju, ali odgovor ne odgovara pravilima gramatike.

- Na pitanje 110. *Što je ukradeno?* sustav odgovara *njegovog oca automobil*. Potpuno točan odgovor bio bi *Automobil njegovog oca*.
- Na pitanje 148. *Što je brzo?* sustav odgovara *za Zagreb vlak*. Potpuno točan odgovor bio bi *Vlak za Zagreb*.

Sustav je točno odgovorio u 206 od 222 slučaja. Neki od tih točnih odgovora prikazani su u tablici 7.14 u dijelu Točni odgovori. Kao točni odgovori prihvaćani su odgovori koji su u potpunosti identični odgovoru čovjeka. Osim toga, kod pitanja tipa "Što netko radi?" kao točan odgovor vrednovao se odgovor koji se sastojao samo od procesnog čvora (na primjer, čita, piše, crta, vozi i slično). Čovjek može svojim znanjem na temelju polaznih rečenica na takve odgovore odgovoriti detaljnije, na primjer, čita knjigu, piše roman, crta drvenim bojicama, vozi crven auto ili ponekad dati odgovor cijelom rečenicom. Zbog toga se za točan odgovor uzimao onaj odgovor koji zadovoljava semantičku i gramatičku ispravnost (sastoji se od jedne ili više riječi). Daljnjom doradom sustava planira se omogućiti sustavu detaljnije odgovaranje na pitanja i u više varijanti.

Na pojedino pitanje moguće je odgovoriti kraćim ili dužim odgovorom pa čak i cijelom rečenicom. Nama je dovoljno da je zadovoljen minimum odgovora, ali da je odgovor semantički i gramatički korektan.

Svi ovi uočeni problemi ostaju za doradu algoritama i budući rad.

### **7.2.2 Rezultati testiranja sustava rečenicama engleskog jezika**

Sustav je testiran i na 100 jednostavnih rečenica engleskog jezika (skraćeno TENG) koje su predstavljale skup ulaznog znanja Uz rečenice u sadašnjem glagolskom vremenu, u skup rečenica nad kojim je sustav testiran stavili smo nekoliko rečenica u Simple past tense, Future simple tense, imperativne rečenice te rečenice s modalnim glagolima.

Na temelju 100 TENG rečenica, postavljeno je 174 pitanja prirodnog jezika (skraćeno QTENG) kojim se propituje ulazno znanje. Sve TENG rečenice i sva QTENG pitanja prikazani su u tablici 22, odnosno 23 u Pravitku 2.

U tablici 7.15 prikazat ćemo samo dio tablice 22 iz Pravitka 2, a u tablici 7.16 dio tablice 23 iz Pravitka 2. Svi TENG i QTENG (tj. rečenice i pitanja) pretvoreni su u FNOK, odnosno QFNOK zapise.

Tablica 7.15. Isječak TENG rečenica i njihovih pripadnih FNOK-ova

RBR	TENG	FNOK
53	The bears eat greedily.	eat ("who?" bears ("art?" the), "how?" greedily)
88	Lucas eats a sandwich.	eats ("who?" Lucas, "what?" sandwich ("art?" a))
89	Students eat chocolate.	eat ("who?" students, "what?" chocolate)

Tablica 7.16. Isječak QTENG pitanja i njihovih pripadnih QFNOK-ova

RBR	QTENG	QFNOK
1	Who eats?	eats ("who?" X)
2	Who eats greedily?	eats ("who?" X, _ greedily)
3	Who eats chocolate?	eats ("who?" X, _ chocolate)
4	How do the bears eat?	eat ( (& do) ( _ bears ( _ the), "how?" X))
5	Does Lucas eat a sandwich?	eat ( (& does) ( _ Lucas, _ sandwich ( & a)))

Na temelju 100 TENG rečenica te 174 QTENG pitanja nastala su 222 odgovora jer je na neko pitanje bilo moguće odgovoriti s dva ili više odgovora. Tablica 24 iz Pravitka 2 prikazuje sve rezultate dobivene sustavom, a u tablici 7.17 izdvojeni su samo neki dijelovi te tablice grupirani po kategoriji odgovora.

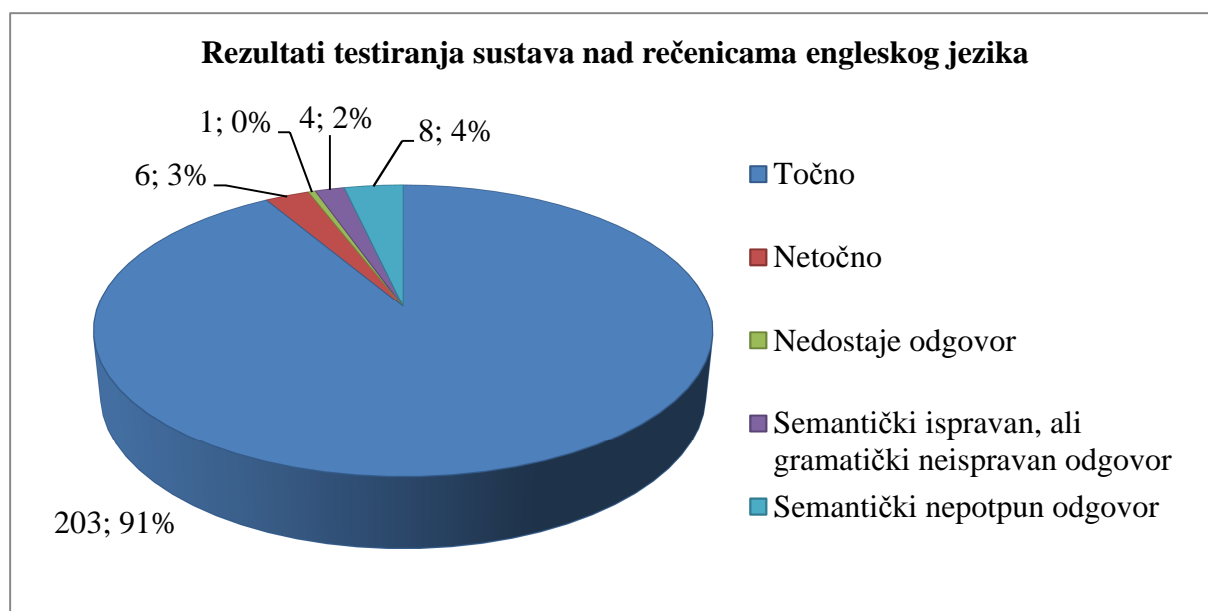
Tablica 7.17. Isječak odgovora sustava

ANS WER_ ID	PR GQ	OR DQ	QTENG	Očekivani odgovor	ANSWER sustava	Analiza odgovora	PR GS	OR DS
<b>Netočni odgovori</b>								
29	333	26	Are competitions in the last week of May?	NO	I DO NOT KNOW	Netočno		
205	333	165	What does Peter do?		is	Netočno	333	15
206	333	165	What does Peter do?		has	Netočno	333	62
207	333	165	What does Peter do?		younger	Netočno	333	79
216	333	174	What do the students do?		are	Netočno	333	49
219	333	174	What do the students do?		are	Netočno	333	95
<b>Nedostaje odgovor</b>								
170	333	134 a	What kind of a table is it?	wooden		Nedostaje odgovor		
<b>Semantički nepotpun odgovor</b>								
75	333	64	When does Lucas do	every day at	at seven	Semantički nepotpun	333	83

			his homework?	seven o'clock.	o'clock	odgovor		
104	333	78	Who did Noa give this book to?	To a friend	friend	Semantički nepotpun odgovor	333	66
105	333	79	Where we don't sleep?	in the hotel Intercontinental	in the hotel	Semantički nepotpun odgovor	333	38
106	333	80	Where we do not sleep?	in the hotel Intercontinental	in the hotel	Semantički nepotpun odgovor	333	38
135	333	107	What do the students talk about?	about the solution	solution	Semantički nepotpun odgovor	333	58
140	333	112	What was Vanessa?	the fifteenth person to win the award since 1999	to win the award	Semantički nepotpun odgovor	333	50
176	333	141	What does Lucas do?	does his homework	does	Semantički nepotpun odgovor	333	83
181	333	145	What is London?	the capital of Great Britain	of Great Britain	Semantički nepotpun odgovor	333	69
184	333	148	What is fast?	The train to New York	the train	Semantički nepotpun odgovor	333	60
<b>Semantički ispravan, ali gramatički neispravan odgovor</b>								
45	333	38	Whose friends are good boys?	mine / my friends	my	Semantički ispravan, ali gramatički neispravan odgovor	333	27
174	333	139	What is dinner?	Lucas's favourite meal	favourite Lukas's meal	Semantički ispravan, ali gramatički neispravan odgovor	333	91
188	333	151	What is Tom?	younger than Peter	than Peter younger	Semantički ispravan, ali gramatički neispravan odgovor	333	79
214	333	172	Whose cat is this?	yours / your cat	your	Semantički ispravan, ali gramatički neispravan odgovor	333	34
<b>Točni odgovori</b>								
1	333	1	Who eats?	The bears	the bears	Točno	333	53
2	333	1	Who eats?	Lucas	Lucas	Točno	333	88
3	333	1	Who eats?	Students	students	Točno	333	89
4	333	2	Who eats greedily?	The bears	the bears	Točno	333	53
5	333	3	Who eats chocolate?	Students	students	Točno	333	89
6	333	4	How do the bears eat?	greedily	greedily	Točno	333	53

7	333	5	Does Lucas eat a sandwich?	YES	YES	Točno	333	88
19	333	16	What kind of books does Maya have?	interesting books	interesting	Točno	333	67
59	333	50	How many pounds of potatoes does Thomas buy?	10 pounds	10	Točno	333	40
63	333	54	What can you do?	buy anything	buy	Točno	333	39
64	333	54	What can you do?	touch sister's notebook	touch	Točno	333	85
77	333	66	What does Joseph draw with?	With wooden crayons	crayons	Točno	333	9
98	333	73	What kind of a car does Catherine drive?	a red car	red	Točno	333	6
143	333	115	How many times this year did Monica go to Italy?	two times	two	Točno	333	51
177	333	141	What does Lucas do?	eats a sandwich	eats	Točno	333	88
217	333	174	What do the students do?	talk about the solution	talk	Točno	333	58
218	333	174	What do the students do?	eat chocolate	eat	Točno	333	89
220	333	174	What do the students do?	play basketball every Saturday	play	Točno	333	96
221	333	174	What do the students do?	individually write a seminar paper	write	Točno	333	98

Answer\_ID je redni broj odgovora, PRGQ i ORDQ su broj paragrafa i broj pitanja iz tablice 23 iz Privitka 2, PRGS i ORDS su broj paragrafa i broj rečenice na temelju koje je odgovor proizašao (iz tablice 22 iz Privitka 2). QTENG je pitanje zadano u obliku prirodnog engleskog jezika i ovdje je ubačen radi lakše provjere rezultata. Očekivani odgovor je odgovor na pitanje prirodnog jezika kojeg je dao čovjek na temelju ulaznog znanja (jednostavnih rečenica prirodnog jezika), ANSWER sustava je odgovor koji je dao sustav, a Analiza odgovora je kategorizacija odgovora u jednu od ranije navedenih kategorija. Slika 7.2 prikazuje rezultate testiranja sustava temeljenog na NOK-u nad rečenicama i pitanjima na engleskom jeziku.



**Slika 7.2. Rezultati testiranja sustava nad rečenicama engleskog jezika**

Sustav je netočno odgovorio u 6 od 222 slučaja. Svi netočni odgovori prikazani su u tablici 7.17 u dijelu Netočni odgovori. Pod netočnim odgovorom podrazumijevamo onaj odgovor sustava koji se ne temelji na pohranjenom znanju, odnosno ako je u odgovoru korišten pogrešan skup riječi. Također, pod netočnim odgovorima sustava podrazumijevamo i one odgovore kada odgovori nisu pronađeni, iako su trebali biti, odnosno kada sustav kao odgovor javlja *I DO NOT KNOW*.

- Sustav odgovara *I DO NOT KNOW* u slučaju pitanja broj 26. *Are competitions in the last week of May?* Kao i za isto pitanje na hrvatskom jeziku, tako i ovdje očekujemo da će sustav odgovoriti *NO* jer *Competitions are in the last week of April*. Razlog za netočan odgovor je u nedostatku kontekstnog znanja. Sustav ima znanje na temelju rečenice 81. *Competitions are in the last week of April*, ali bi trebalo doraditi algoritam za provjeru antonima, hiponima, hiperonima i slično kojima bi se mogle provjeravati varijacije riječi npr. *april – may, the first – the last* i slično te na temelju toga dati odgovor *NE*
- Na pitanja 165. *What does Peter do?* i 174. *What do the students do?* sustav između ostalog, odgovara *is, has*, odnosno *younger, are, are*. Međutim, ove odgovore ne možemo prihvatiti kao točne jer su nepotpuni i ne kazuju radnju subjekta, a u slučaju odgovora *younger* i netočni.

Sustav u 1 od 222 slučaja nije dao nikakav odgovor, iako je odgovor bio očekivan i mogao se pronaći u polaznom znanju. Slučaj kad nedostaje odgovor prikazan je u tablici



7.17 u dijelu Nedostaje odgovor. Na pitanje 134. *What kind of a table is it?* sustav na temelju rečenice 11. *It is a stone table.* odgovara *stone*. Međutim, na temelju rečenice 13. *The table is wooden.* sustav je trebao odgovoriti i *wooden*. Ovdje je opet razlika u razinama hijerarhije kao i kod istog primjera na hrvatskom jeziku.

Sustav je dao semantički nepotpun odgovor u 9 od 222 slučaja. Svi takvi odgovori prikazani su u tablici 7.17 u dijelu Semantički nepotpun odgovor. Kod takve vrste odgovora, sustav djelomično (nepotpuno) odgovara na pitanje. On pronalazi izoliran, djelomičan odgovor koji je izvučen iz ispravnog znanja. Međutim, bilo bi prirodnije da je odgovor na postavljeno pitanje potpuniji, opširniji.

- Na pitanje 64. *When does Lucas do his homework?* na temelju znanja iz rečenice 83. *Lucas does his homework every day at seven o'clock.*, sustav odgovara *at seven o'clock*. Ovaj odgovor je točan, međutim potpuniji odgovor bio bi *every day at seven o'clock*.
- Na pitanje 78. *Who did Noa give this book to?* na temelju znanja iz rečenice 66. *Noa gave this book to a friend.* sustav odgovara *friend*. Ovaj odgovor je točan, međutim potpuniji odgovor bio bi *to a friend*.
- Na pitanja 79. *Where we don't sleep?* i 80. *Where we do not sleep?* na temelju znanja iz rečenice 38. *We don't sleep in the hotel Intercontinental.* sustav odgovara *in the hotel*. Ovaj odgovor je točan, međutim potpuniji odgovor bio bi *in the hotel Intercontinental*.
- Na pitanje 107. *What do the students talk about?* na temelju znanja iz rečenice 58. *Students talk about the solution.* sustav odgovara *solution*. Ovaj odgovor je točan, međutim potpuniji odgovor bio bi *about the solution*.
- Na pitanje 112. *What was Vanessa?* na temelju znanja iz rečenice 50. *Vanessa was the fifteenth person to win the award since 1999.* sustav odgovara *to win the award*. Ovaj odgovor je nepotpun i trebao bi glasiti *the fifteenth person to win the award since 1999*.
- Na pitanje 141. *What does Lucas do?* na temelju znanja iz rečenice 83. *Lucas does his homework every day at seven o'clock.* sustav odgovara *does*. Ovaj odgovor je točan, međutim potpuniji odgovor bio bi *does homework* ili *does his homework*.

- Na pitanje 145. *What is London?* na temelju znanja iz rečenice 69. *London is the capital of Great Britain.* sustav odgovara *of Great Britain.* Ovaj odgovor je nepotpun i trebao bi glasiti *the capital of Great Britain.*
- Na pitanje 148. *What is fast?* na temelju znanja iz rečenice 60. *The train to New York is fast.* sustav odgovara *the train.* Ovaj odgovor je točan, međutim potpuniji odgovor bio bi *The train to New York.*

Sustav je dao semantički ispravan, ali gramatički neispravan odgovor u 4 od 222 slučaja. Svi takvi odgovori prikazani su u tablici 7.17 u dijelu Semantički ispravan, ali gramatički neispravan odgovor. Kod takvih odgovora, sustav daje odgovor temeljen na ispravnom znanju, ali odgovor ne odgovara pravilima gramatike.

- Na pitanja 38. *Whose friends are good boys?* i 172. *Whose cat is this?* sustav odgovara *my* i *your* (na temelju znanja iz rečenica 27. *All my friends are good boys.* i 34. *This is your cat.* Prema pravilima gramatike, točan odgovor na ovo pitanje bio bi *my friends* ili *mine*, odnosno *your cat* ili *yours.*
- Na pitanja 139. *What is dinner?* i 151. *What is Tom?* sustav odgovara *favorite Lukas's meal* i *than Peter younger.* Potpuno točan odgovor bio bi *Lucas's favorite meal* i *younger than Peter* (na temelju znanja iz rečenica 91. *Dinner is Lucas's favorite meal.* i 79. *Tom is younger than Peter.*)

Sustav je točno odgovorio u 202 od 222 slučaja. Neki od tih točnih odgovora prikazani su u tablici 7.17 u dijelu Točni odgovori. Kao točni odgovori prihvaćani su odgovori koji su u potpunosti identični odgovoru čovjeka. Osim toga, kod pitanja tipa *What does someone do?* kao točan odgovor vrednovao se odgovor koji se sastojao samo od procesnog čvora (na primjer, *drives, talk, write, eats* i slično). Čovjek može svojim znanjem na temelju polaznih rečenica na takve odgovore odgovoriti detaljnije, na primjer, *drives his brother's car, write a seminar paper* ili ponekad dati odgovor cijelom rečenicom. Zbog toga se za točan odgovor uzimao onaj odgovor koji zadovoljava semantičku i gramatičku ispravnost (sastoji se od jedne ili više riječi). Iz istog su se razloga prihvaćali odgovori na pitanja tipa *How many pounds of potatoes does Thomas buy?, What kind of books does Maya have?* i slično. Odgovori na ova pitanja mogu biti *10* i *interesting, 10 pounds* i *interesting books* ili čak cijele rečenice *Thomas buys 10 pounds of potatoes in the store.* i *Maya has some interesting books.*

Daljnjom doradom sustava planira se omogućiti sustavu detaljnije odgovaranje na pitanja. Svi ovi uočeni problemi ostaju za doradu algoritama i budući rad.

### **7.3 Usporedba dvaju sustava temeljenih na NOK metodi**

U nastavku će biti prikazana usporedba dvaju sustava temeljenih na NOK metodi. Tijekom razvoja NOK metode, njenih varijanti i formalizama, razvile su se i dvije istraživačke ideje kako razviti sustav temeljen na NOK-u. S obzirom da se nismo htjeli ograničiti samo na jednu ideju, obje ideje su krenule u realizaciju i razvijena su dva sustava. Prvi sustav (QANOK) temelji se na gramatikama, odnosno razvoju pretvarača (engl. *transducer*), a drugi na relacijskim bazama podataka.

Usporedba ovih dvaju sustava pomoći će nam u određivanju napretka koji je postignut u ovom radu. U toku evaluacijskog procesa, oba sustava su testirana koristeći 42 rečenice i 88 pitanja prirodnog engleskog jezika. U nastavku su prikazani rezultati testiranja oba sustava, usporedba dobivenih odgovora iz oba sustava te analiza ispravnosti dobivenih odgovora.

#### **7.3.1 Sustav temeljen na gramatici (engl. Grammar-based system (GBS))**

Prvi razvijeni QA sustav temeljen na NOK metodi nazvan je Question Answering NOK (QANOK) i detaljno je opisan u [90]. U nastavku će se ovaj sustav označavati s GBS skraćeno od Grammar-based system. GBS je računalni pretvarač čija je primarna funkcija osigurati precizne odgovore na pitanja postavljena u prirodnom jeziku. U tu svrhu razvijena su tri algoritma [90]:

- algoritam koji znanje iskazano tekstom u prirodnom jeziku prevodi u FMTEK (engl. *Formalized method for text expressed knowledge*) formaliziranu notaciju,
- algoritam koji pitanje u prirodnom jeziku prevodi u QFMTEK (question FMTEK) formaliziranu notaciju i
- algoritam koji koristi QFMTEK formalizirana pitanja i traži odgovor unutar FMTEK formaliziranog znanja.

GBS je implementiran u Python-u. Sastoji se od "two phrase structure grammars" (PSG). Jedna se koristi za sintaksnu analizu neformaliziranih rečenica, odnosno rečenica iskazanih prirodnim jezikom. Druga se koristi u formaliziranoj notaciji iste rečenice

primjenom dijelova koji su identificirani tijekom sintaksne analize. Ona čuva semantičku vezu između čvorova koja je identificirana koristeći wh-pitanja. Identifikacija moguće semantičke veze između riječi čuva se u proširenom MULTEXT-East leksikonu.

Testovi izvedeni na ovom sustavu pokazali su da on može "razumjeti" jednostavne rečenice u engleskom jeziku i pronaći odgovor na postavljeno pitanje. Tijekom testiranja korišten je leksikon koji se sastojao od oko stotinjak odabranih riječi. Postupak testiranja se sastojao od nekoliko koraka. Rečenice u prirodnom jeziku opisane su pomoću regularne gramatike i ekvivalentnih regularnih izraza. Nakon toga se pomoću sintaksno upravljano prevođenja [25] prevode u FNOK zapis. Za svaku od rečenica formirana je jedna ili više upitnih rečenica koje su sintaksno-kontroliranim translacijama transformirane u QFNOK zapise. U procesu traženja odgovora FNOK i QFNOK izrazi se uparuju te varijable u QFNOK zapisu predstavljaju odgovor u FNOK zapisu.

Sustav je testiran na 42 rečenice na engleskom jeziku (TENG). Na temelju njih sastavljeno je 88 pitanja na engleskom jeziku, odnosno za svaku rečenicu postavljeno je najmanje jedno pitanje. Korištene su dvije vrste pitanja – Wh-pitanja i Da/Ne pitanja (engl. *polar questions*).

Analiza odgovora je pokazala da GBS sustav treba poboljšanja. Jedan od problema GBS sustava je problem jednine i množine. Za pitanje postavljeno u jednini (na primjer, *Who is on the beach?*) odgovor može biti u množini (na primjer, *Girls are on the beach*). GBS sustav u ovom slučaju neće odgovoriti na postavljeno pitanje. Problemi prepoznati tijekom testiranja opisani su u [90] te je ovaj sustav moguće i dalje razvijati.

Razvijen je Sustav temeljen na NOK metodi koji koristi algoritme za transformaciju tekstom iskazanih znanja u formalizirane zapise prilagođene upisu u relacijske baze podataka. On je proširen nizom poboljšanja. U nastavku ćemo ovaj sustav označavati s RDBS (skraćeno od *The system based on relational databases*).

### **7.3.2 Sustav temeljen na NOK metodi**

RDBS sustav (skraćeno od *The system based on relational databases*) je sustav temeljen na NOK metodi. Koristi algoritme za transformaciju iskazanog znanja u FNOK

formaliziranom zapisu u relacijske baze podataka. U sustav je implementiran jednojezični rječnik koji omogućuje uključivanje semantičkih relacija, npr. hijerarhija.

U prethodnim poglavljima detaljno su prikazani modeli podataka i procesa koji opisuju RDBS sustav. Osim toga opisan je razvijeni sustav implementiran koristeći bazu u Oraclu.

Svojstva relacijske baze podataka osiguravaju gotovo neograničeno korištenje algoritama za odgovaranje na pitanja. Obogaćen wh-pitanjima koje objašnjavaju vezu između riječi, tekst se u relacijsku bazu podataka (slijedeći propisane procedure za pretvaranje rečenica i pitanja u formalizirane zapise) pohranjuje bez gubitaka semantike.

Rečenice i pitanja su pripremljeni u FNOK i QFNOK zapise pomoću zasebnog programa prevođenja rečenica u FNOK i QFNOK zapis opisanog u [47] i [90]. Program je doraden i prilagođen pravilima definiranim u poglavlju 4. te prikazan u reduciranom obliku u Pravitku 3. Samim time, ova pravila omogućuju obradu više vrsta riječi od pravila primijenjenih u GBS sustavu. Tablica 7.18 prikazuje neke od razlika u FNOK zapisu u ovisnosti je li korištena postojeća gramatika ili doradena (poboljšana) gramatika.

**Tablica 7.18. Razlika u FNOK zapisu u ovisnosti o gramatici**

<b>FNOK (GBS)</b>	<b>FNOK (RDBS)</b>
<p>Ulaz: The car drives. Dd Nc Vm</p> <p>Izlaz: Vm (QN Dd Nc) drives (what? the car)</p>	<p>Ulaz: The car drives. Dd Nc Vm</p> <p>Izlaz: Vm(QNc Nc(QD Dd)) drives("what?" car("art?" the))</p>
<p>Ulaz: A boy stares. Di Nc Vm</p> <p>Izlaz: Vm (QN Di Nc) stares (who? a boy)</p>	<p>Ulaz: A boy stares. Di Nc Vm</p> <p>Izlaz: Vm(QN1 Nc(QD Di)) stares("who?" boy("art?" a))</p>
<p>Ulaz: The red car drives. Dd Af Nc Vm</p> <p>Izlaz: Vm (QN Dd Nc (QA Af)) drives (what? The car (which? red))</p>	<p>Ulaz: The red car drives. Dd Af Nc Vm</p> <p>Izlaz: Vm(QNc Nc(QD Dd, QA4 Af)) drives("what?" car("art?" the, "which?" red))</p>

### 7.3.3 Rezultati usporedbe dvaju sustava

GBS sustav testiran je u procesu odgovaranja na pitanja. Za skup pitanja, GBS sustav je dao odgovor temeljen na znanju pohranjenom u bazi znanja. Nakon poboljšanja metode za formalizaciju, odnosno definiranja pravila za pretvaranje rečenica u FNOK zapise i pitanja u QFNOK zapise, razvoja algoritama za traženje odgovora, a samim time i razvoja RDBS sustava, RDBS sustav je testiran.

Oba sustava testirana su nad istim skupom pitanja i odgovora. Cilj je bio testirati rezultate, usporediti ih s odgovorima primljenim od GBS sustava te analizirati točnost primljenih odgovora. Polazno znanje je jedan paragraf sastavljen od 42 tekstualne rečenice prirodnog engleskog jezika (TENG). Svaki sustav koristio je svoje metode za pretvaranje rečenica i pitanja u formalizirane zapise. U procesu dobivanja odgovora, korišten je skup od 88 pitanja (QTENG). U tablici 7.19 i tablici 7.20 prikazani su isječci rečenica i pripadni FNOK-ovi, odnosno pitanja i njima pripadni QFNOK-ovi za oba sustava. Cjelokupne tablice prikazane su u tablicama 25 i 26 u Pravitku 2 gdje je vidljiva razlika u FNOK zapisima. Treba napomenuti da je u ovom radu bilo moguće uzeti pravila iz GBS sustava, međutim potrebno je kontinuirano poboljšanje FNOK-a u cilju postizanja boljih odgovora i taj proces treba nastaviti.

Tablica 7.19. Isječak rečenica i FNOK-ova u oba sustava

TENG	FNOK (GBS)	FNOK (RDBS)
The car drives.	drives("what?" the car)	drives ("what?" car ("art?" the))
A boy stares.	stares("who?" a boy)	stares ("who?" boy ("art?" a))
My car drives.	drives("what?" car("whose?" my))	drives ("what?" car ("whose?" my))
The red car drives.	drives("what?" the car("which?" red))	drives ("what?" car ("art?" the, "which?" red))
A tall boy stares.	stares("what?" a boy("which?" tall))	stares ("who?" boy ("art?" a, "which?" tall))
My incredibly tall boy stares.	stares("whose?" my "how?" incredibly "who?" boy("which?" tall))	stares ("who?" boy ("whose?" my, "which?" tall ("how?" incredibly)))
The red car on the road drives.	drives("what?" the car("which?" red) "where?" on "what?" the road)	drives ("what?" car ("art?" the, "which?" red), "where?" road ("where?" on, "art?" the))

Tablica 7.20. Isječak pitanja i QFNOK-ova u oba sustava

QTENG	QFNOK (GBS)	QFNOK (RDBS)
Are girls on the beach?	are (_ girls, _ on _ the beach	are (_ girls, _ beach (_ on, _ the))
Who is on the beach?	is ("who?" X, & _ on _ the beach	is ("who?" X, _ beach (_ on, _ the))
Who are on the beach?	are ("who?" X, & _ on _ the beach	are ("who?" X, _ beach (_ on, _ the))

Who write a letter to a friend?	write ("who?" X, _ a letter _ to _ a friend	write ("who?" X, _ letter ( _ a), _ friend ( _ to, _ a))
What does someone's electrical lamp do?	X ( _ lamp ( _ someone's, _ electrical)	X ( (& does), (& do), _ lamp ( _ someone's, _ electrical))
When does Tom drive a car?	drive ( _ Tom, _ a car "when?" X	drive ( ( & does), _ Tom, _ car ( & a), "when?" X)

U RDBS sustavu korištena su poboljšana pravila za formalizaciju rečenica i pitanja (tj. pretvaranje rečenica i pitanja u FNOK i QFNOK). Primjeri promjena za FNOK prikazani su u tablicama 7.21 i 7.22, a za QFNOK u tablici 7.23. Poboljšanja su vidljiva u nekoliko elemenata: prijelaz s linearnog na hijerarhijski zapis, članovi, implementacija pravila za različite tipove podataka i uvođenje rječnika:

- **Prijelaz s linearnog na hijerarhijski zapis.** Kod GBS sustava korišten je uglavnom linearni zapis koji započinje glagolom, a nakon njega navode se ostale riječi u onom redosljed u kojem se nalaze u rečenici (izuzev pridjeva koji su ispod imenica). U RDBS sustavu, zapisi također kreću s glagolom, ali se detaljno analizira odnos hijerarhije među riječima i taj se redosljed upisuje. Svaka riječ u hijerarhiji je dodijeljena onoj riječi na koju se odnosi, što je regulirano pravilima.
- **Članovi.** Umjesto korištenja članova vezanih na same riječi, poboljšana formalizacija uvodi i koristi ulogu čvora "art?" za svaki član u rečenici. Član je na nižoj razini hijerarhije od onog čvora kojem pripada.
- **Nova pravila transformacije različitih vrsta riječi.** Uvedena su nova pravila transformacije različitih vrsta riječi (posebice za brojeve, prijedloge i članove) za prevođenje TENG u FNOK te QTENG u QFNOK. Osim toga, poboljšana su pravila za imenice, zamjenice, pridjeve, priloge i glagole (uključujući različita glagolska vremena, primjerice, dodana su pravila za prošlo vrijeme (*did*)).
- **Rječnik.** U RDBS sustav uveden je hijerarhijski rječnik u procesu pohrane podataka. Za svaku riječ korištenu u rečenici, traži se da ista postoji u svim formama u rječniku. Na primjer, glagol *drive* povezan je sa svojim formama *drives*, *drove* i *driven*. Ovo omogućuje sustavu pronalazak odgovora na pitanje, iako je u pitanju upotrebljen drugi oblik riječi.

Tablica 7.21. Promjene uvedene u pravila prevođenja rečenica u FNOK

TENG	FNOK (GBS)	FNOK (RDBS)
<b>Hierarchy</b>		
My incredibly tall boy stares.	stares("whose?" my "how?" incredibly "who?" boy("which?" tall))	stares ("who?" boy ("whose?" my, "which?" tall ("how?" incredibly)))
<b>Articles</b>		
The red car on the road drives.	drives("what?" the car("which?" red) "where?" on "what?" the road)	drives ("what?" car ("art?" the, "which?" red), "where?" road ("where?" on, "art?" the))
<b>Nova pravila transformacije različitih vrsta riječi</b>		
The red car on the road drives.	drives("what?" the car("which?" red) "where?" on "what?" the road)	drives ("what?" car ("art?" the, "which?" red), "where?" road ("where?" on, "art?" the))
I talk about the solution with my colleague.	talk("who?" I, "where?" about "what?" the solution "where?" with "whose?" my colleague)	talk ("who?" I, "what?" solution ("what?" about, "art?" the), "who?" colleague ("who?" with, "whose?" my))
Tom has two cars.	has("who?" Tom, "how many?" two "what?" cars)	has ("who?" Tom, "what?" cars ("how many?" two))

Tablica 7.22. Promjene uvedene u pravila prevođenja rečenica u FNOK - Rječnik

TENG	FNOK (GBS)	FNOK (RDBS)	DICT
Tom drives a car today.	drives("who?" Tom, "what?" a car "when?" today)	drives ("who?" Tom, "what?" car ("art?" a), "when?" today)	drive
Tom drove red car on Monday.	drove("who?" Tom, "what?" car("which?" red) "when?" on "who?" Monday)	drove ("who?" Tom, "what?" car ("which?" red), "when?" Monday ("when?" on))	drive

Poboljšanja formalizacije su vidljiva i kod pretvorbe pitanja u QFNOK. Svaka riječ iz rečenice prirodnog jezika postaje čvor u QFNOK zapisu. Uvedena je hijerarhija u QFNOK-u te zatvorena zagrada kao obavezan znak ")" koji se koristi za vraćanje na višu razinu hijerarhije. Najveća promjena je istaknuta u tablici 7.23. Na primjer, za pitanje *What does Julia do to a friend?* uvedeni su oblici (& does) i (& do) te hijerarhija *\_ friend ( \_ to, \_ a)*.

Tablica 7.23. Promjene uvedene u pravila prevođenja pitanja u QFNOK

QTENG	QFNOK (GBS)	QFNOK (RDBS)
Are girls on the beach?	are ( _ girls, _ on _ the beach	are ( _ girls, _ beach ( _ on, _ the))
Who is on the beach?	is ("who?" X, & _ on _ the beach	is ("who?" X, _ beach ( _ on, _ the))
What does Julia do to a friend?	X ( _ Julia & _ to _ a friend	X ( (& does), (& do), _ Julia, _ friend ( _ to, _ a))
What does Julia do on the pool?	X ( _ Julia & _ on _ the pool	X ( (& does), (& do), _ Julia, _ pool ( _ on, _ the))
Where does Julia swim?	swim ( _ Julia & "where?" X	swim ( (& does), _ Julia, "where?" X)
How does Julia swim?	swim ( _ Julia, "how?" X	swim ( (& does), _ Julia, "how?" X)



TENG rečenice i QTENG pitanja bili su jednaki kod oba testirana sustava. Promjene temeljene na novim pravilima uvedene su u ukupno 29 FNOK zapisa (od toga je 12 manjih promjena te 17 većih) te u svakom QFNOK zapisu. Manje promjene odnose se na članove (RDBS sustav smatra svaki čvor kao zaseban objekt) i promjenu *wh*-pitanja korištenog za definiranje uloga (na primjer, *what* je ponekad mijenjan u *which*; *what kind* u *what* ili *which* i tako dalje). Veće promjene odnose se na promjene u hijerarhiji te na nova pravila za brojeve i prijedloge.

Nakon provedenog testiranja RDBS sustava, analizirana je kvaliteta dobivenih odgovora. Svaki je odgovor procijenjen prema svojoj cjelovitosti i točnosti. Odgovori su kategorizirani u nekoliko kategorija:

- Netočan odgovor,
- Nedostaje odgovor,
- Semantički nepotpun odgovor,
- Semantički ispravan, ali gramatički neispravan odgovor i
- Točan odgovor.

Ove kategorije su detaljno objašnjene u ranijim poglavljima (posebice u poglavlju 7.2) te ih ovdje nećemo dodatno objašnjavati. Tablica 27 u Pravitku 2 prikazuje sve rezultate dobivene testiranjem oba sustava, a u tablicu 7.24 izdvojeni su samo neki dijelovi te tablice grupirani po kategoriji odgovora.

Tablica 7.24. Neki od odgovora primljenih iz sustava

QTENG	ANSWER	TENG koja sadrži odgovor
<b>Netočan odgovor</b>		
Who is on the beach?	I do not know	Girls are on the beach.
Whose red car did he drive on Monday?	I do not know	He drove my red car on Monday.
With what I talk?	colleague	I talk about the solution with my colleague.
What does Tom do?	has	Tom has two cars.
<b>Nedostaje odgovor</b>		
What does a pencil do?	writes	A pencil writes.
		Student's pencil writes.
<b>Semantički nepotpun odgovor</b>		
To whom does Julia write a letter?	friend	Julia writes a letter to a friend.
<b>Semantički ispravan, ali gramatički neispravan odgovor</b>		
Whose car drives?	my	My car drives.
<b>Točan odgovor</b>		

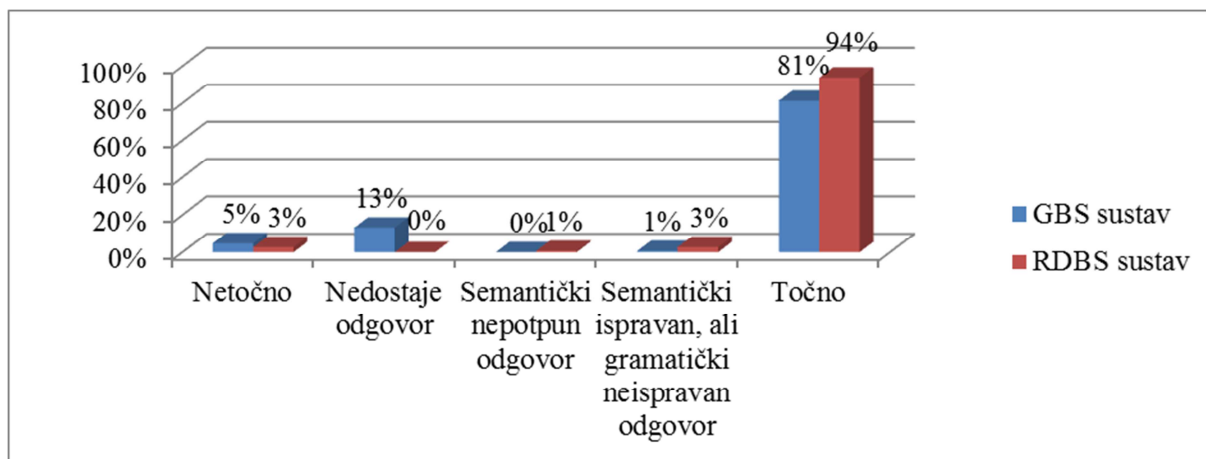
Are girls on the beach?	Yes	Girls are on the beach.
Who is on the beach?	girls	Girls are on the beach.
Where are girls?	on the beach	Girls are on the beach.
What does Julia do to a friend?	writes	Julia writes a letter to a friend.
What glows?	everything	Everything glows.
	someone's lamp	Someone's lamp glows.
	someone's electrical lamp	Someone's electrical lamp glows.

Analiza dobivenih odgovora oba sustava te njihova međusobna usporedba prikazana je u tablici 27 u Privitku 2. Na temelju 42 TENG rečenice i 88 QTENG pitanja ukupno je dobiveno 144 odgovora (neka pitanja imala su više odgovora). Tablica 7.25 prikazuje rezultate analize odgovora.

Tablica 7.25. Analiza odgovora

Kategorija odgovora	GBS	RDBS
Netočno	7	4
Nedostaje odgovor	19	0
Semantički nepotpun odgovor	0	1
Semantički ispravan, ali gramatički neispravan odgovor	1	4
Točno	117	135
Ukupno	144	144

GBS sustav netočno je odgovorio u 7 od 144 slučaja, dok je RDBS sustav netočno odgovorio u 4 od 144 slučaja. Iako je odgovor bio očekivan, GBS sustav nije ga dao u 19 slučajeva, dok je RDBS sustav uvijek dao neki odgovor (ali sa različitim razinama točnosti). U kategoriji *Semantički nepotpun odgovor*, GBS sustav nije imao niti jedan slučaj, dok je RDBS sustav imao 1 takav slučaj. U kategoriji *Semantički ispravan, ali gramatički neispravan odgovor*, GBS sustav imao je 1 slučaj, a RDBS sustav 4 slučaja. GBS sustav točno je odgovorio na 117 od 144 slučaja, dok je RDBS sustav točno odgovorio u 135 od 144 slučaja [5]. Slika 7.3 prikazuje analizu odgovora oba sustava. GBS sustav točno je odgovorio u 81% slučajeva dok je kod RDBS sustava točnost povećana na 94%.



Slika 7.3. Analiza odgovora za GBS i RDBS sustava

Analiza individualnih rezultata dobivenih iz oba sustava (tablica 7.26) pokazala je da je u 3 slučaja RDBS sustav pokazao lošije rezultate od GBS sustava (ponudio je netočan ili semantički nepotpun odgovor, dok je GBS sustav dao točan odgovor). U 119 slučajeva, oba sustava su ponudila odgovore jednake kvalitete (ponekad točno, ponekad netočno). U 21 slučaju RDBS sustav je bio bolji (ponudio je najmanje semantički ispravan, ali gramatički neispravan odgovor, dok je GBS sustav dao netočan odgovor ili je odgovor nedostajao).

Tablica 7.26. Kvaliteta odgovora RDBS sustava u odnosu na GBS sustav

Kvaliteta odgovora	Broj odgovora	%
Lošiji	3	2%
Isti	119	83%
Bolji	22	14%

Treba napomenuti da niti jedan sustav nije bolji niti lošiji. Svaki se može dalje razvijati i postići bolje rezultate od drugog sustava. Oba sustava zasnovana su na istoj metodi i za svakog možemo reći da je napravljen velik pomak u razvoju metode tijekom procesa implementacije.

## 7.4 Dokaz hipoteze H2

Hipoteza H2 glasi: *Semantika jednostavnih rečenica prirodnog hrvatskog jezika (izjavnih rečenica u nominativu te u sadašnjem glagolskom vremenu) ostaje očuvana u relacijskoj bazi podataka NOK-a (BPNOK).*

Ova hipoteza se smatra potvrđenom ako se pokaže da se upisivanjem rečenica prirodnog jezika (koje su NOK metodom i njenim pravilima transformirane u

odgovarajući FNOK zapis prikladan za upis u relacijsku bazu podataka) neće izgubiti semantika u rečenici. Očuvanost semantike dokazuje se propitivanjem relacijske baze podataka pitanjima prirodnog jezika (također transformiranih u odgovarajući QFNOK zapis koristeći NOK metodu), odnosno dobivanjem odgovora na postavljeno pitanje na temelju podataka u relacijskoj bazi podataka.

Očuvanost semantike pokazana je na nekoliko mjesta u ovom radu.

Popunjavanjem relacijske baze podataka NOK-a izjavnim rečenicama prirodnog hrvatskog jezika u sadašnjem glagolskom vremenu te nominativu jednine, pokazali smo da je moguće transformirati rečenice prirodnog jezika u relacijsku bazu podataka te je nad njom omogućeno propitivanje. Na osnovu tablica 5.1 do 5.5 iz poglavlja 5 koje prikazuju primjer jednostavnih rečenica prirodnog hrvatskog jezika pretvorenih u FNOK zapis te transformiranih u tablice relacijske baze podataka nastale na temelju meta modela jednosmjerne NOK metode, zaključujemo da semantika prirodnog jezika ostaje očuvana višestrukim prolascima kroz tablicu 5.5. Dakle, moguće je odgovoriti na pitanje postavljeno prirodnim jezikom bez gubitka semantike i time je dokazana hipoteza H2.

Očuvanost semantike rečenica prirodnog hrvatskog jezika pokazana je u tablici 21. Odgovori sustava na pitanja (hrvatski jezik) u Privitku 2. U toj tablici je pokazano da se koristeći znanje pohranjeno u relacijskoj bazi podataka (rečenice prirodnog jezika iz tablice 19. Odabrane rečenice hrvatskog jezika i pripadni im FNOK-ovi u Privitku 2) mogu postaviti pitanja na prirodnom jeziku (iz tablice 20. Pitanja prirodnog hrvatskog jezika i pripadni QFNOK-ovi u Privitku 2) na koja će se dobiti odgovori temeljeni na znanju pohranjenom u bazi podataka. Analizom odgovora prikazanih u tablici 21. Odgovori sustava na pitanja (hrvatski jezik) u Privitku 2 pokazano je da sustav odgovara *točno* u 93% slučajeva (na temelju 103 rečenice, postavljeno je 180 pitanja te je ukupno dobiveno 222 slučaja odgovora klasificiranih u različite kategorije). S obzirom da je u 103 rečenice ulaznog znanja bilo uključeno i nekoliko rečenica u prošlom i budućem glagolskom vremenu, negacije, imperativne rečenice te rečenice s neizrečenim subjektom koje izlaze iz okvira postavljene definicije jednostavne rečenice, pokazalo se da sustav vrlo dobro odgovara i na ovakva pitanja.

Da do gubitka semantike nije došlo upisom rečenice prirodnog jezika u bazu podataka, pokazat ćemo još na jednom primjeru. Za jednu rečenicu na hrvatskom jeziku postaviti

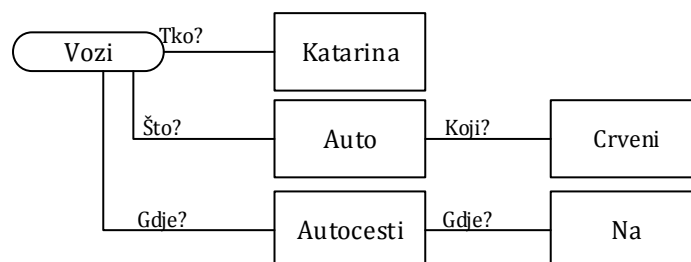
ćemo sva pitanja koja možemo postaviti koristeći riječi iz rečenice, a da pitanja budu smisljena.

Rečenica prirodnog jezika (THJ) glasi: *Katarina vozi crveni auto na autocesti.*

Pripadni FNOK zapis je:

*vozi ("tko?" Katarina, "što?" auto ("koji?" crveni), "gdje?" autocesti ("gdje?" na))*

Pripadni DNOK prikazan je na slici 7.4.



Slika 7.4. DNOK za rečenicu *Katarina vozi crveni auto na autocesti.*

Na temelju te rečenice postavljamo pitanja prikazana u tablici 7.27.

Tablica 7.27. Pitanja postavljena za jednu rečenicu i pripadni QFNOK-ovi

RB	Pitanje (QTHJ)	QFNOK
1	Tko vozi?	vozi ("tko?" X)
2	Tko vozi auto?	vozi ("tko?" X, _ auto)
3	Tko vozi crveni auto?	vozi ("tko?" X, _ auto ( _ crveni))
4	Tko vozi crveni auto na autocesti?	vozi ("tko?" X, _ auto ( _ crveni), _ autocesti ( _ na))
5	Tko vozi na autocesti?	vozi ("tko?" X, _ autocesti ( _ na))
6	Tko vozi auto na autocesti?	vozi ("tko?" X, _ auto, _ autocesti ( _ na))
7	Što vozi?	vozi ("što?" X)
8	Što vozi Katarina?	vozi ( _ Katarina, "što?" X)
9	Što vozi Katarina na autocesti?	vozi ( _ Katarina, "što?" X, _ autocesti ( _ na))
10	Što vozi na autocesti?	vozi ("što?" X, _ autocesti ( _ na))
11	Gdje vozi?	vozi ("gdje?" X)
12	Gdje vozi Katarina?	vozi ( _ Katarina, "gdje?" X)
13	Gdje vozi auto?	vozi ( _ auto, "gdje?" X)
14	Gdje vozi crveni auto?	vozi ( _ auto ( _ crveni), "gdje?" X)
15	Gdje Katarina vozi auto?	vozi ( _ Katarina, _ auto, "gdje?" X)
16	Gdje Katarina vozi crveni auto?	vozi ( _ Katarina, _ auto ( _ crveni), "gdje?" X)
17	Koji auto?	auto ("koji?" X)
18	Koji auto vozi?	vozi ( _ auto ("koji?" X))
19	Koji auto vozi Katarina?	vozi ( _ Katarina, _ auto ("koji?" X))
20	Koji auto vozi na autocesti?	vozi ( _ auto ("koji?" X), _ autocesti ( _ na))
21	Koji auto vozi Katarina na autocesti?	vozi ( _ Katarina, _ auto ("koji?" X), _ autocesti ( _ na))
22	Što Katarina radi?	X ( (& radi), _ Katarina)
23	Katarina vozi crveni auto na autocesti?	vozi ( _ Katarina, _ auto ( _ crveni), _ autocesti ( _ na))

Tablica 7.28 prikazuje analizu odgovora na pitanja iz tablice 7.27 temeljenih na jednoj rečenici. Answer ID je redni broj odgovora u tablici KB\_ANSWER, PRGQ i ORDQ su broj paragrafa i redni broj pitanja, QHJ je pitanje na prirodnom hrvatskom jeziku, PRGS i ORDS je broj paragrafa i redni broj rečenice, Očekivani odgovor je odgovor kojeg je dao čovjek, ANSWER sustava je odgovor koji je pronašao sustav, Usporedba je provjera točnosti prema različitim kategorijama opisanim u prethodnim poglavljima.

Tablica 7.28. Odgovora na pitanja temeljeni na jednoj rečenici

ANSWER_ID	PRGQ	ORDQ	QHJ	Očekivani odgovor	ANSWER sustava	Usporedba	PRGS	ORDS
1	11	1	Tko vozi?	Katarina	Katarina	Točno	11	1
2	11	2	Tko vozi auto?	Katarina	Katarina	Točno	11	1
3	11	3	Tko vozi crveni auto?	Katarina	Katarina	Točno	11	1
4	11	4	Tko vozi crveni auto na autocesti?	Katarina	Katarina	Točno	11	1
5	11	5	Tko vozi na autocesti?	Katarina	Katarina	Točno	11	1
6	11	6	Tko vozi auto na autocesti?	Katarina	Katarina	Točno	11	1
7	11	7	Što vozi?	crveni auto	crveni auto	Točno	11	1
8	11	8	Što vozi Katarina?	crveni auto	crveni auto	Točno	11	1
9	11	9	Što vozi Katarina na autocesti?	crveni auto	crveni auto	Točno	11	1
10	11	10	Što vozi na autocesti?	crveni auto	crveni auto	Točno	11	1
11	11	11	Gdje vozi?	na autocesti	na autocesti	Točno	11	1
12	11	12	Gdje vozi Katarina?	na autocesti	na autocesti	Točno	11	1
13	11	13	Gdje vozi auto?	na autocesti	na autocesti	Točno	11	1
14	11	14	Gdje vozi crveni auto?	na autocesti	na autocesti	Točno	11	1
15	11	15	Gdje Katarina vozi auto?	na autocesti	na autocesti	Točno	11	1
16	11	16	Gdje Katarina vozi crveni auto?	na autocesti	na autocesti	Točno	11	1
17	11	17	Koji auto?	crveni	crveni	Točno	11	1
18	11	18	Koji auto vozi?	crveni	crveni	Točno	11	1
19	11	19	Koji auto vozi Katarina?	crveni	crveni	Točno	11	1
20	11	20	Koji auto vozi na autocesti?	crveni	crveni	Točno	11	1
21	11	21	Koji auto vozi Katarina na autocesti?	crveni	crveni	Točno	11	1
22	11	22	Što Katarina radi?	vozi crveni auto na autocesti	vozi	Točno	11	1
23	11	23	Katarina vozi crveni auto na autocesti?	DA	YES	Točno	11	1

Budući da smo na svako postavljeno pitanje dobili točan odgovor, zaključujemo da je moguće upisati rečenice prirodnog jezika u relacijsku bazu podataka bez gubitka semantike. Drugim riječima, transformacijom jednostavnih rečenica prirodnog jezika

metodom NOK moguće je znanje iz tih rečenica pohraniti u relacijsku bazu podataka NOK (BPNOK) bez gubitka semantike sadržane u DNOK-u i/ili FNOK-u jer metoda NOK polazi od rečenice, u kojoj se pronalazi eksplicitan i implicitan sadržaj i oba se unose u DNOK i/ili FNOK, odnosno u BPNOK.

Time je hipoteza H2 potvrđena.

Dodatno, ova hipoteza je dokazana i za jednostavne rečenice prirodnog engleskog jezika. Programski sustav je testiran i nad jednostavnim rečenicama engleskog jezika koje su predstavljale skup ulaznog znanja. Sustav je točno odgovorio na 97% postavljenih pitanja. Time je također pokazano da semantika rečenica engleskog jezika nije izgubljena transformacijom u formalizirani zapis te upisom u relacijsku bazu podataka.

## **8 Proširenje postojeće baze podataka odabranog informacijskog sustava bazom podataka sustava temeljenog na NOK-u**

Kod projektiranja modela podataka informacijskog sustava treba biti pažljiv i predvidjeti sve buduće zahtjeve jer jednom izgrađenu aplikaciju nije preporučljivo mijenjati. U slučaju da se ukaže potreba za mijenjanjem jednog atributa u aplikaciji, potrebno je mijenjati shemu baze podataka, a zatim i samu aplikaciju.

Podaci poslovnih informacijskih sustava danas se pohranjuju uglavnom u relacijskim bazama podataka. Relacijske baze podataka omogućuju pohranu rečenica prirodnog jezika u tekstualno polje u koje se mogu upisivati rečenice prirodnog jezika u nestrukturiranom obliku. Međutim, nad ovako nestrukturiranim zapisom nije moguće korištenje SQL upitnog jezika. Pretraga po rečenicama se svodi na indeksiranje teksta te traženje jedne ili više riječi u pojedinom tekstu. Rezultat pretrage je prikaz slogova (zapisa) u kojima se pojavljuju navedene riječi. Nije moguće postaviti upit kojim bi dobili točan odgovor sadržan u tekstu. Stoga, da bismo propitivali bazu podataka postojećih informacijskih sustava upitima prirodnog jezika, potrebno je postojeće informacijske sustave proširiti sustavom temeljenim na NOK-u koji omogućuje postavljanje pitanja prirodnim jezikom.

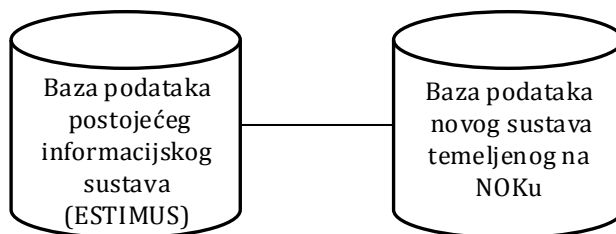
### **8.1 Proširenje baze postojećeg informacijskog sustava**

Za proširenje postojećeg informacijskog sustava odabran je aplikativni softver Estimus tvrtke Ris d.o.o. iz Kastva. Estimus je aplikativni softver za upravljanje projektima koji omogućuje brzu i jednostavnu organizaciju vremena i kadrova uključenih u projekte. Aplikacija pruža jasnu evidenciju rada svakog djelatnika te detalje o projektima na kojima oni rade. Međutim, Estimus ne omogućuje unos dodatnih znanja o djelatnicima ili projektima, primjerice osobne podatke o djelatnicima, činjenice koje se izražavaju rečenicama prirodnog jezika, počevši od onih najjednostavnijih. Sustav temeljen na NOK metodi omogućit će unos dodatnih znanja o djelatnicima, projektima i drugim entitetima i tipovima entiteta koji se nalaze u postojećoj bazi podataka. Na primjer, životopis ili nova znanja iskazana jednostavnim rečenicama: *Ivana poznaje PHP., Petar govori*



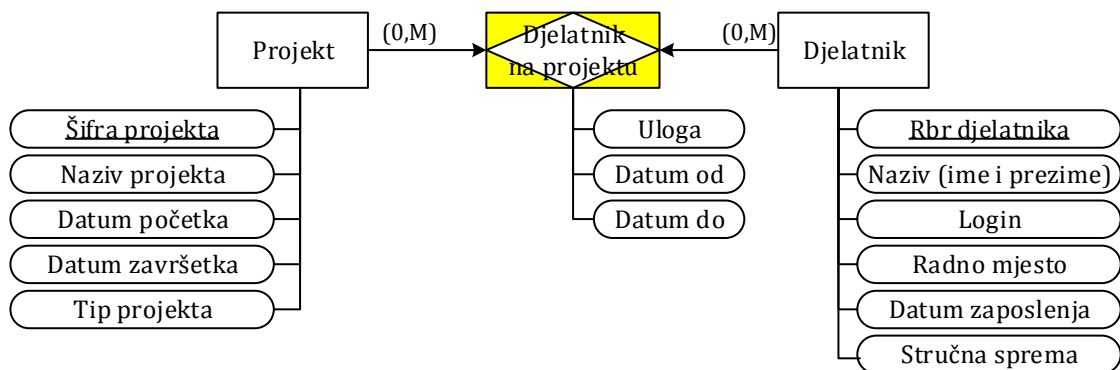
engleski. i slično, gdje su Ivana i Petar djelatnici koji se nalaze u bazi podataka Estimusa, omogućit će potpunije izvještaje iz postojećeg informacijskog sustava.

Proširenje postojećeg informacijskog sustava novim sustavom, znači da ćemo bazu podataka postojećeg informacijskog sustava povezati s bazom podataka novog sustava temeljenog na NOK metodi (slika 8.1).



**Slika 8.1. Proširenje baze podataka postojećeg informacijskog sustava bazom podataka sustava temeljenog na NOK-u**

Kod implementacije, odnosno proširenja postojećeg informacijskog sustava ograničili smo se samo na dio tablica Estimusa (Projekt, Djelatnik, Djelatnik na projektu) (slika 8.2).



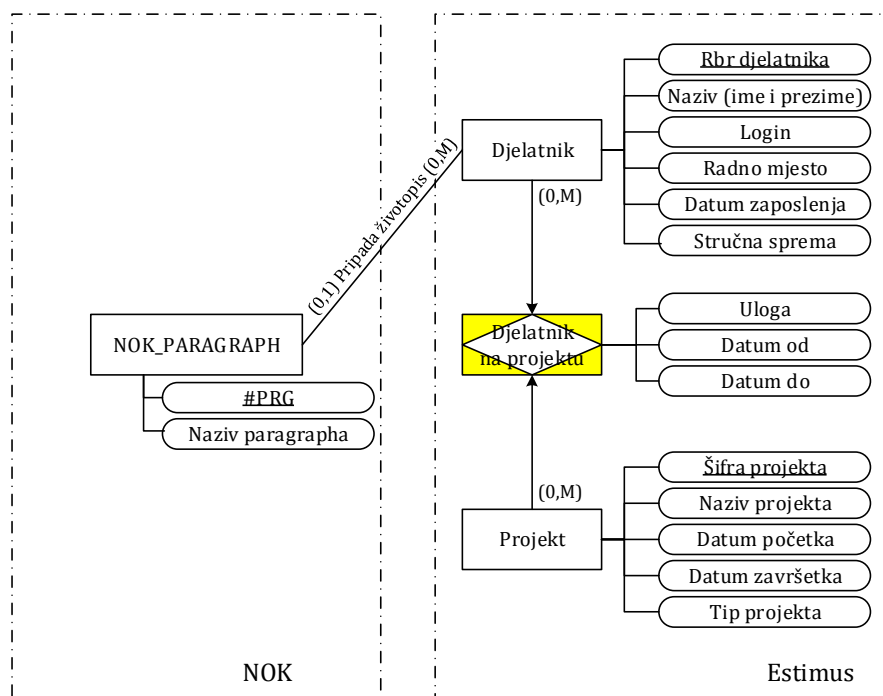
**Slika 8.2. DEV ESTIMUS**

Izdvojeni dio dijagrama entiteta i veza Estimusa sastoji se od dva tipa entiteta *Djelatnik* i *Projekt* te agregiranog tipa entiteta (agregacije) *Djelatnik na projektu*.

Tip entiteta *Djelatnik* s atributima *Rbr. djelatnika*, *Naziv (ime i prezime)*, *Login*, *Radno mjesto*, *Datum zaposlenja* i *Stručna sprema* sadrži podatke o djelatnicima tvrtke. Tip entiteta *Projekt* s atributima *Šifra projekta*, *Naziv projekta*, *Datum početka*, *Datum završetka* te *Tip projekta* sadrži podatke o svim projektima neke tvrtke. S obzirom da na jednom projektu iz tipa entiteta *Projekt* može raditi najmanje niti jedan djelatnik iz tipa entiteta *Djelatnik*, a najviše mnogo njih; te jedan djelatnik iz tipa entiteta *Djelatnik* može raditi na najmanje niti jednom, a najviše na mnogo projekata iz tipa entiteta *Projekt*,

uvodi se agregirani tip entiteta *Djelatnik na projektu* s atributima *Uloga*, *Datum od* i *Datum do*. Agregacija *Djelatnik na projektu* omogućuje pregled projekta i djelatnika, odnosno koji djelatnik je na kojem projektu radio u nekom periodu te koja je bila njegova uloga.

Slika 8.3 prikazuje povezanost postojećeg informacijskog sustava Estimusa i novog sustava temeljenog na NOK-u. Veza između ta dva sustava ostvaruje se vezom "pripada životopis" s brojnostima (0,1) – (0,M) između tipa entiteta (tablice) NOK\_PARAGRAPH iz baze podataka novog sustava temeljenog na NOK-u (BPNOK) te tipa entiteta (odnosno tablice) Djelatnik iz baze podataka postojećeg informacijskog sustava Estimusa (BP Estimus).



Slika 8.3. Povezanost NOK-a i Estimusa

Pretvaranjem dijagrama entiteta i veza sa slike 8.3 u relacijski model dobivamo sljedeće relacije u Estimusu:

- DJELATNIK** (Rbr djelatnika, Naziv (ime i prezime), Login, Radno mjesto, Datum zaposlenja, Stručna sprema)
- PROJEKT** (Šifra projekta, Naziv projekta, Datum početka, Datum završetka, Tip projekta)
- DJELATNIK NA PROJEKTU** (Rbr djelatnika, Šifra projekta, Uloga, Datum od, Datum do)

Osim toga, u dijelu NOK baze imamo relaciju:

**NOK\_PARAGRAPH (#PRG, Naziv paragrapha, RB djelatnika)**

Povezivanje sustava temeljenog na NOK-u i postojećeg informacijskog sustava Estimusa, odnosno njihovih baza podataka pokazat ćemo na primjeru životopisa djelatnika. Za dva životopisa dvaju djelatnika iz Estimusa, odnosno dva paragrafa bit će popunjene tablice u bazi podataka temeljenog na NOK-u. U nastavku će biti definirane tablice u bazi podataka Estimusa te tablica NOK\_PARAGRAPH (koja je dalje povezana s ostalim tablicama u bazi podataka sustava temeljenog na NOK-u (BPNOK)).

Znanje iskazano rečenicama prirodnog jezika, odnosno tekstovima životopisa bit će razrađeno u FNOK zapis prema pravilima te popunjeno u BPNOK. Na kraju, bit će definirane rečenice, odnosno pitanja koja odgovore traže u integralnoj bazi podataka, odnosno pitanja koja moraju koristiti podatke oba sustava da bi se došlo do odgovora.

Dodatno, zbog kontrole FNOK zapisa i provjere sustava temeljenog na NOK-u, definirana su i pitanja, odnosno QFNOK zapisi kojima se propituje tekstualno znanje u sustavu temeljenom na NOK-u.

U tablicama 8.1 do 8.3 prikazana je baza podataka postojećeg informacijskog sustava Estimusa. Tablica 8.4 prikazuje podatke u tablici NOK PARAGRAPH u BPNOK.

**Tablica 8.1. Djelatnik**

Rbr djelatnika	Naziv (ime i prezime)	Login	Radno mjesto	Datum zaposlenja	Stručna sprema
1	Pero Perić	pperic	projektant	01.01.1999.	VSS
2	Hrvoje Horvat	hhorvat	programer	15.03.2009.	VSS
3	Marija Marković	mmarkovic	tajnica	01.12.2000.	SSS
4	Ivana Ivić	iivic	programer	17.03.2008.	SSS
5	Petar Petrović	ppetrovic	programer	21.07.2010.	VSS
6	Jelena Ivić	jivic	projektant	01.09.2009.	VSS

**Tablica 8.2. Projekt**

Šifra projekta	Naziv projekta	Datum početka	Datum završetka	Tip projekta
I	MISUMI	15.03.2013.	15.03.2017.	uz potporu sveučilišta
II	MZOS	01.01.2000.	10.01.2010.	uz potporu sveučilišta
III	SMARTY	17.04.2010.	17.04.2013.	razvojni
IV	LUMENS	18.09.2014.	31.12.2020.	razvojni
V	KREDIS	20.05.2010.	-	održavanje
VI	PRIS	10.04.2012.	-	održavanje

**Tablica 8.3. Djelatnik na projektu**

Rbr djelatnika	Šifra projekta	Uloga	Datum od	Datum do
1	I	Nadzor na projektu	15.03.2013.	15.03.2015.

1	II	Vanjski suradnik	17.04.2001.	17.04.2010.
2	V	Voditelj projekta	20.05.2010.	01.01.2011.
2	IV	Član projektnog tima	18.09.2014.	18.09.2016.
3	V	Korisnička podrška	27.07.2012.	
3	VI	Korisnička podrška	15.04.2013.	
6	III	Član projektnog tima	17.04.2012.	17.04.2017.

**Tablica 8.4. NOK\_PARAGRAPH**

#PRG	Naziv paragrapha	Rbr djelatnika
0	nema	0
2201	Životopis Hrvoje Horvat	2
2202	Životopis Jelena Ivić	6

Životopisi djelatnika, odnosno rečenice prirodnog jezika koje sadrže znanje od djelatnicima, podijeljeni su u dva paragrafa. Životopis Hrvoja Horvata zapisan je pod šifrom #2201, a životopis Jelene Ivić pod šifrom #2202. U nastavku su dani životopisi pripremljeni za pretvorbu u DNOK i FNOK zapise (tablica 8.5 i tablica 8.6) te rečenice kojima možemo propitati znanje o životopisima (tablica 8.7).

### **2201 - Životopis Hrvoje Horvat**

Hrvoje Horvat rođen je 01.12.1978. u Rijeci. Hrvoje Horvat je završio Filozofski fakultet u Rijeci. Hrvoje Horvat je bio nastavnik matematike u "OŠ Rijeka". Hrvoje Horvat ima 5 godina radnog iskustva u projektiranju. Hrvoje Horvat dobro govori engleski jezik. Hrvoje Horvat slabo govori njemački jezik. Hrvoje Horvat programira u Javi. Hrvoje Horvat je vjenčan. Hrvoje Horvat ima dvoje djece. Hrvoje Horvat posjeduje vozačku dozvolu B kategorije. Hrvoje Horvat u slobodno vrijeme vozi automobilske utrke po svijetu.

### **2202 - Životopis Jelena Ivić**

Jelena Ivić je rođena 28.05.1980. u Varaždinu. Jelena Ivić je završila Tehnički fakultet u Zagrebu. Jelena Ivić je radila na nekoliko znanstvenih projekata. Jelena Ivić dobro govori njemački jezik. Jelena Ivić programira u Pythonu. Jelena Ivić je vjenčana. Jelena Ivić ima jedno dijete. Jelena Ivić ne posjeduje vozačku dozvolu. Jelena Ivić u slobodno vrijeme igra odbojku.

U tablici 8.5 prikazane su rečenice prirodnog hrvatskog jezika iz paragrafa 2201 (THJ – tekstualne rečenice na hrvatskom jeziku) te njihov odgovarajući FNOK zapis. Slika 8.4 prikazuje DNOK za rečenice dane u tablici 8.5. U tablici 8.6 prikazane su rečenice prirodnog hrvatskog jezika iz paragrafa 2202 te njihov odgovarajući FNOK zapis. DNOK za rečenice dane u tablici 8.6 vrlo je sličan DNOK-u prikazanom na slici 8.4 pa ga nećemo prikazivati.

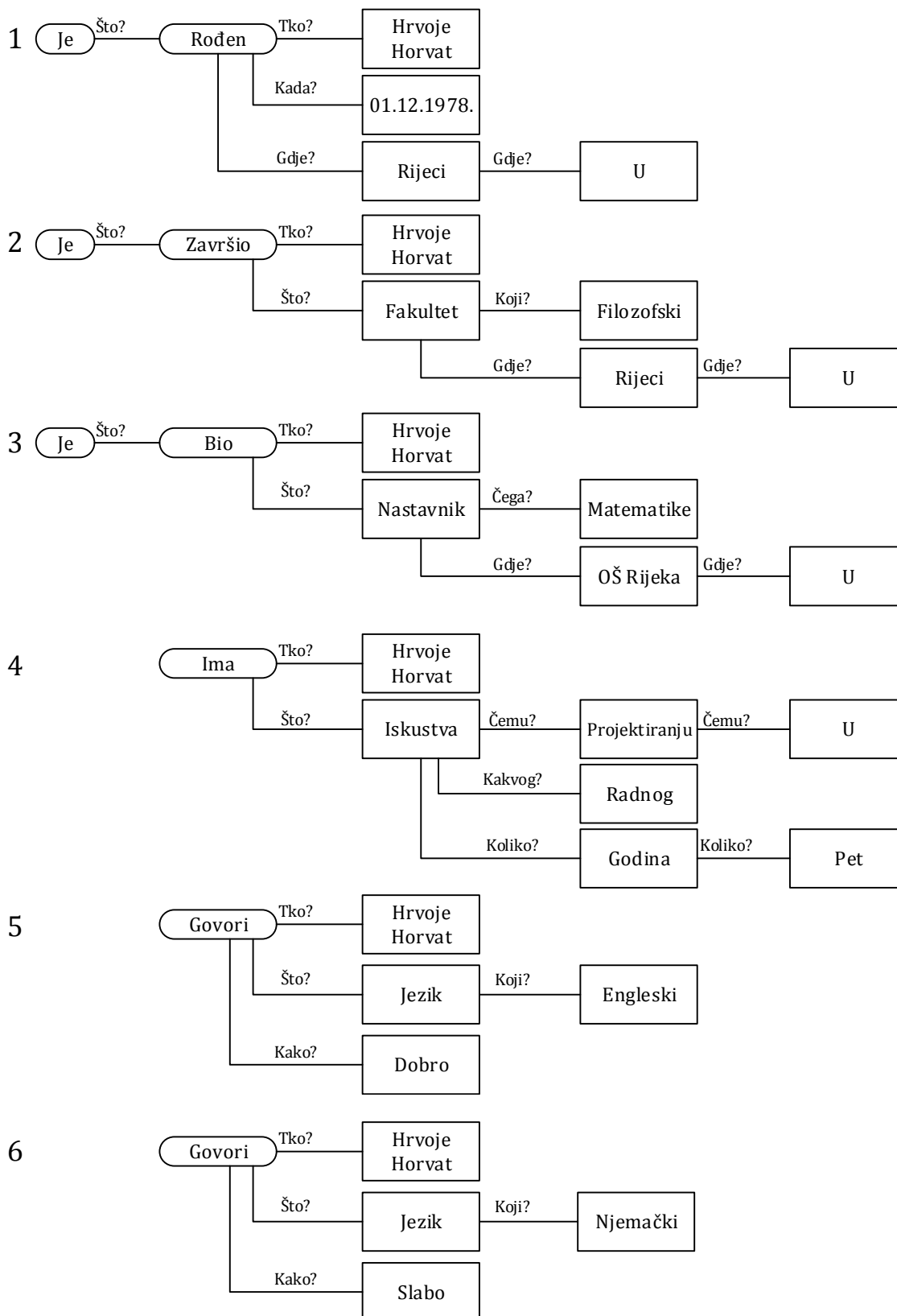
**Tablica 8.5. Životopis Hrvoje Horvat 2201.**

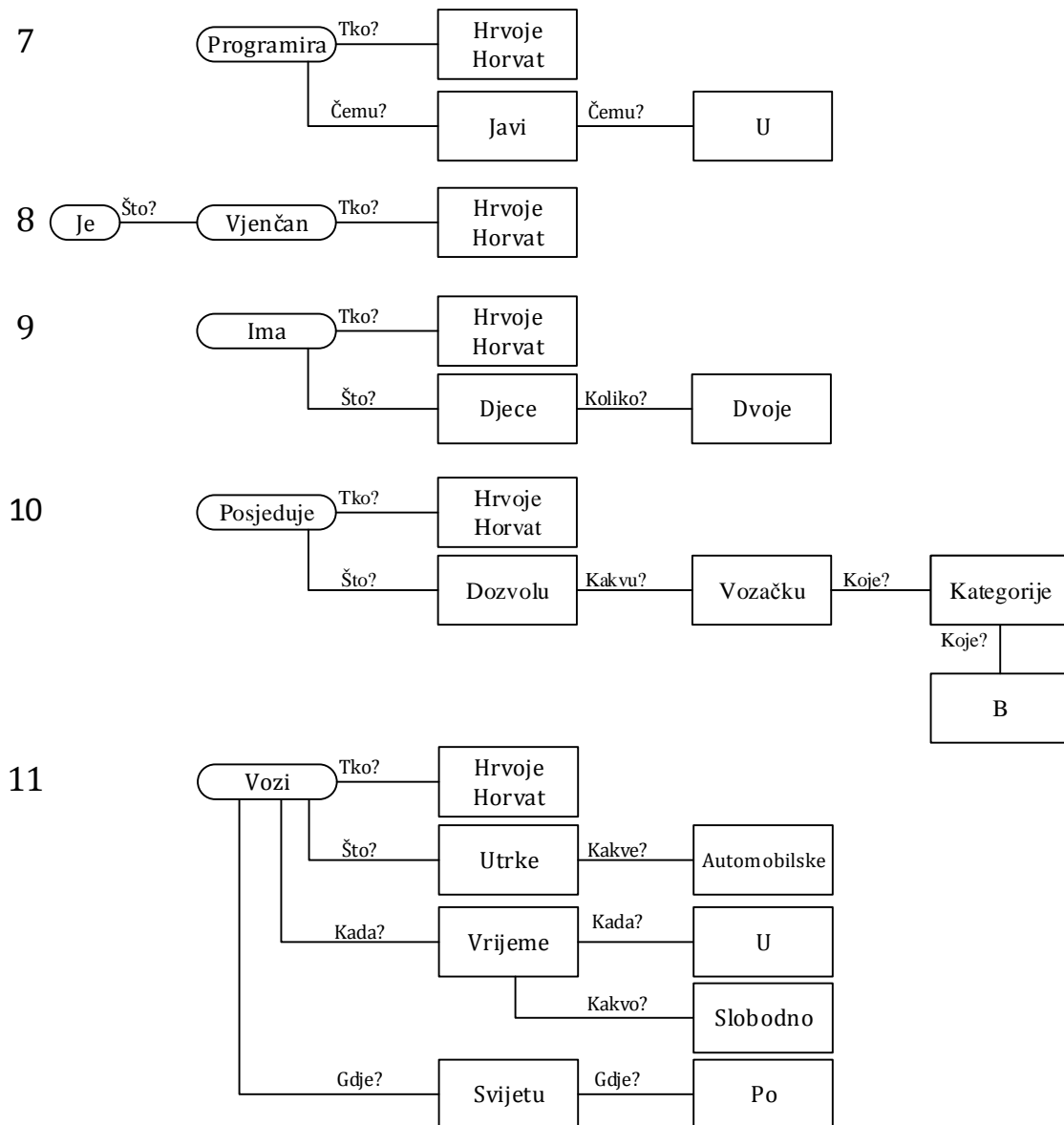
RBR	THJ	FNOK
1	Hrvoje Horvat rođen je 1.12.1978. u Rijeci.	je ("što?" rođen ("tko?" Hrvoje Horvat, "kada?" 01.12.1978., "gdje?" Rijeci ("gdje?" u)))
2	Hrvoje Horvat je završio Filozofski fakultet u Rijeci.	je ("što?" završio ("tko?" Hrvoje Horvat, "što?" fakultet ("koji?" filozofski, "gdje?" Rijeci ("gdje?" u))))
3	Hrvoje Horvat je bio nastavnik matematike u "OŠ Rijeka".	je ("što?" bio ("tko?" Hrvoje Horvat, "što?" nastavnik ("čega?" matematike, "gdje?" OŠ_Rijeka ("gdje?" u))))
4	Hrvoje Horvat ima 5 godina radnog iskustva u projektiranju.	ima ("tko?" Hrvoje Horvat, "što?" iskustva ("kakvog?" radnog, "čemu?" projektiranju ("čemu?" u), "koliko?" godina ("koliko?" 5)))
5	Hrvoje Horvat dobro govori engleski jezik.	govori ("tko?" Hrvoje Horvat, "što?" jezik ("koji?" engleski), "kako?" dobro)
6	Hrvoje Horvat slabo govori njemački jezik.	govori ("tko?" Hrvoje Horvat, "što?" jezik ("koji?" njemački), "kako?" slabo)
7	Hrvoje Horvat programira u Javi.	programira ("tko?" Hrvoje Horvat, "čemu?" Javi ("čemu?" u))
8	Hrvoje Horvat je vjenčan.	je ("što?" vjenčan ("tko?" Hrvoje Horvat))
9	Hrvoje Horvat ima dvoje djece.	ima ("tko?" Hrvoje Horvat, "što?" djece ("koliko?" dvoje))
10	Hrvoje Horvat posjeduje vozačku dozvolu B kategorije.	posjeduje ("tko?" Hrvoje Horvat, "što?" dozvolu ("kakvu?" vozačku ("koje?" kategorije ("koje?" B))))
11	Hrvoje Horvat u slobodno vrijeme vozi automobilske utrke po svijetu.	vozi ("tko?" Hrvoje Horvat, "što?" utrke ("kakve?" automobilske), "kada?" vrijeme ("kada?" u, "kakvo?" slobodno), "gdje?" svijetu ("gdje?" po))

**Tablica 8.6. Životopis Jelena Ivić 2202.**

RBR	THJ	FNOK
12	Jelena Ivić je rođena 28.05.1980. u Varaždinu.	je ("što?" rođena ("tko?" Jelena Ivić, "kada?" 28.05.1980., "gdje?" Varaždinu ("gdje?" u)))
13	Jelena Ivić je završila Tehnički fakultet u Zagrebu.	je ("što?" završila ("tko?" Jelena Ivić, "što?" fakultet ("koji?" tehnički, "gdje?" Zagrebu ("gdje?" u))))
14	Jelena Ivić je radila na nekoliko znanstvenih projekata.	je ("što?" radila ("tko?" Jelena Ivić, "gdje?" projekata ("gdje?" na, "koliko?" nekoliko, "kakvih?" znanstvenih)))
15	Jelena Ivić dobro govori njemački jezik.	govori ("tko?" Jelena Ivić, "što?" jezik ("koji?" njemački), "kako?" dobro)
16	Jelena Ivić programira u Pythonu.	programira ("tko?" Jelena Ivić, "čemu?" Pythonu ("čemu?" u))
17	Jelena Ivić je vjenčana.	je ("što?" vjenčana ("tko?" Jelena Ivić))
18	Jelena Ivić ima jedno dijete.	ima ("tko?" Jelena Ivić, "što?" dijete ("koliko?" jedno))
19	Jelena Ivić ne posjeduje vozačku	ne ("što?" posjeduje ("tko?" Jelena Ivić, "što?" dozvolu

	dozvolu.	("kakvu?" vozačku))
20	Jelena Ivić u slobodno vrijeme igra odbojku.	igra ("tko?" Jelena Ivić, "što?" odbojku, "kada?" vrijeme ("kada?" u, "kakvo?" slobodno))





Slika 8.4. DNOK za rečenice iz tablice 8.5 za životopis Hrvoja Horvata

### 8.1.1 Propitivanje BPNOK baze podataka

U tablici 8.7 prikazane su upitne rečenice, odnosno pitanja (QTHJ – upitne rečenice prirodnog hrvatskog jezika) kojima provjeravamo znanje vezano uz paragrafe 2201 i 2202 (Životopisi Hrvoja Horvata i Jelene Ivić) u BPNOK bazi podatka. Za svaku od rečenica definiran je odgovarajući QFNOK zapis te dan očekivani odgovor (odnosno odgovor koji je rekao čovjek) i odgovor koji je dao sustav temeljen na NOK-u. Osim toga, radi lakše kontrole, uz očekivani odgovor nalazi se i redni broj rečenice iz tablice 8.5 i/ili tablice 8.6 u kojoj je odgovor pronađen. Ispravnost odgovora BPNOK-a dana je u stupcu *Ispravnost odgovora*.

Tablica 8.7. Pitanja i QFNOK zapisi za životopise

RBR	QTHJ	QFNOK	RBR THJ u tablici 8.5 ili 8.6	Očekivani odgovor	Odgovor BPNOK-a	Ispravnost odgovora
1	Kada je rođen Hrvoje Horvat?	je ( _ rođen ( _ Hrvoje Horvat, "kada?" X ))	1	1.12.1978.	01.12.1978.	točno
2	Gdje je rođen Hrvoje Horvat?	je ( _ rođen ( _ Hrvoje Horvat, "gdje?" X ))	1	u Rijeci	u Rijeci	točno
3	Tko je rođen 01.12.1978.?	je ( _ rođen ("tko?" X, _ 01.12.1978.))	1	Hrvoje Horvat	Hrvoje Horvat	točno
4	Koji fakultet je završio Hrvoje Horvat?	je ( _ završio ( _ Hrvoje Horvat, _ fakultet ("koji?" X)))	2	filozofski	filozofski	točno
5	Gdje je Hrvoje Horvat bio nastavnik matematike?	je ( _ bio ( _ Hrvoje Horvat, _ nastavnik ( _ matematike, "gdje?" X)))	3	u OŠ Rijeka	u OŠ Rijeka	točno
6	Koliko godina radnog iskustva u projektiranju ima Hrvoje Horvat?	ima ( _ Hrvoje Horvat, _ iskustva ( _ radnog, _ projektiranju ( _ u), _ godina ("koliko?" X)))	4	5	5	točno
7	U čemu ima Hrvoje Horvat 5 godina radnog iskustva?	ima ( _ Hrvoje Horvat, _ iskustva ( _ radnog, "čemu?" X ( _ u), _ godina ( _ 5)))	4	u projektiranju	projektiranju	točno
8	Koji jezik Hrvoje Horvat dobro govori?	govori ( _ Hrvoje Horvat, _ jezik ("koji?" X), _ dobro)	5	engleski jezik	engleski	točno
9	Kako Hrvoje Horvat govori njemački jezik?	govori ( _ Hrvoje Horvat, _ jezik ( _ njemački), "kako?" X)	6	slabo	slabo	točno
10	Hrvoje Horvat slabo govori engleski jezik?	govori ( _ Hrvoje Horvat, _ jezik ( _ engleski), _ slabo)	5	NE	I DO NOT KNOW	netočno
11	U čemu programira Hrvoje Horvat?	programira ( _ Hrvoje Horvat, "čemu?" X ( _ u))	7	Javi	Javi	točno
12	Tko je vjenčan?	je ( _ vjenčan ("tko?" X))	8	Hrvoje Horvat	Hrvoje Horvat	točno



			17	Jelena Ivić	Jelena Ivić	točno
13	Koliko djece ima Hrvoje Horvat?	ima ( _ Hrvoje Horvat, _ djece ("koliko?" X))	9	Dvoje	dvoje	točno
14	Što posjeduje Hrvoje Horvat?	posjeduje ( _ Hrvoje Horvat, "što?" X)	10	vozačku dozvolu B kategorije	B kategorije vozačku dozvolu	* točno, ali gramatički nepotpuno
15	Kakvu dozvolu posjeduje Hrvoje Horvat?	posjeduje ( _ Hrvoje Horvat, _ dozvolu ("kakvu?" X))	10	vozačku	B kategorije vozačku	točno
16	Kada Hrvoje Horvat vozi automobilske utrke po svijetu?	vozi ( _ Hrvoje Horvat, _ utrke ( _ automobilske), "kada?" X, _ svijetu ( _ po))	11	u slobodno vrijeme	u slobodno vrijeme	točno
17	Tko u slobodno vrijeme vozi automobilske utrke po svijetu?	vozi ("tko?" X, _ utrke ( _ automobilske), _ vrijeme ( _ u, _ slobodno), _ svijetu ( _ po))	11	Hrvoje Horvat	Hrvoje Horvat	točno
18	Što u slobodno vrijeme vozi Hrvoje Horvat?	vozi ( _ Hrvoje Horvat, "što?" X, _ vrijeme ( _ slobodno, _ u))	11	automobilske utrke	automobilske utrke	Točno
19	Gdje Hrvoje Horvat u slobodno vrijeme vozi automobilske utrke?	vozi ( _ Hrvoje Horvat, _ utrke ( _ automobilske), _ vrijeme ( _ u, _ slobodno), "gdje?" X)	11	po svijetu	po svijetu	Točno
20	Kakve utrke vozi Hrvoje Horvat?	vozi ( _ Hrvoje Horvat, _ utrke ("kakve?" X))	11	automobilske	automobilske	Točno
21	Kada je rođena Jelena Ivić?	je ( _ rođena ( _ Jelena Ivić, "kada?" X))	12	28.05.1980.	28.05.1980.	Točno
22	Gdje je rođena Jelena Ivić?	je ( _ rođena ( _ Jelena Ivić, "gdje?" X))	12	u Varaždinu	u Varaždinu	Točno
23	Tko je rođen 28.05.1980.?	je ( _ rođen ("tko?" X, _ 28.05.1980.))	12	Jelena Ivić	Jelena Ivić	Točno
24	Koji fakultet je završila Jelena Ivić?	je ( _ završila ( _ Jelena Ivić, _ fakultet ("koji?" X)))	13	Tehnički fakultet	tehnički	Točno
25	Tko je završio Tehnički	je ( _ završio ("tko?" X, _	13	Jelena Ivić	Jelena Ivić	Točno

	fakultet?	fakultet ( _ tehnički))				
26	Gdje je radila Jelena Ivić?	je ( _ radila ( _ Jelena Ivić, "gdje?" X))	14	na nekoliko znanstvenih projekata	na nekoliko znanstvenih projekata	Točno
27	Na koliko znanstvenih projekata je radila Jelena Ivić?	je ( _ radila ( _ Jelena Ivić, _ projekata ( _ na, "koliko?" X, _ znanstvenih)))	14	nekoliko	nekoliko	Točno
28	Tko dobro govori njemački jezik	govori ("tko?" X, _ jezik ( _ njemački), _ dobro)	15	Jelena Ivić	Jelena Ivić	Točno
29	Tko programira u Pythonu?	programira ("tko?" X, _ Pythonu ( _ u))	16	Jelena Ivić	Jelena Ivić	Točno
30	Tko ima jedno dijete?	ima ( "tko?" X, _ dijete ( _ jedno))	18	Jelena Ivić	Jelena Ivić	Točno
31	Koliko djece ima Jelena Ivić?	ima ( _ Jelena Ivić, _ djece ("koliko?" X))	18	jedno	jedno	Točno
32	Tko posjeduje vozačku dozvolu?	posjeduje ("tko?" X, _ dozvolu ( _ vozačku))	10	Hrvoje Horvat	Hrvoje Horvat	točno
33	Tko ne posjeduje vozačku dozvolu?	ne ( _ posjeduje ("tko?" X, _ dozvolu ( _ vozačku)))	19	Jelena Ivić	Jelena Ivić	Točno
34	Tko igra odbojku?	igra ("tko?" X, _ odbojku)	20	Jelena Ivić	Jelena Ivić	Točno
35	Kada Jelena Ivić igra odbojku?	igra ( _ Jelena Ivić, _ odbojku, "kada?" X)	20	u slobodno vrijeme	u slobodno vrijeme	Točno
36	Što u slobodno vrijeme radi Jelena Ivić?	X ( (& radi), _ Jelena Ivić, _ vrijeme ( _ u, _ slobodno))	20	igra	Igra	Točno
37	Što u slobodno vrijeme igra Jelena Ivić?	igra ( _ Jelena Ivić, "što?" X, _ vrijeme ( _ u, _ slobodno))	20	odbojku	Odbojku	Točno

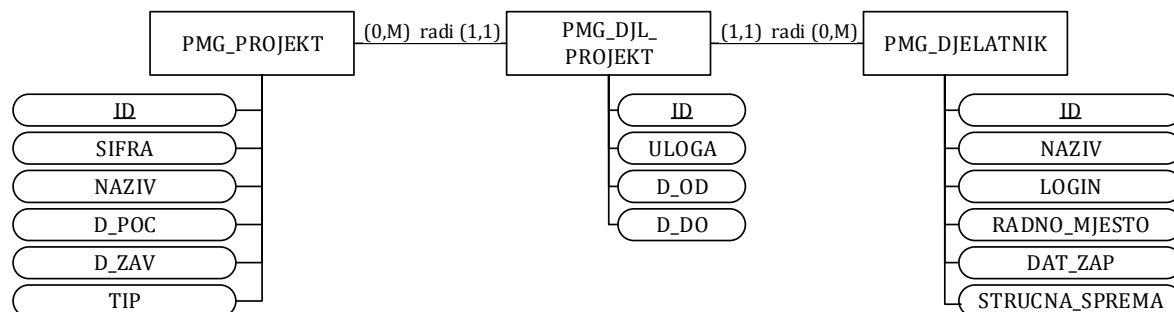
Sustav temeljen na NOK metodi za odgovor na 10. pitanje daje odgovor *I DO NOT KNOW*, što tretiramo kao netočan odgovor, jer sustav nije pronašao odgovor u nekoj od rečenica u bazi. Međutim, ovdje je namjerno postavljeno pitanje koje se odnosi na problem konteksta. U BPNOK postoje znanja o vezi među nekim glagolskim i imeničnim formama, međutim ostali oblici riječi i njihovi odnosi nisu sadržani (sinonimi, antonimi, hiponimi, hiperonimi i slično) te to preostaje za daljnji rad. Samim time, sustav nije niti mogao odgovoriti točno na to pitanje jer nema ugrađena potrebna znanja za taj dio.

\* Točno, ali gramatički nepotpuno - Dobiveni odgovor može se smatrati točnim jer je sadržana ispravna semantika odgovora, međutim potrebno je dodatno formatiranje odgovora. Redosljed riječi u odgovoru ne podudara se s uobičajenom formom gramatike i pravopisa i razlikuje se od redosljeda riječi kojeg bi dao čovjek.

## 8.2 Propitivanje integrirane baze podataka

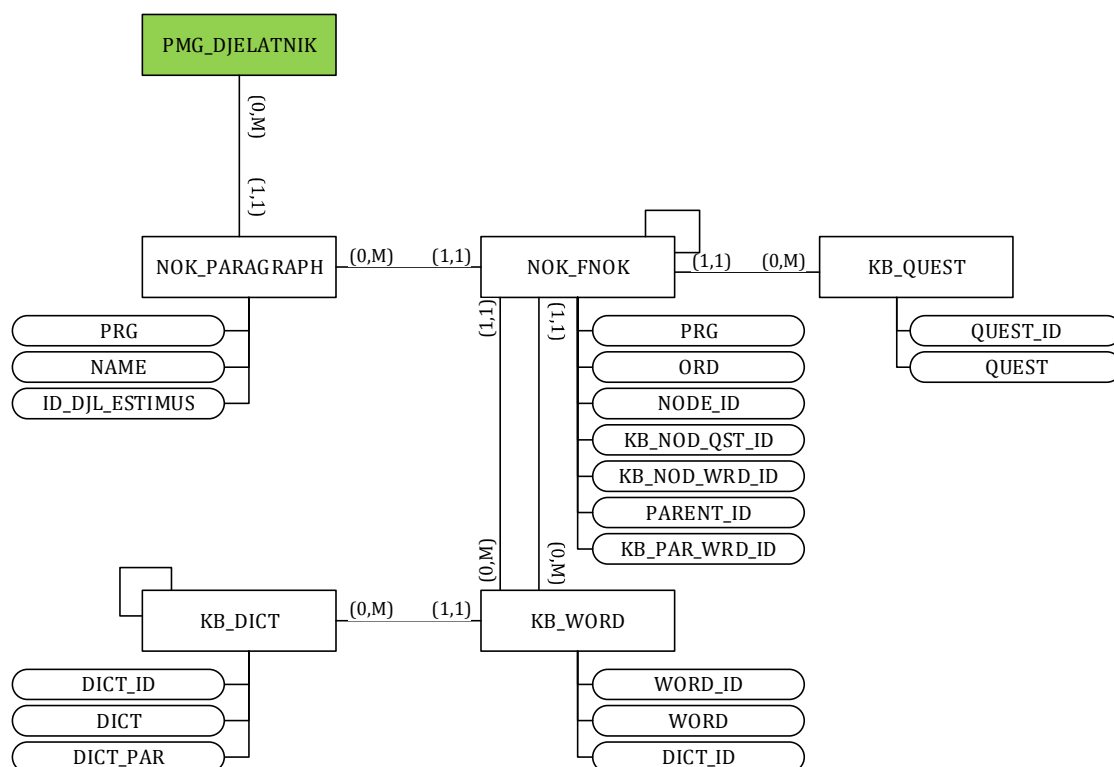
Zbog nemogućnosti propitivanja sheme baze podataka prirodnim jezikom i nemogućnosti upisa rečenica u shemu baze podataka, pokazala se potreba da se postojeći informacijski sustav nadopuni sustavom za prihvata i predstavljanje znanja zapisanih u tekstovima, tj. sustavom koji se temelji na NOK metodi. NOK metoda omogućuje modeliranje znanja, odnosno razvoj mreže znanja za rečenice koje mogu, ali i ne moraju biti međusobno povezane. Dobivenu mrežu znanja iz rečenica prirodnog jezika moguće je transformirati u bazu podataka NOK metode (BPNOK). Time se omogućuje unos tekstualnog znanja u drugu bazu podataka NOK-a koja je povezana s bazom postojećeg informacijskog sustava. Na taj način moguće je upite postavljene prirodnim jezikom transformirati u DNOK te upisati u BPNOK, pretražiti relacijske baze postojećeg IS i BPNOK te dati odgovor na pitanje koji sadrži podatke obje baze.

Proširenjem postojeće baze podataka odabranog informacijskog sustava sustavom temeljenim na NOK-u dobili smo integriranu bazu podataka. Sada je moguće definirati upite koji će biti vezani za obje relacijske baze podataka (bazu podataka postojećeg informacijskog sustava i bazu podataka sustava temeljenog na NOK-u). Osim toga, omogućeno je i dobivanje odgovora na postavljeno pitanje. Slika 8.5 prikazuje implementacijski model postojećeg informacijskog sustava.



Slika 8.5. Implementacijski model postojećeg informacijskog sustava

Slika 8.6 prikazuje implementacijski model sustava temeljenog na NOK-u.



Slika 8.6. Implementacijski model sustava temeljenog na NOK-u

Povezanost ovih dvaju sustava, odnosno njihovih baza realizira se vezom između *PMG\_DJELATNIK* i *NOK\_PARAGRAPH*, odnosno spuštanjem primarnog ključa tablice *PMG\_DJELATNIK* u tablicu *NOK\_PARAGRAPH*, gdje će imati ulogu vanjskog ključa.

Tablica 8.8 sadrži pitanja, odnosno upitne rečenice kojima se propituje integrirana baza podataka. Pitanja su sastavljena tako da se jedan dio u pitanju odnosi na podatke u bazi podataka postojećeg informacijskog sustava, a drugi na znanje iskazano jednostavnim rečenicama prirodnog jezika transformiranih u FNOK zapis i pohranjeno u bazi podataka sustava temeljenog na NOK-u. Da bi odgovorio na pitanje, integrirani sustav treba iskoristiti podatke iz oba sustava. U tablici se još nalaze i odgovori koje je dao čovjek (očekivani odgovori) te odgovori koje je dao integrirani sustav.

Za integrirani sustav pripremljeno je 20 pitanja koja kombiniraju podatke, odnosno znanje iz oba sustava. U 17 pitanja integrirani sustav mora, koristeći znanje iz sustava temeljenog na NOK-u, odgovor tražiti i naći u bazi podataka postojećeg informacijskog sustava (BPEstimus). U tri pitanja, integrirani sustav, koristeći podatke iz BPEstimus, traži odgovor u bazi podataka sustava temeljenom na NOK-u (BPNOK).

U pitanjima je podebljan dio teksta koji je vezan za jednostavne rečenice prirodnog jezika, odnosno odgovor traži u BPNOK.

Tablica 8.8. Pitanja za integriranu bazu podataka

RBR	Pitanje	Odgovor čovjeka	Odgovor integriranog sustava
1	Na kojem projektu radi djelatnik koji <b>programira u Javi?</b>	LUMENS KREDIS	LUMENS KREDIS
2	Na kojem projektu radi djelatnik koji <b>programira u Pythonu?</b>	SMARTY	SMARTY
3	Koji djelatnik na radnom mjestu programera <b>je rođen 01.12.1978.?</b>	Hrvoje Horvat	Hrvoje Horvat
4	Koji djelatnik na radnom mjestu projektanta <b>je rođen 28.05.1980.?</b>	Jelena Ivić	Jelena Ivić
5	<b>Koliko djece ima</b> djelatnik s datumom zaposlenja 15.03.2009.?	dvoje	dvoje
6	<b>Koliko djece ima</b> djelatnik s datumom zaposlenja 01.09.2009.?	jedno	jedno
7	Koji djelatnik <b>je završio Filozofski fakultet u Rijeci</b> i ima login hhorvat?	Hrvoje Horvat	Hrvoje Horvat
8	Koji djelatnik <b>je završio Filozofski fakultet u Rijeci</b> i ima login jivic?	Ne postoji takav djelatnik	nema odgovora
9	Koji djelatnik <b>je završio Tehnički fakultet u Zagrebu</b> i ima login jivic?	Jelena Ivić	Jelena Ivić
10	Na kojem projektu radi djelatnik s VSS koji <b>ima 5 godina radnog iskustva u projektiranju.</b>	LUMENS KREDIS	LUMENS KREDIS
11	Na kojem projektu radi djelatnik s VSS koji <b>je radio na nekoliko znanstvenih projekata.</b>	SMARTY	SMARTY
12	Koju ulogu na projektu čiji je tip projekta "održavanje" ima djelatnik koji <b>je vjenčan?</b>	Voditelj projekta	Voditelj projekta
13	Koji je datum zaposlenja djelatnika koji <b>dobro govori engleski jezik?</b>	15.03.2009.	15.03.2009.
14	Koji je datum zaposlenja djelatnika koji <b>dobro govori njemački jezik?</b>	01.09.2009.	01.09.2009.
15	Koju ulogu na projektu LUMENS ima djelatnik koji <b>posjeduje vozačku dozvolu B kategorije?</b>	Član projektnog tima	Član projektnog tima
16	Koju ulogu na projektu SMARTY ima djelatnik koji <b>ne posjeduje vozačku dozvolu?</b>	Član projektnog tima	Član projektnog tima
17	Na kojem radnom mjestu radi djelatnik koji <b>u slobodno vrijeme vozi automobilske utrke?</b>	programer	Programer
18	Na kojem radnom mjestu radi djelatnik koji <b>u slobodno vrijeme igra odbojku?</b>	projektant	Projektant
19	<b>Što je bio</b> djelatnik koji od 18.09.2014. radi na projektu?	nastavnik matematike u OŠ Rijeka	odgovor prikazan u tablici 8.9
20	Koju ulogu na projektu čiji je tip projekta "razvojni" ima djelatnik koji <b>slabo govori njemački jezik?</b>	Član projektnog tima	Član projektnog tima

## 8.2.1 Odgovori integriranog sustava

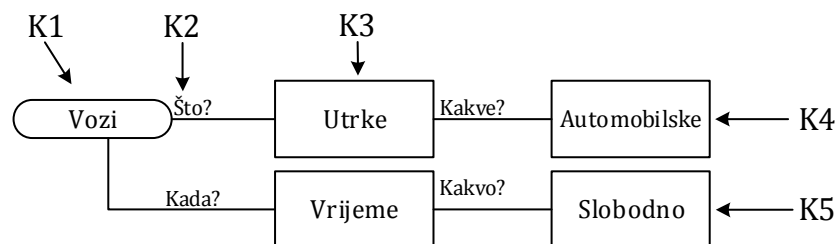
Svi upiti kojima su dobiveni odgovori integriranog sustava nalaze su u Pravitku 4.

Odgovaranje na pitanja integriranog sustava objasniti ćemo na složenijem primjeru za pitanje 17. *Na kojem radnom mjestu radi djelatnik koji u slobodno vrijeme vozi automobilske utrke?* iz tablice 8.8.

Slaganje SQL upita prolazilo je kroz nekoliko faza.

### 1. Obrada pitanja

Pitanje 17. *Na kojem radnom mjestu radi djelatnik koji u slobodno vrijeme vozi automobilske utrke?* sastoji se od dva osnovna dijela. Dio koji se odnosi na sustav Estimus i dio koji se odnosi na sustav temeljen na NOK-u. Dio teksta koji se odnosi na tekstualno znanje iskazano jednostavnim rečenicama prirodnog jezika, koje je upisano u BPNOK, analiziramo prema pravilima pretvorbe u teksta u FNOK ili DNOK. Drugim riječima, dio teksta *slobodno vrijeme vozi automobilske utrke* možemo prikazati u DNOK-u kao na slici 8.7.



Slika 8.7. DNOK za dio teksta

Tekst *slobodno vrijeme vozi automobilske utrke* rastavljamo na dva dijela: *vozi automobilske utrke* i *vozi slobodno vrijeme* s pripadnim ulogama (pitanjima). Time razlikujemo varijable *rijec1* do *rijec5* (podrazumijevaju čvor i procesni čvor) te *quest1* do *quest4* (podrazumijevaju ulogu (pitanje)). Konkretno, vrijednosti koje dodjeljujemo varijablama su:

```
:rijec1 = utrke
:rijec2 = vozi
:rijec3 = automobilske
:rijec4 = vrijeme
:rijec5 = slobodno
:quest1 = što
:quest2 = kakve
:quest3 = kada
:quest4 = kakvo
```

Proces traženja odgovora sastoji se od 6 koraka i to:

1. Traženje oblika riječi (imenica, glagola i sl.)
2. Traženje prvog pitanja (uloge) (uz glagol)
3. Traženje prvog člana s nadređenim pitanjem
4. Filtriranje rezultata cijelog niza podređenih čvorova
5. Filtriranje rezultata više podređenih nizova.

## 2. Upit za traženje oblika riječi

KB\_DICT je rječnik u kojem se nalaze svi hijerarhijski organizirani oblici riječi. KB\_WORD je rječnik svih riječi koje se pojavljuju u tekstu. Svaka riječ u KB\_WORD trebala bi biti povezana na odgovarajući oblik te riječi u KB\_DICT. Upit predviđa nepotpun rječnik tako da riječi osim u hijerarhiji KB\_DICT provjerava i u KB\_WORD. U slučaju kada veza određene riječi na KB\_DICT nije definirana, odgovor na upit je riječ pronađena u KB\_WORD (i to samo u slučaju ako je pronađena).

```
select WRD.WORD_ID
from KB_DICT DCT, KB_WORD WRD
where WRD.DICT_ID = DCT.DICT_ID(+)
and (DCT.DICT_PAR = (select DICT_PAR
from KB_DICT
where upper(DICT) = upper(:rijec))
or upper(WRD.WORD) = upper(:rijec))
```

## 3. Upit za traženje pitanja

Traži se pitanje u KB\_QUESTION.

```
select QUEST_ID from KB_QUESTION where UPPER(QUESTION) = UPPER(:quest)
```

## 4. Upit za traženje čvora

Upit za traženje čvora u kojemu je riječ *vozi*, uloga (pitanje) *što* i odgovor *utrke*:

```
select *
from NOK_FNOK FNOK
where FNOK.KB_NOD_WRD_ID in (select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD
where WRD.DICT_ID = DCT.DICT_ID(+)
and (DCT.DICT_PAR = (select DICT_PAR from KB_DICT where upper(DICT) =
upper(:rijec1)) or upper(WRD.WORD) = upper(:rijec1)))
and FNOK.KB_NOD_QST_ID in (select QUEST_ID from KB_QUESTION where UPPER(QUESTION)
= UPPER(:quest1))
and FNOK.KB_PAR_WRD_ID in (select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD
where WRD.DICT_ID = DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from
KB_DICT where upper(DICT) = upper(:rijec2)) or upper(WRD.WORD) =
upper(:rijec2)))
```

```
:rijec1 = utrke
:rijec2 = vozi
:quest1 = što
```

## 5. Filtriranje rezultata

Pronađene rezultate treba filtrirati tako da ostanu samo oni kojima je podređeni čvor s riječi *utrke*, pitanjem *kakve* i odgovorom *automobilske*.

```
select *
from NOK_FNOK FNOK
where FNOK.KB_NOD_WRD_ID in (select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD
where WRD.DICT_ID = DCT.DICT_ID(+)
and (DCT.DICT_PAR = (select DICT_PAR from KB_DICT where upper(DICT) =
upper(:rijec1)) or upper(WRD.WORD) = upper(:rijec1)))
and FNOK.KB_NOD_QST_ID in (select QUEST_ID from KB_QUESTION where UPPER(QUESTION)
= UPPER(:quest1))
and FNOK.KB_PAR_WRD_ID in (select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD
where WRD.DICT_ID = DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from
KB_DICT where upper(DICT) = upper(:rijec2)) or upper(WRD.WORD) =
upper(:rijec2)))
and (FNOK.PRG, FNOK.ORD, FNOK.NODE_ID) in
(select PRG, ORD, PARENT_ID
from NOK_FNOK
where KB_NOD_WRD_ID in (select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where
WRD.DICT_ID = DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from KB_DICT
where upper(DICT) = upper(:rijec3)) or upper(WRD.WORD) = upper(:rijec3)))
and KB_NOD_QST_ID in (select QUEST_ID from KB_QUESTION where UPPER(QUESTION) =
UPPER(:quest2))
and KB_PAR_WRD_ID in (select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where
WRD.DICT_ID = DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from KB_DICT
where upper(DICT) = upper(:rijec1)) or upper(WRD.WORD) = upper(:rijec1))))

:rijec1 = utrke
:rijec2 = vozi
:rijec3 = automobilske
:quest1 = što
:quest2 = kakve
```

## 6. Dodatno filtriranje rezultata

Na isti način se dobiveni rezultati dodatno filtriraju tako da prikazuju samo one čvorove koji su podređeni početnom čvoru i sadrže izraz *vozi-kada-vrijeme* i *vrijeme-kakvo-slobodno*. Time dobivamo čvor koji je odgovor na dio pitanja vezanog za sustav temeljen na NOK-u.

```
select *
from NOK_FNOK FNOK
where FNOK.KB_NOD_WRD_ID in (select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD
where WRD.DICT_ID = DCT.DICT_ID(+)
and (DCT.DICT_PAR = (select DICT_PAR from KB_DICT where upper(DICT) =
upper(:rijec1)) or upper(WRD.WORD) = upper(:rijec1)))
```



```

    and FNOK.KB_NOD_QST_ID in (select QUEST_ID from KB_QUESTION where UPPER(QUESTION)
= UPPER(:quest1))
    and FNOK.KB_PAR_WRD_ID in (select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD
where WRD.DICT_ID = DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from
KB_DICT where upper(DICT) = upper(:rijec2)) or upper(WRD.WORD) =
upper(:rijec2)))
    and (FNOK.PRG, FNOK.ORD, FNOK.NODE_ID) in
(select PRG, ORD, PARENT_ID
from NOK_FNOK
where KB_NOD_WRD_ID in (select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where
WRD.DICT_ID = DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from KB_DICT
where upper(DICT) = upper(:rijec3)) or upper(WRD.WORD) = upper(:rijec3)))
    and KB_NOD_QST_ID in (select QUEST_ID from KB_QUESTION where UPPER(QUESTION) =
UPPER(:quest2))
    and KB_PAR_WRD_ID in (select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where
WRD.DICT_ID = DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from KB_DICT
where upper(DICT) = upper(:rijec1)) or upper(WRD.WORD) = upper(:rijec1))))
    and (FNOK.PRG, FNOK.ORD, FNOK.PARENT_ID) in
(select PRG, ORD, PARENT_ID
from NOK_FNOK
where KB_NOD_WRD_ID in (select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where
WRD.DICT_ID = DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from KB_DICT
where upper(DICT) = upper(:rijec4)) or upper(WRD.WORD) = upper(:rijec4)))
    and KB_NOD_QST_ID in (select QUEST_ID from KB_QUESTION where UPPER(QUESTION) =
UPPER(:quest3))
    and KB_PAR_WRD_ID in (select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where
WRD.DICT_ID = DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from KB_DICT
where upper(DICT) = upper(:rijec2)) or upper(WRD.WORD) = upper(:rijec2)))
    and (PRG, ORD, NODE_ID) in
(select PRG, ORD, PARENT_ID
from NOK_FNOK
where KB_NOD_WRD_ID in (select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where
WRD.DICT_ID = DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from KB_DICT
where upper(DICT) = upper(:rijec5)) or upper(WRD.WORD) = upper(:rijec5)))
    and KB_NOD_QST_ID in (select QUEST_ID from KB_QUESTION where UPPER(QUESTION) =
UPPER(:quest4))
    and KB_PAR_WRD_ID in (select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where
WRD.DICT_ID = DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from KB_DICT
where upper(DICT) = upper(:rijec4)) or upper(WRD.WORD) = upper(:rijec4))))))

```

Napomena: Dodatno bi trebalo rezultate filtrirati još za izraz *kada-u*. Što je u ovom dijelu zanemareno. Cjelokupan upit koji uključuje cijeli izraz *u slobodno vrijeme vozi automobilske utrke* prikazan je u Privitku 4, 17. pitanje.

## 7. Povezivanje s bazom podataka Estimusa

Pronađeni čvor se povezuje s bazom podataka postojećeg informacijskog sustava (Estimusom) kako bi se dobio odgovor na pitanje – radno mjesto djelatnika.

```

select DJL.RADNO_MJESTO
from NOK_FNOK FNOK, NOK_PARAGRAPH PRG, ESTIMUS.PMG_DJELATNIK DJL
where FNOK.PRG = PRG.PRG

```

```

    and PRG.ID_DJL_ESTIMUS = DJL.ID
    and FNOK.KB_NOD_WRD_ID in (select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where
WRD.DICT_ID = DCT.DICT_ID(+))
and (DCT.DICT_PAR = (select DICT_PAR from KB_DICT where upper(DICT) = upper(:rijec1))
or upper(WRD.WORD) = upper(:rijec1)))
    and FNOK.KB_NOD_QST_ID in (select QUEST_ID from KB_QUEST where UPPER(QUEST) =
UPPER(:quest1))
    and FNOK.KB_PAR_WRD_ID in (select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where
WRD.DICT_ID = DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from KB_DICT where
upper(DICT) = upper(:rijec2)) or upper(WRD.WORD) = upper(:rijec2)))
    and (FNOK.PRG, FNOK.ORD, FNOK.NODE_ID) in
(select PRG, ORD, PARENT_ID
from NOK_FNOK
where KB_NOD_WRD_ID in (select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where
WRD.DICT_ID = DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from KB_DICT where
upper(DICT) = upper(:rijec3)) or upper(WRD.WORD) = upper(:rijec3)))
    and KB_NOD_QST_ID in (select QUEST_ID from KB_QUEST where UPPER(QUEST) =
UPPER(:quest2))
    and KB_PAR_WRD_ID in (select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where
WRD.DICT_ID = DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from KB_DICT where
upper(DICT) = upper(:rijec1)) or upper(WRD.WORD) = upper(:rijec1))))
    and (FNOK.PRG, FNOK.ORD, FNOK.PARENT_ID) in
(select PRG, ORD, PARENT_ID
from NOK_FNOK
where KB_NOD_WRD_ID in (select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where
WRD.DICT_ID = DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from KB_DICT where
upper(DICT) = upper(:rijec4)) or upper(WRD.WORD) = upper(:rijec4)))
    and KB_NOD_QST_ID in (select QUEST_ID from KB_QUEST where UPPER(QUEST) =
UPPER(:quest3))
    and KB_PAR_WRD_ID in (select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where
WRD.DICT_ID = DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from KB_DICT where
upper(DICT) = upper(:rijec2)) or upper(WRD.WORD) = upper(:rijec2)))
    and (PRG, ORD, NODE_ID) in
(select PRG, ORD, PARENT_ID
from NOK_FNOK
where KB_NOD_WRD_ID in (select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where
WRD.DICT_ID = DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from KB_DICT where
upper(DICT) = upper(:rijec5)) or upper(WRD.WORD) = upper(:rijec5)))
    and KB_NOD_QST_ID in (select QUEST_ID from KB_QUEST where UPPER(QUEST) =
UPPER(:quest4))
    and KB_PAR_WRD_ID in (select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where
WRD.DICT_ID = DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from KB_DICT where
upper(DICT) = upper(:rijec4)) or upper(WRD.WORD) = upper(:rijec4))))))

:rijec1 = utrke
:rijec2 = vozi
:rijec3 = automobilske
:rijec4 = vrijeme
:rijec5 = slobodno
:quest1 = što
:quest2 = kakve
:quest3 = kada
:quest4 = kakvo

```

### 8.3 Analiza rezultata

U tablici 8.8 definirano je 20 pitanja na koja integrirani sustav treba dati odgovor uzimajući u obzir podatke iz oba sustava (BPEstimus i BPNOK). Sva pitanja se sastoje od dijela teksta koji je vezan za BPNOK i dijela koji je vezan za BPEstimus.

Pitanje 8. *Koji djelatnik je završio Filozofski fakultet u Rijeci i ima login jivic?*, postavljeno je na način da niti jedan djelatnik ne zadovoljava uvjete, odnosno u bazama podataka nema djelatnika koji je završio Filozofski fakultet u Rijeci i ima login jivic. Odgovor koji bi čovjek dao na ovakvo pitanje je, primjerice, *Ne postoji takav djelatnik*. Integrirani sustav na ovo pitanje ne daje nikakav odgovor, što je u principu i ispravno jer nitko ne zadovoljava uvjete u pitanju.

Pitanje 19. *Što je bio djelatnik koji od 18.09.2014. radi na projektu?* Odgovor pronalazi u BPNOK te se kao rezultat upita dobije tablica 8.9.

**Tablica 8.9. Rezultat upita za pitanje 19.**

	NODE
1	nastavnik
2	matematike
3	OŠ Rijeka
4	u

Dodatno, sustavom temeljenim na NOK-u mogao bi se generirati tekstualan odgovor, međutim to trenutno nije bilo u fokusu istraživanja te preostaje za daljnja istraživanja.

Rezultati upita za sva ostala pitanja (18/20) su u potpunosti ispravni.

Sveukupno gledano, integrirani sustav je ispravno odgovorio na sva pitanja (19/19) koja su odgovor nalazila u BPEstimus. Također, na pitanje koje odgovor pronalazi u BPNOK dobili smo točan odgovor koji je prikazan u obliku tablice, ali poštujući pravila NOK metode može se izgenerirati tekstualan odgovor.

Osim prethodno navedenih 20 pitanja iz tablice 8.8, integriranoj bazi podataka možemo postaviti i sljedeće pitanje.

*Na kojem projektu radi djelatnik koji programira?*

S obzirom na to da možemo imati više djelatnika koji programiraju i rade na više projekata, u ispis ćemo uz *Naziv* projekta iz tablice PMG\_PROJEKT uključiti i *Naziv* djelatnika iz tablice PMG\_DJELATNIK.

SQL upit je:

```
select DJL.NAZIV, PRJ.NAZIV
from NOK_FNOK FNOK, NOK_PARAGRAPH PRG, ESTIMUS.PMG_DJELATNIK DJL,
     ESTIMUS.PMG_DJL_PROJEKT DPR, ESTIMUS.PMG_PROJEKT PRJ
where FNOK.PRG          = PRG.PRG
     and PRG.ID_DJL_ESTIMUS = DJL.ID
     and DPR.ID_DJL       = DJL.ID
     and DPR.ID_PRJ       = PRJ.ID
     and FNOK.KB_NOD_WRD_ID in
     (select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where WRD.DICT_ID =
      DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from KB_DICT where
      upper (DICT) = upper (:rijec1)) or upper (WRD.WORD) = upper (:rijec1)));
```

:rijec1 = programira

Rezultat upita nad integriranom bazom podataka prikazan je u tablici 8.10.

**Tablica 8.10. Rezultat SQL upita nad integriranom bazom podataka**

Naziv	Naziv
Hrvoje Horvat	KREDIS
Jelena Ivić	SMARTY
Hrvoje Horvat	LUMENS

Odgovor na postavljeno pitanje (odnosno ovakvu tablicu) ne bismo mogli dobiti postavljanjem SQL upita koji je vezan samo za bazu podataka postojećeg sustava Estimusa. Na primjer, mogli bismo dobiti odgovor na pitanje *Na kojem projektu radi (koji) djelatnik?* Ili *Koji djelatnik radi na (kojem) projektu?*

SQL upit za postavljeno pitanje je:

```
select DJL.NAZIV, PRJ.NAZIV
from ESTIMUS.PMG_DJELATNIK DJL, ESTIMUS.PMG_DJL_PROJEKT DPR,
     ESTIMUS.PMG_PROJEKT PRJ
where DPR.ID_DJL = DJL.ID
     and DPR.ID_PRJ = PRJ.ID;
```

Time bismo dobili popis djelatnika i naziv projekta na kojima rade (tablica 8.11), odnosno to je najslabije tablici 8.10 što bismo mogli dobiti. Međutim ova tablica sadrži sve djelatnike. Oni djelatnici koji ne zadovoljavaju uvjet (da programiraju) nisu isključeni iz popisa.

**Tablica 8.11. Rezultat SQL upita nad postojećim informacijskim sustavom Estimus**

Naziv (djelatnika)	Naziv (projekta)
Pero Perić	MISUMI
Pero Perić	MZOS
Jelena Ivić	SMARTY
Hrvoje Horvat	LUMENS

Marija Marković	KREDIS
Hrvoje Horvat	KREDIS
Marija Marković	PRIS

#### 8.4 Dokaz hipoteze H3

Hipoteza H3 glasi: *Rezultat proširenja relacijske baze podataka odabranog informacijskog sustava jednostavnim rečenicama zapisanim u BPNOK je integralna semantički bogatija baza podataka informacijskog sustava.*

Hipoteza se smatra potvrđenom ako se na temelju logičkih i implementacijskih modela postojećeg informacijskog sustava i modela sustava temeljenog na NOK metodi razvije integrirani sustav nad kojim će se moći vršiti propitivanje SQL upitima koji će dati ispravne odgovore. Upiti će biti sastavljeni na način da koriste podatke iz obje baze podataka. Ukoliko dobijemo odgovor na upite pokazat ćemo da je rezultat proširenja relacijske baze podataka odabranog informacijskog sustava jednostavnim rečenicama zapisanim u BPNOK, integralna semantički bogatija baza podataka informacijskog sustava. Dodatno, ako postavljanjem upita nad integralnom bazom podataka dobijemo rezultat koji koristi tablice u postojećoj bazi podataka, a taj rezultat ne bismo mogli dobiti samo postavljanjem upita nad postojećom bazom podataka, također možemo zaključiti da je rezultat proširenja relacijske baze podataka odabranog informacijskog sustava jednostavnim rečenicama zapisanim u BPNOK integralna semantički bogatija baza podataka informacijskog sustava.

Pripadni logički modeli su prikazani na slici 8.2. DEV ESTIMUS i slici 8.3. Povezanost NOK-a i Estimusa. Implementacijski modeli prikazani su na slici 8.5. Implementacijski model postojećeg informacijskog sustava i slici 8.6. Implementacijski model sustava temeljenog na NOK-u. Time je dan model integracije dva sustava.

Na temelju tekstova prirodnog jezika prikazanih u tablici 8.5. Životopis Hrvoje Horvat 2201. i tablici 8.6. Životopis Jelena Ivić 2202. popunjene su tablice u BPNOK. Na temelju tih tablica, tablice 8.4. NOK\_PARAGRAPH i tablice BPEstimusa (tablica 8.1. Djelatnik, tablica 8.2. Projekt, tablica 8.3. Djelatnik na projektu) tvori se baza podataka integriranog sustava.

U tablici 8.8 prikazana su pitanja na koja bi integrirani sustav trebao odgovoriti, odgovor koji daje čovjek te odgovor koji je dao integrirani sustav. Usporedbom dvaju odgovora

zaključujemo da integrirani sustav daje u 18/20 (odnosno 90%) slučajeva potpuno točne odgovore. Time je dokazana mogućnost proširenja postojećeg informacijskog sustava Estimus sustavom BPNOK (sustavom temeljenim na NOK metodi) koji sadrži jednostavne rečenice. Takav širi sustav je integralan i semantički bogatiji, što dokazuje hipotezu H3.

U tablici 8.11 prikazan je rezultat SQL upita nad postojećim informacijskim sustavom, a u tablici 8.10 prikazan je rezultat SQL upita nad integriranim sustavom. Oba SQL upita postavljena su za isto pitanje (odnosno koliko je to bilo moguće zbog podataka u bazi).

Rezultat u tablici 8.10 je semantički bogatiji od rezultata u tablici 8.11 te smo time dokazali hipotezu H3.

Dobivanjem odgovora iz integriranog sustava pokazali smo da je rezultat proširenja relacijske baze podataka odabranog informacijskog sustava jednostavnim rečenicama zapisanim u BPNOK, integralna semantički bogatija baza podataka informacijskog sustava te je time hipoteza H3 dokazana.

## 9. Zaključak

U ovom doktorskom radu predstavljeno je istraživanje čiji je cilj obogatiti postojeće informacijske sustave i pripadne im baze podataka sustavom za prihvatanje i predstavljanje znanja zapisanih u tekstovima s jednostavnim rečenicama prirodnog jezika. Naglasak istraživanja je na povezivanju postojeće relacijske baze podataka poslovnog informacijskog sustava s tekstovima napisanim jednostavnim rečenicama prirodnog jezika. Ova integracija omogućuje dodavanje novih znanja vezanih za podatke u bazi podataka postojećeg informacijskog sustava (bez izmjene postojeće sheme baze podataka). Osim toga, omogućuje i postavljanje pitanja na prirodnom jeziku koji će koristeći podatke u obje baze podataka sustava dati odgovor na postavljeno pitanje. Motivacija za ovo istraživanje proizlazi iz činjenice da kada tekstualno znanje iskazano rečenicama prirodnog jezika želimo pohraniti u relacijsku bazu podataka dolazimo do problema kako rečenice prirodnog jezika upisati u relacijsku bazu podataka na način da im je omogućeno semantičko propitivanje. Drugi problem bio je kako propitivati postojeće informacijske sustave upitima definiranim na prirodnom jeziku te kako nova znanja vezana za podatke u bazi podataka postojećeg informacijskog sustava upisati u postojeću bazu podataka bez izmjene sheme baze podataka postojećeg sustava.

U radu su prikazani osnovni pojmovi vezani za područje umjetne inteligencije i predstavljanje znanja. Navedeni su osnovni pristupi za sheme za prikaz znanja s naglaskom na grafičke metode. Također, dan je i pregled područja obrade prirodnog jezika te sustava pitanja i odgovora. Opseg istraživanja je ograničen na jednostavne rečenice hrvatskog i engleskog jezika. U radu su postavljene tri istraživačke hipoteze koje su i potvrđene u ovom doktorskom istraživanju.

Prilikom izrade ovog rada postignuti su sljedeći znanstveni doprinosi: a) Definiranje meta modela sustava za prihvatanje i predstavljanje tekstualnog znanja iskazanog jednostavnim rečenicama prirodnog jezika, b) Definiranje pravila prevođenja koncepata NOK metode u koncepte metode entiteta i veza i obrnuto, c) Definiranje pravila transformacije rečenica prirodnog jezika u FNOK zapis i definiranje pravila transformacije pitanja prirodnog jezika u QFNOK zapis, d) Definiranje i razvoj modela sustava temeljenog na konceptualnom okviru "Node of Knowledge" i e) Oblikovanje

semantički bogatijeg sustava proširenjem postojećeg informacijskog sustava i njegove baze podataka novim znanjima.

Kod pregleda područja koje pokriva istraživanja u području razvoja baza podataka i informacijskih sustava s aspektima povezanim na polje umjetne inteligencije, a posebice obrade prirodnog jezika, uključen je i pregled područja povezan s eksternim izvorima znanja. Iako postoje brojne metode i alati za obradu teksta, ali i OLAP ili *data warehousing* alati koji mogu poboljšati zaključivanje nad podacima u informacijskom sustavu, još uvijek nije riješen problem zapisivanja rečenica prirodnog jezika u relacijsku bazu podataka bez gubitka semantike.

Fokus ovog istraživanja je razvoj konceptualnog okvira koji može prevesti bilo koje tekstualno znanje u podatke pohranjene u relacijskoj bazi podataka, bez obzira na domenu znanja. NOK konceptualni okvir ne koristi statističke pristupe. S obzirom da smo htjeli imati mogućnost integracije bilo kojeg rječnika (na bilo kojem jeziku) s relacijskom bazom podataka, predložena je integracija vlastitog jednojezičnog rječnika s relacijskom bazom podataka koji je kao vanjski izvor znanja povezan s relacijskom bazom podataka. Poseban doprinos rada jesu pravila transformacije rečenica prirodnog jezika u FNOK zapis te pitanja prirodnog jezika u QFNOK zapis za hrvatski i engleski jezik. Pravila prevođenja primjenjuju se na sve riječi u rečenici, a definirana su za glagole, imenice, pridjeve, većinu zamjenica, brojeve, priloge i prijedloge. U ova pravila nisu uključene upitne i odnosne zamjenice jer njihovom upotrebom dolazimo do upitnih i zavisno-složenih rečenica koje izlaze iz ograničenja predmeta istraživanja postavljenog na jednostavne rečenice. Također, nisu obrađeni veznici, usklici i čestice. To je preostalo za daljnju analizu i proširenje pravila.

Važan dio rada su meta modeli NOK metode koji predstavljaju model sustava istraživanja. Definirani su model procesa sustava transformacije teksta u bazu podataka, model hijerarhije tekstualnog znanja povezan s modelom rječnika te meta modeli NOK metode za dvosmjernu i jednosmjernu vezu. Modeli su detaljno objašnjeni. Za oba meta modela NOK-a dani su primjeri popunjavanja tablica u relacijskoj bazi podataka temeljeni na nekoliko rečenica prirodnog jezika. Demonstrirano je postavljanje pitanja i dobivanje odgovora SQL upitima kod jednosmjernog NOK-a.



U zasebnom poglavlju opisani su algoritmi za transformaciju FNOK zapisa u relacijsku bazu podataka, QFNOK zapisa u relacijsku bazu podataka te kreiranje odgovora. Algoritmi su prikazani SQL-om. Rezultati ovog rada bitni su za nastavak istraživanja proširenja postojećih relacijskih baza podataka tekstualnim znanjima.

Pohrana tekstova u baze podataka potrebna je jednako kao i pohrana podataka, ali je do sada bila teško ostvariva. Prednost ovog pristupa postigla se u primjeni NOK metode, koja omogućuje pohranu i izvlačenje znanja. Nedostatak ovog pristupa je u potrebnom prethodnom formaliziranju rečenica NOK metodom. Prilikom razvoja sustava temeljenog na NOK metodi, koristila se jednosmjerna NOK metoda. Rečenice prirodnog jezika u FNOK zapisu i pitanja prirodnog jezika u QFNOK zapisu transformiraju se u relacijsku bazu podataka (BPNOK). Priprema rečenica prirodnog jezika u FNOK zapis, odnosno pitanja prirodnog jezika u QFNOK zapis odvija se prema pravilima transformacije. Rečenice i pitanja su pripremljeni u FNOK i QFNOK zapise pomoću zasebnog programa prevođenja rečenica u FNOK i QFNOK zapis.

U zasebnom poglavlju prikazani su rezultati primjene algoritma transformacije na skupu rečenica prirodnog jezika te skupu pitanja prirodnog jezika u sustavu temeljenom na NOK metodi. Izvršena je transformacija rečenica i pitanja u tablice relacijske baze podataka. Prikazani su rezultati algoritma generiranja odgovora na postavljena pitanja (na temelju polaznih rečenica). Sustav je testiran na tri paragrafa. Prvi paragraf obuhvaćao je 103 jednostavne rečenice hrvatskog jezika koje su predstavljale skup ulaznog znanja. Uz rečenice u sadašnjem glagolskom vremenu, u skup rečenica nad kojim je sustav testiran uključeno je nekoliko rečenica u prošlom i budućem glagolskom vremenu te imperativne rečenice. Također, uvedene su i rečenice koje nemaju izrečen subjekt, ili se on podrazumijeva. Na temelju 103 rečenice na hrvatskom jeziku, definirano je 180 pitanja prirodnog jezika kojima se propitivalo ulazno znanje. Drugi paragraf obuhvaćao je 100 jednostavnih rečenica engleskog jezika koje su predstavljale skup ulaznog znanja te 174 pitanja prirodnog jezika kojim se ulazno znanje propitivalo. Uz rečenice u sadašnjem glagolskom vremenu, u skup rečenica nad kojim je sustav testiran stavili smo nekoliko rečenica u prošlom i budućem glagolskom vremenu te imperativne rečenice. Treći paragraf obuhvaćao je 42 rečenice prirodnog jezika te 88 pitanja. Nakon primjene skupa pitanja na znanje iskazano rečenicama prirodnog jezika dobiveni su odgovori na postavljena pitanja. Dobiveni odgovori su analizirani u ovisnosti

o potpunosti i ispravnosti samog odgovora te usporedbi s odgovorom kojeg bi dao čovjek. Odgovori su kategorizirani u pet kategorija (1. Netočan odgovor, 2. Nedostaje odgovor, 3. Semantički nepotpun odgovor, 4. Semantički ispravan, ali gramatički neispravan odgovor, 5. Točan odgovor) koje možemo grupirati u dvije osnovne kategorije netočni odgovori (1. i 2.) te točni odgovori (3., 4. i 5.). Za slučajeve kada je odgovor iz 3. i 4. kategorije, algoritam za formatiranje odgovora trebalo bi doraditi. Isto tako treba doraditi algoritam za 1. i 2. kategoriju kako bi sustav odgovorio na pitanje i dao točan odgovor. Kod prvog paragrafa, sustav je dao netočan odgovor u 5% slučajeva, a točan u 95% slučajeva. Kod drugog i trećeg paragrafa, postoci su jednaki - sustav je dao netočan odgovor u 3% slučajeva, a točan u 97% slučajeva.

Kao točni odgovori prihvaćani su odgovori koji su u potpunosti identični odgovoru čovjeka. Osim toga, kod pitanja tipa *Što netko radi? / What does someone do?* kao točan odgovor vrednovao se odgovor koji se sastojao samo od procesnog čvora (na primjer, čita, piše i slično). Međutim, čovjek može svojim znanjem na temelju polaznih rečenica na takve odgovore odgovoriti detaljnije, na primjer, čita knjigu, piše zanimljiv roman, ili ponekad dati odgovor cijelom rečenicom. Kao točan odgovor uzimao se onaj odgovor koji zadovoljava semantičku i gramatičku ispravnost (sastoji se od jedne ili više riječi). Daljnjom doradom sustava planira se omogućiti sustavu detaljnije odgovaranje na pitanja i u više varijanti. Na pojedino pitanje moguće je odgovoriti kraćim ili dužim odgovorom pa čak i cijelom rečenicom. U ovom radu bilo je dovoljno da je zadovoljen minimum odgovora, ali da je odgovor semantički i gramatički korektan. Svi ovi uočeni problemi ostaju za doradu algoritama i budući rad.

Treći paragraf testiran je sa svrhom usporedbe algoritama i poboljšane NOK metode sustava temeljenog na NOK metodi s relacijskom bazom podataka, s GBS sustavom koji je testiran nad istim skupom pitanja i odgovora. Pokazalo se da je poboljšanjem pravila transformacije i algoritama sustav temeljen na NOK metodi s relacijskom bazom podataka dao u 2% slučajeva lošiji odgovor, odgovor iste kvalitete u 83% slučajeva te bolji odgovor u 14% slučajeva. Naravno, treba napomenuti da za niti jedan sustav ne možemo reći ni da je bolji, ni da je lošiji. Oba se sustava mogu dalje razvijati i postići bolje rezultate od drugog sustava. Oba su zasnovana na istoj metodi i za njih možemo reći da je napravljen velik pomak u razvoju metode tijekom procesa implementacije.

Zaseban dio istraživanja je proširenje baze podataka postojećeg informacijskog sustava bazom podataka sustava temeljenog na NOK-u. Prikazano je propitivanje ove integrirane baze podataka pitanjima koja kombinira podatke, odnosno znanje iz oba sustava. Pokazalo se da je rezultat proširenja relacijske baze podataka odabranog informacijskog sustava jednostavnim rečenicama zapisanim u BPNOK integralna semantički bogatija baza podataka informacijskog sustava.

Dosadašnjim istraživanjem uočeni su mnogi problemi koji se planiraju riješiti u budućnosti. Neki od planova su: a) dorada pravila transformacije prema vrstama riječi za upitne i odnosne zamjenice, veznike, usklrike i čestice b) dorada algoritama za odgovaranje na pitanja na način da se omoguće višestruki prolasci kroz hijerarhiju kod odgovaranja na pitanja, uključivanje provjere antonima, hiponima, hiperonima i sl., rješavanje problema nabiranja u rečenici, analiza složenih rečenica, analiza više međusobno povezanih rečenica, c) dorada pravila transformacije pitanja u QFNOK za alternativna pitanja d) rješavanje problema homonima i sinonima u rečenicama e) dorada algoritma za odgovaranje na pitanja kod odgovora koji daje samo procesni čvor (omogućavanje potpunijeg odgovora) f) dorada algoritama kod integrirane baze podataka tako da primjenom NOK metode složi odgovor koji je pronašao. Općenito, nastaviti će se raditi na poboljšanju NOK konceptualnog okvira te sustava za integraciju.

## Literatura

- [1] Abulaish, M. i Dey, L. (2007.) Biological relation extraction and query answering from medline abstracts using ontology-based text mining. *Data & Knowledge Engineering*. 61, 2, 228–262.
- [2] Allen, J., Swift, M. i Beaumont, W. (2008.) Deep Semantic Analysis of Text. *Proceedings of the 2008 conference on semantics in text processing*, Stroudsburg, 343–354.
- [3] Allen, J. F., Byron, D. K., Dzikovska, M., Ferguson, G., Galescu, L. i Stent, A. (2001.) Toward conversational human-computer interaction. *AI Magazine*. 22, 4, 27–37.
- [4] Anić, V. (2007.) Rječnik hrvatskoga jezika. Europapress holding d.o.o. i Novi Liber d.o.o.
- [5] Ašenbrener Katić, M., Čandrlić, S. i Pavlić, M. (2017.) Comparison of two versions of formalization method for text expressed knowledge. *Beyond Databases, Architectures and Structures (BDAS) 2017, CCIS (2017)*, prihvaćen za objavljivanje
- [6] Ašenbrener Katić, M., Pavlić, M. i Čandrlić, S. (2015.) The representation of database content and structure using the NOK method. *Procedia Engineering*. 100, 1075-1081.
- [7] Baker, M. C. (2001.) *Syntax. The Handbook of Linguistics* (ed. Aronoff, M; Rees-Miller, J.). Blackwell. 265–295.
- [8] Barić, E., Lončarić, M., Malić, D., Pavešić, S., Peti, M., Zečević, V. i Znika, M. (1997.) *Hrvatska gramatika. Školska knjiga*
- [9] *Basic Knowledge Concepts - Data, Information, Knowledge and Wisdom* (2013.) <http://www.trainmor-knowmore.eu/FBC5DDB3.en.aspx>. (15.03.2013.)
- [10] Biber, D., Johansson, S., Leech, G., Conrad, S. i Finegan, E. (1999.) *Longman grammar of spoken and written english*. Longman.
- [11] *Biblioteka Strani Jezici* (2009.) *Engleska gramatika, English Grammar*. Knjigotisak d.o.o.
- [12] Božić, D. (2006.) *Rječnik englesko-hrvatski, hrvatsko-engleski: s gramatikom*. Marijan tisak.
- [13] Brachman, R. i Levesque, H. (2004.) *Knowledge representation and reasoning. The Morgan Kaufmann Series in Artificial Intelligence*, Morgan Kaufmann.
- [14] Brinton, L.J. i Brinton, D.M. (2010.) *The Linguistic Structure of Modern English*. Philadelphia: John Benjamins B.V.
- [15] Budin, L., Dalbelo Bašić, B., Pavešić, N. i Ribarić, S. (2001.) *Inteligentni sustavi, materijali za seminar kontinuiranog obrazovanja. Inteligentni informatički sustavi, d.o.o.*
- [16] *Cambridge Dictionaries Online (book)* (2015.)

- <http://dictionary.cambridge.org/dictionary/british/book>. (12.03.2015).
- [17] Cambridge Dictionaries Online (land) (2015.)  
<http://dictionary.cambridge.org/dictionary/british/land>. (12.03.2015).
- [18] Chein, M. i Mugnier, M. L. (2009.) Graph-based Knowledge Representation, Computational Foundations of Conceptual Graphs. Springer.
- [19] Chen, Z. Y., Yao, S., Lin, J. Q., Zeng, Y. i Eberlein, A. (2007.) Formalisation of product requirements: from natural language descriptions to formal specifications. International Journal of Manufacturing Research. 2, 3, 362–387.
- [20] Crystal, D. (1997.) The cambridge encyclopedia of language. Cambridge: Cambridge University Press.
- [21] Davenport, T.H. i Prusak, L. (2000.) Working Knowledge. Harvard Business School Press.
- [22] Davis, R., Shrobe, H. i Szolovits, P. (1993.) What is a Knowledge Representation? AI Magazine. 14, 1, 17–33.
- [23] Dendral (2013.) <http://en.wikipedia.org/wiki/Dendral>. (15.09.2013.)
- [24] Dodiya, T. i Jain, S. (2012.) Comparison of question answering systems. Intelligent informatics – proceedings of the international symposium on intelligent informatics ISI'12. Berlin: Springer.
- [25] Dovedan Han, Z. (2013.) FORMALNI JEZICI I PREVODIOCI – prevođenje i primjene. Element.
- [26] Dovedan Han, Z. (2012.) FORMALNI JEZICI I PREVODIOCI – regularni izrazi, gramatike, automati. Element.
- [27] Dovedan Han, Z. (2012.) FORMALNI JEZICI I PREVODIOCI – sintakсна analiza i primjene. Element.
- [28] Exact (2016.) <https://en.oxforddictionaries.com/definition/exact>. (17.02.2016.)
- [29] Fellbaum, C. (1998.) WordNet. Blackwell Publishing Ltd.
- [30] Fougères, A. J. i Trigano, P. (1996.) The formalisation of specifications from specifications written in natural language. Expersys' 96. 369–374.
- [31] Garner, B. J. (1987.) Expert systems: from database to knowledgebase. Information and Software Technology. 29, 2, 60–65.
- [32] Gomez-Perez, A. i Benjamins, V. R. (2002.) Knowledge Engineering and Knowledge Management, Ontologies and the Semantic Web. Proceedings of 13th International Conference, EKAW 2002, Sigüenza, Spain.
- [33] Gottschalk-Mazouz, N. (2007.) Internet and the flow of knowledge: Which ethical and political challenges will we face? Proceedings of the 30. International Ludwig Wittgenstein Symposium. 2, 215–232.
- [34] Greenbaum, S. i Quirk, R. (2006.) A Student's Grammar of the English Language. Longman.
- [35] Grimm, S., Hitzler, P. i Abecker, A. (2007.) Knowledge representation and

ontologies logic. *Ontologies and Semantic Web Languages*.

- [36] Grupa autora (1992.) *Psihologija*. Školska knjiga.
- [37] Gucker, P. (1996.) *Essential English grammar*. Dover Publications, Inc.
- [38] Hallett, C. (2006.) Generic querying of relational databases using natural language generation techniques. *Proceedings of the fourth international natural language generation conference*. 95–102.
- [39] Harmelen, F., Lifschitz, V. i Porter, B. (2008.) *Handbook of Knowledge Representation*. Elsevier.
- [40] Helbig, H. (2006.) *Knowledge Representation and the Semantics of Natural Language*. Springer.
- [41] Hirschman, L. i Gaizauskas, R. (2001.) Natural language question answering: The view from here. *Natural Language Engineering*. 7, 4, 275–300.
- [42] Hrvatska enciklopedija (2015.) [www.enciklopedija.hr](http://www.enciklopedija.hr). (15.09.2016.)
- [43] Hrvatski jezični portal (2014.) <http://hjp.novi-liber.hr/>. (16.01.2014.)
- [44] Infodom d.o.o. (2007.) *Upravljanje znanjem i metodologija uvođenja KM sustava (white paper)*.
- [45] Introduction to RDF. (2000.) <http://www.dbis.informatik.uni-frankfurt.de/~tolle/RDF/DBISResources/RDFIntro.html> (16.06.2013).
- [46] Jach, T. i Xieski, T. (2015.) Inference in expert systems using natural language processing. *Beyond Databases, Architectures and Structures (BDAS)*. Volume 521 of CCIS. Springer Int. Publishing, Berlin Heidelberg, Germany, 288–298.
- [47] Jakupović, A., Pavlić, M. i Dovedan Han, Z. (2014.) Formalisation method for the text expressed knowledge. *Expert Systems with Applications*. 41, 11, 5308–5322.
- [48] Jakupović, A., Pavlić, M., Meštrović, A. i Jovanović, V. (2013.) Comparison of the Nodes of Knowledge method with other graphical methods for knowledge representation. *Proceedings of the 36th international convention /CIS/, MIPRO 2013, Rijeka*, 1276–1280.
- [49] Jespersen, O. (1964.) *Essentials of English grammar*. The University of Alabama Press.
- [50] Kaburlasos, V. G. (2006.) *Towards a Unified Modeling and Knowledge-Representation based on Lattice Theory, Computational Intelligence and Soft Computing Applications Studies*. Springer.
- [51] Kiš, M. (2002.) *Englesko-hrvatski i hrvatsko-engleski INFORMATIČKI RJEČNIK*. Naklada Ljevak.
- [52] Knowledge representation and networked schemes (2013.) [http://stpk.cs.rtu.lv/sites/all/files/stpk/lecture\\_7.pdf](http://stpk.cs.rtu.lv/sites/all/files/stpk/lecture_7.pdf). (11.09.2013.)
- [53] Knowledge representation and reasoning (2013.) [https://en.wikipedia.org/wiki/Knowledge\\_representation\\_and\\_reasoning](https://en.wikipedia.org/wiki/Knowledge_representation_and_reasoning). (15.09.2013.)
- [54] Kovács, L. (2011.) *SQL Generation for Natural Language Interface*. Computer

- Technology and Computer Programming: Research and Strategies. 19–22.
- [55] Lami, G., Gnesi, S., Fabbrini, F., Fusani, M. and Trentanni, G. (2004.) An automatic tool for the analysis of natural language requirements. Informe técnico, CNR Information Science and Technology Institute, Pisa, Italia, Setiembre.
- [56] Leksikologija (2015.) <http://hrvatskijezik.eu/leksikologija/> (10.02.2015)
- [57] Lenat, D. B. (1995.) CYC: A large-scale investment in knowledge infrastructure. *Communications of the ACM*. 38, 11, 33–38.
- [58] Li, F. i Jagadish, H. V. (2014.) NaLIR: An interactive natural language interface for querying relational databases. *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*. ACM. 709–712.
- [59] Li, H. i Shi, Y. (2010.) A WordNet-based natural language interface to relational databases. *The 2nd International Conference on Computer and Automation Engineering, IEEE*. 1, 514–518.
- [60] Liebowitz, J. (2003.) *The Knowledge Management Handbook*. CRC Press LLC.
- [61] Luger, G.F. (2009.) *Artificial intelligence: structures and strategies for complex problem solving*. Pearson Education.
- [62] Lujić, R., Šarić, T. i Šimunović, G. (2007.) Primjena ekspertnog sustava pri određivanju klase prioriteta radnog naloga u pojedinačnoj proizvodnji. *Tehnički vjesnik*. 1, 2, 65–75.
- [63] Lytinen, S. L. (1992.) Conceptual dependency and its descendants. *Computers & Mathematics with Applications*. 23, 2, 51–73.
- [64] Machova, S., Kratochvil, P. i Kleckova, J. (2010.) The elegant mind. A new insight to the deep semantic network. *Annals of DAAAM for 2010 & Proceedings of the 21st International DAAAM Symposium*. 21, 1, 1089–1090.
- [65] Mansuri, I. R. i Sarawagi, S. (2006.) Integrating unstructured data into relational databases. *Proceedings of the 22nd International Conference on Data Engineering, ICDE'06*. IEEE
- [66] Marković, A. (2016.) *Upitne rečenice u novinskim naslovima*, diplomski rad. Sveučilište u Rijeci, Filozofski fakultet u Rijeci.
- [67] Matthews, G., Curtis, J., Hines, K., Kahlert, R. C. i Miraglia, P. (2004.) *Kraken-Knowledge Rich Acquisition of Knowledge from Experts Who Are Non-Logicians*. CYCORP AUSTIN TX.
- [68] Meštrović, A. (2013.) *Prikaz znanja, nastavni materijali*, Odjel za informatiku, Sveučilište u Rijeci.
- [69] Meštrović, A. (2012.) *Semantic Matching Using Concept Lattice*. *Proceedings of Concept Discovery in Unstructured Data, Katholieke Universiteit Leuven, Leuven*, 49–58.
- [70] Meštrović, A. i Čubrilo, M. (2011.) Monolingual dictionary semantic capturing using concept lattice. *International Review on Computers and Software (I. RE. CO*.

- S.). 6, 2, 173–184.
- [71] Mihaljević, M. (2011.) Pitanja u hrvatskome jeziku. Sintaksa hrvatskoga jezika/Književnost i kultura osamdesetih: Zbornik radova 39. seminara Zagrebačke slavističke škole, Filozofski fakultet : Zagrebačka slavistička škola,.
- [72] Miller, G.A. (1995.) WordNet: a lexical database for English. *Communications of the ACM*. 38, 11, 39–41.
- [73] Mollá, D. i Vicedo, J.L. (2007.) Question answering in restricted domains: An overview. *Computational Linguistics*,. 33, 1, 41–61.
- [74] Morgenstern, L. i Thomason, R. H. (2000.) Teaching Knowledge representation: Challenges and Proposals. KR2000 (2000).
- [75] MULTEXT-East - Morphosyntactic Specifications (Version 4). (2010.) <http://nl.ijs.si/ME/V4/msd/html/index.html>. (17.02.2010.)
- [76] MYCIN (2013.) <http://en.wikipedia.org/wiki/MYCIN>. (21.06.2013.)
- [77] Nebel, B. (2001.) *Logics for Knowledge Representation*.
- [78] Nihalani, N. (2010.) An Intelligent Interface for relational databases. *Human-computer interaction*, 6, 7.
- [79] Noun (2017.) <https://www.merriam-webster.com/dictionary/noun>. (03.01.2017.)
- [80] Onyshkevych, B. A. i Nirenburg, S. (1992.) *Lexicon, ontology, and text meaning. Lexical semantics and knowledge representation*. Springer. Berlin. 289–303.
- [81] Oracle (2015.) *SQL- the natural language for analysis*. Oracle white paper.
- [82] Oxford Dictionaries (book), (2015.) <http://www.oxforddictionaries.com/definition/english/book?searchDictCode=all>. (12.03.2015.)
- [83] Pantou, K., Miraglia, P., Salay, N., Kahlert, R.C., Baxter, D. i Reagan, R. (2002.) Knowledge formation and dialogue using the KRAKEN toolset. *AAAI/IAAI*. 900–905.
- [84] Paragraf (2017.) <http://hjp.znanje.hr/index.php?show=search>. (02.01.2017.)
- [85] Paragraf (2017.) <http://proleksis.lzmk.hr/40597/>. (02.01.2017.)
- [86] Pavlić, M. (2011.) *Informacijski sustavi. Školska knjiga*.
- [87] Pavlić, M. (2011.) *Oblikovanje baza podataka*. Odjel za informatiku, Sveučilište u Rijeci.
- [88] Pavlić, M. (2013.) *Razvoj metode za modeliranje znanja*.
- [89] Pavlić, M., Ašenbrener, M., Jakupović, A., Meštović, A., Čandrlić, S. i Ivašić-Kos, M. (2013.) *Inteligentni informacijski sustavi. Razvoj poslovnih i informatičkih sustava CASE*, 25, 25–35.
- [90] Pavlić, M., Dovedan Han, Z. i Jakupović, A. (2015.) Question answering with a conceptual framework for knowledge-based system development “Node of Knowledge.” *Expert Systems with Applications*. 42, 12, 5264–5286.



- [91] Pavlić, M., Jakupović, A. i Čandrlić, S. (2014.) Modeliranje procesa. Odjel za informatiku Sveučilište u Rijeci.
- [92] Pavlić, M., Jakupović, A. i Meštrović, A. (2013.) Nodes of knowledge method for knowledge representation. *Informatologia*. 46, 3, 206–214.
- [93] Pavlić, M., Mestrovic, A. i Jakupović, A. (2013.) Graph-based formalisms for knowledge representation. *Proceedings of the 17th World Multi- Conference on Systemic Cybernetics and Informatics (WMSCI 2013)*, Orlando, 200–204.
- [94] Perovšek, M., Vavpetic, A. i Lavrac, N. (2012.) A wordification approach to relational data mining: Early results. *Late Breaking Papers of the 22nd International Conference on Inductive Logic Programming*, 56–61.
- [95] Petteri Pesonen, J. (2002.) *Concepts and Object-Oriented Knowledge Representation*, MA Thesis. University of Helsinki.
- [96] Pitanja u hrvatskome jeziku (2016.) <http://www.hrvatskiplus.org/article.php?id=1549&naslov=pitanja-u-hrvatskome-jeziku>. (15.10.2016.)
- [97] Poole, D. (2000.) *Logic, Knowledge Representation and Bayesian Decision Theory. Computational Logic — CL 2000*. Springer Berlin Heidelberg. 70–86.
- [98] Prcela, M. (2010.) *Predstavljanje znanja zasnovano na integraciji ontologija i Bayesovih mreža*, doktorska disertacija.
- [99] Proleksis enciklopedija (2013.) <http://proleksis.lzmk.hr/?s=mudrost>. (01.08.2013.)
- [100] Quirk, R., Greenbaum, S., Leech, G. i Svartvik, J. (1985.) *A comprehensive grammar of the English language*. Longman Inc.
- [101] Radovan, M. (2016.) On time and mind: In search of a coherent discourse.
- [102] Raffaelli, I., Tadić, M., Bekavac, B. i Agić, Ž. (2008.) Building Croatian WordNet. *Proceedings of the Fourth Global WordNet Conference*, 349–359.
- [103] Rauker Koch, M., Pavlić, M. i Ašenbrener Katić, M. (2015.) Homonyms and Synonyms in NOK Method. *Procedia Engineering*. 100, 1055–1061.
- [104] Rauker Koch, M., Pavlić, M. i Jakupović, A. (2014.) Application of the NOK method in sentence modelling. *Proceedings of the 37th International Convention MIPRO 2014*, 1426–1431.
- [105] Reddy, R. (1996.) The challenge of artificial intelligence. *Computer*. 10, 10, 86–98.
- [106] Russell, S. i Norvig, P. (2010.) *Artificial Intelligence: A Modern Approach*, Third edition. Prentice Hall.
- [107] Santoso, H. A., Haw, S. C. i Abdul-Mehdi, Z. T. (2011.) Ontology extraction from relational database: Concept hierarchy as background knowledge. *Knowledge-Based Systems*. 24, 3, 457–464.
- [108] Sarawagi, S. (2005.) Models and indices for integrating unstructured data with a relational database. *Knowledge Discovery in Inductive Databases*. Springer Berlin Heidelberg, 1–10.

- [109] Schank, R. C. (1973.) Identification of conceptualizations underlying natural language. Computer models of thought and language, San Francisco, CA: W.H. Freeman. 187–247.
- [110] Silić, J. i Pranjković, I. (2005.) Gramatika hrvatskoga jezika za gimnazije i visoka učilišta. Školska knjiga.
- [111] Singh, P., Lin, T., Mueller, E. T., Lim, G., Perkins, T. i Zhu, W. L. (2002.) Open Mind Common Sense: Knowledge acquisition from the general public. On the move to meaningful internet systems 2002: CoopIS, DOA, and ODBASE, Berlin, 1223–1237.
- [112] Soutou, C. (2001.) Modeling relationships in object relational databases. Data & Knowledge Engineering. 36, 1, 79–107.
- [113] Stanojević, M. i Vraneš, S. (2007.) Knowledge representation with SOUL. Expert Systems with Applications. 33, 122–134.
- [114] Studer, R., Benjamins, V. R. i Fensel, D. (1998.) Knowledge Engineering: Principles and Methods. Data & Knowledge Engineering. 25, 161–197.
- [115] Težak, S. i Babić, S. (2005.) Gramatika hrvatskoga jezika, Priručnik za osnovno jezično obrazovanje. Školska knjiga.
- [116] Thomson, A. J. i Martinet, A. W. (2008.) A Practical English Grammar. Oxford University Press.
- [117] Tomljanović, J., Krsnik, M. i Pavlić, M. (2014.) Inteligentni sustavi pitanja i odgovora. Zbornik Veleučilišta u Rijeci.
- [118] Trask, R.L. (1999.) Language: The Basics (2nd ed.). Psychology Press.
- [119] Trask, R.L. i Stockwell, P. (2007.) Language and linguistics: The key concepts. Taylor & Francis.
- [120] Upravljanje znanjem (2013.) kvaliteta.inet.hr/e-quality/.../6/IFC Upravljanje znanjem.ppt. (12.01.2013.)
- [121] Uvod u umjetnu inteligenciju (2015.) [http://www2.fer.hr/\\_download/repository/UI-1-Uvod.pdf](http://www2.fer.hr/_download/repository/UI-1-Uvod.pdf). (16.03.2015.)
- [122] Uvod u upravljanje znanjem, nastavni materijali: [http://www.fer.unizg.hr/\\_download/repository/UVOD\\_U\\_UPRAVLJANJE\\_ZNANJEM1.pdf](http://www.fer.unizg.hr/_download/repository/UVOD_U_UPRAVLJANJE_ZNANJEM1.pdf). (12.09.2013.)
- [123] Verspoor, M. i Sauter, K. (2000.) English sentence analysis: An introductory course. John Benjamins B.V.
- [124] XCON (2013.) <http://en.wikipedia.org/wiki/XCON> (15.09.2013.)
- [125] Yang, S., Bhowmick, S. S. i Madria, S. (2005.) Bio2X: a rule-based approach for semi-automatic transformation of semi-structured biological data to XML. Data & Knowledge Engineering. 52, 2, 249–271.
- [126] Zadeh, L. A. (2006.) From search engines to question answering systems – the problems of world knowledge, relevance, deduction and precisiation. Fuzzy logic and the semantic web. Amsterdam: Elsevier.
- [127] Zamjenice (2016.) <http://proleksis.lzmk.hr/50932/>. (09.10.2016.)

## Popis slika

Slika 2.1. Hijerarhijski prikaz znanja [9], [60].....	12
Slika 2.2. Jedan pogled na DIKW hijerarhiju (Clark, 2004.) .....	13
Slika 2.3. Osnovni koncepti semantičkih mreža .....	16
Slika 2.4. Primjer semantičke mreže [52] .....	17
Slika 3.1. DNOK za rečenicu <i>Iva vozi auto.</i> .....	35
Slika 3.2. Kontekstna veza između kontekstnog i specijaliziranog čvora.....	35
Slika 3.3. Različite razine mreže znanja.....	35
Slika 3.4. Tri razine znanja.....	36
Slika 3.5. Dvosmjernan DNOK za rečenicu <i>Ivana piše zanimljivu knjigu.</i> .....	38
Slika 3.6. Dvosmjernan DNOK za rečenicu <i>Petra sjedi i čita knjigu.</i> .....	39
Slika 3.7. Jednosmjernan DNOK za rečenicu <i>Ivana piše zanimljivu knjigu.</i> .....	40
Slika 3.8. DEV Osoba u OJ.....	45
Slika 3.9. Hijerarhijska organizacija Organizacijske jedinice (OJ) .....	45
Slika 3.10. Rečenice prikazane DNOK-om.....	48
Slika 3.11. Rečenica u DNOK-u .....	50
Slika 3.12. Rečenica u DEV .....	52
Slika 4.1. Primjer kako jedna riječ otvara mjesto drugoj .....	58
Slika 4.2. Hijerarhijska ovisnost.....	62
Slika 4.3. DNOK za rečenicu <i>Na obje strane svijeta ljudi naporno rade svaki dan.</i> .....	63
Slika 4.4. DNOK za rečenicu <i>On either side of the world people work hard every day.</i> .....	64
Slika 4.5. Tvorba glagolskih vremena u engleskom jeziku i njihova hijerarhija .....	67
Slika 5.1. Model procesa sustava transformacije teksta u bazu podataka .....	94
Slika 5.2. Model hijerarhije tekstualnog znanja povezan s modelom rječnika.....	96
Slika 5.3. Prikaz poglavlja u knjizi S. Russell i P. Norvig, <i>Artificial Intelligence</i> [106] .....	97
Slika 5.4. DEV Raščlane teksta.....	98
Slika 5.5. Meta model NOK-a (model NOKBPT) povezan s dijelom modela raščlane teksta .....	105
Slika 5.6. Pojednostavljeni meta model NOK-a.....	107
Slika 5.7. Pojednostavljeni meta model NOK-a povezan s dijelom modela raščlane teksta .....	107
Slika 5.8. Primjer dvosmjernog NOK-a za rečenicu <i>Petra čita knjigu.</i> .....	108
Slika 5.9. Primjer jednosmjernog NOK-a za rečenicu <i>Petra čita knjigu.</i> .....	108
Slika 5.10. Meta model NOK-a za jednosmjernu vezu.....	108
Slika 5.11. DNOK-ovi za rečenice prirodnog jezika .....	110
Slika 5.12. DNOK-ovi za rečenice prirodnog jezika i rečenice zadane FNOK-om.....	114
Slika 6.1. Model sustava temeljenog na NOK-u.....	121
Slika 6.2. Meta model strukture i sadržaja FNOK zapisa.....	123
Slika 6.3. Prikaz transformacije formaliziranog teksta FNOK u relacijsku bazu podataka .....	130
Slika 6.4. Meta model strukture i sadržaja QFNOK zapisa.....	136

Slika 6.5. Prikaz transformacije QFNOK zapisa u relacijsku bazu podataka .....	138
Slika 6.6. Meta model strukture i sadržaja odgovora .....	140
Slika 6.7. Model procesa niže razine za proces 3. Generiranje odgovora na pitanje.....	144
Slika 7.1. Rezultati testiranja sustava nad rečenicama hrvatskog jezika.....	157
Slika 7.2. Rezultati testiranja sustava nad rečenicama engleskog jezika.....	164
Slika 7.3. Analiza odgovora za GBS i RDBS sustava.....	175
Slika 7.4. DNOK za rečenicu <i>Katarina vozi crveni auto na autocesti</i> .....	177
Slika 8.1. Proširenje baze podataka postojećeg informacijskog sustava bazom podataka sustava temeljenog na NOK-u.....	181
Slika 8.2. DEV ESTIMUS.....	181
Slika 8.3. Povezanost NOK-a i Estimusa.....	182
Slika 8.4. DNOK za rečenice iz tablice 8.5 za životopis Hrvoja Horvata .....	187
Slika 8.5. Implementacijski model postojećeg informacijskog sustava.....	191
Slika 8.6. Implementacijski model sustava temeljenog na NOK-u.....	192
Slika 8.7. DNOK za dio teksta .....	194
Slika 1. Model sustava QANOK .....	289

## Popis tablica

Tablica 2.1. Definicije umjetne inteligencije organizirane u četiri kategorije .....	8
Tablica 2.2. Podjela shema, tehnika i jezika za prikaz znanja prema osnovnim pristupima .....	15
Tablica 2.3. Primjer okvira [68] .....	18
Tablica 3.1. Osnovni koncepti NOK metode .....	32
Tablica 3.2. Osoba .....	46
Tablica 3.3. Inženjer .....	46
Tablica 3.4. Vozač .....	46
Tablica 3.5. Prodavač .....	47
Tablica 3.6. Dijete .....	47
Tablica 3.7. Ima dijete .....	47
Tablica 3.8. Tip nagrade .....	47
Tablica 3.9. Nagrada .....	47
Tablica 3.10. Organizacijska jedinica .....	47
Tablica 3.11. Preslikavanje metode entiteta i veza u NOK metodu .....	49
Tablica 3.12. OJ .....	53
Tablica 3.13. Osoba .....	53
Tablica 3.14. Inženjer .....	53
Tablica 3.15. Vozač .....	53
Tablica 3.16. Nagrada .....	54
Tablica 3.17. Je dobila .....	54
Tablica 3.18. Preslikavanje NOK metode u metodu entiteta i veza .....	54
Tablica 4.1. Primjeri za modalne glagole te različita glagolska vremena u hrvatskom i engleskom jeziku .....	68
Tablica 4.2. Primjeri Da/Ne pitanja .....	89
Tablica 4.3. Primjer rečenica s pripadajućim FNOK zapisima na hrvatskom (A dio) i engleskom (B dio) jeziku .....	91
Tablica 4.4. Upitne rečenice s pripadajućim QFNOK zapisima i odgovorima temeljenim na znanju iz tablice 4.3 na hrvatskom (A dio) i engleskom (B dio) jeziku .....	91
Tablica 4.5. Upitne rečenice, kod kojih je traženi odgovor na najvišoj razini hijerarhije, s pripadajućim QFNOK zapisima i odgovorima temeljenim na znanju iz tablice 4.3.....	93
Tablica 5.1. Rečenice prirodnog jezika i njihov FNOK zapis.....	109
Tablica 5.2. ČVOR.....	110
Tablica 5.3. ULOGA (PITANJE).....	111
Tablica 5.4. ČVOR_TIP .....	111
Tablica 5.5. PITANJA.....	111
Tablica 5.6. Rečenice prirodnog engleskog jezika i njihov FNOK zapis .....	113
Tablica 5.7. Čvor_tip .....	115
Tablica 5.8. Čvor.....	115
Tablica 5.9. Procesni čvor.....	116

Tablica 5.10. Uloga (pitanje) .....	117
Tablica 5.11. Veza .....	117
Tablica 5.12. Procesna veza.....	118
Tablica 6.1. Različite kombinacije za Parent, Node i Question u tablici KB_QUESTION_QUEST .....	142
Tablica 6.2. Različite kombinacije za Parent, Node i Question u tablici KB_QUESTION_QUEST za identifikatore .....	142
Tablica 7.1. Popis paragrafa .....	149
Tablica 7.2. NOK_FNOK_ALL .....	149
Tablica 7.3. KB_QUESTION.....	149
Tablica 7.4. KB_DICT .....	149
Tablica 7.5. KB_WORD .....	150
Tablica 7.6. NOK_FNOK – popis čvorova .....	150
Tablica 7.7. KB_TEXT_QUESTION – popis veza.....	150
Tablica 7.8. NOK_QFNOK_ALL .....	151
Tablica 7.9. NOK_QFNOK – popis čvorova .....	151
Tablica 7.10. KB_QUESTION_QUESTION – popis veza .....	152
Tablica 7.11. KB_ANSWER .....	152
Tablica 7.12. Isječak THJ rečenica i njihovih pripadnih FNOK-ova.....	155
Tablica 7.13. Isječak THJ pitanja i njihovih pripadnih QFNOK-ova .....	155
Tablica 7.14. Isječak odgovora sustava.....	156
Tablica 7.15. Isječak TENG rečenica i njihovih pripadnih FNOK-ova .....	161
Tablica 7.16. Isječak QTENG pitanja i njihovih pripadnih QFNOK-ova .....	161
Tablica 7.17. Isječak odgovora sustava.....	161
Tablica 7.18. Razlika u FNOK zapisu u ovisnosti o gramatici.....	169
Tablica 7.19. Isječak rečenica i FNOK-ova u oba sustava.....	170
Tablica 7.20. Isječak pitanja i QFNOK-ova u oba sustava .....	170
Tablica 7.21. Promjene uvedene u pravila prevođenja rečenica u FNOK.....	172
Tablica 7.22. Promjene uvedene u pravila prevođenja rečenica u FNOK - Rječnik.....	172
Tablica 7.23. Promjene uvedene u pravila prevođenja pitanja u QFNOK .....	172
Tablica 7.24. Neki od odgovora primljenih iz sustava .....	173
Tablica 7.25. Analiza odgovora .....	174
Tablica 7.26. Kvaliteta odgovora RDBS sustava u odnosu na GBS sustav .....	175
Tablica 7.27. Pitanja postavljena za jednu rečenicu i pripadni QFNOK-ovi.....	177
Tablica 7.28. Odgovora na pitanja temeljeni na jednoj rečenici.....	178
Tablica 8.1. Djelatnik.....	183
Tablica 8.2. Projekt .....	183
Tablica 8.3. Djelatnik na projektu .....	183
Tablica 8.4. NOK_PARAGRAPH.....	184
Tablica 8.5. Životopis Hrvoje Horvat 2201.....	185
Tablica 8.6. Životopis Jelena Ivić 2202.....	185
Tablica 8.7. Pitanja i QFNOK zapisi za životopise .....	188

Tablica 8.8. Pitanja za integriranu bazu podataka.....	193
Tablica 8.9. Rezultat upita za pitanje 19.....	199
Tablica 8.10. Rezultat SQL upita nad integriranom bazom podataka.....	200
Tablica 8.11. Rezultat SQL upita nad postojećim informacijskim sustavom Estimus .....	200
Tablica 1. Proces 1. 1. NOK_Import_Data().....	222
Tablica 2. Algoritam procedure NOK_Hierarchy() .....	222
Tablica 3. Proces 1. NOK_Hierarchy_Start().....	223
Tablica 4. Proces 1.2. NOK_Part_Divide().....	224
Tablica 5. Algoritam funkcije NOK_Part_Write().....	225
Tablica 6. Algoritam funkcije NOK_Part_Read().....	225
Tablica 7. Proces 1.3. NOK_Hierarchy_Insert().....	226
Tablica 8. Proces 1.4. NOK_Hierarchy_Create().....	226
Tablica 9. Algoritam procedure KB_Dictionary().....	227
Tablica 10. Algoritam procedure KB_List_Words().....	228
Tablica 11. Upit za vizualni prikaz hijerarhije rječnika .....	229
Tablica 12. Algoritam procedure KB_Update_FNOK() .....	229
Tablica 13. Algoritam procedure KB_Create_QUEST() .....	231
Tablica 14. Algoritam procedure KB_Create_Answer().....	231
Tablica 15. Algoritam funkcije KB_Get_Answer_Word().....	236
Tablica 16. Algoritam funkcije KB_Hierarchical_Answer().....	237
Tablica 17. Algoritam funkcije KB_Write_Answer() .....	238
Tablica 18. Algoritam funkcije KB_Complete_Answers() .....	238
Tablica 19. Odabrane rečenice hrvatskog jezika i pripadni im FNOK-ovi.....	239
Tablica 20. Pitanja prirodnog hrvatskog jezika i pripadni QFNOK-ovi .....	243
Tablica 21. Odgovori sustava na pitanja (hrvatski jezik).....	248
Tablica 22. Odabrane rečenice engleskog jezika i pripadni FNOK-ovi .....	257
Tablica 23. Pitanja engleskog jezika i pripadni QFNOK-ovi.....	261
Tablica 24. Odgovori sustava na pitanja (engleski jezik).....	266
Tablica 25. TENG rečenice i pripadni FNOK zapisi u oba sustava.....	275
Tablica 26. QTENG pitanja i pripadni QFNOK zapisi u oba sustava.....	278
Tablica 27. Odgovori dvaju sustava na pitanja te analiza rezultata dvaju sustava.....	282
Tablica 28. Tablica prijelaza prepoznavača jezika TENG.....	291
Tablica 29. Rezultat SQL upita za 1. pitanje .....	295
Tablica 30. Rezultat SQL upita za 2. pitanje .....	295
Tablica 31. Rezultat SQL upita za 3. pitanje .....	296
Tablica 32. Rezultat SQL upita za 4. pitanje .....	296
Tablica 33. Rezultat SQL upita za 5. pitanje .....	297
Tablica 34. Rezultat SQL upita za 6. pitanje .....	297
Tablica 35. Rezultat SQL upita za 7. pitanje .....	299
Tablica 36. Rezultat SQL upita za 8. pitanje .....	299
Tablica 37. Rezultat SQL upita za 9. pitanje .....	299
Tablica 38. Rezultat SQL upita za 10. pitanje.....	301

Tablica 39. Rezultat SQL upita za 11. pitanje.....	303
Tablica 40. Rezultat SQL upita za 12. pitanje.....	304
Tablica 41. Rezultat SQL upita za 13. pitanje.....	305
Tablica 42. Rezultat SQL upita za 14. pitanje.....	305
Tablica 43. Rezultat SQL upita za 15. pitanje.....	307
Tablica 44. Rezultat SQL upita za 16. pitanje.....	308
Tablica 45. Rezultat SQL upita za 17. pitanje.....	309
Tablica 46. Rezultat SQL upita za 18. pitanje.....	311
Tablica 47. Rezultat SQL upita za 19. pitanje.....	312
Tablica 48. Rezultat SQL upita za 20. pitanje.....	313



## Popis privitaka

Privitak 1. SQL skripte za sustav BPNOK .....	222
Privitak 2. Tablice.....	239
Privitak 3. Formalizacija rečenica i pitanja .....	289
Privitak 4. SQL upiti nad integriranom bazom podataka.....	294

# Privitak 1. SQL skripte za sustav BPNOK

Tablica 1. Proces 1. 1. NOK\_Import\_Data()

```
-----  
Procedure    NOK_Import_Data (   
                PrgId IN Number      -- Paragraph of Sentences/Questions  
                )  
  
Is  
Begin  
--  
-->>> PROCESS 1.1  
-- Cancel old Data of the same Paragraph  
Delete  
From    NOK_FNOK_ALL  
Where   PRG = PrgId      -- Paragraph to delete  
;  
--  
Delete  
From    NOK_QFNOK_ALL  
Where   PRG = PrgId      -- Paragraph to delete  
;  
-- Insert new Data of the same Paragraph  
Insert  
Into    NOK_FNOK_ALL  
      (   
        PRG  
      , ORD  
      , TENG  
      , FNOK  
      )  
  
Select  PrgId                -- Paragraph to insert  
      , ROWNUM + (Select count(*) From NOK_FNOK_ALL Where PRG = PrgId) -- Order Number  
                                                between same Paragraph  
      , TENG  
      , FNOK  
From    NOK_FNOK_IMP  
;  
--  
Insert  
Into    NOK_QFNOK_ALL  
      (   
        PRG  
      , ORD  
      , QTENG  
      , QFNOK  
      )  
  
Select  PrgId                -- Paragraph to insert  
      , ROWNUM + (Select count(*) From NOK_QFNOK_ALL Where PRG = PrgId) -- Order  
                                                Number between same Paragraph  
      , QTENG  
      , QFNOK  
From    NOK_QFNOK_IMP  
;  
--  
End;  
-----
```

Tablica 2. Algoritam procedure NOK\_Hierarchy()

```
-----  
Procedure    NOK_Hierarchy (   
                PrgId IN Number      -- Paragraph of Sentences / Questions  
                )  
-----
```

```

)
Is
  TNOK      Varchar2(5);  -- Type of NOK 'FNOK' or 'QFNOK'
  NumR      Number;      -- Number of Transformed Records
--
Begin
--
---- Create Hierarchy for FNOK
--
  TNOK := 'FNOK';
  NumR := NOK_Hierarchy_Start(TNOK, PrgId);
--
---- Create Hierarchy for QFNOK
--
  TNOK := 'QFNOK';
  NumR := NOK_Hierarchy_Start(TNOK, PrgId);
--
End;
-----

```

**Tablica 3. Proces 1. NOK\_Hierarchy\_Start()**

```

-----
Function      NOK_Hierarchy_Start (
              TNOK IN Varchar2    -- Type of NOK 'FNOK' or 'QFNOK'
              , PrgId IN Number   -- Paragraph of Sentences/Questions
              )
Return Number Is
-----
--Prepare Data and Start Transformation of a FNOK / QFNOK record into Hierarchical order.
-----
  NOKID      Number;          -- NOK IDENTIFIER (used as SEQUENCE)
  Ret        Number;          -- Return Value
  NumS        Number;          -- Number of Sentences / Questions
  OrdNo       Number;          -- Sentence / Question Order Number
  FNOK        Varchar2(400);   -- FNOK string to divide into syntactical Parts
  NumP        Number;          -- Number of syntactical Parts
  Pos         Number;          -- Actual Position
  NodId       Number;          -- Node Identifier
  ParId       Number;          -- Parent Identifier
  Part        Varchar2(99);    -- Single Syntactical Part
  Quest       Varchar2(99);    -- Question
--
Begin
-- Prepare tables and data
--
-- cancel all records from hierarchical table NOK_FNOK / NOK_QFNOK belonging to requested
Paragraph
  If      TNOK = 'FNOK'      Then      Delete      From      NOK_FNOK      Where      PRG = PrgId;
  ElseIf TNOK = 'QFNOK'     Then      Delete      From      NOK_QFNOK     Where      PRG = PrgId;
  Else   Return 0;
  End If;
-- select Number of Sentences / Questions at requested Paragraph
  If TNOK = 'FNOK' Then Select count(*) Into NumS From NOK_FNOK_ALL Where PRG =
PrgId;
  Else                Select count(*) Into NumS From NOK_QFNOK_ALL Where PRG =
PrgId;
  End If;
--
-- Transforming all FNOK / QFNOK records from the Paragraph in their Hierarchical structure
--
  For OrdNo in 1 .. NumS Loop    -- MAIN LOOP - FOR EACH FNOK
  --
-- get a FNOK record
  If TNOK = 'FNOK' Then Select FNOK Into FNOK From NOK_FNOK_ALL Where

```

```

PRG = PrgId And ORD = OrdNo;
  Else
    Select QFNOK || ' ' Into FNOK From NOK_QFNOK_ALL Where
PRG = PrgId And ORD = OrdNo;
  End If;
--
-->>> PROCESS 1.2
-- Divide a FNOK record in his syntactical Parts
  NumP := NOK_Part_Divide(FNOK);
  If NumP = 0 Then
    Return 0;
  End If;
--
-->>> PROCESS 1.3
-- Prepare a First Node
  NOKID := 0; -- NOK IDENTIFIER initialization
  ParId := NOKID; -- Parent Identifier for first Node
  NOKID := NOKID + 1; -- NOK IDENTIFIER increment
  NodId := NOKID; -- Node Identifier for first Node
  Quest := '_'; -- first Question is empty

-- First syntactical Part must be a VERB (but we are not verifying a syntax)
  Pos := 1; -- first Position between
  -- syntactical Parts of FNOK record
  Part := NOK_Part_Read(Pos); -- read a First syntactical Part
  Ret := NOK_Hierarchy_Insert(TNOK, PrgId, OrdNo, ParId, NodId, Part, Quest); --
  -- INSERT THIS NODE

--
-->>> PROCESS 1.4
-- Look for Second syntactical Part that must be '('
  If Pos < NumP Then
    Pos := Pos + 1;
    Part := NOK_Part_Read(Pos); -- read a Second syntactical Part
    If Part = '(' Then -- if next syntactical Part is
      -- equal to '(' then can continue on the next level of hierarchy
-- Go to the next level of hierarchy
    Pos := NOK_Hierarchy_Create(TNOK, NOKID, PrgId, OrdNo, NodId, NumP, Pos);-- go
    -- to the next level of hierarchy

    End If;
  End If;
End Loop; -- END MAIN LOOP - FOR EACH FNOK
--
Return NOKID; -- return Number of Transformed Records
End;
-----

```

**Tablica 4. Proces 1.2. NOK\_Part\_Divide()**

```

-----
Function NOK_Part_Divide (
          FNOK IN Varchar2-- FNOK string to divide into syntactical Parts
        )
Return Number Is
-----
-- Divide a FNOK string into his syntactical Parts
-----
--
Chr1 Varchar (02); -- Single Character
Chr2 Varchar (02); -- Single Character
NumP Number; -- Number of syntactical Parts
Part Varchar2(99); -- Single Syntactical Part
--
Begin
Delete From NOK_PART ; -- cancel a temporary table NOK_PART
--
NumP := 0;

```

```

Part := NULL;
--
For Pos In 1 .. length(FNOK) Loop      -- for each character in the FNOK string
--
Chr1 := substr(FNOK, Pos, 1);          -- get a character
If Chr1 In ('&','_') Then             -- characters '&' and '_' can be a part of
                                     -- text, not only a syntactical delimiters
    Chr2 := substr(FNOK, Pos+1, 1);    -- we must check the successive character
End If;
--
If Chr1 In (('(',')',' ','''')       -- syntactical divisors are '(' ')' ' ','
            '""')
Or (Chr1 In ('&','_') And Chr2 = ' ') -- but also '&' '_' only if the successive
                                     -- character is space
Then
    -- if character is one of the syntactical
    -- divisors (punctuation)
    NumP := NOK_Part_Write(NumP, Part); -- write a precedent Part
    NumP := NOK_Part_Write(NumP, Chr1); -- write the actual syntactical divisor
    Part := NULL;                       -- prepare a buffer for new syntactical Part
Else
    -- if character is one normal text character
    Part := Part || Chr1;               -- add the character to the syntactical Part
End If;
--
End Loop;
--
Return NumP;                          -- Return Number of syntactical Parts
End;
-----

```

**Tablica 5. Algoritam funkcije NOK\_Part\_Write()**

```

-----
Function    NOK_Part_Write ( NumP IN Number    -- Number of precedent syntactical Parts
                             , Part IN Varchar2 -- New syntactical Part
                             )
Return     Number Is
-----
-- Write a new syntactical Part into the temporary table NOK_PART.
-----
--
Begin
If trim(Part) Is NULL Then    -- if Part is NOT OK exit
    Return NumP;
End If;

-- increment Number of Parts and write the Part
Insert Into    NOK_PART    ( ORD    , PART    )
Values        ( NumP + 1, trim(Part) );

Return NumP + 1;            -- Return Number of syntactical Parts
--
End;
-----

```

**Tablica 6. Algoritam funkcije NOK\_Part\_Read()**

```

-----
Function    NOK_Part_Read (
                             NumP IN Number    -- Number of precedent syntactical Parts
                             )
Return     Varchar2 Is
-----
-- Read requested syntactical Part of FNOK from temporary table NOK_PART.
-----

```

```

    Part    Varchar2(99);      -- Single Syntactical Part
Begin
--
    Select  trim(PART)
Into      Part
From      NOK_PART
Where     ORD = NumP;
--
    Return Part;              -- Return Number of syntactical Parts
End;
-----

```

**Tablica 7. Proces 1.3. NOK\_Hierarchy\_Insert()**

```

-----
Function    NOK_Hierarchy_Insert (
            TNOK IN Varchar2  -- Type of NOK 'FNOK' or 'QFNOK'
            , PrgId IN Number  -- Paragraph of Sentences/Questions
            , OrdNo IN Number  -- Order Number
            , ParID IN Number  -- Parent Identifier
            , NodID IN Number  -- Node Identifier
            , Node  IN Varchar2 -- Node content
            , Quest IN Varchar2 -- Question
            )
Return Number Is
-----
-- Insert one hierarchical Node to hierarchical table NOK_FNOK or NOK_QFNOK.
-----
Begin
--
    If    TNOK = 'FNOK' Then
        Insert Into    NOK_FNOK    ( PRG , ORD , PARENT_ID, NODE_ID, NODE, QUEST )
        Values         ( PrgId, OrdNo, ParId  , NodId  , Node, Quest );
    ElseIf TNOK = 'QFNOK' Then
        Insert Into    NOK_QFNOK   ( PRG , ORD , PARENT_ID, NODE_ID, NODE, QUEST )
        Values         ( PrgId, OrdNo, ParId  , NodId  , Node, Quest );
    End If;
--
    Return 1;
End;
-----

```

**Tablica 8. Proces 1.4. NOK\_Hierarchy\_Create()**

```

-----
Function    NOK_Hierarchy_Create (
            TNOK IN    Varchar2  -- Type of NOK 'FNOK' or 'QFNOK'
            , NOKID IN OUT Number  -- NOK IDENTIFIER (used as
            SEQUENCE)
            , PrgID IN    Number  -- Paragraph of Sentences /
            Questions
            , OrdNo IN    Number  -- Order Number
            , ParID IN    Number  -- Parent Identifier
            , NumP  IN    Number  -- Number of syntactical Parts
            , Pos   IN OUT Number  -- precedent Position
            )
Return Number Is
-----
-- Transform a FNOK - QFNOK record in Hierarchical order.
-----
--
    Ret    Number;      -- Return Value
    NodId  Number;      -- Node Identifier
    Part   Varchar2(99); -- Single Syntactical Part

```

```

    Quest  Varchar2(99); -- Question
--
Begin
--
    While Pos < NumP Loop                -- for all - until the last syntactical Part
        Pos := Pos + 1;
        Part := NOK_Part_Read(Pos);      -- read next syntactical Part
        If Part = '(' Then                -- if divisor is equal to '('
                                            -- then can continue on the next level of
                                            -- hierarchy (recursive call)
            Pos := NOK_Hierarchy_Create(TNOK, NOKID, PrgId, OrdNo, ParID, NumP, Pos);
            Continue;
        ElseIf Part = ')' Then Exit;      -- if divisor is equal to ')' then it is the
                                            -- end of this level of hierarchy
        ElseIf Part = ',' Then Continue;  -- if Part is equal to ',' we skip to the
                                            -- next Part
        End If;
--
-- -- we are looking for a QUESTION
    If Part = '_'                         -- character '_' substitute the Question to be
                                            -- answered
    Or Part = '&' Then                    -- character '&' substitute the Question (can
                                            -- be skipped)
        Quest := Part;
    Else                                   -- character '"' open the Question
        Pos := Pos + 1;
        Part := NOK_Part_Read(Pos); -- read QUESTION
        Quest := substr(Part, 1, length(Part) - 1); -- QUESTION without '?'
        Pos := Pos + 1;
        Part := NOK_Part_Read(Pos); -- character '"' close the Question
    End If;
--
    Pos := Pos + 1;
    Part := NOK_Part_Read(Pos); -- read PART as NODE - ANSWER
    NOKID := NOKID + 1; -- NOK IDENTIFIER increment
    NodId := NOKID; -- Node Identifier for first Node
    Ret := NOK_Hierarchy_Insert(TNOK, PrgId, OrdNo, ParId, NodId, Part, Quest);
                                            -- INSERT THIS NODE
    Pos := Pos + 1;
    Part := NOK_Part_Read(Pos); -- read next syntactical divisor (punctuation)
    If Part = '(' Then                -- if divisor is equal to '(' then can
                                            -- continue on the next level of hierarchy (recursive call)
        Pos := NOK_Hierarchy_Create(TNOK, NOKID, PrgId, OrdNo, NodId, NumP, Pos);
    ElseIf Part = ')' Then            --if divisor is equal to ')' then it is the
                                            -- end of this level of hierarchy
        Exit;
    Else                               -- repeat the last read
        Pos := Pos - 1;
    End If;
--
    End Loop;
    Return Pos; -- return Actual Position of the last
                -- syntactical Part
End;
-----

```

**Tablica 9. Algoritam procedure KB\_Dictionary()**

```

-----
Procedure  KB_Dictionary  (
                PrgId IN Number  -- Paragraph of Sentences / Questions
            )
Is
Begin

```

```

-----
-- Insert distinct VERBS to KB_DICT from NOK_FNOK and NOK_QFNOK
-----
-->>> Process 2.1
--
Insert Into KB_DICT ( DICT_ID      , DICT  , DICT_PAR )
--
Select          M.MaxId + ROWNUM, K.NODE, null
--
From ( Select nvl(max(DICT_ID), 0) MaxId
      From KB_DICT
      ) M
      , ( Select distinct NODE
          From
              ( Select  NODE
                From    NOK_FNOK
                Where   PRG      = PrgId          -- requested Paragraph
                  And   PARENT_ID = 0            -- First Node = VERB

              Union
                Select  NODE
                From    NOK_QFNOK
                Where   PRG      = PrgId          -- requested Paragraph
                  And   PARENT_ID = 0            -- First Node = VERB
                  And   NODE     != 'X'        )
          Where NODE Not In (Select DICT From KB_DICT)
          ) K
      Order by NODE ;
--
End;
-----

```

**Tablica 10. Algoritam procedure KB\_List\_Words()**

```

-----
Procedure  KB_List_WORDS (
                PrgId IN Number  -- Paragraph of Sentences / Questions
            )
Is
Begin
--
-->>> Process 2.2
--
-- -- Inserting distinct QUEST to KB_QUESTION from NOK_FNOK
Insert Into KB_QUESTION ( QUEST_ID      , QUEST )
--
Select          M.MaxId + ROWNUM, K.QUEST
--
From ( Select nvl(max(QUEST_ID), 0) MaxId
      From KB_QUESTION
      ) M
      , ( Select distinct QUEST
          From NOK_FNOK
          Where PRG      = PrgId          -- requested Paragraph
            And QUEST != ' '
            And QUEST Not In ( Select QUEST From KB_QUESTION )
          ) K ;
--
-- -- Inserting distinct QUEST to KB_QUESTION from NOK_QFNOK
Insert Into KB_QUESTION ( QUEST_ID      , QUEST )
--
Select          M.MaxId + ROWNUM, K.QUEST
--
From ( Select nvl(max(QUEST_ID), 0) MaxId
      From KB_QUESTION
      ) M

```



```

    , ( Select distinct QUEST
        From   NOK_QFNOK
        Where  PRG      = PrgId          -- requested Paragraph
              And  QUEST != '_'
              And  QUEST Not In ( Select QUEST From KB_QUESTION )
        ) K ;
--
-- -- Inserting distinct NODE to KB_WORD from NOK_FNOK
Insert Into KB_WORD ( WORD_ID          , WORD      , DICT_ID )
--
Select
        M.MaxId + ROWNUM, K.NODE, null
--
From ( Select nvl(max(NODE_ID), 0) MaxId
      From   KB_WORD
      ) M
    , ( Select distinct NODE
        From   NOK_FNOK
        Where  PRG = PrgId          -- requested Paragraph
              And  NODE Not In ( '_' , '&' , 'X' )
              And  NODE Not In ( Select WORD From KB_WORD )
        ) K ;
--
-- -- Inserting distinct NODE to KB_WORD from NOK_QFNOK
Insert Into KB_WORD ( WORD_ID          , WORD      , DICT_ID )
--
Select
        M.MaxId + ROWNUM, K.NODE, null
--
From ( Select nvl(max(NODE_ID), 0) MaxId
      From   KB_WORD
      ) M
    , ( Select distinct NODE
        From   NOK_QFNOK
        Where  PRG = PrgId          -- requested Paragraph
              And  NODE Not In ( '_' , '&' , 'X' )
              And  NODE Not In ( Select WORD From KB_WORD )
        ) K ;
--
-->>> Process 2.3
-- -- Updating DICT_ID from KB_DICT (only for Verbs)
Update  KB_WORD
Set     DICT_ID = Select DICT_ID From KB_DICT Where DICT = KB_WORD.WORD ;
--
End;
-----

```

**Tablica 11. Upit za vizualni prikaz hijerarhije rječnika**

```

-----
Select  DICT_ID          ID
        , lpad(' ', LEVEL * 5) || DICT  VERB
        , DICT_PAR       PARENT
From    KB_DICT
Start With          DICT_ID = DICT_PAR
Connect By NOCYCLE PRIOR DICT_ID = DICT_PAR ;
-----

```

**Tablica 12. Algoritam procedure KB\_Update\_FNOK()**

```

-----
Procedure  KB_Update_FNOK (
                PrgId IN Number  -- Paragraph of Sentences / Questions
            )
Is
Begin
--
-->>> Proces 2.5

```

```

-- -- Update NOK_FNOK From List KB_QUEST And List KB_WORD
--
-- -- Parent Node Word KB_PAR_WRD
Update   NOK_FNOK      K
Set      KB_PAR_WRD   = Select   NODE
                               From   NOK_FNOK
                               Where   PRG      = K.PRIG
                               And     ORD      = K.ORD
                               And     NODE_ID  = K.PARENT_ID

Where    PRG          = PrgId          -- requested Paragraph
And     PARENT_ID    > 0      ;

--
-- -- Parent Node Word Identifier KB_PAR_WRD_ID
Update   NOK_FNOK      K
Set      KB_PAR_WRD_ID = Select   WORD_ID
                               From   KB_WORD
                               Where   WORD    = K.KB_PAR_WRD

Where    PRG          = PrgId ;          -- requested Paragraph

--
-- Node Word Identifier KB_NOD_WRD_ID
Update   NOK_FNOK      K
Set      KB_NOD_WRD_ID = Select   WORD_ID
                               From   KB_WORD
                               Where   WORD    = K.NODE

Where    PRG          = PrgId ;          -- requested Paragraph

--
-- Node Question Identifier KB_NOD_QST_ID
Update   NOK_FNOK      K
Set      KB_NOD_QST_ID = Select   QUEST_ID
                               From   KB_QUEST
                               Where   QUEST   = K.QUEST

Where    PRG          = PrgId ;          -- requested Paragraph

--
-- Node Question Identifier KB_NOD_QST_ID for QUEST = '&'
Update   NOK_FNOK      K
Set      KB_NOD_QST_ID = -1 -- non important Question
Where    QUEST         = '&'
And     PRG            = PrgId          -- requested Paragraph
-- And   KB_NOD_QST_ID is null
;

-- Update NOK_QFNOK From List KB_QUEST And List KB_WORD
--
-- Parent Node Word KB_PAR_WRD
Update   NOK_QFNOK     K
Set      KB_PAR_WRD   = ( Select   NODE
                               From   NOK_QFNOK
                               Where   PRG      = K.PRIG
                               And     ORD      = K.ORD
                               And     NODE_ID  = K.PARENT_ID
                               )

Where    PRG          = PrgId          -- requested Paragraph
-- And   KB_PAR_WRD   is null
And     PARENT_ID    > 0;

--
-- Parent Node Word Identifier KB_PAR_WRD_ID
Update   NOK_QFNOK     K
Set      KB_PAR_WRD_ID =( SELECT WORD_ID FROM KB_WORD WHERE WORD = K.KB_PAR_WRD)
Where    PRG          = PrgId          -- requested Paragraph
-- And   KB_PAR_WRD_ID is null
;

--
-- Node Word Identifier KB_NOD_WRD_ID
Update   NOK_QFNOK     K
Set      KB_NOD_WRD_ID = (SELECT WORD_ID FROM KB_WORD WHERE WORD = K.NODE )

```

```

Where PRG = PrgId -- requested Paragraph
-- And KB_NOD_WRD_ID is null
;
--
-- Node Question Identifier KB_NOD_QST_ID
Update NOK_QFNOK K
Set KB_NOD_QST_ID =( SELECT QUEST_ID FROM KB_QUESTION WHERE QUEST = K.QUEST)
Where PRG = PrgId -- requested Paragraph
-- And KB_NOD_QST_ID is null
;
--
-- Node Question Identifier KB_NOD_QST_ID for QUEST = '&'
Update NOK_QFNOK K
Set KB_NOD_QST_ID = -1 -- non important Question
Where QUEST = '&'
And PRG = PrgId -- requested Paragraph
-- And KB_NOD_QST_ID is null
;
--
End;
-----

```

**Tablica 13. Algoritam procedure KB\_Create\_QUESTION()**

```

-----
Procedure KB_Create_QUESTION (
                PrgId IN Number -- Paragraph of Sentences
                )
Is
Begin
--
-->>> Proces 2.6
-- -- Insert KB_TEXT_QUESTION From NOK_FNOK
Delete
From KB_TEXT_QUESTION
Where PRG = PrgId ; -- requested Paragraph
--
Insert Into KB_TEXT_QUESTION ( ID , PRG , ORD
                , PAR_WRD_ID , NOD_WRD_ID , NOD_QST_ID
                , PAR_WRD , NOD_WRD , NOD_QST )
Select
                M.MaxId + ROWNUM, K.PRQ , K.ORD
                , K.KB_PAR_WRD_ID , K.KB_NOD_WRD_ID, K.KB_NOD_QST_ID
                , K.KB_PAR_WRD , K.NODE , K.QUEST
From ( Select nvl(max(ID), 0) MaxId
        From KB_TEXT_QUESTION
        ) M
, ( Select *
        From NOK_FNOK
        Where PRG = PrgId -- requested Paragraph
        And PARENT_ID > 0
        Order by ORD
                , NODE_ID
        ) K ;
End;
-----

```

**Tablica 14. Algoritam procedure KB\_Create\_Answer()**

```

-----
Procedure KB_Create_Answer (
                PrgIdQ IN NUMBER DEFAULT 0 -- Paragraph of Questions (0 =
                all paragraphs)
                , PrgIds IN NUMBER DEFAULT 0 -- Paragraph of Sentences (0 =
                all paragraphs)
                )
As

```

```

--
PrgQ      Number;      -- Actual Paragraph of Questions
OrdQ      Number;      -- Actual Question
--
NumQ      Number;      -- Number      of Questions from NOK_QFNOK
MinQ      Number;      -- Row Number of First Node of Question
MaxQ      Number;      -- Row Number of Last Node of Question
--
ParQ      Number;      -- Identifier of Question's Parent Node
NodQ      Number;      -- Identifier of Question's Node
QstQ      Number;      -- Identifier of Question's Question
--
ParDictQ  Number;      -- Identifier of Question's Parent Node generic Verb
NodDictQ  Number;      -- Identifier of Question's Node generic Verb
--
PrgS      Number;      -- Actual Paragraph of Sentences
OrdS      Number;      -- Actual Sentence
--
NumS      Number;      -- Number      of Sentences from NOK_FNOK
MinS      Number;      -- Row Number of First Node of Sentence
MaxS      Number;      -- Row Number of Last Node of Sentence
--
ParS      Number;      -- Identifier of Sentence's Parent Node
NodS      Number;      -- Identifier of Sentence's Node
QstS      Number;      -- Identifier of Sentence's Question
--
ParDictS  Number;      -- Identifier of Sentence's Parent Node generic Verb
NodDictS  Number;      -- Identifier of Sentence's Node generic Verb
--
ParR      Number;      -- referent Parent
NodR      Number;      -- referent Node
--
NumNQ     Number;      -- Number of Nodes in the Question
NumAQ     Number;      -- Number of founded Answers for the Question
NumSQ     Number;      -- Number of Sentences which give the Answer
--
Ok        Char(1);     -- Flag of the Node
AnsId     Number;      -- Answer Identifier - Progressive
AnsH      Varchar2(99); -- Hierarchical Answer
Answer    Varchar2(99); -- Answer
--
Begin
--
-- -- cancel precedent Answers
Delete From KB_ANSWER;          -- empty table ANSWER
--
AnsId := 0;                    -- Answer Identifier - Progressive
--
-->>> PROCESS 3.1.1.
--
-- -- prepare data in temporary tables
--
-- XTT_QST
--
Delete From XTT_QST;          -- temporary table for Nodes of the Questions
--
Insert                                -- get all Nodes of the Questions
Into      XTT_QST
( ID, PRG, ORD, PAR          , NOD          , QST          , PAR_DICT, NOD_DICT, OK , FLW,
RID )
Select    ID, PRG, ORD, PAR_WRD_ID, NOD_WRD_ID, NOD_QST_ID, PAR_DICT, NOD_DICT, 'N', FLW,
rownum
From
(
Select    Q.ID, Q.PRG, Q.ORD, Q.PAR_WRD_ID, Q.NOD_WRD_ID, Q.NOD_QST_ID, J.DICT_PAR

```

```

PAR_DICT, K.DICT_PAR NOD_DICT, F.FLW
  From    KB_QUEST_QUEST Q
         , KB_WORD N    , KB_DICT J
         , KB_WORD M    , KB_DICT K
         , (Select PRG, ORD, rownum FLW From (Select distinct PRG, ORD From KB_QUEST_QUEST
Order by PRG, ORD)) F
  Where   (PrgIdQ      = 0              Or   Q.PRGM      = PrgIdQ )
         And   N.WORD_ID (+) = Q.PAR_WRD_ID  And   J.DICT_ID (+) = N.DICT_ID
         And   M.WORD_ID (+) = Q.NOD_WRD_ID  And   K.DICT_ID (+) = M.DICT_ID
         And   F.PRGM      = Q.PRGM        And   F.ORD      = Q.ORD
  Order by Q.PRGM, Q.ORD, Q.ID
) X;

--
-- XTT_TXT
--
Delete From XTT_TXT;          -- temporary table for Nodes of the Sentences
--
Insert          -- get all Nodes of the Sentences
Into    XTT_TXT
  ( ID, PRGM, ORD, PAR, NOD, QST, PAR_DICT, NOD_DICT, OK , FLW, RID )
Select  ID, PRGM, ORD, PAR_WRD_ID, NOD_WRD_ID, NOD_QST_ID, PAR_DICT, NOD_DICT, 'N', FLW,
rownum
  From
  (
  Select  T.ID, T.PRGM, T.ORD, T.PAR_WRD_ID, T.NOD_WRD_ID, T.NOD_QST_ID, J.DICT_PAR
PAR_DICT, K.DICT_PAR NOD_DICT, F.FLW
  From    KB_TEXT_QUEST T
         , KB_WORD N    , KB_DICT J
         , KB_WORD M    , KB_DICT K
         , (Select PRG, ORD, rownum FLW From (Select distinct PRG, ORD From KB_TEXT_QUEST
Order by PRG, ORD)) F
  Where   (PrgIdS      = 0              Or   T.PRGM      = PrgIdS )
         And   N.WORD_ID (+) = T.PAR_WRD_ID  And   J.DICT_ID (+) = N.DICT_ID
         And   M.WORD_ID (+) = T.NOD_WRD_ID  And   K.DICT_ID (+) = M.DICT_ID
         And   F.PRGM      = T.PRGM        And   F.ORD      = T.ORD
  Order by T.PRGM, T.ORD, T.ID
) X;

--
-- -- get Number of Questions NumQ and Number of Sentences NumS
--
Select Max(FLW) Into NumQ From XTT_QST;
Select Max(FLW) Into NumS From XTT_TXT;

--
-->>> PROCESS 3.1.2.
--
-- -- Create Answers
--
For FlwQ In 1 .. NumQ Loop          -- for each Question
--
  Select  Min(RID), Max(RID)          -- get first and last Row Number of
                                     the Node of the Question
         , Min(PRGM), Min(ORD)       -- get Paragraph and Order of Question
  Into    MinQ    , MaxQ
         , PrgQ    , OrdQ
  From    XTT_QST      Where FLW = FlwQ;

--
  NumNQ := MaxQ - MinQ + 1;          -- Number of Nodes in the Question
                                     (MaxQ - MinQ + 1)
  NumSQ := 0;                        -- Number of Sentences which give the
                                     Answer (initial 0)

--
  For FlwS In 1 .. NumS Loop        -- for each Sentence
--
    Select  Min(RID), Max(RID)       -- get first and last Row Number of
                                     the Node in the Sentence

```

```

, Min(PRG), Min(ORD)          -- get Paragraph and Order of Sentence
Into   MinS   , MaxS
, PrgS   , OrdS
From   XTT_TXT Where FLW = FlwS;
--
Update XTT_QST          -- reset all flags for matched question nodes
Set    OK = 'N' Where FLW = FlwQ;
--
Update XTT_TXT          -- reset all flags for marching sentence nodes
Set    OK = 'N' Where FLW = FlwS;
--
Answer := null;        -- initial values
--
-->>> PROCESS 3.1.3.
--
-- -- -- -- first loop - look for equal nodes
-----
--
For Q In MinQ .. MaxQ Loop      -- for each Node of the Question

Select  PAR , NOD , QST , PAR_DICT, NOD_DICT, OK
Into    ParQ, NodQ, QstQ, ParDictQ, NodDictQ, Ok
From    XTT_QST
Where   RID = Q;

If Ok   = 'Y' Then Continue; End If;    -- skip matched Question Nodes

If QstQ = -1 Then                    -- Question Node with '&' can be
                                     ignored

Update XTT_QST Set OK = 'Y' Where RID = Q; -- flag on ignored question node
Continue; End If;
--
-- -- -- --
--
For S IN MinS .. MaxS Loop      -- for each Node of Sentence

Select  PAR , NOD , QST , PAR_DICT, NOD_DICT, OK
Into    ParS, NodS, QstS, ParDictS, NodDictS, Ok
From    XTT_TXT
Where   RID = S;

If Ok   = 'Y'                    -- skip matched Sentence Node
Or QstS = -1                    -- or Sentence Node can be ignored,
                               skipped - non important Question
Then Continue; End If;

If ParQ is not null And          -- if exist Parent and Node
NodQ is not null And

(ParQ = ParS Or (ParDictQ is not null And ParDictQ = ParDictS)) And
(NodQ = NodS Or (NodDictQ is not null And NodDictQ = NodDictS)) Then
-- found (match) question and sentence node
Update XTT_QST Set OK = 'Y' Where RID = Q; -- flag on matched
question node
Update XTT_TXT Set OK = 'Y' Where RID = S; -- flag on matched
sentence node

End If;

End Loop; -- S - Node of Sentence

End Loop; -- Q - Node of Question
--
-- --
--

```

```

Select count(*) Into NumAQ From XTT_QST Where FLW = FlwQ And OK = 'Y';
-- get a Number of Answered Question Nodes

If NumAQ = NumNQ Then -- if all question nodes are
                        matched with sentence nodes
    NumSQ := NumSQ + 1; -- the sentence is OK - increment
                        number of good sentences
    Answer := 'YES'; -- prepare the Answer = 'YES'
    AnsId := KB_Write_Answer ( AnsId, Answer, PrgQ, OrdQ, PrgS, OrdS );
-- write the Answer

    Continue; -- FlwS - next Sentence
End If;
--
-->>> PROCESS 3.1.4.
-- -- -- -- second loop - look for nodes with questions (X, _, qst)
-----
--
    For Q In MinQ .. MaxQ Loop -- for each Node of the Question

        Select PAR , NOD , QST , PAR_DICT, NOD_DICT, OK
        Into ParQ, NodQ, QstQ, ParDictQ, NodDictQ, Ok
        From XTT_QST
        Where RID = Q;

        If Ok = 'Y' Then Continue; End If; -- skip matched Question Nodes
--
-- --
--
        For S IN MinS .. MaxS Loop -- for each Node of Sentence

            Select PAR , NOD , QST , PAR_DICT, NOD_DICT, OK
            Into ParS, NodS, QstS, ParDictS, NodDictS, Ok
            From XTT_TXT
            Where RID = S;

            If Ok = 'Y' -- skip matched Sentence Node
            Or QstS = -1 -- or Sentence Node can be ignored,
                        skipped - non important Question

                Then Continue; End If;

            If ParQ is not null And
                NodQ is not null Then -- if this node is here,
                Exit; -- there is no answer - look at the
                        first loop for equal nodes

            End If;

            If ParQ is not null Then -- if exist Parent, Node NodQ must be
                null

                If ParQ = ParS Or (ParDictQ is not null And ParDictQ = ParDictS) Then

                    If QstQ is null Or QstQ = QstS Then -- if Question of a Node
                        is empty or matches the sentence node question

                        Update XTT_QST Set OK = 'Y' Where RID = Q; -- flag on matched
                            question node
                        Update XTT_TXT Set OK = 'Y' Where RID = S; -- flag on matched
                            sentence node
                        Answer := KB_Get_Answer_Word(NodS); -- we get Answer from
                            Sentence's Node
                        AnsH := KB_Hierarchical_Answer(PrgS, OrdS, NodS); -- look for a
                            hierarchical answer

                        Select decode(AnsH , null, null, AnsH || ' ') || Answer

```

```

                Into      Answer
                From      DUAL;

            End If;
            End If;

            Else -- ParQ is null and NodQ must not be null

                If NodQ = NodS Or (NodDictQ is not null And NodDictQ = NodDictS) Then

                    Update XTT_QST Set OK = 'Y' Where RID = Q;
                    Update XTT_TXT Set OK = 'Y' Where RID = S;

                    Answer      := KB_Get_Answer_Word(ParS);           -- we get Answer from
                                                                    Sentence's Parent Node
                End If;
            End If;

            End Loop;    -- S - Node of Sentence

        End Loop;    -- Q - Node of Question
    -- --
    Select count(*) Into NumAQ From XTT_QST Where FLW = FlwQ And OK = 'Y';
                                                                    -- get a Number of Answered Question Nodes

    If NumAQ >= NumNQ Then                -- if exist an Answer
        NumSQ := NumSQ + 1;                -- increment number of sentences
                                                which give the answer
        AnsId := KB_Write_Answer ( AnsId, Answer, PrgQ, OrdQ, PrgS, OrdS );
                                                -- write the Answer

        Continue;
    End If;
    -- --
    End Loop;
    --
    -->>> PROCESS 3.1.5.
    --
    If NumSQ = 0 Then                    -- we do not found any Answer (any
        Answer := 'I DO NOT KNOW';        Sentence) for the Question
        AnsId := KB_Write_Answer ( AnsId, Answer, PrgQ, OrdQ, null, null );
                                                -- prepare the Answer = 'I DO NOT KNOW'
                                                -- write the Answer
    End If;

    End Loop; -- FlwQ - Question
    --
    AnsID := KB_Complete_Answers();        -- Complete all Answers with original text
                                                of Questions and Sentences
    --
    End;
    -----

```

**Tablica 15. Algoritam funkcije KB\_Get\_Answer\_Word()**

```

-----
Function KB_Get_Answer_Word (
                WrdId      IN Number      -- Word Identifier
            )
Return Varchar2 Is
    --
    -- Get a Word (text) from KB_WORD
    --
    Word      Varchar2 (99);
    --
    Begin

```



```

--
Select  WORD
Into    Word
From    KB_WORD
Where   WORD_ID = WrdId;
--
Return Word;
End;
-----

```

**Tablica 16. Algoritam funkcije KB\_Hierarchical\_Answer()**

```

-----
Function KB_Hierarchical_Answer (
                                PrgS IN Number      -- Paragraph of Sentences
                                , OrdS IN Number      -- Order Number of Sentence
                                , ParS IN Number      -- Parent Node
                                )
Return Varchar2 Is
--
-- Look for Hierarchical Answer
--
AnsH   Varchar2(200);
Answer Varchar2(200);
--
Cursor H_ANSWER
(
  PrgS IN Number      -- Paragraph of Sentences
  , OrdS IN Number      -- Order Number of Sentence
  , ParS IN Number      -- Parent Node
)
Is
Select      ID, PAR, NOD, TXT
From (
  Select    ID
            , PAR_WRD_ID  PAR
            , NOD_WRD_ID  NOD
            , NOD_WRD     TXT
  From      KB_TEXT_QUEST
  Where     (PrgS = 0      Or
            PRG = PrgS)
            And ORD = OrdS
)
Where      LEVEL = 2
Start With NOD = ParS
Connect By PRIOR NOD = PAR
Order By  ID
;
--
Begin
--
Answer := null;
--
For A In H_ANSWER (PrgS, OrdS, ParS) Loop
  AnsH := KB_Hierarchical_Answer(PrgS, OrdS, A.NOD); -- recursive call

  Select  decode(Answer, null, null, Answer || ' ')
          || decode(AnsH , null, null, AnsH  || ' ')
          || A.TXT
  Into    Answer
  From    DUAL;

End Loop;
--
Return Answer;
-----

```

```
--
End;
```

**Tablica 17. Algoritam funkcije KB\_Write\_Answer()**

```
-----
Function KB_Write_Answer (
    AnsId    IN Number    -- Answer Identifier
    , Answer  IN Varchar2  -- Answer
    , PrgQ    IN Number    -- Paragraph of Questions
    , OrdQ    IN Number    -- Order Number of the Question
    , PrgS    IN Number    -- Paragraph of Sentences
    , OrdS    IN Number    -- Order Number of the Sentence
)
Return Number Is
--
-- Write an Answer into table KB_ANSWER
--
Begin
--
    Insert
    Into    KB_ANSWER  ( ANSWER_ID, ANSWER, PRGQ, ORDQ, PRGS, ORDS )
    Values  ( AnsId + 1, Answer, PrgQ, OrdQ, PrgS, OrdS )
    ;
--
    Return AnsId + 1;
End;
```

**Tablica 18. Algoritam funkcije KB\_Complete\_Answers()**

```
-----
Function KB_Complete_Answers ( )
--
Return Number Is
--
-- Complete all Answers with original text of Questions (QTENG, QFNOK) and Sentences (TENG,
FNOK)
--
Begin
--
    Update  KB_ANSWER  A
    Set     QTENG = (Select QTENG From NOK_QFNOK_ALL Where PRG = A.PRGQ And ORD = A.ORDQ)
            , QFNOK = (Select QFNOK From NOK_QFNOK_ALL Where PRG = A.PRGQ And ORD = A.ORDQ)
            , TENG  = (Select TENG  From NOK_FNOK_ALL  Where PRG = A.PRGS And ORD = A.ORDS)
            , FNOK  = (Select FNOK  From NOK_FNOK_ALL  Where PRG = A.PRGS And ORD = A.ORDS)
            ;
--
    Return 1;
End;
```

## Privitak 2. Tablice

Tablica 19. Odabrane rečenice hrvatskog jezika i pripadni im FNOK-ovi

RBR	THJ	FNOK
1	Danijel vozi.	vozi ("tko?" Danijel)
2	Jakov vozi kamion.	vozi ("tko?" Jakov, "što?" kamion)
3	Tomislav vozi auto.	vozi ("tko?" Tomislav, "što?" auto)
4	Ivan vozi automobil.	vozi ("tko?" Ivan, "što?" automobil)
5	Moj prijatelj Vedran vozi auto.	vozi ("tko?" Vedran ("koji?" prijatelj ("čiji?" moj)), "što?" auto)
6	Katarina vozi crveni auto na autocesti.	vozi ("tko?" Katarina, "što?" auto ("koji?" crveni), "gdje?" autocesti ("gdje?" na))
7	Katarina vozi zeleni auto na utrci.	vozi ("tko?" Katarina, "što?" auto ("koji?" zeleni), "gdje?" utrci ("gdje?" na))
8	Petar vozi bratov auto.	vozi ("tko?" Petar, "što?" auto ("čiji?" bratov))
9	Josip crta drvenim bojicama.	crta ("tko?" Josip, "čime?" bojicama ("kakvim?" drvenim))
10	Auto je crn.	je ("što?" auto, "kakav?" crn)
11	To je kameni stol.	je ("što?" to, "što?" stol ("kakav?" kameni))
12	Stol je izrađen od kamena.	je ("što?" izrađen ("što?" stol, "čega?" kamena ("čega?" od)))
13	Stol je drven.	je ("što?" stol, "kakav?" drven)
14	Hrvatska je država.	je ("tko?" Hrvatska, "što?" država)
15	Petar je usamljen čovjek.	je ("tko?" Petar, "što?" čovjek ("kakav?" usamljen))
16	Ovaj sat je njemački.	je ("što?" sat ("koji?" ovaj), "kakav?" njemački)
17	Nož nije oštar.	nije ("što?" nož, "kakav?" oštar)
18	Gramatika je komplicirana.	je ("kakva?" komplicirana ("što?" gramatika))
19	Zadaci su komplicirani.	su ("kakvi?" komplicirani ("što?" zadaci))
20	Život je ponekad kompliciran.	je ("kakav?" kompliciran ("što?" život, "kada?" ponekad))
21	Učitelj piše studentovom olovkom.	piše ("tko?" učitelj, "čime?" olovkom ("čijom?" studentovom))
22	Studentova olovka je na stolu.	je ("što?" olovka ("čija?" studentova), "gdje?" stolu ("gdje?" na))
23	Trebam mnogo novca.	trebam ("što?" novca ("koliko?" mnogo))
24	Andrea, kupi Hariu bilo koju knjigu.	kupi ("tko?" Andrea, "kome?" Hariu, "što?" knjigu ("koju?" bilo koju))
25	Svaki dan je isti.	je ("što?" dan ("koji?" svaki), "kakav?" isti)
26	Na obje strane svijeta ljudi naporno rade svaki dan.	rade ("tko?" ljudi, "kako?" naporno, "kada?" dan ("koji?" svaki), "gdje?" svijeta ("što?" strane ("koje?" obje ("gdje?" na))))
27	Svi su moji prijatelji dobri dečki.	su ("tko?" prijatelji ("koji?" svi, "čiji?" moji), "što?" dečki ("kakvi?" dobri))
28	Netko je u kući.	je ("tko?" netko, "gdje?" kući ("gdje?" u))
29	Ona je nastavnica.	je ("tko?" ona, "što?" nastavnica)
30	Pada snijeg.	pada ("što?" snijeg)

31	To je Anđelin bicikl.	je ("što?" to, "što?" bicikl ("čiji?" Anđelin))
32	Ono su zvijezde.	su ("što?" zvijezde, "koje?" ono)
33	To je doktorov auto.	je ("što?" to, "što?" auto ("čiji?" doktorov))
34	Ovo je tvoja mačka.	je ("što?" ovo, "što?" mačka ("čija?" tvoja))
35	Ja češljam sebe.	češljam ("tko?" ja, "koga?" sebe)
36	On govori o sebi.	govori ("tko?" on, "kome?" sebi ("kome?" o))
37	Jack je u kući.	je ("tko?" Jack, "gdje?" kući ("gdje?" u))
38	Mi ne spavamo u hotelu Intercontinental.	ne ("što?" spavamo ("tko?" mi, "gdje?" hotelu ("gdje?" u, "kojem?" Intercontinental)))
39	Ti možeš kupiti bilo što.	možeš ("što?" kupiti ("što?" bilo što, "tko?" ti))
40	Tomislav kupuje 10 kila krumpira u trgovini.	kupuje ("tko?" Tomislav, "što?" krumpira ("koliko?" kila ("koliko?" 10)), "gdje?" trgovini ("gdje?" u))
41	Nitko nije došao.	nije ("što?" došao, "tko?" nitko)
42	Nikoga nije bilo na plaži.	nije ("što?" bilo, "koga?" nikoga, "gdje?" plaži ("gdje?" na))
43	Ništa nije ostalo.	nije ("što?" ostalo, "što?" ništa)
44	Svatko se njega boji.	se ("što?" boji, "tko?" svatko, "koga?" njega)
45	Ovi glumci su Robertovi prijatelji.	su ("tko?" glumci ("koji?" ovi), "što?" prijatelji ("čiji?" Robertovi))
46	Taj auto je crven.	je ("što?" auto ("koji?" taj), "kakav?" crven)
47	Ovaj dječak je Josipov brat.	je ("tko?" dječak ("koji?" ovaj), "što?" brat ("čiji?" Josipov))
48	Tom ima dva brata.	ima ("tko?" Tom, "što?" brata ("koliko?" dva))
49	U hotelu su sedamdeset i dva studenta.	su ("tko?" studenta ("koliko?" sedamdeset i dva), "gdje?" hotelu ("gdje?" u))
50	Vedrana je petnaesta dobitnica nagrade od 1999 godine.	je ("tko?" Vedrana, "što?" dobitnica ("koja?" petnaesta), "čega?" nagrade, "kada?" godine ("kada?" od, "kada?" 1999))
51	Monika je otišla u Italiju dva puta ove godine.	je ("što?" otišla ("tko?" Monika, "gdje?" Italiju ("gdje?" u), "koliko?" puta ("koliko?" dva), "kada?" godine ("koje?" ove)))
52	Automobil se kreće sporo.	se ("što?" kreće ("što?" automobil, "kako?" sporo))
53	Medvjedi pohlepno jedu.	jedu ("tko?" medvjedi, "kako?" pohlepno)
54	Marko danas vrlo brzo vozi auto.	vozi ("tko?" Marko, "što?" auto, "kada?" danas, "kako?" brzo ("koliko?" vrlo))
55	Ti izgledaš jako dobro.	izgledaš ("tko?" ti, "kako?" dobro ("kako?" jako))
56	Marija svira violinu izuzetno dobro.	svira ("tko?" Marija, "što?" violinu, "kako?" dobro ("kako?" izuzetno))
57	Knjiga je na stolu.	je ("što?" knjiga, "gdje?" stolu ("gdje?" na))
58	Studenti razgovaraju o rješenju.	razgovaraju ("tko?" studenti, "čemu?" rješenju ("čemu?" o))
59	Ja vidim učitelja.	vidim ("tko?" ja, "koga?" učitelja)
60	Vlak za Zagreb je brz.	je ("što?" vlak ("kamo?" Zagreb ("kamo?" za)), "kakav?" brz)
61	Dječak je vrlo mlad.	je ("tko?" dječak, "kakav?" mlad ("koliko?" vrlo))
62	Petar ima ured u Zagrebu.	ima ("tko?" Petar, "što?" ured, "gdje?" Zagrebu ("gdje?" u))
63	Robert ima ured 200 metara od mora u	ima ("tko?" Robert, "što?" ured, "gdje?" Dubrovniku ("gdje?" u), "gdje?" mora ("koliko?" metara ("koliko?" 200), "gdje?"

	Dubrovniku.	od))
64	Tom je prilično dobar dječak.	je ("tko?" Tom, "što?" dječak ("kakav?" dobar ("koliko?" prilično)))
65	Ja naporno radim.	radim ("tko?" ja, "kako?" naporno)
66	Noa je dao ovu knjigu prijatelju.	je ("što?" dao ("tko?" Noa, "koga?" knjigu ("koju?" ovu), "kome?" prijatelju))
67	Maja ima nekoliko zanimljivih knjiga.	ima ("tko?" Maja, "što?" knjiga ("koliko?" nekoliko, "kakvih?" zanimljivih))
68	Ja često idem u kino.	idem ("tko?" ja, "kada?" često, "kamo?" kino ("kamo?" u))
69	London je glavni grad Velike Britanije.	je ("što?" London, "što?" grad ("kakav?" glavni)("čega?" Velike Britanije))
70	Marija je Lucijina prijateljica.	je ("tko?" Marija, "što?" prijateljica ("čija?" Lucijina))
71	Otac je dobar čovjek.	je ("tko?" otac, "što?" čovjek ("kakav?" dobar))
72	Ljepota nije zauvijek.	nije ("što?" ljepota, "što?" zauvijek)
73	Poštenje je najbolja vrлина.	je ("što?" poštenje, "što?" vrлина ("kakva?" najbolja))
74	Marijin otac je star.	je ("tko?" otac ("čiji?" Marijin), "kakav?" star)
75	Šest je sati.	je ("što?" sati ("koliko?" šest))
76	Lucija ide tri puta tjedno u teretanu.	ide ("tko?" Lucija, "koliko često?" tjedno ("koliko?" puta ("koliko?" tri)), "kamo?" teretanu ("kamo?" u))
77	Škola je 50 metara od naše kuće.	je ("što?" škola ("gdje?" kuće ("koliko daleko?" metara ("koliko daleko?" 50), "gdje?" od, "čije?" naše)))
78	Ivan me vidio jučer na ulici.	vidio ("tko?" Ivan, "koga?" me, "kada?" jučer, "gdje?" ulici ("gdje?" na))
79	Tom je mlađi od Petra.	je ("tko?" Tom, "kakav?" mlađi ("koga?" Petra ("koga?" od)))
80	Josip je najmlađi.	je ("tko?" Josip, "kakav?" najmlađi)
81	Natjecanja su u zadnjem tjednu travnja.	su ("što?" natjecanja, "kada?" travnja ("kada?" tjednu ("kada?" u, "kojem?" zadnjem)))
82	Ja radim svoju domaću zadaću svaki dan u pet sati.	radim ("tko?" ja, "što?" zadaću ("čiju?" svoju, "kakvu?" domaću), "kada?" dan ("koji?" svaki), "koliko?" sati ("koliko?" u, "koliko?" pet))
83	Luka radi svoju domaću zadaću svaki dan u sedam sati.	radi ("tko?" Luka, "što?" zadaću ("čiju?" svoju, "kakvu?" domaću), "kada?" dan ("koji?" svaki), "koliko?" sati ("koliko?" u, "koliko?" sedam))
84	On može čitati.	može ("što?" čitati ("tko?" on))
85	Ti možeš dotaknuti sestrinu bilježnicu.	možeš ("što?" dotaknuti ("tko?" ti, "što?" bilježnicu ("čiju?" sestrinu)))
86	Mi možemo biti prijatelji.	možemo ("što?" biti ("tko?" mi, "što?" prijatelji))
87	Ja ću raditi sutra	ću ("što?" raditi ("tko?" ja, "kada?" sutra))
88	Luka jede sendvič.	jede ("tko?" Luka, "što?" sendvič)
89	Studenti jedu čokoladu.	jedu ("tko?" studenti, "što?" čokoladu)
90	Naš brod je prešao Atlantik u 7 dana.	je ("što?" prešao ("što?" brod ("čiji?" naš), "što?" Atlantik, "koje vrijeme?" dana ("koje vrijeme?" u, "koliko?" 7)))
91	Večera je Lukin omiljeni obrok.	je ("što?" večera, "što?" obrok ("kakav?" omiljeni, "čiji?" Lukin))
92	Automobil njegovog oca je ukraden.	je ("što?" ukraden ("što?" automobil ("čiji?" oca ("čijeg?" njegovog)))

93	Ti želiš dobru knjigu.	želiš ("tko?" ti, "što?" knjigu ("kakvu?" dobru))
94	Pada kiša.	pada ("što?" kiša)
95	U učionici je 30 studenata.	je ("tko?" studenata ("koliko?" 30), "gdje?" učionici ("gdje?" u))
96	Studenti igraju košarku svaku subotu.	igraju ("tko?" studenti, "što?" košarku, "kada?" subotu ("koju?" svaku))
97	Čovjek ide na putovanje.	ide ("tko?" čovjek, "kamo?" putovanje ("kamo?" na))
98	Studenti individualno pišu seminarski rad tijekom cijelog semestra.	pišu ("tko?" studenti, "što?" rad ("koji?" seminarski), "kako?" individualno, "kada?" semestra ("kada?" tijekom, "kakvog?" cijelog))
99	Znanstvenici u TV emisiji govore o globalnom zatopljenju.	govore ("tko?" znanstvenici, "gdje?" emisiji ("gdje?" u, "kakvoj?" TV), "čemu?" zatopljenju ("čemu?" o, "kakvom?" globalnom))
100	Marija ne izgleda umorno.	ne ("što?" izgleda ("tko?" Marija, "kako?" umorno))
101	Ivica voli Anu.	voli ("tko?" Ivica, "koga?" Anu)
102	Ana voli Ivicu.	voli ("tko?" Ana, "koga?" Ivicu)
103	Anamarija voli Antona.	voli ("tko?" Anamarija, "koga?" Antona)

Tablica 20. Pitanja prirodnog hrvatskog jezika i pripadni QFNOK-ovi

RBR	QHJ	QFNOK
1	Tko jede?	jede ("tko?" X)
2	Tko pohlepno jede?	jede ("tko?" X, _ pohlepno)
3	Tko jede čokoladu?	jede ("tko?" X, _ čokoladu)
4	Kako medvjedi jedu?	jedu ( _ medvjedi, "kako?" X)
5	Jede li Luka sendvič?	jede ( ( & li), _ Luka, _ sendvič)
6	Tko ima ured u Zagrebu?	ima ("tko?" X, _ ured, _ Zagrebu ( _ u))
7	Tko ima ured u Dubrovniku?	ima ("tko?" X, _ ured, _ Dubrovniku ( _ u))
8	Tko ima ured?	ima ("tko?" X, _ ured)
9	Gdje ima Petar ured?	ima ( _ Petar, _ ured, "gdje?" X)
10	Što Petar ima u Zagrebu?	ima ( _ Petar, "što?" X, _ Zagrebu ( _ u))
11	Gdje ima Robert ured u Dubrovniku?	ima ( _ Robert, _ ured, _ Dubrovniku ( _ u), "gdje?" X)
12	Koliko metara od mora ima Robert ured u Dubrovniku?	ima ( _ Robert, _ ured, _ Dubrovniku ( _ u), _ mora ( _ metara ("koliko?" X), _ od))
13	Tko ima nekoliko zanimljivih knjiga?	ima ("tko?" X, _ knjiga ( _ nekoliko, _ zanimljivih))
14	Što Maja ima?	ima ( _ Maja, "što?" X)
15	Koliko zanimljivih knjiga ima Maja?	ima ( _ Maja, _ knjiga ("koliko?" X, _ zanimljivih))
16	Kakve knjige ima Maja?	ima ( _ Maja, _ knjige ("kakve?" X))
17	Tko ima dva brata?	ima ("tko?" X, _ brata ( _ dva))
18	Što Tom ima?	ima ( _ Tom, "što?" X)
19	Koliko braće ima Tom?	ima ( _ Tom, _ braće ("koliko?" X))
20	Što se sporo kreće?	se ( _ kreće ("što?" X, _ sporo))
21	Kako se automobil kreće?	se ( _ kreće ( _ automobil, "kako?" X))
22	Što je u zadnjem tjednu travnja?	je ("što?" X, _ travnja ( _ tjednu ( _ u, _ zadnjem)))
23	Kada su natjecanja?	su ( _ natjecanja, "kada?" X)
24	U kojem tjednu travnja su natjecanja?	su ( _ natjecanja, _ travnja ( _ tjednu ( _ u, "kojem?" X)))
25	Jesu li natjecanja u zadnjem tjednu travnja?	jesu ( ( & li), _ natjecanja, _ travnja ( _ tjednu ( _ u, _ zadnjem)))
26	Jesu li natjecanja u zadnjem tjednu svibnja?	jesu ( ( & li), _ natjecanja, _ svibnja ( _ tjednu ( _ u, _ zadnjem)))
27	Što je ono?	je ("što?" X, _ ono)
28	Koje su zvijezde?	su ( _ zvijezde, "koje?" X)
29	Što je komplicirano?	je ( _ komplicirano ("što?" X)
30	Što je ponekad komplicirano?	je ( _ komplicirano ("što?" X, _ ponekad))
31	Je li život ponekad kompliciran?	je ( ( & li), _ kompliciran ( _ život, _ ponekad))
32	Koliko je studenata?	je ( _ studenata ("koliko?" X))
33	Gdje je 30 studenata?	je ( _ studenata ( _ 30), "gdje?" X)
34	Tko je u hotelu?	je ("tko?" X, _ hotelu ( _ u))
35	Gdje su studenti?	su ( _ studenti, "gdje?" X)

36	Tko su dobri dečki?	su ("tko?" X, _ dečki ( _ dobri))
37	Kakvi dečki su svi moji prijatelji?	su ( _ prijatelji ( _ svi, _ moji), _ dečki ("kakvi?" X))
38	Čiji prijatelji su dobri dečki?	su ( _ prijatelji ("čiji?" X), _ dečki ( _ dobri))
39	Tko su Robertovi prijatelji?	su ("tko?" X, _ prijatelji ( _ Robertovi))
40	Kakvi su moji prijatelji?	su ( _ prijatelji ( _ moji), "kakvi?" X)
41	Čiji prijatelji su ovi glumci?	su ( _ glumci ( _ ovi), _ prijatelji ("čiji?" X))
42	Tko treba kupiti bilo koju knjigu?	kupiti ( ( & treba), "tko?" X, _ knjigu ( _ bilo koju))
43	Tko može kupiti bilo što?	može ( _ kupiti ("tko?" X, _ bilo što))
44	Tko kupuje?	kupuje ("tko?" X)
45	Što treba Andrea kupiti Hariu?	kupiti ( (& treba), _ Andrea, _ Hariu, "što?" X)
46	Što možeš kupiti?	možeš ( _ kupiti ("što?" X))
47	Kome Andrea treba kupiti bilo koju knjigu?	kupiti ( ( & treba), _ Andrea, "kome?" X, _ knjigu ( _ bilo koju))
48	Koju knjigu treba Andrea kupiti?	kupiti ( ( & treba), _ Andrea, _ knjigu ("koju?" X))
49	Što Tomislav kupuje?	kupuje ( _ Tomislav, "što?" X)
50	Koliko kila krumpira Tomislav kupuje?	kupuje ( _ Tomislav, _ krumpira ( _ kila ("koliko?" X)))
51	Gdje Tomislav kupuje krumpir?	kupuje ( _ Tomislav, _ krumpir, "gdje?" X)
52	Tko nije došao?	nije ( _ došao, "tko?" X)
53	Što on može?	može ("što?" X ( _ on))
54	Što ti možeš?	možeš ("što?" X ( _ ti))
55	Tko može dotaknuti sestrinu bilježnicu?	može ( _ dotaknuti ("tko?" X, _ bilježnicu ( _ sestrinu)))
56	Tko može kupiti bilo što?	može ( _ kupiti ( _ bilo što, "tko?" X))
57	Čiju bilježnicu možeš dotaknuti?	možeš ( _ dotaknuti ( _ bilježnicu ("čiju?" X)))
58	Što mi možemo biti?	možemo ( _ biti ( _ mi, "što?" X))
59	Što je prešlo Atlantik u 7 dana?	je ( _ prešlo ("što?" X, _ Atlantik, _ dana ( _ u, _ 7)))
60	Što je prešao naš brod?	je ( _ prešao ( _ brod ( _ naš), "što?" X))
61	U koje vrijeme je naš brod prešao Atlantik?	je ( _ prešao ( _ brod ( _ naš), _ Atlantik, "koje vrijeme?" X (& u))
62	Tko radi svoju domaću zadaću svaki dan?	radi ("tko?" X, _ zadaću ( _ svoju, _ domaću), _ dan ( _ svaki))
63	Što je radim svaki dan?	radim ( _ ja, "što?" X, _ dan ( _ svaki))
64	Kada Luka radi svoju domaću zadaću?	radi ( _ Luka, _ zadaću ( _ svoju, _ domaću), "kada?" X)
65	Tko crta?	crta ("tko?" X)
66	Čime Josip crta?	crta ( _ Josip, "čime?" X)
67	Kakvim bojicama crta Josip?	crta ( _ Josip, _ bojicama ("kakvim?" X))
68	Tko vozi?	vozi ("tko?" X)
69	Tko vozi auto?	vozi ("tko?" X, _ auto)
70	Čiji auto vozi Petar?	vozi ( _ Petar, _ auto ("čiji?" X))
71	Kako Marko danas vozi auto?	vozi ( _ Marko, _ auto, _ danas, "kako?" X)
72	Kada Marko vrlo brzo vozi	vozi ( _ Marko, _ auto, "kada?" X, _ brzo ( _ vrlo))



	auto?	
73	Koji auto vozi Katarina?	vozi ( _ Katarina, _ auto ("koji?" X))
74	Koji auto vozi Katarina na utrci?	vozi ( _ Katarina, _ auto ("koji?" X), _ utrci ( _ na))
75	Gdje Katarina vozi crveni auto?	vozi ( _ Katarina, _ auto ( _ crveni), "gdje?" X)
76	Što Jakov vozi?	vozi ( _ Jakov, "što?" X)
77	Je li Noa dao ovu knjigu prijatelju?	je ( (& li), _ dao ( _ Noa, _ knjigu ( _ ovu), _ prijatelju))
78	Kome je Noa dao ovu knjigu?	je ( _ dao ( _ Noa, _ knjigu ( _ ovu), "kome?" X))
79	Gdje mi ne spavamo?	ne ( _ spavamo ( _ mi, "gdje?" X))
80	Tko svira?	svira ("tko?" X)
81	Kamo često idem?	idem ( _ često, "kamo?" X)
82	Kamo čovjek ide?	ide ( _ čovjek, "kamo?" X)
83	Koliko puta tjedno ide Lucija u teretanu?	ide ( _ Lucija, _ tjedno ( _ puta ("koliko?" X)), _ teretanu ( _ u))
84	Koliko često ide Lucija u teretanu?	ide ( _ Lucija, "koliko često?" X, _ teretanu ( _ u))
85	Kamo ide Lucija?	ide ( _ Lucija, "kamo?" X)
86	Kako Marija ne izgleda?	ne ( _ izgleda ( _ Marija, "kako?" X))
87	Koga ja vidim?	vidim ("koga?" X, _ ja)
88	Kako izgledaš?	izgledaš ("kako?" X)
89	Koliko novaca trebam?	trebam ( _ novaca ("koliko?" X))
90	Što ja trebam?	trebam ( _ ja, "što?" X)
91	Tko igra?	igra ("tko?" X)
92	Kako Marija svira violinu?	svira ( _ Marija, _ violinu, "kako?" X)
93	Što Marija svira izuzetno dobro?	svira ( _ Marija, "što?" X, _ dobro ( _ izuzetno))
94	Kada studenti igraju košarku?	igraju ( _ studenti, _ košarku, "kada?" X)
95	Što studenti igraju?	igraju ( _ studenti, "što?" X)
96	Pada li kiša?	pada ( (& li), _ kiša)
97	Pada li snijeg?	pada ( (& li), _ snijeg)
98	Gdje me Ivan vidio jučer?	vidio ( _ Ivan, _ me, _ jučer, "gdje?" X)
99	Kada me Ivan vidio?	vidio ( _ Ivan, _ me, "kada?" X)
100	Koga ja vidim?	vidim ( _ ja, "koga?" X)
101	Tko vidi učitelja?	vidim ("tko?" X, _ učitelja)
102	Tko govori o sebi?	govori ("tko?" X, _ sebi ( _ o))
103	Tko razgovara?	razgovara ("tko?" X)
104	O kome on govori?	govori ( _ on, "kome?" X (& o))
105	O čemu znanstvenici govore?	govore ( _ znanstvenici, "čemu?" X (& o))
106	Gdje znanstvenici govore?	govore ( _ znanstvenici, "gdje?" X)
107	O čemu studenti razgovaraju?	razgovaraju ( _ studenti, "čemu?" X (& o))
108	Što želiš?	želiš ("što?" X)
109	Kakvu knjigu želiš?	želiš ( _ knjigu ("kakvu?" X))
110	Što je ukradeno?	je ( _ ukradeno ("što?" X))
111	Je li bilo koga na plaži?	je ( (& li), _ bilo, "koga?" X, _ plaži ( _ na))

112	Tko je Vedrana?	je ( _ Vedrana, "tko?" X)
113	Koga ja češljam?	češljam ( _ ja, "koga?" X)
114	Gdje je Monika otišla?	je ( _ otišla ( _ Monika, "gdje?" X))
115	Koliko je puta ove godine Monika otišla u Italiju?	je ( _ otišla ( _ Monika, _ Italiju ( _ u), _ puta ("koliko?" X, _ godine ( _ ove)))
116	Kada je Monika otišla u Italiju?	je ( _ otišla ( _ Monika, _ Italiju ( _ u), "kada?" X))
117	Tko ide?	ide ("tko?" X)
118	Kakav je stol?	je ( _ stol, "kakav?" X)
119	Tko je otišao?	je ( _ otišao ("tko?" X))
120	Tko će raditi?	će ( _ raditi ("tko?" X))
121	Tko naporno radi?	radi ("tko?" X, _ naporno)
122	Gdje ljudi naporno rade?	rade ( _ ljudi, _ naporno, "gdje?" X)
123	Kada ću ja raditi?	ću ( _ raditi ( _ ja, "kada?" X))
124	Kada ljudi rade?	rade ( _ ljudi, "kada?" X)
125	Kada studenti pišu?	pišu ( _ studenti, "kada?" X)
126	Čime učitelj piše?	piše ( _ učitelj, "čime?" X)
127	Kako studenti pišu seminarski rad?	pišu ( _ studenti, _ rad ( _ seminarski), "kako?" X)
128	Tko piše?	piše ("tko?" X)
129	Što je crno?	je ("što?" X, _ crno)
130	Što je isto?	je ("što?" X, _ isto)
131	Gdje je olovka?	je ( _ olovka, "gdje?" X)
132	Čija olovka je na stolu?	je ( _ olovka ("čija?" X), _ stolu ( _ na))
133	Što je na stolu?	je ("što?" X, _ stolu ( _ na))
134	Kakav je to stol?	je ( _ to, _ stol ("kakav?" X))
135	Što je drveno?	je ("što?" X, _ drveno)
136	Od čega je stol izrađen?	je ( _ izrađen ( _ stol, "čega?" X ( & od)))
137	Što je izrađeno od kamena?	je ( _ izrađeno ("što?" X, _ kamena ( _ od)))
138	Tko se njega boji?	se ( _ boji, "tko?" X, _ njega)
139	Što je večera?	je ( _ večera, "što?" X)
140	Čiji omiljeni obrok je večera?	je ( _ večera, _ obrok ( _ omiljeni, "čiji?" X))
141	Što Luka radi?	X ( (& radi), _ Luka)
142	Što je poštenje?	je ( _ poštenje, "što?" X)
143	Što je najbolja vrlina?	je ("što?" X, _ vrlina ( _ najbolja))
144	Koliko je sati?	je ( _ sati ("koliko?" X))
145	Što je London?	je ( _ London, "što?" X)
146	Gdje je škola?	je ( _ škola ("gdje?" X))
147	Koliko daleko je škola od naše kuće?	je ( _ škola ( _ kuće ("koliko daleko?" X, _ od, _ naše))
148	Što je brzo?	je ("što?" X, _ brzo)
149	Što je ostalo?	je ( _ ostalo, "što?" X)
150	Koliko mlad je dječak?	je ( _ dječak, _ mlad ("koliko?" X))
151	Što je Tom?	je ( _ Tom, "što?" X)
152	Tko je Josipov brat?	je ("tko?" X, _ brat ( _ Josipov))
153	Čiji otac je star?	je ( _ otac ("čiji?" X), _ star)

154	Kakav je otac?	je ( _ otac, "kakav?" X)
155	Tko je najmlađi?	je ("tko?" X, _ najmlađi)
156	Tko je mlad?	je ("tko?" X, _ mlad)
157	Tko je mlađi od Petra?	je ("tko?" X, _ mlađi ( _ Petra ( _ od)))
158	Čija prijateljica je Marija?	je ( _ Marija, _ prijateljica ("čija?" X))
159	Tko je u kući?	je ("tko?" X, _ kući ( _ u))
160	Što je Hrvatska?	je ( _ Hrvatska, "što?" X)
161	Kakav je ovaj sat?	je ( _ sat ( _ ovaj), "kakav?" X)
162	Tko je dobar čovjek?	je ("tko?" X, _ čovjek ( _ dobar))
163	Kakav čovjek je Petar?	je ( _ Petar, _ čovjek ("kakav?" X))
164	Tko je usamljen čovjek?	je ("tko?" X, _ čovjek ( _ usamljen))
165	Što Petar radi?	X ( (& radi), _ Petar)
166	Čiji je to bicikl?	je ( _ to, _ bicikl ("čiji?" X))
167	Čiji je to auto?	je ( _ to, _ auto ("čiji?" X))
168	Što je crveno	je ("što?" X, _ crveno)
169	Što nije zauvijek?	nije ("što?" X, _ zauvijek)
170	Što nije oštro?	nije ("što?" X, _ oštro)
171	Što nije ostalo?	nije ( _ ostalo, "što?" X)
172	Čija je ovo mačka?	je ( _ ovo, _ mačka ("čija?" X))
173	Što je ona?	je ( _ ona, "što?" X)
174	Što studenti rade?	X ( (& rade), _ studenti)
175	Koliko krumpira Tomislav kupuje?	kupuje ( _ Tomislav, _ krumpira ("koliko?" X))
176	Tko voli Anu?	voli ("tko?" X, _ Anu)
177	Tko voli Ivicu?	voli ("tko?" X, _ Ivicu)
178	Tko voli Antona?	voli ("tko?" X, _ Antona)
179	Što trebam?	trebam ("što?" X)
180	Tko treba mnogo novaca?	treba ("tko?" X, _ novaca ( _ mnogo))

Tablica 21. Odgovori sustava na pitanja (hrvatski jezik)

ANSWER_ID	PRGQ	ORDQ	QHJ	Očekivani odgovor	ANSWER sustava	Analiza odgovora	PRGS	ORDS
1	444	1	Tko jede?	Medvjedi	medvjedi	Točno	444	53
2	444	1	Tko jede?	Luka	Luka	Točno	444	88
3	444	1	Tko jede?	Studenti	studenti	Točno	444	89
4	444	2	Tko pohlepno jede?	Medvjedi	medvjedi	Točno	444	53
5	444	3	Tko jede čokoladu?	Studenti	studenti	Točno	444	89
6	444	4	Kako medvjedi jedu?	pohlepno	pohlepno	Točno	444	53
7	444	5	Jede li Luka sendvič?	YES	YES	Točno	444	88
8	444	6	Tko ima ured u Zagrebu?	Petar	Petar	Točno	444	62
9	444	7	Tko ima ured u Dubrovniku?	Robert	Robert	Točno	444	63
10	444	8	Tko ima ured?	Petar	Petar	Točno	444	62
11	444	8	Tko ima ured?	Robert	Robert	Točno	444	63
12	444	9	Gdje ima Petar ured?	u Zagrebu	u Zagrebu	Točno	444	62
13	444	10	Što Petar ima u Zagrebu?	ured	ured	Točno	444	62
14	444	11	Gdje ima Robert ured u Dubrovniku?	200 metara od mora	200 metara od mora	Točno	444	63
15	444	12	Koliko metara od mora ima Robert ured u Dubrovniku?	200	200	Točno	444	63
16	444	13	Tko ima nekoliko zanimljivih knjiga?	Maja	Maja	Točno	444	67
17	444	14	Što Maja ima?	nekoliko zanimljivih knjiga	nekoliko zanimljivih knjiga	Točno	444	67
18	444	15	Koliko zanimljivih knjiga ima Maja?	nekoliko	nekoliko	Točno	444	67
19	444	16	Kakve knjige ima Maja?	zanimljive	I DO NOT KNOW	Netočno		
20	444	17	Tko ima dva brata?	Tom	Tom	Točno	444	48
21	444	18	Što Tom ima?	dva brata	dva brata	Točno	444	48
22	444	19	Koliko braće ima Tom?	dva	dva	Točno	444	48
23	444	20	Što se sporo kreće?	automobil	automobil	Točno	444	52
24	444	21	Kako se automobil kreće?	sporo	sporo	Točno	444	52
25	444	22	Što je u zadnjem tjednu travnja?	natjecanja	natjecanja	Točno	444	81
26	444	23	Kada su natjecanja?	u zadnjem tjednu	u zadnjem tjednu	Točno	444	81

				travnja	travnja			
27	444	24	U kojem tjednu travnja su natjecanja?	zadnjem	zadnjem	Točno	444	81
28	444	25	Jesu li natjecanja u zadnjem tjednu travnja?	YES	YES	Točno	444	81
29	444	26	Jesu li natjecanja u zadnjem tjednu svibnja?	NE	I DO NOT KNOW	Netočno		
30	444	27	Što je ono?	zvijezde	zvijezde	Točno	444	32
31	444	28	Koje su zvijezde?	ono	ono	Točno	444	32
32	444	29	Što je komplicirano?	Gramatika	gramatika	Točno	444	18
33	444	29	Što je komplicirano?	Zadaci	zadaci	Točno	444	19
34	444	29	Što je komplicirano?	Život	život	Točno	444	20
35	444	30	Što je ponekad komplicirano?	Život	život	Točno	444	20
36	444	31	Je li život ponekad kompliciran?	YES	YES	Točno	444	20
37	444	32	Koliko je studenata?	sedamdeset i dva	sedamdeset i dva	Točno	444	49
38	444	32	Koliko je studenata?	30	30	Točno	444	95
39	444	33	Gdje je 30 studenata?	u učionici	u učionici	Točno	444	95
40	444	34	Tko je u hotelu?	sedamdeset i dva studenta	sedamdeset i dva studenta	Točno	444	49
41	444	35	Gdje su studenti?	U hotelu	u hotelu	Točno	444	49
42	444	35	Gdje su studenti?	U učionici	u učionici	Točno	444	95
43	444	36	Tko su dobri dečki?	Svi moji prijatelji	svi moji prijatelji	Točno	444	27
44	444	37	Kakvi dečki su svi moji prijatelji?	dobri	dobri	Točno	444	27
45	444	38	Čiji prijatelji su dobri dečki?	moji	moji	Točno	444	27
46	444	39	Tko su Robertovi prijatelji?	Ovi glumci	ovi glumci	Točno	444	45
47	444	40	Kakvi su moji prijatelji?	dobri dečki	I DO NOT KNOW	Netočno		
48	444	41	Čiji prijatelji su ovi glumci?	Robertovi	Robertovi	Točno	444	45
49	444	42	Tko treba kupiti bilo koju knjigu?	Andrea	Andrea	Točno	444	24
50	444	43	Tko može kupiti bilo što?	Ti	ti	Točno	444	39
51	444	44	Tko kupuje?	Andrea	Andrea	Točno	444	24
52	444	44	Tko kupuje?	Ti	ti	Točno	444	39
53	444	44	Tko kupuje?	Tomislav	Tomislav	Točno	444	40
54	444	45	Što treba Andrea kupiti Hariu?	bilo koju knjigu	bilo koju knjigu	Točno	444	24
55	444	46	Što možeš kupiti?	bilo što	bilo što	Točno	444	39

56	444	47	Kome Andrea treba kupiti bilo koju knjigu?	Hariu	Hariu	Točno	444	24
57	444	48	Koju knjigu treba Andrea kupiti?	bilo koju	bilo koju	Točno	444	24
58	444	49	Što Tomislav kupuje?	10 kila krumpira	10 kila krumpira	Točno	444	40
59	444	50	Koliko kila krumpira Tomislav kupuje?	10	10	Točno	444	40
60	444	51	Gdje Tomislav kupuje krumpir?	u trgovini	u trgovini	Točno	444	40
61	444	52	Tko nije došao?	Nitko	nitko	Točno	444	41
62	444	53	Što on može?	čitati	čitati	Točno	444	84
63	444	54	Što ti možeš?	kupiti bilo što	kupiti	Točno	444	39
64	444	54	Što ti možeš?	dotaknuti sestrinu bilježnicu	dotaknuti	Točno	444	85
65	444	55	Tko može dotaknuti sestrinu bilježnicu?	Ti	ti	Točno	444	85
66	444	56	Tko može kupiti bilo što?	Ti	ti	Točno	444	39
67	444	57	Čiju bilježnicu možeš dotaknuti?	sestrinu	sestrinu	Točno	444	85
68	444	58	Što mi možemo biti?	prijatelji	prijatelji	Točno	444	86
69	444	59	Što je prešlo Atlantik u 7 dana?	Naš brod	naš brod	Točno	444	90
70	444	60	Što je prešao naš brod?	Atlantik	Atlantik	Točno	444	90
71	444	61	U koje vrijeme je naš brod prešao Atlantik?	u 7 dana	u 7 dana	Točno	444	90
72	444	62	Tko radi svoju domaću zadaću svaki dan?	Ja	ja	Točno	444	82
73	444	62	Tko radi svoju domaću zadaću svaki dan?	Luka	Luka	Točno	444	83
74	444	63	Što je radim svaki dan?	svoju domaću zadaću	svoju domaću zadaću	Točno	444	82
75	444	64	Kada Luka radi svoju domaću zadaću?	svaki dan u sedam sati	svaki dan	Semantički nepotpun odgovor	444	83
76	444	65	Tko crta?	Josip	Josip	Točno	444	9
77	444	66	Čime Josip crta?	drvenim bojicama	drvenim bojicama	Točno	444	9
78	444	67	Kakvim bojicama crta Josip?	drvenim	drvenim	Točno	444	9
79	444	68	Tko vozi?	Danijel	Danijel	Točno	444	1
80	444	68	Tko vozi?	Jakov	Jakov	Točno	444	2
81	444	68	Tko vozi?	Tomislav	Tomislav	Točno	444	3

82	444	68	Tko vozi?	Ivan	Ivan	Točno	444	4
83	444	68	Tko vozi?	Moj prijatelj Vedran	moj prijatelj Vedran	Točno	444	5
84	444	68	Tko vozi?	Katarina	Katarina	Točno	444	6
85	444	68	Tko vozi?	Katarina	Katarina	Točno	444	7
86	444	68	Tko vozi?	Petar	Petar	Točno	444	8
87	444	68	Tko vozi?	Marko	Marko	Točno	444	54
88	444	69	Tko vozi auto?	Tomislav	Tomislav	Točno	444	3
89	444	69	Tko vozi auto?	Ivan	Ivan	Točno	444	4
90	444	69	Tko vozi auto?	Moj prijatelj Vedran	moj prijatelj Vedran	Točno	444	5
91	444	69	Tko vozi auto?	Katarina	Katarina	Točno	444	6
92	444	69	Tko vozi auto?	Katarina	Katarina	Točno	444	7
93	444	69	Tko vozi auto?	Petar	Petar	Točno	444	8
94	444	69	Tko vozi auto?	Marko	Marko	Točno	444	54
95	444	70	Čiji auto vozi Petar?	bratov	bratov	Točno	444	8
96	444	71	Kako Marko danas vozi auto?	vrlo brzo	vrlo brzo	Točno	444	54
97	444	72	Kada Marko vrlo brzo vozi auto?	danas	danas	Točno	444	54
98	444	73	Koji auto vozi Katarina?	crveni	crveni	Točno	444	6
99	444	73	Koji auto vozi Katarina?	zeleni	zeleni	Točno	444	7
100	444	74	Koji auto vozi Katarina na utrci?	zeleni	zeleni	Točno	444	7
101	444	75	Gdje Katarina vozi crveni auto?	na autocesti	na autocesti	Točno	444	6
102	444	76	Što Jakov vozi?	kamion	kamion	Točno	444	2
103	444	77	Je li Noa dao ovu knjigu prijatelju?	YES	YES	Točno	444	66
104	444	78	Kome je Noa dao ovu knjigu?	Prijatelju	prijatelju	Točno	444	66
105	444	79	Gdje mi ne spavamo?	u hotelu Intercontinental	u Intercontinental hotelu	Točno	444	38
106	444	80	Tko svira?	Marija	Marija	Točno	444	56
107	444	81	Kamo često idem?	u kino	u kino	Točno	444	68
108	444	82	Kamo čovjek ide?	na putovanje	na putovanje	Točno	444	97
109	444	83	Koliko puta tjedno ide Lucija u teretanu?	tri puta	tri	Točno	444	76
110	444	84	Koliko često ide Lucija u teretanu?	tri puta tjedno	tri puta tjedno	Točno	444	76
111	444	85	Kamo ide Lucija?	u teretanu	u teretanu	Točno	444	76

112	444	86	Kako Marija ne izgleda?	umorno	umorno	Točno	444	100
113	444	87	Koga ja vidim?	Učitelja	učitelja	Točno	444	59
114	444	88	Kako izgledaš?	jako dobro	jako dobro	Točno	444	55
115	444	89	Koliko novaca trebam?	mного	mного	Točno	444	23
116	444	90	Što ja trebam?	mного novca	I DO NOT KNOW	Netočno		
117	444	91	Tko igra?	Studenti	studenti	Točno	444	96
118	444	92	Kako Marija svira violinu?	izuzetno dobro	izuzetno dobro	Točno	444	56
119	444	93	Što Marija svira izuzetno dobro?	violinu	violinu	Točno	444	56
120	444	94	Kada studenti igraju košarku?	svaku subotu	svaku subotu	Točno	444	96
121	444	95	Što studenti igraju?	košarku	košarku	Točno	444	96
122	444	96	Pada li kiša?	YES	YES	Točno	444	94
123	444	97	Pada li snijeg?	YES	YES	Točno	444	30
124	444	98	Gdje me Ivan vidio jučer?	na ulici	na ulici	Točno	444	78
125	444	99	Kada me Ivan vidio?	jučer	jučer	Točno	444	78
126	444	100	Koga ja vidim?	Učitelja	učitelja	Točno	444	59
127	444	101	Tko vidi učitelja?	Ja	ja	Točno	444	59
128	444	102	Tko govori o sebi?	On	on	Točno	444	36
129	444	103	Tko razgovara?	Studenti	studenti	Točno	444	58
130	444	104	O kome on govori?	o sebi	o sebi	Točno	444	36
131	444	105	O čemu znanstvenici govore?	o globalnom zatopljenju	o globalnom zatopljenju	Točno	444	99
132	444	106	Gdje znanstvenici govore?	u TV emisiji	u TV emisiji	Točno	444	99
133	444	107	O čemu studenti razgovaraju?	o rješenju	o rješenju	Točno	444	58
134	444	108	Što želiš?	dobru knjigu	dobru knjigu	Točno	444	93
135	444	109	Kakvu knjigu želiš?	dobru	dobru	Točno	444	93
136	444	110	Što je ukradeno?	Automobil njegovog oca	njegovog oca automobil	Semantički ispravan, ali gramatički neispravan odgovor	444	92
137	444	111	Je li bilo koga na plaži?	Nikoga	nikoga	Točno	444	42
138	444	112	Tko je Vedrana?	petnaesta dobitnica	petnaesta	Semantički	444	50



				nagrade od 1999 godine	dobitnica	nepotpun odgovor		
139	444	113	Koga ja češljam?	Sebe	sebe	Točno	444	35
140	444	114	Gdje je Monika otišla?	u Italiju	u Italiju	Točno	444	51
141	444	115	Koliko je puta ove godine Monika otišla u Italiju?	dva	dva	Točno	444	51
142	444	116	Kada je Monika otišla u Italiju?	ove godine	ove godine	Točno	444	51
143	444	117	Tko ide?	Ja	ja	Točno	444	68
144	444	117	Tko ide?	Lucija	Lucija	Točno	444	76
145	444	117	Tko ide?	Čovjek	čovjek	Točno	444	97
146	444	118	Kakav je stol?	kameni		Nedostaje odgovor		
146a	444	118	Kakav je stol?	drven	drven	Točno	444	13
147	444	119	Tko je otišao?	Monika	Monika	Točno	444	51
148	444	120	Tko će raditi?	Ja	ja	Točno	444	87
149	444	121	Tko naporno radi?	Ljudi	ljudi	Točno	444	26
150	444	121	Tko naporno radi?	Ja	ja	Točno	444	65
151	444	122	Gdje ljudi naporno rade?	Na obje strane svijeta	na obje strane svijeta	Točno	444	26
152	444	123	Kada ću ja raditi?	Sutra	sutra	Točno	444	87
153	444	124	Kada ljudi rade?	svaki dan	svaki dan	Točno	444	26
154	444	125	Kada studenti pišu?	tijekom cijelog semestra	tijekom cijelog semestra	Točno	444	98
155	444	126	Čime učitelj piše?	studentovom olovkom	studentovom olovkom	Točno	444	21
156	444	127	Kako studenti pišu seminarski rad?	individualno	individualno	Točno	444	98
157	444	128	Tko piše?	Učitelj	učitelj	Točno	444	21
158	444	128	Tko piše?	Studenti	studenti	Točno	444	98
159	444	129	Što je crno?	Auto	auto	Točno	444	10
160	444	130	Što je isto?	Svaki dan	svaki dan	Točno	444	25
161	444	131	Gdje je olovka?	na stolu	na stolu	Točno	444	22
162	444	132	Čija olovka je na stolu?	Studentova	studentova	Točno	444	22
163	444	133	Što je na stolu?	Studentova olovka	studentova olovka	Točno	444	22

164	444	133	Što je na stolu?	Knjiga	knjiga	Točno	444	57
165	444	134	Kakav je to stol?	kameni	kameni	Točno	444	11
166	444	135	Što je drveno?	Stol	stol	Točno	444	13
167	444	136	Od čega je stol izrađen?	kamena	od kamena	Točno	444	12
168	444	137	Što je izrađeno od kamena?	Stol	stol	Točno	444	12
169	444	138	Tko se njega boji?	Svatko	svatko	Točno	444	44
170	444	139	Što je večera?	Lukin omiljeni obrok	omiljeni Lukin obrok	Točno	444	91
171	444	140	Čiji omiljeni obrok je večera?	Lukin	Lukin	Točno	444	91
172	444	141	Što Luka radi?	radi svoju domaću zadaću	radi	Točno	444	83
173	444	141	Što Luka radi?	jede sendvič	jede	Točno	444	88
174	444	142	Što je poštenje?	najbolja vrlina	najbolja vrlina	Točno	444	73
175	444	143	Što je najbolja vrlina?	poštenje	poštenje	Točno	444	73
176	444	144	Koliko je sati?	šest	šest	Točno	444	75
177	444	145	Što je London?	glavni grad Velike Britanije	glavni grad	Semantički nepotpun odgovor	444	69
178	444	146	Gdje je škola?	50 metara od naše kuće	50 metara od naše kuće	Točno	444	77
179	444	147	Koliko daleko je škola od naše kuće?	50 metara	50 metara	Točno	444	77
180	444	148	Što je brzo?	Vlak za Zagreb	za Zagreb vlak	Semantički ispravan, ali gramatički neispravan odgovor	444	60
181	444	149	Što je ostalo?	ništa	ništa	Točno	444	43
182	444	150	Koliko mlad je dječak?	vrlo	vrlo	Točno	444	61
183	444	151	Što je Tom?	dobar dječak	prilično dobar dječak	Točno	444	64
184	444	152	Tko je Josipov brat?	Ovaj dječak	ovaj dječak	Točno	444	47
185	444	153	Čiji otac je star?	Marijin	Marijin	Točno	444	74
186	444	154	Kakav je otac?	dobar čovjek		Nedostaje odgovor		

186a	444	154	Kakav je otac?	star	star	Točno	444	74
187	444	155	Tko je najmlađi?	Josip	Josip	Točno	444	80
188	444	156	Tko je mlađi?	Dječak	dječak	Točno	444	61
189	444	157	Tko je mlađi od Petra?	Tom	Tom	Točno	444	79
190	444	158	Čija prijateljica je Marija?	Lucijina	Lucijina	Točno	444	70
191	444	159	Tko je u kući?	Netko	netko	Točno	444	28
192	444	159	Tko je u kući?	Jack	Jack	Točno	444	37
193	444	160	Što je Hrvatska?	država	država	Točno	444	14
194	444	161	Kakav je ovaj sat?	njemački	njemački	Točno	444	16
195	444	162	Tko je dobar čovjek?	Otac	otac	Točno	444	71
196	444	163	Kakav čovjek je Petar?	usamljen	usamljen	Točno	444	15
197	444	164	Tko je usamljen čovjek?	Petar	Petar	Točno	444	15
198	444	165	Što Petar radi?	vozi bratov auto	vozi	Točno	444	8
199	444	165	Što Petar radi?		je	Netočno	444	15
200	444	165	Što Petar radi?		ima	Netočno	444	62
201	444	166	Čiji je to bicikl?	Anđelin	Anđelin	Točno	444	31
202	444	167	Čiji je to auto?	doktorov	doktorov	Točno	444	33
203	444	168	Što je crveno	taj auto	taj auto	Točno	444	46
204	444	169	Što nije zauvijek?	Ljepota	ljepota	Točno	444	72
205	444	170	Što nije oštro?	Nož	nož	Točno	444	17
206	444	171	Što nije ostalo?	ništa	ništa	Točno	444	43
207	444	172	Čija je ovo mačka?	tvoja	tvoja	Točno	444	34
208	444	173	Što je ona?	nastavnica	nastavnica	Točno	444	29
209	444	174	Što studenti rade?		su	Netočno	444	49
210	444	174	Što studenti rade?	razgovaraju o rješenju	razgovaraju	Točno	444	58
211	444	174	Što studenti rade?	jedu čokoladu	jedu	Točno	444	89
212	444	174	Što studenti rade?		je	Netočno	444	95
213	444	174	Što studenti rade?	igraju košarku	igraju	Točno	444	96
214	444	174	Što studenti rade?	pišu seminarski rad	pišu	Točno	444	98
215	444	175	Koliko krumpira Tomislav kupuje?	10 kila	10 kila	Točno	444	40
216	444	176	Tko voli Anu?	Ivica	Ivica	Točno	444	101

217	444	177	Tko voli Ivicu?	Ana	Ana	Točno	444	102
218	444	178	Tko voli Antona?	Anamarija	Anamarija	Točno	444	103
219	444	179	Što trebam?	mного novaca	mного novaca	Točno	444	23
220	444	180	Tko treba mnogo novaca?	ja	I DO NOT KNOW	Netočno		

Tablica 22. Odabrane rečenice engleskog jezika i pripadni FNOK-ovi

RBR	TENG	FNOK
1	Daniel drives.	drives ("who?" Daniel)
2	Jacob drives a truck.	drives ("who?" Jacob, "what?" truck ("art?" a))
3	Thomas drives a car.	drives ("who?" Thomas, "what?" car ("art?" a))
4	John drives an automobile.	drives ("who?" John, "what?" automobile ("art?" an))
5	My friend Sean drives a car.	drives ("who?" Sean ("who?" friend ("whose?" my)), "what?" car ("art?" a))
6	Catherine drives a red car on the highway.	drives ("who?" Catherine, "what?" car ("art?" a, "what?" red), "where?" highway ("where?" on, "art?" the))
7	Catherine drives a green car in the race.	drives ("who?" Catherine, "what?" car ("art?" a, "what?" green), "where?" race ("where?" in, "art?" the))
8	Peter drives his brother's car.	drives ("who?" Peter, "what?" car ("whose?" brother's ("whose?" his)))
9	Joseph draws with wooden crayons.	draws ("who?" Joseph, "what?" crayons ("what?" with, "what?" wooden))
10	The car is black.	is ("what?" car ("art?" the), "what?" black)
11	It is a stone table.	is ("what?" it, "what?" table ("art?" a, "what?" stone))
12	The table is made of stone.	is ("what?" made ("what?" table ("art?" the), "what?" stone ("what?" of)))
13	The table is wooden.	is ("what?" table ("art?" the), "what?" wooden)
14	Croatia is a country.	is ("what?" Croatia, "what?" country ("art?" a))
15	Peter is a lonely man.	is ("who?" Peter, "what?" man ("art?" a, "what?" lonely))
16	This clock is German.	is ("what?" clock ("what?" this), "what?" German)
17	The knife isn't sharp.	isn't ("what?" knife ("art?" the), "what?" sharp)
18	Grammar is complicated.	is ("what?" complicated ("what?" grammar))
19	The tasks are complicated.	are ("what?" complicated ("what?" tasks ("art?" the)))
20	Life is sometimes complicated.	is ("what?" complicated ("what?" life, "when?" sometimes))
21	The teacher writes with student's pencil.	writes ("who?" teacher ("art?" the), "what?" pencil ("what?" with, "whose?" student's))
22	Student's pencil is on the table.	is ("what?" pencil ("whose?" student's), "where?" table ("where?" on, "art?" the))
23	I need a lot of money.	need ("who?" I, "what?" money ("how much?" lot ("art?" a), "what?" of))
24	Andrea, buy any book for Harry.	buy ("who?" Andrea, "whom?" Harry ("whom?" for), "what?" book ("which?" any))
25	Every day is the same.	is ("what?" day ("what?" every), "what?" same ("art?" the))
26	On either side of the world people work hard every day.	work ("who?" people, "how?" hard, "when?" day ("what?" every), "where?" world ("what?" side ("what?" either ("what?" on)), "where?" of, "art?" the))
27	All my friends are good boys.	are ("who?" friends ("what?" all, "whose?" my), "what?" boys ("what?" good))
28	Somebody is in the house.	is ("who?" somebody, "where?" house ("where?" in, "art?" the))
29	She is a teacher.	is ("who?" she, "what?" teacher ("art?" a))
30	It is snowing.	is ("what?" snowing ("what?" it))
31	It is Angela's bicycle.	is ("what?" it, "what?" bicycle ("whose?" Angela's))

32	Those are stars.	are ("what?" stars, "what?" those)
33	That is doctor's car.	is ("what?" that, "what?" car ("whose?" doctor's))
34	This is your cat.	is ("what?" this, "what?" cat ("whose?" your))
35	I brush myself.	brush ("who?" I, "who?" myself)
36	He speaks about himself.	speaks ("who?" he, "who?" himself ("who?" about))
37	Jack is in the house.	is ("who?" Jack, "where?" house ("where?" in, "art?" the))
38	We don't sleep in the hotel Intercontinental.	don't ("what?" sleep ("who?" we, "where?" hotel ("where?" in, "art?" the) ("which?" Intercontinental)))
39	You can buy anything.	can ("what?" buy ("who?" you, "what?" anything))
40	Thomas buys 10 pounds of potatoes in the store.	buys ("who?" Thomas, "what?" potatoes ("how many?" pounds ("how many?" 10), "what?" of), "where?" store ("where?" in, "art?" the))
41	Nobody came.	came ("who?" nobody)
42	There was no-one on the beach.	was ("where?" there, "who?" no-one, "where?" beach ("where?" on, "art?" the))
43	There is nothing left.	is ("where?" there, "what?" left ("what?" nothing))
44	Everybody is afraid of him.	is ("what?" afraid, "who?" everybody, "whom?" him ("whom?" of))
45	These actors are Rob's friends.	are ("who?" actors ("which?" these), "what?" friends ("whose?" Rob's))
46	That car is red.	is ("what?" car ("which?" that), "what?" red)
47	This boy is Joseph's brother.	is ("who?" boy ("which?" this), "what?" brother ("whose?" Joseph's))
48	Tom has two brothers.	has ("who?" Tom, "what?" brothers ("how many?" two))
49	There are seventy-two students in the hotel.	are ("where?" there, "who?" students ("how many?" seventy-two), "where?" hotel ("where?" in, "art?" the))
50	Vanessa was the fifteenth person to win the award since 1999.	was ("who?" Vanessa, "what?" person ("what?" fifteenth ("art?" the)), "what?" award ("what?" to, "what?" win, "art?" the) ("when?" 1999 ("when?" since)))
51	Monica went to Italy two times this year.	went ("who?" Monica, "where?" Italy ("where?" to), "how many?" times ("how many?" two), "when?" year ("which?" this))
52	The car moves slowly.	moves ("what?" car ("art?" the), "how?" slowly)
53	The bears eat greedily.	eat ("who?" bears ("art?" the), "how?" greedily)
54	Mark drives a car very quickly today.	drives ("who?" Mark, "what?" car ("art?" a), "when?" today, "how?" quickly ("how?" very))
55	You look absolutely fabulous!	look ("who?" you, "how?" fabulous ("how?" absolutely))
56	Mary plays the violin extremely well.	plays ("who?" Mary, "what?" violin ("art?" the), "how?" well ("how?" extremely))
57	The book is on the table.	is ("what?" book ("art?" the), "where?" table ("where?" on, "art?" the))
58	Students talk about the solution.	talk ("who?" students, "what?" solution ("what?" about, "art?" the))
59	I see a teacher.	see ("who?" I, "who?" teacher ("art?" a))
60	The train to New York is fast.	is ("what?" train ("art?" the) ("where?" New York ("where?" to)), "what?" fast)
61	The boy is very young.	is ("who?" boy ("art?" the), "what?" young ("how?" very))
62	Peter has an office in New York.	has ("who?" Peter, "what?" office ("art?" an), "where?" New York ("where?" in))
63	Rob has an office 200 meters from the sea in	has ("who?" Rob, "what?" office ("art?" an), "where?" sea ("how many?" meters ("how many?" 200), "where?" from,

	Dubrovnik.	"art?" the), "where?" Dubrovnik ("where?" in))
64	Tom is quite a good boy.	is ("who?" Tom, "what?" boy ("what?" good ("how?" quite, "art?" a)))
65	I work hard.	work ("who?" I, "how?" hard)
66	Noa gave this book to a friend.	gave ("who?" Noa, "what?" book ("which?" this), "who?" friend ("who?" to, "art?" a))
67	Maya has some interesting books.	has ("who?" Maya, "what?" books ("how many?" some, "what?" interesting))
68	I often go to the cinema.	go ("who?" I, "when?" often, "where?" cinema ("where?" to, "art?" the))
69	London is the capital of Great Britain.	is ("what?" London, "what?" capital ("art?" the) ("what?" Great Britain ("what?" of)))
70	Mary is Lucy's friend.	is ("who?" Mary, "what?" friend ("whose?" Lucy's))
71	Father is a good man.	is ("who?" father, "what?" man ("art?" a, "what?" good))
72	Beauty is not forever.	is not ("what?" beauty, "what?" forever)
73	Honesty is the best policy.	is ("what?" honesty, "what?" policy ("what?" best ("art?" the)))
74	Mary's father is old.	is ("who?" father ("whose?" Mary's), "what?" old)
75	It is six o'clock.	is ("what?" it, "what time?" o'clock ("what time?" six))
76	Lucy goes to the gym three times a week.	goes ("who?" Lucy, "how often?" week ("how many?" times ("how many?" three), "art?" a), "where?" gym ("where?" to, "art?" the))
77	School is 50 meters from our house.	is ("what?" school, "where?" house ("how far?" meters ("how far?" 50), "where?" from, "which?" our))
78	John saw me on the street yesterday.	saw ("who?" John, "whom?" me, "where?" street ("where?" on, "art?" the), "when?" yesterday)
79	Tom is younger than Peter.	is ("who?" Tom, "what?" younger ("who?" Peter ("who?" than)))
80	Joseph is the youngest.	is ("who?" Joseph, "what?" youngest ("art?" the))
81	Competitions are in the last week of April.	are ("what?" competitions, "when?" April ("when?" week ("when?" in, "what?" last ("art?" the)), "what?" of))
82	I do my homework every day at five o'clock.	do ("who?" I, "what?" homework ("whose?" my), "when?" day ("what?" every), "when?" o'clock ("when?" at, "when?" five))
83	Lucas does his homework every day at seven o'clock.	does ("who?" Lucas, "what?" homework ("whose?" his), "when?" day ("what?" every), "when?" o'clock ("when?" at, "when?" seven))
84	He can read.	can ("what?" read ("who?" he))
85	You can touch sister's notebook.	can ("what?" touch ("who?" you, "what?" notebook ("whose?" sister's))
86	We could be friends.	could ("what?" be ("who?" we, "what?" friends))
87	I will work tomorrow.	will ("what?" work ("who?" I, "when?" tomorrow))
88	Lucas eats a sandwich.	eats ("who?" Lucas, "what?" sandwich ("art?" a))
89	Students eat chocolate.	eat ("who?" students, "what?" chocolate)
90	Our ship crossed the Atlantic in 7 days.	crossed ("what?" ship ("whose?" our), "what?" Atlantic ("art?" the), "what time?" days ("what time?" in, "how many?" 7))
91	Dinner is Lucas's favorite meal.	is ("what?" dinner, "what?" meal ("what?" favorite, "whose?" Lucas's))
92	His father's car was stolen.	was ("what?" stolen ("what?" car ("whose?" father's ("whose?" his))))
93	You want a good book.	want ("who?" you, "what?" book ("art?" a, "which?" good))

94	It rains.	rains ("what?" it)
95	There are 30 students in the classroom.	are ("where?" there, "who?" students ("how many?" 30), "where?" classroom ("where?" in, "art?" the))
96	Students play basketball every Saturday.	play ("who?" students, "what?" basketball, "when?" Saturday ("what?" every))
97	A man is going on a journey	is ("what?" going ("who?" man ("art?" a), "where?" journey ("where?" on, "art?" a)))
98	Students individually write a seminar paper during the whole semester.	write ("who?" students, "how?" individually, "what?" paper ("art?" a, "what?" seminar), "when?" semester ("what?" during, "art?" the, "what?" whole))
99	Scientists talk about global warming on the TV show.	talk ("who?" scientists, "what?" warming ("what?" about, "what?" global), "where?" show ("where?" on, "art?" the, "what?" TV))
100	Mary doesn't look tired.	doesn't ("what?" look ("who?" Mary, "how?" tired))



Tablica 23. Pitanja engleskog jezika i pripadni QFNOK-ovi

RBR	QTENG	QFNOK
1	Who eats?	eats ("who?" X)
2	Who eats greedily?	eats ("who?" X, _ greedily)
3	Who eats chocolate?	eats ("who?" X, _ chocolate)
4	How do the bears eat?	eat ( (& do) ( _ bears ( _ the), "how?" X))
5	Does Lucas eat a sandwich?	eat ( (& does) ( _ Lucas, _ sandwich ( & a)))
6	Who has an office in New York?	has ("who?" X, _ office ( _ an), _ New York ( _ in))
7	Who has an office in Dubrovnik?	has ("who?" X, _ office ( _ an), _ Dubrovnik ( _ in))
8	Who has an office?	has ("who?" X, _ office ( _ an))
9	Where does Peter have an office?	have ( (& does) ( _ Peter, _ office ( _ an), "where?" X))
10	What does Peter have in New York?	have ( (& does) ( _ Peter, "what?" X, _ New York ( _ in)))
11	Where does Rob have an office in Dubrovnik?	have ( (& does) ( _ Rob, _ office ( _ an), _ Dubrovnik ( _ in), "where?" X))
12	How many meters from the sea does Rob have an office in Dubrovnik?	have ( (& does) ( _ Rob, _ office ( _ an), _ Dubrovnik ( _ in), _ sea ( _ meters ("how many?" X), _ from, _ the)))
13	Who has some interesting books?	has ("who?" X, _ books ( _ interesting, _ some)))
14	What does Maya have?	have ( (& does) ( _ Maya, "what?" X))
15	How many interesting books does Maya have?	have ( (& does) ( _ Maya, _ books ( _ interesting, "how many?" X)))
16	What kind of books does Maya have?	have ( (& does) ( _ Maya, _ books ("what?" X ( & kind, & of))))
17	Who has two brothers?	has ("who?" X, _ brothers ( _ two))
18	What does Tom have?	have ( (& does) ( _ Tom, "what?" X))
19	How many brothers does Tom have?	have ( (& does) ( _ Tom, _ brothers ("how many?" X)))
20	What moves slowly?	moves ("what?" X, _ slowly)
21	How does the car move?	move ( (& does) ( _ car ( & the), "how?" X))
22	What is in the last week of April?	is ("what?" X, _ April ( _ of, _ week ( _ in, _ last ( _ the))))
23	When are competitions?	are ( _ competitions, "when?" X)
24	In what week of April are competitions?	are ( _ competitions, _ April ( _ of, _ week ( _ in, "what?" X))
25	Are competitions in the last week of April?	are ( _ competitions, _ April ( _ of, _ week ( _ in, _ last ( _ the))))
26	Are competitions in the last week of May?	are ( _ competitions, _ May ( _ of, _ week ( _ in, _ last ( _ the))))
27	What are those?	are ("what?" X, _ those)
28	What are stars?	are ( _ stars, "what?" X)
29	What is complicated?	is ( _ complicated ("what?" X))
30	What is sometimes complicated?	is ( _ complicated ("what?" X, _ sometimes))
31	Is life sometimes complicated?	is ( _ complicated ( _ life, _ sometimes))
32	How many students are	are ( _ there, _ students ("how many?" X))

	there?	
33	Where are 30 students?	are ("where?" X, _ students ( _ 30))
34	Who is in the hotel?	is ("who?" X, _ hotel ( _ the, _ in))
35	Where are the students?	are ("where?" X, _ students ( & the))
36	Who are good boys?	are ("who?" X, _ boys ( _ good))
37	What kind of boys are all my friends?	are ( _ friends ( _ my, _ all), _ boys ("what?" X ( & kind, & of)))
38	Whose friends are good boys?	are ( _ friends ("whose?" X), _ boys ( _ good))
39	Who are Rob's friends?	are ("who?" X, _ friends ( _ Rob's))
40	What are my friends like?	are ( _ friends ( _ my), "what?" X ( & like))
41	Whose friends are these actors?	are ( _ actors ( _ these), _ friends ("whose?" X))
42	Who should buy any book?	buy ( (& should) ("who?" X, _ book ( _ any)))
43	Who can buy anything?	can ( _ buy ("who?" X, _ anything))
44	Who buys?	buys ("who?" X)
45	What should Andrea buy Harry?	buy ( (& should) ( _ Harry, "what?" X, _ Andrea))
46	What can you buy?	can ( _ buy ( _ you, "what?" X))
47	Whom should Andrea buy any book for?	buy ( (& should) ( _ book ( _ any), "whom?" X ( _ for), _ Andrea))
48	Which book should Andrea buy?	buy ( (& should) ( _ Andrea, _ book ("which?" X)))
49	What does Thomas buy?	buy ( (& does) ( _ Thomas, "what?" X))
50	How many pounds of potatoes does Thomas buy?	buy ( (& does) ( _ Thomas, _ potatoes ( _ pounds ("how many?" X), _ of)))
51	Where does Thomas buy potatoes?	buy ( (& does) ( _ Thomas, _ potatoes, "where?" X))
52	Who came?	came ("who?" X)
53	What can he do?	can ( (& do) "what?" X ( _ he))
54	What can you do?	can ( (& do) "what?" X ( _ you))
55	Who can touch sister's notebook?	can ("what?" touch ("who?" X, _ notebook ( _ sister's)))
56	Who can buy anything?	buy ( (& can) ("who?" X, _ anything))
57	Whose notebook can you touch?	touch ( (& can) ( _ you, _ notebook ("whose?" X)))
58	What could we be?	be ( (& could) ( _ we, "what?" X))
59	What crossed the Atlantic in 7 days?	crossed ("what?" X, _ Atlantic ( _ the), _ days ( _ in, _ 7))
60	What did our ship cross?	cross ( (& did) ( _ ship ( _ our), "what?" X))
61	In what time did our ship cross the Atlantic?	cross ( (& did) ( _ ship ( _ our), _ Atlantic ( _ the), "what time?" X))
62	Who does homework every day?	does ("who?" X, _ homework, _ day ( _ every))
63	What do I do every day?	do ( (& do) ( _ I, "what?" X, _ day ( _ every)))
64	When does Lucas do his homework?	do ( (& does) ( _ Lucas, _ homework ( _ his), "when?" X))
65	Who draws?	draws ("who?" X)
66	What does Joseph draw with?	draw ( (& does) ( _ Joseph, "what?" X ( _ with)))
67	What crayons does Joseph	draw ( (& does) ( _ Joseph, _ crayons ( _ with, "what?"

	draw with?	X)))
68	Who drives?	drives ("who?" X)
69	Who drives a car?	drives ("who?" X, _ car ( & a))
70	Whose car does Peter drive?	drive ( (& does) ( _ Peter, _ car ("whose?" X)))
71	How does Mark drive today?	drive ( (& does) ( _ Mark, _ today, "how?" X))
72	When does Mark drive very quickly?	drive ( (& does) ( _ Mark, "when?" X, _ quickly ( _ very)))
73	What kind of a car does Catherine drive?	drive ( (& does) ( _ Catherine, _ car ( & a, "what?" X ( & kind, & of)))
74	What car does Catherine drive in the race?	drive ( (& does) ( _ Catherine, _ car ( "what?" X), _ race ( _ the, _ in)))
75	Where does Catherine drive a red car?	drive ( (& does) ( _ Catherine, _ car ( _ a, _ red), "where?" X))
76	What does Jacob drive?	drive ( (& does) ( _ Jacob, "what?" X))
77	Did Noa give this book to a friend?	give ( (& did) ( _ Noa, _ book ( _ this), _ friend ( _ to, _ a)))
78	Who did Noa give this book to?	give ( (& did) ( _ Noa, _ book ( _ this), "who?" X ( _ to)))
79	Where we don't sleep?	don't ( _ sleep ( _ we, "where?" X))
80	Where we do not sleep?	do_not ( _ sleep ( _ we, "where?" X))
81	Where do I often go?	go ( (& do) ( _ I, _ often, "where?" X))
82	Where is the man going?	going ( (& is) ( _ man ( & the), "where?" X))
83	How many times a week does Lucy go to the gym?	go ( (& does) ( _ Lucy, _ week ( _ a, _ times ("how many?" X)), _ gym ( _ to, _ the)))
84	How often does Lucy go to the gym?	go ( (& does) ( _ Lucy, "how often?" X, _ gym ( _ to, _ the)))
85	Where does Lucy go?	go ( (& does) ( _ Lucy, "where?" X))
86	How does not Mary look?	does_not ( _ look ( _ Mary, "how?" X)))
87	How doesn't Mary look?	doesn't ( _ look ( _ Mary, "how?" X))
88	How do you look?	look ( (& do) ( _ you, "how?" X))
89	How much money do I need?	need ( (& do) ( _ I, _ money ("how much?" X)))
90	What do I need?	need ( (& do) ( _ I, "what?" X))
91	Who plays?	plays ("who?" X)
92	How does Mary play the violin?	play ( (& does) ( _ Mary, _ violin ( _ the), "how?" X))
93	What does Mary play extremely well?	play ( (& does) ( _ Mary, "what?" X, _ well ( _ extremely)))
94	When do the students play basketball?	play ( (& do) ( _ students ( & the), _ basketball, "when?" X))
95	What do the students play?	play ( (& do) ( _ students ( & the), "what?" X))
96	Does it rain?	rain ( (& does) ( _ it))
97	Is it snowing?	is ( _ snowing ( _ it))
98	Where did John see me yesterday?	see ( (& did) ( _ John, _ me, "where?" X, _ yesterday))
99	When did John see me?	see ( (& did) ( _ John, _ me, "when?" X))
100	Who do I see?	see ( (& do) ( _ I, "who?" X))
101	Who sees a teacher?	sees ("who?" X, _ teacher ( & a))
102	Who speaks about himself?	speaks ("who?" X, _ himself ( _ about))
103	Who talks?	talks ("who?" X)
104	Who does he speak about?	speak ( (& does) ( _ he, "who?" X ( _ about)))
105	What do the scientists talk	talk ( (& do) ( _ scientists ( & the), "what?" X ( &

	about?	about)))
106	Where do the scientists talk?	talk ( (& do) (_ scientists (& the), "where?" X))
107	What do the students talk about?	talk ( (& do) (_ students (& the), "what?" X (_ about)))
108	What do you want?	want ( (& do) (_ you, "what?" X))
109	Which book do you want?	want ( (& do) (_ you, _ book ("which?" X)))
110	What was stolen?	was (_ stolen ("what?" X))
111	Who was on the beach?	was ("who?" X, _ beach (_ on, _ the))
112	What was Vanessa?	was (_ Vanessa, "what?" X)
113	Who I brush?	brush (_ I, "who?" X)
114	Where did Monica go?	go ( (& did) (_ Monica, "where?" X))
115	How many times this year did Monica go to Italy?	go ( (& did) (_ Monica, _ Italy (_ to), _ times ("how many?" X), _ year (_ this)))
116	When did Monica go to Italy?	go ( (& did) (_ Monica, _ Italy (_ to), "when?" X))
117	Who goes?	goes ("who?" X)
118	Who is going?	is (_ going ("who?" X))
119	Who went?	went ("who?" X)
120	Who will work?	will (_ work ("who?" X))
121	Who works hard?	works ("who?" X, _ hard)
122	Where do people work hard?	work (_ people, _ hard, "where?" X)
123	When will I work?	will (_ work (_ I, "when?" X))
124	When do people work?	work ( (& do) (_ people, "when?" X))
125	When do the students write?	write ( (& do) (_ students (& the), "when?" X))
126	What does the teacher write with?	write ( (& does) (_ teacher (& the), "what?" X (& with)))
127	How do the students write a seminar paper?	write ( (& do) (_ students (& the), _ paper (_ seminar, _ a), "how?" X))
128	Who writes?	writes ("who?" X)
129	What is black?	is ("what?" X, _ black)
130	What is the same?	is ("what?" X, _ same (_ the))
131	Where is the pencil?	is (_ pencil (& the), "where?" X)
132	Whose pencil is on the table?	is (_ pencil ("whose?" X), _ table (_ the, _ on))
133	What is on the table?	is ("what?" X, _ table (_ the, _ on))
134	What kind of a table is it?	is (& it, _ table ("what?" X (& kind, & of), (& a)))
135	What is wooden?	is ("what?" X, _ wooden)
136	What is the table made of?	is (_ made (_ table (_ the), "what?" X(_ of)))
137	What is made of stone?	is (_ made ("what?" X, _ stone (_ of)))
138	Who is afraid of him?	is (_ afraid, "who?" X, _ him (_ of))
139	What is dinner?	is (_ dinner, "what?" X)
140	Whose favorite meal is dinner?	is (_ dinner, _ meal (_ favorite, "whose?" X))
141	What does Lucas do?	X ( (& does), (& do), _ Lucas)
142	What is honesty?	is (_ honesty, "what?" X)
143	What is the best policy?	is ("what?" X, _ policy (_ best (_ the)))
144	What time is it?	is (_ it, "what time?" X)
145	What is London?	is (_ London, "what?" X)
146	Where is school?	is (_ school, "where?" X)
147	How far is school from our house?	is (_ school, _ house ("how far?" X, _ from, _ our))
148	What is fast?	is ("what?" X, _ fast)
149	What is left?	is (_ left ("what?" X))

150	How young is the boy?	is ( _ boy ( & the), _ young ("how?" X))
151	What is Tom?	is ( _ Tom, "what?" X)
152	Who is Joseph's brother?	is ("who?" X, _ brother ( _ Joseph's))
153	Whose father is old?	is ( _ father ("whose?" X), _ old)
154	What is father like?	is ( _ father, "what?" X ( & like))
155	Who is the youngest?	is ("who?" X, _ youngest ( _ the))
156	Who is young?	is ("who?" X, _ young)
157	Who is younger than Peter?	is ("who?" X, _ younger ( _ Peter ( _ than)))
158	Whose friend is Mary?	is ( _ Mary, _ friend ("whose?" X))
159	Who is in the house?	is ("who?" X, _ house ( _ in, _ the))
160	What is Croatia?	is ( _ Croatia, "what?" X)
161	What kind of clock is this?	is ( _ clock ( _ this), "what?" X ( & kind, & of))
162	Who is a good man?	is ("who?" X, _ man ( _ good, _ a))
163	What kind of a man is Peter?	is ( _ Peter, _ man ( _ a, "what?" X ( & kind, & of)))
164	Who is a lonely man?	is ("who?" X, _ man ( _ a, _ lonely))
165	What does Peter do?	X ( (& does), (& do), _ Peter)
166	Whose bicycle is it?	is ( _ it, _ bicycle ("whose?" X))
167	Whose car is that?	is ( _ that, _ car ("whose?" X))
168	What is red?	is ("what?" X, _ red)
169	What is not forever?	is_not ("what?" X, _ forever)
170	What isn't sharp?	isn't ("what?" X, _ sharp)
171	What is not sharp?	is_not ("what?" X, _ sharp))
172	Whose cat is this?	is ( _ this, _ cat ("whose?" X))
173	What is she?	is ( _ she, "what?" X)
174	What do the students do?	X ( (& do), (& do) ( _ students ( & the)))

Tablica 24. Odgovori sustava na pitanja (engleski jezik)

ANSWER_ID	PRGQ	ORDQ	QTENG	Očekivani odgovor	ANSWER sustava	Analiza odgovora	PRGS	ORDS
1	333	1	Who eats?	The bears	the bears	Točno	333	53
2	333	1	Who eats?	Lucas	Lucas	Točno	333	88
3	333	1	Who eats?	Students	students	Točno	333	89
4	333	2	Who eats greedily?	The bears	the bears	Točno	333	53
5	333	3	Who eats chocolate?	Students	students	Točno	333	89
6	333	4	How do the bears eat?	greedily	greedily	Točno	333	53
7	333	5	Does Lucas eat a sandwich?	YES	YES	Točno	333	88
8	333	6	Who has an office in New York?	Peter	Peter	Točno	333	62
9	333	7	Who has an office in Dubrovnik?	Rob	Rob	Točno	333	63
10	333	8	Who has an office?	Peter	Peter	Točno	333	62
11	333	8	Who has an office?	Rob	Rob	Točno	333	63
12	333	9	Where does Peter have an office?	in New York	in New York	Točno	333	62
13	333	10	What does Peter have in New York?	an office	an office	Točno	333	62
14	333	11	Where does Rob have an office in Dubrovnik?	200 meters from the sea	200 meters from the sea	Točno	333	63
15	333	12	How many meters from the sea does Rob have an office in Dubrovnik?	200	200	Točno	333	63
16	333	13	Who has some interesting books?	Maya	Maya	Točno	333	67
17	333	14	What does Maya have?	some interesting books	some interesting books	Točno	333	67
18	333	15	How many interesting books does Maya have?	some	some	Točno	333	67
19	333	16	What kind of books does Maya have?	interesting books	interesting	Točno	333	67
20	333	17	Who has two brothers?	Tom	Tom	Točno	333	48
21	333	18	What does Tom have?	two brothers	two brothers	Točno	333	48
22	333	19	How many brothers does Tom have?	two	two	Točno	333	48
23	333	20	What moves slowly?	The car	the car	Točno	333	52
24	333	21	How does the car move?	slowly	slowly	Točno	333	52
25	333	22	What is in the last week of April?	Competitions	competitions	Točno	333	81

26	333	23	When are competitions?	in the last week of April	in the last week of April	Točno	333	81
27	333	24	In what week of April are competitions?	the last	the last	Točno	333	81
28	333	25	Are competitions in the last week of April?	YES	YES	Točno	333	81
29	333	26	Are competitions in the last week of May?	NO	I DO NOT KNOW	Netočno		
30	333	27	What are those?	stars	stars	Točno	333	32
31	333	28	What are stars?	those	those	Točno	333	32
32	333	29	What is complicated?	Grammar	grammar	Točno	333	18
33	333	29	What is complicated?	The tasks	the tasks	Točno	333	19
34	333	29	What is complicated?	Life	life	Točno	333	20
35	333	30	What is sometimes complicated?	Life	life	Točno	333	20
36	333	31	Is life sometimes complicated?	YES	YES	Točno	333	20
37	333	32	How many students are there?	seventy-two	seventy-two	Točno	333	49
38	333	32	How many students are there?	30	30	Točno	333	95
39	333	33	Where are 30 students?	in the classroom	in the classroom	Točno	333	95
40	333	34	Who is in the hotel?	seventy-two students	seventy-two students	Točno	333	49
41	333	35	Where are the students?	in the hotel	in the hotel	Točno	333	49
42	333	35	Where are the students?	in the classroom	in the classroom	Točno	333	95
43	333	36	Who are good boys?	All my friends	all my friends	Točno	333	27
44	333	37	What kind of boys are all my friends?	good	good	Točno	333	27
45	333	38	Whose friends are good boys?	mine / my friends	my	Semantički ispravan, ali gramatički neispravan odgovor	333	27
46	333	39	Who are Rob's friends?	These actors	these actors	Točno	333	45
47	333	40	What are my friends like?	good boys	good boys	Točno	333	27
48	333	41	Whose friends are these actors?	Rob's	Rob's	Točno	333	45
49	333	42	Who should buy any book?	Andrea	Andrea	Točno	333	24
50	333	43	Who can buy anything?	You	you	Točno	333	39
51	333	44	Who buys?	Andrea	Andrea	Točno	333	24

52	333	44	Who buys?	you	you	Točno	333	39
53	333	44	Who buys?	Thomas	Thomas	Točno	333	40
54	333	45	What should Andrea buy Harry?	Any book	any book	Točno	333	24
55	333	46	What can you buy?	Anything	anything	Točno	333	39
56	333	47	Whom should Andrea buy any book for?	Harry	Harry	Točno	333	24
57	333	48	Which book should Andrea buy?	Any	any	Točno	333	24
58	333	49	What does Thomas buy?	10 pounds of potatoes	10 pounds of potatoes	Točno	333	40
59	333	50	How many pounds of potatoes does Thomas buy?	10 pounds	10	Točno	333	40
60	333	51	Where does Thomas buy potatoes?	in the store	in the store	Točno	333	40
61	333	52	Who came?	Nobody	nobody	Točno	333	41
62	333	53	What can he do?	read	read	Točno	333	84
63	333	54	What can you do?	buy anything	buy	Točno	333	39
64	333	54	What can you do?	touch sister's notebook	touch	Točno	333	85
65	333	55	Who can touch sister's notebook?	You	you	Točno	333	85
66	333	56	Who can buy anything?	You	you	Točno	333	39
67	333	57	Whose notebook can you touch?	sister's	sister's	Točno	333	85
68	333	58	What could we be?	Friends	friends	Točno	333	86
69	333	59	What crossed the Atlantic in 7 days?	Our ship	our ship	Točno	333	90
70	333	60	What did our ship cross?	the Atlantic	the Atlantic	Točno	333	90
71	333	61	In what time did our ship cross the Atlantic?	in 7 days	in 7 days	Točno	333	90
72	333	62	Who does homework every day?	I	I	Točno	333	82
73	333	62	Who does homework every day?	Lucas	Lucas	Točno	333	83
74	333	63	What do I do every day?	My homework	my homework	Točno	333	82
75	333	64	When does Lucas do his homework?	every day at seven o'clock.	at seven o'clock	Semantički nepotpun odgovor	333	83
76	333	65	Who draws?	Joseph	Joseph	Točno	333	9
77	333	66	What does Joseph draw with?	With wooden crayons	crayons	Točno	333	9



78	333	67	What crayons does Joseph draw with?	wooden	wooden	Točno	333	9
79	333	68	Who drives?	Daniel	Daniel	Točno	333	1
80	333	68	Who drives?	Jacob	Jacob	Točno	333	2
81	333	68	Who drives?	Thomas	Thomas	Točno	333	3
82	333	68	Who drives?	John	John	Točno	333	4
83	333	68	Who drives?	My friend Sean	my friend Sean	Točno	333	5
84	333	68	Who drives?	Catherine	Catherine	Točno	333	6
85	333	68	Who drives?	Catherine	Catherine	Točno	333	7
86	333	68	Who drives?	Peter	Peter	Točno	333	8
87	333	68	Who drives?	Mark	Mark	Točno	333	54
88	333	69	Who drives a car?	Thomas	Thomas	Točno	333	3
89	333	69	Who drives a car?	John	John	Točno	333	4
90	333	69	Who drives a car?	My friend Sean	my friend Sean	Točno	333	5
91	333	69	Who drives a car?	Catherine	Catherine	Točno	333	6
92	333	69	Who drives a car?	Catherine	Catherine	Točno	333	7
93	333	69	Who drives a car?	Peter	Peter	Točno	333	8
94	333	69	Who drives a car?	Mark	Mark	Točno	333	54
95	333	70	Whose car does Peter drive?	His brother's	his brother's	Točno	333	8
96	333	71	How does Mark drive today?	very quickly	very quickly	Točno	333	54
97	333	72	When does Mark drive very quickly?	today	today	Točno	333	54
98	333	73	What kind of a car does Catherine drive?	a red car	red	Točno	333	6
99	333	73	What kind of a car does Catherine drive?	a green car	green	Točno	333	7
100	333	74	What car does Catherine drive in the race?	a green car	green	Točno	333	7
101	333	75	Where does Catherine drive a red car?	on the highway	on the highway	Točno	333	6
102	333	76	What does Jacob drive?	A truck	a truck	Točno	333	2
103	333	77	Did Noa give this book to a friend?	YES	YES	Točno	333	66
104	333	78	Who did Noa give this book to?	To a friend	friend	Semantički nepotpun odgovor	333	66
105	333	79	Where we don't sleep?	in the hotel Intercontinental	in the hotel	Semantički nepotpun odgovor	333	38

106	333	80	Where we do not sleep?	in the hotel Intercontinental	in the hotel	Semantički nepotpun odgovor	333	38
107	333	81	Where do I often go?	to the cinema	to the cinema	Točno	333	68
108	333	82	Where is the man going?	on a journey	on a journey	Točno	333	97
109	333	83	How many times a week does Lucy go to the gym?	three times	three	Točno	333	76
110	333	84	How often does Lucy go to the gym?	three times a week	three times a week	Točno	333	76
111	333	85	Where does Lucy go?	to the gym	to the gym	Točno	333	76
112	333	86	How does not Mary look?	tired	tired	Točno	333	100
113	333	87	How doesn't Mary look?	tired	tired	Točno	333	100
114	333	88	How do you look?	absolutely fabulous	absolutely fabulous	Točno	333	55
115	333	89	How much money do I need?	a lot	a lot	Točno	333	23
116	333	90	What do I need?	a lot of money	a lot of money	Točno	333	23
117	333	91	Who plays?	Mary	Mary	Točno	333	56
118	333	91	Who plays?	Students	students	Točno	333	96
119	333	92	How does Mary play the violin?	extremely well	extremely well	Točno	333	56
120	333	93	What does Mary play extremely well?	The violin	the violin	Točno	333	56
121	333	94	When do the students play basketball?	every Saturday	every Saturday	Točno	333	96
122	333	95	What do the students play?	basketball	basketball	Točno	333	96
123	333	96	Does it rain?	YES	YES	Točno	333	94
124	333	97	Is it snowing?	YES	YES	Točno	333	30
125	333	98	Where did John see me yesterday?	on the street	on the street	Točno	333	78
126	333	99	When did John see me?	Yesterday	yesterday	Točno	333	78
127	333	100	Who do I see?	A teacher	a teacher	Točno	333	59
128	333	101	Who sees a teacher?	I	I	Točno	333	59
129	333	102	Who speaks about himself?	He	he	Točno	333	36
130	333	103	Who talks?	Students	students	Točno	333	58
131	333	103	Who talks?	Scientists	scientists	Točno	333	99
132	333	104	Who does he speak about?	himself	himself	Točno	333	36
133	333	105	What do the scientists talk about?	about global	about global	Točno	333	99

				warming	warming			
134	333	106	Where do the scientists talk?	on the TV show	on the TV show	Točno	333	99
135	333	107	What do the students talk about?	about the solution	solution	Semantički nepotpun odgovor	333	58
136	333	108	What do you want?	a good book	a good book	Točno	333	93
137	333	109	Which book do you want?	a good book	good	Točno	333	93
138	333	110	What was stolen?	His father's car	his father's car	Točno	333	92
139	333	111	Who was on the beach?	No-one	no-one	Točno	333	42
140	333	112	What was Vanessa?	the fifteenth person to win the award since 1999	to win the award	Semantički nepotpun odgovor	333	50
141	333	113	Who I brush?	Myself	myself	Točno	333	35
142	333	114	Where did Monica go?	to Italy	to Italy	Točno	333	51
143	333	115	How many times this year did Monica go to Italy?	two times	two	Točno	333	51
144	333	116	When did Monica go to Italy?	this year	this year	Točno	333	51
145	333	117	Who goes?	Monica	Monica	Točno	333	51
146	333	117	Who goes?	I	I	Točno	333	68
147	333	117	Who goes?	Lucy	Lucy	Točno	333	76
148	333	118	Who is going?	A man	a man	Točno	333	97
149	333	119	Who went?	Monica	Monica	Točno	333	51
150	333	119	Who went?	I	I	Točno	333	68
151	333	119	Who went?	Lucy	Lucy	Točno	333	76
152	333	120	Who will work?	I	I	Točno	333	87
153	333	121	Who works hard?	People	people	Točno	333	26
154	333	121	Who works hard?	I	I	Točno	333	65
155	333	122	Where do people work hard?	On either side of the world	on either side of the world	Točno	333	26
156	333	123	When will I work?	Tomorrow	tomorrow	Točno	333	87
157	333	124	When do people work?	every day	every day	Točno	333	26
158	333	125	When do the students write?	During the whole semester	during the whole semester	Točno	333	98

159	333	126	What does the teacher write with?	with student's pencil	with student's pencil	Točno	333	21
160	333	127	How do the students write a seminar paper?	individually	individually	Točno	333	98
161	333	128	Who writes?	The teacher	the teacher	Točno	333	21
162	333	128	Who writes?	Students	students	Točno	333	98
163	333	129	What is black?	The car	the car	Točno	333	10
164	333	130	What is the same?	Every day	every day	Točno	333	25
165	333	131	Where is the pencil?	on the table	on the table	Točno	333	22
166	333	132	Whose pencil is on the table?	Student's	student's	Točno	333	22
167	333	133	What is on the table?	Student's pencil	student's pencil	Točno	333	22
168	333	133	What is on the table?	The book	the book	Točno	333	57
169	333	134	What kind of a table is it?	stone	stone	Točno	333	11
169a	333	134a	What kind of a table is it?	wooden		Nedostaje odgovor	333	13
170	333	135	What is wooden?	the table	the table	Točno	333	13
171	333	136	What is the table made of?	stone	stone	Točno	333	12
172	333	137	What is made of stone?	The table	the table	Točno	333	12
173	333	138	Who is afraid of him?	Everybody	everybody	Točno	333	44
174	333	139	What is dinner?	Lucas's favorite meal	favorite Lukas's meal	Semantički ispravan, ali gramatički neispravan odgovor	333	91
175	333	140	Whose favorite meal is dinner?	Lukas's	Lukas's	Točno	333	91
176	333	141	What does Lucas do?	does his homework	does	Semantički nepotpun odgovor	333	83
177	333	141	What does Lucas do?	eats a sandwich	eats	Točno	333	88
178	333	142	What is honesty?	the best policy	the best policy	Točno	333	73
179	333	143	What is the best policy?	honesty	honesty	Točno	333	73
180	333	144	What time is it?	six o'clock	six o'clock	Točno	333	75
181	333	145	What is London?	the capital of Great Britain	of Great Britain	Semantički nepotpun	333	69

						odgovor		
182	333	146	Where is school?	50 meters from our house	50 meters from our house	Točno	333	77
183	333	147	How far is school from our house?	50 meters	50 meters	Točno	333	77
184	333	148	What is fast?	The train to New York	the train	Semantički nepotpun odgovor	333	60
185	333	149	What is left?	nothing	nothing	Točno	333	43
186	333	150	How young is the boy?	very	very	Točno	333	61
187	333	151	What is Tom?	a good boy	quite a good boy	Točno	333	64
188	333	151	What is Tom?	younger than Peter	than Peter younger	Semantički ispravan, ali gramatički neispravan odgovor	333	79
189	333	152	Who is Joseph's brother?	This boy	this boy	Točno	333	47
190	333	153	Whose father is old?	Mary's	Mary's	Točno	333	74
191	333	154	What is father like?	a good man	a good man	Točno	333	71
192	333	154	What is father like?	Old	old	Točno	333	74
193	333	155	Who is the youngest?	Joseph	Joseph	Točno	333	80
194	333	156	Who is young?	The boy	the boy	Točno	333	61
195	333	157	Who is younger than Peter?	Tom	Tom	Točno	333	79
196	333	158	Whose friend is Mary?	Lucy's	Lucy's	Točno	333	70
197	333	159	Who is in the house?	Somebody	somebody	Točno	333	28
198	333	159	Who is in the house?	Jack	Jack	Točno	333	37
199	333	160	What is Croatia?	a country	a country	Točno	333	14
200	333	161	What kind of clock is this?	German	German	Točno	333	16
201	333	162	Who is a good man?	Father	father	Točno	333	71
202	333	163	What kind of a man is Peter?	a lonely man	lonely	Točno	333	15
203	333	164	Who is a lonely man?	Peter	Peter	Točno	333	15
204	333	165	What does Peter do?	drives his brother's car	drives	Točno	333	8
205	333	165	What does Peter do?		is	Netočno	333	15
206	333	165	What does Peter do?		has	Netočno	333	62

207	333	165	What does Peter do?		younger	Netočno	333	79
208	333	166	Whose bicycle is it?	Angela's	Angela's	Točno	333	31
209	333	167	Whose car is that?	doctor's	doctor's	Točno	333	33
210	333	168	What is red?	That car	that car	Točno	333	46
211	333	169	What is not forever?	Beauty	beauty	Točno	333	72
212	333	170	What isn't sharp?	The knife	the knife	Točno	333	17
213	333	171	What is not sharp?	The knife	the knife	Točno	333	17
214	333	172	Whose cat is this?	yours / your cat	your	Semantički ispravan, ali gramatički neispravan odgovor	333	34
215	333	173	What is she?	a teacher	a teacher	Točno	333	29
216	333	174	What do the students do?		are	Netočno	333	49
217	333	174	What do the students do?	talk about the solution	talk	Točno	333	58
218	333	174	What do the students do?	eat chocolate	eat	Točno	333	89
219	333	174	What do the students do?		are	Netočno	333	95
220	333	174	What do the students do?	play basketball every Saturday	play	Točno	333	96
221	333	174	What do the students do?	individually write a seminar paper	write	Točno	333	98

Tablica 25. TENG rečenice i pripadni FNOK zapisi u oba sustava

RBR	TENG	FNOK (GBS)	FNOK (RDBS)
1	They stare.	stare("who?" they)	stare ("who?" they)
2	Everything glows.	glows("what?" everything)	glows ("what?" everything)
3	Tom swims.	swims("who?" Tom)	swims ("who?" Tom)
4	Pencil writes.	writes("what?" a pencil)	writes ("what?" pencil ("art?" a))
5	The car drives.	drives("what?" the car)	drives ("what?" car ("art?" the))
6	A boy stares.	stares("who?" a boy)	stares ("who?" boy ("art?" a))
7	This girl sings.	sings("who?" girl("which?" this))	sings ("who?" girl ("which?" this))
8	Someone's lamp glows.	glows("what?" lamp("whose?" someone's))	glows ("what?" lamp ("whose?" someone's))
9	Tom's brother swims.	swims("who?" brother("whose?" Tom's))	swims ("who?" brother ("whose?" Tom's))
10	Student's pencil writes.	writes("what?" pencil("whose?" student's))	writes ("what?" pencil ("whose?" student's))
11	My car drives.	drives("what?" car("whose?" my))	drives ("what?" car ("whose?" my))
12	The red car drives.	drives("what?" the car("which?" red))	drives ("what?" car ("art?" the, "which?" red))
13	A tall boy stares.	stares("what?" a boy("which?" tall))	stares ("who?" boy ("art?" a, "which?" tall))
14	Several girls sing.	sing("who?" girls("how_many?" several))	sing ("who?" girls ("how_many?" several))
15	Someone's electrical lamp glows.	glows("what?" lamp("whose?" someone's, "what_kind?" electrical))	glows ("what?" lamp ("whose?" someone's, "what?" electrical))
16	This artificial system works.	works("what?" system("which?" this, "what_kind?" artificial))	works ("what?" system ("which?" artificial ("which?" this)))
17	Enough information remains.	remains("what?" information("how_much?" enough))	remains ("what?" information ("how_much?" enough))
18	American's very tall boy stares.	stares("who?" boy("whose?" American's, "which?" tall("how?" very)))	stares ("who?" boy ("whose?" American's, "which?" tall ("how?" very)))
19	My incredibly tall boy stares.	stares("whose?" my "how?" incredibly "who?" boy("which?" tall))	stares ("who?" boy ("whose?" my, "which?" tall ("how?" incredibly)))
20	Julia's very incredibly tall boy stares.	stares("who?" boy("whose?" Julia's, "which?" tall("how?" incredibly("how?" very))))	stares ("who?" boy ("whose?" Julia's, "which?" tall ("how?" incredibly ("how?" very))))
21	The red car on the road drives.	drives("what?" the car("which?" red) "where?" on "what?" the road)	drives ("what?" car ("art?" the, "which?" red), "where?" road ("where?" on, "art?" the))
22	Several girls from behind the green door sing.	sing("who?" girls("how_many?" several) "where?" from "where?" behind "what?" the door("what?" green))	sing ("who?" girls ("how_many?" several), "where?" door ("where?" behind ("where?" from),

			"art?" the, "what?" green))
23	Julia's very incredibly tall boy two feet behind me stares.	stares("who?" boy("whose?" Julia's, "which?" tall("how?" incredibly("how?" very))) "how_many?" two "what?" feet "where?" behind "who?" me)	stares ("who?" boy ("whose?" Julia's, "which?" tall ("how?" incredibly ("how?" very))), "where?" me ("where?" behind ("what?" feet ("how_many?" two))))
24	Enough information right on the paper remains.	remains("what?" information("how_much?" enough) "how?" right "where?" on "what?" the paper)	remains ("what?" information ("how_much?" enough), "where?" paper ("where?" right, "where?" on, "art?" the))
25	This artificial system two feet from behind the table works.	works("what?" system("which?" this, "what_kind?" artificial) "how_many?" two "what?" feet "where?" from "where?" behind "what?" the table)	works ("what?" system ("which?" artificial ("which?" this)), "where?" table ("where?" behind ("what?" feet ("how_many?" two), "where?" from), "art?" the))
26	American's very tall boy right from behind the door stares.	stares("who?" boy("whose?" American's, "which?" tall("how?" very)) "how?" right "where?" from "where?" behind "what?" the door)	stares ("who?" boy ("whose?" American's, "which?" tall ("how?" very)), "where?" door ("art?" the, "where?" behind ("where?" from ("what?" right))))
27	I look myself.	look("who?" I, "whom?" myself)	look ("who?" I, "whom?" myself)
28	The student reads a wonderful book.	reads("who?" the student, "what?" a book("what?" wonderful))	reads ("who?" student ("art?" the), "what?" book ("art?" a, "what?" wonderful))
29	Julia writes a letter to a friend	writes("who?" Julia, "what?" a letter "where?" to "what?" a friend)	writes ("who?" Julia, "what?" letter ("art?" a) "whom?" friend ("whom?" to, "art?" a))
30	He drove my red car on Monday.	drove("who?" he, "what?" car("whose?" my, "which?" red) "when?" on "who?" Monday)	drove ("who?" he, "what?" car ("whose?" my, "which?" red), "when?" Monday ("when?" on))
31	A boy gives an apology to the girl.	gives("who?" a boy, "what?" an apology "where?" to "who?" the girl)	gives ("who?" boy ("art?" a), "what?" apology ("art?" an), "whom?" girl ("whom?" to, "art?" the))
32	She makes the room red.	makes("who?" she, "what?" the room "which?" red)	makes ("what?" red, "who?" she, "what?" room ("art?" the))
33	The view makes him incredibly happy on the hill.	makes("what?" the view, "who?" him "how?" incredibly "what?" happy "where?" on "what?" the hill)	makes ("what?" view ("art?" the), "whom?" him "what?" happy ("how?" incredibly), "where?" hill ("where?" on, "art?" the))
34	The toy is very old.	is("what?" the toy, "what?" old("how?" very))	is ("what?" toy ("art?" the), "what?" old ("how?" very))
35	Julia swims fast on the pool.	swims("who?" Julia, "how?" fast "where?" on "what?" the pool)	swims ("who?" Julia, "how?" fast, "where?" pool ("where?"



			on, "art?" the))
36	Girls are on the beach.	are("who?" girls, "where?" on "what?" the beach)	are ("who?" girls, "where?" beach ("where?" on, "art?" the))
37	I talk with my book.	talk("who?" I, "where?" with "whose?" my book)	talk ("who?" I, "what?" book ("what?" with, "whose?" my))
38	I talk about the solution with my colleague.	talk("who?" I, "where?" about "what?" the solution "where?" with "whose?" my colleague)	talk ("who?" I, "what?" solution ("what?" about, "art?" the), "who?" colleague ("who?" with, "whose?" my))
39	Tom has two cars.	has("who?" Tom, "how_many?" two "what?" cars)	has ("who?" Tom, "what?" cars ("how_many?" two))
40	Tom drives a car today.	drives("who?" Tom, "what?" a car "when?" today)	drives ("who?" Tom, "what?" car ("art?" a), "when?" today)
41	Tom drove red car on Monday.	drove("who?" Tom, "what?" car("which?" red) "when?" on "who?" Monday)	drove ("who?" Tom, "what?" car ("which?" red), "when?" Monday ("when?" on))
42	Tom drove Mery's red car on Monday.	drove("who?" Tom, "what?" car("whose?" Mery's, "which?" red) "when?" on "who?" Monday)	drove ("who?" Tom, "what?" car ("whose?" Mary's, "which?" red), "when?" Monday ("when?" on))

Tablica 26. QTENG pitanja i pripadni QFNOK zapisi u oba sustava

RBR	QTENG	QFNOK (BDAS)	QFNOK (RDBS)
1	Are girls on the beach?	are ( _ girls, _ on _ the beach	are ( _ girls, _ beach ( _ on, _ the))
2	Who is on the beach?	is ("who?" X, & _ on _ the beach	is ("who?" X, _ beach ( _ on, _ the))
3	Who are on the beach?	are ("who?" X, & _ on _ the beach	are ("who?" X, _ beach ( _ on, _ the))
4	Where are girls?	are ( _ girls, & "where?" X	are ( _ girls, "where?" X)
5	To whom does Julia write a letter?	write ( _ Julia, _ a letter _ to _ X	write ( (& does), _ Julia, _ letter ( _ a), "whom?" X ( _ to))
6	Who writes a friend a letter?	writes ("who?" X, _ a friend _ a letter	writes ("who?" X, _ letter ( _ a), _ friend ( _ to, _ a))
7	Who write a letter to a friend?	write ("who?" X, _ a letter _ to _ a friend	write ("who?" X, _ letter ( _ a), _ friend ( _ to, _ a))
8	What does Julia do to a friend?	X ( _ Julia & _ to _ a friend	X ( (& does), (& do), _ Julia, _ friend ( _ to, _ a))
9	What does Julia do on the pool?	X ( _ Julia & _ on _ the pool	X ( (& does), (& do), _ Julia, _ pool ( _ on, _ the))
10	Where does Julia swim?	swim ( _ Julia & "where?" X	swim ( (& does), _ Julia, "where?" X)
11	How does Julia swim?	swim ( _ Julia, "how?" X	swim ( (& does), _ Julia, "how?" X)
12	Who swim fast on the pool?	swim ("who?" X, _ fast _ on _ the pool	swims ("who?" X, _ fast, _ pool ( _ on, _ the))
13	What does Julia's very incredibly tall boy do two feet behind me?	X ( _ boy ( _ Julia's, _ tall ( _ incredibly ( _ very))) _ two _ feet _ behind _ me	X ( (& does), (& do), _ boy ( _ Julia's, _ tall ( _ incredibly ( _ very))) _ me ( _ behind ( _ feet ( _ two)))
14	What does Julia's very incredibly tall boy do?	X ( _ boy ( _ Julia's, _ tall ( _ incredibly ( _ very)))	X ( (& does), (& do), _ boy ( _ Julia's, _ tall ( _ incredibly ( _ very)))
15	Who swim?	swim ("who?" X	swim ("who?" X)
16	Does a tall boy stare?	stare ( _ a boy ( _ tall)	stare ( (& does), _ boy ( _ a, _ tall))
17	How old is the toy?	is ( _ the toy, _ old ("how?" X)	is ( _ toy ( _ the), _ old ("how?" X))
18	Is the toy very old?	is ( _ the toy, _ old ( _ very)	is ( _ toy ( _ the), _ old ( _ very))
19	What did Tom do with Mery's red car on Monday?	X ( _ Tom & _ car ( _ red) _ on _ Monday	X ( (& did), (& do), (& with), _ Tom, _ car ( _ Mary's, _ red), _ Monday ( _ on))
20	What did Tom do with red car?	X ( _ Tom & _ car ( _ red)	X ( (& did), (& do), (& with), _ Tom, _ car ( _ red))
21	What do I do about the solution with my colleague?	X ( _ I, _ about _ the solution _ with _ my colleague	X ( (& do), (& do), _ I, _ solution ( _ about, _ the), _ colleague ( _ with, _ my))
22	With whose colleague I talk about the solution?	talk ( _ I, & with _ X & _ about _ the solution	talk ( _ I, _ solution ( _ about, _ the), _ colleague ( _ with, "whose?" X))
23	What do I do with my book?	X ( _ I, _ with _ my book	X ( (& do), (& do), _ I, _ book ( _ with, _ my))
24	What do I do?	X ( _ I	X ( (& do), (& do), _ I)

25	What do several girls do?	X (_ girls (_ several)	X ( (& do), (& do), _ girls (_ several))
26	What do several girls from behind the green door do?	X (_ girls (_ several) _ from _ behind _ the door (_ green)	X ( (& do), (& do), _ girls (_ several), _ door (_ behind (_ from), _ the, _ green))
27	What do they do?	X (_ they	X ( (& do), (& do), _ they)
28	What does someone's electrical lamp do?	X (_ lamp (_ someone's, _ electrical)	X ( (& does), (& do), _ lamp (_ someone's, _ electrical))
29	How many cars has Tom?	has (_ Tom, "how_many?" X _ cars	has (_ Tom, _ cars ("how_many?" X))
30	What does Tom do today?	X (_ Tom & _ today	X ( (& does), (& do), _ Tom, _ today)
31	What does Tom do with a car today?	X (_ Tom & _ a car _ today	X ( (& does), (& do), _ Tom, _ car (_ a), _ today, & with)
32	What does Tom do?	X (_ Tom	X ( (& does), (& do), _ Tom)
33	What does Tom's brother do?	X (_ brother (_ Tom's	X ( (& does), (& do), _ brother (_ Tom's))
34	Who give an apology to the girl?	give ("who?" X, _ an apology _ to _ the girl	give ("who?" X, _ apology (_ an), _ girl (_ to, _ the))
35	What does a boy do to the girl?	X (_ a boy, _ Y & _ to _ the girl	X ( (& does), (& do), _ boy (_ a), _ girl (_ to, _ the))
36	What does a boy do?	X (_ a boy	X ( (& does), (& do), _ boy (_ a))
37	What does a pencil do?	X (_ a pencil	X ( (& does), (& do), _ pencil (& a))
38	What does a tall boy do?	X (_ a boy (_ tall)	X ( (& does), (& do), _ boy (_ a, _ tall))
39	What does American's very tall boy do?	X (_ boy (_ American's, _ tall (_ very)	X ( (& does), (& do), _ boy (_ American's, _ tall (_ very)))
40	What does American's very tall boy right from behind the door do?	X (_ boy (_ American's, _ tall (_ very)) _ right _ from _ behind _ the door	X ( (& does), (& do), _ boy (_ American's, _ tall (_ very)), _ door (_ the, _ behind (_ from (_ right))))
41	What does enough information do?	X (_ information (_ enough)	X ( (& does), (& do), _ information (_ enough))
42	What does enough information right on the paper do?	X (_ information (_ enough) _ right _ on _ the paper	X ( (& does), (& do), _ information (_ enough), _ paper (_ right, _ on, _ the))
43	What does everything do?	X (_ everything	X ( (& does), (& do), _ everything)
44	What does my car do?	X (_ car (_ my	X ( (& does), (& do), _ car (_ my))
45	What does my incredibly tall boy do?	X (_ my _ incredibly _ boy (_ tall)	X ( (& does), (& do), _ boy (_ my, _ tall (_ incredibly)))
46	What does she do to the room red?	X (_ she & _ the room _ red	X ( (& does), (& do), _ red, _ she, _ room (_ the))
47	What does someone's lamp do?	X (_ lamp (_ someone's	X ( (& does), (& do), _ lamp (_ someone's))
48	What does the car do?	X (_ the car	X ( (& does), (& do), _ car (_ the))
49	What does the red car do?	X (_ the car (_ red)	X ( (& does), (& do), _ car (_ red))

50	What does the red car on the road do?	X (_ the car (_ red) _ on _ the road	X ( (& does), (& do), _ car (_ red, _ the), _ road (_ on, _ the))
51	What does the student do?	X (_ the student	X ( (& does), (& do), _ student (_ the))
52	What does student's pencil do?	X (_ pencil (_ student's	X ( (& does), (& do), _ pencil (_ student's))
53	What does this artificial system do?	X (_ system (_ this, _ artificial)	X ( (& does), (& do), _ system (_ artificial (_ this)))
54	What does this girl do?	X (_ girl (_ this	X ( (& does), (& do), _ girl (_ this))
55	What drives on the road?	drives ("what?" X, & _ on _ the road	drives ("what?" X, _ road (_ on, _ the))
56	What drives?	drives ("what?" X	drives ("what?" X)
57	What glows?	glows ("what?" X	glows ("what?" X)
58	What is very old?	is ("what?" X, _ old (_ very)	is ("what?" X, _ old (_ very))
59	What makes him incredibly happy?	makes ("what?" X & _ incredibly _ happy	makes ("what?" X, _ him, _ happy (_ incredibly))
60	What remains?	remains ("what?" X	remains ("what?" X)
61	Which system works two feet from behind the table?	works (_ system (_ X) & _ two _ feet _ from _ behind _ the table	works (_ system ("which?" X), _ table (_ behind (_ feet (_ two), _ from), _ the))
62	What works?	works ("what?" X	works ("what?" X)
63	What writes?	writes ("what?" X	writes ("what?" X)
64	Who drove Mery's red car?	drove (_ X, _ car (_ Mery's, _ red)	drove ("who?" X, _ car (_ Mary's, _ red))
65	When did Tom drove Mery's red car?	drove (_ Tom, _ car (_ Mery's, _ red) "when?" X	drive ( (& did), _ Tom, _ car (_ Mary's, _ red), "when?" X)
66	When does Tom drive a car?	drive (_ Tom, _ a car "when?" X	drive ( (& does), _ Tom, _ car (& a), "when?" X)
67	Who drive today?	drive ("who?" X & _ today	drive ("who?" X, _ today)
68	Who drive?	drive ("who?" X	drive ("who?" X)
69	Where does enough information remain?	remain (_ information (_ enough) & "where?" X	remain ( (& does) _ information (_ enough), "where?" X)
70	Which girl sings?	sings (_ girl, "which?" X	sings (_ girl ("which?" X))
71	Who drove my red car on Monday?	drove ("who?" X, _ car (_ my, _ red) _ on _ Monday	drove ("who?" X, _ car (_ my, _ red), _ Monday (_ on))
72	Who make the room red?	make ("who?" X, _ the room _ red	make (_ red, "who?" X, _ room (_ the))
73	Who read a wonderful book?	read ("who?" X, _ a book (_ wonderful)	read ("who?" X, _ book (_ a, _ wonderful))
74	Who sing?	sing ("who?" X	sing ("who?" X)
75	Who stare?	stare ("who?" X	stare ("who?" X)
76	Who talk about the solution with my colleague?	talk ("who?" X, & _ about _ the solution _ with _ my colleague	talk ("who?" X, _ solution (_ about, _ the), _ colleague (_ with, _ my))
77	With what I talk?	talk (_ I, _ with _ X	talk (_ I, "what?" X (_ with))
78	Whose brother swims?	swims (_ brother, "whose?" X	swims (_ brother ("whose?" X))
79	Whose car drives?	drives (_ car, "whose?" X	drives (_ car ("whose?" X))
80	Whose lamp glows?	glows (_ lamp, "whose?" X	glows (_ lamp ("whose?" X))
81	Whose pencil writes?	writes (_ pencil, "whose?" X	writes (_ pencil ("whose?" X))

82	Are girls on the pool?	are (_ girls, _ on _ the pool	are (_ girls, _ pool (_ on, _ the))
83	Whom do I look?	look (_ I, _ X)	look (_ I, "whom?" X)
84	Who looks himself?	looks ("who?" X, _ himself	looks ("who?" X, _ himself)
85	Who look myself?	look ("who?" X, _ myself	look ("who?" X, _ myself)
86	Who is him?	is (_ him & "who?" X	is ("who?" X, _ him)
87	Whose red car did he drive on Monday?	drive (_ he, _ car (_ X, _ red) _ on _ Monday	drive ( (& did), _ he, _ car ("whose?" X, _ red), _ Monday (_ on))
88	Whose red car did Tom drive on Monday?	drive (_ Tom, _ car (_ X, _ red) _ on _ Monday	drive ( (& did), _ Tom, _ car ("whose?" X, _ red), _ Monday (_ on))

Tablica 27. Odgovori dvaju sustava na pitanja te analiza rezultata dvaju sustava

RBR	RB QTENG	QTENG	Answer (GBS)	RBR TENG (GBS)	Answer (RDBS)	RBR TENG (RDBS)	Analiza (GBS)	ANALIZA (RDBS)	Usporedba
1	1	Are girls on the beach?	YES	36	YES	36	Točno	Točno	isto
2	2	Who is on the beach?	I DO NOT KNOW		girls	36	Netočno	Točno	bolji
3	3	Who are on the beach?	girls	36	girls	36	Točno	Točno	isto
4	4	Where are girls?	on the beach	36	on the beach	36	Točno	Točno	isto
5	5	To whom does Julia write a letter?	a friend	29	friend	29	Točno	Semantički nepotpun odgovor	lošiji
6	6	Who writes a friend a letter?	I DO NOT KNOW		Julia	29	Netočno	Točno	bolji
7	7	Who write a letter to a friend?	Julia	29	Julia	29	Točno	Točno	isto
8	8	What does Julia do to a friend?	writes	29	writes	29	Točno	Točno	isto
9	9	What does Julia do on the pool?	swims	35	swims	35	Točno	Točno	isto
10	10	Where does Julia swim?	on the pool	35	on the pool	35	Točno	Točno	isto
11	11	How does Julia swim?	fast	35	fast	35	Točno	Točno	isto
12	12	Who swim fast on the pool?	Julia	35	Julia	35	Točno	Točno	isto
13	13	What does Julia's very incredibly tall boy do two feet behind me?	stares	23	stares	23	Točno	Točno	isto
14	14	What does Julia's very incredibly tall boy do?	stares	20	stares	20	Točno	Točno	isto
15	14	What does Julia's very incredibly tall boy do?	stares	23	stares	23	Točno	Točno	isto
16	15	Who swim?	Tom	3	Tom	3	Točno	Točno	isto
17	15	Who swim?	Tom's brother	9	Tom's brother	9	Točno	Točno	isto
18	15	Who swim?	Julia	35	Julia	35	Točno	Točno	isto
19	16	Does a tall boy stare?	YES	13	YES	13	Točno	Točno	isto
20	17	How old is the toy?	very	34	very	34	Točno	Točno	isto
21	18	Is the toy very old?	YES	34	YES	34	Točno	Točno	isto

22	19	What did Tom do with Mery's red car on Monday?	drove	42	drove	42	Točno	Točno	isto
23	20	What did Tom do with red car?	drove	41	drove	41	Točno	Točno	isto
24	20	What did Tom do with red car?	drove	42	drove	42	Točno	Točno	isto
25	21	What do I do about the solution with my colleague?	talk	38	talk	38	Točno	Točno	isto
26	22	With whose colleague I talk about the solution?	my colleague	38	my	38	Točno	Semantički ispravan, ali gramatički neispravan odgovor	lošiji
27	23	What do I do with my book?	talk	37	talk	37	Točno	Točno	isto
28	24	What do I do?	look	27	look	27	Točno	Točno	isto
29	24	What do I do?	talk	37	talk	37	Točno	Točno	isto
30	24	What do I do?	talk	38	talk	38	Točno	Točno	isto
31	25	What do several girls do?	sing	14	sing	14	Točno	Točno	isto
32	25	What do several girls do?	sing	22	sing	22	Točno	Točno	isto
33	26	What do several girls from behind the green door do?	sing	22	sing	22	Točno	Točno	isto
34	27	What do they do?	stare	1	stare	1	Točno	Točno	isto
35	28	What does someone's electrical lamp do?	glows	15	glows	15	Točno	Točno	isto
36	29	How many cars has Tom?	two	39	two	39	Točno	Točno	isto
37	30	What does Tom do today?	drives	40	drives	40	Točno	Točno	isto
38	31	What does Tom do with a car today?	drives	40	drives	40	Točno	Točno	isto
39	32	What does Tom do?	swims	3	swims	3	Točno	Točno	isto
40	32	What does Tom do?	has	39	has	39	Netočno	Netočno	isto
41	32	What does Tom do?	drives	40	drives	40	Točno	Točno	isto
42	32	What does Tom do?	drove	41	drove	41	Točno	Točno	isto
43	32	What does Tom do?	drove	42	drove	42	Točno	Točno	isto
44	33	What does Tom's brother do?	swims	9	swims	9	Točno	Točno	isto
45	34	Who give an apology to the girl?	a boy	31	a boy	31	Točno	Točno	isto
46	35	What does a boy do to the girl?	gives an	31	gives	31	Točno	Točno	isto

			apology						
47	36	What does a boy do?	stares	6	stares	6	Točno	Točno	isto
48	36	What does a boy do?	stares	13	stares	13	Točno	Točno	isto
49	36	What does a boy do?	gives	31	gives	31	Točno	Točno	isto
50	37	What does a pencil do?	writes	4	writes	4	Točno	Točno	isto
51	37	What does a pencil do?			writes	10	Nedostaje odgovor	Točno	bolji
52	38	What does a tall boy do?	stares	13	stares	13	Točno	Točno	isto
53	39	What does American's very tall boy do?	stares	18	stares	18	Točno	Točno	isto
54	39	What does American's very tall boy do?	stares	26	stares	26	Točno	Točno	isto
55	40	What does American's very tall boy right from behind the door do?	stares	26	stares	26	Točno	Točno	isto
56	41	What does enough information do?	remains	17	remains	17	Točno	Točno	isto
57	41	What does enough information do?	remains	24	remains	24	Točno	Točno	isto
58	42	What does enough information right on the paper do?	remains	24	remains	24	Točno	Točno	isto
59	43	What does everything do?	glows	2	glows	2	Točno	Točno	isto
60	44	What does my car do?	drives	11	drives	11	Točno	Točno	isto
61	44	What does my car do?			drove	30	Nedostaje odgovor	Točno	bolji
62	45	What does my incredibly tall boy do?	stares	19	stares	19	Točno	Točno	isto
63	46	What does she do to the room red?	makes	32	makes	32	Točno	Točno	isto
64	47	What does someone's lamp do?	glows	8	glows	8	Točno	Točno	isto
65	47	What does someone's lamp do?	glows	15	glows	15	Točno	Točno	isto
66	48	What does the car do?	drives	5	drives	5	Točno	Točno	isto
67	48	What does the car do?	drives	12	drives	12	Točno	Točno	isto



68	48	What does the car do?	drives	21	drives	21	Točno	Točno	isto
69	49	What does the red car do?	drives	12	drives	12	Točno	Točno	isto
70	49	What does the red car do?	drives	21	drives	21	Točno	Točno	isto
71	49	What does the red car do?			drove	30	Nedostaje odgovor	Točno	bolji
72	49	What does the red car do?			drove	41	Nedostaje odgovor	Točno	bolji
73	49	What does the red car do?			drove	42	Nedostaje odgovor	Točno	bolji
74	50	What does the red car on the road do?	drives	21	drives	21	Točno	Točno	isto
75	51	What does the student do?	reads	28	reads	28	Točno	Točno	isto
76	52	What does student's pencil do?	writes	10	writes	10	Točno	Točno	isto
77	53	What does this artificial system do?	works	16	works	16	Točno	Točno	isto
78	53	What does this artificial system do?	works	25	works	25	Točno	Točno	isto
79	54	What does this girl do?	sings	7	sings	7	Točno	Točno	isto
80	55	What drives on the road?	the red car	21	the red car	21	Točno	Točno	isto
81	56	What drives?	the car	5	the car	5	Točno	Točno	isto
82	56	What drives?	my car	11	my car	11	Točno	Točno	isto
83	56	What drives?	the red car	12	the red car	12	Točno	Točno	isto
84	56	What drives?	the red car	21	the red car	21	Točno	Točno	isto
85	56	What drives?			my red car	30	Nedostaje odgovor	Točno	bolji
86	56	What drives?			a car	40	Nedostaje odgovor	Točno	bolji
87	56	What drives?			red car	41	Nedostaje odgovor	Točno	bolji
88	56	What drives?			Mary's red car	42	Nedostaje odgovor	Točno	bolji
89	57	What glows?	everything	2	everything	2	Točno	Točno	isto
90	57	What glows?	someone's	8	someone's	8	Točno	Točno	isto

			lamp		lamp				
91	57	What glows?	someone's electrical lamp	15	someone's electrical lamp	15	Točno	Točno	isto
92	58	What is very old?	the toy	34	the toy	34	Točno	Točno	isto
93	59	What makes him incredibly happy?	the view	33	the view	33	Točno	Točno	isto
94	60	What remains?	enough information	17	enough information	17	Točno	Točno	isto
95	60	What remains?	enough information	24	enough information	24	Točno	Točno	isto
96	61	Which system works two feet from behind the table?	this artificial	25	this artificial	25	Točno	Točno	isto
97	62	What works?	this artificial system	16	this artificial system	16	Točno	Točno	isto
98	62	What works?	this artificial system	25	this artificial system	25	Točno	Točno	isto
99	63	What writes?	a pencil	4	a pencil	4	Točno	Točno	isto
100	63	What writes?	student's pencil	10	student's pencil	10	Točno	Točno	isto
101	63	What writes?			a letter	29	Nedostaje odgovor	Točno	bolji
102	64	Who drove Mery's red car?	Tom	42	Tom	42	Točno	Točno	isto
103	65	When did Tom drove Mery's red car?	on Monday	42	on Monday	42	Točno	Točno	isto
104	66	When does Tom drive a car?	today	40	today	40	Točno	Točno	isto
105	66	When does Tom drive a car?			on Monday	41	Nedostaje odgovor	Točno	bolji
106	66	When does Tom drive a car?			on Monday	42	Nedostaje odgovor	Točno	bolji
107	67	Who drive today?	Tom	40	Tom	40	Točno	Točno	isto
108	68	Who drive?	Tom	40	Tom	40	Točno	Točno	isto
109	68	Who drive?			he	30	Nedostaje	Točno	bolji

							odgovor		
110	68	Who drive?			Tom	41	Nedostaje odgovor	Točno	bolji
111	68	Who drive?			Tom	42	Nedostaje odgovor	Točno	bolji
112	69	Where does enough information remain?	on the paper	24	right on the paper	24	Točno	Točno	isto
113	70	Which girl sings?	this	7	this	7	Točno	Točno	isto
114	71	Who drove my red car on Monday?	he	30	he	30	Točno	Točno	isto
115	72	Who make the room red?	she	32	she	32	Točno	Točno	isto
116	73	Who read a wonderful book?	the student	28	the student	28	Točno	Točno	isto
117	74	Who sing?	this girl	7	this girl	7	Točno	Točno	isto
118	74	Who sing?	several girls	14	several girls	14	Točno	Točno	isto
119	74	Who sing?	several girls	22	several girls	22	Točno	Točno	isto
120	75	Who stare?	they	1	they	1	Točno	Točno	isto
121	75	Who stare?	a boy	6	a boy	6	Točno	Točno	isto
122	75	Who stare?			a tall boy	13	Nedostaje odgovor	Točno	bolji
123	75	Who stare?	American's very tall boy	18	American's very tall boy	18	Točno	Točno	isto
124	75	Who stare?			my incredibly tall boy	19	Nedostaje odgovor	Točno	bolji
125	75	Who stare?	Julia's very incredibly tall boy	20	Julia's very incredibly tall boy	20	Točno	Točno	isto
126	75	Who stare?	Julia's very incredibly tall boy	23	Julia's very incredibly tall boy	23	Točno	Točno	isto
127	75	Who stare?	American's very tall boy	26	American's very tall boy	26	Točno	Točno	isto
128	76	Who talk about the solution with my colleague?	I	38	I	38	Točno	Točno	isto

129	77	With what I talk?	my book	37	book	37	Točno	Točno	isto
130	77	With what I talk?			colleague	38	Točno	Netočno	lošiji
131	78	Whose brother swims?	Tom's	9	Tom's	9	Točno	Točno	isto
132	79	Whose car drives?	my	11	my	11	Semantički ispravan, ali gramatički neispravan odgovor	Semantički ispravan, ali gramatički neispravan odgovor	isto
133	79	Whose car drives?			my	30	Nedostaje odgovor	Semantički ispravan, ali gramatički neispravan odgovor	isto
134	79	Whose car drives?			Mary's	42	Nedostaje odgovor	Točno	bolji
135	80	Whose lamp glows?	someone's	8	someone's	8	Točno	Točno	isto
136	80	Whose lamp glows?	someone's	15	someone's	15	Točno	Točno	isto
137	81	Whose pencil writes?	student's	10	student's	10	Točno	Točno	isto
138	82	Are girls on the pool?	NO	36	I DO NOT KNOW		Točno	Točno	isto
139	83	Whom do I look?	myself	27	myself	27	Točno	Točno	isto
140	84	Who looks himself?	I DO NOT KNOW		I DO NOT KNOW		Netočno	Netočno	isto
141	85	Who look myself?	I	27	I	27	Točno	Točno	isto
142	86	Who is him?	I DO NOT KNOW		I DO NOT KNOW		Netočno	Netočno	isto
143	87	Whose red car did he drive on Monday?	I DO NOT KNOW		my	30	Netočno	Semantički ispravan, ali gramatički neispravan odgovor	bolji
144	88	Whose red car did Tom drive on Monday?	I DO NOT KNOW		Mary's	42	Netočno	Točno	bolji

### Privitak 3. Formalizacija rečenica i pitanja

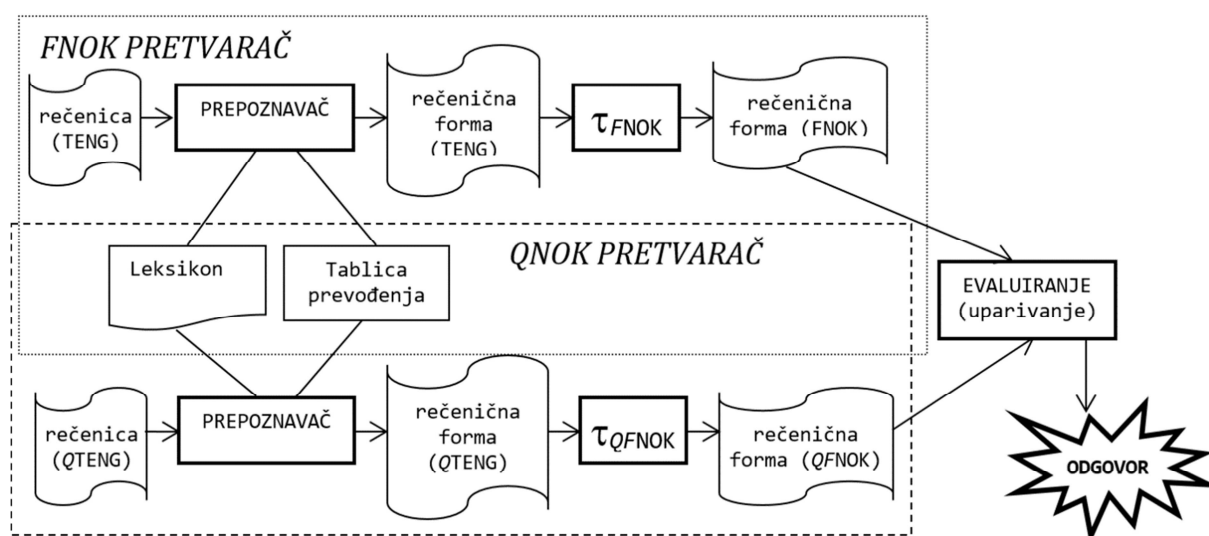
Osnovni proces [90] koji se odvija u podsustavu "Postavljanja pitanja od strane korisnika i dobivanje odgovora" (prikazan na slici 1 u Privitku 3), odnosno u podsustavu sustava QANOK preko kojega korisnik postavlja pitanja BPNOKU na koja ovaj daje odgovore, sastoji se iz sljedećih aktivnosti:

1. prihvaćanje pitanja u prirodnom engleskom jeziku (QTENG)
2. prevođenje pitanja s prirodnoga engleskog jezika u formalizirani oblik (QFNOK)
3. usporedba QFNOK s FNOK rečenicama u BPNOK-u

Prije provedbe opisanoga osnovnog procesa potrebno je bazu znanja BPNOK napuniti formaliziranim znanjem. To znači da u BPNOK trebaju postojati FNOK formalizirane rečenice koje se dobivaju prevođenjem iz njihova TENG oblika. Za potrebe razvoja sustava QANOK razvit će se sustav koji se sastoji iz dva pretvarača:

1. FNOK pretvarač koji ulaznu rečenicu napisanu u jeziku TENG prevodi u rečenicu u jeziku FNOK.
2. QFNOK pretvarač koji ulaznu rečenicu (pitanje), napisanu u jeziku QTENG prevodi u rečenicu u jeziku QFNOK.

i evaluatora koji na temelju prevedenog teksta i pitanja u FNOK i QFNOK notaciju i njihovog uparivanja pronalazi odgovor (slika 1). QFNOK pretvarač i evaluator predstavljaju dijelove QANOK sustava.



Slika 1. Model sustava QANOK

U nastavku ćemo opisati sintaksu jezika TENG i QTENG, formalizme njihova prevođenja u FNOK i QFNOK notaciju i postupak evaluiranja odgovora na postavljena pitanja.

## Jezik TENG

Jezik TENG predstavlja jednostavne rečenice engleskog jezika definirane nad rječnikom (leksikonom) danim u [90] koji sadrži sljedeće klase riječi:

<i>Af</i> - qualificative adjective	<i>Dd</i> - demonstrative determiner	<i>Di</i> - indefinite determiner	
<i>Dg</i> - general determiner	<i>Ds</i> - possessive determiner	<i>Nc</i> - common noun	
<i>Np</i> - proper noun	<i>Mc</i> - cardinal numeral	<i>Pg</i> - general pronoun	
<i>Pp</i> - personal pronoun	<i>Ps</i> - possessive pronoun	<i>Px</i> - reflexive pronoun	(1)
<i>Rm</i> - modifier adverb	<i>Rmp</i> - positive modifier adverb	<i>Rs</i> - specifier adverb	
<i>Sp</i> - preposition adposition	<i>Vm</i> - main verb		

Leksikon koji se koristio utemeljen je na rezultatima projekta "MULTEXT-East". Svaka riječ leksikona sadrži tri atributa [75]:

- word-form – izmijenjen oblik riječi,
- lemma – osnovni oblik riječi,
- MSD – morfosintaktički opis leksika – u (4) su prikazane klase riječi primjenom MSD simbola.

## Prepoznač

U [47] dana je gramatika jezika TENG koja je ovdje prikazana u reduciranom obliku:

<i>S</i>	→ NP <i>Vm</i> [ (( NP   <i>Px</i> ) [ ( NP   <i>Px</i>   PP   AP [ PP ] )   ( AP   PP ) [ PP ] ) ]	
<i>NP</i>	→ [ ( <i>Dd</i>   <i>Di</i>   <i>Dg</i>   <i>Ds</i>   <i>Nc</i> 's   <i>Np</i> 's   <i>Pg</i> 's ) ] { AP } <i>Nc</i> { PP }   <i>Np</i>   <i>Pg</i>   <i>Pp</i>	(2)
<i>AP</i>	→ [ ( <i>Rs</i>   [ <i>Rs</i> ] <i>Rm</i> ) ] <i>Af</i>	
<i>PP</i>	→ [ ( <i>Rm</i>   <i>Mc</i> <i>Nc</i> ) ] <i>Sp</i> ( NP   PP )	

Osim parsiranja (engl. *parsing*), sintaksne analize jezika  $L(G)$  definiranog gramatikom, postoji i sintaksna analiza nazvana prepoznavanje (eng. *recognizing*). U radu je rabljen prepoznač primjenljiv na sve tipove formalnih jezika, nazvanih jezici sa svojstvima [27] i [25].

Prepoznač jezika sa svojstvima je sedmorka:

$$J = (Q, \mathcal{V}, \delta, q_0, F, \alpha, M)$$

gdje su:

<i>Q</i>	konačan skup stanja
$\mathcal{V}$	rječnik, $\mathcal{V} \subseteq \Sigma^+$

- $\delta$  funkcija prijelaza, definirana kao  $\delta: Q \times \mathcal{V} \cup \{\text{@}\} \rightarrow P(Q)$  gdje je  $P(Q)$  particija od  $Q$  ; @ je oznaka kraja ulaznog niza
- $q_0$  početno stanje,  $q_0 \in Q$
- $F$  skup završnih stanja,  $F \subseteq Q$
- $\alpha$  skup akcija pridruženih svakom paru  $(q_i, s_j)$ ,  $q_i \in Q$ ,  $s_j \in \mathcal{V}$ , za koji je definirana funkcija prijelaza
- $\mathcal{M}$  pomoćna memorija

Gramatika (2) poslužila je za definiranje jezika TENG kao jezika sa svojstvima i potom je realiziran odgovarajući prepoznavać (TENG prepoznavać). Ako je ulazna rečenica sintaksno korektna prepoznavać će emitirati njezinu strukturu rečeničnoj formi (SF).

Tablica 28 prikazuje tablicu prijelaza prepoznavaća jezika TENG definiranog s gramatikom (2).

Tablica 28. Tablica prijelaza prepoznavaća jezika TENG

	Af	Dd	Di	Dg	Ds	Mc	Nc	Np	Pg	Pp	Px	Rm	Rs	Sp	Vm	Nc's	Np's	Pg's	.
<b>q</b>	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
<b>0</b>	4	1	1	9	9		11	11	11	11		3	2			9	9	9	
<b>1</b>	4						11	11	11	11		3	2						
<b>2</b>	4											3							
<b>3</b>	4																		
<b>4</b>	4						5	5				3	2						
<b>5</b>						6	6	6				6		7	12				
<b>6</b>							5							7	12				
<b>7</b>		8	8	9	9	6	11	11	11	11				7					
<b>8</b>	10	1	1	1	1	6	11												
<b>9</b>	4						11	11	11	11		3	2						
<b>10</b>							11												
<b>11</b>															12				
<b>12</b>	16	13	13	22	22		20	20	20	20	20	15	14	19		16	16	16	0
<b>13</b>	16						24	20				15	14						
<b>14</b>	16											15							
<b>15</b>	16																		0
<b>16</b>	16						17	17				15	14	19					0
<b>17</b>						18	18	18				18		19					0
<b>18</b>							17							19					0
<b>19</b>		21	21	22	22	18	20	20	20	20		18		19					0
<b>20</b>	16	21	21	22	22							15		19					0
<b>21</b>	23					18	20												

22	16					20					15	14							
23						20													
24											15	14	12						0

### Primjeri rečenica sa sintaksnom pogreškom

Julia writes a letter a friend. (Julia writes a letter **to** a friend)

GREŠKA 24 3 a 24

The red car on the road drives. (The red car **drives** on the road???)

GREŠKA 5 14 on 5

### Prevođenje

Ako je  $\Sigma$  ulazni, a  $\Delta$  izlazni rječnik, *prijevod* iz jezika  $L_1 \subseteq \Sigma^*$  u jezik  $L_2 \subseteq \Delta^*$  je relacija  $T$  iz  $\Sigma^* \times \Delta^*$ , tako da je  $L_1$  domena, a  $L_2$  kodomena od  $T$ . Rečenica  $y$ , za koju je  $(x, y)$  u  $T$ , naziva se *izlaz* od  $x$ . Umjesto  $(x, y)$  piše se

$$y = \tau(x) \text{ ili } x \rightarrow y$$

i čita "x se prevodi u y". Kaže se da je  $x$  rečenica izvornog jezika, a  $y$  rečenica ciljnog jezika. Općenito se proces prevođenja sastoji od nekoliko faza: leksičke i sintaksne analize ulaznog niza (rečenice  $x$ ) i generiranje izlazne rečenice,  $y$ . Program za prevođenje naziva se *translator*. Ako se u fazi sintaksne analize prevodioca koristi prepoznavač, takav se translator naziva *pretvarač*.

### Jezik FNOK

Općenito se struktura rečeničnih formi gramatike (2) može prikazati kao:

$$s = \alpha V_m [ \beta ]$$

Rečenice jezika FNOK dobiju se prevođenjem rečenica jezika TENG, prema pravilima:

$$\begin{aligned} \alpha V_m &\rightarrow V_m ( \tau(\alpha) ) \\ \alpha V_m \beta &\rightarrow V_m ( \tau(\alpha) , \tau(\beta) ) \end{aligned} \tag{3}$$

Pravila prevođenja su sljedeća:

$X$	$\tau(X)$	$X$	$\tau(X)$
Pp	QPp Pp	$\_d Px$	$QPx \_d Px$
Pg	QPg Pg	$\_d Af Nc$	$QN Nc ( QD \_d , QA4 Af )$
Np	QNP Np	$\_x Nc$	$QN Nc ( QD \_x )$
Nc	QN Nc	$Mc Nc$	$QN Nc ( QSpec Mc )$
QN	QN1   QNc	$Af Nc$	$QN Nc ( QA Af )$
Px	QPx Px	$Sp Np$	$QNP Np ( QS Sp )$
Dg	QDg Dg	$Sp N1$	$QNC N1 ( QRM Sp )$
Ds	QDs Ds	$Sp Pp$	$, QPp Pp ( QSp Sp )$
QA	QA1   QA2   QA3   QA4   QA5   QA6	$\_x Af Nc$	$QN Nc ( QD \_x , QA Af )$



Af	QA Af	_x Rs Af Nc	QN Nc ( QD _x , QA Af ( QR Rs ) )
Rs	, QRs Rs	_x Rs Rm Af Nc	QN Nc ( QD _x , QA Af ( QR Rm ( QR Rs ) ) )
Rm	, QRm Rm	_x Rs Rs Af Nc	QN Nc ( QD _x , QA Af ( QR Rs ( QR Rs ) ) )
Sp	QS Sp	_x Rm Af Nc	QN Nc ( QD _x , QA4 Af ( QR1 Rm ) )
Rs Af	QA Af (QRs Rs)	_d Af Nc	QNc Nc ( QD _d , QA4 Af )
Rs Af Nc	QN Nc ( QA Af ( QR Rs ) )	Sp _d Nc	, QN Nc ( QS Sp , QD _d )
_d Nc	QN Nc ( QD _d )	Sp _x Nc	, QN Nc ( QSp Sp , QD _x )
_d Px	QD _d _x		

gdje je:

```

_d = ( Dd | Di )_x = ( Dg |Ds |Pg's |Np's |Nc's )
QD = "art"
QSp = ( "which?" | "where?" | "when?" | "whom?" | "what?" )    QNp = ( "who?" | "whose?" | "whom?" )
QN1 = ( "who?" | "whose?" | "whom?" )    QNc = ( "what?" | "whose?" )
QA1 = ( "what_kind?" | "what?" ) QA2 = ( "how_much?" | "how_many?" | "what?" )
QPg = ( "what?" )    QA3 = ( "what?" | "which?" | "how?" )
QPp = ( "who?" | "whom?" )    QA5 = ( "how?" )
QDs = ( "whose?" )    QPx = ( "whom?" )
QA4 = ( "which?" | "what?" )    QDg = ( "which?" )
QA6 = ( "how_many?" )    QRm = ( "when?" )

```

Neka je  $\alpha = \alpha_0 \alpha_1 \dots \alpha_m$  i  $\beta = \varepsilon$  ili  $\beta = \beta_0 \beta_1 \dots \beta_n$  rečenična forma ulazne rečenice napisane u jeziku TENG. U (5) je prikazan algoritam prevođenja u rečeničnu formu Y u jeziku FNOK u Pythonu sličnom kodu:

```

def FNOK (X) :
    x = list (X); k = len(X); y = []
    while True :
        if  $\tau(x) <> []$ : y +=  $\tau(x)$ ; X = X[k+1:]; k = len(X)
        else : k -= 1
        x = X[:k]
        if x == [] : break
    return y

```

(5)

```

Y = [Vm, '('] +FNOK( $\alpha$ )
if  $\beta <> \varepsilon$  : Y += [','] +FNOK( $\beta$ )
Y += [')']

```

## Privitak 4. SQL upiti nad integriranom bazom podataka

U nastavku su navedeni SQL upiti koji omogućuju odgovaranje na pitanja iz tablice 8.8. Pitanja za integriranu bazu podataka. Za svako pitanje bit će naveden tekst pitanja zadan rečenicama prirodnog jezika te definirane varijable koje prosljeđujemo upitu. Tamo gdje je moguće, na istom SQL upitu bit će riješeno više pitanja. Na primjer, za prvo i drugo pitanje (*1. Na kojem projektu radi djelatnik koji programira u Javi?* i *2. Na kojem projektu radi djelatnik koji programira u Pythonu?*) koristi se isti upit. Razlika je samo u varijablama koje se prosljeđuju upitu (*programira – što – Javi – čemu –u* za prvi upit, *programira – što – Pythonu – čemu –u* za drugi upit).

### 1. Na kojem projektu radi djelatnik koji programira u Javi

*programira – čemu – Javi – čemu -u*

```
select PRJ.NAZIV
from NOK_FNOK FNOK, NOK_PARAGRAPH PRG, ESTIMUS.PMG_DJELATNIK DJL,
     ESTIMUS.PMG_DJL_PROJEKT DPR, ESTIMUS.PMG_PROJEKT PRJ
where FNOK.PRG          = PRG.PRG
     and PRG.ID_DJL_ESTIMUS = DJL.ID
     and DPR.ID_DJL       = DJL.ID
     and DPR.ID_PRJ       = PRJ.ID
     and FNOK.KB_NOD_WRD_ID in
     (select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where WRD.DICT_ID =
DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from KB_DICT where upper
(DICT) = upper (:rijec1)) or upper (WRD.WORD) = upper (:rijec1)))
     and FNOK.KB_NOD_QST_ID = (select QUEST_ID from KB_QUEST where UPPER (QUEST)
= UPPER (:quest1))
     and FNOK.KB_PAR_WRD_ID in
     (select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where WRD.DICT_ID =
DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from KB_DICT where upper
(DICT) = upper (:rijec2)) or upper (WRD.WORD) = upper (:rijec2)))
     and (FNOK.PRG, FNOK.ORD, FNOK.NODE_ID) in (select PRG, ORD, PARENT_ID from
NOK_FNOK where KB_NOD_WRD_ID in
     (select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where WRD.DICT_ID =
DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from KB_DICT where upper
(DICT) = upper (:rijec3)) or upper (WRD.WORD) = upper (:rijec3)))
     and KB_NOD_QST_ID = (select QUEST_ID from KB_QUEST where UPPER (QUEST) =
UPPER (:quest2))
     and KB_PAR_WRD_ID in (select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD
where WRD.DICT_ID = DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from
KB_DICT where upper (DICT) = upper (:rijec1)) or upper (WRD.WORD) = upper
(:rijec1))));

:rijec1 = Javi
:rijec2 = programira
:rijec3 = u
:quest1 = čemu
:quest2 = čemu
```

Tablica 29. prikazuje rezultat upita na 1. pitanje.

Tablica 29. Rezultat SQL upita za 1. pitanje

	NAZIV
1	LUMENS
2	KREDIS

## 2. Na kojem projektu radi djelatnik koji programira u Pythonu?

*programira – čemu – Pythonu – čemu –u*

```
:rijec1 = Pythonu
:rijec2 = programira
:rijec3 = u
:quest2 = čemu
:quest2 = čemu
```

Tablica 30. prikazuje rezultat upita na 2. pitanje.

Tablica 30. Rezultat SQL upita za 2. pitanje

	NAZIV
1	SMARTY

## 3. Koji djelatnik na radnom mjestu programera je rođen 01.12.1978.?

*Je – što - rođen – kada – 01.12.1978.*

```
select DJL.NAZIV
from NOK_FNOK FNOK, NOK_PARAGRAPH PRG, ESTIMUS.PMG_DJELATNIK DJL
where FNOK.PRG = PRG.PRG
and PRG.ID_DJL_ESTIMUS = DJL.ID
and DJL.RADNO_MJESTO = :radnoMjesto
and FNOK.KB_NOD_WRD_ID in
(select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where WRD.DICT_ID =
DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from KB_DICT where upper
(DICT) = upper (:rijec1)) or upper (WRD.WORD) = upper (:rijec1)))
and FNOK.KB_NOD_QST_ID =
(select QUEST_ID from KB_QUESTION where UPPER (QUEST) = UPPER (:quest1))
and FNOK.KB_PAR_WRD_ID in
(select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where WRD.DICT_ID =
DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from KB_DICT where upper
(DICT) = upper (:rijec2)) or upper (WRD.WORD) = upper (:rijec2)))
and (FNOK.PRG, FNOK.ORD, FNOK.PARENT_ID) in (select PRG, ORD, NODE_ID from
NOK_FNOK where KB_NOD_WRD_ID in
(select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where WRD.DICT_ID =
DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from KB_DICT where upper
(DICT) = upper (:rijec2)) or upper (WRD.WORD) = upper (:rijec2)))
and KB_NOD_QST_ID = (select QUEST_ID from KB_QUESTION where UPPER (QUEST) =
UPPER (:quest2))
and KB_PAR_WRD_ID in
(select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where WRD.DICT_ID =
DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from KB_DICT where upper
(DICT) = upper (:rijec3)) or upper (WRD.WORD) = upper (:rijec3)))
```

);

```
:rijec1 = 01.12.1978.  
:rijec2 = rođen  
:rijec3 = je  
:quest1 = kada  
:quest2 = što  
:radnoMjesto = programer
```

Tablica 31 prikazuje rezultat upita na 3. pitanje.

Tablica 31. Rezultat SQL upita za 3. pitanje

	NAZIV
1	Hrvoje Horvat

#### 4. Koji djelatnik na radnom mjestu projektanta je rođen 28.05.1980.?

*Je – što - rođen – kada – 28.05.1980.*

```
:rijec1 = 28.05.1980.  
:rijec2 = rođen  
:rijec3 = je  
:quest1 = kada  
:quest2 = što  
:radnoMjesto = projektant
```

Tablica 32 prikazuje rezultat upita na 4. pitanje.

Tablica 32. Rezultat SQL upita za 4. pitanje

	NAZIV
1	Jelena Ivić

#### 5. Koliko djece ima djelatnik sa datumom zaposlenja 15.03.2009.?

*ima – što - djece – koliko – X*

```
select FNOK.NODE  
from NOK_FNOK FNOK, NOK_PARAGRAPH PRG, ESTIMUS.PMG_DJELATNIK DJL  
where FNOK.PRG = PRG.PRG  
and PRG.ID_DJL_ESTIMUS = DJL.ID  
and DJL.DAT_ZAP = TO_DATE (:datum, 'DD.MM.YYYY.')
```

```
and FNOK.KB_NOD_QST_ID =  
(select QUEST_ID from KB_QUESTION where UPPER (QUEST) = UPPER (:quest1))  
and FNOK.KB_PAR_WRD_ID in (select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD  
where WRD.DICT_ID = DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from  
KB_DICT where upper (DICT) = upper (:rijec1)) or upper (WRD.WORD) =  
upper(:rijec1)))  
and (FNOK.PRG, FNOK.ORD, FNOK.PARENT_ID) in (  
select PRG, ORD, NODE_ID  
from NOK_FNOK  
where KB_NOD_WRD_ID in  
(select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where WRD.DICT_ID =  
DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from KB_DICT where upper  
(DICT) = upper (:rijec1)) or upper (WRD.WORD) = upper (:rijec1)))
```

```

        and KB_NOD_QST_ID = (select QUEST_ID from KB_QUEST where UPPER (QUEST) =
UPPER (:quest2))
        and KB_PAR_WRD_ID in (select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD
where WRD.DICT_ID = DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from
KB_DICT where upper (DICT) = upper (:rijec2)) or upper (WRD.WORD) =
upper(:rijec2)))) ;

:rijec1 = djece
:rijec2 = ima
:quest1 = koliko
:quest2 = što
:datum = 15.03.2009.

```

Tablica 33 prikazuje rezultat upita na 5. pitanje.

Tablica 33. Rezultat SQL upita za 5. pitanje

	NODE
1	dvoje

## 6. Koliko djece ima djelatnik s datumom zaposlenja 01.09.2009.?

*ima – što - djece – koliko – X*

```

:rijec1 = dijete
:rijec2 = ima
:quest1 = koliko
:quest2 = što
:datum = 01.09.2009.

```

Tablica 34 prikazuje rezultat upita na 6. pitanje.

Tablica 34. Rezultat SQL upita za 6. pitanje

	NODE
1	jedno

## 7. Koji djelatnik je završio Filozofski fakultet u Rijeci i ima login hhorvat?

*Je – što - završio – što – fakultet – koji – filozofski*

*– gdje – Rijeci – gdje – u*

```

select DJL.NAZIV
from NOK_FNOK FNOK, NOK_PARAGRAPH PRG, ESTIMUS.PMG_DJELATNIK DJL
where FNOK.PRG          = PRG.PRG
      and DJL.ID        = PRG.ID_DJL_ESTIMUS
      and DJL.LOGIN     = :login
      and KB_NOD_WRD_ID in ( select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD
where WRD.DICT_ID      = DCT.DICT_ID(+) and (DCT.DICT_PAR = ( select DICT_PAR
from KB_DICT where upper (DICT) = upper (:rijec)) or upper (WRD.WORD) = upper
(:rijec)))
      and KB_NOD_QST_ID = (select QUEST_ID from KB_QUEST where UPPER (QUEST) =
UPPER (:quest))
      and KB_PAR_WRD_ID in (select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where
WRD.DICT_ID          = DCT.DICT_ID(+) and (DCT.DICT_PAR = ( select DICT_PAR from

```

```

KB_DICT where upper (DICT) = upper (:rijec1) ) or upper (WRD.WORD) = upper
(:rijec1)))
and (FNOK.PRG, ORD, PARENT_ID) in ( select PRG, ORD, NODE_ID from NOK_FNOK
where KB_NOD_WRD_ID in ( select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD
where WRD.DICT_ID = DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from
KB_DICT where upper (DICT) = upper (:rijec1)) or upper (WRD.WORD) = upper
(:rijec1)))
and KB_NOD_QST_ID = (select QUEST_ID from KB_QUESTION where UPPER (QUEST) =
UPPER (:quest3))
and KB_PAR_WRD_ID in (select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD
where WRD.DICT_ID = DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from
KB_DICT where upper (DICT) = upper (:rijec5)) or upper (WRD.WORD) = upper
(:rijec5)))
) and (FNOK.PRG, ORD, NODE_ID) in ( select PRG, ORD, PARENT_ID from NOK_FNOK
where KB_NOD_WRD_ID in ( select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD
where WRD.DICT_ID = DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from
KB_DICT where upper (DICT) = upper (:rijec2)) or upper (WRD.WORD) = upper
(:rijec2)))
and KB_NOD_QST_ID = (select QUEST_ID from KB_QUESTION where UPPER (QUEST) =
UPPER (:quest1))
and KB_PAR_WRD_ID in (select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD
where WRD.DICT_ID = DCT.DICT_ID(+) and (DCT.DICT_PAR = ( select DICT_PAR
from KB_DICT where upper (DICT) = upper (:rijec)) or upper (WRD.WORD) = upper
(:rijec)))
) and (FNOK.PRG, ORD, NODE_ID) in (select PRG, ORD, PARENT_ID from NOK_FNOK
where KB_NOD_WRD_ID in ( select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD
where WRD.DICT_ID = DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from
KB_DICT where upper (DICT) = upper (:rijec3)) or upper (WRD.WORD) = upper
(:rijec3)))
and KB_NOD_QST_ID = (select QUEST_ID from KB_QUESTION where UPPER (QUEST) =
UPPER (:quest2))
and KB_PAR_WRD_ID in ( select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD
where WRD.DICT_ID = DCT.DICT_ID(+) and (DCT.DICT_PAR =( select DICT_PAR from
KB_DICT where upper (DICT) = upper (:rijec)) or upper (WRD.WORD) = upper
(:rijec)))
and (PRG, ORD, NODE_ID) in (select PRG, ORD, PARENT_ID from NOK_FNOK
where KB_NOD_WRD_ID in (select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where
WRD.DICT_ID = DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from
KB_DICT where upper (DICT) = upper (:rijec4)) or upper (WRD.WORD) = upper
(:rijec4)))
and KB_NOD_QST_ID =(select QUEST_ID from KB_QUESTION where UPPER
(QUEST) = UPPER (:quest2))
and KB_PAR_WRD_ID in (select WRD.WORD_ID from KB_DICT DCT, KB_WORD
WRD where WRD.DICT_ID = DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR
from KB_DICT where upper (DICT) = upper (:rijec3)) or upper (WRD.WORD) = upper
(:rijec3)))
));

```

```

:rijec = fakultet
:rijec1 = završio
:rijec2 = filozofski
:rijec3 = Rijeci
:rijec4 = u
:rijec5 = je
:quest = što

```

:quest1 = koji  
:quest2 = gdje  
:quest3 = što  
:login = hhorvat

Tablica 35 prikazuje rezultat upita na 7. pitanje.

Tablica 35. Rezultat SQL upita za 7. pitanje

	NAZIV
1	Hrvoje Horvat

### 8. Koji djelatnik je završio Filozofski fakultet u Rijeci i ima login jivic?

:rijec = fakultet  
:rijec1 = završio  
:rijec2 = filozofski  
:rijec3 = Rijeci  
:rijec4 = u  
:rijec5 = je  
:quest = što  
:quest1 = koji  
:quest2 = gdje  
:quest3 = što  
:login = jivic

Tablica 36 prikazuje rezultat upita na 8. pitanje.

Tablica 36. Rezultat SQL upita za 8. pitanje

	NAZIV
1	

Nema odgovora jer niti jedan djelatnik ne zadovoljava uvjete u upitu.

### 9. Koji djelatnik je završio Tehnički fakultet u Zagrebu i ima login jivic?

:rijec = fakultet  
:rijec1 = završio  
:rijec2 = tehnički  
:rijec3 = Zagrebu  
:rijec4 = u  
:rijec5 = je  
:quest = što  
:quest1 = koji  
:quest2 = gdje  
:quest3 = što  
:login = jivic

Tablica 37 prikazuje rezultat upita na 9. pitanje.

Tablica 37. Rezultat SQL upita za 9. pitanje

	NAZIV
1	Jelena Ivić

## 10. Na kojem projektu radi djelatnik sa VSS koji ima 5 godina radnog iskustva u projektiranju?

*ima – što – iskustva – koliko – godina – koliko – 5*

*– kakvog – radnog*

*– čemu – projektiranju – čemu - u*

```
select PRJ.NAZIV
from NOK_FNOK FNOK, NOK_PARAGRAPH PRG, ESTIMUS.PMG_DJELATNIK DJL,
     ESTIMUS.PMG_DJL_PROJEKT DPR, ESTIMUS.PMG_PROJEKT PRJ
where FNOK.PRG          = PRG.PRG
     and PRG.ID_DJL_ESTIMUS = DJL.ID
     and DPR.ID_DJL        = DJL.ID
     and DPR.ID_PRJ        = PRJ.ID
     and DJL.STRUCNA_SPREMA = :vss
     and KB_NOD_WRD_ID     in (select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD
where WRD.DICT_ID = DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from
KB_DICT where upper (DICT) = upper (:rijec1)) or upper (WRD.WORD) = upper
(:rijec1)))
     and KB_NOD_QST_ID = ( select QUEST_ID from KB_QUESTION where UPPER (QUEST) =
UPPER (:quest1))
     and KB_PAR_WRD_ID in (select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where
WRD.DICT_ID = DCT.DICT_ID(+) and (DCT.DICT_PAR = ( select DICT_PAR from
KB_DICT where upper (DICT) = upper (:rijec2) ) or upper (WRD.WORD) = upper
(:rijec2)))
     and (FNOK.PRG, ORD, NODE_ID) in ( select PRG, ORD, PARENT_ID from NOK_FNOK
where KB_NOD_WRD_ID in
( select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where WRD.DICT_ID =
DCT.DICT_ID(+) and (DCT.DICT_PAR = ( select DICT_PAR from KB_DICT where upper
(DICT) = upper (:rijec3)) or upper (WRD.WORD) = upper (:rijec3)))
     and KB_NOD_QST_ID = (select QUEST_ID from KB_QUESTION where UPPER (QUEST) =
UPPER (:quest2))
     and KB_PAR_WRD_ID in ( select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD
where WRD.DICT_ID = DCT.DICT_ID(+) and (DCT.DICT_PAR = ( select DICT_PAR
from KB_DICT where upper (DICT) = upper (:rijec1)) or upper (WRD.WORD) = upper
(:rijec1)))
     and (PRG, ORD, NODE_ID) in ( select PRG, ORD, PARENT_ID from NOK_FNOK where
KB_NOD_WRD_ID in ( select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where
WRD.DICT_ID = DCT.DICT_ID(+) and (DCT.DICT_PAR = ( select DICT_PAR from
KB_DICT where upper (DICT) = upper (:rijec4)) or upper (WRD.WORD) = upper
(:rijec4)))
     and KB_NOD_QST_ID = (select QUEST_ID from KB_QUESTION where UPPER (QUEST) =
UPPER (:quest3))
     and KB_PAR_WRD_ID in (select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where
WRD.DICT_ID = DCT.DICT_ID(+) and (DCT.DICT_PAR = ( select DICT_PAR from
KB_DICT where upper (DICT) = upper (:rijec3)) or upper (WRD.WORD) = upper
(:rijec3)))
     ))
     and (FNOK.PRG, ORD, NODE_ID) in (select PRG, ORD, PARENT_ID from NOK_FNOK
where KB_NOD_WRD_ID in (select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where
WRD.DICT_ID = DCT.DICT_ID(+) and (DCT.DICT_PAR = ( select DICT_PAR from
KB_DICT where upper (DICT) = upper (:rijec5)) or upper (WRD.WORD) = upper
(:rijec5)))
```



```

    and KB_NOD_QST_ID = (select QUEST_ID from KB_QUEST where UPPER (QUEST) =
UPPER (:quest4))
    and KB_PAR_WRD_ID in (select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where
WRD.DICT_ID = DCT.DICT_ID(+) and (DCT.DICT_PAR = ( select DICT_PAR from
KB_DICT where upper (DICT) = upper (:rijec1) ) or upper (WRD.WORD) = upper
(:rijec1)))
)
    and (FNOK.PRG, ORD, NODE_ID) in (select PRG, ORD, PARENT_ID from NOK_FNOK
where KB_NOD_WRD_ID in (select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where
WRD.DICT_ID = DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from
KB_DICT where upper (DICT) = upper (:rijec6)) or upper (WRD.WORD) = upper
(:rijec6)))
    and KB_NOD_QST_ID = (select QUEST_ID from KB_QUEST where UPPER (QUEST) =
UPPER (:quest5))
    and KB_PAR_WRD_ID in (select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD
where WRD.DICT_ID = DCT.DICT_ID(+) and (DCT.DICT_PAR = ( select DICT_PAR
from KB_DICT where upper (DICT) = upper (:rijec1)) or upper (WRD.WORD) = upper
(:rijec1)))
    and (PRG, ORD, NODE_ID) in ( select PRG, ORD, PARENT_ID from NOK_FNOK
where KB_NOD_WRD_ID in (select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where
WRD.DICT_ID = DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from
KB_DICT where upper (DICT) = upper (:rijec7)) or upper (WRD.WORD) = upper
(:rijec7)))
    and KB_NOD_QST_ID = (select QUEST_ID from KB_QUEST where UPPER
(QUEST) = UPPER (:quest6))
    and KB_PAR_WRD_ID in (select WRD.WORD_ID from KB_DICT DCT, KB_WORD
WRD where WRD.DICT_ID = DCT.DICT_ID(+) and (DCT.DICT_PAR = ( select DICT_PAR
from KB_DICT where upper (DICT) = upper (:rijec6)) or upper (WRD.WORD) = upper
(:rijec6)))
)
);

```

```

:rijec1 = iskustva
:rijec2 = ima
:rijec3 = godina
:rijec4 = 5
:rijec5 = radnog
:rijec6 = projektiranju
:rijec7 = u
:quest1 = što
:quest2 = koliko
:quest3 = koliko
:quest4 = kakvog
:quest5 = čemu
:quest6 = čemu
:vss = VSS

```

Tablica 38 prikazuje rezultat upita na 10. pitanje.

**Tablica 38. Rezultat SQL upita za 10. pitanje**

	NAZIV
1	LUMENS
2	KREDIS

## 11. Na kojem projektu radi djelatnik s VSS koji je radio na nekoliko znanstvenih projekata?

*je – što – radio – gdje – projekata – koliko – nekoliko*

*– kakvih – znanstvenih*

```
select PRJ.NAZIV
from NOK_FNOK FNOK, NOK_PARAGRAPH PRG, ESTIMUS.PMG_DJELATNIK DJL,
     ESTIMUS.PMG_DJL_PROJEKT DPR, ESTIMUS.PMG_PROJEKT PRJ
where FNOK.PRG          = PRG.PRG
     and PRG.ID_DJL_ESTIMUS = DJL.ID
     and DPR.ID_DJL        = DJL.ID
     and DPR.ID_PRJ        = PRJ.ID
     and DJL.STRUCNA_SPREMA = :vss
     and KB_NOD_WRD_ID     in
       (select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where WRD.DICT_ID =
        DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from KB_DICT where upper
        (DICT) = upper (:rijec1)) or upper (WRD.WORD) = upper (:rijec1)))
     and KB_NOD_QST_ID = (select QUEST_ID from KB_QUESTION where UPPER (QUEST) =
UPPER (:quest1))
     and KB_PAR_WRD_ID in
       (select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where WRD.DICT_ID =
        DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from KB_DICT where upper
        (DICT) = upper (:rijec2)) or upper (WRD.WORD) = upper (:rijec2)))
     and (FNOK.PRG, ORD, NODE_ID) in
       (
         select PRG, ORD, PARENT_ID
         from NOK_FNOK
         where KB_NOD_WRD_ID in
           (select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where WRD.DICT_ID =
            DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from KB_DICT where upper
            (DICT) = upper (:rijec3)) or upper (WRD.WORD) = upper (:rijec3)))
           and KB_NOD_QST_ID = (select QUEST_ID from KB_QUESTION where UPPER (QUEST) =
UPPER (:quest2))
           and KB_PAR_WRD_ID in
             (select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where WRD.DICT_ID =
              DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from KB_DICT where upper
              (DICT) = upper (:rijec1)) or upper (WRD.WORD) = upper (:rijec1)))
             and (PRG, ORD, NODE_ID) in
               (
                 select PRG, ORD, PARENT_ID
                 from NOK_FNOK
                 where KB_NOD_WRD_ID in
                   (select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where WRD.DICT_ID
= DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from KB_DICT where upper
(DICT) = upper (:rijec4)) or upper (WRD.WORD) = upper (:rijec4)))
                   and KB_NOD_QST_ID = (select QUEST_ID from KB_QUESTION where UPPER
(QUEST) = UPPER (:quest3))
                   and KB_PAR_WRD_ID in
                     (select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where WRD.DICT_ID
= DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from KB_DICT where upper
(DICT) = upper (:rijec3)) or upper (WRD.WORD) = upper (:rijec3)))
                     )
                 and (PRG, ORD, NODE_ID) in
```

```

(
  select PRG, ORD, PARENT_ID
  from NOK_FNOK
  where KB_NOD_WRD_ID in
    (select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where WRD.DICT_ID
= DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from KB_DICT where upper
(DICT) = upper (:rijec5)) or upper (WRD.WORD) = upper (:rijec5)))
    and KB_NOD_QST_ID = (select QUEST_ID from KB_QUESTION where UPPER
(QUEST) = UPPER (:quest4))
    and KB_PAR_WRD_ID in
    (select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where WRD.DICT_ID
= DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from KB_DICT where upper
(DICT) = upper (:rijec3)) or upper (WRD.WORD) = upper (:rijec3)))
)
);

```

```

:rijec1 = radila
:rijec2 = je
:rijec3 = projekata
:rijec4 = nekoliko
:rijec5 = znanstvenih
:quest1 = što
:quest2 = gdje
:quest3 = koliko
:quest4 = kakvih
:vss = VSS

```

Tablica 39 prikazuje rezultat upita na 11. pitanje.

Tablica 39. Rezultat SQL upita za 11. pitanje

	NAZIV
1	SMARTY

## 12. Koju ulogu na projektu čiji je tip projekta "održavanje" ima djelatnik koji je vjenčan?

*je – što – vjenčan*

```

select DPR.ULOGA
from NOK_FNOK FNOK, NOK_PARAGRAPH PRG, ESTIMUS.PMG_DJELATNIK DJL,
  ESTIMUS.PMG_DJL_PROJEKT DPR, ESTIMUS.PMG_PROJEKT PRJ
where FNOK.PRG = PRG.PRG
  and PRG.ID_DJL_ESTIMUS = DJL.ID
  and DPR.ID_DJL = DJL.ID
  and DPR.ID_PRJ = PRJ.ID
  and PRJ.TIP = :odrzavanje
  and FNOK.KB_NOD_WRD_ID in
  (select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where WRD.DICT_ID =
DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from KB_DICT where upper
(DICT) = upper (:rijec1)) or upper (WRD.WORD) = upper (:rijec1)))
  and FNOK.KB_NOD_QST_ID = (select QUEST_ID from KB_QUESTION where UPPER (QUEST)
= UPPER (:quest))
  and FNOK.KB_PAR_WRD_ID in

```

```
(select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where WRD.DICT_ID =
DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from KB_DICT where upper
(DICT) = upper (:rijec2)) or UPPER (WRD.WORD) = UPPER (:rijec2)));
```

```
:rijec1 = vjenčan
:rijec2 = je
:quest = što
:odrzavanje = održavanje
```

Tablica 40 prikazuje rezultat upita na 12. pitanje.

**Tablica 40. Rezultat SQL upita za 12. pitanje**

	<b>ULOGA</b>
1	Voditelj projekta

### **13. Koji je datum zaposlenja djelatnika koji dobro govori engleski jezik?**

*govori – što – jezik – koji – engleski*

*– kako – dobro*

```
select TO_CHAR (DJL.DAT_ZAP, 'DD.MM.YYYY.') as DAT_ZAP
from NOK_FNOK FNOK, NOK_PARAGRAPH PRG, ESTIMUS.PMG_DJELATNIK DJL
where FNOK.PRG = PRG.PRG
and PRG.ID_DJL_ESTIMUS = DJL.ID
and KB_NOD_WRD_ID in
(select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where WRD.DICT_ID =
DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from KB_DICT where upper
(DICT) = upper (:rijec1)) or upper (WRD.WORD) = upper (:rijec1)))
and KB_NOD_QST_ID = (select QUEST_ID from KB_QUESTION where UPPER (QUEST) =
UPPER (:quest1))
and KB_PAR_WRD_ID in
(select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where WRD.DICT_ID =
DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from KB_DICT where upper
(DICT) = upper (:rijec2)) or upper (WRD.WORD) = upper (:rijec2)))
and (FNOK.PRG, ORD, NODE_ID) in
(
select PRG, ORD, PARENT_ID
from NOK_FNOK
where KB_NOD_WRD_ID in
(select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where WRD.DICT_ID =
DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from KB_DICT where upper
(DICT) = upper (:rijec3)) or upper (WRD.WORD) = upper (:rijec3)))
and KB_NOD_QST_ID = (select QUEST_ID from KB_QUESTION where UPPER (QUEST) =
UPPER (:quest2))
and KB_PAR_WRD_ID in
(select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where WRD.DICT_ID =
DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from KB_DICT where upper
(DICT) = upper (:rijec1)) or upper (WRD.WORD) = upper (:rijec1)))
)
and (FNOK.PRG, ORD, PARENT_ID) in
(
select PRG, ORD, PARENT_ID
from NOK_FNOK
where KB_NOD_WRD_ID in
```

```

(select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where WRD.DICT_ID =
DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from KB_DICT where upper
(DICT) = upper (:rijec4)) or upper (WRD.WORD) = upper (:rijec4)))
and KB_NOD_QST_ID = (select QUEST_ID from KB_QUESTION where UPPER (QUEST) =
UPPER (:quest3))
and KB_PAR_WRD_ID in
(select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where WRD.DICT_ID =
DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from KB_DICT where upper
(DICT) = upper (:rijec2)) or upper (WRD.WORD) = upper (:rijec2)))
);

```

```

:rijec1 = jezik
:rijec2 = govori
:rijec3 = engleski
:rijec4 = dobro
:quest1 = što
:quest2 = koji
:quest3 = kako

```

Tablica 41 prikazuje rezultat upita na 13. pitanje.

Tablica 41. Rezultat SQL upita za 13. pitanje

	DAT_ZAP
1	15.03.2009.

#### 14. Koji je datum zaposlenja djelatnika koji dobro govori njemački jezik

```

:rijec1 = jezik
:rijec2 = govori
:rijec3 = njemački
:rijec4 = dobro
:quest1 = što
:quest2 = koji
:quest3 = kako

```

Tablica 42 prikazuje rezultat upita na 14. pitanje.

Tablica 42. Rezultat SQL upita za 14. pitanje

	DAT_ZAP
1	01.09.2009.

#### 15. Koju ulogu na projektu LUMENS ima djelatnik koji posjeduje vozačku dozvolu B kategorije?

*posjeduje – što – dozvolu – kakvu – vozačku – što – kategorije – koje – B*

```

select DPR.ULOGA
from NOK_FNOK FNOK, NOK_PARAGRAPH PRG, ESTIMUS.PMG_DJELATNIK DJL,
ESTIMUS.PMG_DJL_PROJEKT DPR, ESTIMUS.PMG_PROJEKT PRJ
where FNOK.PRG = PRG.PRG
and PRG.ID_DJL_ESTIMUS = DJL.ID
and DPR.ID_DJL = DJL.ID
and DPR.ID_PRJ = PRJ.ID
and PRJ.NAZIV = :projekt

```

```

    and KB_NOD_WRD_ID      in
      (select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where WRD.DICT_ID =
DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from KB_DICT where upper
(DICT) = upper (:rijec1)) or upper (WRD.WORD) = upper (:rijec1)))
    and KB_NOD_QST_ID = (select QUEST_ID from KB_QUESTION where UPPER (QUEST) =
UPPER (:quest1))
    and KB_PAR_WRD_ID in
      (select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where WRD.DICT_ID =
DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from KB_DICT where upper
(DICT) = upper (:rijec2)) or upper (WRD.WORD) = upper (:rijec2)))
    and (FNOK.PRG, ORD, NODE_ID) in
      (
        select PRG, ORD, PARENT_ID
        from NOK_FNOK
        where KB_NOD_WRD_ID in
          (select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where WRD.DICT_ID =
DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from KB_DICT where upper
(DICT) = upper (:rijec3)) or upper (WRD.WORD) = upper (:rijec3)))
          and KB_NOD_QST_ID = (select QUEST_ID from KB_QUESTION where UPPER (QUEST) =
UPPER (:quest2))
          and KB_PAR_WRD_ID in
            (select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where WRD.DICT_ID =
DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from KB_DICT where upper
(DICT) = upper (:rijec1)) or upper (WRD.WORD) = upper (:rijec1)))
            and (PRG, ORD, NODE_ID) in
              (
                select PRG, ORD, PARENT_ID
                from NOK_FNOK
                where KB_NOD_WRD_ID in
                  (select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where WRD.DICT_ID
= DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from KB_DICT where upper
(DICT) = upper (:rijec4)) or upper (WRD.WORD) = upper (:rijec4)))
                  and KB_NOD_QST_ID = (select QUEST_ID from KB_QUESTION where UPPER
(QUEST) = UPPER (:quest3))
                  and KB_PAR_WRD_ID in
                    (select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where WRD.DICT_ID
= DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from KB_DICT where upper
(DICT) = upper (:rijec3)) or upper (WRD.WORD) = upper (:rijec3)))
                    and (PRG, ORD, NODE_ID) in
                      (
                        select PRG, ORD, PARENT_ID
                        from NOK_FNOK
                        where KB_NOD_WRD_ID in
                          (select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where
WRD.DICT_ID = DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from KB_DICT
where upper (DICT) = upper (:rijec5)) or upper (WRD.WORD) = upper(:rijec5)))
                          and KB_NOD_QST_ID = (select QUEST_ID from KB_QUESTION where UPPER
(QUEST) = UPPER (:quest4))
                          and KB_PAR_WRD_ID in
                            (select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where
WRD.DICT_ID = DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from KB_DICT
where upper (DICT) = upper (:rijec4)) or upper (WRD.WORD) = upper (:rijec4)))
                            )
                        )
                      )
                )
              )
            )
          )
        )
      )
    );

```

:rijec1 = dozvolu  
 :rijec2 = posjeduje  
 :rijec3 = vozačku  
 :rijec4 = kategorije  
 :rijec5 = B  
 :quest1 = što  
 :quest2 = kakvu  
 :quest3 = koje  
 :quest4 = koje  
 :projekt = LUMENS

Tablica 43 prikazuje rezultat upita na 15. pitanje.

Tablica 43. Rezultat SQL upita za 15. pitanje

	ULOGA
1	Član projektnog tima

### 16. Koju ulogu na projektu SMARTY ima djelatnik koji ne posjeduje vozačku dozvolu?

*ne posjeduje – što – dozvolu – kakvu – vozačku*

```
select DPR.ULOGA
from NOK_FNOK FNOK, NOK_PARAGRAPH PRG, ESTIMUS.PMG_DJELATNIK DJL,
     ESTIMUS.PMG_DJL_PROJEKT DPR, ESTIMUS.PMG_PROJEKT PRJ
where FNOK.PRG          = PRG.PRG
     and PRG.ID_DJL_ESTIMUS = DJL.ID
     and DPR.ID_DJL        = DJL.ID
     and DPR.ID_PRJ        = PRJ.ID
     and PRJ.NAZIV         = :projekt
     and KB_NOD_WRD_ID     in
     (select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where WRD.DICT_ID =
      DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from KB_DICT where upper
      (DICT) = upper (:rijec1)) or upper (WRD.WORD) = upper (:rijec1)))
     and KB_NOD_QST_ID = (select QUEST_ID from KB_QUESTION where UPPER (QUEST) =
      UPPER (:quest1))
     and KB_PAR_WRD_ID in
     (select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where WRD.DICT_ID =
      DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from KB_DICT where upper
      (DICT) = upper (:rijec2)) or upper (WRD.WORD) = upper (:rijec2)))
     and (FNOK.PRG, ORD, NODE_ID) in
     (
       select PRG, ORD, PARENT_ID
       from NOK_FNOK
       where KB_NOD_WRD_ID in
       (select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where WRD.DICT_ID =
        DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from KB_DICT where upper
        (DICT) = upper (:rijec3)) or upper (WRD.WORD) = upper (:rijec3)))
        and KB_NOD_QST_ID = (select QUEST_ID from KB_QUESTION where UPPER (QUEST) =
        UPPER (:quest2))
        and KB_PAR_WRD_ID in
        (select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where WRD.DICT_ID =
        DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from KB_DICT where upper
        (DICT) = upper (:rijec1)) or upper (WRD.WORD) = upper (:rijec1)))
```

);

:rijec1 = dozvolu  
:rijec2 = ne posjeduje  
:rijec3 = vozačku  
:quest1 = što  
:quest2 = kakvu  
:projekt = SMARTY

Tablica 44 prikazuje rezultat upita na 16. pitanje.

Tablica 44. Rezultat SQL upita za 16. pitanje

	ULOGA
1	Član projektnog tima

### 17. Na kojem radnom mjestu radi djelatnik koji u slobodno vrijeme vozi automobilske utrke?

*vozi – što – utrke – kakve – automobilske  
– kada – vrijeme – kakvo – slobodno  
– kada- u*

```
select DJL.RADNO_MJESTO
from NOK_FNOK FNOK, NOK_PARAGRAPH PRG, ESTIMUS.PMG_DJELATNIK DJL
where FNOK.PRG = PRG.PRG
and PRG.ID_DJL_ESTIMUS = DJL.ID
and FNOK.KB_NOD_WRD_ID in
(select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where WRD.DICT_ID =
DCT.DICT_ID(+) and (DCT.DICT_PAR = ( select DICT_PAR from KB_DICT where upper
(DICT) = upper (:rijec1) ) or upper (WRD.WORD) = upper (:rijec1)))
and FNOK.KB_NOD_QST_ID in ( select QUEST_ID from KB_QUESTION where UPPER
(QUEST) = UPPER (:quest1))
and FNOK.KB_PAR_WRD_ID in (select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD
where WRD.DICT_ID = DCT.DICT_ID(+) and (DCT.DICT_PAR = ( select DICT_PAR
from KB_DICT where upper (DICT) = upper (:rijec2) ) or upper (WRD.WORD) =
upper (:rijec2)))
and (FNOK.PRG, FNOK.ORD, FNOK.NODE_ID) in ( select PRG, ORD, PARENT_ID from
NOK_FNOK where KB_NOD_WRD_ID in
( select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where WRD.DICT_ID =
DCT.DICT_ID(+) and (DCT.DICT_PAR = ( select DICT_PAR from KB_DICT where upper
(DICT) = upper (:rijec3)) or upper (WRD.WORD) = upper (:rijec3)))
and KB_NOD_QST_ID in ( select QUEST_ID from KB_QUESTION where UPPER (QUEST)
= UPPER (:quest2))
and KB_PAR_WRD_ID in ( select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD
where WRD.DICT_ID = DCT.DICT_ID(+) and (DCT.DICT_PAR = ( select DICT_PAR from
KB_DICT where upper (DICT) = upper (:rijec1) ) or upper (WRD.WORD) = upper
(:rijec1)))
)
and (FNOK.PRG, FNOK.ORD, FNOK.PARENT_ID) in ( select PRG, ORD, PARENT_ID
from NOK_FNOK where KB_NOD_WRD_ID in
```



```

( select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where WRD.DICT_ID =
DCT.DICT_ID(+) and (DCT.DICT_PAR = ( select DICT_PAR from KB_DICT where upper
(DICT) = upper (:rijec4)) or upper (WRD.WORD) = upper (:rijec4)))
and KB_NOD_QST_ID in (select QUEST_ID from KB_QUESTION where UPPER (QUEST)
= UPPER (:quest3))
and KB_PAR_WRD_ID in
( select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where WRD.DICT_ID =
DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from KB_DICT where upper
(DICT) = upper (:rijec2)) or upper (WRD.WORD) = upper (:rijec2)))
and (PRG, ORD, NODE_ID) in (select PRG, ORD, PARENT_ID from NOK_FNOK
where KB_NOD_WRD_ID in
(select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where WRD.DICT_ID =
DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from KB_DICT where upper
(DICT) = upper (:rijec6)) or upper (WRD.WORD) = upper (:rijec6)) )
and KB_NOD_QST_ID = ( select QUEST_ID from KB_QUESTION where UPPER
(QUEST) = UPPER (:quest5))
and KB_PAR_WRD_ID in ( select WRD.WORD_ID from KB_DICT DCT, KB_WORD
WRD where WRD.DICT_ID = DCT.DICT_ID(+) and (DCT.DICT_PAR = ( select DICT_PAR
from KB_DICT where upper (DICT) = upper (:rijec4)) or upper (WRD.WORD) = upper
(:rijec4)))
)
and (PRG, ORD, NODE_ID) in (select PRG, ORD, PARENT_ID from NOK_FNOK
where KB_NOD_WRD_ID in (select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where
WRD.DICT_ID = DCT.DICT_ID(+) and (DCT.DICT_PAR = ( select DICT_PAR from
KB_DICT where upper (DICT) = upper (:rijec5)) or upper (WRD.WORD) = upper
(:rijec5)))
and KB_NOD_QST_ID in (select QUEST_ID from KB_QUESTION where UPPER
(QUEST) = UPPER (:quest4))
and KB_PAR_WRD_ID in (select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where
WRD.DICT_ID = DCT.DICT_ID(+) and (DCT.DICT_PAR = ( select DICT_PAR from
KB_DICT where upper (DICT) = upper (:rijec4)) or upper (WRD.WORD) = upper
(:rijec4)))
)
);

```

```

:rijec1 = utrke
:rijec2 = vozi
:rijec3 = automobilske
:rijec4 = vrijeme
:rijec5 = slobodno
:rijec6 = u
:quest1 = što
:quest2 = kakve
:quest3 = kada
:quest4 = kakvo
:quest5 = kada

```

Tablica 45 prikazuje rezultat upita na 17. pitanje.

Tablica 45. Rezultat SQL upita za 17. pitanje

	<b>RADNO_MJESTO</b>
1	Programer

## 18. Na kojem radnom mjestu radi djelatnik koji u slobodno vrijeme igra odbojku?

*igra – što – odbojku*

*- kada - vrijeme - kakvo - slobodno*

*- kada - u*

```
select DJL.RADNO_MJESTO
from NOK_FNOK FNOK, NOK_PARAGRAPH PRG, ESTIMUS.PMG_DJELATNIK DJL
where FNOK.PRG = PRG.PRG
  and PRG.ID_DJL_ESTIMUS = DJL.ID
  and FNOK.KB_NOD_WRD_ID in
    ( select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where WRD.DICT_ID =
      DCT.DICT_ID(+) and (DCT.DICT_PAR = ( select DICT_PAR from KB_DICT where upper
      (DICT) = upper (:rijec1)) or upper (WRD.WORD) = upper (:rijec1)))
  and FNOK.KB_NOD_QST_ID in ( select QUEST_ID from KB_QUESTION where UPPER
  (QUEST) = UPPER (:quest1))
  and FNOK.KB_PAR_WRD_ID in ( select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD
  where WRD.DICT_ID = DCT.DICT_ID(+) and (DCT.DICT_PAR = ( select DICT_PAR
  from KB_DICT where upper (DICT) = upper (:rijec2)) or upper (WRD.WORD) = upper
  (:rijec2)))
  and (FNOK.PRG, FNOK.ORD, FNOK.PARENT_ID) in ( select PRG, ORD, PARENT_ID
  from NOK_FNOK where KB_NOD_WRD_ID in
  ( select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where WRD.DICT_ID =
  DCT.DICT_ID(+) and (DCT.DICT_PAR = ( select DICT_PAR from KB_DICT where upper
  (DICT) = upper (:rijec4)) or upper (WRD.WORD) = upper (:rijec4)))
  and KB_NOD_QST_ID in (select QUEST_ID from KB_QUESTION where UPPER (QUEST) =
  UPPER (:quest3))
  and KB_PAR_WRD_ID in (select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where
  WRD.DICT_ID = DCT.DICT_ID(+) and (DCT.DICT_PAR = ( select DICT_PAR from
  KB_DICT where upper (DICT) = upper (:rijec2) ) or upper (WRD.WORD) = upper
  (:rijec2)))
  and (PRG, ORD, NODE_ID) in ( select PRG, ORD, PARENT_ID from NOK_FNOK where
  KB_NOD_WRD_ID in ( select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where
  WRD.DICT_ID = DCT.DICT_ID(+) and (DCT.DICT_PAR = ( select DICT_PAR from
  KB_DICT where upper (DICT) = upper (:rijec6)) or upper (WRD.WORD) = upper
  (:rijec6)))
  and KB_NOD_QST_ID = ( select QUEST_ID from KB_QUESTION where UPPER (QUEST) =
  UPPER (:quest5))
  and KB_PAR_WRD_ID in (select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where
  WRD.DICT_ID = DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from
  KB_DICT where upper (DICT) = upper (:rijec4)) or upper (WRD.WORD) = upper
  (:rijec4)))
  )
  and (PRG, ORD, NODE_ID) in (select PRG, ORD, PARENT_ID from NOK_FNOK where
  KB_NOD_WRD_ID in (select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where
  WRD.DICT_ID = DCT.DICT_ID(+) and (DCT.DICT_PAR = ( select DICT_PAR from
  KB_DICT where upper (DICT) = upper (:rijec5)) or upper (WRD.WORD) = upper
  (:rijec5)))
  and KB_NOD_QST_ID in (select QUEST_ID from KB_QUESTION where UPPER (QUEST) =
  UPPER (:quest4))
  and KB_PAR_WRD_ID in (select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where
  WRD.DICT_ID = DCT.DICT_ID(+) and (DCT.DICT_PAR = ( select DICT_PAR from
  KB_DICT where upper (DICT) = upper (:rijec4)) or upper (WRD.WORD) = upper
  (:rijec4)))
  )
  )
```

);

```
:rijec1 = odbojku
:rijec2 = igra
:rijec4 = vrijeme
:rijec5 = slobodno
:rijec6 = u
:quest1 = što
:quest3 = kada
:quest4 = kakvo
:quest5 = kada
```

Tablica 46 prikazuje rezultat upita na 18. pitanje.

Tablica 46. Rezultat SQL upita za 18. pitanje

	<b>RADNO_MJESTO</b>
1	projektant

### 19. Što je bio djelatnik koji od 18.09.2014. radi na projektu?

*je – što – bio – što - X*

```
select NODE
from NOK_FNOK NOK
  connect by prior PRG = PRG
  and prior ORD       = ORD
  and prior NODE_ID   = PARENT_ID
start with
(
  PRG, ORD, NODE_ID
)
in
(
  select FNOK.PRG, FNOK.ORD, FNOK.NODE_ID
  from NOK_FNOK FNOK, NOK_PARAGRAPH PRG, ESTIMUS.PMG_DJELATNIK DJL,
  ESTIMUS.PMG_DJL_PROJEKT DPR
  where FNOK.PRG          = PRG.PRG
  and PRG.ID_DJL_ESTIMUS = DJL.ID
  and DPR.ID_DJL         = DJL.ID
  and DPR.D_OD           = :datum
  and FNOK.KB_NOD_QST_ID in (select QUEST_ID from KB_QUESTION where UPPER
(QUEST) = UPPER (:quest1))
  and FNOK.KB_PAR_WRD_ID in
  (select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where WRD.DICT_ID =
DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from KB_DICT where upper
(DICT) = upper (:rijec1)) or upper (WRD.WORD) = upper (:rijec1)))
  and (FNOK.PRG, FNOK.ORD, FNOK.PARENT_ID) in
  (
    select PRG, ORD, NODE_ID
    from NOK_FNOK
    where KB_NOD_WRD_ID in
```

```

        (select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where WRD.DICT_ID
= DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from KB_DICT where upper
(DICT) = upper (:rijec1)) or upper (WRD.WORD) = upper (:rijec1)))
        and KB_NOD_QST_ID in (select QUEST_ID from KB_QUESTION where UPPER
(QUEST) = UPPER (:quest2))
        and KB_PAR_WRD_ID in
        (select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where WRD.DICT_ID
= DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from KB_DICT where upper
(DICT) = upper (:rijec2)) or upper (WRD.WORD) = upper (:rijec2)))
    )
);

:rijec1 = bio
:rijec2 = je
:quest1 = što
:quest2 = što
:datum = 18.09.2014

```

Prvi dio upita je hijerarhijski upit koji će pronaći sve podređene povezane čvorove pronađenog čvora.

Tablica 47 prikazuje rezultat upita na 19. pitanje.

Tablica 47. Rezultat SQL upita za 19. pitanje

	<b>NODE</b>
1	nastavnik
2	matematike
3	OŠ Rijeka
4	u

## **20. Koju ulogu na projektu čiji je tip projekta "razvojni" ima djelatnik koji slabo govori njemački jezik?**

*govori – što – jezik – koji – njemački*

*– kako – slabo*

```

select DPR.ULOGA
from NOK_FNOK FNOK, NOK_PARAGRAPH PRG, ESTIMUS.PMG_DJELATNIK DJL,
ESTIMUS.PMG_DJL_PROJEKT DPR, ESTIMUS.PMG_PROJEKT PRJ
where FNOK.PRG = PRG.PRG
and PRG.ID_DJL_ESTIMUS = DJL.ID
and DPR.ID_DJL = DJL.ID
and DPR.ID_PRJ = PRJ.ID
and PRJ.TIP = :razvojni
and KB_NOD_WRD_ID in
(select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where WRD.DICT_ID =
DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from KB_DICT where upper
(DICT) = upper (:rijec1)) or upper (WRD.WORD) = upper (:rijec1)))
and KB_NOD_QST_ID = (select QUEST_ID from KB_QUESTION where UPPER (QUEST) =
UPPER (:quest1))

```

```

and KB_PAR_WRD_ID in
(select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where WRD.DICT_ID =
DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from KB_DICT where upper
(DICT) = upper (:rijec2)) or upper (WRD.WORD) = upper (:rijec2)))
and (FNOK.PRG, ORD, NODE_ID) in
(
select PRG, ORD, PARENT_ID
from NOK_FNOK
where KB_NOD_WRD_ID in
(select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where WRD.DICT_ID =
DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from KB_DICT where upper
(DICT) = upper (:rijec3)) or upper (WRD.WORD) = upper (:rijec3)))
and KB_NOD_QST_ID = (select QUEST_ID from KB_QUESTION where UPPER (QUEST) =
UPPER (:quest2))
and KB_PAR_WRD_ID in
(select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where WRD.DICT_ID =
DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from KB_DICT where upper
(DICT) = upper (:rijec1)) or upper (WRD.WORD) = upper (:rijec1)))
)
and (FNOK.PRG, ORD, PARENT_ID) in
(
select PRG, ORD, PARENT_ID
from NOK_FNOK
where KB_NOD_WRD_ID in
(select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where WRD.DICT_ID =
DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from KB_DICT where upper
(DICT) = upper (:rijec4)) or upper (WRD.WORD) = upper (:rijec4)))
and KB_NOD_QST_ID = (select QUEST_ID from KB_QUESTION where UPPER (QUEST) =
UPPER (:quest3))
and KB_PAR_WRD_ID in
(select WRD.WORD_ID from KB_DICT DCT, KB_WORD WRD where WRD.DICT_ID =
DCT.DICT_ID(+) and (DCT.DICT_PAR = (select DICT_PAR from KB_DICT where upper
(DICT) = upper (:rijec2)) or upper (WRD.WORD) = upper (:rijec2)))
);

```

```

:rijec1 = jezik
:rijec2 = govori
:rijec3 = njemački
:rijec4 = slabo
:quest1 = što
:quest2 = koji
:quest3 = kako
:razvojni = razvojni

```

Tablica 48 prikazuje rezultat upita na 20. pitanje.

**Tablica 48. Rezultat SQL upita za 20. pitanje**

ULOGA	
1	Član projektnog tima

## Životopis

Martina Ašenbrener Katić rođena je 04. listopada 1985 godine u Rijeci. Nakon završetka osnovnoškolskog i srednjoškolskog obrazovanja (Srednja škola Delnice, Prirodoslovno – matematička gimnazija) upisuje Filozofski fakultet u Rijeci, smjer matematika i informatika. Diplomirala je u svibnju 2009. godine obranom diplomskog rada pod nazivom "Znanstvene metode na polju informacijskih sustava" (mentor: prof. dr. sc. Mile Pavlić). Od prosinca 2009. godine zaposlena je na Odjelu za informatiku Sveučilišta u Rijeci, na radnom mjestu znanstvenog novaka u suradničkom zvanju asistenta. Aktivno priprema i izvodi vježbe iz kolegija na preddiplomskom studiju (Osnove informatike 2, Informacijski sustavi, Modeliranje podataka, Uvod u programsko inženjerstvo) te na diplomskom studiju (Programsko inženjerstvo).

Poslijediplomski doktorski studij informacijskih znanosti upisala je 2009. godine na Filozofskom fakultetu u Zagrebu te ga nastavlja 2012. godine na Odjelu za informatiku Sveučilišta u Rijeci.

Sudjelovala je na nekoliko nacionalnih znanstvenih te međunarodnih stručnih projekata. Sudjelovala je na brojnim međunarodnim znanstvenim i stručnim konferencijama te objavila više znanstvenih i stručnih radova (samostalno ili u koautorstvu).

Znanstveni interesi obuhvaćaju područje metodologije razvoja informacijskih sustava, informacijske sustavi, softversko inženjerstvo, modeliranje podataka, prikaz znanja.

Ima iskustva u pisanju i prijavi nacionalnih (UNIRI potpora, HZZZ, MZOS, BICRO) znanstvenih i stručnih projekata.

U nastavku slijedi popis objavljenih radova.

1. Ašenbrener Katić, M., Pavlić, M., Čandrlić, S., (2015.) The Representation of Database Content and Structure Using the NOK Method, *Procedia Engineering*. 100, 1075-1081.
2. Rauker Koch, M., Pavlić, M., Ašenbrener Katić, M., (2015.) Homonyms and Synonyms in NOK Method, *Procedia Engineering*. 100, 1055-1061.
3. Čandrlić, S., Pavlić, M., Ašenbrener, M., (2013.) Code merging analysis for different procedure types., *International Journal Of Systems Applications, Engineering & Development*. 7, 239-246.

4. Ašenbrener, M., Šimunović, D., Dovedan Han, Z. (2013.) Model informacijskog sustava planiranja TV emisija., *Media, culture and public relations*. 4, 1, 61-69.
5. Ašenbrener Katić, M., Čandrlić, S., Pavlić, M. (2017.) Comparison of two versions of formalization method for text expressed knowledge. *Beyond Databases, Architectures and Structures (BDAS) 2017, CCIS (2017)*, prihvaćen za objavljivanje
6. Ašenbrener Katić, M., Čandrlić, S., Holenko Dlab, M., (2016.) Introducing collaborative e-learning activities to the e-course "Information systems", *Proceedings of the 39th International Convention MIPRO 2016, Rijeka*, 917-922
7. Holenko Dlab, M., Ašenbrener Katić, M., Čandrlić, S. (2015.) Ensuring Formative Assessment in E-Course with Online Tests, *Proceedings of the 10th International Conference on Computer Science & Education. IEEE*, 322-327.
8. Čandrlić, S., Ašenbrener Katić, M., Holenko Dlab, M. (2014.) Online vs. Paper-Based Testing: A Comparison of Test Results, *Proceedings of the 37th International Convention MIPRO 2014*, 775-780.
9. Tomljanović, J., Pavlić, M., Ašenbrener Katić, M. (2014.) Intelligent Question – Answering Systems: Review of research, *Proceedings of the 37th International Convention MIPRO 2014*, 1478-1483.
10. Mrković, B., Ašenbrener, M. (2012.) The simple CMOS negative capacitance with improved frequency response, *Proceedings of the 35th international convention on information and communication technology, electronics and microelectronics / MEET / MIPRO 2012*, 93-96.
11. Mrković, B., Ašenbrener, M. (2011.) A 10-b Fully Differential CMOS SAR ADC for Wireless Sensor Networks, *Proceedings of Microelectronics, Electronics and Electronic Technology /MEET, Zagreb, DENONA*, 98-103.
12. Jakupović, A., Pavlić, M., Ašenbrener, M. (2010.) Measuring the Size and Contribution of ERP Solutions through Covered Business Sectors, *Proceedings of The International Multi-Conference on Complexity, Informatics and Cybernetics (IMCIC 2010)*, Orlando, Florida, USA, 397-402.
13. Pavlić, M.; Dovedan Han Z., Jakupović, A., Čandrlić, S., Ašenbrener Katić, M., Tomljanović, J., Rauker Koch, M (2015.) Sustav za odgovaranje na pitanja prirodnim jezikom, *Razvoj poslovnih i informatičkih sustava CASE 27, Rijeka: CASE d.o.o.* 5-16

14. Pavlić, M., Ašenbrener, M., Jakupović, A., Meštrović, A., Čandrlić, S., Ivašić-Kos, M. (2013.) Inteligentni informacijski sustavi, Razvoj poslovnih i informatičkih sustava CASE 25, Rijeka: CASE d.o.o. Rijeka, 25-35.
15. Ašenbrener, M., Mrković, B. (2012.) Definicija efikasnog komunikacijskog protokola za bežične senzorske mreže, Proceedings of the 35th international convention on information and communication technology, electronics and microelectronics / CTS / MIPRO 2012, Rijeka, 1088-1090.
16. Ašenbrener, M., Šimunović, D., Dovedan Han, Z. (2012.) Model informacijskog sustava planiranja TV emisija, Book of Manuscripts, Society and Technology 2012 - Dr. Juraj Plenković. Zagreb, 117-125
17. Ašenbrener, M., Pavlić, M., Mrković, B. (2011.) Sustav za lociranje i zaštitu plovila u marini, Proceedings of the 34th international convention on information and communication technology, electronics and microelectronics, Proceedings Vol. III, CTS & CIS, MIPRO 2011. Zagreb, 229-232.
18. Pavlić, M., Ašenbrener, M., Čandrlić, S., Francetić, S., Pošćić, P. (2009.) Znanstveni pristup u području informacijskih sustava, CASE 21.