

# Proširivanje konceptualnog razvojnog okvira NOK-a za transformaciju složenih rečenica i pitanja u zapise relacijske baze podataka

---

**Rauker Koch, Marina**

**Doctoral thesis / Disertacija**

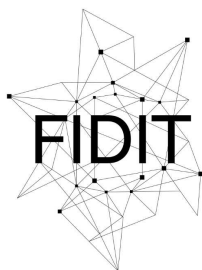
**2024**

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

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:195:807201>

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

*Download date / Datum preuzimanja:* **2025-02-01**



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  
FAKULTET INFORMATIKE I DIGITALNIH TEHNOLOGIJA

Marina Rauker Koch

**PROŠIRIVANJE KONCEPTUALNOG RAZVOJNOG  
OKVIRA NOK-A ZA TRANSFORMACIJU  
SLOŽENIH REČENICA I PITANJA U ZAPISE  
RELACIJSKE BAZE PODATAKA**

DOKTORSKI RAD

Rijeka, 2024.

SVEUČILIŠTE U RIJECI  
FAKULTET INFORMATIKE I DIGITALNIH TEHNOLOGIJA

Marina Rauker Koch

**PROŠIRIVANJE KONCEPTUALNOG RAZVOJNOG  
OKVIRA NOK-A ZA TRANSFORMACIJU  
SLOŽENIH REČENICA I PITANJA U ZAPISE  
RELACIJSKE BAZE PODATAKA**

DOKTORSKI RAD

Mentorica: doc. dr. sc. Martina Ašenbrener Katić, Sveučilište u  
Rijeci, Fakultet informatike i digitalnih tehnologija

Rijeka, 2024.

UNIVERSITY OF RIJEKA  
FACULTY OF INFORMATICS AND DIGITAL TECHNOLOGIES

Marina Rauker Koch

**EXTENSION OF THE NOK CONCEPTUAL  
DEVELOPMENT FRAMEWORK FOR  
TRANSFORMATION OF COMPLEX SENTENCES  
AND QUESTIONS IN RELATIONAL DATABASE  
RECORDS**

DOCTORAL THESIS

Rijeka, 2024

Mentorica: doc. dr. sc. Martina Ašenbrener Katić, Sveučilište u Rijeci, Fakultet informatike i digitalnih tehnologija

Doktorski rad obranjen je dana 11. prosinca 2024. na Fakultetu informatike i digitalnih tehnologija Sveučilišta u Rijeci,

Pred povjerenstvom u sastavu:

1. prof. dr. sc. Božidar Kovačić, predsjednik
2. prof. dr. sc. Patrizia Pošćić, članica
3. izv. prof. dr. sc. Dijana Oreški, članica

## Zahvale

Na kraju ovog dugog i izazovnog putovanja želim izraziti svoju iskrenu zahvalnost svima koji su me podržavali i ohrabivali tijekom izrade ovog doktorata.

Najprije želim zahvaliti svojim mentorima, mentoru prof. dr. sc. Mili Pavliću s kojim sam započela ovo istraživanje i mentorici doc. dr. sc. Martini Ašenbrener Katić s kojom sam ga nastavila i završila. Hvala vam na savjetima, stručnosti, razumijevanju, podršci i motivaciji koja mi je posebno značila u najizazovnijim trenutcima.

Sanjinu Maržiću hvala na pomoći pri izradi prototipa sustava uključenih u ovo istraživanje.

Hvala ekipi s posla koja je uvijek bila tu kad je bila potrebna podrška i razumijevanje. Hvala od srca mojim „curama iz knjižnice“ koje su bile uz mene kroz sve uspone i padove, a posebno Nataši koja je strpljivo rješavala sve moje jezične nedoumice. Hvala vam na smijehu i podršci. Bez vas bi sve ovo bilo znatno teže.

Na kraju zahvaljujem svojoj obitelji na pruženom strpljenju i osobnoj podršci tijekom mog rada, posebno kada se činilo da je cilj nedostižan i dalek.

## Sažetak

Konceptualni okvir Node of Knowledge (NOK) opisuje način modeliranja rečenica prirodnog jezika u formalizirani FNOK (*Formalized Node of Knowledge*) zapis koje je u ovom obliku moguće zapisati u relacijsku bazu podataka. Osim za rečenice, opisuje i način modeliranja pitanja, odnosno upitnih rečenica prirodnog jezika, u formalizirani QFNOK (*Question Formalized Node of Knowledge*) zapis koje je također moguće zapisati u relacijsku bazu podataka. Iz tako definirane i kreirane baze podataka moguće je iz rečenica prirodnog jezika dobiti odgovore na pitanja. U dosadašnjim istraživanjima izrađen je sustav koji je omogućio zapisivanje jednostavnih rečenica i pitanja prirodnog jezika u relacijsku bazu podataka NOK-a (BPNOK) te je testiran i primijenjen na jednostavne rečenice hrvatskog i engleskog jezika.

Cilj ovog rada i istraživanja je proširiti postojeći sustav zapisivanja jednostavnih rečenica prirodnog jezika u relacijsku bazu podataka BPNOK s mogućnošću zapisivanja složenih rečenica engleskog jezika. Promatrane su složene rečenice koje sadrže veznike i/ili se sastoje od dvije radnje, odnosno dva procesna čvora, ako promatramo sa stajališta konceptualnog okvira NOK. S ovim ciljem, u radu su definirana pravila transformacije veznika u formalizirane zapise rečenica FNOK i pitanja QFNOK. Na razini rječnika sustav je proširen implementacijom rješenja problema sinonima. Ove izmjene, u formaliziranim zapisima rečenica i pitanja te implementacija sinonima, generirale su i proširenja u modelu sustava, shemi relacijske baze podataka, algoritama za transformaciju formaliziranih FNOK i QFNOK zapisa u relacijsku bazu BPNOK i algoritama traženja odgovora. Provjera valjanosti i funkcionalnosti svih navedenih proširenja i njihove implementacije provedena je razvojem sustava i provjerom točnosti odgovora koje taj sustav daje na postavljena pitanja.

**Ključne riječi:** relacijska baza podataka, predstavljanje znanja, veznici, sinonimi, metoda „Node of Knowledge“

## **Abstract**

The Node of Knowledge (NOK) conceptual framework describes a way of modeling natural language sentences into The Formalized Node of Knowledge (FNOK) record which in this form, can be used in a relational database. In addition to sentences, it also describes the method of modeling questions, i.e. interrogative sentences of natural language, into a formalized QFNOK record (Question Formalized Node of Knowledge). From the database defined and created in this way, it is possible to get answers to questions from natural language sentences. In previous research, a system was developed that enabled the recording of simple sentences and questions in natural language into the relational database of the Node of Knowledge (DBNOK), and it has been tested and applied to simple sentences in Croatian and English languages.

The aim of this paper and research is to extend the existing system of writing simple natural language sentences to the relational database DBNOK with the possibility of writing complex English sentences. Complex sentences containing the conjunctions and/or and consisting of two actions or two process nodes, if viewed from the standpoint of the NOK conceptual framework are observed. The rules for transforming conjunctions into formalized FNOK and QFNOK records are defined in the paper with this aim. At the dictionary level, the system is extended by implementation of a solution to the problem of synonyms. These changes, in the formalized records of sentences and questions as well as the implementation of synonyms, also generated extension in the system model, the relational database schema, algorithms for transforming formalized FNOK and QFNOK records into the relational database DBNOK and answer – seeking algorithms. The verification of the validity and functionality of all mentioned extension and their implementation was carried out by developing a system and verifying the validity of the answers that the system gives to the questions raised.

**Key words:** relational database, knowledge presentation, conjunctions, synonyms, „Node of Knowledge“ method



## Sadržaj

1.	Uvod.....	1
1.1.	Motivacija .....	2
1.2.	Ciljevi, hipoteze i znanstveni doprinosi .....	2
1.3.	Struktura rada.....	4
2.	Pregled područja.....	6
2.1.	Predstavljanje znanja.....	7
2.1.1.	Obrada prirodnog jezika .....	8
2.1.2.	Rječnici prirodnih jezika.....	9
2.2.	QA sustavi .....	10
2.2.1.	Large Language Models (LLM).....	11
2.2.2.	Knowledge Graph .....	16
3.	Konceptualni okvir Nodes of Knowledge (NOK).....	18
3.1.	Primjena NOK-a na modeliranje složenih rečenica .....	19
3.2.	Automatsko prevođenje uporabom PSA – FNOK aplikacije.....	26
3.2.1.	Atributi riječi .....	27
3.2.2.	Primjena PSA – FNOK aplikacije.....	29
4.	Pravila transformacije složenih rečenica prirodnog jezika u Formalized Node of Knowledge (FNOK).....	37
4.1.	Pravila prevođenja prema vrsti riječi.....	38
4.2.	Veznici.....	39
4.2.1.	Veznici koji povezuju dijelove rečenice .....	40
4.2.2.	Veznici koji povezuju dvije rečenice .....	42
4.3.	Veznici u QFNOK-u.....	43
5.	Meta modeli proširivanja rječnika u NOK metodi .....	45
5.1.	Meta model hijerarhije glagolskih vremena.....	45
5.2.	Meta modeli sinonima i homonima .....	50
6.	Proširivanje sustava temeljenog na NOK metodi .....	54
6.1.	Proširivanje modela podataka BPNOK-a i relacijske sheme baze podataka ..	55
6.1.1.	Implementacija rješenja problema sinonima .....	58
6.1.2.	Proširivanje relacijske sheme baze podataka.....	59
6.2.	Algoritmi transformacije FNOK i QFNOK zapisa u BPNOK.....	59

6.2.1.	Proširivanje algoritma transformacije FNOK zapisa u BPNOK.....	61
6.2.2.	Proširivanje algoritma transformacije QFNOK zapisa u BPNOK.....	64
6.3.	Kreiranje odgovora .....	65
7.	Primjena sustava temeljenog na NOK metodi na složene rečenice engleskog jezika	71
7.1.	Primjena proširenog sustava temeljenog na NOK metodi .....	72
7.2.	Dokaz hipoteze H1.....	79
7.3.	Analiza rezultata .....	80
7.3.1.	Točan odgovor .....	81
7.3.2.	Gramatički neispravan odgovor .....	82
7.3.3.	Djelomično točan odgovor .....	85
7.3.4.	Nema odgovora .....	88
7.3.5.	Zaključci analize rezultata.....	90
7.4.	Dokaz hipoteze H2.....	91
7.5.	Testiranje implementacije sinonima .....	93
7.6.	Testiranje sustava na rečenicama i pitanjima hrvatskog jezika .....	98
8.	Zaključak .....	103
	Literatura .....	107
	Popis slika .....	115
	Popis tablica.....	117
	Popis grafikona .....	117
	Popis privitaka .....	118
	Privitak 1. PHP skripta <i>answer.php</i> .....	119
	Privitak 2. PHP skripta <i>insert.php</i> .....	127
	Privitak 3. PHP skripta <i>insertQ.php</i> .....	135
	Privitak 4. PHP skripta <i>functions.php</i> .....	143
	Privitak 5. Rečenice korištene u inicijalnom testiranju .....	165
	Privitak 6. Pitanja korištena u inicijalnom testiranju i pripadajući QFNOK-ovi .....	166
	Privitak 7. Testirane rečenice engleskog jezika i pripadajući FNOK-ovi .....	168
	Privitak 8. Testirana pitanja engleskog jezika i pripadajući QFNOK-ovi .....	173
	Privitak 9. Odgovori sustava na pitanja .....	182
	Životopis .....	191

## 1. Uvod

Dobivanje odgovora iz tekstualnog znanja iskazanog rečenicama prirodnog jezika na pitanja koja su također postavljena na prirodnom jeziku, predmet je mnogih istraživanja i razvoja različitih *Question Answering* (QA) sustava. QA sustavi koriste različite metode i algoritme kako bi analizirali i interpretirali tekstualne podatke, te na temelju njih generirali odgovore koji što preciznije i točnije odgovaraju na postavljena pitanja. Rješavanjem ovog problema bavi se i konceptualni okvir *Node of Knowledge* (NOK). Metoda *Node of Knowledge* (NOK), koja je sastavni dio konceptualnog okvira NOK za zapisivanje tekstualnog znanja koristi relacijsku bazu podataka. Relacijska baza podataka sastoji se od međusobno povezanih relacija (tablica) koje inicijalno nisu namijenjene za upisivanje tekstualnog znanja.

Analizom rečenica prirodnog jezika, riječi i njihova značenja te načina i redoslijeda na koji se riječi povezuju u rečenicu, metoda NOK definira pravila transformacije kojima se rečenice prilagođavaju na način da se riječi u njima hijerarhijski organiziraju što ih priprema za unos u relacijsku bazu podataka. Nad podacima u relacijskoj bazi podataka moguće je postavljati SQL upite kojima se pretražujući po relacijama, odnosno tablicama u bazi, može dobiti odgovor na dani upit. Kako NOK metoda priprema i omogućava upisivanje rečenica i pitanja na prirodnom jeziku u relacijsku bazu podataka, moguće je i te podatke propitivati odgovarajućim SQL upitima i dobiti tražene odgovore.

Ovo istraživanje nastavlja i proširuje istraživanje [1], kojim je opisan, izrađen i testiran sustav temeljen na NOK metodi, provedeno na:

- jednostavnim rečenicama hrvatskog jezika pod kojima se podrazumijevalo 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.
- i jednostavnim rečenicama engleskog jezika pod kojima se podrazumijevalo izjavne rečenice s jednim glagolom u *Simple present tense* ili *Present continuous tense*, subjektom, objektom i/ili priložnim oznakama.

Ovo istraživanje je provedeno na složenim rečenicama isključivo engleskog jezika. Za istraživanje su odabrane rečenice koje sadrže konačan skup odabranih veznika: veznike *and, or, but, because, if, when, that, unless, whenever, while, until* i *so*. Veznici u rečenicama prirodnog jezika mogu u rečenici prirodnog jezika imati različite uloge i to na sljedećim razinama:

- veznici spajaju subjekte,
- veznici spajaju predikate,
- veznici spajaju objekte,

- veznici spajaju dvije jednostavne rečenice od kojih svaka sadrži jedan glagol/predikat i subjekt i/ili ostale dijelove rečenice kao što je objekt i/ili priložne oznake.

Uporaba veznika u rečenicama ima značajan utjecaj na složenost rečenice, pogotovo u slučaju kada veznici spajaju dvije rečenice. Veznici tada povezuju dvije jednostavne rečenice u jednu složenu koja može biti zavisno ili nezavisno složena, što ovisi i o vezniku koji se koristi u rečenici. Detaljnije o vrstama veznika i njihovoj ulozi u rečenici opisano je u poglavlju 4.2.

## **1.1. Motivacija**

Za pohranjivanje rečenica prirodnog jezika u relacijsku bazu podataka potrebno je riješiti problem načina njihovog upisivanja u relacijsku bazu na način da se iz takve relacijske baze podataka upitima mogu dobiti odgovori na postavljena pitanja. Problem postaje složeniji što su složenije rečenice koje se žele zapisati i propitivati. I za rečenice prirodnog jezika i za pitanja prirodnog jezika kojima se propituju potrebno je definirati pravila kojima se definira njihova prilagodba, odnosno transformacija, u zapise koje je moguće upisati u relacijsku bazu podataka.

U dosadašnjim istraživanjima [1] definiran je model sustava koji omogućuje pohranu tekstualnog znanja u relacijsku bazu podataka bez gubitka semantike temeljen na konceptualnom okviru Node of Knowledge (NOK) i primijenjen na jednostavne rečenice prirodnog jezika. U ovom radu proširit će se postojeći model na odabrane složene rečenice prirodnog jezika koje sadrže veznike i/ili dvije radnje (glagola), a implementirat će se i problem rješavanja sinonima. Motivacija za proširivanje sustava temeljenog na NOK metodi je unaprijediti postojeći sustav na način da se proširi skup rečenica prirodnog jezika koje je moguće upisati u relacijsku bazu podataka i iz njih propitivanjem pitanjima prirodnog jezika dobiti odgovore.

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

U ovom istraživanju nastavlja se ranije istraživanje grupe autora vezano uz konceptualni okvir Node of Knowledge (NOK) i razvoj sustava za odgovaranje na pitanja. Cilj razvoja je i unaprjeđenje sustava za generiranje odgovora na pitanja. Za to je potrebno:

1. naći pravila transformacije složenih rečenica prirodnog jezika u formalizirani oblik (FNOK),
2. zapisivati rečenice u formaliziranom FNOK zapisu u relacijsku bazu podataka,

### 3. razviti sustav za odgovaranje na pitanja.

U ovom istraživanju planira se definirati pravila koja će obuhvatiti odabrane veznike i sinonime u engleskom jeziku, a kako bi se omogućila transformacija složenih rečenica u FNOK zapis. Provjera valjanosti postavljenih pravila transformacije provest će se razvojem sustava za generiranje odgovora na pitanja i provjerom točnosti odgovora koje taj sustav daje na postavljena pitanja.

U radu su postavljene dvije istraživačke hipoteze:

**H1: Primjenom konceptualnog razvojnog okvira NOK, određene vrste složenih rečenica engleskog jezika se mogu transformirati u FNOK i relacijski BPNOK zapis.**

Hipoteza će se smatrati potvrđenom ako se NOK metodom i definiranim pravilima složene rečenice engleskog jezika transformirane u FNOK zapis mogu upisati u relacijsku bazu podataka (BPNOK). Potvrđenost ove hipoteze opisana je u poglavlju 7.2.

**H2: Koristeći prirodni upitni jezik, relacijska BPNOK može pronaći i formulirati odgovore na postavljena pitanja.**

Hipoteza će se smatrati potvrđenom ako se NOK metodom i definiranim pravilima upitne rečenice engleskog jezika transformirane u QFNOK zapišu u relacijsku bazu podataka BPNOK i na temelju njih izvršenim propitivanjem dobiju odgovori na postavljena pitanja. Potvrđenost ove hipoteze opisana je u poglavlju 7.4.

Prilikom izrade ovog rada i proširivanjem sustava temeljenog na NOK metodi postignuti su sljedeći znanstveni doprinosi:

- Analiza prednosti i nedostataka različitih varijanti hijerarhijskog odnosa glavnih i pomoćnih glagola u FNOK zapisu.
- Proširenje pravila prevođenja složenih rečenica u FNOK zapis (kao što su povratne zamjenice, veznici, više procesnih čvorova).
- Proširenje pravila prevođenja pitanja u QFNOK zapis (veznici, više procesnih čvorova).
- Proširenje modela procesa sustava za transformaciju rečenica u relacijsku bazu podataka.
- Proširenje meta modela podataka sustava za transformaciju složenih rečenica engleskog jezika u različitim glagolskim vremenima u relacijsku bazu podataka.
- Poboljšanje algoritama traženja odgovora na postavljeno pitanje.
- Određivanje tipa veznika kao elementa traženja odgovora na pitanja.
- Implementacija sinonima u rječnik i u proces traženja odgovora na pitanja.
- Putem korisničkog sučelja jasnije prikazivanje uparivanja čvorova u pitanju i rečenici te preglednije iščitavanje odgovora ili razloga nedobivanja odgovora.

### 1.3. Struktura rada

U prvom poglavlju dan je uvod u temu i opisani su elementi ovog doktorskog rada. Opisana je motivacija i cilj istraživanja, te su iskazane hipoteze i znanstveni doprinosi kojima je pridonijelo ovo istraživanje.

Drugo poglavlje bavi se pregledom područja istraživanja te definiranjem pojmova povezanih s istraživanjem. Samim pregledom područja obuhvaćena su istraživanja iz područja predstavljanja znanja, obrade prirodnog jezika, rječnika prirodnog jezika, sustava pitanja i odgovora (*Question Answering - QA*), velikih jezičnih modela (*Large Language Models – LLM*), grafova znanja (*Knowledge Graphs*) i relacijskih baza podataka kao elemenata QA sustava. Pregledom područja se željelo i usporediti konceptualni okvir Node of Knowledge s aktualnim istraživanjima u području predstavljanja znanja.

U trećem poglavlju opisan je konceptualni okvir Node of Knowledge i njegove mogućnosti. Također, dan je pregled mogućnosti primjene NOK metode na modeliranje složenih rečenica kao priprema za proširivanje modela sustava NOK za rad sa složenim rečenicama. Opisana je postojeća PSA – FNOK aplikacija za automatsku transformaciju rečenica prirodnog jezika u formalizirane FNOK zapise i analizirana učinkovitost njene primjene na jednostavne i složene rečenice prirodnog jezika.

U četvrtom poglavlju analizirana su postojeća pravila transformacije rečenica prirodnog jezika u formalizirani FNOK zapis i pitanja prirodnog jezika u formalizirani QFNOK zapis, općenita i prema vrsti riječi, s ciljem utvrđivanja potrebe za njihovim izmjenama i proširenjima. Proširen je i izmijenjen temeljni algoritam transformacije rečenica zapisanih u prirodnom jeziku u formalizirani FNOK zapis i definirana su pravila za veznike.

U petom poglavlju opisani su meta modeli proširivanja rječnika u BPNOK relacijskoj bazi podataka i to meta model hijerarhije glagolskih vremena i meta modeli sinonima i homonima.

U šestom poglavlju opisano je proširivanje sustava temeljenog na NOK metodi. Opisano je proširivanje modela podataka relacijske baze podataka BPNOK i njezine relacijske sheme te implementacija rješenja problema sinonima kao dio proširenja rječnika. U ovom poglavlju opisano je i proširivanje algoritama transformacije formaliziranih FNOK i QFNOK zapisa u relacijsku bazu podataka te algoritama za kreiranje odgovora.

U sedmom poglavlju prikazani su rezultati testiranja uspješnosti primjene proširenog sustava temeljenog na NOK metodi na skupu odabranih složenih rečenica i pitanja engleskog jezika koje sadrže veznike na njihovo zapisivanje u relacijsku bazu podataka. Prikazani su i rezultati propitivanja zapisanih rečenica pitanjima prirodnog jezika, odnosno funkcionalnost proširenih algoritama traženja odgovora na pitanja. U

testiranje su uključene rečenice i pitanja prirodnog jezika koje u sebi sadrže sinonime te je i na njima provjerena točnost dobivenih odgovora. Sustav je testiran i na primjerima rečenica na hrvatskom jeziku kako bi se pokazala njegova primjenjivost neovisno o prirodnom jeziku na kojem su rečenice i pitanja kojima se propituju.

U osmom poglavlju doneseni su zaključci ovog istraživanja te planovi za buduća istraživanja i daljnju nadogradnju i proširenje sustava.

## 2. Pregled područja

Strojevi s ljudskim sposobnostima bili su i ostali predmet interesa, ne samo u znanstvenim krugovima, već i u umjetnosti. Pojam umjetna inteligencija (engl. Artificial Intelligence, AI) prvi je upotrijebio J. McCarthy kako bi prikupio sredstva za organizaciju konferencije u Dartmouthu 1956. [2] i od tada AI postaje istraživačka disciplina.

Postoji mnogo definicija umjetne inteligencije i niti jedna nije opće prihvaćena. One se mogu razvrstati u 4 skupine [3]:

### 1. razmišljati ljudski:

- „Automatizacija aktivnosti koje asociramo s ljudskim razmišljanjem, poput donošenja odluka, rješavanja problema, učenja...” (Bellmann, 1978.)
- „Uzbudljivi novi pokušaj da se omogući razmišljanje računalima...strojevi s umovima, u punom i doslovnom smislu.” (Haugeland, 1985.)

### 2. razmišljati racionalno

- „Proučavanje mentalnih svojstava kroz uporabu računalnih modela“ (Charniak and McDermott, 1985.)
- „Proučavanje postupaka koji mogućim čine percipiranje, rasuđivanje i reagiranje“ (Winston, 1992.)

### 3. ponašati se ljudski

- „Proces stvaranja strojeva koji obavljaju funkcije koje zahtijevaju inteligenciju koju imaju ljudi“ (Kurzweil, 1990.)
- „Proučavanje kako učiniti da računala rade stvari u kojima su, trenutno, ljudi bolji“ (Rich and Knight, 1991.)

### 4. ponašati se racionalno

- „Polje rada koje želi objasniti i emulirati inteligentno ponašanje u smislu računalnih procesa“ (Schalkoff, 1990.)
- „Grana računalnih znanosti koja se bavi automatizacijom inteligentnog ponašanja“ (Luger and Stubblefield, 1993.)

Umjetna inteligencija se bavi problemom „inteligentnog“ ponašanja stroja [4], a može ju se klasificirati prema vrsti problema koji rješava [4], [5]:

- sustavi za rješavanje uobičajenih ljudskih zadataka (prevođenje prirodnih jezika, prepoznavanje slika i govora, itd.),
- sustavi za rješavanje formalnih zadataka (logičke igre, matematička logika, geometrija, itd.),
- sustavi za rješavanje ekspertnih zadataka (znanstvena i financijska analiza, planiranje proizvodnje, itd.).



Sustavi koji se smatraju inteligentnim imaju sljedeća svojstva [4], [6]:

- pokazuju prilagodljivo ciljno usmjereno ponašanje,
- uče na temelju iskustva,
- koriste velike količine znanja,
- pokazuju svojstva svjesnosti,
- komuniciraju s čovjekom prirodnim jezikom i govorom,
- sustav mora komunicirati s čovjekom i drugim inteligentnim sustavima na prijateljski način,
- toleriraju pogreške i nejasnoće u komunikaciji,
- odgovaraju u stvarnom vremenu.

Turingov test [7], kojeg je predložio Alan Turing, bio je osmišljen na način da pruži zadovoljavajuću operativnu definiciju inteligencije. Turing je definirao inteligentno ponašanje kao sposobnost računala da postigne čovjekovu razinu performansi u kognitivnim zadacima (poznavanje, shvaćanje, primjena, analiza, sinteza i vrednovanje), dovoljno da zavara ispitivača koji je imao cilj otkriti u kojoj situaciji razgovara s čovjekom, a u kojoj s računalom. Ukoliko ih ne uspije razlikovati, za računalno bi se smatralo da je „prošlo“ Turingov test. Da bi računalno prošlo test trebalo bi imati sljedeće sposobnosti [8]:

- sposobnost prirodnog govora kako bi se omogućila komunikacija na određenom jeziku,
- prikaz ili reprezentaciju znanja u svrhu pohrane informacija do kojih se došlo prije ili tijekom ispitivanja,
- automatizirano zaključivanje, sposobnost da koristi pohranjene informacije kako bi odgovorio na pitanja i donosio zaključke i
- sposobnost učenja, odnosno prilagođavanja novim okolnostima i sposobnost djelovanja izvan nametnutih okvira.

## 2.1. Predstavljanje znanja

Predstavljanje znanja (engl. Knowledge Representation, KR) jedno je od važnih potpodručja umjetne inteligencije [4], [7], [9], [10], [11], a bavi se metodama formalnog zapisivanja znanja u računalu. Mnoga istraživanja bave se pronalaženjem metode za prikaz znanja. Računalni sustav temeljen na metodi formalizacije tekstualnog znanja treba imati sljedeće mogućnosti [12]:

- Na temelju postojećeg formaliziranog znanja sustav je sposoban za postavljanje pitanja za odgovore koji će se koristiti za obogaćivanje postojećeg znanja.

- Na temelju riječi koje stvaraju postojeće formalizirano znanje ili se nalaze u rječniku, sustav je u mogućnosti samostalno zaključiti prijedlog novih izjava. Ako su te izjave potvrđene, postojeće znanje se povećava.
- Na temelju izjava koje predstavljaju postojeće formalizirano znanje, sustav je u mogućnosti samostalno predložiti nove izjave (donositi zaključke) i ako su te izjave provjerene, postojeće znanje se povećava (primjerice, kroz dedukciju, indukciju i analogiju).
- Korištenjem metode analogije, sustav je u stanju predložiti zaključak na temelju formaliziranog znanja iz drugog sustava, neovisno o njegovom području primjene.

Za predstavljanje znanja razvijeni su različiti pristupi, formalizmi, metode i jezici koji variraju po složenosti i semantici [13]. Jedan oblik predstavljanja znanja čine grafičke metode koje u svom prikazu, koriste čvorove i veze među njima. Neke od tih metoda su [13], [14], [15], [16], [17], [18]: Basic Conceptual Graphs (BG), Multi-layered extended semantic networks (MULTINET), Hierarchical Semantic Form (HSF), Resource Description Framework (RDF) i Nodes of Knowledge (NOK).

### 2.1.1. Obrada prirodnog jezika

Obrada prirodnog jezika (engl. natural language processing - NLP) automatizacija je obrade ljudskog prirodnog jezika koja također može biti i poluautomatska obrada [19]. U aplikacijama za rad s prirodnim jezikom značajno je semantičko zaključivanje koje predstavlja problem u njihovoj izradi. Semantički zaključci su od temeljne važnosti za razumijevanje prirodnog jezika i sastavni su dio mnogih aplikacija za rad s prirodnim jezikom kao što su i aplikacije za odgovaranje na pitanja [20].

Za uspoređivanje različitih metoda za rad s prirodnim jezikom Sowa koristi tri različita stupnja semantike [21]:

- teška semantika (engl. heavyweight) - semantika je prikazana formalnom logikom s detaljnim aksiomima koji mogu podržati prošireno zaključivanje;
- srednje teška semantika (engl. middleweight) - temelji se na formalnim ili neformalnim zapisima koji podržavaju malu količinu zaključivanja;
- lagana semantika (engl. lightweight) - koristi oznake za klasifikaciju podataka, kako bi provjerila jednostavna ograničenja tipova i veza, ali ne i za izvođenje proširenog zaključivanja.

Mnogi sustavi koriste varijacije ove tri vrste semantike. Ljudska komunikacija rijetko kad je precizna i logična, ali svejedno ju je moguće razumjeti, a umjetna inteligencija nastoji dostići tu razinu razumijevanja znanja. Kako bi NLP postao dostupniji

za korištenje ljudima i iz drugih područja, osim lingvistike i računalne znanosti, razvijaju se alati koji šire NLP pismenost [22], [23].

### 2.1.2. Rječnici prirodnih jezika

NLP je povezan s lingvistikom, jer alati za obradu prirodnog jezika omogućavaju automatiziranu lingvističku analizu [24], [25]. Lingvistička analiza odnosi se na upotrebu jezika i na to kako specifične riječi i fraze zrcale ljudsko razmišljanje [26], [27]. U sustavima za obradu prirodnih jezika važnu ulogu imaju rječnici koji su strojno razumljivi i sadrže opisanu sintaksu i semantiku pojmova. Rječnici u kontekstu informatičkih znanosti uglavnom su knjige pojmova ili informacija o nekom području ljudskoga djelovanja s karakterističnim sustavom unakrsnoga referenciranja među pojmovima koji se rabe za organiziranje kolekcije pojmova da bi se oni objasnili i ponovno upotrijebili [28].

Prije desetak godina izvori informacija na internetu bili su namijenjeni ljudskom korisniku koji je njima upravljao posredstvom jednostavnih veza (linkova), dok u današnje vrijeme postoje programi za obradu podataka kreirani neovisno od podataka. Ideja semantičkog weba podrazumijeva da se Internet može pretraživati ne samo korištenjem riječi, već i kroz upotrebu značenja. To očigledno zahtijeva i semantičku i sintaktičku interoperabilnost predmetnog rječnika, budući da je dobro poznato da su za temeljit i sveobuhvatan opis predmeta potrebni ne samo izolirani pojmovi već i propozicijska logika [29], [30].

Za utvrđivanje semantičkih odnosa i hijerarhije u skupu riječi koje se nalaze u jednojezičnom rječniku može se koristiti pristup temeljen na formalnoj konceptualnoj analizi (engl. Formal Concept Analysis, FCA). Rezultat ovog postupka je koncept rešetke (engl. Galois Lattice) koji omogućuje hijerarhijsku organizaciju podataka, uspostavljanje veze između objekata, vizualizaciju veza između podataka i navigaciju do objekta u hijerarhijskoj strukturi [31], [32], [33].

Croatian wordnet (CroWN) [34] je semantička mreža hrvatskog jezika. Izrađena je na temelju Princeton WordNet-a [35] razvijenog na Sveučilištu u Princetonu u kojem se imenice, glagoli, pridjevi i prilozi grupiraju u skupove kognitivnih sinonima (synsets), svaki izražavajući drugačiji koncept. Rezultat je mreža smisleno povezanih riječi i pojmova kojima se može kretati uporabom preglednika [35].

Rječnik hrvatskih jezika autora Igalya [36] je rječnik koji ima za cilj obuhvatiti sve riječi koje se javljaju u tekstovima pisanim na hrvatskom jeziku od stručnih izraza do izraza u žargonu, regionalnih izraza i sl. Autor predviđa mnogobrojne primjene ovog rječnika: izrada jezičnih resursa potrebnih za računalnu provjeru pravopisa, prevođenje

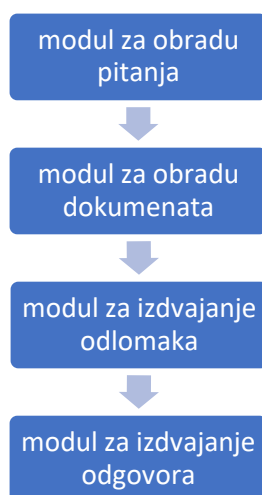
s hrvatskog na neki drugi jezik, testiranje algoritama iz teorije grafova, u kriptografiji, u sastavljanju križaljki, itd.

## 2.2. QA sustavi

Question Answering (QA) sustavi imaju za cilj odgovarati na pitanja postavljena prirodnim jezikom, odnosno nude automatizirani pristup traženja odgovora na ta pitanja [37]. Zajedničko svim QA sustavima su osnovne tri faze kroz koje se prolazi u pronalaženju odgovora [38], [39], [40], [41]:

- analiza pitanja,
- traženje kandidata za odgovor,
- odabir odgovora.

Ove faze prepoznatljive su i u okviru (engl. framework) QA sustava koji je podijeljen u 4 modula [42]: modul za obradu pitanja, modul za obradu dokumenata, modul za izdvajanje odlomaka i modul za izdvajanje odgovora (Slika 2.1).



**Slika 2.1. Framework Question Answering sustava [42]**

Mnoga istraživanja klasificirala su QA sustave [43], [44], [45], [42], [46], [41], [47] na temelju različitih kriterija, kao što su upiti koje postavljaju korisnici, vrste pitanja, značajke korištenih baza podataka, priroda generiranih odgovora, pristupi i tehnike odgovaranju na pitanja, a u [41] dana je sljedeća klasifikacija:

1. Klasifikacija temeljena na domeni:
  - Otvorena domena – odgovaraju na bilo koje pitanje,
  - Zatvorena ili ograničena domena – odgovaraju na pitanja na određenu temu.

2. Klasifikacija temeljena na tipu pitanja
  - Činjenična pitanja (*what, when, which, who, how*),
  - Pitanja na koja je odgovor nabranje,,
  - Pitanja na koja je odgovor definicija
  - Hipotetska pitanja,
  - Uzročna pitanja (*why, how*),
  - Potvrдна pitanja – odgovor je „da“ ili „ne“.
3. Klasifikacija temeljena na izvoru podataka
  - Sustav temeljen na dohvatanju podataka (*Information Retrieval*),
  - Sustav zasnovan na znanju (*Knowledge Based*),
  - Hibridni sustav – temelji se na višestrukim izvorima.

### 2.2.1. Large Language Models (LLM)

Veliki jezični modeli (*engl. Large Language Models – LLM*) u posljednje vrijeme dominiraju u području obrade prirodnog jezika i sustava koji odgovaraju na pitanja te je postignut značajan napredak u ovom području. Za razliku od baza znanja (*engl. Knowledge Bases*) koje obuhvaćaju velike količine podataka, LLM-ovi demonstriraju snažnu sposobnost generalizacije na širokom rasponu tekstualnih, tabličnih, općih i čak matematičkih zadataka za odgovaranje na pitanja s malo uzoraka u kontekstu [48].

U [49] su definirane tri paradigme koje, njih ili njihovu kombinaciju, koriste veliki pre-trenirani jezični modeli (*engl. pretrained language models – PLM*):

1. pre-treniranje zatim fino podešavanje – prvo se provodi općenito pre-treniranje s velikom količinom nespecifičnih izvora, a zatim „fino podešavanje“ na specifičnom zadatku,
2. učenje putem uputa – rješavanje zadatka se svodi na rješavanje onih zadataka koji su slični pre-treniranom modelu,
3. generiranje teksta – tekst se generira iz znanja kodiranog u generativnom jezičnom modelu (na primjer GPT).

Ispitivanja s modelima različitih veličina i u različitim formalnim jezicima pokazuju da razumijevanje logičkih oblika LLM-ova se uveliko približilo ljudskoj razini razumijevanja, ali i dalje ima mnogo prostora za poboljšanja u generiranju ispravnih logičkih oblika, što sugerira da je učinkovitije koristiti LLM-ove za generiranje više podataka na prirodnom jeziku kao izvor nekom manjem modelu nego izravno odgovaranje na pitanja. Također, rezultati evaluacije pokazali su da LLM-ovi pokazuju različitu osjetljivost na različite formalne jezike, gdje su oni s nižom razinom formalizacije za njih prihvatljiviji [50].

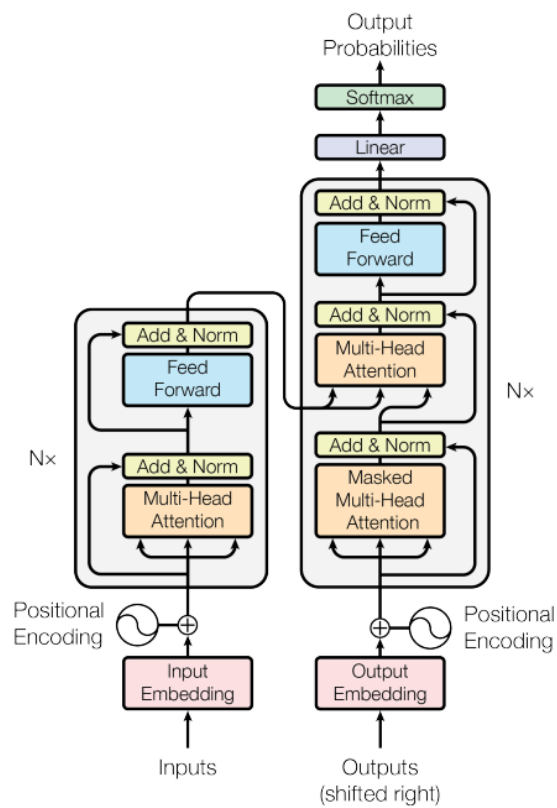
Trenutno najpoznatiji i najrašireniji LLM je GPT [52] kojeg je razvio OpenAI i danas je u uporabi u svoje tri inačice: ChatGPT 3.5 koja je dostupna besplatno, te GPT-4 i GPT-4o koja su dostupne za korištenje uz naknadu. U [51] na nizu različitih primjera testirane su verzije ChatGPT 3.5 i GPT 4 i prikazana usporedba dobivenih rezultata, ali i dokazi tvrdnji da su sposobnosti GPT-4 usporedive s ljudskom razinom za mnoge od njih, ali još puno toga treba učiniti kako bi se stvorio sustav koji bi se mogao kvalificirati kao potpuna generalna umjetna inteligencija (*engl. artificial general intelligence – AGI*). GPT – 4o dostupna je od svibnja 2024.

Model procesa GPT-a (*engl. Generative Pre-trained Transformer*) uključuje složenu arhitekturu baziranu na konceptu transformatorskih mreža. Ključne faze u tom procesu su [52]:

1. Predtreniranje koje uključuje učenje bez nadzora velikih količina teksta preuzetih iz različitih izvora za treniranje bez specifičnih zadatah zadataka; tokenizaciju kojom se tekst razbija na tokene, koji mogu biti riječi ili dijelovi riječi; ugrađivanje tokena pretvaranjem u numerički vektor, što omogućuje modelu da obrađuje tekst kao seriju brojevanih vrijednosti i transformatorske blokove koji se sastoje od niza slojeva koji koriste mehanizme pažnje da bi shvatili kontekst svakog tokena u odnosu na sve ostale tokene u upitu.
2. Fina podešavanja koja uključuju treniranje modela na manjim, specifičnim zadacima poput odgovaranja na pitanja, prevođenja ili pisanja teksta i na specifičnim skupovima podataka koji su direktno vezani za zadatke koje model treba izvršavati.
3. Generacija teksta – uključuje odabir tokena na temelju vjerojatnosti naučenih tijekom predtreniranja i fino podešavanja. Model uzima u obzir sve prethodne tokene u sekvenci kako bi predvidio najvjerojatniji sljedeći token.
4. Izlaz - generirani tekst se predstavlja korisniku kao konačni proizvod, bilo da je to odgovor na pitanje, dio priče, ili neki drugi oblik teksta.

Model GPT koristi duboko učenje (*engl. deep learning*) i napredne algoritme kako bi omogućio kompleksne zadatke generiranja jezika, čineći ga jednim od vodećih alata u području umjetne inteligencije za obradu prirodnog jezika.

Transformer predstavlja temeljni element modela funkcioniranja GPT-a, a njegovu arhitekturu prikazuje Slika 2.2, koja se sastoji od dva dijela Encodera (na slici lijevo) i Decodera (na slici desno) [53].



**Slika 2.2. Transformer - arhitektura modela [53]**

GPT i Node of Knowledge (NOK) su oba alati koji se koriste za obradu jezika i razumijevanje informacija, ali imaju različite pristupe i primjene. Slika 2.3 prikazuje odgovor GPT-a 4o na upit kojim se traži usporedba GPT-a i NOK-a.

Usporedba između ChatGPT-a i konceptualnog okvira Node of Knowledge (NOK) može se detaljno razmotriti kroz nekoliko ključnih aspekata: arhitektura, način funkcioniranja, primjena i specifične prednosti i nedostaci.

#### Arhitektura

##### ChatGPT:

Modeliranje: ChatGPT je model temeljen na dubokom učenju, posebno na Transformira arhitekturi. Trenirano je na velikim količinama tekstualnih podataka kako bi se postiglo razumijevanje i generiranje prirodnog jezika.

Podaci: Koristi širok raspon podataka prikupljenih iz različitih izvora na internetu, uključujući članke, knjige, web stranice, i još mnogo toga.

##### Node of Knowledge (NOK):

Struktura: NOK je okvir za organiziranje znanja koji koristi čvorove (nodes) za predstavljanje koncepta ili dijela informacija. Ti čvorovi su povezani na način koji omogućava hijerarhijsko ili mrežno predstavljanje znanja.

Podaci: Obično se temelji na specifično strukturiranim i kuriranim izvorima podataka, često u obliku baza podataka ili specijaliziranih repozitorija.

#### Način funkcioniranja

##### ChatGPT:

Interakcija: Korisnici postavljaju pitanja ili daju upite na prirodnom jeziku, a ChatGPT generira odgovore koristeći svoje sposobnosti razumijevanja i generiranja teksta.

Fleksibilnost: Može odgovarati na širok spektar pitanja i prilagoditi se različitim kontekstima i temama.

##### Node of Knowledge (NOK):

Navigacija: Korisnici pretražuju ili navigiraju kroz mrežu čvorova kako bi pronašli informacije.

Struktura čvorova omogućava lakšu identifikaciju odnosa između različitih koncepta.

Preciznost: Pruža precizne i dobro strukturirane informacije unutar specifičnih domena, omogućujući dublje razumijevanje specifičnih tema.

#### Primjena

##### ChatGPT:

Široka primjena: Koristi se za različite svrhe kao što su generiranje teksta, pomoć u pisanju, odgovor na pitanja, pružanje preporuka, i podrška korisnicima.

Razgovorni agent: Često se koristi kao chatbot za interakciju s korisnicima u stvarnom vremenu.

##### Node of Knowledge (NOK):

Specijalizirane domene: Idealno za upotrebu u akademskim, znanstvenim ili tehničkim domenama gdje je potrebno precizno i hijerarhijsko organiziranje znanja.

Obrazovanje i istraživanje: Koristi se za obrazovne svrhe, istraživanje i kao alat za podršku učenju i razumijevanju složenih tema.

#### Specifične prednosti i nedostaci

##### ChatGPT:

Prednosti: Visoka fleksibilnost, sposobnost obrade širokog spektra tema, interaktivnost i prilagodljivost različitim kontekstima.

Nedostaci: Može dati netočne ili neprecizne odgovore, ovisno o kvaliteti podataka na kojima je treniran. Nedostatak dubinske specifičnosti u određenim domenama.

##### Node of Knowledge (NOK):

Prednosti: Visoka preciznost i točnost u specifičnim domenama, strukturirano i hijerarhijsko predstavljanje informacija, korisnost za dublje istraživanje i razumijevanje.

Nedostaci: Manje fleksibilan u odnosu na ChatGPT, može zahtijevati više vremena za navigaciju i pronalaženje informacija.

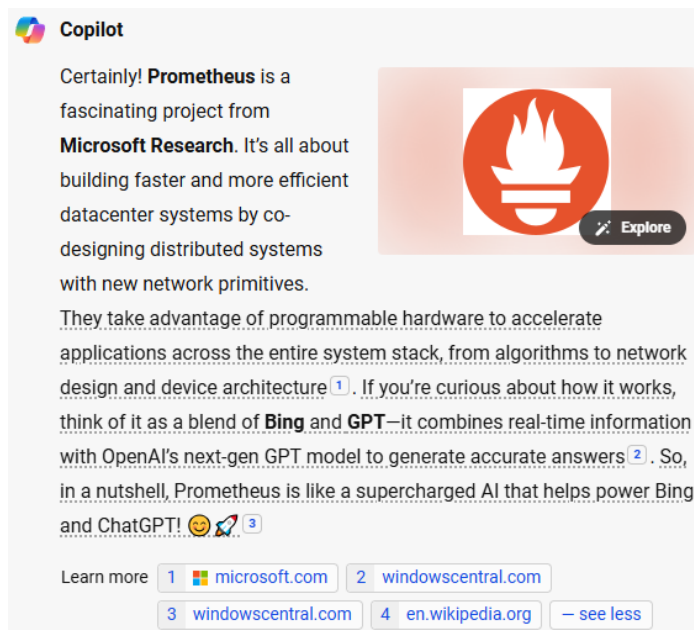
#### Zaključak

ChatGPT i Node of Knowledge (NOK) imaju različite prednosti i primjene, a odabir između njih ovisi o specifičnim potrebama korisnika. ChatGPT je svestraniji i bolje prilagođen za općenitu upotrebu i razgovornu interakciju, dok je NOK idealan za specijalizirane domene gdje je potrebno precizno i strukturirano znanje.

**Slika 2.3. Usporedba između ChatGPT-a i metode Node of Knowledge (NOK) koju je generirao ChatGPT 4o**

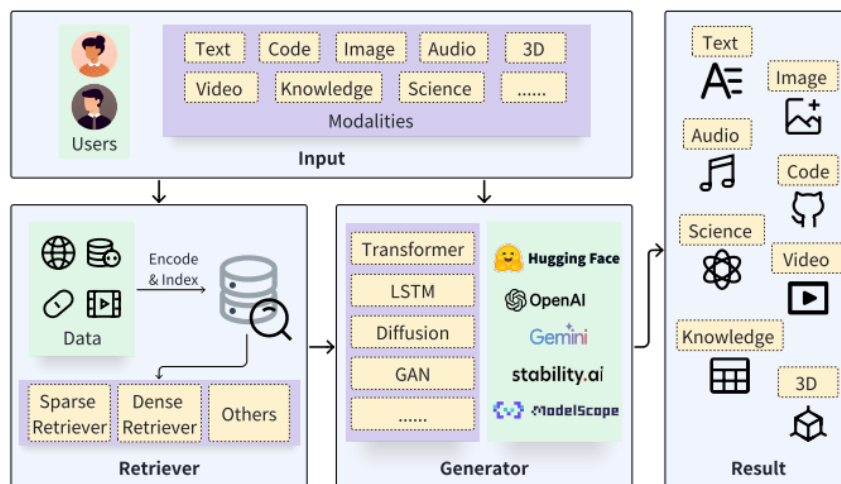


Generativni AI alat i LLM je i Microsoftov *Copilot* integriran u tražilicu *Bing* i kao Microsoft 365 alat. *Copilot* je izgrađen OpenAI-jevimi jezičnim modelima koje koristi i ChatGPT, ali je nadograđen Microsoftovim vlastitim Prometheus modelom [54] koji omogućuje pretraživanje i referenciranje online izvora pri samom kreiranju odgovora na upit. To mu omogućuje izdvajanje i izradu sažetaka informacija koje klasificira kao relevantne, obogaćujući tako svoje odgovore kontekstualno značajnim sadržajem izravno preuzetim s weba [55] (Slika 2.4).



**Slika 2.4.** Odgovor Copilota na upit „Što je Prometheus?“ s uključenim poveznicama na izvore

Veliki jezični modeli za traženje odgovora na upite pretražuju velike količine znanja i stoga dolazi u pitanje točnost dobivenih odgovora. Ovaj nedostatak često se naziva "halucinacijom" [56], [55], [57]. Metoda poboljšanja LLM-ova uključivanjem informacija iz vanjskih izvora znanja i povećanjem njihove točnosti sa stvarnim informacijama općenito je poznata kao Retrieval Augmented Generation (RAG) [56]. RAG je predložen je kao rješenje ovog i sličnih izazova (na primjer održavanje ažuriranog znanja, sprečavanje curenja privatnih podataka i slično) kroz svoj prilagodljivi repozitorij podataka [58]. Tipičan RAG proces (Slika 2.5) započinje ulaznim upitom nakon kojeg dohvatitelj (*engl. Retriever*) locira i pretražuje relevantne izvore podataka, a zatim dohvaćeni rezultati stupaju u interakciju s generatorom s ciljem poboljšanja cjelokupnog procesa generiranja [58].



Slika 2.5. Generička RAG arhitektura [58]

*Open Domain Question Answering* (ODQA) metode općenito sadrže dvostupanjsku arhitekturu u kojoj su *Retriever* i *Generator* trenirani odvojeno, dok RAG spaja ova dva dijela u jednu arhitekturu [59]. Korištenje RAG-a, odnosno parametarske i ne-parametarske memorije općenito dovodi do smanjenja „halucinacija“ i veće interpretabilnosti u zadacima kao što su odgovaranje na pitanja i sažimanje [59], [60], [61].

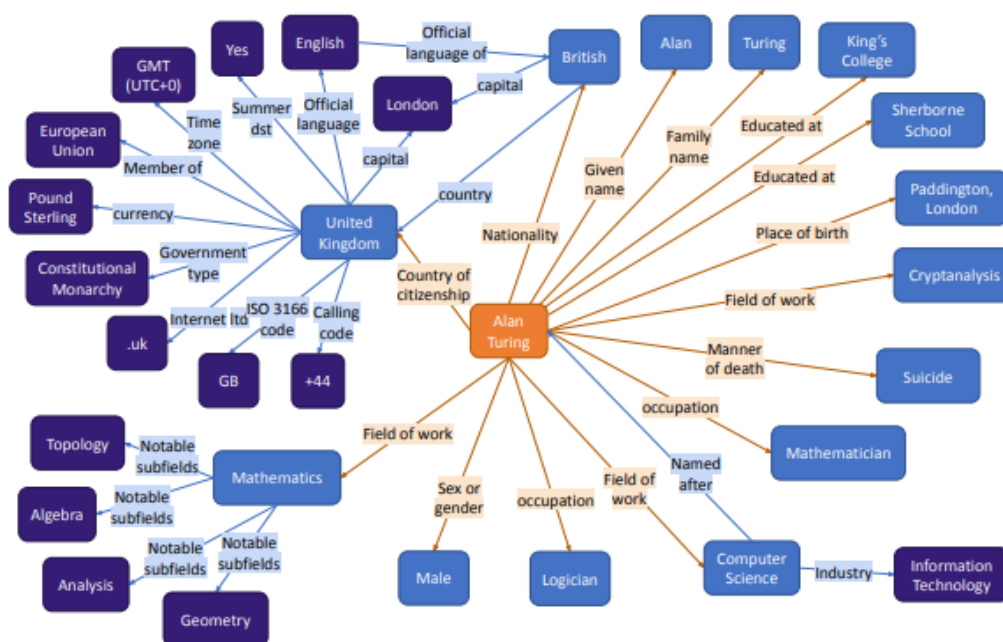
Veliki jezični modeli, kao što su GPT i Copilot, koriste za traženje odgovora na pitanja velike količine nestrukturiranog znanja, dok NOK konceptualni okvir karakterizira upravo strukturiranost. To NOK metodu čini prikladnom za rješavanje problema koji zahtijevaju strukturirano znanje i točno odabrani skup znanja u kojem se traže odgovori na pitanja. Kada se odgovori traže u konačnom skupu odabranog znanja vjerojatnost za generiranje krivog odgovora svodi se na minimum.

### 2.2.2. Knowledge Graph

Oblik predstavljanja znanja u jezičnim modelima su i grafovi znanja (engl. *Knowledge Graph* – KG). Grafovi znanja predstavljaju baze znanja strukturirane kao grafovi sastavljeni od entiteta i veza između njih [62]. U [63] definiraju grafove znanja kao grafove podataka namijenjene za akumulaciju i prijenos znanja o stvarnom svijetu, čiji čvorovi predstavljaju entitete od interesa, dok bridovi predstavljaju potencijalno različite odnose između tih entiteta.

Sustavi odgovaranja na pitanja korištenjem grafova znanja (engl. *Knowledge Graph Question Answering* - KGQA) imaju za cilj odgovoriti na pitanja prirodnog jezika pronalaskom entiteta iz grafa znanja (KG) [64]. Na primjer u metodi LMCRAWL KG je graf

čiji su čvorovi entiteti, a veze su predstavljene odnosima između entiteta, što ih čini čitljivim i usporedivim. Slika 2.6 prikazuje primjer grafa znanja za entitet „Alan Turing“ [65] koji se promatra kao postupak pretraživanja koji započinje od početnog entiteta i rekurzivno se proširuje na pojmove na sljedećoj razini kako bi se prikazale dodatne činjenice.



**Slika 2.6. Primjer generiranog grafa znanja oko entiteta ALAN TU RING, primjenom LMCRAWL [65]**

I metoda NOK je, kao i KG-ovi, grafička metoda koja za prikaz i strukturiranje znanja koristi entitete i veze, a međusobno se razlikuju u vrsti baza podataka koje koriste. NOK koristi relacijske baze podataka, a KG specijalizirane baze podataka kao što je su na primjer *Resource Description Framework (RDF)* i *Property Graphs* [66].

Postoje istraživanja koja se bave postavljanjem upita u prirodnom jeziku [67], [68], [69], [70], [71], gdje se oni pretvaraju u SQL upite na koje relacijska baza daje odgovor. U ovim istraživanjima još uvijek nije riješen problem zapisivanja rečenica prirodnog jezika u relacijsku bazu, a da se pritom čuva njihova semantika. Relacijsku bazu podataka, uz ostale baze podataka, u svom radu može koristiti i NLP metoda - *Natural Language Query Interface (NLQI)*, a njezina metoda prilagodbe za unos teksta u relacijsku bazu podataka opisana je u [72], [73], [74].

### 3. Konceptualni okvir Nodes of Knowledge (NOK)

Konceptualni okvir Nodes of Knowledge (NOK) pripada grupi semantičkih mreža kojima se znanje prikazuje kao graf [18]. Njezin cilj je prikazati znanje u tekstualnom obliku kao mrežu znanja, odnosno čvorova međusobno povezanih vezama. Svaki čvor predstavlja jedinicu znanja, koja se dalje može dijeliti u dijelove na nižoj razini apstrakcije ili imati attribute koji tom čvoru dodjeljuju podatke [17].

Karakteristike NOK-a su [75]:

- 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.

U usporedbi s drugim metodama, NOK metoda se ističe zbog jednostavnosti čitanja i implementacije te mogućnosti široke primjene [13].

Konceptualni okvir "Node of Knowledge (NOK)" je skup metoda i pravila te odgovarajućih alata za analizu i prikaz semantike sadržane u rečenicama prirodnog jezika, a obuhvaća [1]:

- 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.

Tekstualno znanje grafički se prikazuje u obliku dijagrama DNOK (Diagram Nodes Of Knowledge) [17]. Rečenice zapisane u ljudskom, prirodnom jeziku pretvaraju se u grafički model iz kojeg je moguće interpretirati njihovo značenje koje je sačuvano u izrađenom modelu. DNOK se sastoji iz čvorova, veza između njih, te identifikatora uloge u vezi. Oni zajedno čine mrežu znanja. Čvor je najmanji element znanja koji se ne može dijeliti. U tekstualnom znanju, pod najmanji element znanja se podrazumijeva riječ ili

skupina riječi u nekoj rečenici. Ime čvora je atribut čvora i on mu daje semantičko značenje. Ime se može sastojati i od više riječi koje imaju neko značenje. Za implementaciju metode neophodno je svakom čvoru dodijeliti jedinstveni identifikator u mreži.

Čvorove dijelimo na [18, 76, 12]:

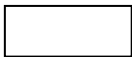
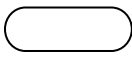


- čvor – obični, statični, entitet, koncept, termin,
- kontekstni čvor – čvor većeg stupnja apstrakcije (skupina, vrsta, klasa, tip),
- podatkovni čvor – mjesto gdje se čuvaju podaci,
- procesni čvor – čvor koji vezama spaja čvorove i s njima kreira složeniju prezentaciju znanja. Njime se prezentira znanje koje se ne može prikazati običnim čvorom kao što su veze između više čvorova, aktivnosti, pojave i slično. Najčešća vrsta riječi u imenima procesnih čvorova su glagoli.

Veze povezuju čvorove u mrežu znanja. Veze nemaju imena, ali mogu imati uloge koje pripadaju čvorovima [76]. Imena uloga su pitanja kojim se povezuju dva susjedna čvora. Veza se prikazuje linijom, sa strelicom ili bez nje. Strelice se upotrebljavaju kada se spajaju čvorovi na različitim razinama apstrakcije [17] ili za direktno povezivanje procesnih čvorova. Brojnost veze uvijek je (1,1):(1,1) [76]. Iz dobivenog modela se, bez uvida u originalni tekst, mogu dobiti odgovori na postavljena pitanja.

### 3.1. Primjena NOK-a na modeliranje složenih rečenica

Primjena grafičkog prikaza znanja dana je u [77, 78] na primjeru modeliranja basne Ezopove basne „Zlatna jaja u kokoši“ [79] na hrvatskom jeziku i "The Goose with the golden egg" [80] na engleskom jeziku. Za grafički prikaz čvorova i koncepata u NOK metodi korišteni su koncepti koje prikazuje Tablica 3.1.

**Tablica 3.1. Koncepti NOK metode**

Simbol	Značenje
	ČVOR – dio znanja u stvarnosti ili mislima
	PROCESNI ČVOR – radnja, događaj, zbivanje
	VEZNIK – složena rečenica od više dijelova
	VEZA jednosmjerna i pitanje uz vezu – povezivanje čvorova

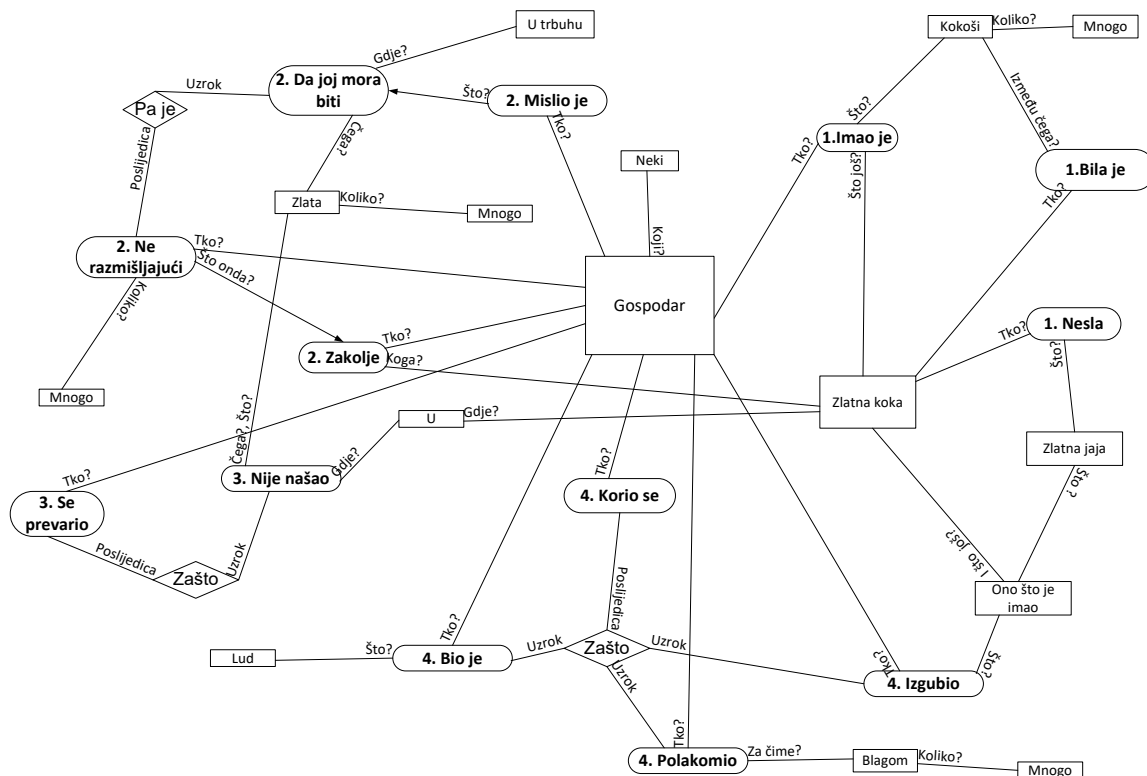
Basna „Zlatna jaja u kokoši“ [79] na hrvatskom jeziku ima sljedeće 4 rečenice:

1. *„Neki gospodar imao je među mnogim svojim kokošima i jednu takovu koja je nesla zlatna jaja.*
2. *On je mislio, da joj u trbuhu mora biti mnogo zlata, pa je, ne razmišljajući mnogo, zakolje.*
3. *Ali se ljuto prevario, ne našavši u njoj ništa drugo nego li i u ostalim kokošima.*
4. *Tad se uzalud sam korio što je bio lud, da se polakomio za mnogim blagom, pa tako izgubio i ono što je imao.“*

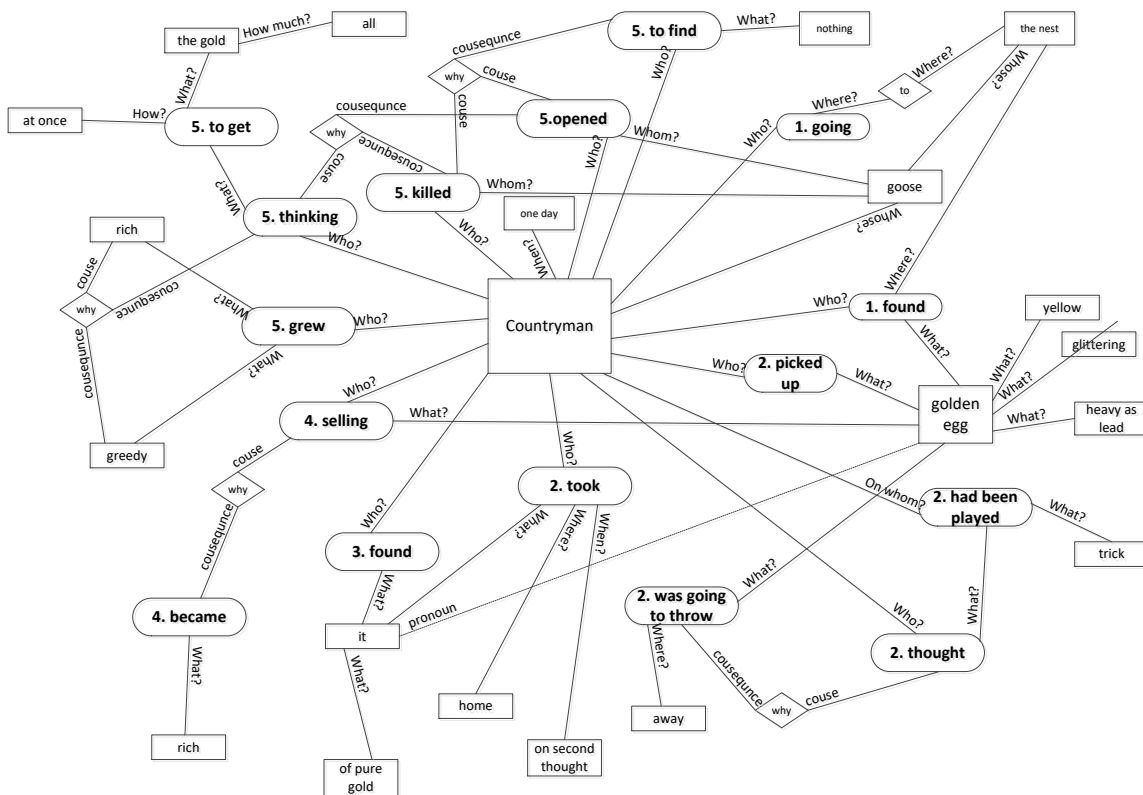
Ista basna, „*The Goose with the golden egg*“ [80] na engleskom jeziku glasi:

1. *„One day a countryman going to the nest of his Goose found there an egg all yellow and glittering.*
2. *When he took it up it was as heavy as lead and he was going to throw it away, because he thought a trick had been played upon him.*
3. *But he took it home on second thoughts, and soon found to his delight that it was an egg of pure gold.*
4. *Every morning the same thing occurred, and he soon became rich by selling his eggs.*
5. *As he grew rich he grew greedy; and thinking to get at once all the gold the Goose could give, he killed it and opened it only to find nothing.“*

Određivanjem čvorova, procesnih čvorova i veznika, gdje se za njima ukazala potreba, i povezivanjem vezama kojima je pitanjima dana uloga i značenje čvorova u rečenici složena su oba grafička prikaza, na hrvatskom (Slika 3.1) i na engleskom jeziku (Slika 3.2).



Slika 3.1. Prikaz basne "Zlatna jaja u kokoši" primjenom NOK metode

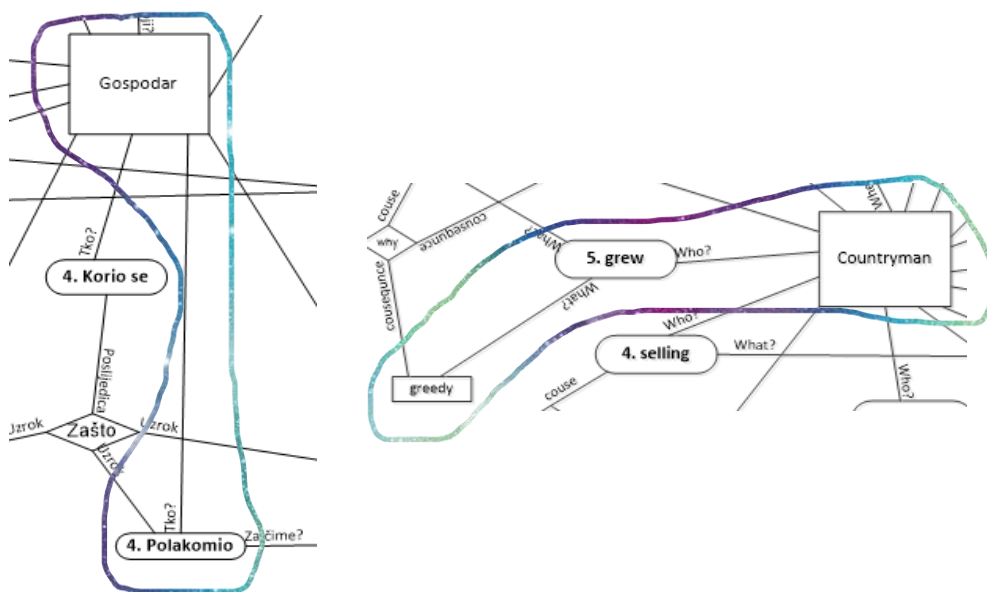


Slika 3.2. Prikaz basne "The Goose with the golden egg" primjenom NOK metode

Na prvi pogled je vidljivo, usporedbom teksta basni, da postoji razlika u prijevodu koja je uvjetovana običajima, narodnom predajom i podnebljem za koje je namijenjena i, naravno, slobodom prevoditelja. Već u samom naslovu uočljiva je razlika u odabiru životinje koja je nesla zlatna jaja: u hrvatskoj verziji to je kokoš, a u engleskoj guska. Iz ovoga možemo zaključiti da prevođenje nije matematički strogo preslikavanje pojmova u jednom jeziku u iste pojmove u drugom jeziku. Također, dva ista prevoditelja neće isti tekst prevesti na isti način. Prema tome u prevođenju ne postoji jedno jedino najbolje rješenje. Prevoditelji smatraju da je njihov prijevod sačuvao poruku odnosno znanja koje je autor htio prenijeti čitatelju. Unatoč tome basna ima istu poantu i ispričana je na sličan način. To znači da se znanje sadržano u basni može sačuvati u oba jezika na različit način. Ovo dovodi do zaključka da je moguće čuvati znanje u modelima različitim od zapsanog teksta.

Pogledom na grafički prikaz (Slika 3.1 i Slika 3.2) uočava se razlika u prikazu. Jasna je i razlika u složenosti njegove izrade. Engleski prijevod basne je složeniji u pogledu broja čvorova i veza kojima je prikazan. Ima i veći broj rečenica.

Značajno za NOK metodu je dobivanje odgovora na pitanja iz izrađenog modela. Ako zanemarimo razliku u prijevodu kokoš/guska iz modela možemo dobiti slične ili iste odgovore na ista pitanja iako je raspored čvorova u svakom od modela različit: na primjer „*Tko je nesao zlatna jaja?*“, „*Tko se polakomio?*“ (Slika 3.3), „*Tko je zaklao kokoš/gusku?*“.



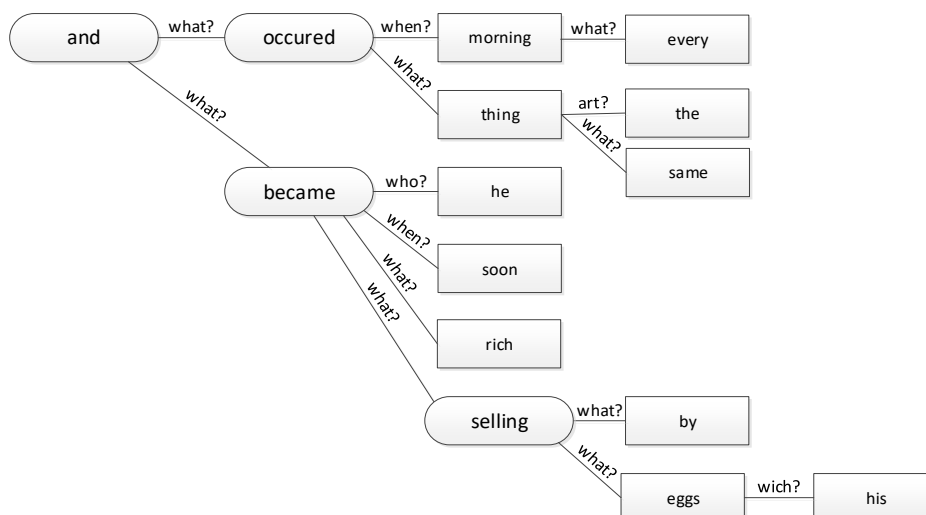
**Slika 3.3. Odgovor na pitanje „Tko se polakomio?“/“Who grew greedy?“**



S druge strane postoje i pitanja primjenjiva samo na jedan model, kao što je na primjer „Tko je bio lud?“ u hrvatskoj inačici ili „*Who grew rich?*“ u engleskoj. NOK metoda primjenjiva je na modeliranje rečenica zapisanih u različitim prirodnim ljudskim jezicima. S obzirom na slobodu u prijevodu neki dijelovi modela ostaju slični, ali u nekima postoji razlika u istoj priči, na istim rečenicama. Prijevodom i spajanjem različitih modela, a ne samo modeliranjem prijevoda, može se dobiti model na bilo kojem jeziku iz kojeg bi se mogla iščitavati ista značenja. Moguće je odabrati tekstove koji su sličnije prevedeni i u tom slučaju dobilo bi se veće preklapanje modela. Dodatno, moguće je mjeriti razliku u modelima i prijevodima po broju koncepata i riječi te istraživati kako NOK metodu koristiti za prevođenje tekstova te poboljšavati proces prevođenje jednog jezika u DNOK i obrnuto.

U [78] za basnu na engleskom jeziku „*The Goose with the golden egg*“ [80] za svaku rečenicu posebno izrađen je grafički zapis u obliku dijagrama DNOK i u formaliziranom FNOK obliku. Ovako zapisane rečenice priprema su za zapisivanje u relacijsku bazu podataka. Za svaku od rečenica analizirana je svaka riječ u njoj i određeno je kojoj vrsti riječi pripada. Između svih riječi izdvojeni su glagoli kojih u svakoj rečenici, s obzirom na njihovu složenost, ima više. U jednostavnim rečenicama sa samo jednim glagolom glagol se stavlja na prvo mjesto. U složenim rečenicama potrebno je odrediti točnu semantičku pripadnost i hijerarhiju između glagola, odnosno redoslijed odvijanja radnji i prvi glagol iz hijerarhije stavlja se na prvo mjesto. Ukoliko je rečenica složena od dvije rečenice koje su međusobno povezane veznikom, veznik se postavlja na prvo mjesto u hijerarhiji, kao što je slučaj u drugoj, trećoj, četvrtoj i petoj rečenici promatrane basne. Nakon veznika u svakoj rečenici se pronalazi glagol koji se postavlja na nižu razinu hijerarhije, a zatim na sljedeće razine hijerarhije postavljaju se sve ostale riječi s kojima je glagol povezan.

Na primjer, u četvrtoj rečenici basne na engleskom jeziku „*Every morning the same thing occurred, and he soon became rich by selling his eggs.*“ nalaze se tri glagola, od kojih su dva povezana veznikom. Analizom rečenice pronađeni glagoli u ovoj rečenici su: *occured, became i selling*. Dvije jednostavne rečenice povezane su u jednu složenu i međusobno povezane veznikom *and*. U tom slučaju veznik *and* promatra se kao procesni čvor i postavlja na vrh hijerarhije ispod kojeg su upisani glagoli *occured i became*, a glagol *selling* stavljen je na novu razinu ispod *became* jer se nalazi nakon *became*. Na sljedeću razinu idu riječi povezane s glagolima (na primjer *morning, thing*) i dalje se gradi hijerarhija ovisno o međusobnoj povezanosti među riječima (Slika 3.4).



**Slika 3.4. DNOK četvrte rečenice basne na engleskom jeziku**

Kako bi se rečenice iz basne pripremile za upisivanje u relacijsku bazu podataka potrebno ih je zapisati u formaliziranom obliku FNOK. Formalizacija NOK metode prikazana je u [12], [81]. Ova metoda analizira prirodni ljudski jezik, rečenice, riječi i njihovo značenje, kao i povezane nizove riječi koje stvaraju složenije fraze [82]. NOK metoda prvo pretvara znanje u model i zatim pretražuje znanje u modelu. Nakon toga je u stanju odgovoriti na pitanja povezana s tim znanjem [83].

Kod zapisivanja u FNOK zapise prati se hijerarhija vidljiva u DNOK-u tako što se za svaku novu razinu otvaraju zagrade, a riječi iz rečenica s pripadajućim pitanjima na istoj razini odvajaju se zarezom. FNOK zapis 4. rečenice basne glasi:

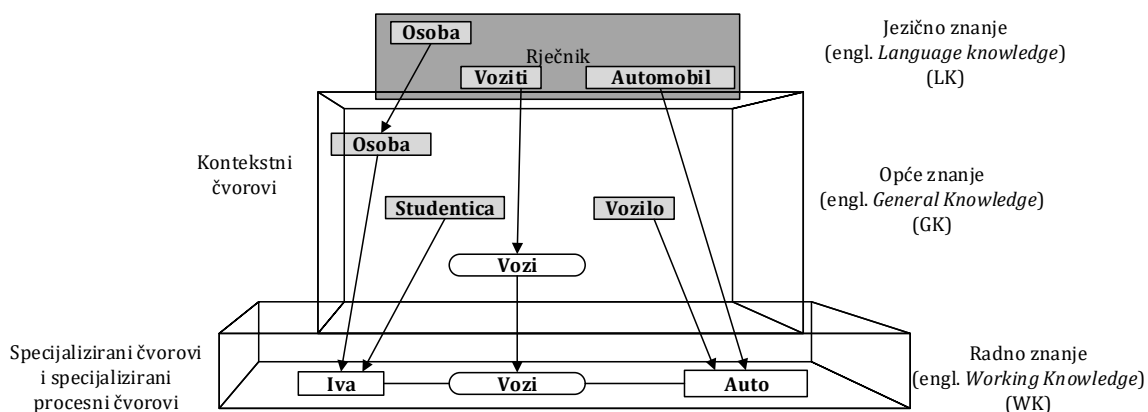
*and („what?“ occurred („when?“ morning („what?“ every), „what?“ thing („art?“ the, „what?“ same), „what?“ became („who?“ he, „when?“ soon, „what?“ rich, „what?“ selling („what?“ by, „what?“ eggs („which?“ his))))))*

Analiza rečenica iz basne i njihovi zapisi u DNOK i FNOK obliku poslužili su kao temelj za izradu pravila zapisivanja složenih rečenica u FNOK zapise opisanim u poglavlju 4 na temelju kojih je provedeno cjelokupno istraživanje u ovom radu.

U [1] je opisan meta model baze podataka složene NOK metode (koja u vezi ima dvije uloge) a u [84] je opisan model s jednom ulogom. Cilj je uzeti predloženi okvir za pretvaranje teksta u relacijsku bazu podataka te njega doraditi kako bi se poboljšali odgovori na pitanja i kako bi se mogle nove vrste rečenica transformirati u bazu znanja. Do sada su definirana pravila transformacije rečenica prirodnog jezika u FNOK zapis za glagole [85], imenice [86], pridjeve [87], većinu zamjenica, brojeve, priloge i prijedloge [88] te pitanja prirodnog jezika u QFNOK zapis za hrvatski i engleski jezik [1]. Opisani su i algoritmi za transformaciju FNOK zapisa u relacijsku bazu podataka, QFNOK zapisa u

relacijsku bazu podataka te kreiranje odgovora [1]. Ovi rezultati temelj su za daljnja istraživanja proširenja postojećih relacijskih baza podataka tekstualnim znanjima.

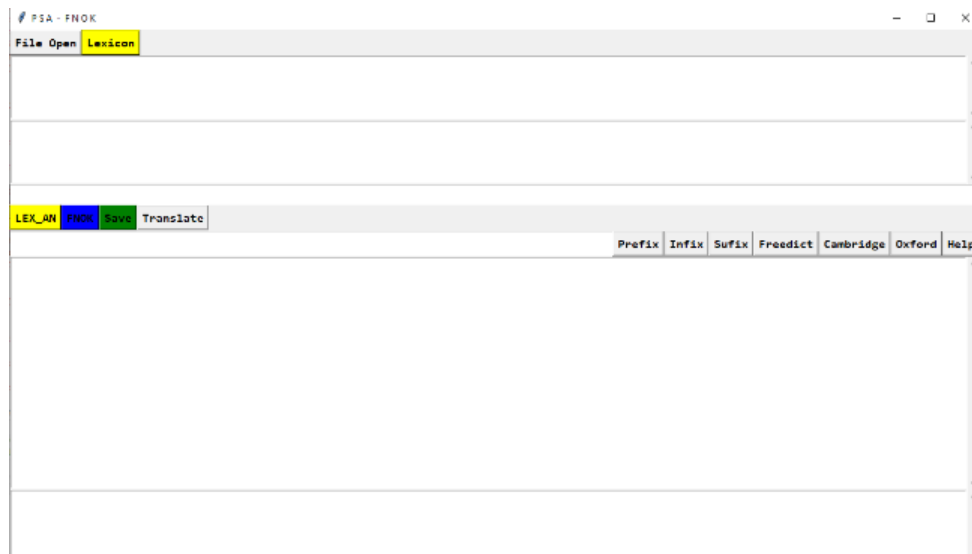
Rečenice zapisane u prirodnom jeziku pretvaraju se u model i iz kojeg je moguće interpretirati značenje koje je u njemu sačuvano. Model se sastoji od čvorova i veza između njih koje zajedno čine mrežu znanja. Čvor je najmanji element znanja koji se ne može dijeliti, a svaki novi pojam ili novo značenje je novi čvor i on je nedjeljiv [13]. Tekstualno znanje u prirodnom jeziku NOK metodom organizira se u mrežu znanja (knowledge network, KN) koja se sastoji od tri različite razina znanja [17]. To su jezično znanje (LK, language knowledge) koje je na najvišoj razini, zatim opće znanje (GK, general knowledge) na sljedećoj razini i radno znanje (WK, working knowledge) koje je složeno od gornjih razina i predstavlja konkretno znanje što prikazuje Slika 3.5. NOK metoda omogućuje prikazivanje rečenica grafički u obliku dijagrama DNOK (Diagram Nodes Of Knowledge) [77] i u formaliziranom obliku FNOK [12], [81].



**Slika 3.5. Razine znanja [1]**

### 3.2. Automatsko prevođenje uporabom PSA – FNOK aplikacije

Za automatsko pretvaranje rečenice engleskog jezika u formalizirani NOK zapis (FNOK), u programskom jeziku Python izrađena je aplikacija PSA – FNOK opisana u [81, 12, 89]. Slika 3.6 prikazuje korisničko sučelje aplikacije koja je testirana na jednostavnim rečenicama objavljenim u [90], a provjeravala se njihova primjenjivost na postojećem QA sustavu.



*Slika 3.6. Korisničko sučelje aplikacije PSA - FNOK*

Temeljni dio ove aplikacije predstavlja leksikon. Leksikon je tekstualna datoteka u koju su upisane riječi i njihovi oblici s pripadajućim atributima, a temeljen je na rezultatima projekta „MULTEXT-East“ za engleski jezik [91], [92], [93]. MULTEXT-East projekt (Multilingual Text Tools and Corpora for Eastern and Central European Languages) je započeo 1995., trajao je dvije godine i njime je razvijen standardizirani jezični izvor za šest jezika centralne i istočne Europe i za engleski jezik [94]. Pored ostalih rezultata ovoga projekta, dva su ključna za formalizaciju znanja iskazanoga tekstem – leksikon i morfosintaktička specifikacija leksičkih elemenata koji su standardizirani za sve uključene jezike [12].

MULTEXT-East leksikoni imaju jednostavnu strukturu, gdje se svaki leksički unos sastoji od tri polja:

1. oblik riječi, kako se pojavljuje u tekstu;
2. lema, koja je osnovni oblik riječi i
3. MSD, tj. tj. morfosintaktički opis.

Prva oznaka u morfosintaktičkom opisu predstavlja kategoriju, odnosno vrstu riječi, a ostali atributi definirani su za svaku pojedinu vrstu prema njenim značajkama. Kategorije definirane MULTEXT-East leksikonu su sljedeće [91]:

- imenica - N Noun,
- glagol - V Verb,
- pridjev - A Adjective,
- zamjenica - P Pronoun,
- odrednica - D Determiner,
- prilog - R Adverb,
- prijedlog - S Adposition,
- veznik - C Conjunction,
- broj - M Numeral,
- Čestica - Q Particple,
- Usklik - I Interjection,
- Kratica - Y Abbreviation,
- Ostalo - X Residual.

Za točnije i lakše definiranje atributa u aplikaciji su kreirane su poveznice na dva online rječnika Cambridge [95] i Oxford [96] gdje su za svaku riječ detaljno opisana sva svojstva i uloge u rečenici.

Prva faza u formalizaciji tekstem iskazanoga znanja je faza parsiranja - provjera sintaktičke ispravnosti rečenice i utvrđivanje njezinih sastavnih dijelova. U svrhu sintaktičke analize rečenice potrebno je uspostaviti generativnu gramatiku - leksikon i pravila pretvaranja. Uspostavljena gramatika koristi morfosintaksički leksikon engleskoga jezika MULTEXT-East, dok će se pravila izgraditi prema gramatici frazne strukture (Phrase structure grammar - PSG) za engleski jezik, a sukladno s [97], [98] i [99], ali sa simbolima prilagođenima notaciji MULTEXT-East morfosintaksičkom opisu (MSD). Morfosintaksički leksikon engleskoga jezika MULTEXT-East proširen je semantičkim dijelom kojim se za svaki oblik riječi identificiraju i wh-pitanja (pitanja koja počinju upitnim riječima: *who, what, when, where, why, how, how often*, i slično) na koja riječ može primiti odgovor, odnosno na koja riječ može dati odgovor. Na taj način izgrađuje se leksikon kojeg autori [12] nazivaju morfološko sintaktičko-semantički leksikon.

### 3.2.1. Atributi riječi

Za svaku pojedinu vrstu riječi definirane su različite skupine atributa koje opisuju njihova svojstva. Iza prvog slova koji označava vrstu riječi slijede ostali atributi riječi. Na primjer, imenice imaju 5 različitih skupina atributa: vrstu, rod, broj, živost i prefiks, kao i glagoli: vrsta, oblik, vrijeme, lice i broj.

Tablica 3.2 prikazuje attribute imenica, a Tablica 3.3 attribute glagola. U tablicama su prikazani atributi za glagole i imenice (prema [93]) sa sljedećim stupcima:

- Pozicija - položaj atributa u nizu morfosintaktičkog opisa;
- Atribut - naziv atributa;
- Vrijednost – naziv vrijednosti;
- Kod - kod vrijednosti.

**Tablica 3.2. Atributi imenica**

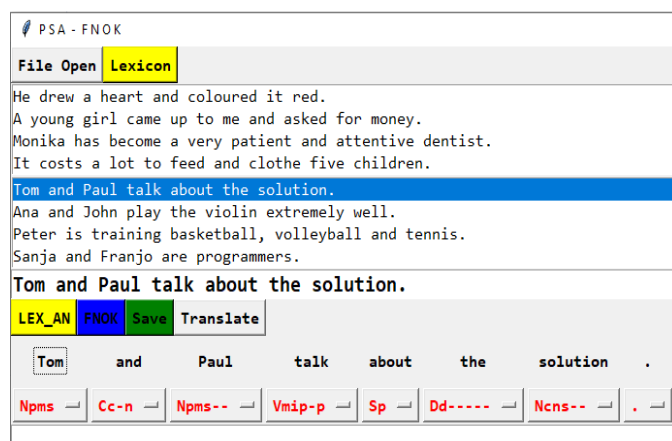
Pozicija	Atribut	Vrijednost	Kod
1	vrsta	zajednička	c
		vlastita	p
		zbirna	x
2	rod	muški	m
		ženski	f
		neodređeni	n
3	broj	jednina	s
		množina	m
		nebrojiva	0
4	živost	neživa	0
		živa	l
		gradivna	m
		apstraktna	a
5	prefiks	suglasnik	c
		samoglasnik	v

**Tablica 3.3. Atributi glagola**

Pozicija	Atribut	Vrijednost	Kod
1	vrsta	glavni	m
		pomoćni	a
		modalni	o
		osnovni	b
2	oblik	indikativni	i
		kondicional	c
		infinitiv	n
		particip	p
3	vrijeme	sadašnjost	p
		prošlost	s
		imperpekt	i
		futur I.	f
4	lice	prvo	1
		drugo	2
		treće	3
5	broj	jednina	s
		množina	p

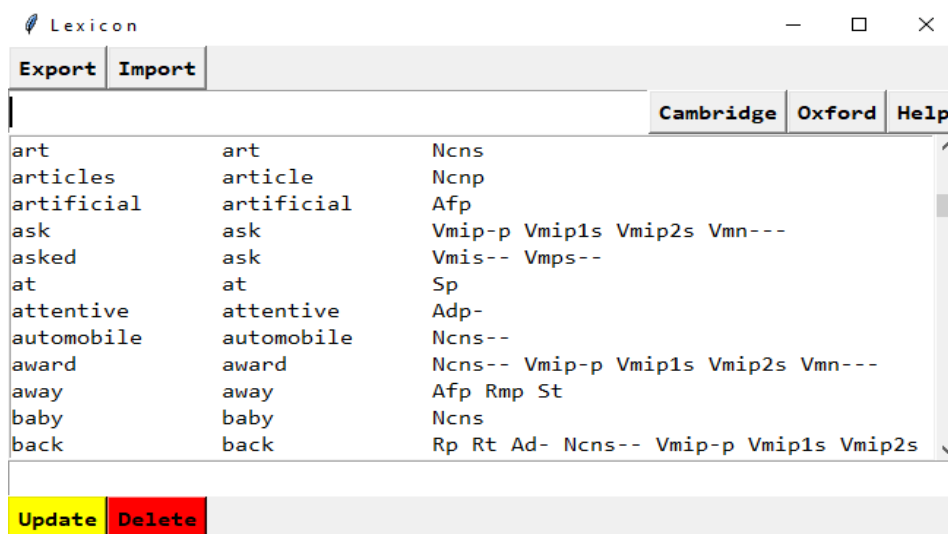
### 3.2.2. Primjena PSA – FNOK aplikacije

Prvi korak u uporabi PSA FNOK aplikacije je unos rečenice u aplikaciju što je moguće učiniti na dva načina: učitavanjem iz tekstualne datoteke niza rečenica i odabira željene rečenice (Slika 3.7) ili direktnim upisivanjem rečenice.



Slika 3.7. Rečenice učitane iz datoteke

Sljedeći korak je leksička analiza rečenice koja se pokreće naredbom LEX\_AN. Za leksičku analizu koristi se unaprijed izrađen *Lexicon* (Slika 3.8), tekstualna datoteka koja sadrži riječi engleskog jezika i njihove oblike opisane unaprijed definiranim atributima. U slučaju kad pojedini oblik riječi ne postoji u leksikonu dodaje se u njega uporabom naredbe *Update*. Na isti način se dodaje i riječ koja nije pronađena u *Lexiconu*, što je najčešće slučaj s osobnim imenima.



Slika 3.8. Lexicon

U prvom stupcu *Lexicona* nalazi se oblik riječi, u drugom osnovni oblik riječi, a u trećem su skupine atributa kojima se definiraju mogući načini njenog korištenja. Na primjer riječ „*kiss*“ (Slika 3.9) ima u *Lexiconu* definirano 5 skupina svojstava atributa ovisno o njenom značenju. „*Kiss*“ u rečenici može, ovisno o kontekstu, pojavljivati se i kao imenica i kao glagol. Na prvo mjesto u *Lexicon* se upisuje ona skupina atributa koja se najčešće pojavljuje u rečenicama. Redoslijed skupina atributa može se promijeniti u *Lexiconu* u svakom trenutku ili prilikom leksičke analize odabrati odgovarajuću skupinu.

kiss	kiss	Ncns-- Vmip-p Vmip1s Vmip2s Vmn---
------	------	------------------------------------

**Slika 3.9. Atributi riječi "kiss"**

Prva skupina atributa opisuje riječ „*kiss*“ kao imenicu čiji su atributi „zajednička, neodređenog spola, u jednini“. Za preostala dva atributa, koji za ovu riječ nisu definirani, upisane su crtice. Na ovaj način se obilježavaju atributi koji nisu definirani. Preostale skupine atributa ovu riječ opisuju kao „glavni“ glagol koji se pojavljuje u obliku „indikativ“ u „sadašnjosti“, ali u različitim licima i brojnosti. Kako je ovdje u attribute uključeno samo glagolsko vrijeme „sadašnjost“, preostaje doraditi *Lexicon* i s ostalim glagolskim vremenima.

Opisani atributi u aplikaciji PSA – FNOK pridružuju se svakoj riječi prilikom leksičke analize (naredba LEX\_AN), a značajni su prilikom pretvaranja u FNOK zapis jer određuju hijerarhiju i pitanja sadržana u FNOK zapisu u skladu s pravilima pretvaranja. Slika 3.10 prikazuje dio koda iz aplikacije kojim se prema atributima definiraju pitanja za FNOK zapis.



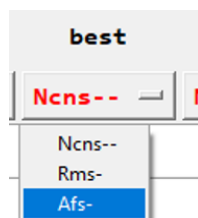
```

PA = [ (r"D[di]"      , "'art?'"      ),
        (r"St"       , "'when?'"    ),
        (r"Sp"       , "'where?'"   ),
        (r"Sm"       , "'how?'"     ),
        (r"N[cp][fm]" , "'who?'"     ),
        (r"N[cp]n"    , "'what?'"    ),
        (r"(Dg|Ds|N[cp]\')", "'whose?'"   ),
        (r"Np[fm]"    , "'who?'"     ),
        (r"Pg"       , "'what?'"    ),
        (r"Pp"       , "'who?'"     ),
        (r"Px"       , "'whom?'"    ),
        (r"Pw"       , "'"'         ),
        (r"Ab"       , "'from what?'" ),
        (r"A[df]p"    , "'how?'"     ),
        (r"A[dfx]"    , "'what?'"    ),
        (r"As"       , "'whose?'"   ),
        (r"Mc"       , "'how_many?'" ),
        (r"Md"       , "'when?'"    ),
        (r"Mr"       , "'what?'"    ),
        (r"Rm"       , "'how?'"     ),
        (r"Rf"       , "'how_often?'" ),
        (r"Rd"       , "'how_much?'" ),
        (r"Rt"       , "'when?'"    ),
        (r"R(p|pp)"  , "'where?'"   ), ]

```

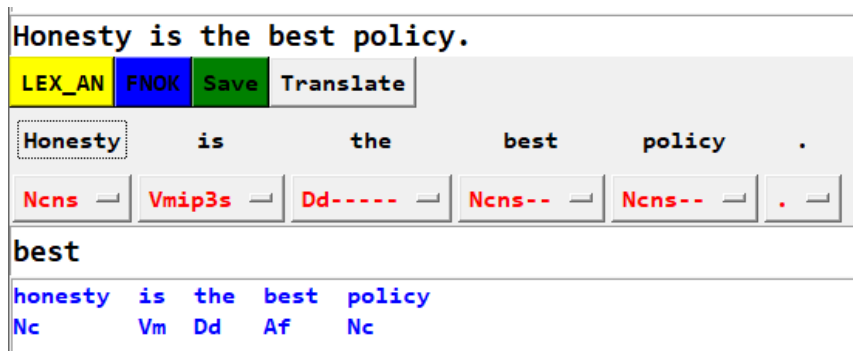
Slika 3.10. Kod kojim se određuju pitanja prema atributima uz riječi

Na primjeru rečenice „*Honesty is the best policy.*“ pokazat će se funkcioniranje aplikacije PSA – FNOK. Prvi korak nakon upisivanja rečenice u aplikaciju, direktnim unosom ili učitavanjem iz datoteke je pokrenuti leksičku analizu naredbom LEX\_AN kojom se svakoj riječi pridruže atributi iz prve skupine atributa pridruženih toj riječi u Lexiconu. U većini slučajeva pridruženi atributi odgovaraju vrsti i ulozi riječi u rečenici, ali u ovom trenutku je potrebno provjeriti je li to stvarno tako. Leksičku analizu testne rečenice prikazuje Slika 3.11 na kojoj se može uočiti da je analizom dobiveno da je riječ „*best*“ imenica, što u ovoj rečenici nije slučaj i ova riječ se javlja kao pridjev.



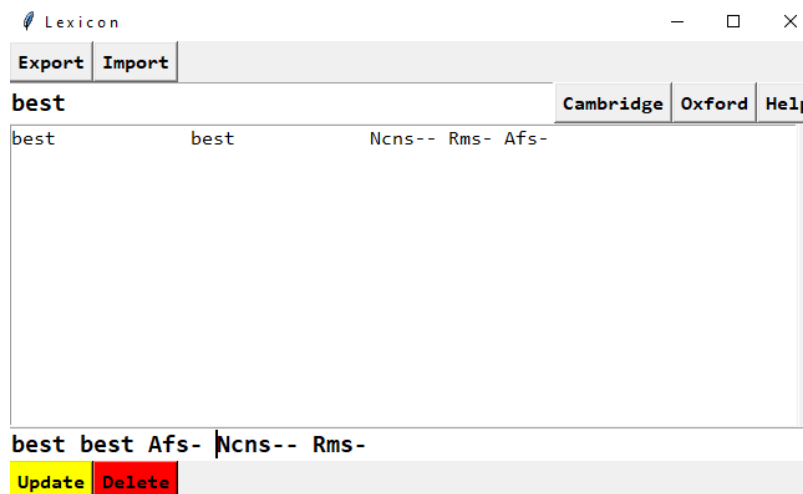
Slika 3.11. Promjena atributa riječi "best"

Klikom na gumb s atributima riječi „best“ otvaraju se svi atributi upisani u *Lexicon* među kojima se odabere odgovarajući. U ovom slučaju odabire se skupina atributa koja ovu riječ opisuje kao pridjev (Slika 3.12).



Slika 3.12. Leksička analiza rečenice "Honesty is the best policy."

S obzirom da se riječ „best“ najčešće u rečenicama pojavljuje kao pridjev (Afs-), a rjeđe kao imenica (Ncns--) i prilog (Rms-), u *Lexiconu* je poželjno izmijeniti redoslijed atributa kako bi automatska leksička analiza rečenice bila što točnija i sa što manje intervencija. Atribute pridjeva (Afs-) potrebno je postaviti na prvo mjesto za primjenu na budućim leksičkim analizama rečenica u kojima se „best“ pojavljuje. Izmijenjeni redoslijed atributa za riječ „best“ prikazuje Slika 3.13.



Slika 3.13. Atributi riječi „best“

Sljedeći korak je pokrenuti pretvaranje u FNOK zapis klikom na istoimenu gumb. Na ovaj način za svih 100 rečenica iz [90] generirani su automatski FNOK zapisi i uspoređeni s FNOK zapisima izrađenim ručno. Za one rečenice koje imaju identične

FNOK zapise onima izrađenim ručno smatrat će se da zadovoljavaju ovo testiranje. One rečenice kojima se FNOK razlikuje upisat će se u relacijsku bazu podataka i pokrenuti upiti i ukoliko se za rečenicu dobije minimalno jedan točan odgovor na postavljeno pitanje smatrat će se da je uspješno automatski prevedena.

Usporedbom je utvrđeno da 37 rečenica ima u potpunosti identične FNOK zapise izrađene ručno i uz pomoć aplikacije te za ove rečenice može se tvrditi da zadovoljavaju postavljene uvjete. 33 rečenice, iako im FNOK zapisi nisu bili u potpunosti identični, upisivanjem u relacijsku bazu i postavljanjem upita dalo je minimalno jedan točan odgovor na postavljena pitanja te se i za njih može smatrati da su uspješno automatski pretvoreni. Detaljnijom analizom i usporedbom ovih FNOK-ova uočeno je da je kod njih razlika u redosljedu riječi u zapisu što pokazuje fleksibilnost NOK metode u njenoj primjeni i cijelog QA sustava.

Za 30 rečenica uočene su, kod same usporedbe FNOK zapisa, veće razlike i pretpostavilo se da neće biti odgovarajući za trenutačnu strukturu QA sustava i zapisivanja u relacijsku bazu podataka pa i očekivano nisu davali točne odgovore na postavljena pitanja, a 7 rečenica se uopće nije moglo učitati u QA sustav. Značajno u ovom testiranju je da su za 93 rečenice FNOK zapisi bili prikladni za upis u relacijsku bazu.

Slika 3.14 prikazuje usporedbu dijela rečenica (od 61. do 75.) među kojima su vidljive rečenice iz sva tri prethodno opisana slučaja. Prvi stupac prikazuje rečenice na engleskom jeziku, drugi stupac njihove FNOK zapise dobivene ručno, a treći FNOK zapise dobivene uz pomoć aplikacije PSA – FNOK. U trećem stupcu identični FNOK zapisi dobiveni automatski označeni su zelenom bojom, a žutom bojom zapisi koji nisu identični, ali daju odgovore na postavljena pitanja. Neoznačeni dijelovi trećeg stupca (bijele boje) su rečenice za koje nisu dobiveni odgovori sustava na postavljena pitanja.

61.	The boy is very young.	is ("who?" boy ("art?" the), "what?" young ("how?" very))	is ("who?" boy ("art?" the), "what?" young ("how?" very))
62.	Peter has an office in London.	has ("who?" Peter, "what?" office ("art?" an), "where?" London ("where?" in))	has ("who?" Peter, "where?" London ("where?" in ("what?" office ("art?" an))))
63.	Rob has an office 200 meters from the sea in Dubrovnik.	has ("who?" Rob, "what?" office ("art?" an), "where?" sea ("how many?" meters ("how many?" 200), "where?" from, "art?" the), "where?" Dubrovnik ("where?" in))	has ("who?" Rob, "what?" Dubrovnik ("where?" in ("what?" sea ("art?" the ("where?" from ("what?" meters ("how_much?" 200 ("what?" office ( an))))))))
64.	Tom is quite a good boy.	is ("who?" Tom, "what?" boy ("what?" good ("how?" quite, "art?" a)))	is ("who?" Tom, "what?" boy ("what?" good ("art?" a ("how?" quite))))
65.	I work hard.	work ("who?" I, "how?" 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))	gave ("who?" Noa, "who?" friend ("art?" a ("who?" to ("what?" book ("what?" this))))
67.	Maya has some interesting books.	has ("who?" Maya, "what?" books ("how many?" some, "what?" interesting))	has ("who?" Maya, "what?" books ("what?" interesting ("how many?" some)))
68.	I often go to the cinema.	go ("who?" I, "when?" often, "where?" cinema ("where?" to, "art?" the))	go ("when?" often ("who?" I), "where?" cinema ("art?" the ("where?" to)))
69.	London is the capital of Great Britain.	is ("what?" London, "what?" capital ("art?" the) ("what?" Great Britain ("what?" of)))	is ("what?" London, "what?" Britain ("what?" Great ("where?" of ("what?" capital ("art?" the))))
70.	Mary is Lucy's friend.	is ("who?" Mary, "what?" friend ("whose?" Lucy's))	is ("who?" Mary, "what?" friend ("whose?" Lucy's))
71.	Father is a good man.	is ("who?" father, "what?" man ("art?" a, "what?" good))	is ("who?" Father, "what?" man ("what?" good ("art?" a)))
72.	Beauty is not forever.	is not ("what?" beauty, "what?" forever)	is ("what?" beauty, "how?" forever ("what?" not))
73.	Honesty is the best policy.	is ("what?" honesty, "what?" policy ("what?" best ("art?" the)))	is ("what?" honesty, "what?" policy ("what?" best ("art?" the)))
74.	Mary's father is old.	is ("who?" father ("whose?" Mary's), "what?" 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))	is ("what?" it, "what time?" o'clock ("what time?" six))

**Slika 3.14. Dio uspoređenih rečenica**

Zaključak prethodne analize uspešnost uporabe aplikacije PSA – FNOK za automatsko pretvaranje rečenica engleskog jezika u FNOK je sljedeći:

- za 37% rečenica FNOK je identičan onom izrađenom ručno
- 93% FNOK zapisa uspešno je upisano u QA sustav
- za 70% rečenica QA sustav je dao točan odgovor na minimalno jedno postavljeno pitanje.

Aplikacija PSA-FNOK primijenjena je na složene rečenice i to na 20 rečenica odabranih za inicijalno testiranje sustava opisano u poglavlju 7 (Privitak 2). Tablica 3.4. Usporedba ručno generiranih FNOK zapisa i generiranih automatski PSA-FNOK aplikacijom prikazuje rečenice na engleskom jeziku (TENG), formalizirane FNOK zapise izrađene ručno prema pravilima opisanim u poglavlju 4 te FNOK zapise koje je automatski generirala PSA-FNOK aplikacija.

**Tablica 3.4. Usporedba ručno generiranih FNOK zapisa i generiranih automatski PSA-FNOK aplikacijom**

R.B.	TENG	FNOK	FNOK (automatski)
1.	Tom and Paul talk about the solution.	talk ("who?" and ("who?" Tom, "who?" Paul), "what?" solution ("art?" the, "what?" about))	talk ("who?" Paul ( and ("who?" Tom)), "what?" solution ( the ("what?" about)))
2.	Peter trains basketball, volleyball and tennis.	trains ("who?" Peter, "what?" and ("what?" basketball, "what?" volleyball, "what?" tennis))	trains ("who?" Peter, "what?" tennis ( and ("what?" volleyball ( , ("what?" basketball))))))
3.	We were wet and tired.	were ("who?" we, "what?" and ("what?" wet, "what?" tired))	were ("who?" we, "what?" tired ( and ("what?" wet)))
4.	He drew a heart and coloured it red.	"what?" and (drew ("who?" he, "what?" heart ("art?" a)), coloured ("what?" it, "how?" red)))	and (drew ("who?" he ("what?" heart ( a)), coloured ("how?" red ("what?" it)))
5.	Lucija is from Zagreb or Rijeka.	is ("who?" Lucija, "where?" from ("where?" or ("where?" Zagreb, "where?" Rijeka)))	is ("who?" Lucija, "where?" Rijeka ( or ("where?" Zagreb ("where?" from)))
6.	You should eat more, or you'll make yourself ill.	"what?" or (should (eat ("who?" you, "what?" more)), will (make ("who?" you, "whom?" yourself, "what?" ill )))	or (eat ( should ("who?" you) , ("what?" more)), make ( will ("who?" you), "what?" ill ("whom?" yourself)))
7.	Sanja is happy because she loves her interesting hobby.	"why?" because (is ("who?" Sanja, "what?" happy), loves ("who?" she, "what?" hobby ("whose?" her, "what?" interesting)))	because (is ("who?" Sanja, "what?" happy), loves ("who?" she, "what?" hobby ("what?" interesting ("whose?" her)))
8.	I didn't go out because I had no money.	"why?" because (didn't (go ("who?" I, "where?" out)), had ("who?" I, "what?" money ("how_much?" no)))	because (did ("who?" I, "where?" out ("what?" go (? not))), had ("who?" I, "what?" money ("how?" no)))
9.	Vanessa is sleepy because she has been studying all night.	"why?" because (is ("who?" Vanessa, "what?" sleepy), has (been (studying ("who?" she, "when?" night ("what?" all))))))	because (is ("who?" Vanessa, "what?" sleepy), studying ( been ( has ("who?" she)), "what?" night ("what?" all)))
10.	We ran after him, but he escaped.	"what?" but (ran ("who?" we, "how?" after, "whom?" him ), escaped ("who?" he))	but (ran ("who?" we) , ("who?" him ("how?" after))), escaped ("who?" he))
11.	He felt nothing but contempt for her.	felt ("who?" he, "what?" nothing, "what?" but ("what?" contempt), "whom?" her ("whom?" for))	but (felt ("who?" he, "what?" nothing), "whom?" her ( for ("what?" contempt)))
12.	I loved maths when I was at school.	"when?" when (loved ("who?" I, "what?" math), was ("who?" I, "where?" school ("where?" at)))	when (loved ("who?" I, "what?" maths), was ("who?" I, "where?" school ("where?" at)))
13.	I had a great time when I went to the coast.	"when?" when (had ("who?" I, "what?" time ("what?" great ("art?" a))), went ("who?" I, "where?" coast ("art?" the ("where?" to)))	when (had ("who?" I, "what?" time ("what?" great ( a))), went ("who?" I, "what?" coast ( the ("where?" to)))
14.	I wonder if she noticed.	"what?" if (wonder ("who?" I), noticed ("who?" she))	if (wonder ("who?" I), noticed ("who?" she))
15.	You can go here if you get the point.	"what?" if (can (go ("who?" you, "where?" here)), get ("who?" you, "what?" point ("art?" the)))	if (go ( can ("who?" you), "where?" here), get ("who?" you, "what?" point ( the)))
16.	You will feel cold if you don't wear a coat.	"what?" if (will (feel ("who?" you, "what?" cold)), don't (wear ("who?" you, "what?" coat ("art?" a))))	if (feel ( will ("who?" you), "what?" cold), wear (? not ( do ("who?" you)), "what?" coat ( a)))
17.	Monika is glad that workers aren't losing their jobs.	"what?" that (is ("who?" Monika, "what?" glad, aren't (losing ("who?" workers, "what?" jobs ("whose?" their )))))	that (is ("who?" Monika, "what?" glad), are ("what?" workers, "what?" jobs ("whose?" their ( losing (? not))))))
18.	It was still painful, so I went to see a specialist.	"why?" so (was ("what?" it, "what?" still, "how?" painful), went ("who?" I, "what?" see ("what?" to), "who?" specialist ("art?" a)))	so (was ("what?" it) , ("how?" painful ("what?" still))), went ("who?" I, "what?" specialist ( a ( see ("how?" to))))
19.	You can ask for help whenever you need it.	"when?" whenever (can (ask ("who?" you, "what?" help ("what?" for))), need ("who?" you, "what?" it))	whenever (ask ( can ("who?" you), "what?" help ( for)), need ("who?" you, "what?" it))
20.	He minded the dog while I was away.	"when?" while (minded ("who?" he, "what?" dog ("art?" the)), was ("who?" I, "where?" away))	while (minded ("who?" he, "what?" dog ( the)), was ("who?" I, "where?" away))

Usporedba formaliziranih FNOK zapisa izrađenih ručno i dobivenih automatski uporabom PSA-FNOK aplikacije pokazala je da aplikacija ne dodjeljuje ulogu/pitanje veznicima kada su nalaze u ulozi procesnog čvora i postavlja veznik na vrh hijerarhije (npr. (*"who?" Paul ( and ("who?" Tom))*)) što je značajno u traženju odgovora na postavljena pitanja. Slijedom ove usporedbe zaključeno je da automatski dobiveni FNOK-ovi nisu u skladu s definiranim pravilima transformacije i algoritmima zapisivanja u bazu podataka BPNOK i traženja odgovora na pitanja te će se u ovom radu koristiti ručno izrađeni formalizirani FNOK zapisi.

## 4. Pravila transformacije složenih rečenica prirodnog jezika u Formalized Node of Knowledge (FNOK)

Postupak prevođenja rečenica prirodnog jezika u FNOK zapis temelji se na vrstama riječi koje je moguće odrediti pomoću rječnika.

Rečenica (engl. *sentence*) u engleskom jeziku, kao i u hrvatskom jeziku, je skupina riječi koja obično sadrži glagol, koja izražava misao u obliku izjave, pitanja, upute ili uzvika i kada je zapisana počinje velikim slovom [95, 100]. Osnovni dijelovi rečenice su subjekt, predikat (često, ali ne uvijek) objekt. Subjekt je obično imenica - riječ koja imenuje osobu, mjesto ili stvar. Predikat (glagol) obično slijedi subjekt i identificira radnju ili stanje postojanja. Objekt prima radnju i obično slijedi glagol [101].

U engleskom jeziku postoje četiri osnovne strukture rečenice [101]:

- Jednostavna rečenica (engl. *simple sentence*) – rečenica sa samo jednom nezavisnom rečenicom.
- Nezavisno složena rečenica (engl. *compound sentence*) - sadrži najmanje dvije nezavisne rečenice.
- Zavisno složena rečenica (engl. *complex sentence*) - sadrži nezavisnu rečenicu i najmanje jednu zavisnu rečenicu.
- Nezavisno – zavisna složena rečenica (engl. *compound-complex sentence*) rečenica koja sadrži dvije ili više nezavisnih rečenica i najmanje jednu zavisnu rečenicu.

U dosadašnjim istraživanjima proučavane su jednostavne rečenice prirodnog jezika, dok u ovom istraživanju proučavaju složene rečenice. Do sada primjenjivani temeljni algoritam transformacije rečenica zapisanih u prirodnom jeziku u formalizirani FNOK zapis, koji se primjenjuje na sve vrste riječi u rečenici, sastoji se od 8 koraka [1]:

1. Iz odlomka (paragrafa) uzima se rečenica.
2. Pronalazi se prva riječ u rečenici.
3. Određuje se vrsta riječi kojoj ta riječ pripada.
4. Traži se druga riječ u rečenici (i kasnije sve ostale riječi do kraja rečenice) te za svaku od njih određuje kojoj vrsti riječi pripadaju.
5. Kad je za sve riječi iz rečenice poznato kojoj vrsti riječi svaka od njih pripada, traže se riječi koje imaju homonime te se iz konteksta rečenice određuje vrsta riječi kojoj pripada.
6. Od svih riječi u rečenici traži se i odabire ona koja je po vrsti riječi glagol. Glagol se stavlja na prvo mjesto i otvara zagradu.
7. Nakon toga se traži veza među riječima, odnosno traži se veza između pronađenog glagola i riječi koje su s njime povezane (koje pripadaju tom

glagolu). To su sljedeće vrste riječi: imenice, zamjenice, prilozi i drugi glagoli ili glagolski oblici.

8. Ako rečenica ima više glagola, imenica i priloga, potrebno je odrediti točnu pripadnost.

S obzirom da se u dosadašnjim istraživanjima algoritam primjenjivao na jednostavne rečenice koje su sadržavale samo jedan glagol, 8. korak nije bio korišten. Proširivanjem promatranih rečenica na one koje sadrže veznike, čime se povećava složenost rečenica, potrebno je proširiti algoritam novim postupcima, odnosno izmijeniti postojeće korake. Kako veznici u rečenicama mogu povezivati dvije rečenice, odnosno dvije radnje, pa sadrže i dva glagola potrebno je u korake prevođenja uključiti i 8. korak. Analizom postojećeg algoritma i prijedloga prevođenja veznika u FNOK, posebno u složenim rečenicama povezanim veznikom, zaključeno je da je promjenu potrebno izvršiti od 6. koraka, kojeg je potrebno proširiti i u njega uključiti aktivnosti koje su u postojećim pravilima bile sadržane u 8. koraku. Korak 8 više ne postoji, a koraci 6 i 7 glase:

6. Od svih riječi u rečenici traži se veznik i ovisno o nađenom moguću su sljedeći slučajevi:
  - 6.1. Ako u rečenici nema veznika, glagol se stavlja na prvo mjesto i otvara se zagrada.
  - 6.2. Ako u rečenici postoji veznik, provjerava se što veznik povezuje i koliko je glagola u rečenici:
    - 6.2.1. Ako je u rečenici jedan glagol, postavlja se na prvo mjesto i otvara zagrada.
    - 6.2.2. Ako je u rečenici više glagola, na prvo mjesto se postavlja veznik i otvara se zagrada, a nakon nje glagoli i otvara se zagrada za svaki glagol.
7. Traži se veza među riječima, odnosno traži se veza između pronađenih glagola i riječi koje su s njime povezane (koje pripadaju tom glagolu). To su sljedeće vrste riječi: imenice, zamjenice, prilozi, veznici i drugi glagoli ili glagolski oblici.

#### **4.1. Pravila prevođenja prema vrsti riječi**

Svaka riječ u rečenici prirodnog jezika odgovara jednom čvoru. Iznimno, dvije ili više usko poveznih riječi, najčešće u nazivima, zapisuju se u jedan čvor i povezuju podvlakom \_ (na primjer *New\_York*). U dosadašnjim istraživanjima [85, 84, 86, 87, 1] su definirana pravila prevođenja za jednostavne rečenice koje sadrže jednu radnju odnosno glagol kao procesni čvorovi. Kako je procesni čvor čvor koji se nalazi na najvišoj razini hijerarhije, on povezuje čvorove i predstavlja odnose među čvorovima, aktivnosti, veze više čvorova, radnje, zbivanja i slično [1]. Proširivanjem opsega istraživanih rečenica na



rečenice koje sadrže veznike, procesni čvor može sadržavati i veznike kad su na najvišoj razini hijerarhije u rečenici. Razlika između glagola i veznika kao procesnih čvorova je u tome što uz veznik stoji pitanje, a uz glagol ne. Pitanje stoji i uz čvorove na nižim razinama hijerarhije, a uloga pitanja je da poveže čvor s čvorom iznad njega, odnosno čvor na nižoj razini hijerarhije je odgovor na to pitanje.

U pravilima prevođenja zadržava se definiranje odnosa između riječi, hijerarhijska ovisnost i redosljed čvorova kako je definirano u [1]. Do sada su definirana pravila transformacije rečenica prirodnog jezika u FNOK zapis za glagole [85], imenice [86], pridjeve [87], priloge i prijedloge [88], brojeve i većinu zamjenica [1].

## 4.2. Veznici

U hrvatskom jeziku veznici su nepromjenljive vrste riječi kojima se povezuju dvije riječi, dvije skupine riječi ili dvije rečenice. U funkciji veznika javljaju se i veznički izrazi koji se sastoje od više riječi [102]. S obzirom na njihovu funkciju razlikuju se razne vrste veznika koje se može svrstati u grupe [103]:

- rastavni – ili,
- sastavni – i, pa, te, ni, niti, pak,
- zaključni – stoga, zato, dakle,
- isključni – jedino, samo, samo što, osim, osim što, tek, tek što,
- vremenski – kako, čim, kada, dok, nakon što, pošto,
- posljednični – te, da,
- suprotni – nego, a, god, ali, već, dok, pa, kad ono, kad li, pak, a ono, kad to, a to, no,
- izrični – kako, e, gdje, da li, da,
- namjerni – neka, da, kako, e da, ne bih li,
- dopusni – ako i, makar, iako, premda,
- uzročni – što, gdje, jer, budući dan, kako,
- pogodbeni – kad li, da, kada, ako,
- načinski i poredbeni – kao, nego, kako, no, kao što.

Postoje i veznici koji se koriste samo u parovima. Koristi li se jedan veznik, mora se koristiti i njemu pripadajući par. To su veznici: ni/niti, ukoliko/utoliko, ne samo/nego i [103].

U engleskom jeziku, kao i u hrvatskom, veznici (*engl. conjunction*) su također riječi koje povezuju dvije riječi, dvije skupine riječi ili dvije rečenice, kao što su *and*, *or*, *but*, *because* i drugi. Za razliku od hrvatskog jezika gdje se po funkciji dijele u 13 skupina, u engleskom postoji osnovna podjela u dvije skupine [100, 104, 95]:

1. *Coordinating Conjunctions* - povezuju dijelove koji su iste gramatičke vrste kao što su riječi, fraze ili neovisne rečenice. Primjeri ove vrste veznika su: *and, or, so, but*.
2. *Subordinating Conjunctions* - povezuju neovisnu rečenicu (koja sadrži i subjekt i glagol i može djelovati kao potpuna rečenica) i zavisnu rečenicu (koja također sadrži subjekt i glagol, ali nije potpuna rečenica). Ovi veznici uvijek dolaze na početku zavisne rečenice, a u slučaju da se zavisna rečenica pojavljuje prva onda se one međusobno odvajaju zarezom. Primjeri ove vrste veznika su: *because, if, when, until, while, as* i dr.

Prevođenje rečenica prirodnog jezika koje sadrže veznike ovise o ulozi veznika u rečenici koja može biti:

- Veznici povezuju dijelove rečenice kao što su subjekt, objekt, atribut i apozicija.
- Veznici povezuju dvije rečenice.

#### 4.2.1. Veznici koji povezuju dijelove rečenice

U slučaju kada veznici povezuju dijelove unutar jedne rečenice glagol je procesni čvor i nalazi se na vrhu hijerarhije. Veznik najčešće spaja dvije imenice ili dva pridjeva. U hijerarhiji u FNOK-u veznik se smješta ispod glagola, ovisno o dijelovima rečenice koje spaja, i zapisuje iznad njih. Pridružuje mu se ista uloga (pitanje) kao i dijelovima rečenice koje povezuje. Hijerarhijski odnos veznika i riječi koje spajaju može se zapisati:

$C, X, Y \rightarrow C(X, Y)$ , gdje je  $C$  veznik (engl. *conjunction*), a  $X$  i  $Y$  riječi između kojih se nalazi.

Preciznije,

$C, X, Y \rightarrow \text{„role?“ } C(\text{„role?“ } X, \text{„role?“ } Y)$ , gdje je  $C$  veznik (engl. *conjunction*), a  $X$  i  $Y$  riječi između kojih se nalazi.

Tablica 4.1 prikazuje primjere transformacije rečenica u FNOK zapise u kojima veznik povezuje riječi.

Tablica 4.1. Primjeri pretvorbe rečenica prirodnog jezika u kojima veznici spajaju riječi u FNOK zapis

Rečenica u prirodnom jeziku (hrvatski ili engleski)	FNOK zapis rečenice
Ana <b>and</b> John play the violin.	play ("who?" and ("who?" Ana, "who?" John), "what?" violin ("art?" the))
Ana i Ivan sviraju violinu.	sviraju („tko?“ i („tko?“ Ana, „tko?“ Ivan), „što?“ violinu)
Lucija is from Zagreb <b>or</b> Rijeka.	is ("who?" Lucija, "where?" from ("where?" <b>or</b> ("where?" Zagreb, "where?" Rijeka)))
Lucija je iz Zagreba <b>ili</b> Rijeke.	je („tko?“ Lucija, „odakle?“ iz („odakle?“ <b>ili</b> („odakle?“ Zagreb, „odakle?“ Rijeka)))
Tom is happy <b>and</b> healthy boy.	is ("who?" Tom, "what?" boy ("what?" <b>and</b> ("what?" happy, "what?" healthy)))
Tom je sretan i zdrav dječak.	Je („tko?“ Tom, „što?“ dječak („kakav?“ i („kakav?“ sretan, „kakav?“ zdrav))

Na sličan način se prevodi rečenica u kojoj veznici povezuju fraze. Fraze se mogu sastojati od jedne ili više riječi. I u ovom slučaju veznik se stavlja u hijerarhiji iznad fraza koje spaja, što se može zapisati:

$C, F1, F2 \rightarrow C (F1, F2)$ , gdje je C veznik (engl. *conjunction*), F1 i F2 fraze između kojih se nalazi.

Preciznije,

$C, F1, F2 \rightarrow „role?“ C („role?“ F1, „role?“ F2)$ , gdje je C veznik (engl. *conjunction*), a F1 i F2 fraze između kojih se nalazi.

Npr:

Tom is **good student and healthy boy**.  $\rightarrow$  is ("who?" Tom, "what?" **and** ("what?" student („what?“ good), „what?“ boy ("what?" healthy)))

Odnosno na hrvatskom:

Tom je **dobar učenik i zdrav dječak**.  $\rightarrow$  je ("tko?" Tom, "što?" **and** ("što?" učenik („kakav?“ dobar), „što?“ dječak ("kakav?“ zdrav)))

U danim primjerima fraze koje spaja veznik *and* (i) su „good student“ („dobar učenik“) i „healthy boy“ („zdrav dječak“). Svaka fraza se sastoji od dvije riječi. Te riječi se prevode u FNOK zapis prema pravilima definiranim za vrste riječi kojima pripadaju, a u ovom slučaju to su imenice i pridjevi.

#### 4.2.2. Veznici koji povezuju dvije rečenice

Kada veznici povezuju dvije rečenice ili dvije radnje, veznik postaje procesni čvor i stavlja se na vrh hijerarhije. Nakon veznika, na sljedećoj razini hijerarhije su glagoli, a zatim slijede ostali dijelovi rečenice koji su po hijerarhiji ispod glagola. Prema dosadašnjim pravilima na najvišoj razini hijerarhije su bili glagoli i uz njih nije stajala uloga/pitanje. U proširenim pravilima na najvišoj razini hijerarhije u ulozi procesnog čvora, osim glagola, može biti i veznik. Uz veznik uvijek stoji uloga/pitanje, a uz glagol koji je na prvom mjestu formaliziranog zapisa nema uloge/pitanja. Glagole od veznika razlikujemo po tome što uz prvi, procesni čvor postoji uloga/pitanje ako je procesni čvor veznika, dok ako je on glagol uloge/pitanja nema. Ova distinkcija značajna je za definiranje algoritama transformacije FNOK i QFNOK zapisa u BPNOK kao i algoritama dobivanja odgovora na pitanja.

To se može zapisati:

$C, V1, V2, X1, X2 \rightarrow C (V1 (X1), V2(X2))$ , gdje je C veznik (engl. *conjunction*), V1 i V2 glagoli (engl. *verb*), X1 preostali dio/dijelovi prve rečenice koji su hijerarhijski ispod prvog glagola V1, a X2 preostali dio/dijelovi druge rečenice koji su hijerarhijski ispod drugog glagola V1.

Preciznije:

$C, V1, V2, X1, X2 \rightarrow „role?“ C (V1 („role?“ X1), V2(„role?“ X2))$ , gdje je C veznik (engl. *conjunction*), V1 i V2 glagoli (engl. *verb*), X1 preostali dijelovi prve rečenice koji su hijerarhijski ispod prvog glagola V1, a X2 preostali dijelovi druge rečenice koji su hijerarhijski ispod drugog glagola V1.

Tablica 4.2 prikazuje primjere rečenica prirodnog jezika u kojima veznici povezuju dvije rečenice i njihove formalizirane FNOK zapise dobivene korištenjem prethodno opisanih pravila.

Tablica 4.2. Primjeri pretvorbe rečenica prirodnog jezika u kojima veznici povezuju dvije rečenice

Rečenica u prirodnom jeziku (hrvatski ili engleski)	FNOK zapis rečenice
Marko is happy because he love his job.	"why?" because (is ("who?" Marko, "what?" happy), love ("who?" he, "what?" job ("whose?" his)))
Marko je sretan jer voli svoj posao.	„zašto?“ jer (je („tko?“ Marko, „što?“ sretan), voli („što?“ posao?“ („čiji?“ svoj)))
I loved math when I went to school.	"when?" when (loved ("who?" I, "what?" math), went ("who?" I, "where?" school ("where?" to)))
Ja sam volio matematiku kada sam išao u školu.	„kada?“ kada (sam (volio („tko?“ ja, „što?“ matematiku)), sam (išao („gdje?“ školu („gdje?“ u))))
He minded the dog while I was away.	"when?" while (minded ("who?" he, "what?" dog ("art?" the)), was ("who?" I, "where?" away))
On je brinuo o psu dok sam bio odsutan.	„kada?“ dok (je (brinuo, „tko?“ on, „kome?“ psu („kome?“ o)), sam (bio („što?“ odsutan)))

Uključivanjem veznika u FNOK formalizirane zapise, pogotovo u slučajevima kada povezuju dvije rečenice, značajno se proširio opseg istraživanja. Na ovaj način u rad se uključuju složene rečenice koje mogu biti zavisno složene i nezavisno složene, ovisno o vrsti veznika koji ih spaja. Osim sadržajnog proširenja veznici imaju utjecaj na strukturu formaliziranih FNOK zapisa, jer na vrhu hijerarhije i u ulozi procesnog čvora nije više samo glagol, već to može biti i veznik.

#### 4.3. Veznici u QFNOK-u

Kako bi se pitanja prirodnog jezika mogla zapisati u relacijsku bazu podataka i njih je, kao priprema za upisivanje u relacijsku bazu podataka, potrebno transformirati u formalizirani oblik QFNOK zapis. Kako bi se kod propitivanja moglo uspoređivati FNOK i QFNOK zapise, oni trebaju biti slične strukture jer to omogućuje dobivanje odgovora na postavljena pitanja. Za razliku od FNOK-a gdje uz svaki čvor osim glagola stoji uloga/pitanje, u QFNOK-u ulogu zapisujemo samo uz čvor koji se traži kao odgovor, a njega obilježavamo s "X". Ostalim čvorovima umjesto uloge piše se znak za podcrtavanja "\_". Osim navedenih simbola, u QFNOK-u se koristi i simbol "&" koji se stavlja ispred čvorova koji se kod propitivanja zanemaruju, a nastali su zbog načina tvorbe upitnih rečenica koji nastaju kroz tvorbu glagolskih vremena (*do/does, am/is/are*), *article (a, an, the)* i slično.

S obzirom da su rečenice koje se prevode u FNOK zapise proširene uključivanjem veznika, posljedično će se javiti i pitanja za te rečenice i prema istim pravilima će se veznici zapisivati i u QFNOK zapise. I u QFNOK-u javljaju se slučajevi u kojima veznici

povezuju riječi i fraze i slučajevi gdje veznici povezuju dvije radnje odnosno dvije rečenice, što je vidljivo iz primjera koje prikazuje Tablica 4.3.

*Tablica 4.3. Primjeri pitanja i pripadajućih QFNOK-ova koji sadrže veznike*

Upitna rečenica na prirodnom jeziku	QFNOK zapis pitanja
What are Tom <b>and</b> Paul talking about?	talking ( & are) ( _ <b>and</b> ( _ Paul, _ Tom), "what?" X ( _ about))
Who is happy <b>and</b> healthy boy?	is ("who?" X, _ boy ( _ <b>and</b> ( _ happy, _ healthy)))
What did I love <b>when</b> I was at school?	_ <b>when</b> (love ( & did) ( _ I, "what?" X), was ( _ I, _ school ( _ at)))

## 5. Meta modeli proširivanja rječnika u NOK metodi

Do sada korišteni rječnik u NOK metodi sastojao se od tri elementa, odnosno tri tablice (relacije) u relacijskoj bazi podataka: tablica koja sadrži popis pitanja, tablica koja sadrži popis svih riječi i tablica koja kroz povratnu vezu za promjenjive vrste riječi spaja njihovo značenje kroz dva različita oblika pojavljivanja te riječi čime se modelira hijerarhija kontekstnog odnosa između njih.

Osim toga uočene su mogućnosti dodatnih proširivanja rječnika kojima bi se omogućilo proširivanje kontekstnog znanja upisanog u bazu podataka i time povećala preciznost i točnost generiranih odgovora na postavljena pitanja. U nastavku ovog poglavlja opisani su sljedeći meta modeli:

- Meta model hijerarhije glagolskih vremena – za preciziranje odgovora s obzirom na vrijeme događanja: sada, u prošlosti ili u budućnosti.
- Meta model sinonima – za prepoznavanje i pronalaženje riječi koje imaju isto ili blisko značenje.
- Meta model homonima – za prepoznavanje riječi koje su istog oblika, ali različitog značenja.

### 5.1. Meta model hijerarhije glagolskih vremena

Glagoli i u hrvatskom i u engleskom jeziku su promjenjive vrste riječi kojima se izriče radnja, stanje ili zbivanje [102, 96]. U oba jezika glagoli imaju sljedeća svojstva [105, 106]:

1. Broj (jednina i množina),
2. Lice (tri jednine i tri množine),
3. Vrijeme (sadašnjost, prošlost i budućnost),
4. Vid (svršeni i nesvršeni),
5. Način (imperativ, indikativ, konjunktiv).

I u hrvatskom i u engleskom jeziku postoje glagolska vremena koja možemo grupirati u tri osnovne skupine: prošlo glagolsko vrijeme, sadašnje glagolsko vrijeme i buduće glagolsko vrijeme. U hrvatskom jeziku postoje četiri prošla vremena, jedno sadašnje i dva buduća, dok ih je u engleskom jeziku po četiri i za prošlo i za sadašnje te za buduće vrijeme (Tablica 5.1).

Tablica 5.1. Glagolska vremena u hrvatskom i engleskom jeziku

Vrijeme	Hrvatski jezik	Engleski jezik
<b>Prošlost</b>	Perfekt Pluskvamperfekt Aorist Imperfekt	Past Simple Past Continuous Past Perfect Past Perfect Continuous
<b>Sadašnjost</b>	Prezent	Present Simple Present Continuous Present Perfect Present Perfect Continuous
<b>Budućnost</b>	Futur I. Futur II.	Future Simple Future Continuous Future Perfect Future Perfect Continuous

Svojstvo glagolskog vremena može imati značajan utjecaj na točnost dobivenih odgovora na pitanja u QA sustavima i može utjecati na razliku u interpretaciji nečega što se dogodilo u prošlosti, događa se u sadašnjosti ili će se tek dogoditi u budućnosti. U pravilima transformacije rečenica prirodnog jezika u formalizirani FNOK zapis za glagole [1] za zapisivanje složenih glagolskih vremena razmatrane su dvije mogućnosti njihovog zapisivanje u hijerarhiju:

- Svi elementi glagolskog vremena (pomoćni i glavni glagol/i) nalaze se u hijerarhiji jedan ispod drugog tako da je glavni glagol na najnižoj razini hijerarhije.
- Ili obrnuto, svi elementi glagolskog vremena (pomoćni i glavni glagol/i) nalaze se u hijerarhiji jedan ispod drugog, ali tako da je glavni glagol na najvišoj razini hijerarhije.

Na primjer za rečenicu

„When the new park opens, I will go there every day.“

- za prvi način zapisivanja formalizirani FNOK zapis glasi:

*"when?" when (opens ("what?" park ("what?" new ("art?" the))),*  
**will (go ( "who?" I, "where?" there, "when?" day ("what?" every))))**

- a obrnuti redosljed hijerarhije u formaliziranom FNOK zapisu bi glasio:

*"when?" when (opens ("what?" park ("what?" new ("art?" the))),*  
**go (will ( "who?" I, "where?" there, "when?" day ("what?" every))))**

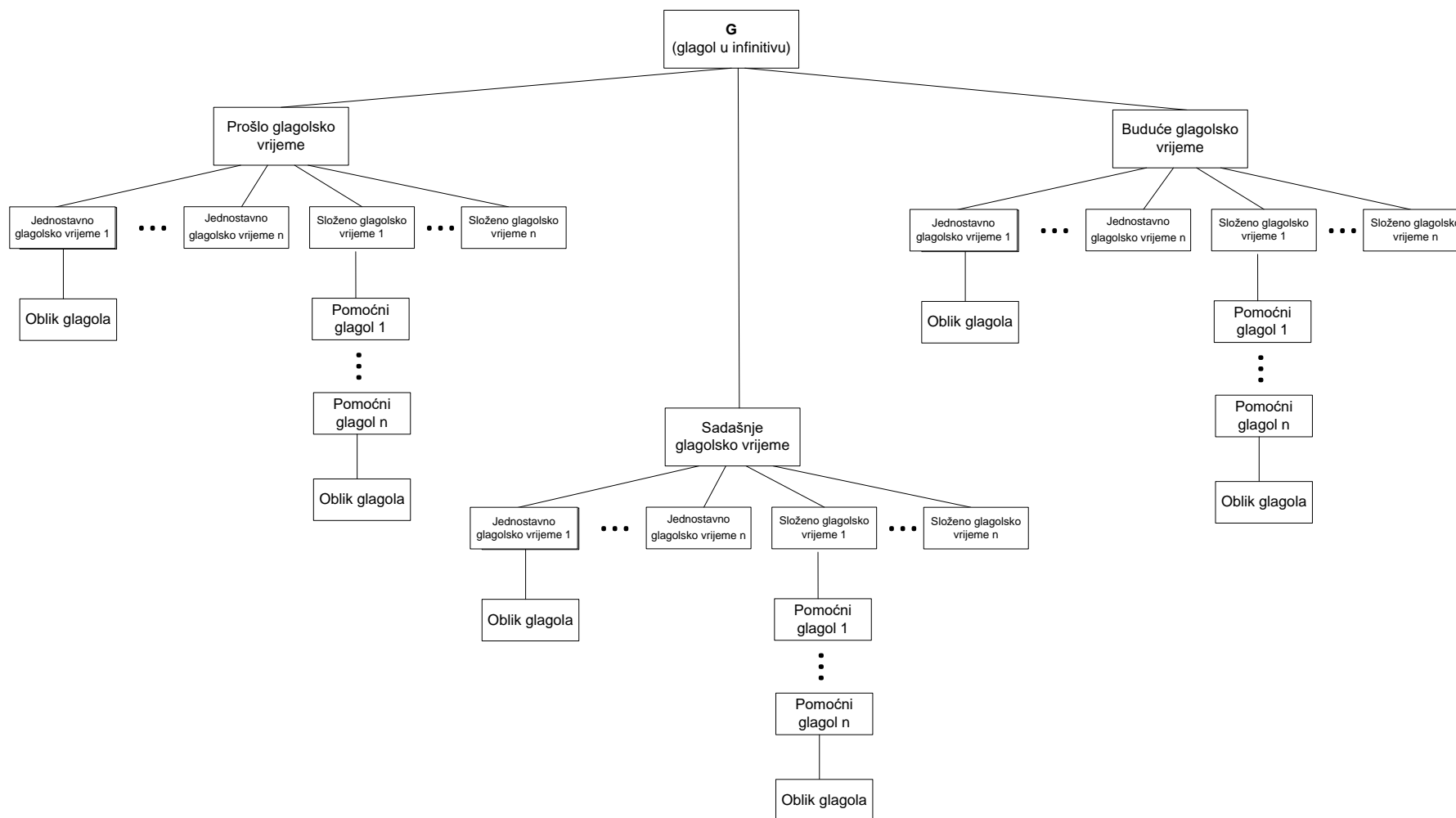
Analizom ova dva pristupa utvrđeno je da su oba navedena načina zapisivanja prihvatljiva za implementaciju i nemaju značajnih nedostataka. Kod implementacije je



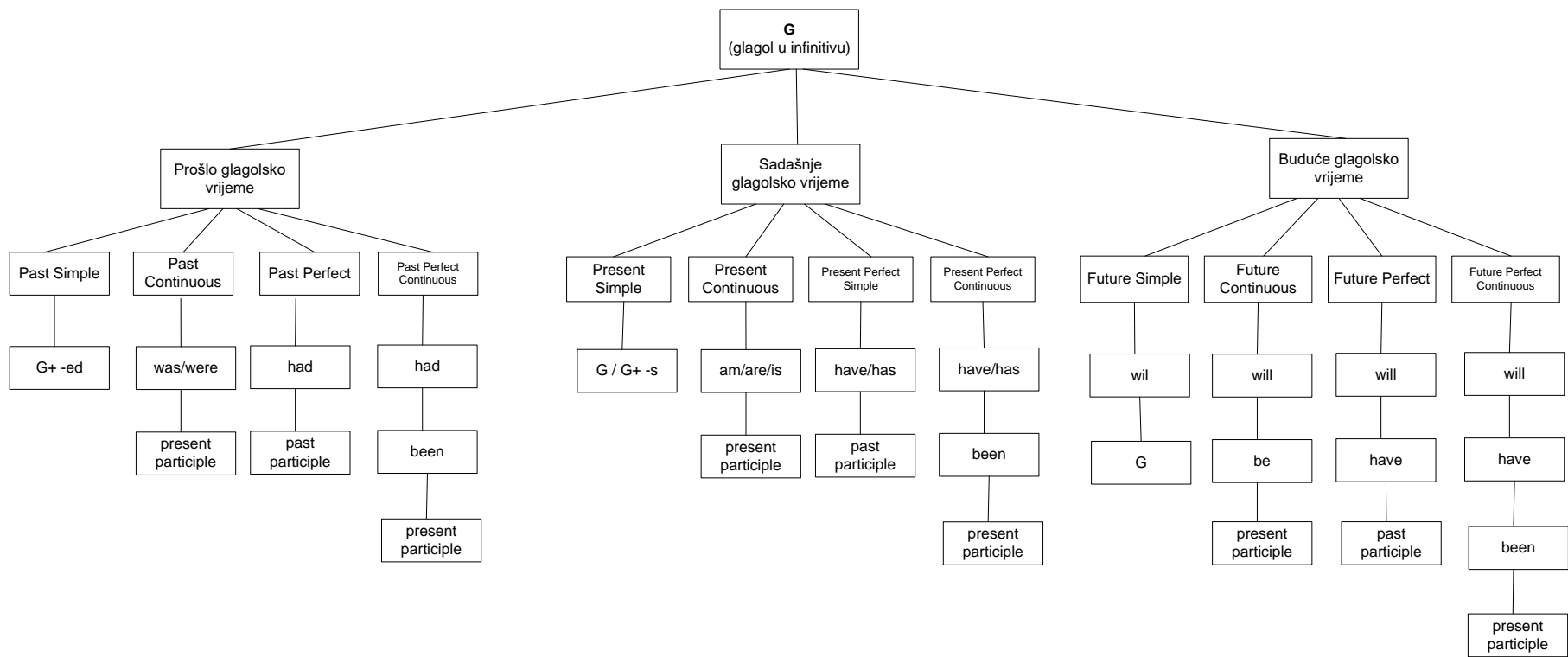
jedino važno biti dosljedan u primijenjenom načinu transformacije glagolskih vremena u formalizirane FNOK i QFNOK zapise. Daljnjom analizom mogućnosti implementacije i utjecaja na algoritme traženja odgovora utvrđeno je da je učinkovitije implementirati prvi prijedlog, u kojem je glavni glagol na najnižoj razini. Prednost ovog prijedloga je u tome što nakon glavnog glagola, koji je na najnižoj razini u odnosu na pomoćne glagole, na sljedećoj razini se nalaze ostali dijelovi rečenice povezani s glavnim glagolom, razinu za razinom, što skraćuje proces traženja odgovora, dok je uloga pomoćnih glagola samo u identifikaciji glagolskog vremena.

U skladu s odabranom hijerarhijom, u kojoj je glavni glagol na najnižoj razini, izrađen je meta model hijerarhije glagolskih vremena (Slika 5.1) za transformaciju formaliziranih zapisa u relacijsku bazu podataka u kojem je predloženo da se glagol u infinitivu (oblik zapisan u rječniku) povezuje hijerarhijski sa svim svojim oblicima u različitim glagolski vremenima. U hijerarhiji na razini ispod samog glagola je temeljna podjela glagolskih vremena na prošlo, sadašnje i buduće koja je ista i za hrvatski i engleski jezik, ali i za ostale prirodne jezike. Na nižoj razini nalaze se glagolska vremena povezana s vremenom kojem pripadaju. Ispod svakog glagolskog vremena slijede razine, jedna ispod druge, a ovisno o načinu tvorbe pojedinih glagolski vremena. Ako je glagolsko vrijeme složeno i tvori se korištenjem pomoćnih glagola, svaki pojedini dio zapisuje se na svojoj razini, od pomoćnih glagola do na najnižoj razini oblika glavnog glagola. Na taj način omogućeno je povezivanje infinitiva glagola sa svim glagolskim oblicima i vremenima, što će prilikom implementacije omogućiti osjetljivost propitivanja i s obzirom na vremensku odrednicu (prošlost, sadašnjost ili budućnost).

Slika 5.2 prikazuje primjenu ovog meta modela na glagolska vremena u engleskom jeziku na kojoj je uključeno svih 12 glagolskih vremena od kojih su tri glagolska vremena jednostavna i u njima se ne koriste pomoćni glagoli, pet glagolskih vremena uz glavni glagol imaju po jedan pomoćni glagol, pet ih ima dva pomoćna glagola, a jedno glagolsko vrijeme, *Future Perfect Continuous*, se sastoji od tri pomoćna i glavnog glagola. Na slici je vidljiva hijerarhija između pomoćnih i glavnog glagola koji je uvijek na najnižoj razini dok se pomoćni glagoli hijerarhijski smještaju jedan ispod drugog prema redoslijedu tvorbe glagolskog vremena.



**Slika 5.1. Meta model hijerarhije glagolskih vremena**



**Slika 5.2. Primjena meta modela hijerarhije glagolskih vremena na glagolska vremena u engleskom jeziku**

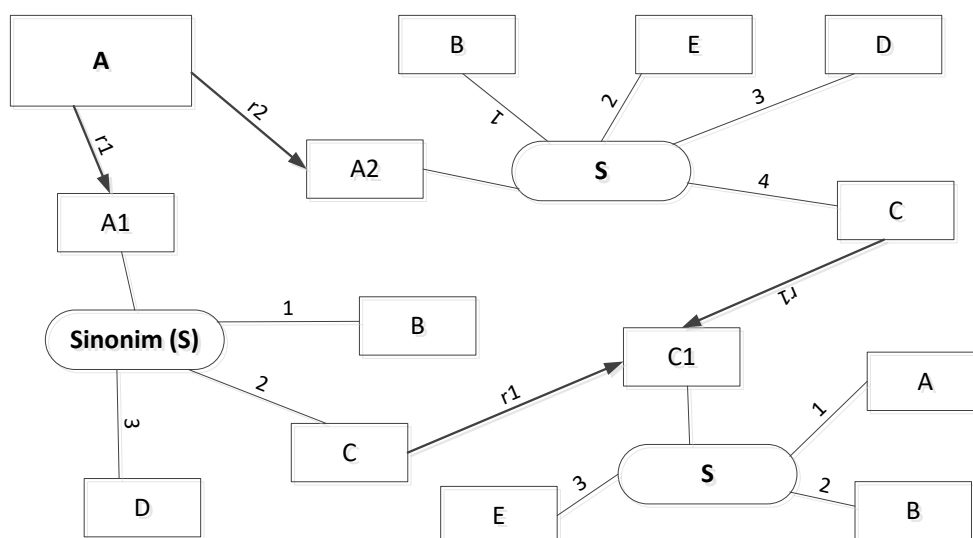
## 5.2. Meta modeli sinonima i homonima

U obradi prirodnog jezika javlja se problem sinonima i homonima, pa tako i u NOK metodi. Homonimi i sinonimi se smiju upotrebljavati [16, 76] ali ih treba povezati s originalom na kontekstnoj razini znanja. Zapisivanje pojmova mora biti nedvosmisleno, a problem nastaje ako se u rečenici koja se modelira pojavljuju homonimi, riječi koje imaju više značenja, i sinonimi, riječi koje imaju isto značenje. Taj problem moguće je riješiti izradom mreže znanja svih riječi, odnosno rječnika, u kojoj bi se nalazili svi pojmovi i njihovi odnosi, bilo kao homonimi ili sinonimi. Ovaj problem razmatran je u [107] i dani su meta modeli rješenja ovih problema na razini rječnika.

Sinonim [108, 100] je lingvistički riječ koja se glasovnim sastavom razlikuje od druge s kojom ima:

1. isto značenje (na primjer tisuća/hiljada, kiša/dažd), pravi sinonim; istoznačnica,
2. blisko značenje (na primjer kuća/dom, obitelj/porodica), nepravi sinonim; bliskoznačnica, sličnoznačnica.

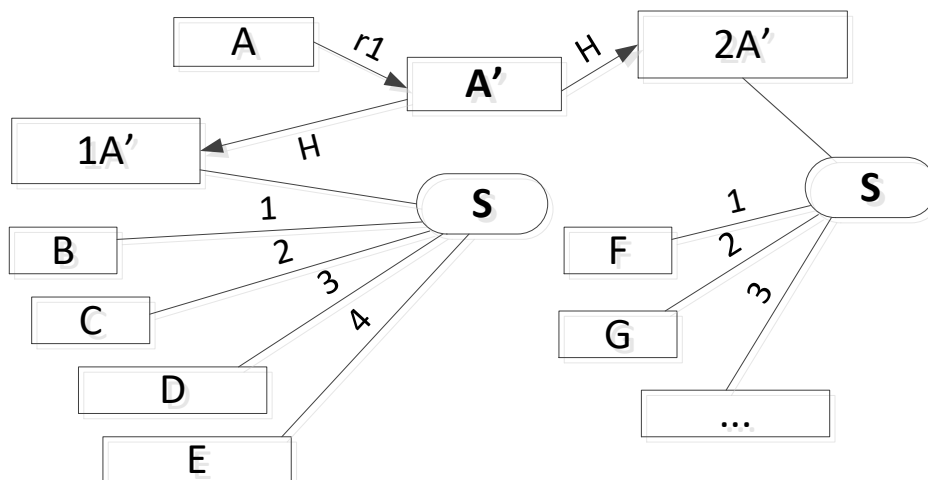
Model znanja treba sadržavati znanja iz različitih rječnika, tako neki opći pojam "A" možemo predstaviti kao A1 u rječniku "r1" ali i kao "A2" u rječniku "r2". U različitim rječnicima jedan pojam može imati iste ali i različite sinonime. Na primjer, pojam A1 u jednom rječniku ima sinonime B, C, D, a u drugom isti pojam A2 ima sinonime ovim redom B, E, D i C. Različite sinonime jednog pojma, povezujemo preko procesnog čvora „Sinonim“, skraćeno "S". Objašnjenja pojma u različitim rječnicima povezujemo kontekstnom vezom između njih koju imenujemo nazivom rječnika. To znači da opći pojam A je naveden u rječniku r1 kao A1. Vezu procesnog čvora sinonim označimo rednim brojem objašnjenja u rječniku. Svaki sinonim (na primjer C) je istovremeno i pojam u tom rječniku (na primjer C1) koji može imati definirane svoje sinonime. Na taj način razvija se mreža povezanih pojmova. Slika 5.3 prikazuje opisani meta model sinonima.



**Slika 5.3. Meta model sinonima prikazan NOK metodom**

Homonim je lingvistički riječ ista oblika, ali različita značenja spram neke druge riječi; istozvučnica [109, 108].

Postavlja se pitanje kako pojam homonima prikazati metodom NOK, hoće li to biti procesni čvor, kao što je korišteno za sinonim, ili običan čvor. Prijedlog je da se pojam homonima u rječnicima modelira konceptom kontekstne veze u NOK metodi. To znači da je neki pojam protumačen nadpojmom, odnosno pripada u grupu nadpojma. Pretpostavka na kojoj leži ovaj koncept za modeliranje znanja je da se dio znanja može prikazati i organizirati u čvorove među kojima postoji odnos: apstraktni pojam – konkretni pojam. Slika 5.4 prikazuje meta model homonima izrađen metodom NOK. Neki opći pojam A može u nekom rječniku biti prikazan kao A'. Imena homonima se ne navode u rječnicima već se navodi samo broj homonima kao 1., 2.a, 2.b i slično. Uzmimo da pojam A' u odabranom rječniku ima dva značenja, dva homonima, i to 1A' i 2A'. Homonim se na DNOK prikazuje konceptom kontekstne veze "H". Homonim se dalje može opisivati pomoću svojih sinonima (prikazano procesnim čvorom "S"). Svako značenje pojma objašnjava se novim pojmovima (B, C, ...) koji imaju konkretno značenje i sinonimi su za taj pojam.



Slika 5.4. Meta model homonima

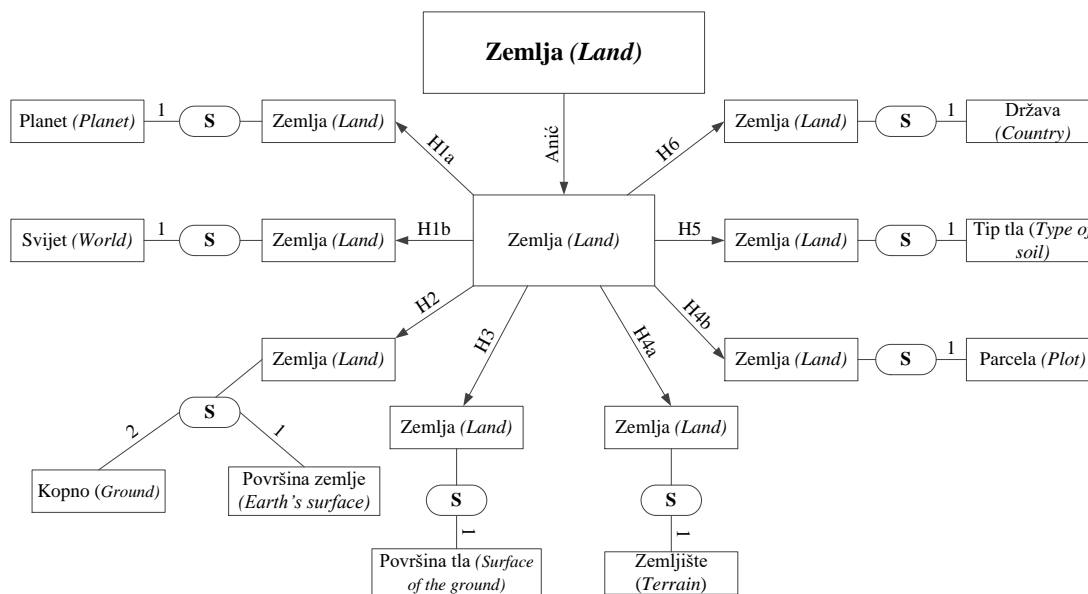
Primjena ovih meta modela prikazana je na primjeru riječ “zemlja” iz Rječnika hrvatskog jezika Vladimira Anića [110] danom na hrvatskom i direktnim prijevodom na engleski jezik (Tablica 5.2).

Tablica 5.2. Značenje riječi zemlja [110]

Hrvatski	Prijevod na engleski
<p>“zèmlja ž N mn zemlje, G zemáljā</p> <p>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</p> <p>2. površina Zemlje, kopno, suho, opr. voda</p> <p>3. površina tla, gornji sloj Zemljine kore [nad zemljom i pod zemljom]</p> <p>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]</p> <p>5. tip tla koje se obrađuje, na kojem se gradi itd. [pjeskovita zemlja; raskvašena zemlja]</p> <p>6. pov. državno-upravna jedinica u smislu državne, administrativne i političke podjele Habsburške Monarhije i Austro-Ugarske Monarhije”</p>	<p>“land n. pl lands</p> <p>1.a. (Earth) planet where we live, the third inner planet of the Solar System with one natural satellite (Moon) b. place, habitation of humans and place of human activity, world</p> <p>2. the solid part of the earth's surface, ground, mainland (opp. water, air, sea)</p> <p>3. the surface of the ground, the top layer of the Earth's crust [above and under the earth's surface]</p> <p>4.a. a terrain, field, soil as a source of goods and food</p> <p>b. (also in pl.) plot, area that can be cultivated, exploited; land as assets, property, estates [private land, collective land, state land]</p> <p>5. type of soil which is cultivated, built on, etc. [sandy land, sodden land]</p> <p>6. his. a country, a state, a nation with its own government, occupying a particular territory</p>

Slika 5.5 prikazuje model riječi „zemlja“ i njenih sinonima i homonima. Riječ “zemlja” prema gornjoj definiciji ima šest različitih značenja, od kojih prvo i četvrto ima dva podznačenja koja su međusobno slična, i to su sve njezini homonimi. U modelu je riječ zemlja povezana kontekstnim vezama označenim slovom H i brojem s pripadajućim

objašnjenjem iz rječnika (H1a do H6) koje se dalje povezuje preko procesnog čvora S sa sinonimima (prema objašnjenju H1a zemlja je povezana procesnim čvorom S sa sinonimom planet). Iz direktnog prijevoda „zemljinih“ sinonima i homonima vidljivo je da direktno preslikavanje sinonima između dva različita prirodna jezika nije moguće. Unatoč tome, proširenjem modela sinonima i homonima na značenje na drugim jezicima omogućilo bi se traženje odgovora na pitanja neovisno o jeziku na kojem je pitanje postavljeno i na kojem je jeziku rečenica u kojoj se nalazi odgovor.



**Slika 5.5. Model riječi "zemlja" (Land)**

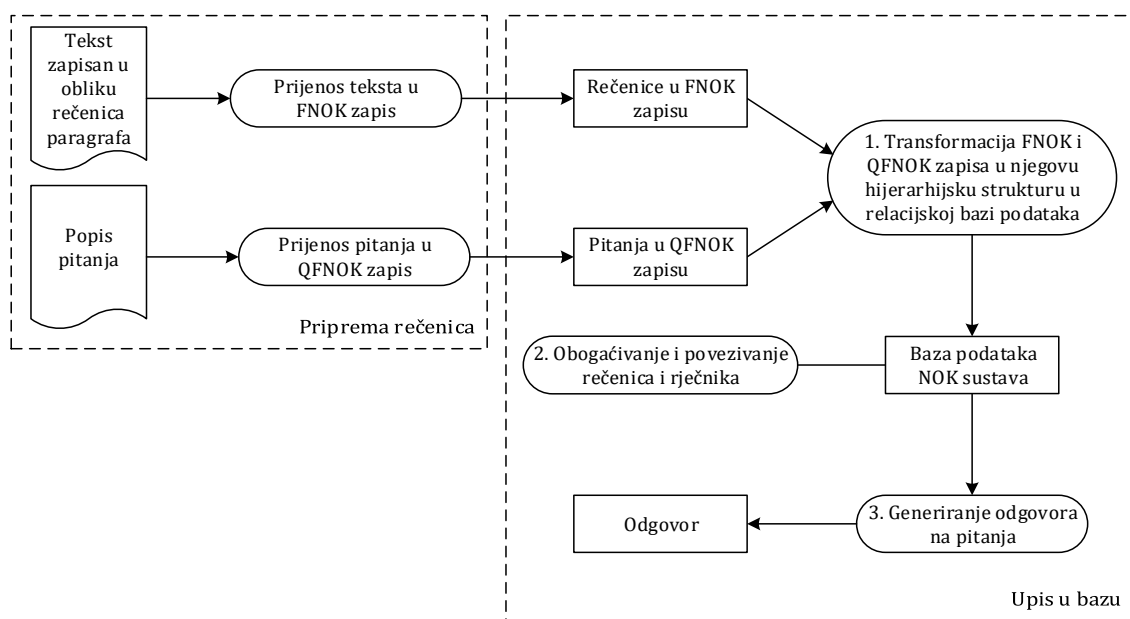
Predloženom metodom moguće je izraditi povezanu mrežu svih pojmova iz jednog rječnika, pri čemu se tumačenje jednih pojmova daje kroz druge pojmove kroz sinonimnu ili homonimnu vezu. Svaki pojedini rječnik predstavlja zasebnu DNOK mrežu. Moguće je povezati različite rječnike tako da se povežu različiti dijagrami DNOK. Spajanje se jednostavno ostvaruje preko općeg rječnika koji ima sve riječi svih rječnika bez definicija značenja. Pojmovi općeg rječnika, te pojmovi iz raznih rječnika predstavljaju najvišu razinu jezičnog znanja.

U ovom radu homonimi nisu razmatrani i preostaje ih implementirati u budućim istraživanjima, a u sljedećem poglavlju je dan opis implementacije sinonima u relacijskoj bazi podataka BPNOK.

## 6. Proširivanje sustava temeljenog na NOK metodi

U [1] analizirani su načini na koji tekstualno znanje upisati u relacijsku bazu podataka i prikazani su modeli podataka i procesa koji opisuju sustav za transformaciju. Ovi modeli temeljeni su na procesu transformacije jednostavnih rečenica koje sadrže jednu radnju/glagol i primijenjeni za transformaciju hijerarhijski složenih FNOK zapisa u skup nezavisnih iskaza među kojima nema direktnih zavisnosti. Proširivanje područja primjene sustava za transformaciju na složene rečenice koje sadrže veznike i dvije radnje/glagola zahtijevalo je i izmjene i nadogradnju prethodno izrađenih modela.

Za izradu modela podataka korištena je metodologija za razvoj informacijskih sustava – MIRIS [111] koja se sastoji od skupa metoda i uputa čiji je ukupni cilj projektirati i izgraditi informacijski sustav, odnosno njegov model. Pored elemenata metodologije MIRIS koriste se i koncepti NOK metode [12]. Slika 6.1 prikazuje model procesa sustava koji omogućuje upisivanje rečenica prirodnog jezika u bazu podataka.



Slika 6.1. Model procesa sustava transformacije teksta u bazu podataka [1]

Kako bi se tekst pripremio za upisivanje u relacijsku bazu podataka potrebno ga je pripremiti, odnosno rečenice prirodnog jezika zapisati u FNOK, a pitanja odnosno upitne rečenice u QFNOK zapis. Ovo pretvaranje provodi se prema pravilima opisanim u [1] i njihovim proširivanjem opisanim u poglavljima 4.2 i 4.3.

Ovaj sustav, koji se sastoji od tri procesa, omogućuje učitavanje tekstualnog znanja u relacijsku bazu podataka i generiranje odgovora na pitanja. Prvi proces priprema tekst za transformaciju. U drugom se tekst transformira uz pomoć NOK metode, obogaćuje i



povezuje s rječnikom i upisuje u relacijsku bazu podataka da bi se u trećem postavljanjem upita iz relacijske baze generirali odgovori na postavljena pitanja.

## 6.1. Proširivanje modela podataka BPNOK-a i relacijske sheme baze podataka

S obzirom na proširivanje područja istraživanja na određene oblike složenih rečenica definiranih u poglavlju 1 potrebno je proširiti meta modele podataka FNOK i QFNOK formalizama i na temelju njih definirati relacijsku shemu baze podataka. Za modeliranje podataka koristi se metoda entiteta i veza metodologije MIRIS.

Meta modeli strukture i sadržaja FNOK i QFNOK zapisa primijenjeni na jednostavne rečenice hrvatskog i engleskog jezika opisani su u [1], u kojem je dana i sljedeća relacijska shema baze podataka BPNOK:

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

Ovaj model implementiran je u Oracle relacijsku bazu podatka sa SQL procedurama i time prilagođen potrebama izgradnje programskog sustava.

Slika 6.2 prikazuje prošireni model podataka zapisa FNOK i QFNOK formalizama u relacijsku bazu podataka BPNOK.



(tip **B**) opisane u poglavlju 4.2. Određivanje i zapisivanje atributa tip veznika (CONJ\_TYPE) u bazu podataka potreban je zbog definiranja procesa traženja odgovora na pitanja koja se postavljaju na sam veznik. U slučaju *Coordinating Conjunctions* veznika, koji povezuju dvije nezavisne rečenice, pitanja na veznik nisu indicirana jer nema uzročno – posljedične veze između dvije rečenice i radnje u njima odvijaju se neovisno jedna o drugoj. Ali kada su u rečenici *Subordinating Conjunctions* onda to znači da veznici (na primjer *because*) povezuju zavisno složene rečenice i može se postaviti pitanje (uz veznik *because* veže se pitanje *why?*) koje uključuje jednu rečenicu, a odgovor je druga rečenica. Na primjer, na pitanje

„*Why is Marko happy?*“

odgovor se može naći u rečenici

„*Marko is happy because he loves his job.*“

i glasi

„*because he loves his job.*“

U ovom slučaju pitanje se postavlja na sam veznik *because* što je definirano u samom FNOK formalizmu ove rečenice:

*"why?" because (is ("who?" Marko, "what?" happy), loves ("who?" he, "what?" job ("whose?" his)))*

Iako se veznik *or* ubraja u *Coordinating Conjunctions* i po prethodno opisanoj notifikaciji to je veznik tipa A, ali prilikom traženja odgovora na pitanja postoje slučajevi koji se razlikuju od ostalih veznika ovog tipa i unificirani algoritam ne nalazi odgovore. Zbog specifičnosti traženja odgovora na pitanja kada je u pitanju veznik *or* uveden je i treći tip veznika **O**. Algoritam traženja odgovora ovisno o tipu veznika opisan je u poglavlju 6.3.

KB\_CONJ (Dictionary conjunctions) i KB\_WORD (Dictionary) povezani su tipom veze *Is*, tako da po gornjim granicama jedna riječ odgovara samo jednoj riječi, a donje granice su nule (0), jer nisu sve riječi iz rječnika veznici.

Drugi novi tip entiteta je KB\_SYNONYM (Dictionary synonym) kojim je omogućeno definiranje sinonima u bazi. Meta modeli ugradnje sinonima i homonima dani su poglavlju 5.2. U ovom istraživanju u bazu podataka implementirani su sinonimi što je opisano u sljedećem poglavlju.

### 6.1.1. Implementacija rješenja problema sinonima

Rječnik (*Dictionary*), s ciljem rješavanja problema sinonima, proširen je s još jednim tipom entiteta KB\_SYNONYM (*Dictionary Synonym*) i ovaj tip entiteta sadrži identifikatore KB\_WORD riječi koje su međusobni sinonimi. Atributi ovog entiteta su: identifikator sinonima (SYN\_ID), identifikator prve riječi (WORD\_ID1) i identifikator druge riječi koja je sinonim prvoj (WORD\_ID2). KB\_SYNONYM je povezan s tipom entiteta KB\_WORD. Postoje dvije veze: KB\_SYN\_WRD\_ID1 i KB\_SYN\_WRD\_ID2. Kroz te veze prvi čvor (WORD\_ID1) povezuje se s drugim čvorom. Identifikacijska veza KB\_SYN\_WRD\_ID1 dodjeljuje pojedinom čvoru u rečenici identifikator te riječi u rječniku. Identifikacijska veza KB\_SYN\_WRD\_ID2 dodjeljuje pojedinom čvoru u rečenici identifikator riječi njegovog sinonima. Ovdje ne vrijedi bijektivno preslikavanje i ako je jedna riječ drugoj sinonim, ne mora nužno vrijediti i obrnuto što je vidljivo već u samoj definiciji riječi sinonim u [110, 100] prema kojoj sinonimi mogu biti riječi koje imaju isto, ali i blisko značenje.

To je također vidljivo i na primjeru dvije rečenice (Tablica 6.1) koje koriste glagole *talk* i *discuss* koji nemaju u potpunosti isto značenje. Netko tko priča (*talk*) ne mora istovremeno i diskutirati (*discuss*), ali osoba koja diskutira (*discuss*) sigurno i priča (*talk*) pa ovdje sinonim vrijedi u jednom smjeru, odnosno glagolu *discuss* je sinonim glagol *talk*, ali ne i obrnuto. Ovi glagoli povezani su u tablici KB\_SYNONYM na način da je identifikator riječi/glagola *talk* iz rječnika KB\_WORD upisan na mjesto atributa WORD\_ID1, a identifikator riječi/glagola *discuss* na mjesto atributa WORD\_ID2.

Tablica 6.1. Primjeri rečenica koje koriste glagole "talk" i "discuss"

TENG	FNOK
<i>Lucija and Ana discuss about the weather.</i>	<i>discuss ("who?" and ("who?" Lucija, "who?" Ana), "what?" weather ("art?" the, "what?" about))</i>
<i>Tom and Paul talk about the solution.</i>	<i>talk ("who?" and ("who?" Tom, "who?" Paul), "what?" solution ("art?" the, "what?" about))</i>

Na postavljeno pitanje „*Who is talking?*“ očekivani i dobiveni odgovori su „*Lucija and Ana*“ i „*Tom and Paul*“ dok je za pitanje „*Who is discussing?*“ očekivan i dobiven samo jedan odgovor, odnosno „*Lucija and Ana*“.

Prethodni primjer prikaz je slučaja kada sinonim nije istoznačnica. U slučaju da su riječi istoznačnice potrebno je za svaku upisati u tablicu KB\_SYNONIM upisati identifikator riječi s kojom je povezana.

### 6.1.2. Proširivanje relacijske sheme baze podataka

Proširivanje modela (

Slika 6.2) dovelo je i do proširivanja relacijske sheme baze podataka koja primjenom pravila za prevođenje iz DEV-a sada glasi:

```
NOK_PARAGRAPH (PRG, NAME)
KB_QUEST (QUEST_ID, QUEST)
KB_DICT (DICT_ID, DICT, DICT_PAR)
KB_WORD (WORD_ID, WORD, DICT_ID)
KB_CONJ (CONJ_ID, CONJ, CONJ_TYPE, WORD_ID)
KB_SYNONYM (SYN_ID, WORD_ID1, WORD_ID2)
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,
PAR_WRD_ID, NOD_WRD_ID, NOD_QST_ID)
KB_ANSWER (ANSWER_ID, PRGQ, ORDQ, QTENG, ANSWER, PRGS, ORDS, TENG,
QFNOK, FNOK)
```

Ljubičastom bojom označene su tablice KB\_CONJ i KB\_SYNONYM koje predstavljaju proširivanje u odnosu na relacijsku shemu BPNOK definiranu u [1].

### 6.2. Algoritmi transformacije FNOK i QFNOK zapisa u BPNOK

U ovom poglavlju opisana su proširenja i izmjene algoritama transformacije rečenica zapisanih u formaliziranim FNOK i QFNOK zapisima koji se prema NOK metodi zapisuju u relacijsku bazu podataka. Za implementaciju relacijske baze podataka i dalje se koristi Oracle relacijska baza, ali za potrebe ovog istraživanja nove funkcije su izrađene korištenjem PHP-a kojim je izrađeno i korisničko sučelje za jednostavnije i preglednije testiranje (Slika 6.3).

## Odgovori na pitanja

**Pitanje**

Who is glad? Odaberi

---

**Formalizirani zapis**  
is ("who?" X, \_ glad)

Answered: 0.58943200111389

**Hijerarhijski zapis pitanja**

Node ID	Node	Question	Parent ID
1	is	-	0
2	X	who	1
3	glad	-	1

**Obrađeno pitanje**

ID	PAR_WRD_ID	NOD_QST_ID	NOD_WRD_ID
1028	149 is is are be was were am would been	3 who	579 X X
1029	149 is is are be was were am would been	-1 -	467 glad glad happy

✓ [99-35] Marko is happy because he love his job. ^

"why?" because (is ("who?" Marko, "what?" happy), love ("who?" he, "what?" job ("whose?" his)))

**ODGOVOR:** Marko

PARENT	QUESTION	NODE
[1] because	-	[2] is
[2] is	who	[3] Marko
[2] is	what	[4] happy
[1] because	-	[5] love
[5] love	who	[6] he
[5] love	what	[7] job
[7] job	whose	[8] his

PARENT	QUESTION	NODE
is	who	X
is	-	glad

Slika 6.3. Prikaz odgovora na pitanja u aplikaciji NOK

Korisničko sučelje dano je u PHP skripti *answer.php* (Privitak 1). Korisničko sučelje prikazuje hijerarhijski zapis pitanja i rečenica u kojima je pronađen odgovor i svih ostalih rečenica iz promatranog paragrafa te hijerarhijski prikaz njihovih čvorova. Sučelje također omogućuje korisniku odabrati željeno pitanje u padajućem izborniku koji prikazuje sva pitanja upisana u bazu vezana uz taj paragraf. Ispod formaliziranog zapisa pitanja prikazano je vrijeme u sekundama utrošeno na traženje odgovora i njegove vrijednosti se kreću oko 0,5 sekundi iz čega je vidljivo da je pretraživanje brzo.

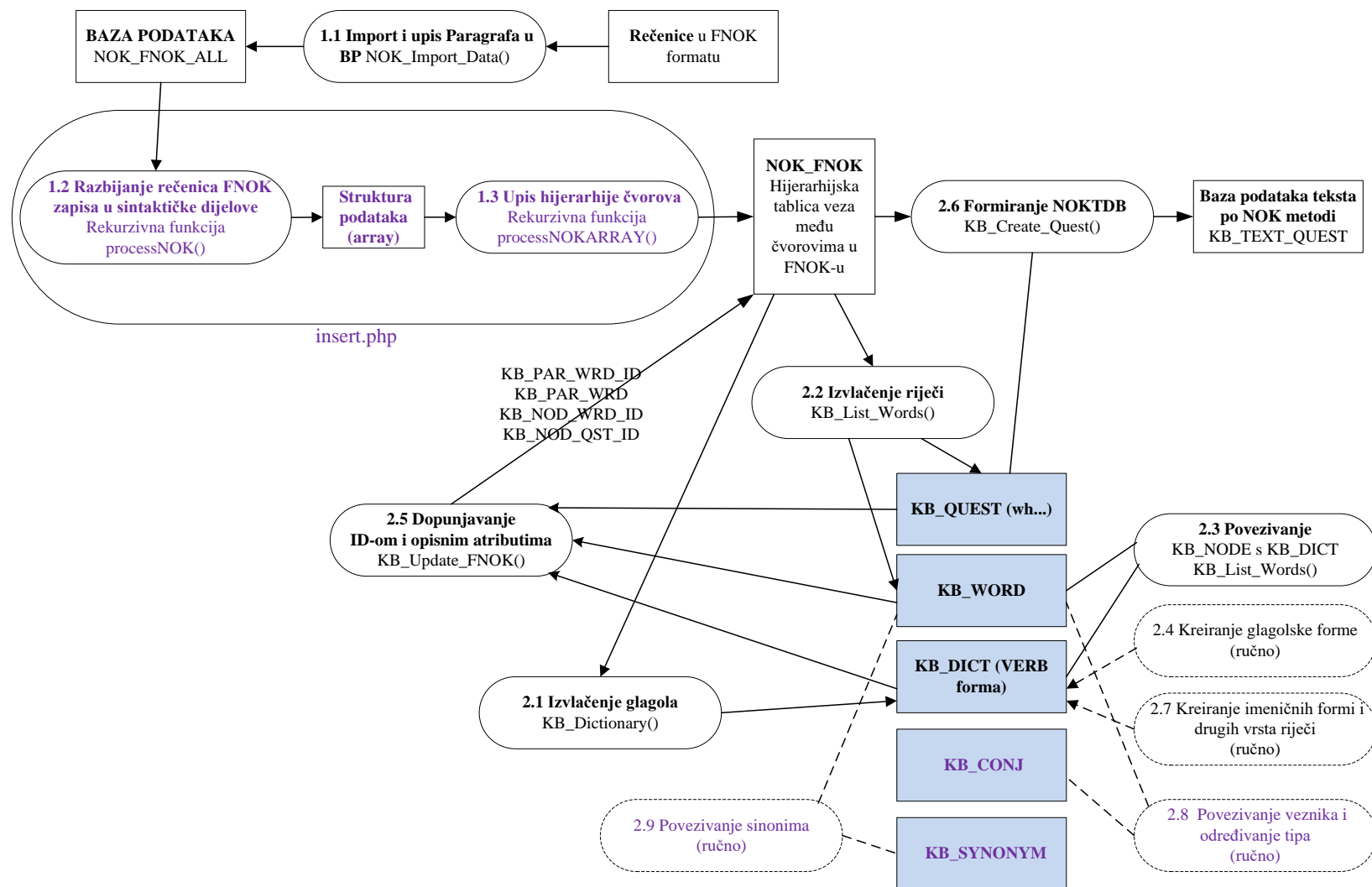
Skripta *answer.php* se sastoji od dvije funkcije `ansSortCallback($a1, $a2)` i `displayAnswers($indAnswer)`. Funkcija `ansSortCallback` sortira rečenice iz paragrafa prema broju pronađenih preklapanja čvorova između postavljenog pitanja i pojedine rečenice kako bi se ispisale prvo one koje daju točan odgovor i imaju najveći broj preklapanja do onih koje ih nemaju. Funkcija formatira i ispisuje odgovore i rečenice hijerarhijski prikazujući rečenice i pitanja i u njima označavajući zelenom bojom čvorove koji se preklapaju. Pri formatiranju koristi komponentu *Acordion* koja omogućuje

skrivanje i proširivanje sadržaja za pojedine rečenice i *front-end framework Bootstrap* za responzivnost i prilagodljivost korisničkog sučelja.

### **6.2.1. Proširivanje algoritma transformacije FNOK zapisa u BPNOK**

Rečenice engleskog jezika zapisane u formaliziranom FNOK zapisu kroz niz koraka zapisuju se u relacijsku podatka definiranih NOK metodom. S obzirom na proširivanje područja istraživanja, odnosno vrsta rečenica na kojima se provodi testiranje, bilo je potrebno izmijeniti i procese zapisivanja rečenica u relacijsku bazu.

Slika 6.4 prikazuje razradu i proširivanje procesa *1. Transformacija FNOK i QFNOK zapisa u njegovu hijerarhijsku strukturu u relacijskoj bazi podataka* i procesa *2. Obogaćivanje i povezivanje rečenica i rječnika* (Slika 6.1) primijenjenog na upisivanje rečenica. Novi i izmijenjeni procesi u odnosu na one definirane u [1] označeni su ljubičastom bojom. Proces koji nisu automatizirani označeni su isprekidanom linijom.



Slika 6.4. Prikaz transformacije formaliziranog teksta FNOK u relacijsku bazu podataka



Transformacija rečenica zapisanih u FNOK formalizmu u hijerarhijski zapis provodi se PHP skriptom *insert.php* koja sadrži funkcije `processNOK($nok)` i `processNOKArray($nok, $prg, $ord, $parent, $parWrd, $parWrdId, $parQuest)`. Prije samog procesa kreiranja hijerarhije potrebno je prepisati rečenice prirodnog jezika i njihove FNOK zapise iz tablice `NOK_FNOK_IMP` u tablicu `NOK_FNOK_ALL`, za što se više ne koristi SQL procedura `NOK_Import_Data` već su naredbe sadržane u samoj skripti *insert.php* (Privitak 2).

Funkcija `processNOK($nok)` je rekurzivna funkcija koja dijeli FNOK zapis u čvorove (proces 1.2.) na temelju sintakse definirane NOK metodom i zapisuje ih u strukturu podataka (*array*), odnosno u višedimenzionalno polje u čijem se svakom retku nalazi jedan čvor, a stupci su razine hijerarhije koje određuju zagrade u FNOK zapisu. Kada se, u izvršavanju funkcije, dođe do otvorene zagrade, funkcija rekurzivno poziva samu sebe i nastavlja obrađivati sadržaj u zagradi. Izmjenama u pravilima prevođenja u FNOK formalizirani zapis opisanim u poglavlju 0 promijenila se struktura FNOK zapisa i više prvi čvor nije uvijek glagol već može biti i veznik. Glagol u sintaksi FNOK zapisa prepoznaje se po tome jer uz njega ne stoji pitanje/uloga. Za definiranje čvorova i elemenata u strukturi podataka koriste se sljedeći ključevi:

- 'V' – glagol (*Verb*),
- 'F' – činjenice (*Fact*), odnosno ostali dijelovi rečenice koji se vežu uz glagol, a nalaze se hijerarhijski ispod glagola,
- 'Q' – pitanje (*Question*),
- 'A' – odgovor (*Answer*),
- 'C' – kontekst (*Context*), sadržaj, element fraze koji se nalazi hijerarhijski ispod ostalih vrsta riječi isključivši glagole i veznike,
- 'O' – opcije (*Options*) koje povezuje veznik i hijerarhijski se nalaze ispod veznika.

Funkcija `processNOK($nok)` čita string znak po znak i trenutni čvor znak po znak upisuje u varijablu 'elt' i provjerava je li znak razmak, zarez ili navodnik, što označava krajeve elementa. Ukoliko je znak razmak, zarez ili navodnik, provjerava se je li 'elt' prazan i ako nije 'elt' se zapisuje u 'elts', odnosno niz u koji se upisuju čvorovi. Sljedeće se provjerava je li znak otvorena zagrada '(' koji znači prelazak u novu razinu hijerarhije na način da prva otvorena zagrada mijenja varijablu 'state' u 1, a svaka sljedeća za jedan povećava varijablu 'brack' dok se za zatvorenu zagradu ')' smanjuje za jedan. Svaki drugi znak upisuje se u 'elt'. Kada 'brack' ponovno postane 0 rekurzivno se poziva funkcija `processNOK($nok)`, kojom se dalje obrađuje sadržaj unutar zagrada, rezultat dodaje u 'elts' i 'state' se vraća u 0.

Funkcija `processNOKArray($nok, $prg, $ord, $parent, $parWrd, $parWrdId, $parQuest)` hijerarhijski strukturirane podatke, koji su rezultat izvršavanja funkcije `processNOK` upisuje u relacijsku bazu podataka u tablicu `NOK_FNOK`.

Nakon izvršavanja funkcija `processNOK` i `processNOKArray` pokreću se procedure korištene u [1]:

- za obogaćivanje i povezivanje rečenica i rječnika, `KB_Dictionary()` koja čita iz tablice `NOK_FNOK` glagole, a koja je, za razliku od dosadašnje procedure kad se u tablici `KB_NOK` tražio zapis za koji je `First PARENT = 0`, izmijenjena i sada se traže

zapisi koji nemaju pitanje (QEST = \_ ), jer zbog proširivanja primjene na složene rečenice na prvom mjestu nije uvijek glagol već može biti i veznik i imati vrijednost First PARENT = 0;

- KB\_Update\_FNOK() – povezuje riječi iz tablice NOK\_FNOK s riječima iz rječnika;
- KB\_Create\_QUEST(PrgId) – popunjava tablicu KB\_TEXT\_QUEST koristeći zapise iz tablice NOK\_FNOK. Tablica KB\_TEXT\_QUEST u koju su zapisane rečenice, odnosno pojedini čvorovi od kojih su sastavljene, u nastavku se koristi u procesu dobivanja odgovora na pitanja.

Prije pokretanja algoritma za traženje odgovora na pitanja potrebno je ručno urediti tablicu KB\_CONJ koja sadrži veznike (proces 2.8) i KB\_SYNONIM koja služi za povezivanje sinonima (proces 2.9). Tablica KB\_CONJ sadrži atribute CONJ\_ID, CONJ, CONJ\_TYPE, WORD\_ID. CONJ je veznik, CONJ\_TYPE njegov tip (A, B ili O), a WORD\_ID je identifikator riječi u rječniku KB\_WORD. Svi veznici razmatrani u ovom radu upisani u su tablicu u fazi inicijalnog testiranja kada je sustav testiran na 20 rečenica u kojima su svi oni bili zastupljeni i u drugoj fazi testiranja nije bilo potrebno unositi nove veznike.

Kako bi se u bazi odredili sinonimi potrebno je prema meta modelu opisanom u poglavlju 6.1.1 u tablicu KB\_SYNONIM upisati identifikatore riječi koje su u sinonimnoj vezi.

## 6.2.2. Proširivanje algoritma transformacije QFNOK zapisa u BPNOK

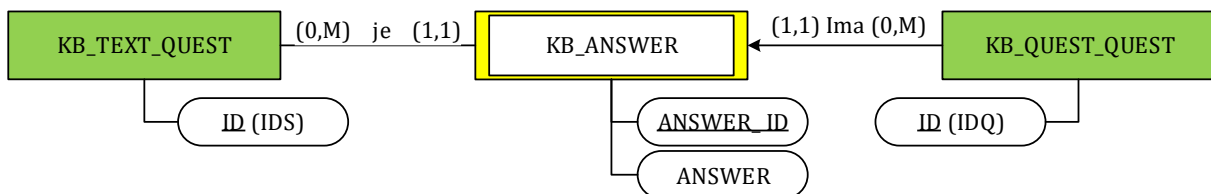
Transformacija pitanja zapisanih u formaliziranom QFNOK obliku u relacijsku bazu BPNOK provodi se na analogan način kao transformacija rečenica zapisanih u FNOK pokretanjem PHP skripte *insertQ.php* (Privitak 3). I ova skripta, kao i skripta *insert.php*, sadrži dvije funkcije `processNOK` i `processNOKArray` koje u slučaju transformacije pitanja čitaju i upisuju podatke iz i u tablice rezervirane za rad s pitanjima: NOK\_QFNOK\_IMP, NOK\_QFNOK\_ALL, NOK\_QFNOK, i KB\_QUEST\_QUEST.

Rekurzivna funkcija `processNOK($nok)` dijeli QFNOK zapis u čvorove i zapisuje ih u strukturu podataka (*array*). Za definiranje čvorova i elemenata u strukturi podataka koriste se ključevi kao i kod FNOK-a: 'V' – glagol (*Verb*), 'F' – činjenice (*Fact*), 'Q' – pitanje (*Question*), 'A' – odgovor (*Answer*), 'C' – kontekst (*Context*), 'O' – opcije (*Options*).

Funkcija `processNOKArray($nok, $prg, $ord, $parent, $parWrd, $parWrdId, $parQuest)` hijerarhijski strukturirane podatke, koji su rezultat izvršavanja funkcije `processNOK` upisuje u relacijsku bazu podataka u tablicu NOK\_QFNOK i KB\_QUEST\_QUEST. Tablica KB\_QUEST\_QUEST u koju su zapisana pitanja, odnosno pojedini čvorovi od kojih su sastavljena, u nastavku se koristi u procesu traženja odgovora.

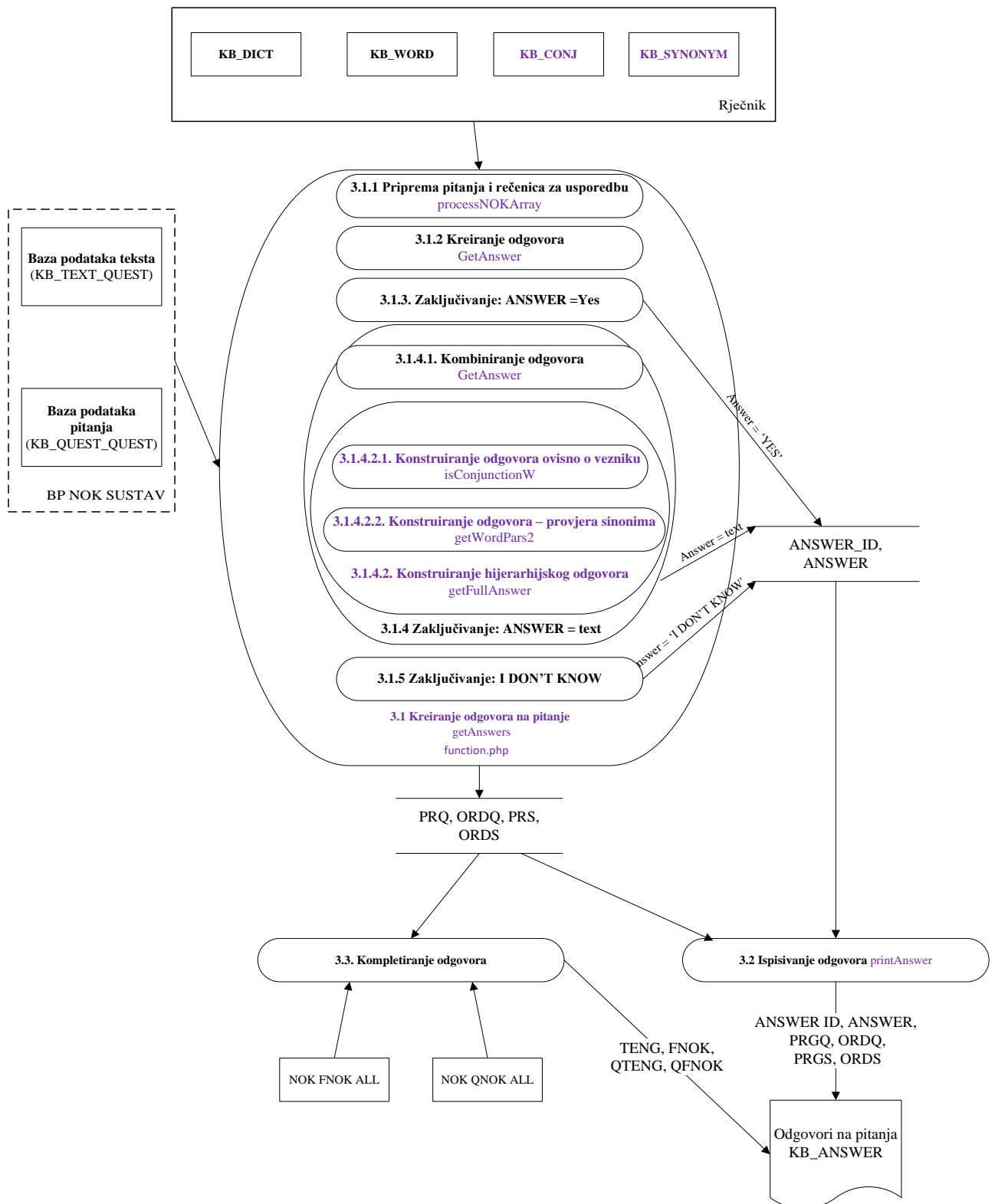
### 6.3. Kreiranje odgovora

Prethodno opisanim algoritmima u relacijsku bazu podataka upisane su rečenice i pitanja i tablice KB\_TEXT\_QUESTION i KB\_QUESTION\_QUESTION napunjene podacima. Ove tablice temeljni su izvor podataka za algoritam traženja odgovora. Proširivanje opsega područja istraživanja nije utjecao na meta model strukture i sadržaja odgovora (Slika 6.5) [1]. Prema ovom meta modelu za jedan zapis čvorova pitanja u KB\_QUESTION\_QUESTION postoji najmanje nijedan, a najviše mnogo odgovora iz slabog tipa entiteta KB\_ANSWER, a obrnuto za jedan odgovor iz KB\_ANSWER postoji najmanje jedan i najviše jedan zapis čvorova pitanja u KB\_QUESTION\_QUESTION. S druge strane, za jedan odgovor iz KB\_ANSWER postoji najmanje jedan i najviše jedan zapis čvorova rečenice u KB\_TEXT\_QUESTION, a za jedan zapis čvorova rečenica u KB\_TEXT\_QUESTION postoji najmanje nijedan, a najviše mnogo odgovora u KB\_ANSWER.



Slika 6.5. Meta model strukture i sadržaja odgovora [1]

Iako se meta model strukture i sadržaja odgovora nije promijenio, u samom procesu traženja odgovora došlo je do proširenja koje je uvjetovano vrstama rečenica, odnosno uključivanjem veznika i provjerom sinonima. Slika 6.6 prikazuje prošireni, u odnosu na model procesa dan u [1], model procesa niže razine za proces 3. Generiranje odgovora na pitanje (Slika 6.1).



Slika 6.6. Proširivanje modela procesa niže razine za proces 3. Generiranje odgovora na pitanje

Sve funkcije za kreiranje odgovora element su PHP skripte *functions.php* (Privitak 4). Ova skripta sadrži niz funkcija čijim izvršavanjem se traže odgovori na pitanja. Prva od njih je funkcija `connect()` čija je uloga spajanje na Oracle bazu.

Funkcija `getWordPar($wId)` traži i vraća identifikator roditeljskog (*parent*) čvora, odnosno osnovni oblik imenice ili glagola, za ulazni čvor iz tablice `KB_DICT`.

Funkcija `getWordPars2($wId, $findSynonims = false)` rekurzivna je funkcija koja pronalazi povezane riječi u rječniku (različite oblike riječi) i sinonime ako je tako zadano. Sinonimi se traže samo za pitanja kako bi se očuvala nebijektivnost preslikavanja sinonima. Ovu funkciju poziva funkcija `getWord($wId, $findSynonims = false)` koja vraća riječ, njene sinonime i roditeljski čvor. Funkcija `getWord` poziva i funkciju `printPars($pars)` koja niz dobiven izvršavanjem funkcije `getWordPars2` vraća kao string.

Funkcija `getQuest($qId)` dohvaća pitanje prema zadanom identifikatoru pitanja.

Osnovna funkcija koja traži odgovore na pitanja je funkcija `getAnswers($prg, $ord)`. Njezine ulazne varijable su identifikator paragrafa u kojem se traže odgovori i identifikator pitanja za koje se traži odgovor. Ova funkcije prvo iz baze podataka čita podatke o pitanju iz tablice `NOK_QFNOK` koristeću SQL naredbu (Slika 6.7).

```
$q = 'select q.*, level ' .  
    'from NOK_QFNOK q ' .  
    'where PRG = ' . $prg . ' ' .  
    'and ORD = ' . $ord . ' ' .  
    'connect by prior node_id = parent_id and prior prg = prg and  
    prior ord = ord ' .  
    'start with parent_id = 0';
```

**Slika 6.7. SQL naredba za čitanje podataka o pitanju iz tablice `NOK_FNOK`**

Zatim se dohvaćaju elementi pitanja, odnosno njihov hijerarhijski zapis iz tablice `KB_QUEST_QUEST`. Svi dohvaćeni podaci upisuju su u strukture, odnosno polja.

Nakon što funkcija `getAnswers` obriše prethodne odgovore na pitanje koje se obrađuje iz tablice `KB_ANSWER`, pretražuju se sve rečenice iz paragrafa i traže odgovori na pitanje. Funkcijom `getSentences()` dohvaćaju se sve rečenice iz tablica `NOK_FNOK_ALL` i `KB_TEXT_QUEST` nakon čega se započinje s usporedbom čvorova iz rečenica i pitanja za sljedeće slučajeve:

- Traži se preklapanje svih čvorova u pitanju i rečenici. Provjera se provodi funkcijom `function equalWords($w1, $w2)` koja traži preklapanja čvorova uključujući sve oblike tražene riječi iz rječnika i provjeravajući sinonime.
- Ukoliko se nađe isti čvor roditelj (*parent*), ali sljedeći čvor nema preklapanje, provjerava se je li taj čvor veznik pozivanjem funkcije `isConjunctionW($word)`, a ako je veznik, traže se preklapanja čvorova koji se hijerarhijski nalaze ispod veznika, odnosno provjeravaju se direktna djeca koja daju odgovor zajedno s veznikom. U obrnutom slučaju, kada je veznik prvi element, odnosno roditelj pronađenom čvoru, provjeravaju se roditelji veznika.

- Ukoliko u pitanju se nalazi veznik tipa O, što se provjerava funkcijom `getConjType($wordId)`, dovoljno je upariti samo jedan čvor ispod veznika da bi se dobio odgovor. Na primjer za pitanje „*Who is from Zagreb or Rijeka?*“ dovoljno je naći u rečenici jedan od navedenih gradova da bi se dobio odgovor. To omogućava da sustav nađe odgovor i u rečenici „*Marko and Ana are from Zagreb.*“ (Slika 6.8), jer je kao odgovor dovoljno da netko bude u jednom iz ova dva grada („*Rijeka or Zagreb*“).

**Pitanje**

Who is from Zagreb or Rijeka? Odaberi

---

**Formalizirani zapis**  
 is ("who?" X, \_ from ( \_ or ( \_ Zagreb, \_ Rijeka)))

OK!!

Answered: 1.0516281032562

### Hijerarhijski zapis pitanja

Node ID	Node	Question	Parent ID
1	is	..	0
2	X	who	1
3	from	..	1
4	or	..	3
5	Zagreb	..	4
6	Rijeka	..	4

### Obrađeno pitanje

ID	PAR_WRD_ID	NOD_QST_ID	NOD_WRD_ID
326	149 is is are be was were am would been	3 who	579 X X
327	149 is is are be was were am would been	-1 ..	124 from from
328	124 from from	-1 ..	302 or or
329	302 or or	-1 ..	306 Zagreb Zagreb
330	302 or or	-1 ..	369 Rijeka Rijeka

✓ [99-24] Lucija is from Zagreb or Rijeka. ^

is ("who?" Lucija, "where?" from ("where?" or ("where?" Zagreb, "where?" Rijeka)))

**ODGOVOR:** Lucija

PARENT	QUESTION	NODE
[1] is	who	[2] Lucija
[1] is	where	[3] from
[3] from	where	[4] or
[4] or	where	[5] Zagreb
[4] or	where	[6] Rijeka

PARENT	QUESTION	NODE
is	who	X
is	..	from
from	..	or
or	..	Zagreb
or	..	Rijeka

✓ [99-7] Marko and Ana are from Zagreb. ^

are ("who?" and ("who?" Marko, "who?" Ana), "where?" from ("where?" Zagreb))

**ODGOVOR:** Marko and Ana

PARENT	QUESTION	NODE
[1] are	who	[2] and
[2] and	who	[3] Marko
[2] and	who	[4] Ana
[1] are	where	[5] from
[5] from	where	[6] Zagreb

PARENT	QUESTION	NODE
is	who	X
is	..	from
from	..	or
or	..	Zagreb
or	..	Rijeka

**Slika 6.8. Odgovori na pitanje „Who is from Zagreb or Rijeka?“**

- Kada je u pitanju veznik tipa B (*Subordinating Conjunctions*) koji povezuje dvije zavisno složene rečenice može se dogoditi slučaj da su upareni svi čvorovi pitanja sa svim čvorovima jedne od rečenica koje povezuje veznik tipa B. U tom slučaju se zaključuje da je odgovor cijela druga rečenica uključujući veznik. Ovo je vidljivo na primjeru

pitanja „Why Daniel loves summer?“ (Slika 6.9.) na koje sustav daje odgovor sastavljen od svih preostalih čvorova koji nisu upareni s čvorovima iz pitanja što se vidi na slici lijevo, u dijelu s čvorovima rečenice, gdje su upareni čvorovi osjenčani zelenom bojom, a iz neosjenčanih čvorova je generiran odgovor.

✓ [99-43] Daniel loves summer because he loves swimming. ^

"why?" because (loves ("who?" Daniel, "what?" summer), loves ("who?" he, "what?" swimming))

**ODGOVOR:** because loves he swimming

PARENT	QUESTION	NODE	PARENT	QUESTION	NODE
[1] because	-	[2] loves	loves	-	Daniel
[2] loves	who	[3] Daniel	loves	-	summer
[2] loves	what	[4] summer	loves	why	X
[1] because	-	[5] loves			
[5] loves	who	[6] he			
[5] loves	what	[7] swimming			

**Slika 6.9. Odgovor sustava na pitanje „Why Daniel loves summer?“**

Prethodno navedeni slučajevi, odnosno koraci samog algoritma, nakon provjere i uparivanja svih čvorova rečenica i pitanja pronađene parove spremaju u php strukturu (listu) *\$foundPars*, dok su čvorovi pitanja zapisani u *\$qElts* i čvorovi rečenica u *\$sElts*. Prije samog traženja odgovora inicijaliziraju se varijable koje će se koristiti za praćenje odgovora (*\$answer*), identificiranje riječi koja je odgovor (*\$answerWrd*), praćenje jesu li pronađeni svi čvorovi pitanja (*\$allFound*) i varijabla za pripremu odgovora (*\$answerPrep*). Nakon toga slijedi traženje vrijednosti za X i generiranje odgovora. Traženje odgovora X provodi se provjerom jednakosti između čvora pitanja i čvora rečenice pozivanjem funkcije `equalWords`:

```
if(equalWords($qElt['PAR_WRD_ID'], $sElt['PAR_WRD_ID']))
```

U slučaju da su čvorovi jednaki provjerava se je li vrijednost 'X' u pitanju:

```
if($qElt['NOD_WRD'] === 'X')
```

Ako je za 'X' pronađen upareni čvor (odgovor) upisuje se u varijablu u varijablu *\$answer*. Ako već postoji pronađeni čvor upisuje se u varijablu *\$answerWrd*:

```
$answerWrd = $sElt['NOD_WRD_ID'] . '-' . $sElt['NODE_ID'];
```

Kada algoritam prođe kroz sve čvorove generira se odgovor ako je pronađen. Prvo se provjerava postoji li preklapanje svih čvorova pitanja i rečenice i u tom slučaju odgovor je 'YES', odnosno u tom slučaju je postavljeno pitanje na koje se može odgovoriti s 'YES' ili 'NO'. Nakon toga se provjerava jesu li pronađeni odgovori i poziva funkcija `getFullAnswer($s, $wordId, $answerB)` koja generira potpuni odgovor koji ispisuje funkcija `printAnswer($elts, $idx, $ans)`.



## 7. Primjena sustava temeljenog na NOK metodi na složene rečenice engleskog jezika

U ovom poglavlju opisan je postupak testiranja i rezultati primjene algoritama iz poglavlja 6.2 i 6.3 na rečenice prirodnog jezika definirane u poglavlju 1.

Testiranje se provodilo u dvije faze:

1. faza – faza inicijalnog testiranja,
2. faza – faza glavnog testiranja.

Prva faza provedena je na 20 odabranih reprezentativnih rečenica i 72 pitanja postavljenih na njih. Rečenice su odabrane prema kriteriju da se u barem jednoj od njih pojavljuju svi testirani veznici: *and, or, but, because, if, when, that, unless, whenever, while, until* i *so*. U ovoj fazi dopunjen je i obogaćen rječnik, odnosno tablica KB\_CONJUNCTION s testiranim veznici koji su upisani u tablicu nakon prvog pojavljivanja pri transformaciji rečenica prirodnog jezika u tablice relacijske baze podataka, odnosno relacijski BPNOK zapis. Prilikom generiranja odgovora na pitanja za koje se odgovor nalazi u testiranim rečenicama uočena je potreba određivanja tipa veznika čime je omogućeno unaprjeđenje algoritma traženja odgovora na pitanja.

U obje faze testiranja obogaćivao se rječnik i tablica KB\_SYNONIM novim sinonimima koji se upisuju pri prvom pojavljivanju u rečenici, odnosno pri njenom upisivanju u relacijsku bazu podataka.

Druga faza, nakon unaprjeđenja algoritma, provedena je ponovno na 20 rečenica iz inicijalnog testiranja i još dodatnih 80 rečenica prirodnog jezika, koje sadrže veznike, i ukupno 280 pitanja temeljenih na svih 100 testiranih rečenica.

U prilogima se nalaze tablice s testiranim rečenicama i pitanjima:

- tablica koja sadrži 20 rečenica engleskog jezika i odgovarajućih FNOK zapisa nad kojima je izvršeno inicijalno testiranje (Privitak 5)
- tablica koja sadrži 72 pitanja engleskog jezika i odgovarajućih QFNOK zapisa s kojima se propitivalo znanje i tražili odgovori na pitanja u testiranim rečenicama u inicijalnom testiranju (Privitak 6),
- tablica koja sadrži svih 100 rečenica engleskog jezika i odgovarajućih FNOK zapisa nad kojima je izvršeno glavno testiranje (Privitak 7),
- tablica koja sadrži svih 280 pitanja engleskog jezika i odgovarajućih QFNOK zapisa s kojima se propitalo znanje u glavnom testiranju (Privitak 8),
- tablica koja prikazuje svih 280 pitanja engleskog jezika i pripadajućih očekivanih odgovora uspoređenih s odgovorima koje je dao sustav (Privitak 9).

## 7.1. Primjena proširenog sustava temeljenog na NOK metodi

U ovom poglavlju bit će prikazan dio sadržaja relacijske baze podataka BPNOK, odnosno onaj dio sadržaja na kojem je vidljiva funkcionalnost primjene sustava na složene rečenice testirane u ovom radu i sadržaj koji nije bio element sustava opisanog u [1].

Tablice NOK\_PARAGRAPH, NOK\_FNOK\_ALL, KB\_WORD, KB\_DICT, KB\_QUEST, NOK\_FNOK\_ALL, NOK\_QFNOK\_ALL se nisu mijenjale u odnosu na prethodnu inačicu sustava i u njih se inicijalno upisuje sadržaj koji će sustav obrađivati.

U rječnik su dodane dvije tablice KB\_CONJ i KB\_SYNONIM. U KB\_CONJ (CONJ\_ID, CONJ, CONJ\_TYPE, WORD\_ID) su upisani svi korišteni veznici (CONJ), njihovi tipovi (CONJ\_TYPE) i njihovi identifikatori (WORD\_ID) preko kojih su povezani s tablicom KB\_WORD. Slika 7.1 prikazuje sadržaj tablice KB\_CONJ.

CONJ...	CONJ	CONJ_TYPE	WORD_ID
1	and	A	294
2	or	O	302
4	because	B	475
5	but	A	384
6	when	B	483
7	if	B	380
8	that	B	235
9	unless	B	478
10	whenever	B	507
11	while	B	509
12	until	B	474
13	so	A	460

Slika 7.1. KB\_CONJ

Tablica KB\_SYNONYM (SYN\_ID, WORD\_ID1, WORD\_ID2) sadrži vanjske ključeve na način da je identifikator WORD\_ID2 identifikator riječi u rječniku KB\_WORD koja je sinonim riječi koja u KB\_WORD ima identifikator WORD\_ID1. Implementacija sinonima opisana je u poglavlju 6.1.1. Sadržaj tablice KB\_SYNONIM prikazuje Slika 7.2.

..	WORD_ID1	WORD_ID2
2	616	249
3	231	609
4	622	105
5	467	303

Slika 7.2. KB\_SYNONIM

Funkcijama `processNOK` i `processNOKArray` omogućeno je u tablicu `NOK_FNOK` transformirati rečenice i koje sadrže veznike i više procesnih čvorova zapisane u tablici `NOK_FNOK_ALL` u popis čvorova koji su hijerarhijski povezani s pridruženim nadređenim čvorom (`PARENT_ID`). Čvorovi se zapisuju iz strukture podataka `array` u tablicu `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`). Slika 7.3 prikazuje zapis rečenice „*Vanessa is sleepy because she has been studying all night.*“ u tablici `NOK_FNOK`, a Slika 7.4 hijerarhijski zapis u polju `array` u kojem je pripremljena rečenica za zapis u tu tablicu.

PRG	...	NODE_ID	NODE	QUEST	PARENT_ID	KB_PAR_WRD_ID	KB_PAR_WRD	KB_NOD_WRD_ID	KB_NOD_QST_ID	ID
99	42	1	because	why	0	(null)	(null)	475	16	686
99	42	2	is	_	1	475	because	149	-1	687
99	42	3	Vanessa	who	2	149	is	55	3	688
99	42	4	sleepy	what	2	149	is	518	2	689
99	42	5	has	_	1	475	because	136	-1	690
99	42	6	been	_	5	136	has	519	-1	691
99	42	7	studying	_	6	519	been	520	-1	692
99	42	8	she	who	7	520	studying	208	3	693
99	42	9	night	when	7	520	studying	521	11	694
99	42	10	all	what	9	521	night	60	2	695

**Slika 7.3. Tablica `NOK_FNOK`**

[42] Vanessa is sleepy because she has been studying all night.

```
array (  
  0 =>  
    array (  
      'Q' => 'why',  
      'C' => 'because',  
      'O' =>  
        array (  
          0 =>  
            array (  
              'V' => 'is',  
              'F' =>  
                array (  
                  0 =>  
                    array (  
                      'Q' => 'who',  
                      'A' => 'Vanessa',  
                    ),  
                  1 =>  
                    array (  
                      'Q' => 'what',  
                      'A' => 'sleepy',  
                    ),  
                ),  
              ),  
            ),  
          1 =>  
            array (  
              'V' => 'has',  
              'F' =>  
                array (  
                  'V' => 'been',  
                  'F' =>  
                    array (  
                      'V' => 'studying',  
                      'F' =>  
                        array (  
                          0 =>  
                            array (  
                              'Q' => 'who',  
                              'A' => 'she',  
                            ),  
                          1 =>  
                            array (  
                              'Q' => 'when',  
                              'C' => 'night',  
                              'O' =>  
                                array (  
                                  'Q' => 'what',  
                                  'A' => 'all',  
                                ),  
                              ),  
                            ),  
                          ),  
                        ),  
                      ),  
                    ),  
                  ),  
                ),  
              ),  
            ),  
          ),  
        ),  
      ),  
    ),  
  ),  
)
```

*Slika 7.4. Hijerarhijski zapis rečenice u polje*

U tablici NOK\_FNOK rečenice su zapisane čvor po čvor tako da je svaki čvor zapisan u jedan redak tablice. Svaki redak osim čvora sadrži i identifikator hijerarhijski nadređenog čvora (PARENT\_ID), identifikator riječi u tablici KB\_WORD (KB\_PAR\_WRD\_ID) preko kojeg je povezan s rječnikom, identifikator pitanja koje je postavljeno u formaliziranom FNOK zapisu.

Slika 7.5 prikazuje konačnu tablicu KB\_TEXT\_QUEST pripremljenu za traženje odgovora na pitanja i zapis čvorova rečenice „*Vanessa is sleepy because she has been studying all night.*“ Osnovni atributi ove tablice su PAR\_WRD\_ID, NOD\_WRD\_ID i NOD\_QST\_ID, a preostala četiri atributa dodana su radi lakšeg praćenja (PAR\_WRD, NOD\_QST, NODE\_ID i PARENT\_ID). NOD\_QST\_ID je pitanje pridruženo čvoru-roditelju PAR\_WRD\_ID na koje je NOD\_WRD\_ID odgovor.

ID	PRG	...	PAR_WRD_ID	NOD_WRD_ID	NOD_QST_ID	PAR_WRD	NOD_WRD	NOD_QST	NODE_ID	PARENT_ID
286	99	42	475	149	-1	because	is	_	2	1
287	99	42	149	55	3	is	Vanessa	who	3	2
288	99	42	149	518	2	is	sleepy	what	4	2
289	99	42	475	136	-1	because	has	_	5	1
290	99	42	136	519	-1	has	been	_	6	5
291	99	42	519	520	-1	been	studying	_	7	6
292	99	42	520	208	3	studying	she	who	8	7
293	99	42	520	521	11	studying	night	when	9	7
294	99	42	521	60	2	night	all	what	10	9

**Slika 7.5. KB\_TEXT\_QUEST**

Na analogan način se pitanja i njihov formalizirani QFNOK zapis zapisuju u tablice relacijske baze podataka. Funkcijama `processNOK` i `processNOKArray` omogućeno je, u tablicu NOK\_QFNOK, transformirati pitanja koje sadrže veznike zapisane u tablici NOK\_QFNOK\_ALL u popis čvorova koji su hijerarhijski povezani s pridruženim nadređenim čvorom (PARENT\_ID). Čvorovi se zapisuju iz strukture podataka array u tablicu 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).

Slika 7.6 prikazuje hijerarhijski zapis u polju array u kojem je pripremljeno pitanje za upisivanje u tablicu NOK\_QFNOK.

```

[156] Why is Vanessa sleepy?

is ( _ sleepy, _ Vanessa, "why?" X)

array (
  'V' => 'is',
  'F' =>
  array (
    0 =>
    array (
      'Q' => '_',
      'A' => 'sleepy',
    ),
    1 =>
    array (
      'Q' => '_',
      'A' => 'Vanessa',
    ),
    2 =>
    array (
      'Q' => 'why',
      'A' => 'X',
    ),
  ),
)

```

**Slika 7.6. Hijerarhijski zapis pitanja u polje**

Slika 7.7 prikazuje zapis pitanja „Why is Vanessa sleepy?“ u tablici NOK\_QFNOK, a Slika 7.8 prikazuje konačnu tablicu KB\_QUEST\_QUEST pripremljenu za traženje odgovora na pitanja i zapis čvorova pitanja.

PRG	...	NODE_ID	NODE	QUEST	PARENT_ID	KB_PAR_WRD_ID	KB_PAR_WRD	KB_NOD_WRD_ID	KB_NOD_QST_ID
99	156	1	is	_	0	(null)	(null)	149	-1
99	156	2	sleepy	_	1	149	is	518	-1
99	156	3	Vanessa	_	1	149	is	55	-1
99	156	4	X	why	1	149	is	579	16

**Slika 7.7. NOK\_QFNOK**

ID	PRG	...	PAR_WRD_ID	NOD_WRD_ID	NOD_QST_ID	PAR_WRD	NOD_WRD	NOD_QST
527	99	156	149	518	-1	is	sleepy	_
528	99	156	149	55	-1	is	Vanessa	_
529	99	156	149	579	16	is	X	why

**Slika 7.8. KB\_QUEST\_QUEST**

Nakon pripremljenih tablica KB\_TEXT\_QUEST (Slika 7.5) i KB\_QUEST\_QUEST (Slika 7.8) te popunjavanja podacima može se krenuti s traženjem odgovora na pitanja i popunjavanjem tablice KB\_ANSWER (ANSWER\_ID, PRGQ, ORDQ, QTENG, ANSWER, PRGS, ORDS, TENG, QFNOK, FNOK). Tablica 7.1 prikazuje odgovore na pitanja u tablici KB\_ANSWER za rečenicu „*Vanessa is sleepy because she has been studying all night.*“

Kao u do sada korištenom sustavu u tablici KB\_ANSWER ostavljena su polja QTENG, TENG, QFNOK i FNOK koja olakšavaju praćenje i čitanje podataka iz tablice kao što je prikazano i u korisničkom sučelju.

Tablica 7.1. KB\_ANSWER

ANSWER_ID	PRGQ	ORDQ	QTENG	ANSWER	PRGS	ORDS	TENG	QFNOK	FNOK
2801	99	155	Who has been studying all night?	she	99	42	Vanessa is sleepy because she has been studying all night.	has (been (studying ("who?" X, _ night ( _ all))))	"why?" because (is ("who?" Vanessa, "what?" sleepy), has (been (studying ("who?" she, "when?" night ("what?" all))))))
2816	99	154	What is Vanessa?	sleepy	99	42	Vanessa is sleepy because she has been studying all night.	is ( _ Vanessa, "what?" X)	"why?" because (is ("who?" Vanessa, "what?" sleepy), has (been (studying ("who?" she, "when?" night ("what?" all))))))
2840	99	156	Why is Vanessa sleepy?	because has been studying she all night	99	42	Vanessa is sleepy because she has been studying all night.	is ( _ sleepy, _ Vanessa, "why?" X)	"why?" because (is ("who?" Vanessa, "what?" sleepy), has (been (studying ("who?" she, "when?" night ("what?" all))))))
1556	99	153	Who is sleepy?	Vanessa	99	42	Vanessa is sleepy because she has been studying all night.	is ("who?" X, _ sleepy)	"why?" because (is ("who?" Vanessa, "what?" sleepy), has (been (studying ("who?" she, "when?" night ("what?" all))))))



## 7.2. Dokaz hipoteze H1

Hipoteza H1 glasi: *Primjenom konceptualnog razvojnog okvira NOK, određene vrste složenih rečenica engleskog jezika se mogu transformirati u FNOK i relacijski BPNOK zapis.*

Ova hipoteza se smatra potvrđenom ako se pokaže da se definirane vrste složenih rečenica prirodnog jezika prema definiranim pravilima mogu transformirati u formalizirani FNOK zapis, a taj zapis definiranim algoritmom zapisati u relacijski BPNOK zapis. Pod relacijskim BPNOK zapisom smatraju se formalizirani FNOK zapisi rečenica upisani u tablice relacijske baze podataka (NOK\_FNOK, KB\_TEXT\_QUEST) na način da se svaka riječ (čvor) upisuje u svoj redak zajedno s identifikatorom riječi koja je njezin „roditelj“ čime je zapisana hijerarhijska povezanost između čvorova.

Slika 7.9. prikazuje zapise rečenica od rednog broja 48 do 52 u tablicu KB\_TEXT\_QUEST.

U poglavlju 4 definirana su pravila transformacije rečenica prirodnog jezika u formalizirani FNOK zapis. Ova pravila primijenjena su na 100 rečenica engleskog jezika koje zadovoljavaju opseg ovog istraživanja te su transformacijom dobiveni formalizirani FNOK zapisi (Privitak 7). Algoritmima opisanim i definiranim u poglavlju 6.2. sve rečenice, odnosno njihovi formalizirani FNOK zapisi transformirani su u relacijski BPNOK zapis i upisani u tablice NOK\_FNOK (Slika 7.3.) i KB\_TEXT\_QUEST (Slika 7.5.) čime su pripremljene za traženje odgovora na pitanja. Time je pokazano da se određene vrste složenih rečenica mogu transformirati u FNOK zapis, a takvi zapisi se mogu transformirati i upisati u zapise tablice relacijske baze podataka BPNOK. Popunjavanjem relacijske baze podataka BPNOK odabranim složenim rečenicama engleskog jezika koje sadrže veznike i definirane su u poglavlju 1 dokazana je hipoteza H1.

ID	PRG	PAR_WRD_ID	NOD_WRD_ID	NOD_QST_ID	PAR_WRD	NOD_WRD	NOD_QST	NODE_ID	PARENT_ID
331	99	48	384	255	-1 but	was	_	2	1
332	99	48	255	237	12 was	there	where	3	2
333	99	48	255	101	-1 was	do	_	4	2
334	99	48	101	245	12 do	to	where	5	4
335	99	48	245	174	2 to	nothing	what	6	5
336	99	48	384	401	2 but	wait	what	7	1
337	99	49	384	149	-1 but	is	_	2	1
338	99	49	149	393	-1 is	caused	_	3	2
339	99	49	393	392	-1 caused	not	_	4	3
340	99	49	392	239	2 not	this	what	5	4
341	99	49	392	394	2 not	evil	what	6	4
342	99	49	394	395	2 evil	by	what	7	6
343	99	49	393	396	2 caused	ignorance	what	8	3
344	99	49	396	397	2 ignorance	simple	what	9	8
345	99	49	397	395	2 simple	by	what	10	9
346	99	50	384	398	-1 but	ran	_	2	1
347	99	50	398	256	3 ran	we	who	3	2
348	99	50	398	399	5 ran	after	how	4	2
349	99	50	398	139	3 ran	him	who	5	2
350	99	50	384	400	-1 but	escaped	_	6	1
351	99	50	400	137	3 escaped	he	who	7	6
352	99	51	384	149	-1 but	is	_	2	1
353	99	51	149	152	2 is	it	what	3	2
354	99	51	149	385	2 is	dress	what	4	2
355	99	51	385	386	2 dress	nice	what	5	4
356	99	51	386	5	1 nice	a	art	6	5
357	99	51	384	387	-1 but	creases	_	7	1
358	99	51	387	152	2 creases	it	what	8	7
359	99	51	387	388	5 creases	easily	how	9	7
360	99	51	388	251	5 easily	very	how	10	9
361	99	52	283	564	-1 have	received	_	2	1
362	99	52	564	30	3 received	I	who	3	2
363	99	52	564	565	2 received	replies	what	4	2
364	99	52	564	384	3 received	but	who	5	2
365	99	52	384	111	14 but	everybody	whom	6	5
366	99	52	111	124	14 everybody	from	whom	7	6
367	99	52	384	566	3 but	Jane	who	8	5

Slika 7.9. Zapisi rečenica 48 do 52 u tablicu KB\_TEXT\_QUEST

### 7.3. Analiza rezultata

Sustav je testiran na 100 složenih rečenica engleskog jezika (Privitak 7) i 280 pitanja (Privitak 8) za koja se odgovor nalazi u tim rečenicama. Za svako pitanje unaprijed je definiran očekivani odgovor ili odgovori koje bi na postavljena pitanja dao čovjek, a koji su uspoređivani s odgovorima koje je dao sustav.

Dobivene odgovore može se svrstati u jednu od sljedeći 4 kategorije:

1. Točan odgovor

2. Gramatički neispravan odgovor
3. Djelomično točan odgovor
4. Nema odgovora

U nastavku su dana pojašnjenja i primjeri za svaku od kategorija.

### 7.3.1. Točan odgovor

U ovome slučaju odgovori koje je dao sustav su istovjetni očekivanom odgovoru koji je dao čovjek kao što je prikazano kroz pet primjera koje prikazuje Tablica 7.2. Točni odgovori dobiveni su na 198 od postavljenih 280 pitanja, odnosno 70,71 %.

**Tablica 7.2. Točni odgovori**

Red. br.	QTENG	Očekivani odgovor	ANSWER sustava	Analiza odgovora
1.	What are Tom and Paul talking about?	about the solution	about the solution	Točno
2.	What does Sanja train?	volleyball and tennis	volleyball and tennis	Točno
3.	Who did we kiss and hug?	each other	each other	Točno
4.	Who trains volleyball and tennis?	Sanja, Peter	Sanja, Peter	Točno
5.	Who is playing?	Ana and John, Martina and Josip	Ana and John, Martina and Josip	Točno

U prva tri primjera očekivani i dobiveni odgovori su iz jedne rečenice zapisane u BPNOK. U prvom primjeru nalazi se veznik koji je u ulozi povezivanja subjekata u pitanju. U drugom pitanju/primjeru veznik povezuje objekte u odgovoru, a u trećem veznik u pitanju povezuje radnje, odnosno glagole. Za sve ove opcije u ulozi veznika, u sva tri dana primjera, dobiveni su točni odgovori.

U četvrtom i petom primjeru očekivana su dva odgovora koja se nalaze u dvije različite rečenice zapisane u BPNOK. U četvrtom primjeru veznik je korišten u pitanju „*Who trains volleyball and tennis?*“ i povezuje dva objekta, a odgovori su subjekti iz rečenica: „*Sanja is training volleyball and tennis.*“ i „*Peter trains basketball, volleyball and tennis.*“ U petom primjeru odgovori na pitanje „*Who is playing?*“ se nalaze u rečenicama „*Ana and John play the violin.*“ i „*Martina and Josip play the violin extremely well.*“. Veznik se u ovom slučaju nalazi u odgovoru i povezuje dva subjekta. Slika 7.10 prikazuje odgovore koje je dao sustav na pitanje „*Who is playing?*“ na kojoj je vidljivo da su odgovori pronađeni u obje rečenice i jednaki su očekivanim odgovorima.

## Odgovori na pitanja

Pitanje

Who is playing? Odobri

Formalizirani zapis  
`playing ( & is) ( "who?" X)`

Answered: 0.59096002578735

### Hijerarhijski zapis pitanja

Node ID	Node	Question	Parent ID
1	playing	-	0
2	is	&	1
3	X	who	1

### Obrađeno pitanje

ID	PAR WRD ID	NOD QST ID	NOD WRD ID
20	614 playing playing_and play plays	4 &	149 is is are be was were am would been
21	614 playing playing_and play plays	3 who	579 X X

✓ [99-2] Ana and John play the violin. ^

`play ("who?" and ("who?" Ana, "who?" John), "what?" violin ("art?" the))`

**ODGOVOR:** Ana and John

PARENT	QUESTION	NODE
[1] play	who	[2] and
[2] and	who	[3] Ana
[2] and	who	[4] John
[1] play	what	[5] violin
[5] violin	art	[6] the

PARENT	QUESTION	NODE
playing	&	is
playing	who	X

✓ [99-3] Martina and Josip play the violin extremely well. ^

`play ("who?" and ("who?" Martina, "who?" Josip), "what?" violin ("art?" the), "how?" well ("how?" extremely))`

**ODGOVOR:** Martina and Josip

PARENT	QUESTION	NODE
[1] play	who	[2] and
[2] and	who	[3] Martina
[2] and	who	[4] Josip
[1] play	what	[5] violin
[5] violin	art	[6] the
[1] play	how	[7] well
[7] well	how	[8] extremely
[9] extremely	what	[10] violin
[10] violin	art	[11] the

PARENT	QUESTION	NODE
playing	&	is
playing	who	X

Slika 7.10. Prikaz odgovora na pitanje "Who is playing?"

### 7.3.2. Gramatički neispravan odgovor

U kategoriji gramatički neispravan odgovor promatrana su pitanja za koja sustav daje odgovor koji nije gramatički ispravan, ali je sadržajno razumljiv i prihvatljiv čovjeku, s obzirom na sposobnost razumijevanja iz konteksta, ali i svojstvo zalihosti (redundancije) koje čovjeku

omogućava popunjavanje praznina nastalih u prijenosu informacije [112, 113]. Od 280 pitanja na njih 42 (15 %) dobiveni su odgovori koji su kategorizirani u gramatički neispravne odgovore.

Primjere takvih pitanja i odgovora prikazuje Tablica 7.3.

**Tablica 7.3. Primjeri pitanja s gramatički neispravnim odgovorom**

Red. br.	QTENG	Očekivani odgovor	ANSWER sustava	Analiza odgovora
1.	Where is Lucija from?	from Zagreb or Rijeka	Zagreb or Rijeka from; is Zagreb or Rijeka from	Gramatički neispravno
2.	Why you have to go to college for a lot of years?	if you want to be a doctor	if want to a doctor	Gramatički neispravno
3.	Why do I went to see a specialist?	it was still painful	so was it still painful	Gramatički neispravno
4.	Why is Monika glad?	that workers aren't losing their jobs	that aren't losing workers their jobs	Gramatički neispravno
5.	When the springs in the armchair creak?	whenever I change position	whenever change I position	Gramatički neispravno

U prva tri primjera prisutno je svojstvo zalihosti i to na dva različita načina. U prvom primjeru odgovor sustava je širi, odnosno sadrži više riječi/čvorova iz rečenice u kojoj je pronađen odgovor što ne mijenja semantičko značenje pronađenog odgovora, dok je u drugom i trećem primjeru uži od očekivanog te sadrži manji broj riječi/čvorova u odnosu na očekivani odgovor. U oba slučaja odgovorima je preneseno traženo znanje i odgovor se u tom kontekstu može smatrati točnim.

U preostala dva primjera greška je u redosljedu riječi u odgovoru. Na primjer u četvrtom primjeru na pitanje „*Why is Monika glad?*“ odgovor je pronađen u rečenici „*Monika is glad that workers aren't losing their jobs.*“ Dobiven je odgovor „*that aren't losing workers their jobs*“ (Slika 7.11), a razlog krivog poretka je u strukturi FNOK zapisa i hijerarhijski organiziranih čvorova u odnosu na procesni čvor.

## Odgovori na pitanja

Pitanje

Why is Monika glad? Odaberi

Formalizirani zapis  
is ( \_ Monika, \_ glad, "why?" X)

Answered: 0.61394786834717

### Hijerarhijski zapis pitanja

Node ID	Node	Question	Parent ID
1	is	-	0
2	Monika	-	1
3	glad	-	1
4	X	why	1

### Obrađeno pitanje

ID	PAR_WRD_ID	NOD_QST_ID	NOD_WRD_ID
1041	149 is is are be was were am would been	-1 -	365 Monika Monika
1042	149 is is are be was were am would been	-1 -	467 glad glad happy
1043	149 is is are be was were am would been	16 why	579 X X

✓ [99-89] Monika is glad that workers aren't losing their jobs. ^

"why?" that (is ("who?" Monika, "what?" glad), aren't (losing ("who?" workers, "what?" jobs ("whose?" their ))))

ODGOVOR: that aren't losing workers their jobs

PARENT	QUESTION	NODE
[1] that	-	[2] is
[2] is	who	[3] Monika
[2] is	what	[4] glad
[1] that	-	[5] aren't
[5] aren't	-	[6] losing
[6] losing	who	[7] workers
[6] losing	what	[8] jobs
[8] jobs	whose	[9] their

PARENT	QUESTION	NODE
is	-	Monika
is	-	glad
is	why	X

Slika 7.11. Odgovor sustava na pitanja "Why is Monika glad?"

Prema definiranim pravilima i, na temelju njih definiranim algoritmima, procesni čvorovi (glagoli ili veznik) se nalazi uvijek na vrhu hijerarhije, kao što je i u promatranom primjeru:

*"why?" that (is ("who?" Monika, "what?" glad),  
aren't (losing ("who?" workers, "what?" jobs ("whose?" their ))))*

Algoritam definiran za kreiranje odgovora od pronađenih čvorova ispisuje odgovor najprije ispisujući procesne čvorove, a zatim se ispisuju ostali čvorovi koje se nalaze na nižim razinama hijerarhije. Ispis ostalih čvora provodi se od najniže razine hijerarhije (od unutarnjih prema vanjskim zgradama). Ukoliko neka zagrada sadrži dva i više čvorova, čvorovi se ispisuju s lijeva na desno. U ovom primjeru algoritam je ispisao procesne čvorove „that“, „aren't“ i „losing“, a zatim čvorove ispod njih redom „workers their jobs“. Problem je nastao u ispisu

subjekta „workers“, koji se nalazi u hijerarhiji ispod procesnih čvorova. Kako u trenutno definiranim formaliziranim FNOK zapisima nije pravilima definirano i predviđeno određivanje dijelova rečenice (subjekt, predikat, objekt i dr.) pa nije definiran ni algoritam kojim bi se subjekt smještao na predviđeno mjesto u odgovoru. Rješenje ovog problema preostaje za buduća istraživanja.

### 7.3.3. Djelomično točan odgovor

Djelomično točan odgovor dobiven je na 11 od 280 postavljenih pitanja (3,93 %). prikazuje primjere pitanja za koje je dobiven djelomično točan odgovor. Primjere pitanja na koje su dobiveni djelomično točni odgovori prikazuje Tablica 7.4.

**Tablica 7.4. Primjeri pitanja s djelomično točnim odgovorima**

R.B.	QTENG	Očekivani odgovor	Odgovor sustava	Analiza odgovora
1.	What did he colour?	heart, a heart, it	it	Djelomično točno
2.	Where I will go?	there, new park	there	Djelomično točno
3.	Who love his job?	Marko, he	he	Djelomično točno
4.	Who loves her hobby?	she, Sanja	she	Djelomično točno
5.	What they could do?	nothing, but swallow their pride	nothing	Djelomično točno

U svim pitanjima koja su kategorizirana kao „djelomično točna“, kao što su i ova u primjerima iz tablice, u odgovoru nedostaje element koji ne može biti iščitan niti iz postavljenog pitanja niti odgovora. Razlozi djelomičnih odgovora mogu se svrstati u tri skupine:

- odgovor je objekt koji se nalazi u drugom dijelu rečenice,
- odgovor je subjekt koji se nalazi u drugom dijelu rečenice,
- odgovor se sastoji od elemenata zavisne rečenice.

U prvom i drugom primjeru (Tablica 7.4) očekivana su dva odgovora od kojih je sustav odgovorio samo na jedan od njih. Za drugo pitanje sustav je pronašao odgovor u rečenici „When the new park opens, I'll go there every day.“ Očekivani odgovori koje bi čovjek odgovorio su „new park“ i „there“. „There“ se nalazi u drugom dijelu rečenice i upućuje na to da se radnja odvija u „new park“ koji se nalazi u prvom dijelu rečenice. Međutim, iz formaliziranog zapisa nije moguće pronaći tu vezu, jer glagol „go“ koji se pojavljuje u pitanju, u hijerarhijskoj je vezi s objektom „there“, ali nije u vezi s „new park“ koji je pak u hijerarhijskoj

vezi s glagolom „*opens*“. Ove veze vidljive su i u zeleno osjenčanim dijelovima pitanja i odgovora koje prikazuje Slika 7.12.

✓ [99-59] When the new park opens, I'll go there every day. ^

"when?" when (opens ("what?" park ("what?" new ("art?" the))), will (go ("who?" I, "where?" there, "when?" day ("what?" every))))

ODGOVOR: there

PARENT	QUESTION	NODE
[1] when	-	[2] opens
[2] opens	what	[3] park
[3] park	what	[4] new
[4] new	art	[5] the
[1] when	-	[6] will
[6] will	-	[7] go
[7] go	who	[8] I
[7] go	where	[9] there
[7] go	when	[10] day
[10] day	what	[11] every

PARENT	QUESTION	NODE
go	&	will
go	-	I
go	where	X

**Slika 7.12. Odgovor na pitanje "Where will I go?"**

U trećem i četvrtom primjeru, uzrok nedobivanja odgovora je sličan kao u prethodna dva, samo što su u ovom slučaju očekivana dva subjekta, odnosno subjekt iz prve rečenice i njegova zamjenica iz druge rečenice. Na primjer na pitanje „*Who loves his job?*“ sustav je dao odgovor „*he*“ iako bi čovjek odgovorio i „*Marko*“ jer je čvor „*he*“ hijerarhijski povezan s glagolom „*love*“ koji se nalazi u pitanju, a čvor „*Marko*“ s glagolom „*is*“ (Slika 7.13).

✓ [99-35] Marko is happy because he love his job. ^

"why?" because (is ("who?" Marko, "what?" happy), love ("who?" he, "what?" job ("whose?" his)))

ODGOVOR: he

PARENT	QUESTION	NODE
[1] because	-	[2] is
[2] is	who	[3] Marko
[2] is	what	[4] happy
[1] because	-	[5] love
[5] love	who	[6] he
[5] love	what	[7] job
[7] job	whose	[8] his

PARENT	QUESTION	NODE
love	who	X
love	-	job
job	-	his

**Slika 7.13. Odgovor na pitanje "Who loves his job?"**

U ova dva slučaja u kojima odgovori nisu pronađeni, odgovori su subjekt ili objekt iz druge rečenice koji nisu direktno povezani s glagolom koji se nalazi u pitanju, odnosno ne



nalaze se u hijerarhiji ispod tog glagola. To je i razlog što odgovor na pitanje tada nije pronađen. Kao rješenje ovog problema potrebno je naći rješenje za povezivanja različitih rečeničnih dijelova dviju zavisno složenih rečenica. Ovaj problem ne može se riješiti samo na razini algoritma za traženje odgovora, već zahtjeva nadogradnju pravila transformacije rečenica u formalizirani FNOK zapis, ali i nadogradnju algoritama transformacije u BPNOK. Iako postoje definirana pravila za transformaciju priloga (na primjer „there“ u drugom primjeru) i zamjenica (osobne zamjenice „it“, „he“ i „she“ u prvom, trećem i četvrtom primjeru) u formalizirani FNOK zapis, prethodna četiri primjera su pokazala da, s obzirom da ove riječi upućuju na imenicu ili je mijenjaju, bit će potrebno proširiti pravila transformacije ovih vrsta riječi kako bi formalizirani zapis sačuvao tu vezu. Posebno se to odnosi na osobne zamjenice koje najčešće mijenjaju subjekt u zavisnoj rečenici, dok kod ostalih zamjenica (povratnih, posvojnih, pokaznih, neodređenih i drugih) nisu uočeni slični problemi i njihova uloga u rečenici je u testiranim primjerima bila jasna i nije utjecala na ispravnost odgovora.

U petom primjeru na pitanje „What they could do?“ sustav je odgovorio samo „nothing“ (Slika 7.14), a očekivani odgovor je bio „nothing, but swallow their pride“. U ovom primjeru sustav je dao odgovor iz prvog dijela rečenice, a čovjek je očekivao uz to i cijelu zavisnu rečenicu u odgovoru. Kroz ovaj i slične primjere primijećeno je da su ovakvi slučajevi prisutni kada dvije rečenice povezuje veznik „but“ jer se njime u značenju rečenice isključuje sadržaj druge rečenice. Ovo dovodi u zaključka da će biti potrebno proširiti sustav i uvesti još jedan tip veznika, a time onda proširiti i algoritam traženja odgovora ovisno o njemu.

✓ [99-47] There was nothing they could do but swallow their pride. ^

what? but (was (could (do ("who?" they, "where?" there, "what?" nothing))), swallow ("what?" pride ("whose?" their)))

**ODGOVOR:** nothing

PARENT	QUESTION	NODE
[1] but	-	[2] was
[2] was	-	[3] could
[3] could	-	[4] do
[4] do	who	[5] they
[4] do	where	[6] there
[4] do	what	[7] nothing
[1] but	-	[8] swallow
[8] swallow	what	[9] pride
[9] pride	whose	[10] their

PARENT	QUESTION	NODE
do	&	could
do	-	they
do	what	X

**Slika 7.14. Odgovor na pitanje „What they could do?“**

### 7.3.4. Nema odgovora

Od testiranih 280 pitanja, za njih 29 (10,36 %) sustav nije dao odgovor, odnosno kao odgovor ispisuje „NEMA ODGOVORA“. Za 28 pitanja sustav je pronašao rečenicu u kojoj se odgovor nalazi, a samo za jedno pitanje rečenica nije pronađena.

Među pitanjima na koja sustav nije dao odgovor uočeni su slični razlozi nedobivanja odgovora i za svaki od njih je uključen primjer (Tablica 7.5).

Red. br.	QTENG	Očekivani odgovor	Rečenica u kojoj je odgovor
1.	Who talks to Tom?	Paul?	Tom and Paul talk about the solution.
2.	Who talks to Paul?	Tom?	Tom and Paul talk about the solution.
3.	What did we do to each other?	kissed and hugged	We kissed and hugged each other.
4.	What can you do if you get the point?	go here	You can go here if you get the point.
5.	How did he colour the heart?	red	He drew a heart and coloured it red.

**Tablica 7.5. Primjeri pitanja za koja nije pronađen odgovor, ali je pronađena rečenica u kojoj se odgovor nalazi**

Prva dva primjera su pitanja „*Who talks to Tom?*“ i „*Who talks to Paul?*“ za koja se odgovor može pronaći u rečenici „*Tom and Paul talk about the solution.*“ Na ova pitanja ni čovjek ne može dati odgovor sa stopostotnom sigurnošću i zato u tablici kraj odgovora „Tom“ i „Paul“ stoji upitnik „?“, jer rečenica tvrdi da Tom i Paul govore (talk) o istoj temi, ali može i ne mora značiti da oni razgovaraju međusobno.

U trećem i četvrtom primjeru očekivani odgovor je glagol, odnosno glagoli (3. primjer). U ovim pitanjima glagol koji je u pitanju ne nalazi se i u rečenici iz koje je očekivani odgovor, što je u ovim slučajevima uzrok nepronalaženja odgovora. U 3. primjeru „*What did we do to each other?*“ očekivan je odgovor „*kissed and hugged*“ koji je moguće pronaći u rečenici „*We kissed and hugged each other.*“ U prikazu odgovora sustava vidljivo je da glagol „do“ nema uparene čvorove s čvorovima u rečenici (Slika 7.15), a uz glagol „*kissed*“ i „*hugged*“ nema pitanja na koje bi se moglo odgovoriti.

× [99-9] We kissed and hugged each other.

"what?" and (kissed ("who?" we), hugged ("who?" other ("who?" each)))

NEMA ODGOVORA

PARENT	QUESTION	NODE	PARENT	QUESTION	NODE
[1] and	-	[2] kissed	X	-	do
[2] kissed	who	[3] we	do	&	did
[1] and	-	[4] hugged	do	-	we
[4] hugged	who	[5] other	do	-	other
[5] other	who	[6] each	other	-	each

Slika 7.15. Prikaz odgovora sustava na pitanje "What did we do to each other?"

U budućim istraživanjima bit će potrebno doraditi algoritam na način da može odgovoriti i na pitanje koje upućuje na procesni čvor koji u sebi sadrži glagol. U formaliziranom FNOK zapisu, prema pravilima korištenim u ovom radu, glagol kao procesni čvor nema pridruženo pitanje pa se pridruživanje pitanja glagolu nameće kao moguće rješenje ovakvih i sličnih slučajeva, a pronalazak konkretnog rješenja značit će i doradu kako formaliziranih FNOK zapisa tako i samih algoritama za traženje odgovora.

Slika 7.16 prikazuju odgovor sustava na pitanje „What they swallowed?“, jedino na koje sustav nije dao odgovor i nije pronađena rečenica u kojem se odgovor potencijalno nalazi („There was nothing they could do but swallow their pride.“).

## Odgovori na pitanja

**Pitanje**

What they swallowed? Odaberi

---

**Formalizirani zapis**  
swallowed ( \_ they, "what?" X)

Answered: 0.53437495231628

**Hijerarhijski zapis pitanja**

Node ID	Node	Question	Parent ID
1	swallowed	-	0
2	they	-	1
3	X	what	1

**Obrađeno pitanje**

ID	PAR_WRD_ID	NOD_QST_ID	NOD_WRD_ID
571	635 swallowed swallowed swallow swallows	-1 -	402 they they
572	635 swallowed swallowed swallow swallows	2 what	579 X X

× [99-11] He drew a heart and coloured it red. ↓

× [99-2] Ana and John play the violin. ↓

× [99-42] Vanessa is sleepy because she has been studying all night. ↓

× [99-4] Sanja is training volleyball and tennis. ↓

× [99-5] Peter trains basketball, volleyball and tennis. ↓

Slika 7.16. Odgovor sustava na pitanje "What they swallowed?"

Analiza formaliziranih FNOK i QFNOK zapisa rečenice i pitanja (Slika 7.17) pokazala je da odgovor nije pronađen zbog toga što se subjekt „they“ nalazi u prvom dijelu rečenice „*There was nothing they could do*“ dok ga u drugom dijelu rečenice „*swallow their pride*“, na koji se odnosi pitanje, nema. Stoga u rečenici nema pronađenih čvorova iz koji bi generirali odgovor. I u ovom slučaju bi određivanje dijelova rečenice moglo biti potencijalno rješenje problema, a istražiti će se u budućim istraživanjima.

× [99-47] There was nothing they could do but swallow their pride. ^

what? but (was (could (do ("who?" they, "where?" there, "what?" nothing))), swallow ("what?" pride ("whose?" their)))

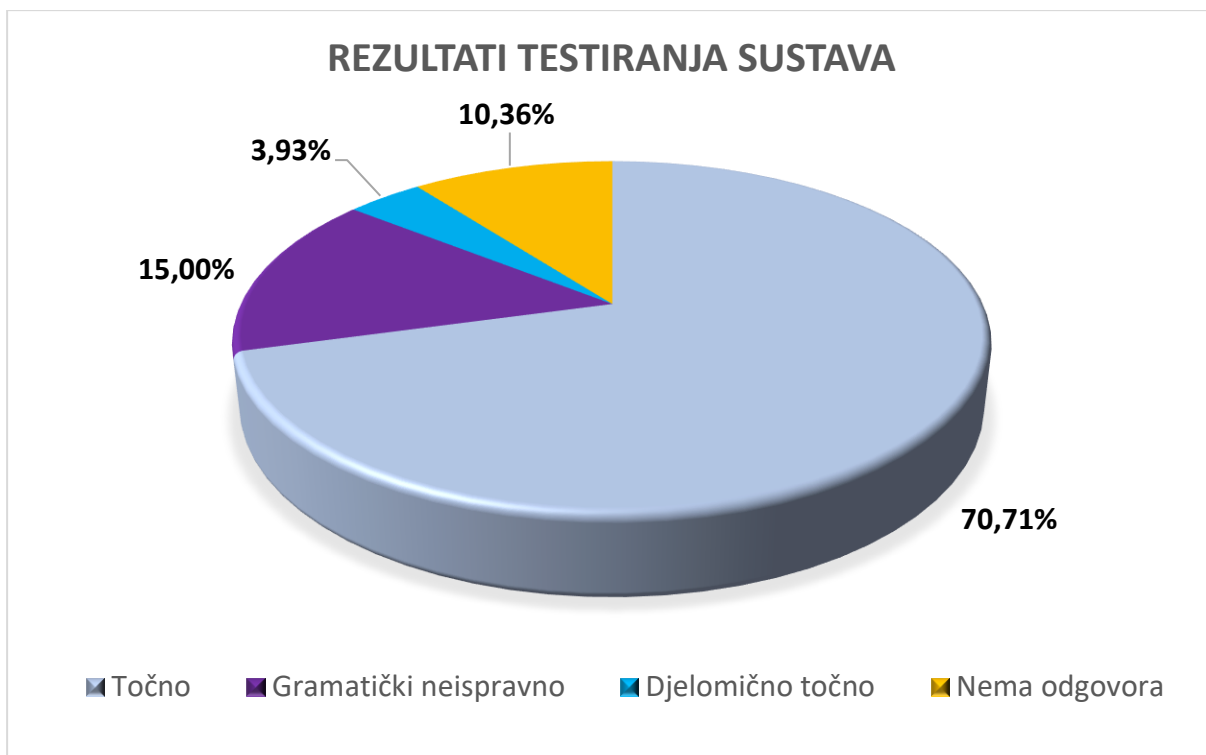
NEMA ODGOVORA

PARENT	QUESTION	NODE	PARENT	QUESTION	NODE
[1] but	-	[2] was	swallowed	-	they
[2] was	-	[3] could	swallowed	what	X
[3] could	-	[4] do			
[4] do	who	[5] they			
[4] do	where	[6] there			
[4] do	what	[7] nothing			
[1] but	-	[8] swallow			
[8] swallow	what	[9] pride			
[9] pride	whose	[10] their			

**Slika 7.17. Pregled pronađenih čvorova u rečenici „*There was nothing they could do but swallow their pride.*“ za pitanje „*What they swallowed?*“**

### 7.3.5. Zaključci analize rezultata

Sustav je testiran na 280 upitnih rečenica odnosno pitanja prirodnog jezika na koje je sustav tražio odgovore među 100 rečenica prirodnog jezika. Na 198 pitanja sustav je odgovorio točno (70,71 %), na 42 pitanje odgovori su bili gramatički neispravni ali razumljivi čovjeku (15,00 %), na 11 pitanja odgovor je bio djelomično točan (3,93 %), a na 29 pitanja sustav nije dao odgovor (10,36 %) što je prikazano na Grafikonu 1.



**Grafikon 1. Rezultati testiranja sustava**

S obzirom da se gramatički neispravne odgovore može smatrati točnima, može se zaključiti da je sustav odgovorio na 85,71 % pitanja što ukazuje na njegovu djelotvornost. Analiza dobivenih odgovora ukazala je na potrebna poboljšanja i moguće nadogradnje sustava NOK u budućim istraživanjima:

- definiranje sintaktičkih dijelova rečenice s ciljem točnijeg definiranja odgovora na pitanja i redoslijeda riječi u dobivenim odgovorima,
- preispitivanje uloge procesnih čvorova, posebno glagola i pravila njihove transformacije u formalizirani zapis (dodavanje uloge/pitanja glagolu),
- definiranje posebnih pravila za transformaciju osobnih zamjenica s obzirom na njihovu specifičnu ulogu kada je u pitanju traženje subjekta u rečenici.

#### **7.4. Dokaz hipoteze H2**

Hipoteza H2 glasi: *Koristeći prirodni upitni jezik, relacijska BPNOK može pronaći i formulirati odgovore na postavljena pitanja.*

Hipoteza će se smatrati potvrđenom ako se NOK metodom i definiranim pravilima upitne rečenice engleskog jezika transformirane u QFNOK zapišu u relacijsku bazu podataka BPNOK i na temelju njih izvršenim propitivanjem dobiju odgovori na postavljena pitanja.

U poglavlju 0 definirana su pravila transformacije upitnih rečenica prirodnog jezika u formalizirani QFNOK zapis. Ova pravila primijenjena su na 280 upitnih rečenica engleskog jezika i transformacijom su dobiveni formalizirani QFNOK zapisi (Privitak 8). Algoritmima

opisanim i definiranim u poglavlju 6.2. sva pitanja, odnosno njihovi formalizirani QFNOK zapisi transformirani su u relacijsku bazu podataka BPNOK i upisani u tablice NOK\_QFNOK (Slika 7.3.) i KB\_TEXT\_QUEST (Slika 7.5.) čime su pitanja pripremljena za traženje odgovora.

Testiranjem na skupu od 280 upitnih rečenica, sustav je pronašao točne odgovore (Privitak 9) jednake očekivanima za 71,43 % pitanja. Za 14,64 % pitanja sustav je dao gramatički neispravne odgovore, ali ti odgovori su semantički razumljivi čovjeku i time u skladu s očekivanim odgovorima.

Kao primjer u nastavku će se prikazati sva pitanja i dobiveni odgovori za rečenicu „I didn't go out because I had no money.“ Tablica 7.6 prikazuje pitanja (u stupcu QTENG) koja su postavljena za tu rečenicu i pripadajuće formalizirane QFNOK zapise za svako testirano pitanje.

**Tablica 7.6. Pitanja postavljena za rečenicu „I didn't go out because I had no money.“ i pripadajući QFNOK-ovi**

RB	Pitanje (QTENG)	QFNOK
1.	Who didn't go out?	didn't (go ("who?" X, _ out))
2.	Where I didn't go?	didn't ( go ( _ I, "where?" X))
3.	Who had no many?	had ("who?" X, _ money ( _ no))
4.	What I don't had?	had ( & don't) ( _ I, "what?" X)
5.	Who didn't go out?	go (& didn't) ("who?" X, _ out)
6.	Why didn't I go out?	didn't (go ( _ I, _ out), "why?" X)

Tablica 7.7 prikazuje očekivane odgovore, odgovore sustava i njihovu analizu za pitanja koja prikazuje Tablica 7.6. Od 6 postavljenih pitanja na 5 je sustav odgovorio identično očekivanom odgovoru. Šesti odgovor, iako u njemu nije točan redoslijed riječi, možemo smatrati točnim jer je razumljiv čovjeku i sadrži iste riječi kao i očekivani odgovor.

**Tablica 7.7. Odgovori na pitanja postavljena za rečenicu „I didn't go out because I had no money.“ i usporedba s očekivanim odgovorima**

Red. br.	QTENG	Očekivani odgovor	ANSWER sustava	Analiza odgovora
1.	Who didn't go out?	I	I	Točno
2.	Where I didn't go?	out	out	Točno
3.	Who had no many?	I	I	Točno
4.	What I don't had?	no money	no money	Točno
5.	Who didn't go out?	I	I	Točno
6.	Why didn't I go out?	because I had no money	because had I no money	Gramatički neispravno

Provedenim testiranjem je pokazano da se postavljanjem pitanja korištenjem prirodnog upitnog jezika koji se prema definiranim pravilima transformiraju u formalizirani

QFNOK zapis a kojeg je moguće upisati u relacijsku bazu podataka BPNOK, mogu pronaći i formulirati odgovori na postavljena pitanja čime je potvrđena hipoteza H2.

## 7.5. Testiranje implementacije sinonima

U ovom radu implementirano je rješenje problema sinonima što je opisano u poglavlju 6.1.1. u kojem su prikazani primjeri pitanja i rečenica (Tablica 6.1) koje koriste glagole „*talk*“ i „*discuss*“. S obzirom da je glagol *discuss* sinonim glagolu *talk*, ali ne i obrnuto, razlikuju se i očekivani odgovori na postavljena pitanja. Među testiranim rečenicama su dvije rečenice od kojih jedna koristi glagol „*talk*“ („*Tom and Paul talk about the solution.*“), a druga glagol „*discuss*“ (*Lucija and Ana discuss about the weather.*) Na postavljeno pitanje „*Who is talking?*“ očekivani i dobiveni odgovori su „*Lucija and Ana*“ i „*Tom and Paul*“ dok je za pitanje „*Who is discussing?*“ očekivan i dobiven samo jedan odgovor, odnosno „*Lucija and Ana*“.

Slika 7.18 prikazuje odgovor sustava na pitanje „*Who is talking?*“ i odgovor je pronađen u obje prethodno navedene rečenice. U dijelu korisničkog sučelja „obrađeno pitanje“ vidljivo je da je pronađen sinonim „*discuss*“ i njegov oblik „*discussing*“ kojeg algoritam pronalazi u rječniku, u tablici KB\_SYNONIM. Zbog pronađenog sinonima traži se odgovor i na pitanje „*Who is discussing?*“ koje u ovom slučaju nije bilo postavljeno. Kad se postavi pitanje „*Who is discussing?*“ sinonim neće biti pronađen jer preslikavanje nije bijektivno i neće ga biti niti u dijelu sučelja „obrađeno“ pitanje“ (Slika 7.19).

**Pitanje**

Who is talking? Odaberi

---

**Formalizirani zapis**  
 talking ( & is) ( "who?" X)

Answerid: 0.55514001846313

### Hijerarhijski zapis pitanja

Node ID	Node	Question	Parent ID
1	talking	-	0
2	is	&	1
3	X	who	1

### Obrađeno pitanje

ID	PAR WRD ID	NOD QST ID	NOD WRD ID
21	295 talking talking talk talks talking_to discussing discuss discuss	4 &	149 is is are be was were am would been
22	295 talking talking talk talks talking_to discussing discuss discuss	3 who	579 X X X

✓ [99-100] Lucija and Ana discuss about the weather. ^

discuss ("who?" and ("who?" Lucija, "who?" Ana), "what?" weather ("art?" the, "what?" about))

**ODGOVOR:** Lucija and Ana

PARENT	QUESTION	NODE
[1] discuss	who	[2] and
[2] and	who	[3] Lucija
[2] and	who	[4] Ana
[1] discuss	what	[5] weather
[5] weather	art	[6] the
[5] weather	what	[7] about

PARENT	QUESTION	NODE
talking	&	is
talking	who	X

✓ [99-1] Tom and Paul talk about the solution. ^

talk ("who?" and ("who?" Tom, "who?" Paul), "what?" solution ("art?" the, "what?" about))

**ODGOVOR:** Tom and Paul

PARENT	QUESTION	NODE
[2] and	who	[3] Tom
[2] and	who	[4] Paul
[1] talk	what	[5] solution
[5] solution	art	[6] the
[5] solution	what	[7] about
[1] talk	who	[2] and

PARENT	QUESTION	NODE
talking	&	is
talking	who	X

**Slika 7.18. Odgovor sustava na pitanje „Who is talking?“**



**Pitanje**

Who is discussing? Odaberi

---

**Formalizirani zapis**  
 discussing ( & is) ("who?" X)

Answered: 0.58258199691772

### Hijerarhijski zapis pitanja

Node ID	Node	Question	Parent ID
1	discussing	-	0
2	is	&	1
3	X	who	1

### Obrađeno pitanje

ID	PAR_WRD_ID	NOD_QST_ID	NOD_WRD_ID
1147	609 discussing discussing discuss discuss	4 &	149 is is are be was were am would been
1148	609 discussing discussing discuss discuss	3 who	579 X X

✓ [99-100] Lucija and Ana discuss about the weather. ^

discuss ("who?" and ("who?" Lucija, "who?" Ana), "what?" weather ("art?" the, "what?" about))

**ODGOVOR:** Lucija and Ana

PARENT	QUESTION	NODE
[1] discuss	who	[2] and
[2] and	who	[3] Lucija
[2] and	who	[4] Ana
[1] discuss	what	[5] weather
[5] weather	art	[6] the
[5] weather	what	[7] about

PARENT	QUESTION	NODE
discussing	&	is
discussing	who	X

✗ [99-11] He drew a heart and coloured it red. v

**Slika 7.19. Odgovor sustava na pitanje „Who is discussing?“**

U prethodnom primjeru prikazani su sinonimi za glagole, ali na isti način su sinonimi implementirani i za ostale vrste riječi. Na primjer, pridjevi „happy“ i „glad“ povezani su na način da je „happy“ sinonim od „glad“, ali ne i obrnuto. Ove pridjeve koriste rečenice:

- *Monika is glad that workers aren't losing their jobs.*
- *Sanja is happy because she loves her interesting hobby.*
- *Marko is happy because he loves his job.*

Na pitanje „Who is happy?“ sustav je odgovorio Sanja i Marko, ali ne i Monika, dok je na pitanje „Who is glad?“ odgovorio i Sanja i Marko i Monika (Slika 7.20).

**Pitanje**

Who is happy? Odaberi

---

**Formalizirani zapis**  
 is ("who?" X, \_ happy)

Answered: 0.57111597861157

**Hijerarhijski zapis pitanja**

Node ID	Node	Question	Parent ID
1	is	-	0
2	X	who	1
3	happy	-	1

**Obrađeno pitanje**

ID	PAR WRD ID	NOD_QST_ID	NOD_WRD_ID
437	149 is is are be was were am would been	3 who	579 X X
438	149 is is are be was were am would been	-1 -	303 happy happy

✓ [99-35] Marko is happy because he love his job. ^

"why?" because (is ("who?" Marko, "what?" happy), love ("who?" he, "what?" job ("whose?" his)))

**ODGOVOR:** Marko

PARENT	QUESTION	NODE
[1] because	-	[2] is
[2] is	who	[3] Marko
[2] is	what	[4] happy
[1] because	-	[5] love
[5] love	who	[6] he
[5] love	what	[7] job
[7] job	whose	[8] his

PARENT	QUESTION	NODE
is	who	X
is	-	happy

---

✓ [99-36] Sanja is happy because she love her interesting hobby. ^

"why?" because (is ("who?" Sanja, "what?" happy), loves ("who?" she, "what?" hobby ("whose?" her, "what?" interesting)))

**ODGOVOR:** Sanja

PARENT	QUESTION	NODE
[1] because	-	[2] is
[2] is	who	[3] Sanja
[2] is	what	[4] happy
[1] because	-	[5] loves
[5] loves	who	[6] she
[5] loves	what	[7] hobby
[7] hobby	whose	[8] her
[7] hobby	what	[9] interesting

PARENT	QUESTION	NODE
is	who	X
is	-	happy

**Slika 7.20. Odgovor na pitanje "Who is happy?"**

**Pitanje**

Who is glad? Odaberi

---

**Formalizirani zapis**  
 is ("who?" X, \_ glad)

Answered: 0.72984409332275

**Hijerarhijski zapis pitanja**

Node ID	Node	Question	Parent ID
1	is	-	0
2	X	who	1
3	glad	-	1

**Obrađeno pitanje**

ID	PAR_WRD_ID	NOD_QST_ID	NOD_WRD_ID
1029	149 is is are be was were am would been	3 who	579 X X
1030	149 is is are be was were am would been	-1 -	467 glad glad happy

✓ [99-35] Marko is happy because he love his job. ^

"why?" because (is ("who?" Marko, "what?" happy), love ("who?" he, "what?" job ("whose?" his)))

ODGOVOR: Marko

PARENT	QUESTION	NODE
[1] because	-	[2] is
[2] is	who	[3] Marko
[2] is	what	[4] happy
[1] because	-	[5] love
[5] love	who	[6] he
[5] love	what	[7] job
[7] job	whose	[8] his

PARENT	QUESTION	NODE
is	who	X
is	-	glad

✓ [99-36] Sanja is happy because she love her interesting hobby. ^

"why?" because (is ("who?" Sanja, "what?" happy), loves ("who?" she, "what?" hobby ("whose?" her, "what?" interesting)))

ODGOVOR: Sanja

PARENT	QUESTION	NODE
[1] because	-	[2] is
[2] is	who	[3] Sanja
[2] is	what	[4] happy
[1] because	-	[5] loves
[5] loves	who	[6] she
[5] loves	what	[7] hobby
[7] hobby	whose	[8] her
[7] hobby	what	[9] interesting

PARENT	QUESTION	NODE
is	who	X
is	-	glad

✓ [99-89] Monika is glad that workers aren't losing their jobs. ^

"why?" that (is ("who?" Monika, "what?" glad), aren't (losing ("who?" workers, "what?" jobs ("whose?" their ))))

ODGOVOR: Monika

PARENT	QUESTION	NODE
[1] that	-	[2] is
[2] is	who	[3] Monika
[2] is	what	[4] glad
[1] that	-	[5] aren't
[5] aren't	-	[6] losing
[6] losing	who	[7] workers
[6] losing	what	[8] jobs
[8] jobs	whose	[9] their

PARENT	QUESTION	NODE
is	who	X
is	-	glad

**Slika 7.21. Odgovor na pitanje "Who is glad?"**

## 7.6. Testiranje sustava na rečenicama i pitanjima hrvatskog jezika

Konceptualni okvir Node of Knowledge primjenjiv je na različite prirodne jezike, odnosno na bilo koje tekstualno znanje, a algoritmi ne ovise o sadržaju već strukturi formaliziranih zapisa. S obzirom na navedeno, sustav je testiran i na manjem setu rečenica i pitanja na hrvatskom jeziku. Za testiranje na hrvatskom jeziku, odabrano je nekoliko rečenica engleskog jezika koje su prevedene na hrvatski jezik. Odabrane rečenice engleskog jezika ranije su sudjelovale u testiranju na engleskom jeziku.

Tablica 7.8. prikazuje odabrane rečenice na engleskom jeziku (TENG), njima pripadajuće formalizirane zapise na engleskom jeziku, prijevod rečenica na hrvatski jezik (THJ) i pripadajuće FNOK zapise na hrvatskom jeziku.

*Tablica 7.8. Odabrane rečenice za testiranje na hrvatskom jeziku*

R.B.	TENG	FNOK	THJ	FNOK
1.	Sanja is training volleyball and tennis.	is (training ("who?" Sanja, "what?" and ("what?" volleyball, "what?" tennis)))	Sanja trenira odbojku i tenis.	trenira ("tko?" Sanja, "što?" i ("što?" odbojku, "što?" tenis))
2.	Marko is happy because he love his job.	"why?" because (is ("who?" Marko, "what?" happy), love ("who?" he, "what?" job ("whose?" his)))	Marko je sretan jer voli svoj posao.	"zašto?" jer (je ("tko?" Marko, "što?" sretan), voli ("što?" posao ("čiji?" svoj)))
3.	Monika is glad that workers aren't losing their jobs.	"what?" that (is ("who?" Monika, "what?" glad), aren't (losing ("who?" workers, "what?" jobs ("whose?" their )))	Moniki je drago što radnici neće izgubiti svoje poslove.	"zašto?" što (je ("kome?" Moniki, "što?" drago), neće (izgubiti ("tko?" radnici, "što?" poslove ("čije?" svoje )))
4.	The trip was canceled because of bad weather.	was ("what?" trip ("art?" the), "who?" canceled, "why?" because ("what?" weather ("how?" bad ("how?" of)))	Putovanje je otkazano zbog lošeg vremena.	je ("što?" putovanje, "što?" otkazano, "zašto?" zbog ("što?" vremena ("kakvog?" lošeg)))
5.	Saturday is the day when I get my hair done.	"when?" when (is ("what?" saturday, "what?" day ("art?" the)), get (done ("who?" I, "what?" hair ("whose?" my)))	Subota je dan kad perem svoju kosu.	"kad?" kad (je ("što?" subota, "što?" dan), perem ("što?" kosu ("čiju?" svoju)))

Rečenice na hrvatskom jeziku i njihovi formalizirani FNOK zapisi upisani u bazu podataka BPNOK, odnosno u tablicu NOK\_FNOK\_IMP i transformirani pokretanjem funkcija `processNOK` i `processNOKArray` u hijerarhiju čvorova u tablicama NOK\_FNOK i KB\_TEXT\_QUESTION. Slika 7.22 prikazuje hijerarhijski zapis u polje za rečenicu „Marko je sretan jer voli svoj posao.“, a Slika 7.23 njen hijerarhijski zapis u tablicu NOK\_FNOK.

# Unos u bazu podataka

[102] Marko je sretan jer voli svoj posao.

```
array (  
  0 =>  
    array (  
      'Q' => 'zasto',  
      'C' => 'jer',  
      'O' =>  
        array (  
          0 =>  
            array (  
              'V' => 'je',  
              'F' =>  
                array (  
                  0 =>  
                    array (  
                      'Q' => 'tko',  
                      'A' => 'Marko',  
                    ),  
                  1 =>  
                    array (  
                      'Q' => 'sto',  
                      'A' => 'sretan',  
                    ),  
                ),  
              ),  
            ),  
          ),  
        ),  
      ),  
    ),  
  1 =>  
    array (  
      'V' => 'voli',  
      'F' =>  
        array (  
          0 =>  
            array (  
              'Q' => 'sto',  
              'C' => 'posao',  
              'O' =>  
                array (  
                  'Q' => 'ciji',  
                  'A' => 'svoj',  
                ),  
              ),  
            ),  
          ),  
        ),  
      ),  
    ),  
),  
)
```

Slika 7.22. Hijerarhijski zapis rečenice „Marko je sretan jer voli svoj posao.“ u polje

PRG	ORD	NODE_ID	NODE	QUEST	PARENT...	KB_PAR_WRD_ID	KB_PAR_WRD	KB_NOD_WRD_ID	KB_NOD_QST_ID	ID
99	102	1	jer	zasto	0	(null)	(null)	649	27	2746
99	102	2	je	_	1	649	jer	650	-1	2747
99	102	5	voli	_	1	649	jer	652	-1	2750
99	102	4	sretan	sto	2	650	je	651	26	2749
99	102	3	Marko	tko	2	650	je	357	24	2748
99	102	6	posao	sto	5	652	voli	653	26	2751
99	102	7	svoj	ciji	6	653	posao	654	28	2752

Slika 7.23. Zapis rečenice „Marko je sretan jer voli svoj posao.“ u tablici NOK\_FNOK

Iz prethodnog primjera vidljivo je da je i složene rečenice hrvatskog jezika koje sadrže veznike moguće zapisati u relacijsku bazu podataka BPNOK. Iz tih rečenica mogu se dobiti odgovori na pitanja koji su također postavljena na hrvatskom jeziku. Odabrano je 10 pitanja za koja se odgovori mogu pronaći u odabranih pet rečenica (Tablica 7.8). Slika 7.24 prikazuje odgovor sustava na pitanje „Zašto je Marko sretan?“.

**Pitanje**

Zasto je Marko sretan? Odaberi

---

**Formalizirani zapis**  
je ( \_ Marko, \_ sretan, "zasto?" X)

Answered: 0.60846519470215

**Hijerarhijski zapis pitanja**

Node ID	Node	Question	Parent ID
1	je	_	0
2	Marko	_	1
3	sretan	_	1
4	X	zasto	1

**Obrađeno pitanje**

ID	PAR_WRD_ID	NOD_QST_ID	NOD_WRD_ID
1164	650 je je	-1 -	357 Marko Marko
1165	650 je je	-1 -	651 sretan sretan
1166	650 je je	27 zasto	579 X X

✓ [99-102] Marko je sretan jer voli svoj posao. ^

"zasto?" jer (je ("tko?" Marko, "sto?" sretan), voli ("sto?" posao ("ciji?" svoj)))

**ODGOVOR:** jer voli svoj posao

PARENT	QUESTION	NODE
[1] jer	_	[2] je
[2] je	tko	[3] Marko
[2] je	sto	[4] sretan
[1] jer	_	[5] voli
[5] voli	sto	[6] posao
[6] posao	ciji	[7] svoj

PARENT	QUESTION	NODE
je	_	Marko
je	_	sretan
je	zasto	X

**Slika 7.24. Odgovor na pitanje „Zašto je Marko sretan?“**

Tablica 7.9 prikazuje pitanja na engleskom jeziku (QTENG) i pripadajuće QFNOK-ove, pitanja na hrvatskom jeziku (QHJ) i pripadajuće QFNOK-ove, očekivane odgovore i odgovore koje je dao sustav te analizu odgovora. Na pet pitanja sustav je odgovorio točno, na četiri pitanja gramatički neispravno, a na jedno pitanje nije bilo odgovora. Iz analize odgovora koji su gramatički neispravni uočen je isti problem identificiran pri testiranju rečenica i pitanja na engleskom jeziku, a to je krivi redoslijed riječi u odgovoru što ne utječe na razumljivost odgovora (1., 5. 8. i 10. pitanje među testiranim primjerima). Ovaj problem detaljnije je opisan u poglavlju 7.3.2.

Tablica 7.9. Odgovori sustava na 10 postavljenih pitanja na hrvatskom jeziku

R.B.	QTENG	QFNOK	QHJ	QFNOK	Očekivani odgovor	Odgovor sustava	Analiza odgovora
1.	What does Sanja train?	train ( & does) ( _ Sanja, "what?" X)	Što trenira Sanja?	trenira ( _ Sanja, "što?" X)	odbojku i tenis	odbojku tenis i	Gramatički neispravno
2.	Who trains volleyball and tennis?	trains ("who?" X, _and ( _ volleyball, _ tennis))	Tko trenira odbojku i tenis?	trenira ("tko?" X, _ i ( _ odbojku, _ tenis))	Sanja	NEMA ODGOVORA	Nema odgovora
3.	What is Marko?	is ( _ Marko, "what?" X)	Što je Marko?	je ( _ Marko, "što?" X)	sretan	sretan	Točno
4.	Why is Marko happy?	is ( _ Marko, _ happy, "why?" X)	Zašto je Marko sretan?	je ( _ Marko, _ sretan, "zašto?" X)	jer voli svoj posao	jer voli svoj posao	Točno
5.	Why is Monika glad?	is ( _ Monika, _ glad, "why?" X)	Zašto je Moniki drago?	je ( _ Moniki, _ drago, "zašto?" X)	što radnici neće izgubiti svoje poslove	sto neće izgubiti radnici svoje poslove	Gramatički neispravno
6.	Who aren't losing their jobs?	aren't (losing ( "who?" X, _ jobs ( _ their))	Tko neće izgubiti svoje poslove?	neće (izgubiti ( "tko?" X, _ poslove ( _ svoje))	radnici	radnici	Točno
7.	What was cancelled?	was ("what?" X, _ canceled)	Što je otkazano?	je ("što?" X, _ otkazano)	putovanje	putovanje	Točno
8.	Why the trip was cancelled?	was ( _ trip ( _ the), _ canceled, "why?" X)	Zašto je putovanje otkazano?	je ( _ putovanje, _ otkazano, "zašto?" X)	zbog lošeg vremena	lošeg vremena zbog	Gramatički neispravno
9.	What is Saturday?	is ( _ Saturday, "what?" X)	Što je subota?	je ( _ subota, "što?" X)	dan	dan	Točno
10.	When I get my hair done?	get (done ( _ I, _ hair ( _ my)), "when?" X)	Kad perem svoju kosu?	perem ( _ kosu ( _ svoju)), "kad?" X)	u subotu	kad je subota dan	Gramatički neispravno

Pitanje na koje nije pronađen odgovor je ono koje u sebi sadrži veznik „i“. Problem je nastao zbog toga što u rječniku već postoji riječ „I“ koja na engleskom jeziku nije veznik već osobna zamjenica. U pitanju „Tko trenira odbojku i tenis?“ zbog algoritma traženja odgovora važno je „i“ definirati kao veznik i upisati ga u tablicu KB\_CONJ. U ovom slučaju to nije moguće jer u engleskom jeziku „I“ (ja) je zamjenica, a ne veznik. Može se zaključiti da je „i“ homonim za dva različita jezika. Implementacijom rješenja homonima u sustav NOK može se pretpostaviti da će sustav moći tražiti odgovore na pitanja neovisno na kojem je prirodnom jeziku pitanje postavljeno i na kojem je jeziku rečenica u kojoj se nalazi odgovor. Implementacija problema homonima riješit će se u budućim istraživanjima.



## 8. Zaključak

Ovaj doktorski rad nastao je kao nastavak istraživanja temeljenog na konceptualnom okviru Node of Knowledge (NOK) u okviru kojeg je izrađen sustav koji je testiran na jednostavnim rečenicama hrvatskog i engleskog jezika. Cilj ovog istraživanja bio je proširiti opseg primjene postojećeg sustava na odabrane složene rečenice engleskog jezika. Motivacija za proširivanje sustava temeljenog na NOK metodi je unaprijediti postojeći sustav na način da se proširi skup rečenica prirodnog jezika koje je moguće upisati u relacijsku bazu podataka i iz njih propitivanjem pitanjima prirodnog jezika dobiti odgovore. Promatrane su složene rečenice engleskog koje sadrže veznike i/ili se sastoje od dvije radnje (glagola). Ovo proširivanje značilo je proširivanje pravila transformacije rečenica i pitanja prirodnog jezika u formalizirane FNOK i QFNOK zapise, modela procesa, modela podataka, algoritama transformacije formaliziranih zapisa u relacijsku bazu podataka i algoritme traženja odgovora na pitanja.

Opseg istraživanja ograničen je na složene rečenice engleskog jezika i to one koje sadrže konačan skup odabranih veznika: veznike *and, or, but, because, if, when, that, unless, whenever, while, until* i *so*. U tim rečenicama veznici mogu imati različite uloge i to na sljedećim razinama: veznici spajaju subjekte, veznici spajaju predikate, veznici spajaju objekte, veznici spajaju dvije jednostavne rečenice od kojih svaka sadrži jedan glagol/predikat i subjekt i/ili ostale dijelove rečenice kao što je objekt i/ili priložne oznake. Uporaba veznika u rečenicama ima značajan utjecaj na složenost rečenice, pogotovo u slučaju kada veznici spajaju dvije rečenice.

Istraživanjem su dokazane dvije postavljene istraživačke hipoteze:

H1: Primjenom konceptualnog razvojnog okvira NOK, određene vrste složenih rečenica engleskog jezika se mogu transformirati u FNOK i relacijski BPNOK zapis.

H2: Koristeći prirodni upitni jezik, relacijska BPNOK može pronaći i formulirati odgovore na postavljena pitanja.

Provedbom istraživanja ostvareni su sljedeći znanstveni doprinosi:

- Analiza prednosti i nedostataka različitih varijanti hijerarhijskog odnosa glavnih i pomoćnih glagola u FNOK zapisu.
- Proširenje pravila prevođenja složenih rečenica u FNOK zapis (kao što su povratne zamjenice, veznici, više procesnih čvorova).
- Proširenje pravila prevođenja pitanja u QFNOK zapis (veznici, više procesnih čvorova).
- Proširenje modela procesa sustava za transformaciju rečenica u relacijsku bazu podataka.
- Proširenje meta modela podataka sustava za transformaciju složenih rečenica engleskog jezika u različitim glagolskim vremenima u relacijsku bazu podataka.
- Poboljšanje algoritama traženja odgovora na postavljeno pitanje.
- Određivanje tipa veznika kao elementa traženja odgovora na pitanja.
- Implementacija sinonima u rječnik i u proces traženja odgovora na pitanja.

- Putem korisničkog sučelja jasnije prikazivanje uparivanja čvorova u pitanju i rečenici te preglednije iščitavanje odgovora ili razloga nedobivanja odgovora.

Pregledom područja obuhvaćena su istraživanja iz područja: predstavljanja znanja, obrade prirodnog jezika, rječnika prirodnog jezika, sustava pitanja i odgovora (*Question Answering - QA*), velikih jezičnih modela (*Large Language Models – LLM*), grafova znanja (*Knowledge Graphs*) i relacijskih baza podataka kao elemenata QA sustava. Pregledom područja se željelo usporediti konceptualni okvir Node of Knowledge s aktualnim istraživanjima u području predstavljanja znanja. Najznačajnije što NOK metodu izdvaja od drugih metoda je što je usmjerena na konkretan, zadani opseg znanja i njegovo propitivanje na prirodnom jeziku čime je relevantnost dobivenih odgovora neupitna. Ostali sustavi većinom propituju velike količine znanja, za koje se ne može sa sigurnošću reći da je točno, što onda može utjecati i na točnost odgovora.

Kako bi se nadgradio postojeći sustav temeljen na NOK konceptualnom okviru prvo se nadgradilo i proširilo pravila transformacije rečenica i pitanja prirodnog jezika u formalizirane FNOK i QFNOK zapise, a definirana su i pravila prevođenja koja se primjenjuju na prevođenje veznika koji su temeljni dio opsega istraživanja. Prije same definicije pravila, primjena NOK metode na rečenice basni koje se sastoje upravo od složenih rečenica, dala je uvid u eventualne probleme i načine njihove transformacije u formalizirane zapise. Testirana je i PSA-NOK aplikacija kojom je moguće automatski transformirati jednostavne rečenice engleskog jezika, ali za transformaciju složenih rečenica potrebna je nadogradnja ove aplikacije.

Proširivanje sustava NOK započelo je proširivanjem rječnika koji se sastojao od tri elementa, odnosno tri tablice (relacije) u relacijskoj bazi podataka. S tim ciljem, u radu su opisani sljedeći meta modeli: meta model hijerarhije glagolskih vremena, meta model sinonima i meta model homonima. Proširivanje modela podataka i relacijske sheme baze podataka također je učinjeno na razini rječnika koji je proširen s još dva tipa entiteta: KB\_CONJ (*Dictionary conjunctions*) i KB\_SYNONYM (*Dictionary synonym*), čime su u rječnik implementirani veznici i sinonimi.

Korištenje i implementacija veznika u rečenicama i implementacija sinonima utjecala je na izmjene modela procesa transformacije formaliziranog teksta u FNOK-u u relacijsku bazu podataka i generiranje odgovora na pitanja.

Izmjene algoritama transformacije rečenica zapisanih u formaliziranim FNOK i QFNOK zapisima koji se prema NOK metodi zapisuju u relacijsku bazu podataka i algoritmi traženja i kreiranja odgovora na pitanja izrađeni su korištenjem PHP-a kojim je izrađeno i korisničko sučelje za jednostavnije i preglednije testiranje sustava. Za implementaciju relacijske baze podataka i dalje se koristi Oracle relacijska baza, a u PHP skripte su integrirane i neke SQL procedure koje se i dalje koriste iz prethodne verzije NOK sustava.

Testiranje je provedeno na ukupno 100 rečenica i 280 pitanja engleskog jezika koje zadovoljavaju definirani opseg istraživanja. Testiranje se provelo u dvije faze. Prva faza provedena je na 20 odabranih reprezentativnih rečenica i 72 pitanja postavljenih na njih, a

rečenice su odabrane prema kriteriju da se u barem jednoj od njih pojavljuju svi testirani veznici: *and, or, but, because, if, when, that, unless, whenever, while, until* i *so*. U ovoj fazi dopunjen je i obogaćen rječnik, odnosno tablica KB\_CONJUNCTION i uočena je potreba određivanja tipa veznika čime je omogućeno unaprjeđenje algoritma traženja odgovora na pitanja. Druga faza provedena je na svim odabranim rečenicama i pitanjima. Za svako pitanje unaprijed je definiran očekivani odgovor ili odgovori koje bi na postavljena pitanja dao čovjek, a koji su uspoređivani s odgovorima koje je dao sustav. Na 200 pitanja sustav je odgovorio točno (71,43 %), na 41 pitanje odgovori su bili gramatički neispravni ali razumljivi čovjeku (14,64 %), na 10 pitanja odgovor je bio djelomično točan (3,57 %), a na 29 pitanja sustav nije dao odgovor (10,36 %). U testiranim rečenicama koristili su se i sinonimi i za svaki korišteni sinonim provjera rezultata pokazala je da sustav daje očekivane odgovore za obje međusobno povezane riječi, neovisno o njihovoj vrsti.

Iako je fokus ovog istraživanja bio na rečenicama engleskog jezika, provedeno je i testiranje na rečenicama hrvatskog jezika. Željelo se pokazati da je konceptualni okvir Node of Knowledge primjenjiv na različite prirodne jezike, odnosno na bilo koje tekstualno znanje, a algoritmi ne ovise o sadržaju već strukturi formaliziranih zapisa. Prošireni sustav dodatno je testiran na pet rečenica i deset pitanja na hrvatskom jeziku. Za 9 od 10 pitanja, sustav je pokazao iste ili slične rezultate kao kod testiranja odgovarajućih rečenica na engleskom jeziku. Na jedno pitanje nije dobiven odgovor jer je riječ „i“ homonim u hrvatskom i engleskom jeziku, ali s različitim značenjem i pripadanjem različitim vrstama riječi i različitim jezicima. Implementacijom rješenja homonima na razini različitih jezika omogućilo bi se dobivanje odgovora na postavljena pitanja neovisno o prirodnom jeziku koji je korišten u pitanju i rečenici.

Ovim istraživanjem uočeni su problemi, ali i moguća poboljšanja i unaprjeđenja NOK sustava. Neki od planova za buduća istraživanja su sljedeći:

- Implementacija homonima na razini jednog prirodnog jezika, ali i različitih prirodnih jezika s ciljem dobivanja odgovora na pitanja u različitim jezicima. Dakle, pitanje može biti postavljeno na jednom jeziku, a odgovor može biti pronađen u rečenicama na drugom prirodnom jeziku. Drugim riječima, sustav bi bio neovisan o prirodnom jeziku na kojem je pitanje postavljeno i prirodnom jeziku u kojem je zapisana rečenica u kojoj se odgovor nalazi.
- Definiranje sintaktičkih dijelova rečenice s ciljem točnijeg definiranja odgovora na pitanja i redoslijeda riječi u njima.
- Definiranje posebnih pravila za transformaciju osobnih zamjenica s obzirom na njihovu specifičnu ulogu kada je u pitanju traženje subjekta u rečenici.
- Preispitivanje uloge procesnih čvorova, posebno glagola i pravila njihove transformacije u formalizirani zapis (dodavanje uloge / pitanja glagolu).
- Automatizacija transformacije rečenica prirodnog jezika u FNOK zapise i nadogradnja korisničkog sučelja za omogućavanje unosa rečenica i pitanja te pripadajućih formaliziranih zapisa (FNOK i QFNOK) u bazu podataka.

- Nadogradnja automatizacije transformacije složenih rečenica u formalizirane FNOK zapise u aplikaciji PSA-NOK, a u planu je i direktna integracija u NOK sustav.

## Literatura

- [1] M. Ašenbrener Katić, *Sustav za integraciju relacijske baze podataka i jednostavnih rečenica prirodnog jezika, doktorska disertacija*, Rijeka: Odjel za informatiku, Sveučilište u Rijeci, 2017.
- [2] P. McCorduck i C. Cfe, *Machines who think: A personal inquiry into the history and prospects of artificial intelligence*, CRC Press, 2004.
- [3] S. J. Russell i P. Novig, *Artificial Intelligence - A Modern Approach*, New Jersey: Prentice Hall, Englewood Cliffs, 1995.
- [4] R. Lujić, T. Šarić i G. Šimunović, »Primjena ekspertnog sustava pri određivanju klase prioriteta,« *Tehnički vijesnik 14, svez. 1, br. 2*, pp. 66-75, 2007.
- [5] N. Majdandžić, R. Lujić, G. Matičević, G. Šimunović i I. Majdandžić, *Upravljanje proizvodnjom*, Slavonski Brod: Strojarski fakultet Slavonski Brod, 2001.
- [6] R. Reddy, »The challenge of artificial intelligence,« *Computer, svez. 10, br. 10*, pp. 86-98, 1996.
- [7] A. Turing, »Computing machinery and Intelligence,« *Mind 59*, pp. 433-460, 1950.
- [8] S. M. Shieber, »Lessons from a restricted Turing test,« *Communications of the ACM 37.6*, pp. 70-79, 1994.
- [9] J. McCarthy, M. Minsky, N. Rochester i C. Shannon, »A proposal for the Dartmouth summer research project on artificial intelligence, August 31, 1955.,« *AI Magazine 27(4)*, 2006.
- [10] K. Warwick i H. Shah, »Can machines think? A report on Turing test experiments at the Royal Society,« *Journal of Experimental & Theoretical Artificial Intelligence 28.6*, pp. 989-1007, 2016.
- [11] L. Morgenstern i R. H. Thomason, »Teaching knowledge representation: Challenges and proposals,« *Ann Arbor 1001*, pp. 48109-2210, 2000.
- [12] A. Jakupović, M. Pavlić i Z. Dovedan Han, »Formalisation method for the text expressed knowledge,« *Expert Systems with Applications, 41(11)*, pp. 5308-5322, 2014.
- [13] A. Jakupović, M. Pavlić, A. Meštrović i V. Jovanović, »Comparison of the Nodes of Knowledge method with other graphical methods for knowledge representation,« u *Proceedings of the 36th international convention (CIS), MIPRO, Opatija*, 2013.
- [14] M. Mugnier i M. Chein, »Graph-based Knowledge Representation,« *Advanced Information and Knowledge Processing. Springer, London*, 2009.
- [15] H. Helbig, »Knowledge Representation and the Semantics of Natural Language,« *Springer*, 2006.

- [16] M. Pavlić, M. Ašenbrener, A. Jakupović, A. Meštović, S. Čandrić i M. Ivašić-Kos, »Inteligentni informacijski sustavi,« u *Razvoj poslovnih i informatičkih sustava CASE 25*, Rijeka, 2013.
- [17] M. Pavlić, »Development of a method for knowledge modeling,« Odjel za informatiku Sveučilišta u Rijeci, Rijeka, 2013.
- [18] M. Pavlić, A. Jakupović i A. Meštović, »Nodes of knowledge method for knowledge representation,« *Informatologija* 46,3, pp. 206-214, 2013.
- [19] M. Elallaoui, K. Nafil i R. Touahni, »Automatic transformation of user stories into UML use case diagrams using NLP techniques,« *Procedia computer science*, 130, pp. 42-49, 2018.
- [20] R. Bhagat, P. Pantel, E. Hovy i M. Rey, »LEDIR: An Unsupervised Algorithm for Learning Directionality of Inference Rules,« *In EMNLP-CoNLL*, pp. 161-170, 2007.
- [21] J. F. Sowa, »Future directions for semantic systems,« *In Intelligence-based systems engineering, Springer Berlin Heidelberg*, pp. 23-47, 2011.
- [22] P. Libbrecht, T. Declerck, T. Schlippe, T. Mandl i D. Schiffner, »LP for Student and Teacher: Concept for an AI based Information Literacy Tutoring System,« *CIKM (Workshops)*, 2020.
- [23] R. Baglini i H. Hjorth, »Natural Language Processing 4 All (NLP4All): A New Online Platform for Teaching and Learning NLP Concepts,« *Proceedings of the Fifth Workshop on Teaching NLP*, pp. 28-33, June 2021.
- [24] A. Lash, K. Murray i G. Mocko, »Natural language processing applications in requirements engineering.,« u *ASME 2012 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Vol. 45011*, 2012.
- [25] T. Güneş i F. B. Aydemir, »Automated goal model extraction from user stories using NLP,« *2020 IEEE 28th International Requirements Engineering Conference (RE) (pp. 382-387). IEEE.*, pp. 382-387, August 2020.
- [26] M. H. Amirhosseini, H. B. Kazemian, K. Ouazzane i C. Chandler, »Natural Language Processing approach to NLP Meta model automation,« u *2018 International Joint Conference on Neural Networks (IJCNN) (pp. 1-8)*, 2018.
- [27] J. Hale, L. Campanelli, J. Li, S. Bhattasali, C. Pallier i J. Brennan, »Neuro-computational models of language processing,« *Annual Review of Linguistics*, 2021.
- [28] M. Štorga, *Model rječnika za računalnu razmjenu informacija u distribuiranom razvoju proizvoda, doktorska disertacija*, Zagreb: Sveučilište u Zagrebu, Fakultet strojarstva i brodogradnje, 2005.
- [29] A. Slavić, *Semantički Web, sustavi za organizaciju znanja i mrežni standardi*, London: School of Library, Archives and Information Studies, University College London, 2004.

- [30] K. H. Veltman, »Syntactic and semantic interoperability: new approaches to knowledge and the semantic web,« *New Review of Information Networking 7.1*, pp. 159-183, 2001.
- [31] A. Meštrović i M. Čubrilo, »Monolingual dictionary semantic capturing using concept lattice,« *International Review on Computer and Software (IRECOS) 6.2*, pp. 173-184, 2011.
- [32] A. Meštrović, »Semantic Matching Using Concept Lattice,« *Proceedings of Concept Discovery in Unstructured Data, CDUD 2012, Katholieke Universitet Leuven, Belgium*, pp. 49-58.
- [33] A. Meštrović, *F-logika kao mehanizam za modeliranje rešetki za manipuliranje polustrukturiranim podacima i analizu koncepata, doktorska disertacija*, Varaždin: Fakultet organizacije i informatike, 2009.
- [34] »Croatian WordNet,« [Mrežno]. Available: <http://meta-share.ffzg.hr/repository/browse/croatian-wordnet/61471f206c1811e28a985ef2e4e6c59ead7d69526eb449efa116f9e2fe512d73/>. [Pokušaj pristupa 20. lipanj 2021.].
- [35] »WordNet,« [Mrežno]. Available: <https://wordnet.princeton.edu/>. [Pokušaj pristupa 20. lipanj 2021.].
- [36] G. Igalj, »Rječnik hrvatskih jezika,« [Mrežno]. Available: <https://github.com/gigaly/rjecnik-hrvatskih-jezika>. [Pokušaj pristupa 20. lipanj 2021.].
- [37] M. Caballero, »A brief survey of question answering systems,« *International Journal of Artificial Intelligence & Applications (IJAA)*, 12(5), 2021.
- [38] J. Prager, E. Brown, A. Coden i D. Radev, »Question-answering by predictive annotation,« *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 184-191, 2000.
- [39] E. H. Hovy, L. Gerber, U. Hermjakob, M. Junk i C. Y. Lin, »Question Answering in Webclopedia,« *TREC, Vol. 52*, pp. 53-56, 2000.
- [40] A. Iftene, *Textual Entailment. Ph.D Thesis*, Cuza University, Faculty of Computer Science, 2009.
- [41] B. Ojokoh i E. Adebisi, »A review of question answering systems,« *Journal of Web Engineering 17.8*, pp. 717-758, 2018.
- [42] A. M. Pundge, S. A. Khillare i C. N. Mahender, »Question answering system, approaches and techniques: a review,« *International Journal of Computer Applications*, 141(3), pp. 35-39, 2016.
- [43] A. Mishra i S. K. Jain, »A survey on question answering systems with classification,« *Journal of King Saud University-Computer and Information Sciences*, 28(3), pp. 345-361, 2016.

- [44] J. Kaur i V. Gupta, »Effective question answering techniques and their evaluation metrics,« *International Journal of Computer Applications* 65.12 , pp. 30-37, 2013.
- [45] S. Tirpude, »Closed Domain Question Answering System: A Survey,« *International Journal of Informative and Futuristic Research (IJIFR)* 2.9, pp. 334-335, 2015.
- [46] D. Yogis, T. N. Manjunath i R. S. Ravindra, » A survey of intelligent question answering system using nlp and information retrieval techniques,« *International Journal of Advanced Research in Computer and Communication Engineering*, 5(5), pp. 536-540, 2016.
- [47] D. Diefenbach, V. Lopez, K. Singh i P. Maret, »Core techniques of question answering systems over knowledge bases: a survey,« *Knowledge and Information systems*, 55, pp. 529-569, 2018.
- [48] T. Li, X. Ma, A. Zhuang, Y. Gu, Y. Su i W. Chen, »Few-shot In-context Learning on Knowledge Base Question Answering,« *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6966-6980, 2023.
- [49] B. Min, H. Ross, E. Sulem, A. P. B. Veyseh, T. H. Nguyen, O. Sainz, E. Agirre, I. Heinz i D. Roth, »Recent advances in natural language processing via large pre-trained language models: A survey,« *ACM Computing Surveys*, 56(2), pp. 1-40, 2023.
- [50] J. Liu, S. Cao, J. Shi, T. Zhang, L. Hou i J. Li, »Probing Structured Semantics Understanding and Generation of Language Models via Question Answering,« *arXiv:2401.05777*, 2024.
- [51] S. Bubeck, V. Chandrasekaran, R. Eldan, J. Gehrke, E. Horvitz, E. Kamar, P. Lee, Y. T. Lee, Y. Li, S. Lundberg, H. Nori, H. Palangi, M. T. Ribeiro i Y. Zhang, »Sparks of artificial general intelligence: Early experiments with gpt-4,« *arXiv preprint arXiv:2303.12712*, 2023.
- [52] 12 5 2024. [Mrežno]. Available: <https://chatgpt.com/>.
- [53] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser i I. Polusukhin, » Attention is all you need,« *Advances in neural information processing systems*, 30, 2017.
- [54] Y. Mehdi, »Reinventing search with a new AI-powered Microsoft Bing and Edge, your copilot for the web,« Microsoft, 2023.
- [55] S. Romano, R. Angius, N. Kerby, P. Bouchaud, J. Amidei i A. Kaltenbrunner, »A Dataset to Assess Microsoft Copilot Answers in the Context of Swiss, Bavarian and Hessian Elections,« *Proceedings of the International AAI Conference on Web and Social Media. Vol. 18.*, pp. 2040-2050, 2024.
- [56] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, n. Goyal, H. Kuttler, M. Lewis, W.-t. Yih, T. Rocktaschel, S. Riedel i D. Kiela, »Retrieval-augmented generation for knowledge-intensive nlp tasks,« *Advances in Neural Information Processing Systems*, 33, pp. 9459-9474, 2020.



- [57] S. A. Athaluri, S. V. Manthena, V. K. M. Keesapragada, V. Yarlagadda, T. Dave i R. T. S. Duddumpudi, »Yarlagadda, V., Dave, T., & Duddumpudi, R. T. S. (2023). Exploring the Boundaries of Reality: Investigating the Phenomenon of Artificial Intelligence Hallucination in Scientific Writing Through ChatGPT,« *Cureus*, 15(4), 2023.
- [58] P. Zhao, H. Zhang, Q. Yu, Z. Wang, Y. Geng, F. Fu, L. Yang, W. Zhang i B. Cui, »Retrieval-Augmented Generation for AI-Generated Content: A Survey,« *CoRR*, 2024.
- [59] S. Siriwardhana, R. Weerasekera, E. Wen, T. Kaluarachchi, R. Rana i S. Nanaykkara, »Improving the Domain Adaptation of Retrieval Augmented Generation,« *Transactions of the Association for Computational Linguistics*, 11, pp. 1-17, 2023.
- [60] M. Komeili, K. Shuster i J. Weston, »Internet-Augmented Dialogue Generation,« *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8460-8478, 2022.
- [61] K. Guu, K. Lee, Z. Tung, P. Pasupat i M.-W. Chang, »Retrieval augmented language model pre-training,« *International conference on machine learning*, PMLR, pp. 3929-3938.
- [62] M. Nickel, K. Murphy, V. Tresp i E. Gabrilovich, »A review of relational machine learning for knowledge graphs,« *Proceedings of the IEEE 104.1*, pp. 11-33, 2015.
- [63] A. Hogan, E. Blomquist, M. Cochez, C. D'Amato, G. De Melo, C. Giutierrez, S. Kirrane, J. E. Labra Gayo, R. Navigli, S. Neumaier, A.-C. Ngogna Ngomo, A. Polleres, S. M. Rashid, A. Rula i L. Schmelzeisen, »Knowledge graphs,« *ACM Computing Surveys (Csur)*, 54(4), pp. 1-37, 2021.
- [64] G. Zhang, J. Liu, G. Zhou, K. Zhao, Z. Xie i B. Huang, »Question-Directed Reasoning With Relation-Aware Graph Attention Network for Complex Question Answering Over Knowledge Graph,« *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32, pp. 1915-1927, 2024.
- [65] R. Cohen, M. Geva, J. Berant i A. Globerson, »Crawling The Internal Knowledge-Base of Language Models,« *In Findings of the Association for Computational Linguistics: EACL*, pp. 1856-1869, 2023.
- [66] H. V. Thakker, *On Supporting Interoperability between RDF and Property Graph Databases (Doctoral dissertation)*, Universitäts-und Landesbibliothek Bonn, 2021.
- [67] S. Kanhe, V. Udawant, P. Bodke i A. Chikhale, »SQL Generation and PL/SQL Execuion from Natural Languae Processing,« *International Journal of Engineering Research & Tehnology (IJERT)*, Vol. 4 Issue 02, February 2015.
- [68] Oracle, »SQL - the natural language for analysis, Oracle white paper,« Oracle, 2015..
- [69] N. Nihalani, »An Intelligent Interface for relational database,« *Human-computer interaction*, 2010.

- [70] F. Li i H. Jagadish, »NaLIR: A interactive natural language interface for quering relational databases,« *Proceedings of the fourth international natural language generation conference, AM*, pp. 709-712, June 2014.
- [71] L. Kovacs, »SQL Generation for Natural Language Interface,« *Computer Tehnology and Computer Programming: Research and Strategies*, pp. 19-22, 2011.
- [72] R. A. Frost i S. M. Peelar, »A New Data Structure for Processing Natural Language Database Queries,« u *WEBIST*, 2019.
- [73] A. Stoica, K. Q. Pu i H. Davoudi, »NLP Relational Queries and Its Application,« *2020 IEEE 21st International Conference on Information Reuse and Integration for Data Science (IRI)*, pp. 359-398, August 2020.
- [74] S. Walelign, D. Tesfaye i T. Kebebew, »Modelling and Designing of NLP Interface to Database for Afaan Oromoo,« *International Journal of Innovative Research in Computer Science & Technology (IJIRCST) ISSN*, pp. 2347-5552, 2021.
- [75] A. Jakupović, M. Pavlić, A. Meštrović i V. Jovanović, »Comparison of the Nodes of Knowledge method with other graphical methods for knowledge representation,« u *36th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 2013.
- [76] M. Pavlić, A. Meštrović i A. Jakupović, »Graph-Based Formalisams for Knowledge Representation,« *Proceedings of the 17th World Multi-Conference on Systemics Cybernetics and Informatics (WMSCI 2013), Vol. 2*, pp. 200-204, 2013.
- [77] M. Rauker Koch, M. Pavlić i A. Jakupović, »Aplication of the NOK method in sentence modelling,« u *Proceedings of the 37th international convention /CIS/, MIPRO 2014.*, Opatija, 2014..
- [78] M. Rauker Koch, M. Ašenbrener Katić i M. Pavlić, »Fable representation in FNOK and DNOK formalisms using the NOK conceptual framework,« *Annals of DAAAM and Proceedings of the International DAAAM Symposium*, pp. 439-445, 2017.
- [79] Ezop, »Basne,« 2018. [Mrežno]. Available: [https://lektire.skole.hr/wp-content/uploads/2020/01/ezop\\_basnee.pdf](https://lektire.skole.hr/wp-content/uploads/2020/01/ezop_basnee.pdf). [Pokušaj pristupa 25. 3. 2024].
- [80] Aesop, »The Goose With the Golden Egg,« [Mrežno]. Available: <https://www.bartleby.com/lit-hub/hc/fables/the-goose-with-the-golden-egg/>. [Pokušaj pristupa 25. 3. 2024].
- [81] M. Pavlić, Z. Dovedan Han i A. Jakupović, »Question Answering with Conceptual Framework for Knowledge-Based System Development "Node of Knowledge",« *Expert System with applications*, pp. 5264-5286, 2015.
- [82] M. Ašenbrener Katic, M. Pavlic i S. Candrlic, »The representation of database content and structure using the NOK method,« *Procedia Engineering*, 100, pp. 1075-1081, 2015.

- [83] J. Tomljanović, M. Pavlić i M. Ašenbrener Katić, »Intelligent Question – Answering Systems: Review of research,« u *Proceedings of the 37th International Convention MIPRO 2014*, Opatija, 2014.
- [84] M. Ašenbrener Katić, M. Pavlić i A. Meštrović, »The framework for transformation of text expressed knowledge into relational database« *.neobjavljeno*.
- [85] M. Ašenbrener Katić, S. Čandrlić i M. Pavlić, »Modeling of verbs using the node of knowledge conceptual framework,« u *41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 2018.
- [86] M. Ašenbrener Katić, S. Čandrlić i M. Pavlić, »Nouns in the Conceptual Framework “Node of Knowledge”,« *Tehnički vjesnik / Technical Gazette, Vol. 28/No. 6*, p. in the process of publication, 2021.
- [87] M. Pavlic, Z. D. Han, A. Jakupovic, M. Asenbrener Katic i S. Candrlic, » Adjective representation with the method Nodes of Knowledge.,« u *In 2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 2017.
- [88] M. Ašenbrener Katić, S. Čandrlić i M. Rauker Koch, »Adverb and Preposition Representation in Croatian and English using the Node of Knowledge Method,« u *Central European Conference on Information and Intelligent Systems*, Dubrovnik, 2022.
- [89] Z. Dovedan Han, *Progovorimo pythonski*, Zagreb: Vlastita naklada, 2021.
- [90] S. Candrlic, M. Asenbrener Katic i M. Pavlic, »A system for transformation of sentences from the enriched formalized Node of Knowledge record into relational database,« *Expert Systems with Applications*, 115, pp. 442-464, 2019.
- [91] N. Ide, G. Priest-Dorman, T. Erjavec i T. Varadi, »MULTEXT-East,« [Mrežno]. Available: <https://nl.ijs.si/ME/V6/msd/html/msd-en.html>. [Pokušaj pristupa 15. 8. 2021.].
- [92] C. Chiarcos i T. Erjavec, »OWL/DL formalization of the MULTEXT-East morphosyntactic specifications,« *Proceedings of the 5th Linguistic Annotation Workshop*, pp. 11-20, 2011.
- [93] T. Erjavec, »MULTEXT-East,« *Handbook of Linguistic Annotation. Springer, Dordrecht*, pp. 441-462, 2017.
- [94] L. Dimitrova, T. Erjavec, N. Ide, H. J. Kaalep, V. Petkevic i D. Tufis, »Multext-East: Parallel and Comparable Corpora and Lexicons for Six Central and Eastern European Languages,« *Proceedings of the COLING-ACL '98. Montreal: ACL*, pp. 315-319, 1998.
- [95] »Cambridge Dictionary,« [Mrežno]. Available: <https://dictionary.cambridge.org/>. [Pokušaj pristupa 9. 8. 2021.].
- [96] »Oxford Dictionaries - The World's Most Trusted Dictionary Provider,« , . [Mrežno]. Available: <http://oxforddictionaries.com>. [Pokušaj pristupa 9 8 2021].

- [97] L. J. Brinton i D. M. Brinton, »The Linguistic Structure of Modern English,« *Philadelphia: John Benjamins B.V.*, 2010.
- [98] M. Celce-Murcia i D. Larsen-Freeman, *The Grammar Book: An ESL/EFL Teacher's Course*, Boston: Heinle & Heinle., 1999.
- [99] R. Quirk, S. Greenbaum, G. Leech i J. Svartvik, *A comprehensive grammar of the English language*, New York: Longman Inc., 1985.
- [100] »OED - Oxford English Dictionary,« 15. 3. 2024.. [Mrežno]. Available: [https://www.oed.com/dictionary/synonym\\_n?tab=factsheet#19371818](https://www.oed.com/dictionary/synonym_n?tab=factsheet#19371818).
- [101] E. Suhrob i V. Kilicheva, »Parts of speech and sentence structure in English grammar,« *Galaxy International Interdisciplinary Research Journal 10.7*, pp. 156-160, 2022.
- [102] »Hrvatski jezični portal,« [Mrežno]. Available: [https://hjp.znanje.hr/index.php?show=search\\_by\\_id&id=f19uXRI%2F](https://hjp.znanje.hr/index.php?show=search_by_id&id=f19uXRI%2F). [Pokušaj pristupa 11 2 2024].
- [103] »Veznici,« [Mrežno]. Available: <https://www.lektire.hr/veznici/>. [Pokušaj pristupa 2 2024.].
- [104] »Oxford institute of education,« [Mrežno]. Available: <https://oxford-institute-education.com/wp-content/uploads/2023/01/conjunctions-list.pdf>.
- [105] E. Poszer, »škole.hr,« 6 ožujak 2020. [Mrežno]. Available: <https://www.skole.hr/glagoli-i-njihove-gramaticke-kategorije/>. [Pokušaj pristupa travanj 2024.].
- [106] B. Grgić i J. Brihta, *Engleska gramatika za svakoga*, Zagreb: Školska knjiga, 1982.
- [107] M. Rauker Koch, M. Pavlić i M. Ašenbrener Katić, »Homonyms and Synonyms in NOK Method,« *Procedia engineering*, 100, pp. 1055-1061, 2015.
- [108] 15. 3. 2024.. [Mrežno]. Available: <https://hjp.znanje.hr/>.
- [109] »OED - Oxford English Dictionary,« 15. 3. 2024.. [Mrežno]. Available: <https://www.oed.com/search/dictionary/?scope=Entries&q=homonym>.
- [110] V. Anić, *Riječnik hrvatskog jezika*, Zagreb: Novi liber, 1991.
- [111] M. Pavlić, *Oblikovanje baze podataka*, Rijeka: Odjel za informatiku, Sveučilište u Rijeci, 2011.
- [112] S. Zrinjanin, *HRVATSKI JEZIK (AZ)*, Zagreb: Alfa, 2018.
- [113] T. Marčan, *Putokazi 1*, Zagreb: Školska knjiga, 2019.

## Popis slika

Slika 2.1. Framework Question Answering sustava [42].....	10
Slika 2.2. Transformer - arhitektura modela [53].....	13
Slika 2.3. Usporedba između ChatGPT-a i metode Node of Knowledge (NOK) koju je generirao ChatGPT 4o.....	14
Slika 2.4. Odgovor Copilota na upit „Što je Prometheus?“ s uključenim poveznicama na izvore .....	15
Slika 2.5. Generička RAG arhitektura [58].....	16
Slika 2.6. Primjer generiranog grafa znanja oko entiteta ALAN TU RING, primjenom LMCRAWL [65] .....	17
Slika 3.1. Prikaz basne “Zlatna jaja u kokoši” primjenom NOK metode .....	21
Slika 3.2. Prikaz basne "The Goose with the golden egg" primjenom NOK metode .....	21
Slika 3.3. Odgovor na pitanje “Tko se polakomio?”/“Who grew greedy?” .....	22
Slika 3.4. DNOK četvrte rečenice basne na engleskom jeziku .....	24
Slika 3.5. Razine znanja [1] .....	25
Slika 3.6. Korisničko sučelje aplikacije PSA - FNOK .....	26
Slika 3.7. Rečenice učitane iz datoteke .....	29
Slika 3.8. Lexicon .....	29
Slika 3.9. Atributi riječi "kiss" .....	30
Slika 3.10. Kod kojim se određuju pitanja prema atributima uz riječi .....	31
Slika 3.11. Promjena atributa riječi "best" .....	31
Slika 3.12. Leksička analiza rečenice "Honesty is the best policy." .....	32
Slika 3.13. Atributi riječi „best“ .....	32
Slika 3.14. Dio uspoređenih rečenica .....	34
Slika 5.1. Meta model hijerarhije glagolskih vremena .....	48
Slika 5.2. Primjena meta modela hijerarhije glagolskih vremena na glagolska vremena u engleskom jeziku .....	49
Slika 5.3. Meta model sinonima prikazan NOK metodom .....	51
Slika 5.4. Meta model homonima .....	52
Slika 5.5. Model riječi "zemlja" (Land) .....	53
Slika 6.1. Model procesa sustava transformacije teksta u bazu podataka [1].....	54
Slika 6.2. Prošireni model podataka zapisa FNOK i QFNOK formalizama u relacijsku bazu podataka BPNOK .....	56
Slika 6.3. Prikaz odgovora na pitanja u aplikaciji NOK .....	60
Slika 6.4. Prikaz transformacije formaliziranog teksta FNOK u relacijsku bazu podataka .....	62
Slika 6.5. Meta model strukture i sadržaja odgovora [1] .....	65
Slika 6.6. Proširivanje modela procesa niže razine za proces 3. Generiranje odgovora na pitanje .....	66
Slika 6.7. Odgovori na pitanje „Who is from Zagreb or Rijeka?“ .....	69
Slika 6.8. Odgovor sustava na pitanje „Why Daniel loves summer?“ .....	70

Slika 7.1. KB_CONJ.....	72
Slika 7.2. KB_SYNONIM .....	72
Slika 7.3. Tablica NOK_FNOK.....	73
Slika 7.4. Hijerarhijski zapis rečenice u polje.....	74
Slika 7.5. KB_TEXT_QUEST .....	75
Slika 7.6. Hijerarhijski zapis pitanja u polje .....	76
Slika 7.7. NOK_QFNOK .....	76
Slika 7.8. KB_QUEST_QUEST .....	76
Slika 7.9. Zapisi rečenica 48 do 52 u tablicu KB_TEXT_QUEST.....	80
Slika 7.10. Prikaz odgovora na pitanje "Who is playing?" .....	82
Slika 7.11. Odgovor sustava na pitanja "Why is Monika glad?" .....	84
Slika 7.12. Odgovor na pitanje "Where will I go?" .....	86
Slika 7.13. Odgovor na pitanje "Who loves his job?" .....	86
Slika 7.14. Odgovor na pitanje „What they could do?“ .....	87
Slika 7.15. Prikaz odgovora sustava na pitanje "What did we do to each other?" .....	89
Slika 7.16. Odgovor sustava na pitanje "What they swallowed?" .....	89
Slika 7.17. Pregled pronađenih čvorova u rečenici „There was nothing they could do but swallow their pride.“ za pitanje "What they swallowed?" .....	90
Slika 7.18. Odgovor sustava na pitanje „Who is talking?“ .....	94
Slika 7.19. Odgovor sustava na pitanje „Who is discussing?“ .....	95
Slika 7.20. Odgovor na pitanje "Who is happy?" .....	96
Slika 7.21. Odgovor na pitanje "Who is glad?" .....	97
Slika 7.22. Hijerarhijski zapis rečenice „Marko je sretan jer voli svoj posao.“ u polje.....	99
Slika 7.23. Zapis rečenice „Marko je sretan jer voli svoj posao.“ u tablici NOK_FNOK.....	99
Slika 7.24. Odgovor na pitanje „Zašto je Marko sretan?“ .....	100

## Popis tablica

Tablica 3.1. Koncepti NOK metode .....	19
Tablica 3.2. Atributi imenica.....	28
Tablica 3.3. Atributi glagola.....	28
Tablica 3.4. Usporedba ručno generiranih FNOK zapisa i generiranih automatski PSA-FNOK aplikacijom .....	35
Tablica 4.1. Primjeri pretvorbe rečenica prirodnog jezika u kojima veznici spajaju riječi u FNOK zapis.....	41
Tablica 4.2. Primjeri pretvorbe rečenica prirodnog jezika u kojima veznici povezuju dvije rečenice .....	43
Tablica 4.3. Primjeri pitanja i pripadajućih QFNOK-ova koji sadrže veznike.....	44
Tablica 5.1. Glagolska vremena u hrvatskom i engleskom jeziku .....	46
Tablica 5.2. Značenje riječi zemlja [110] .....	52
Tablica 6.1. Primjeri rečenica koje koriste glagole "talk" i "discuss" .....	58
Tablica 7.1. KB_ANSWER.....	78
Tablica 7.2. Točni odgovori .....	81
Tablica 7.3. Primjeri pitanja s gramatički neispravnim odgovorom.....	83
Tablica 7.4. Primjeri pitanja s djelomično točnim odgovorima .....	85
Tablica 7.5. Primjeri pitanja za koja nije pronađen odgovor, ali je pronađena rečenica u kojoj se odgovor nalazi.....	88
Tablica 7.6. Pitanja postavljena za rečenicu „I didn't go out because I had no money.“ i pripadajući QFNOK-ovi .....	92
Tablica 7.7. Odgovori na pitanja postavljena za rečenicu „I didn't go out because I had no money.“ i usporedba s očekivanim odgovorima.....	92
Tablica 7.8. Odabrane rečenice za testiranje na hrvatskom jeziku.....	98
Tablica 7.9. Odgovori sustava na 10 postavljenih pitanja na hrvatskom jeziku .....	101

## Popis grafikona

Grafikon 1. Rezultati testiranja sustava .....	91
--	----

## Popis privitaka

Privitak 1. PHP skripta <i>answer.php</i> .....	119
Privitak 2. PHP skripta <i>insert.php</i> .....	127
Privitak 3. PHP skripta <i>insertQ.php</i> .....	135
Privitak 4. PHP skripta <i>functions.php</i> .....	143
Privitak 5. Rečenice korištene u inicijalnom testiranju .....	165
Privitak 6. Pitanja korištena u inicijalnom testiranju i pripadajući QFNOK-ovi.....	166
Privitak 7. Testirane rečenice engleskog jezika i pripadajući FNOK-ovi.....	168
Privitak 8. Testirana pitanja engleskog jezika i pripadajući QFNOK-ovi.....	173
Privitak 9. Odgovori sustava na pitanja.....	182



## Privitak 1. PHP skripta *answer.php*

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <title>NOK</title>
    <link
      href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
      rel="stylesheet"
      integrity="sha384-T3c6CoIi6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwvykc2MPK8M2HN"
      crossorigin="anonymous"
    />
    <link href="style.css" rel="stylesheet" />
  </head>
  <body>
    <div class="container">
      <h1>Odgovori na pitanja</h1>
      <?php

        require_once('config.php');
        require_once('functions.php');

        connect();
        loadData();

        $prg = isset($_GET['prg']) ? intval($_GET['prg']) : 99;
        $ord = isset($_GET['ord']) ? intval($_GET['ord']) : 1;

        if(isset($_GET['q'])) {
          $elts = explode('-', $_GET['q']);
          if(count($elts) === 2) {
            $prg = intval($elts[0]);
            $ord = intval($elts[1]);
          }
        }

        // popis pitanja i forma za odabir pitanja
        $q = 'select * from NOK_QFNOK_ALL where PRG = 99 order by ORD';
        $st = oci_parse($conn, $q);
        $r = oci_execute($st);

        $form = '<form method="get" action="?" class="row">' .
          '<div class="col-5">' .
            '<select class="form-select" name="q" id="q">';

        $question = '';
        $qfnok = '';
        while($row = oci_fetch_assoc($st)) {
          $selected = $row['PRG'] == $prg && $row['ORD'] == $ord;
          $form .= '<option value="' . $row['PRG'] . '-' . $row['ORD'] . '" . ($selected
? ' selected="selected"' : '') . '>' . $row['QTENG'] . '</option>';
          if($selected) {
            $question = $row['QTENG'];
            $qfnok = $row['QFNOK'];
          }
        }
        $form .= '</select>' .
          '</div>' .
          '<div class="col-5 d-grid"><button class="btn btn-
secondary">Odaberi</button></div>' .
          '</form>';
      </?php
    </div>
  </body>
</html>
```

```

echo '<div class="alert alert-light" role="alert">' .
    '<h4 class="alert-heading">Pitanje</h4>' .
    '<p>' . $form . '</p>' .
    '<hr>' .
    '<p class="mb-0"><strong>Formalizirani zapis</strong><br /></code>' . $qfnok
. '</code></p>' .
    '</div>';

$ansStart = microtime(true);
$ans = getAnswers($prg, $ord);
$ansEnd = microtime(true);
echo '<pre>Answered: ' . ($ansEnd - $ansStart) . '</pre>';

// prikaz strukture pitanja

echo '<div class="row">' .
    '<div class="col-6">' .
    '<h3>Hijerarhijski zapis pitanja</h3>' .
    '<table class="table table-bordered nok-table-1">' .
    '<thead><tr><th>Node ID</th><th>Node</th><th>Question</th><th>Parent
ID</th></tr></thead>' .
    '<tbody>';
//while($row = oci_fetch_assoc($st)) {
foreach($ans['qfnok'] as $idx => $val) {
    //echo '<tr><td><pre>' . var_export($row, true) . '</pre></td></tr>';
    $class = false;
    if($val['QUEST'] === '&') {
        $class = 'table-secondary';
    } elseif($val['NODE'] === 'X') {
        $class = 'table-primary';
    }
    echo '<tr>' . ($class ? ' class="' . $class . '"' : '') . '>' .
        '<td>' . $val['NODE_ID'] . '</td>' .
        '<td style="padding-left: ' . (20 * ($val['LEVEL']-1) + 8) . 'px">' .
$val['NODE'] . '</td>' .
        '<td>' . $val['QUEST'] . '</td>' .
        '<td>' . $val['PARENT_ID'] . '</td>' .
        '</tr>';
}
echo '</tbody>' .
    '</table>' .
    '</div>' .
    '<div class="col-6">';

echo '<h3>Obradeno pitanje</h3>';
echo '<table class="table table-bordered nok-table-2">' .
    '<thead><tr>' .
    '<th>ID</th>' .
    '<th>PAR_WRD_ID</th>' .
    '<th>NOD_QST_ID</th>' .
    '<th>NOD_WRD_ID</th>' .
    '</tr></thead>' .
    '<tbody>';
foreach($ans['elts'] as $idx => $val) {
    echo '<tr>' .
        '<td>' . $val['ID'] . '</td>' .
        '<td>' . $val['PAR_WRD_ID'] . '<br />' . $val['w1'] . '</td>' .
        '<td>' . $val['NOD_QST_ID'] . '<br />' .
(isset($quest[$val['NOD_QST_ID']]) ? $quest[$val['NOD_QST_ID']]['QUEST'] : '') . '</td>' .
        '<td>' . $val['NOD_WRD_ID'] . '<br />' . ($val['w2'] ? $val['w2'] :
$val['NOD_WRD']) . '</td>' .
        '</tr>';

    $qElts[] = $row;
}

```

```

echo '</tbody>' .
    '</table>' .
    '</div>' .
    '</div>';

//sort pitanja
function ansSortCallback($a1, $a2) {
    return $a2['numFound'] - $a1['numFound'];
}
usort($ans['s'], 'ansSortCallback');

echo '<div class="accordion" id="accordion-answers">';
displayAnswers(true);
displayAnswers(false);
echo '</div>';
function displayAnswers($indAnswer) {
    global $ans;
    foreach($ans['s'] as $idx => $s) {
        if($indAnswer !== $s['answerOk']) continue;
        echo '<div class="accordion-item">' .
            '<h2 class="accordion-header">' .
                '<button class="accordion-button' . (!$s['answerOk'] &&
                '$s["numFound"] === 0 ? "collapsed" : "") . "' type="button" data-bs-toggle="collapse"
                data-bs-target="#accordion-' . $idx . '" aria-expanded="' . (!$s['answerOk'] &&
                '$s["numFound"] === 0 ? "false" : "true") . "' aria-controls="accordion-' . $idx . '">' .
                    '<span class="answer-icon">' . ($s['answerOk'] ? '<i class="bi
                    bi-check2"></i>' : '<i class="bi bi-x-lg"></i>') . '</span>' .
                    '[' . $s['sentence']['prg'] . '-' . $s['sentence']['ord'] . ']'
                .
                '$s["sentence"]["sentence"] .
                '</button>' .
                '</h2>' .
                '<div id="accordion-' . $idx . '" class="accordion-collapse
                collapse' . (!$s['answerOk'] && $s["numFound"] === 0 ? '' : ' show') . '" data-bs-
                parent="#accordion-answers">' .
                    '<div class="accordion-body">' .
                        '<p><code>' . $s['sentence']['fnoks'] . '</code></p>' .
                        '<p>' . ($s['answerOk'] ? '<strong>ODGOVOR:</strong>' .
                $s['answer'] : 'NEMA ODGOVORA') . '</p>' .
                        '<div class="row">';
        echo '<div class="col-6">' .
            '<table class="table table-bordered">' .
                '<thead><tr><th>PARENT</th><th>QUESTION</th><th>NODE</th></tr><
                /thead>' .
                '<tbody>';

        foreach($s['sentence']['elts'] as $idx => $val) {
            $class = '';
            if($val['FOUND'] === true) {
                $class = 'table-success';
            }
            echo '<tr class="' . $class . '">' .
                '<td>[' . $val['PARENT_ID'] . ']' . $val['PAR_WRD'] . '</td>' .
                .
                '<td>' . $val['NOD_QST'] . '</td>' .
                '<td>[' . $val['NODE_ID'] . ']' . $val['NOD_WRD'] . '</td>' .
                .
                '</tr>';
        }
        echo '</tbody>' .
            '</table>' .
            '</div>' .
            '<div class="col-6">';
    }
}

```

```

        echo '<table class="table table-bordered">' .
            '<thead><tr><th>PARENT</th><th>QUESTION</th><th>NODE</th></tr></thead>' .
            '<tbody>';
        foreach($s['qElts'] as $qIdx => $qElt) {
            $class = '';
            if($qElt['FOUND'] === true) {
                $class = 'table-success';
            }
            echo '<tr class="' . $class . '">' .
                '<td>' . $qElt['PAR_WRD'] . '</td>' .
                '<td>' . $qElt['NOD_QST'] . '</td>' .
                '<td>' . $qElt['NOD_WRD'] . '</td>' .
                '</tr>';
        }
        echo '</tbody>' .
            '</table>';

        echo '</div>';
        echo '        </div>' .
            '</div>' .
            '</div>' .
            '</div>';
    }
}
if(false) {
    $q = 'select * from NOK_FNOK_ALL';
    $st = oci_parse($conn, $q);
    $r = oci_execute($st);
    echo '<h2>Traženje odgovora</h2>';

    echo '<div class="accordion" id="fnok-accordion">';
    $num = 0;
    while($row = oci_fetch_assoc($st)) {
        $num++;
        echo '<div class="accordion-item">' .
            '<h2 class="accordion-header">' .
                '<button class="accordion-button" type="button" data-bs-' .
toggle="collapse" data-bs-target="#collapse-' . $num . '">
            ' . $row['TENGL'] . '</button>' .
            '</h2>' .
            '<div id="collapse-' . $num . '" class="accordion-collapse collapse show"
style="">' .
                '<div class="accordion-body">';

        $sentence = $row['TENGL'];

        $q2 = 'select q.*, level ' .
            'from NOK_FNOK q ' .
            'where PRG = ' . $row['PRG'] . ' ' .
            'and ORD = ' . $row['ORD'] . ' ' .
            'connect by prior node_id = parent_id and prior prg = prg and prior ord =
ord ' .
            'start with parent_id = 0';
        $s2 = oci_parse($conn, $q2);
        $r2 = oci_execute($s2);
        echo '//<h3>Zapis rečenice</h3>' .
            '<div class="row"><div class="col-4">' .
                '<p><code>' . $row['FNOK'] . '</code></p>' .
                '<table class="table table-bordered table-sm">' .
                    '<thead><tr><th>Node
ID</th><th>Node</th><th>Question</th><th>Parent ID</th></tr></thead>' .
                    '<tbody>';
                while($row2 = oci_fetch_assoc($s2)) {

```

```

        echo '<tr>' .
            '<td>' . $row2['NODE_ID'] . '</td>' .
            '<td style="padding-left:' . (($row2['LEVEL'] - 1) * 20 + 8) .
'px">' . $row2['NODE'] . '</td>' .
            '<td>' . $row2['QUEST'] . '</td>' .
            '<td>' . $row2['PARENT_ID'] . '</td>' .
            '</tr>';
    }
    echo '</tbody></table>' .
        '</div>' . // col-6
        '<div class="col-8">';
        //'<h3>Obrađena rečenica</h3>' .
    $sPrg = $row['PRG'];
    $sOrd = $row['ORD'];
    $q2 = 'select * ' .
        'from KB_TEXT_QUEST Q ' .
        'where PRG = ' . $row['PRG'] . ' ' .
        'and ORD = ' . $row['ORD'] . ' ' .
    $st2 = oci_parse($conn, $q2);
    $r2 = oci_execute($st2);
    echo '<table class="table table-bordered">' .
        '<thead><tr>' .
            '<th>PAR_WRD_ID</th>' .
            '<th>NOD_WRD_ID</th>' .
            '<th>NOD_QST_ID</th>' .
        '</tr></thead>' .
        '<tbody>';

    foreach($qElts as $idx => $val) {
        if($val['NOD_QST_ID'] === '-1') {
            $qElts[$idx]['FOUND'] = true;
        } else {
            $qElts[$idx]['FOUND'] = false;
        }
    }
}

$foundIdx = [];
$sElts = [];
while($row = oci_fetch_assoc($st2)) {
    $w1 = getWord($row['PAR_WRD_ID']);
    $w2 = getWord($row['NOD_WRD_ID']);
    $w3 = getQuest($row['NOD_QST_ID']);

    $p1 = getWordPar($row['PAR_WRD_ID']);
    $p2 = getWordPar($row['NOD_WRD_ID']);
    $row['PAR_WRD_ID2'] = $p1;
    $row['NOD_WRD_ID2'] = $p2;

    $found = false;
    foreach($qElts as $idx => $elt) {
        if(($elt['PAR_WRD_ID'] === $row['PAR_WRD_ID'] || ($elt['PAR_WRD_ID2']
=== $row['PAR_WRD_ID2'] && $elt['PAR_WRD_ID2'] !== false)) &&
($elt['NOD_WRD_ID'] === $row['NOD_WRD_ID'] || ($elt['NOD_WRD_ID2']
=== $row['NOD_WRD_ID2'] && $elt['NOD_WRD_ID2'] !== false))) {

            $found = true;
            $qElts[$idx]['FOUND'] = true;
            $row['FOUND'] = true;
            break;
        } elseif(($elt['PAR_WRD_ID'] === $row['PAR_WRD_ID'] ||
($elt['PAR_WRD_ID2'] === $row['PAR_WRD_ID2'] && $elt['PAR_WRD_ID2'] !== false)) &&
($row['NOD_WRD'] === 'and')) {

            // word is AND -> check all direct children

```

```

        $qAnd = 'select * ' .
            'from nok_fnok ' .
            'where prg = ' . $row['PRG'] . ' and ord = ' .
$row['ORD'] . ' and level = 1 ' .
            'connect by prior node_id = parent_id and prior prg =
prg and prior ord = ord ' .
            'start with kb_par_wrd_id = ' . $row['NOD_WRD_ID'];
        $sAnd = oci_parse($conn, $qAnd);
        $rAnd = oci_execute($sAnd);
        while($rowAnd = oci_fetch_assoc($sAnd)) {
            if($rowAnd['KB_NOD_WRD_ID'] === $elt['NOD_WRD_ID'] ||
($elt['NOD_WRD_ID2'] !== false && $elt['NOD_WRD_ID2'] ===
getWordPar($rowAnd['KB_NOD_WRD_ID']))) {
                $found = true;
                $qElts[$idx]['FOUND'] = true;
                $row['FOUND'] = true;
                break;
            }
        }

        echo '<pre>' . $elt['PAR_WRD'] . '-' . $elt['NOD_WRD'] . ' / ' .
$row['PAR_WRD'] . '-' . $row['NOD_WRD'] . ' ' . 'AND?</pre>';

        if($found) {
            break;
        }

        } elseif($elt['PAR_WRD_ID'] === $row['PAR_WRD_ID'] ||
($elt['PAR_WRD_ID2'] === $row['PAR_WRD_ID2'] && $elt['PAR_WRD_ID2'] !== false)) {
            // parent is correct, node is not -> check if there is AND after NODE
            $qAnd = 'select * ' .
                'from nok_fnok ' .
                'where prg = ' . $row['PRG'] . ' and ord = ' . $row['ORD']
. . . .
            'connect by prior node_id = parent_id and prior prg = prg
and prior ord = ord ' .
            'start with kb_par_wrd_id = ' . $row['NOD_WRD_ID'];
            $sAnd = oci_parse($conn, $qAnd);
            $rAnd = oci_execute($sAnd);
            $andFirst = true;
            while($rowAnd = oci_fetch_assoc($sAnd)) {
                if($andFirst) {
                    if($rowAnd['NODE'] !== 'and') {
                        //echo '<p><code>NO AND</code></p>';
                        break;
                    }
                }
                echo '<p><code>Found AND</code></p>';
                $andFirst = false;
                continue;
            } else {
                if($rowAnd['KB_NOD_WRD_ID'] === $elt['NOD_WRD_ID'] ||
($elt['NOD_WRD_ID2'] !== false && $elt['NOD_WRD_ID2'] ===
getWordPar($rowAnd['KB_NOD_WRD_ID']))) {
                    $found = true;
                    $qElts[$idx]['FOUND'] = true;
                    $row['FOUND'] = true;
                    break;
                }
            }
        }
    }
    if($found) {
        break;
    }
}
}
}

```

```

    if(!$found) {
        $row['FOUND'] = $found;
        $sElts[] = $row;
    }

    echo '<tr>' . ($found ? ' class="table-success"' : '') . '>' .
        '<td>' . $row['PAR_WRD_ID'] . '<br />' . $w1 . '</td>' .
        '<td>' . $row['NOD_WRD_ID'] . '<br />' . $w2 . '</td>' .
        '<td>' . $row['NOD_QST_ID'] . '<br />' . $w3 . '</td>' .
        '</tr>';
}
echo '</tbody>' .
    '</table>';

$qEltsRemain = '';
foreach($qElts as $idx => $elt) {
    if($elt['FOUND'] !== true) {
        $qEltsRemain .= '<pre>' . var_export($elt, true) . '</pre>';
    }
}
$sEltsRemain = '';
foreach($sElts as $idx => $elt) {
    if($elt['FOUND'] !== true) {
        $sEltsRemain .= '<pre>' . var_export($elt, true) . '</pre>';
    }
}
if($showQuestionDetails) {
    echo '<div class="row">' .
        '<div class="col-6"><h6>QUESTION</h6>' . $qEltsRemain . '</div>' .
        '<div class="col-6"><h6>SENTENCE</h6>' . $sEltsRemain . '</div>' .
        '</div>';
}

$answer = false;
$answerWrld = false;
foreach($qElts as $qIdx => $qElt) {
    if($qElt['FOUND'] === true) { continue; }
    foreach($sElts as $sIdx => $sElt) {
        if($sElt['FOUND'] === true) { continue; }
        if(($qElt['PAR_WRD_ID'] === $sElt['PAR_WRD_ID'] ||
($qElt['PAR_WRD_ID2'] === $sElt['PAR_WRD_ID2'] && $qElt['PAR_WRD_ID2'] !== false))) {
            if($qElt['NOD_WRD'] === 'X') {
                if($answer === false) {
                    $answer = $sElt['NOD_WRD'];
                    $answerWrld = $sElt['NOD_WRD_ID'];
                    $qElts[$qIdx]['FOUND'] = true;
                    break;
                } else {
                    if($answer === $sElt['NOD_WRD']) {
                        $qElts[$qIdx]['FOUND'] = true;
                        break;
                    } else {
                        $qElts[$qIdx]['FOUND'] = true;
                        $answer .= ', ' . $sElt['NOD_WRD'];
                        $answerWrld = $sElt['NOD_WRD_ID'];
                        break;
                    }
                }
            }
        }
    }
} elseif(($qElt['NOD_WRD_ID'] === $sElt['NOD_WRD_ID'] ||
($qElt['NOD_WRD_ID2'] === $sElt['NOD_WRD_ID2'] && $qElt['NOD_WRD_ID2'] !== false))) {
    if($qElt['PAR_WRD'] === 'X') {
        if($answer === false) {
            $answer = $sElt['PAR_WRD'];
            $answerWrld = $sElt['PAR_WRD_ID'];
        }
    }
}

```

```

        $qElts[$qIdx]['FOUND'] = true;
        break;
    } else {
        if($answer === $sElt['PAR_WRD']) {
            $qElts[$qIdx]['FOUND'] = true;
            break;
        } else {
            $qElts[$qIdx]['FOUND'] = true;
            $answer .= ', ' . $sElt['PAR_WRD'];
            $answerWrđ = $sElt['PAR_WRD_ID'];
            break;
        }
    }
}
}
}
}
}

$ok = true;
foreach($qElts as $idx => $val) {
    if($val['FOUND'] === false) {
        $ok = false;
        break;
    }
}
if($ok) {

    // obrada odgovora ->
    $qA = 'select * ' .
        'from nok_fnok ' .
        'where prg = ' . $sPrg . ' and ord = ' . $sOrd . ' ' .
        'connect by prior node_id = parent_id and prior prg = prg and prior ord
= ord ' .

        'start with kb_par_wrd_id = ' . $answerWrđ;
    $stA = oci_parse($conn, $qA);
    $rA = oci_execute($stA);
    $tmpAnswer = '';
    while($rowA = oci_fetch_assoc($stA)) {
        $tmpAnswer .= $rowA['NODE'] . ' ';
    }
    echo '<div class="alert alert-success"><h4 class="alert-heading">Odgovor
pronađen</h4><p>' . $question . '</p><p><small>' . $sentence . '</small></p><p><strong>' .
$tmpAnswer . $answer . '</strong></p></div>';
    } else {
        echo '<div class="alert alert-danger"><h4 class="alert-heading">Odgovor
NIJE pronađen</h4><p>' . $question . '</p><p><small>' . $sentence . '</small></p><!--
<p><strong>' . $answer . '</strong></p>--></div>';
    }

    echo '</div>' . // col-6
        '</div>'; // row
    echo '</div></div></div>'; // collapse
}
echo '</div>'; //.accordion
}
?>
</div>
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-C6RzsynM9kWDrmMNeT87bh950GNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL"
crossorigin="anonymous"
></script>
</body>
</html>

```



## Privitak 2. PHP skripta insert.php

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>NOK insert</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
T3c6CoIi6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwvykc2MPK8M2HN" crossorigin="anonymous">
    <link href="style.css" rel="stylesheet" />
  </head>
  <body>
    <div class="container">
      <h1>Unos u bazu podataka</h1>
      <?php
require_once('config.php');
require_once('functions.php');

$s0 = microtime(true);

connect();
loadData();

$q = 'select * from nok_fnok_imp order by id';
//$q = 'select * from nok_fnok_imp';
$st = oci_parse($conn, $q);
$r = oci_execute($st);

$doProcess = 1;
if(isset($_GET['p'])) {
    $doProcess = intval($_GET['p']);
}

$rbr = 0;
$prg = 99;

$debug1 = true;

while($row = oci_fetch_assoc($st)) {
    $start = microtime(true);
    $rbr++;

    if($debug1) {
        if(intval($row['ID']) !== $doProcess) {
            continue;
        }
    }
}
```

```

}

echo '<pre>[' . $row['ID'] . ']' . $row['TENG'] . '</pre>';

$tmpFnok = str_replace(') (' , ' , ' , $row['FNOK']);

$elts = processNOK($tmpFnok);
$t1 = microtime(true);
if($debug1) {
echo '<strong><pre>' . "\n" . var_export($elts, true) . '</pre></strong>';
}

//cleanup
$q = 'delete from NOK_FNOK where PRG = ' . $prg . ' and ORD = ' . $row['ID'];
$st2 = oci_parse($conn, $q);
oci_execute($st2);

$q = 'delete from NOK_FNOK_ALL where PRG = ' . $prg . ' and ORD = ' .
$row['ID'];
$st2 = oci_parse($conn, $q);
$ok = oci_execute($st2);
if(!$ok) {
    echo 'ERROR 0';
}

$q = 'insert into NOK_FNOK_ALL (PRG, ORD, TENG, FNOK) values (' . $prg . ', ' .
$row['ID'] . ', :teng, :fnok)';
$st2 = oci_parse($conn, $q);
oci_bind_by_name($st2, ':teng', $row['TENG']);
oci_bind_by_name($st2, ':fnok', $row['FNOK']);
$ok = oci_execute($st2);
if(!$ok) {
    echo 'ERROR 1';
}

$t2 = microtime(true);
$currNodeId = 0;
processNOKArray($elts, $prg, $row['ID'], 0, '', 'null', '_');
$t3 = microtime(true);
if($debug1) {
if(intval($row['ID']) === $doProcess) {
    $q = 'begin KB_CREATE_QUEST(' . $prg . '); end;';
    $st2 = oci_parse($conn, $q);
    oci_execute($st2);
    die();
}
}
}

```

```

        $end = microtime(true);
        echo '<code>T: ' . number_format($end - $start, 2, ',', '.') . ' s</code><br
/>';
    }

    $q = 'begin KB_CREATE_REQUEST(' . $prg . '); end;';
    $st2 = oci_parse($conn, $q);
    oci_execute($st2);

    $e0 = microtime(true);
    echo '<code>TOTAL TIME: ' . number_format($e0 - $s0, 2, ',', '.') . ' s</code><br
/>';

function processNOK($nok) {

    $debug = false;
    /*if($nok === 'come ( can ("who?" you)), see ("who?" me, "when?" tomorrow') {
        $debug = true;
    }*/

    $elts = [];
    $state = 0;
    $elt = '';
    $brack = 0;
    for($i = 0; $i < strlen($nok); $i++) {
        $char = substr($nok, $i, 1);
        // početak
        if($state === 0) {
            if($char === " " || $char === "," || $char === "'") {
                if($elt === '') {
                    continue;
                } else {
                    $elts[] = $elt;
                    $elt = '';
                }
            } elseif($char === '(') {
                if($elt === '') {
                    $brack++;
                    $state = 1;
                    $elt .= $char;
                }
            } else {
                $elt .= $char;
            }
        } elseif($state === 1) {
            $elt .= $char;
            if($char === '(') {
                $brack++;
            }
        }
    }
}

```

```

    } elseif($char === ')') {
        $brack--;
        if($brack === 0) {
            $subElts = processNOK(substr($elt, 1, strlen($elt)-2));
            $elts[] = $subElts;
            $elt = '';
            $state = 0;
        }
    }
}

if($state !== 0) {
    echo '<pre>ERROR [' . $nok . ']</pre>';
    return [];
} else {
    if($elt !== '' && $elt !== ')') {
        $elts[] = $elt;
        $elt = '';
    }
}

if($debug) {
    echo '<strong>ProcessNOK: ' . $nok . '</strong><br />';
}

if(is_array($elts) && is_array($elts[0])) {
    return $elts;
}

if(count($elts) === 2) {
    if(strpos($elts[0], '?') !== false) {
        $q = $elts[0];
        if(substr($q, 0, 1) === '') {
            $q = substr($q, 1);
        }
        if(substr($q, strlen($q) - 1, 1) === '') {
            $q = substr($q, 0, strlen($q) - 1);
        }
        if(substr($q, strlen($q) - 1, 1) === '?') {
            $q = substr($q, 0, strlen($q) - 1);
        }
        $ret = ['Q' => $q, 'A' => $elts[1]];
    } else {
        $ret = ['V' => $elts[0], 'F' => $elts[1]];
    }
}

return $ret;
} else {
    $newElts = [];
    $newNodes = 0;
    if($debug) {

```

```

echo '<div class="row">' .
    '<div class="col-6"><pre>' . var_export($elts, true) .
'</pre></div>' .
    '<div class="col-6"><pre>';
}
for($i = 0; $i < count($elts); $i++) {
    $elt = $elts[$i];
    if(is_array($elt)) {
        $newElts[] = $elt;
        continue;
    }
    if(strpos($elt, '?') !== false || $elt === '_' || $elt === '&') {
        // question word
        $q = $elt;
        if(substr($q, 0, 1) === '') {
            $q = substr($q, 1);
        }
        if(substr($q, strlen($q) - 1, 1) === '') {
            $q = substr($q, 0, strlen($q) - 1);
        }
        if(substr($q, strlen($q) - 1, 1) === '?') {
            $q = substr($q, 0, strlen($q) - 1);
        }
        if(isConjunctionW($elts[$i+1])) {
            $newElts[] = [
                'Q' => $q,
                'C' => $elts[$i+1],
                'O' => $elts[$i+2],
            ];
            $i += 2;
            continue;
        } else {
            $ans = $elts[$i+1];
            if(substr($ans, strlen($ans) - 1, 1) === ',') {
                $ans = substr($ans, 0, strlen($ans) - 1);
            }

            if(isset($elts[$i+2]) && is_array($elts[$i+2])) {
                $newElts[] = [
                    'Q' => $q,
                    'C' => $ans,
                    'O' => $elts[$i+2]
                ];
                $i += 2;
            } else {
                $newElts[] = [
                    'Q' => $q,
                    'A' => $ans
                ];
            }
        }
    }
}

```

```

        ];
        $i++;
    }

        continue;
    }
} else {
    // VERB
    $newElts[] = [
        'V' => $elts[$i],
        'F' => $elts[$i + 1],
    ];
    $i++;
    continue;
    // conjunction -> other node are conjuncted
}
}
/*echo '<div class="row">' .
if($debug) {
    echo var_export($newElts, true) . '</pre></div>' .
        '</div>' . '<hr />';
}
return $newElts;
}
return $elts;
}

function processNOKArray($nok, $prg, $ord, $parent, $parWrId, $parWrId, $parQuest)
{
    global $conn;

    global $currNodeId;
    $currNodeId++;
    $proposedNodeId = $currNodeId;
    $nodeId = $proposedNodeId;
    $node = '';
    $quest = '';
    $kbParWrId = '';
    $kbParWr = '';
    $kbNodWrId = '';
    $kbNodQstId = '';

    $toProcess = false;
    // two options -> VERB or QUESTION
    if(isset($nok['V'])) {
        echo '<p><code>case-V [' . $nok['V'] . ']</code></p>';
        $node = $nok['V'];
    }
}

```

```

    $quest = '_';
    if(is_array($nok['F'])) {
        $toProcess = $nok['F'];
    } else {
        $node = $nok['F'];
    }
} elseif(isset($nok['Q'])) {
    $quest = $nok['Q'];
    // two options -> ANSWER or CONJUNCTION
    if(isset($nok['C'])) {
        $node = $nok['C'];
        $toProcess = $nok['O'];
    } else {
        $node = $nok['A'];
    }
} else {
    $currNodeId--;
    foreach($nok as $idx => $val) {
        processNOKArray($val, $prg, $ord, $parent, $parWrd, $parWrdId, '_');
    }
    return;
}

echo '<pre>GET QUEST ID: ' . var_export($quest, true) . '</pre>';
$questId = getQuestId($quest);
$nodWrdId = getWordId($node);

$q = 'insert into NOK_FNOK (prg, ord, node_id, node, quest, parent_id,
kb_par_wrd_id, kb_par_wrd, kb_nod_wrd_id, kb_nod_qst_id) ' .
    "\n" .
    'values (' . $prg . ', ' . $ord . ', ' . $nodeId . ', :node, \'' .
$quest . '\', ' . $parent . ', ' . $parWrdId . ', :parWrd, ' . $nodWrdId . ', ' . $questId
. ')';

echo '<pre>' . $q . '</pre>';

$st = oci_parse($conn, $q);
$bindOk = oci_bind_by_name($st, ':node', $node);
$bindOk = oci_bind_by_name($st, ':parWrd', $parWrd);
$ok = oci_execute($st);
if($ok) {
} else {
    echo '<code><pre>' . $q . '</pre></code>';
}

if($toProcess) {
    foreach($toProcess as $idx => $val) {
        if(!is_numeric($idx)) {

```

```

        processNOKArray($toProcess, $prg, $ord, $nodeId, $node, $nodWrdId,
$quest);
        break;
    }
    processNOKArray($val, $prg, $ord, $nodeId, $node, $nodWrdId, $quest);
}
}

}

// unos pitanja
// KB_QUESTION_QUESTION
// NOK_QUESTION_QUESTION
// NOK_QUESTION_QUESTION_ALL -> prvi
// NOK_QUESTION_QUESTION_IMP -> prvi
// unos rečenice
// KB_TEXT_QUESTION
// NOK_QUESTION_QUESTION
// NOK_QUESTION_QUESTION_ALL -> prvi
// NOK_QUESTION_QUESTION_IMP -> prvi
?>
</div>
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-C6RzsynM9kWDrmNeT87bh950GnyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL"
crossorigin="anonymous"></script>
</body>
</html>

```



## Privitak 3. PHP skripta insertQ.php

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>NOK insert</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
T3c6CoIi6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwvykc2MPK8M2HN" crossorigin="anonymous">
    <link href="style.css" rel="stylesheet" />
  </head>
  <body>
    <div class="container">
      <h1>Unos u bazu podataka</h1>
      <?php
require_once('config.php');
require_once('functions.php');

$s0 = microtime(true);

connect();
loadData();

$q = 'select * from nok_qfnok_imp order by qfnok';
//$q = 'select * from nok_fnok_imp';
$st = oci_parse($conn, $q);
$r = oci_execute($st);

$doProcess = 1;
if(isset($_GET['p'])) {
    $doProcess = intval($_GET['p']);
}

$rbr = 0;
$prg = 99;

$debug1 = true;

while($row = oci_fetch_assoc($st)) {
    $start = microtime(true);
    $rbr++;

    if($debug1) {
        if(intval($row['ID']) !== $doProcess) {
            continue;
        }
    }
}
```

```

}

echo '<pre>[' . $row['ID'] . ']' . $row['QTENG'] . '</pre>';
echo '<pre> ' . $row['QFNOK'] . '</pre>';

$tmpQfnok = str_replace(' ' , ' ', $row['QFNOK']);

$elts = processNOK($tmpQfnok);
$t1 = microtime(true);
if($debug1) {
echo '<strong><pre>' . "\n" . var_export($elts, true) . '</pre></strong>';
}

//cleanup
$q = 'delete from NOK_QFNOK where PRG = ' . $prg . ' and ORD = ' . $row['ID'];
$st2 = oci_parse($conn, $q);
oci_execute($st2);

$q = 'delete from NOK_QFNOK_ALL where PRG = ' . $prg . ' and ORD = ' .
$row['ID'];
$st2 = oci_parse($conn, $q);
$ok = oci_execute($st2);
if(!$ok) {
    echo 'ERROR 0';
}

$q = 'insert into NOK_QFNOK_ALL (PRG, ORD, QTENG, QFNOK) values (' . $prg . ',
' . $row['ID'] . ', :teng, :fnok)';
$st2 = oci_parse($conn, $q);
oci_bind_by_name($st2, ':teng', $row['QTENG']);
oci_bind_by_name($st2, ':fnok', $row['QFNOK']);
$ok = oci_execute($st2);
if(!$ok) {
    echo 'ERROR 1';
}

$t2 = microtime(true);
$currNodeId = 0;
processNOKArray($elts, $prg, $row['ID'], 0, '', 'null');
$t3 = microtime(true);
if($debug1) {
if(intval($row['ID']) === $doProcess) {
    $q = 'begin KB_CREATE_QUEST(' . $prg . '); end;';
    $st2 = oci_parse($conn, $q);
    oci_execute($st2);
    die();
}
}
}

```

```

}

$qq = 'begin KB_CREATE_QUEST(' . $prg . '); end;';
$st2 = oci_parse($conn, $qq);
oci_execute($st2);

function processNOK($nok) {

    $debug = false;
    $elts = [];
    $state = 0;
    $elt = '';
    $brack = 0;
    for($i = 0; $i < strlen($nok); $i++) {
        $char = substr($nok, $i, 1);
        // početak
        if($state === 0) {
            if($char === " " || $char === "," || $char === "'") {
                if($elt === '') {
                    continue;
                } else {
                    $elts[] = $elt;
                    $elt = '';
                }
            } elseif($char === '(') {
                if($elt === '') {
                    $brack++;
                    $state = 1;
                    $elt .= $char;
                }
            } else {
                $elt .= $char;
            }
        } elseif($state === 1) {
            $elt .= $char;
            if($char === '(') {
                $brack++;
            } elseif($char === ')') {
                $brack--;
                if($brack === 0) {
                    $subElts = processNOK(substr($elt, 1, strlen($elt)-2));
                    $elts[] = $subElts;
                    $elt = '';
                    $state = 0;
                }
            }
        }
    }
}

```

```

}

if($state !== 0) {
    echo '<pre>ERROR [' . $nok . ']</pre>';
    return [];
} else {
    if($elt !== '' && $elt !== ')') {
        $elts[] = $elt;
        $elt = '';
    }
}
if($debug) {
echo '<strong>ProcessNOK: ' . $nok . '</strong><br />';
}
if(is_array($elts) && is_array($elts[0])) {
    return $elts;
}
if(count($elts) === 2) {
    if(strpos($elts[0], '?') !== false) {
        $q = $elts[0];
        if(substr($q, 0, 1) === '') {
            $q = substr($q, 1);
        }
        if(substr($q, strlen($q) - 1, 1) === '') {
            $q = substr($q, 0, strlen($q) - 1);
        }
        if(substr($q, strlen($q) - 1, 1) === '?') {
            $q = substr($q, 0, strlen($q) - 1);
        }
        $ret = ['Q' => $q, 'A' => $elts[1]];
    } else {
        $ret = ['V' => $elts[0], 'F' => $elts[1]];
    }
    return $ret;
} else {
    $newElts = [];
    $newNodes = 0;
    if($debug) {
        echo '<div class="row">' .
            '<div class="col-6"><pre>' . var_export($elts, true) .
'</pre></div>' .
            '<div class="col-6"><pre>';
    }
    for($i = 0; $i < count($elts); $i++) {
        $elt = $elts[$i];
        if(is_array($elt)) {
            $newElts[] = $elt;
            continue;

```

```

}
if(strpos($elt, '?') !== false || $elt === '_' || $elt === '&') {
    // question word
    $q = $elt;
    if(substr($q, 0, 1) === '') {
        $q = substr($q, 1);
    }
    if(substr($q, strlen($q) - 1, 1) === '') {
        $q = substr($q, 0, strlen($q) - 1);
    }
    if(substr($q, strlen($q) - 1, 1) === '?') {
        $q = substr($q, 0, strlen($q) - 1);
    }
    if(isConjunctionW($elts[$i+1])) {
        $newElts[] = [
            'Q' => $q,
            'C' => $elts[$i+1],
            'O' => $elts[$i+2],
        ];
        $i += 2;
        continue;
    } else {
        $ans = $elts[$i+1];
        if(substr($ans, strlen($ans) - 1, 1) === ',') {
            $ans = substr($ans, 0, strlen($ans) - 1);
        }

        if(isset($elts[$i+2]) && is_array($elts[$i+2])){
            $newElts[] = [
                'Q' => $q,
                'C' => $ans,
                'O' => $elts[$i+2]
            ];
            $i += 2;
        } else {
            $newElts[] = [
                'Q' => $q,
                'A' => $ans
            ];
            $i++;
        }

        continue;
    }
} else {
    // VERB
    $newElts[] = [

```

```

        'V' => $elts[$i],
        'F' => $elts[$i + 1],
    ];
    $i++;
    continue;
    // conjunction -> other node are conjuncted
}
}
if($debug) {
    echo var_export($newElts, true) . '</pre></div>' .
        '</div>' . '<hr />';
}
return $newElts;
}
return $elts;
}

function processNOKArray($nok, $prg, $ord, $parent, $parWrd, $parWrdId) {
    global $conn;

    global $currNodeId;
    $currNodeId++;
    $proposedNodeId = $currNodeId;
    $nodeId = $proposedNodeId;
    $node = '';
    $quest = '';
    $kbParWrdId = '';
    $kbParWrd = '';
    $kbNodWrdId = '';
    $kbNodQstId = '';

    $toProcess = false;

    if(isset($nok['V']) && $nok['V'] === '&') {
        $nok = [
            'Q' => '&',
            'A' => $nok['F'],
        ];
    }

    // two options -> VERB or QUESTION
    if(isset($nok['V'])) {
        echo '<p><code>case-V</code></p>';
        $node = $nok['V'];
        $quest = '_';
        if(is_array($nok['F'])) {
            $toProcess = $nok['F'];
        } else {

```

```

        $node = $nok['F'];
    }
    echo '<p><code>toProcess1: ' . var_export($nok['F'], true) . '</code></p>';
} elseif(isset($nok['Q'])) {
    $quest = $nok['Q'];
    // two options -> ANSWER or CONJUNCTION
    if(isset($nok['C'])) {
        $node = $nok['C'];
        $toProcess = $nok['O'];
        echo '<p><code>case-QC</code></p>';
        echo '<p><code>toProcess2: ' . var_export($nok['O'], true) .
'</code></p>';
    } else {
        $node = $nok['A'];
    }
} else {
    // array of Q-A
    $currNodeId--;
    foreach($nok as $idx => $val) {
        processNOKArray($val, $prg, $ord, $parent, $parWrd, $parWrdId);
    }
    return;
}

$questId = getQuestId($quest);
$nodWrdId = getWordId($node);

$q = 'insert into NOK_QFNOK (prg, ord, node_id, node, quest, parent_id,
kb_par_wrd_id, kb_par_wrd, kb_nod_wrd_id, kb_nod_qst_id) ' .
    "\n" .
    'values (' . $prg . ', ' . $ord . ', ' . $nodeId . ', :node, \'' .
$quest . '\', ' . $parent . ', ' . $parWrdId . ', :parWrd, ' . $nodWrdId . ', ' . $questId
. ')';

echo '<pre>' . $q . '</pre>';

$st = oci_parse($conn, $q);
$bindOk = oci_bind_by_name($st, ':node', $node);
$bindOk = oci_bind_by_name($st, ':parWrd', $parWrd);

echo '<pre>' . $parWrd . ' - ' . $quest . ' - ' . $node . '</pre>';

$ok = oci_execute($st);
if($ok) {
    //echo '<pre>' . $q . '</pre>';
} else {
    echo '<code><pre>' . $q . '</pre></code>';
}

```

```

    if($toProcess) {
        foreach($toProcess as $idx => $val) {
            if(!is_numeric($idx)) {
                processNOKArray($toProcess, $prg, $ord, $nodeId, $node, $nodWrdId);
                break;
            }
            processNOKArray($val, $prg, $ord, $nodeId, $node, $nodWrdId);
        }
    }

}

// unos pitanja
// KB_QUESTION_QUESTION
// NOK_QUESTION_QUESTION
// NOK_QUESTION_QUESTION_ALL -> prvi
// NOK_QUESTION_QUESTION_IMP -> prvi

// unos rečenice
// KB_TEXT_QUESTION
// NOK_QUESTION_QUESTION
// NOK_QUESTION_QUESTION_ALL -> prvi
// NOK_QUESTION_QUESTION_IMP -> prvi
?>
</div>
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-C6RzsynM9kWDrmNeT87bh950GNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL"
crossorigin="anonymous"></script>
</body>
</html>

```



## Privitak 4. PHP skripta functions.php

```
<?php
$conn = false;
function connect() {
    global $conn;
    $conn = oci_connect(USER, PASS, DBNAME);
    if($conn) {
        return $conn;
    }

    $e = oci_error();
    echo '<div class="alert alert-danger" role="alert">' .
        '<p>Greška kod povezivanja s bazom podataka</p>' .
        '<p class="mb-0"><strong>' . $e['message'] . '</strong></p>' .
        '</div>';

    die();
}

function getWordPar($wId) {
    global $dict;
    global $words;
    if(!$wId) {
        return false;
    }
    $w = $words[$wId];
    if(isset($words[$wId]['DICTIONARY_ID'])) {
        $par = $dict[$words[$wId]['DICTIONARY_ID']]['PAR'];
        return $par;
    }
    return false;
}

$getWordParsCache = [];
function getWordPars2($wId, $findSynonyms = false) {

    global $getWordParsCache;
    $key = $wId . '/' . $findSynonyms;
    if(isset($getWordParsCache[$key])) {
        return $getWordParsCache[$key];
    }

    global $dict;
    global $words;
    global $synonyms2;
    if(!$wId) {
        return [];
    }
}
```

```

}
$ret = [];

if(isset($words[$wId])) {
    $ret[] = $wId;
}

$i = 0;
while(true) {
    if(!isset($ret[$i])) break;

    $r = $ret[$i];

    if($words[$r]['DICT_ID']) {
        $dicts = [$words[$r]['DICT_ID']];
        if($dict[$words[$r]['DICT_ID']]['DICT_PAR']) {
            $dicts[] = $dict[$words[$r]['DICT_ID']]['DICT_PAR'];
        }
        while(true) {
            $done = true;
            foreach($dict as $idx => $val) {
                if(in_array($val['DICT_ID'], $dicts)) {
                    if(!in_array($val['DICT_PAR'], $dicts)) {
                        $dicts[] = $val['DICT_PAR'];
                        $done = false;
                    }
                }
                if(in_array($val['DICT_PAR'], $dicts)) {
                    if(!in_array($val['DICT_ID'], $dicts)) {
                        $dicts[] = $val['DICT_ID'];
                        $done = false;
                    }
                }
            }
            if($done) break;
        }
        $newW = getDictWords($dicts);
        foreach($newW as $idx => $val) {
            if(!in_array($val, $ret)) {
                $ret[] = $val;
            }
        }
    }
}

if($findSynonims) {
    if(isset($synonims2[$r])) {
        foreach($synonims2[$r] as $rbr => $syn) {
            if(!in_array($syn['WORD_ID2'], $ret)) {

```

```

        $ret[] = $syn['WORD_ID2'];
    }
}
}
}

    $i++;
}
$getWordParsCache[$key] = $ret;
return $ret;
}
function getDictWords($dictId) {
    global $words;
    $ret = [];

    foreach($words as $idx => $val) {
        if(in_array($val['DICT_ID'], $dictId)) {
            $ret[] = $val['WORD_ID'];
        }
    }

    return $ret;
}
function getWordPars($wId, $findSynonyms = true) {
    global $dict;
    global $words;
    global $synonyms;
    if(!$wId) {
        return [];
    }
    $ret = [];
    if(isset($words[$wId]) && $words[$wId]['DICT_ID']) {
        $ret[] = $words[$wId]['DICT_ID'];
    }

    $debug = false;

    $i = 0;
    while(true) {
        if(!isset($ret[$i])) { break; }
        $r = $ret[$i];
        if(isset($dict[$r])) {
            if(!in_array($dict[$r]['DICT_PAR'], $ret)) {
                $ret[] = $dict[$r]['DICT_PAR'];
            }
        }
        foreach($dict as $d) {
            if($d['DICT_PAR'] === $r) {

```

```

        if(!in_array($d['DICT_PAR'], $ret)) {
            $ret[] = $d['DICT_PAR'];
        }
    }
}
if($findSynonims) {
    if(isset($synonims[$r])) {
        foreach($synonims[$r] as $syn) {
            if(!in_array($syn['DICT_ID2'], $ret)) {
                $ret[] = $syn['DICT_ID2'];
            }
        }
    }
}
$i++;
}

return $ret;
}

function printPars2($pars) {
    global $words;
    $ret = '';
    foreach($pars as $p) {
        $ret .= $words[$p]['WORD'] . ' ';
    }
    return $ret;
}

function printPars($pars) {
    global $dict;
    $ret = '';
    foreach ($pars as $p) {
        $ret .= $dict[$p]['DICT'] . ' ';
    }
    return $ret;
}

function getWord($wId, $findSynonims = false) {
    global $dict;
    global $words;
    if(!$wId) {
        return '';
    }
    $w = $words[$wId];
    $d = '';
    if(isset($words[$wId]['DICT_ID'])) {
        $par = $dict[$words[$wId]['DICT_ID']]['DICT_PAR'];
        foreach($dict as $idx => $val) {
            if($val['DICT_PAR'] === $par) {
                $d .= '[' . $val['DICT'] . ' ';
            }
        }
    }
}

```

```

        }
    }
}
if($d) {
    $d = '<div><small>' . $d . '</small></div>';
}

$pars = getWordPars2($wId, $findSynonims);
$pars2 = printPars2($pars);

return $w['WORD'] . '<div><small>' . $pars2 . '</small></div>';
}
function getQuest($qId) {
    global $quest;
    if(!$qId) {
        return '';
    }
    $q = $quest[$qId];
    return $q['QUEST'];
}

// dohvaćanje globalnih podataka (rječnik, pitanja, veznici)
$words = [];
$dict = [];
$quest = [];
$conj = [];
$synonims = [];
$synonims2 = [];
function loadData() {
    global $words, $dict, $quest, $conn, $conj, $synonims, $synonims2;
    // KB_WORD
    $q = 'select * from KB_WORD';
    $st = oci_parse($conn, $q);
    $r = oci_execute($st);
    $words = [];
    while($row = oci_fetch_assoc($st)) {
        $words[$row['WORD_ID']] = $row;
    }

    // KB_DICT
    $q = 'select * from KB_DICT';
    $st = oci_parse($conn, $q);
    $r = oci_execute($st);
    $dict = [];
    while($row = oci_fetch_assoc($st)) {
        $dict[$row['DICT_ID']] = $row;
    }
}

```

```

// KB_QUESTION
$q = 'select * from KB_QUESTION';
$st = oci_parse($conn, $q);
$r = oci_execute($st);
$quest = [];
while($row = oci_fetch_assoc($st)) {
    $quest[$row['QUESTION_ID']] = $row;
}

// KB_CONJ
$q = 'select * from KB_CONJ';
$st = oci_parse($conn, $q);
$r = oci_execute($st);
$conj = [];
while($row = oci_fetch_assoc($st)) {
    $conj[$row['CONJ_ID']] = $row;
}

// KB_SYNONYM
$q = 'select * from KB_SYNONYM';
$st = oci_parse($conn, $q);
$r = oci_execute($st);
$synonyms = [];
$synonyms2 = [];
while($row = oci_fetch_assoc($st)) {
    if(!isset($synonyms[$row['DICT_ID1']])) {
        $synonyms[$row['DICT_ID1']] = [];
    }
    $synonyms[$row['DICT_ID1']][] = $row;
    if(!isset($synonyms2[$row['WORD_ID1']])) {
        $synonyms2[$row['WORD_ID1']] = [];
    }
    $synonyms2[$row['WORD_ID1']][] = $row;
}
}

// traženje odgovora na odabrano pitanje
function getAnswers($prg, $ord) {
    global $conn;

    $ret = [];

    $t1 = microtime(true);

    // čitanje podataka o pitanju
    $qfnok = [];
    $q = 'select q.*, level ' .

```



```

}

// reset strukture za pitanje
$numFound = 0;
foreach($qElts as $i => $q) {
    if(intval($qElts[$i]['NOD_QST_ID']) === 4) {
        $qElts[$i]['FOUND'] = true;
    } else {
        $qElts[$i]['FOUND'] = false;
    }
}

foreach($s['elts'] as $iS => $sElt) {
    $s['elts'][$iS]['FOUND'] = false;
}

$foundPars = [];

foreach($s['elts'] as $iS => $sElt) {
    $found = false;
    foreach($qElts as $iQ => $qElt) {
        if($qElts[$iQ]['FOUND'] === true) continue;
        if(equalWords($qElt['PAR_WRD_ID'], $sElt['PAR_WRD_ID']) &&
            equalWords($qElt['NOD_WRD_ID'], $sElt['NOD_WRD_ID'])) {

            // jednaki čvorovi
            $found = true;
            $qElts[$iQ]['FOUND'] = true;
            $numFound++;
            $qElts[$iQ]['FOUND_ID'] = $iS . '/1';
            $sentences[$idx]['elts'][$iS]['FOUND'] = true;
            $s['elts'][$iS]['FOUND'] = true;
            $foundPars[] = $s['elts'][$iS]['PARENT_ID'];
            $foundPars[] = $s['elts'][$iS]['NODE_ID'];

            break;
        } elseif(equalWords($qElt['PAR_WRD_ID'], $sElt['PAR_WRD_ID']) &&
            isConjunction($sElt['NOD_WRD_ID'])) {

            // prvi element odgovara, drugi u rečenici je veznik -> provjera direktne djece
            $parNode = false;
            foreach($s['fnok'] as $iF => $f) {
                if($parNode === false && $f['KB_NOD_WRD_ID'] ===
                    $sElt['NOD_WRD_ID']) {
                    $parNode = $f['NODE_ID'];
                } elseif($f['PARENT_ID'] === $parNode) {
                    if(equalWords($qElt['NOD_WRD_ID'], $f['KB_NOD_WRD_ID'])) {

```



```

    $found = true;
    $qElts[$iQ]['FOUND'] = true;
    $numFound++;
    $qElts[$iQ]['FOUND_ID'] = $iS . '/2';
    foreach($s['elts'] as $i1 => $v1) {
        if($v1['NOD_WRD_ID'] === $f['KB_NOD_WRD_ID']) {
            $sentences[$idx]['elts'][$i1]['FOUND'] = true;
            $s['elts'][$i1]['FOUND'] = true;
            $foundPars[] = $s['elts'][$i1]['PARENT_ID'];
            $foundPars[] = $s['elts'][$i1]['NODE_ID'];
            $s['elts'][$iS]['FOUND'] = true;
            $foundPars[] = $s['elts'][$iS]['PARENT_ID'];
            $foundPars[] = $s['elts'][$iS]['NODE_ID'];
            $numFound++;
            break;
        }
    }
    break;
}
}
}
if($found) {
    break;
}
}

    if(equalWords($qElt['NOD_WRD_ID'], $sElt['NOD_WRD_ID']) &&
isConjunction($qElt['PAR_WRD_ID'])) {
// drugi element odgovara, prvi element u pitanju je veznik -> provjera čvora roditelja od
veznika

    $qPar = false;
    foreach($qElts as $tmpIdx => $tmpQ) {
        if($tmpQ['NOD_WRD_ID'] === $qElt['PAR_WRD_ID']) {
            $qPar = $tmpQ;
            break;
        }
    }
    if($qPar) {
// ako je roditelj veznika jednak roditelju trenutnog čvora
        if(equalWords($qPar['PAR_WRD_ID'], $sElt['PAR_WRD_ID'])) {
            echo '<p>OK!!</p>';
            $found = true;
        }
    }

// ako je roditelj čvora u rečenici veznik -> provjera njegovog roditelja i roditelja
veznika iz pitanja

    if(!$found && isConjunction($sElt['PAR_WRD_ID'])) {
        $sPar = false;
    }
}

```

```

foreach($s['elts'] as $tmpSIdx => $tmpS) {
    if($tmpS['NOD_WRD_ID'] === $sElt['PAR_WRD_ID']) {
        $sPar = $tmpS;
        break;
    }
}
if($sPar) {

    if(equalWords($qPar['PAR_WRD_ID'], $sPar['PAR_WRD_ID'])) {
        $found = true;
    }
}
}

if($found) {
    // čvor je pronađen, provjera tipa veznika
    $conjType = getConjType($qElt['PAR_WRD_ID']);

    // trenutni čvor je sigurno pronađen
    $qElts[$iQ]['FOUND'] = true;
    $numFound++;
    $sentences[$idx]['elts'][$iS]['FOUND'] = true;
    $s['elts'][$iS]['FOUND'] = true;
    $foundPars[] = $s['elts'][$iS]['NODE_ID'];
    $foundPars[] = $s['elts'][$iS]['PARENT_ID'];

    if($conjType === '0') {
        // veznik je ILI -> jedan pronađeni čvor = svi su pronađeni
        foreach($qElts as $tmpIdx => $tmpQ) {
            if($tmpQ['NOD_WRD_ID'] === $qElt['PAR_WRD_ID'] ||
$tmpQ['PAR_WRD_ID'] === $qElt['PAR_WRD_ID']) {
                $qElts[$tmpIdx]['FOUND'] = true;
            }
        }
    }

    break;
}
}

if(equalWords($qElt['NOD_WRD_ID'], $sElt['NOD_WRD_ID'])) {
    // drugi element odgovara (npr. u pitanju je colour-he, u rečenici
drew-he, pronašli smo HE)

    // roditelj prvog elementa mora biti veznik (mora postojati and-drew)
    $found = false;
    $conjId = false;
}

```

```

foreach($s['elts'] as $tmpIdx => $tmpS) {
    if($tmpS['NOD_WRD_ID'] === $sElt['PAR_WRD_ID']) {
        if(isConjunction($tmpS['PAR_WRD_ID'])) {
            $found = true;
            $conjId = $tmpS['PAR_WRD_ID'];
            break;
        }
    }
}

if($found) {
    // dijete tog istog veznika mora biti roditelj kojeg tražimo (mora postojati and-colour)
    $found = false;
    foreach($s['elts'] as $tmpIdx => $tmpS) {
        if($tmpS['PAR_WRD_ID'] === $conjId &&
equalWords($qElt['PAR_WRD_ID'], $tmpS['NOD_WRD_ID'])) {
            $found = true;
            break;
        }
    }

    if($found) {

        // ne postoji čvor s odgovorom na isto pitanje za roditelja (ne postoji colour-who)
        $found = false;
        foreach($s['elts'] as $tmpIdx => $tmpS) {
            if(equalWords($qElt['PAR_WRD_ID'], $tmpS['PAR_WRD_ID']) &&
$tmpS['NOD_QST_ID'] === $sElt['NOD_QST_ID']) {
                $found = true;
                break;
            }
        }

        if(!$found) {
            // zadovoljeni su svi uvjeti
            $qElts[$iQ]['FOUND'] = true;
            $numFound++;
            $sentences[$idx]['elts'][$iS]['FOUND'] = true;
            $s['elts'][$iS]['FOUND'] = true;
            $foundPars[] = $s['elts'][$iS]['NODE_ID'];
            $foundPars[] = $s['elts'][$iS]['PARENT_ID'];

            // moraju biti pronađena i ostala djeca veznika koji je
            roditelj pronađenog čvora, inače neće pronaći roditelje
            foreach($s['elts'] as $tmpIdx => $tmpS) {
                if($tmpS['PAR_WRD_ID'] === $conjId) {
                    $s['elts'][$tmpIdx]['FOUND'] = true;

```

```

        if(!in_array($s['elts'][$tmpIdx]['NODE_ID'],
$foundPars)) $foundPars[] = $s['elts'][$tmpIdx]['NODE_ID'];
        if(!in_array($s['elts'][$tmpIdx]['PARENT_ID'],
$foundPars)) $foundPars[] = $s['elts'][$tmpIdx]['PARENT_ID'];
    }
}
break;
}
}
}
}

// ako čvor nije pronađen, a nema X -> nema odgovora u ovoj rečenici
// zasad se traži do kraja da vidimo ako sve radi jer performanse nisu u fokusu
}

if($debug) {
    echo '<pre>FP: ' . var_export($foundPars, true) . '</pre>';
}

$answer = false;
$answerWrđ = false;
$allFound = true;
$answerPrep = '';
foreach($qElts as $iQ => $qElt) {
    if($qElt['FOUND']) {
        continue;
    }
    $allFound = false;
    $answerB = false;
    foreach($s['elts'] as $iS => $sElt) {
        if($sElt['FOUND'] === true) {
            if($debug) echo '<pre>F1: ' . $sElt['PAR_WRD'] . ' - ' .
$sElt['NOD_WRD'] . '</pre>';
            continue;
        }
        if($qElt['NOD_QST_ID'] > 0 && $qElt['NOD_QST_ID'] !== $sElt['NOD_QST_ID'])
{
            // ako pitanje nije jednako, ali je čvor iznad veznik tipa B (because)
-> gledaj to pitanje
            $found = false;
            if(isConjunction($sElt['PAR_WRD_ID']) &&
getConjType($sElt['PAR_WRD_ID']) === 'B') {
                $found = true;
                foreach($s['fnok'] as $tmpI => $tmpS) {

```



```

    } else {
        if($answer === $sElt['PAR_WRD']) {
            $qElts[$iQ]['FOUND'] = true;
            $qElts[$iQ]['FOUND_ID'] = '/8';
            break;
        } else {
            $qElts[$iQ]['FOUND'] = true;
            $qElts[$iQ]['FOUND_ID'] = '/9';
            $answer .= ', ' . $sElt['PAR_WRD'];
            if($debug) {
                echo '<pre>C9 - ' . $answer . '</pre>';
            }
            $answerWrđ .= ", " . $sElt['PAR_WRD_ID'] . '- ' .
$sElt['PARENT_ID'];

            break;
        }
    }
}

if($debug && $qElt['NOD_WRD'] === 'X') {
    echo '<pre>X: ' . $qElt['NOD_WRD'] . ' / ' .
isConjunction($sElt['PAR_WRD_ID']) . ' / ' . getConjType($sElt['PAR_WRD_ID']) . ' / ' .
$sElt['NOD_WRD'] . '</pre>';
}
if($qElt['NOD_WRD'] === 'X' && isConjunction($sElt['PAR_WRD_ID']) &&
getConjType($sElt['PAR_WRD_ID']) === 'B' &&
equalWords($qElt['PAR_WRD_ID'], $sElt['NOD_WRD_ID']) &&
(count($foundPars) === 0 || in_array($sElt['NODE_ID'], $foundPars))
) {

    if($debug) echo '<pre>NOVI slučaj [' . $qElt['PAR_WRD'] . '- ' .
$qElt['NOD_WRD'] . ']' [' . $sElt['PAR_WRD'] . '- ' . $sElt['NOD_WRD'] . ']' [oldA: ' .
var_export($answer, true) . ' / ' . var_export($answerWrđ, true) .
']</pre>';
    // start
    $ok = false;
    foreach($s['fnok'] as $tmpI => $tmpS) {
        if($tmpS['KB_NOD_WRD_ID'] === $sElt['PAR_WRD_ID'] &&
$tmpS['KB_NOD_QST_ID'] === $qElt['NOD_QST_ID']) {
            $ok = true;
            break;
        }
    }
    // end

    if($debug) {
        echo '<pre>OK: ' . var_export($ok, true) . '</pre>';
    }
}

```

```

        $found = false;
        if($ok) {
            foreach($s['elts'] as $tmpI => $tmpS) {
                if($debug) {
                    echo '<pre>[' . $tmpI . '] ' . $tmpS['PAR_WRD'] . ' - ' .
$tmpS['NOD_QST'] . ' - ' . $tmpS['NOD_WRD'] . '</pre>';
                    echo '<pre>  ' . $sElt['PAR_WRD'] . ' - ' .
$sElt['NOD_QST'] . ' - ' . $sElt['NOD_WRD'] . '</pre>';
                    echo '<pre>' . $tmpS['NOD_QST_ID'] . ' === ' .
$sElt['NOD_QST_ID'] . '</pre>';
                    echo '<pre>' . $tmpS['PAR_WRD_ID'] . ' === ' .
$sElt['PAR_WRD_ID'] . '</pre>';
                    echo '<pre>' . $tmpS['NOD_WRD_ID'] . ' === ' .
$sElt['NOD_WRD_ID'] . '</pre>';
                    echo '<pre>' . $tmpS['NODE_ID'] . ' !== ' .
$sElt['NODE_ID'] . '</pre>';
                }
                if($tmpS['NOD_QST_ID'] === $sElt['NOD_QST_ID'] &&
$tmpS['PAR_WRD_ID'] === $sElt['PAR_WRD_ID'] &&
                if($debug) {
                    echo '<pre>FOUND-NS</pre>';
                }
                $answer = $tmpS['NOD_WRD'];
                $answerWrds = $tmpS['NOD_WRD_ID'] . '-' . $tmpS['NODE_ID'];
                $qElts[$iQ]['FOUND'] = true;
                $found = true;
                $answerPrep = $sElt['PAR_WRD'] . ' ';
                $answerB = true;
                break;
            }
        }
        if($found) break;
    }
}
if($debug) {
    echo '<p>A: ' . $answer . ', W: ' . $answerWrds . '</p>';
}
if($allFound === true) {
    $answer = 'YES';
} elseif($answerWrds) {
    if($debug) {
        echo '<p>Get FULL answer [' . $answerWrds . ']</p>';
    }
    if(strpos($answerWrds, ',')) {
        $wrds = explode(',', $answerWrds);
        $answer = '';
    }
}

```

```

        $done = [];
        foreach($wrds as $idx => $wrđ) {
            if(in_array($wrđ, $done)) continue;
            $done[] = $wrđ;
            $answer .= ($idx > 0 ? ';' : '') . getFullAnswer($s, $wrđ, $answerB);
        }
    } else {
        $answer = $answerPrep . getFullAnswer($s, $answerWrđ, $answerB);
    }
}

$ok = true;
foreach($qElts as $iQ => $qElt) {
    if($qElt['FOUND'] === false) {
        $ok = false;
        break;
    }
}

if($ok) {
    $q = 'insert into KB_ANSWER(ANSWER_ID, PRGQ, ORDQ, QTENG, ANSWER, PRGS, ORDS,
TENG, QFNOK, FNOK) ' .
        'values ((select NVL(MAX(ANSWER_ID), 0)+1 from KB_ANSWER), :prgq, :ordq,
(select QTENG from NOK_QFNOK_ALL where PRG = :prgq and ORD = :ordq), :answer, :prgs, :ords,
:teng, (select QFNOK from NOK_QFNOK_ALL where PRG = :prgq and ORD = :ordq), :fnok)';
    $st = oci_parse($conn, $q);
    oci_bind_by_name($st, ':prgq', $prg);
    oci_bind_by_name($st, ':ordq', $ord);
    oci_bind_by_name($st, ':answer', $answer);
    oci_bind_by_name($st, ':prgs', $s['prg']);
    oci_bind_by_name($st, ':ords', $s['ord']);
    oci_bind_by_name($st, ':teng', $s['sentence']);
    oci_bind_by_name($st, ':fnok', $s['fnoks']);
    $r = oci_execute($st);
    if(!$r) {
        echo '<pre>ERROR: ' . oci_error($conn) . '</pre>';
    }
}

$ret['s'][] = [
    'answerOk' => $ok,
    'numFound' => $numFound,
    'answer' => $answer,
    'answerWrđ' => $answerWrđ,
    'sentence' => $s,
    'qElts' => $qElts,
];
}

```



```

    return $ret;
}

function getFullAnswer($s, $wordId, $answerB) {
    $debug = true;
    if(false) {
        echo '<pre>GET FULL ANSWER (' . $wordId . ') / ' . '</pre>';
        $debug = true;
    }

    $state = 0;
    $pars = [];
    $elts = [];

    $wrdfound = [];
    if($answerB === false) {
        foreach($s['elts'] as $idx => $val) {
            if($val['FOUND']) {
                $wrdfound[] = $val['NOD_WRD_ID'];
            }
        }
    }

    $wordId = explode('-', $wordId);
    $nodeID = $wordId[1];
    $wordId = $wordId[0];

    foreach($s['fnok'] as $idx => $n) {
        if($state === 0 && ($n['KB_NOD_WRD_ID'] == $wordId && $n['NODE_ID'] === $nodeID)) {
            $state = 1;
            $pars[] = $n['NODE_ID'];
            $elts[] = $n;
        } elseif($state === 1) {
            if(in_array($n['KB_NOD_WRD_ID'], $wrdfound)) {
                continue;
            }
            if(in_array($n['PARENT_ID'], $pars)) {
                $pars[] = $n['NODE_ID'];
                $elts[] = $n;
            } else {
                break;
            }
        }
    }

    $v1 = '';
    $a1 = printAnswer($elts, 0, '');
    return $a1;
}

```

```

$answer = '';
$currConj = false;
$conjIdx = false;
foreach($elts as $idx => $elt) {
    if(isConjunction($elt['KB_NOD_WRD_ID'])) {
        $currConj = $elt;
        $conjIdx = $idx;
        continue;
    } else {
        if($currConj && $elt['PARENT_ID'] !== $currConj['NODE_ID']) {
            $currConj = false;
            $conjIdx = false;
        }
        if($currConj !== false && $idx > $conjIdx + 1) {
            $answer .= ' ' . $currConj['NODE'];
        }
        $answer .= ' ' . $elt['NODE'];
    }
}
return $answer;
}

function printAnswer($elts, $idx, $ans) {
    $elt = $elts[$idx];
    if($elt['KB_NOD_QST_ID'] === '-1') {
        $ans .= ' ' . $elt['NODE'];
        foreach($elts as $i => $v) {
            if($v['PARENT_ID'] === $elt['NODE_ID']) {
                $ans .= printAnswer($elts, $i, '');
            }
        }
        return $ans;
    } else {
        $art = '';
        $conj = false;
        $first = true;
        if(isConjunction($elt['KB_NOD_WRD_ID'])) {
            $conj = true;
        }
        foreach($elts as $i => $v) {
            if($v['PARENT_ID'] === $elt['NODE_ID']) {
                // KB_NOD_QST_ID=1 => ART -> članovi idu prije riječi na koju se odnose
                if($v['KB_NOD_QST_ID'] === '1') {
                    $art = $v['NODE'] . ' ';
                } else {
                    if($conj) {
                        if(!$first) {
                            $ans .= ' ' . $elt['NODE'];
                        }
                    }
                }
            }
        }
    }
}

```

```

        }
        $first = false;
    }
    $ans .= printAnswer($elts, $i, '');
}
}
}
}
$ans .= ' ' . $art . ($conj ? '' : $elt['NODE']);
return $ans;
}
}

function isConjunction($wordId) {
    global $conj;
    foreach($conj as $idx => $c) {
        if($c['WORD_ID'] == $wordId) {
            return true;
        }
    }
    return false;
}

function getConjType($wordId) {
    global $conj;
    foreach($conj as $idx => $c) {
        if($c['WORD_ID'] == $wordId) {
            return $c['CONJ_TYPE'];
        }
    }
    return false;
}

function isConjunctionW($word) {
    global $conj;
    foreach($conj as $idx => $c) {
        if(strtoupper($c['CONJ']) == strtoupper($word)) {
            return true;
        }
    }
    return false;
}

function getSentences() {
    global $conn;

    $q = 'select * from NOK_FNOK_ALL';
    $st = oci_parse($conn, $q);
    $r = oci_execute($st);

```

```

$ret = [];

$q2 = 'select q.*, level ' .
      'from NOK_FNOK q ' .
      'connect by prior node_id = parent_id and prior prg = prg and prior ord = ord ' .
      'start with parent_id = 0';
$s2 = oci_parse($conn, $q2);
$r2 = oci_execute($s2);
$hierarchy = [];
while($row2 = oci_fetch_assoc($s2)) {
    $key = $row2['PRG'] . '-' . $row2['ORD'];
    if(!isset($hierarchy[$key])) {
        $hierarchy[$key] = [];
    }
    $hierarchy[$key][] = $row2;
}

$q2 = 'select * from KB_TEXT_QUEST';
$s2 = oci_parse($conn, $q2);
$r2 = oci_execute($s2);
$sentElts = [];
while($row2 = oci_fetch_assoc($s2)) {
    $key = $row2['PRG'] . '-' . $row2['ORD'];
    if(!isset($sentElts[$key])) {
        $sentElts[$key] = [];
    }
    $w1 = getWord($row2['PAR_WRD_ID']);
    $w2 = getWord($row2['NOD_WRD_ID']);
    $w3 = getQuest($row2['NOD_QST_ID']);
    $row2['w1'] = $w1;
    $row2['w2'] = $w2;
    $row2['w3'] = $w3;
    $sentElts[$key][] = $row2;
}

while($row = oci_fetch_assoc($st)) {
    $sentence = $row['TENG'];
    $fnokS = $row['FNOK'];
    $sPrg = $row['PRG'];
    $sOrd = $row['ORD'];

    $fnok = $hierarchy[$row['PRG'] . '-' . $row['ORD']];

    $sElts = $sentElts[$row['PRG'] . '-' . $row['ORD']];

    $ret[] = [

```

```

        'sentence' => $sentence,
        'prg' => $sPrg,
        'ord' => $sOrd,
        'fnoks' => $fnokS,
        'fnok' => $fnok,
        'elts' => $sElts
    ];
}

return $ret;
}

function getQuestId($questWrd) {
    if($questWrd === '_' ) {
        return -1;
    }
    global $quest;
    foreach($quest as $idx => $val) {
        if(strtoupper($val['QUEST']) === strtoupper($questWrd)) {
            return $idx;
        }
    }

    echo '<pre>New QUEST</pre>';

    global $conn;
    $q = 'select max(QUEST_ID) + 1 as QUEST_ID from KB_QUESTION';
    $st = oci_parse($conn, $q);
    $r = oci_execute($st);
    $row = oci_fetch_assoc($st);

    $newId = $row['QUEST_ID'];

    $q = 'insert into KB_QUESTION (QUEST_ID, QUESTION) values (' . $newId . ', :quest)';
    $st = oci_parse($conn, $q);
    oci_bind_by_name($st, ':quest', $questWrd);
    $r = oci_execute($st);

    $quest[$newId] = [
        'QUEST_ID' => $newId,
        'QUESTION' => $questWrd
    ];

    return $newId;
}

function getWordId($word) {
    global $words;
    foreach($words as $idx => $val) {

```

```

        if(strtoupper($val['WORD'])
        === strtoupper($word)) {
            return $idx;
        }
    }

    echo '<pre>New WORD</pre>';
    global $conn;
    $q = 'select max(WORD_ID) + 1 as WORD_ID from KB_WORD';
    $st = oci_parse($conn, $q);
    $r = oci_execute($st);
    $row = oci_fetch_assoc($st);

    $newId = $row['WORD_ID'];

    $q = 'insert into KB_WORD (WORD_ID, WORD, DICT_ID) values (' . $newId . ', :word,
null)';
    $st = oci_parse($conn, $q);
    oci_bind_by_name($st, ':word', $word);
    $r = oci_execute($st);

    $words[$newId] = [
        'WORD_ID' => $newId,
        'WORD' => $word,
        'DICT_ID' => ''
    ];

    return $newId;
}

function equalWords($w1, $w2) {
    if($w1 === $w2) {
        return true;
    }
    $par1 = getWordPars2($w1, true);
    $par2 = getWordPars2($w2);
    foreach($par1 as $p1) {
        foreach($par2 as $p2) {
            if($p1 === $p2) {
                return true;
            }
        }
    }
    return false;
}

```

## Privitak 5. Rečenice korištene u inicijalnom testiranju

R.B.	TENG	FNOK
1.	Tom and Paul talk about the solution.	talk ("who?" and ("who?" Tom, "who?" Paul), "what?" solution ("art?" the, "what?" about))
2.	Peter trains basketball, volleyball and tennis.	trains ("who?" Peter, "what?" and ("what?" basketball, "what?" volleyball, "what?" tennis))
3.	We were wet and tired.	were ("who?" we, "what?" and ("what?" wet, "what?" tired))
4.	He drew a heart and coloured it red.	"what?" and (drew ("who?" he, "what?" heart ("art?" a)), coloured ("what?" it, "how?" red))
5.	Lucija is from Zagreb or Rijeka.	is ("who?" Lucija, "where?" from ("where?" or ("where?" Zagreb, "where?" Rijeka))
6.	You should eat more, or you'll make yourself ill.	"what?" or (should (eat ("who?" you, "what?" more)), will (make ("who?" you, "whom?" yourself, "what?" ill "))))
7.	Sanja is happy because she loves her interesting hobby.	"why?" because (is ("who?" Sanja, "what?" happy), loves ("who?" she, "what?" hobby ("whose?" her, "what?" interesting)))
8.	I didn't go out because I had no money.	"why?" because (didn't (go ("who?" I, "where?" out)), had ("who?" I, "what?" money ("how_much?" no)))
9.	Vanessa is sleepy because she has been studying all night.	"why?" because (is ("who?" Vanessa, "what?" sleepy), has (been (studying ("who?" she, "when?" night ("what?" all))))))
10.	We ran after him, but he escaped.	"what?" but (ran ("who?" we, "how?" after, "whom?" him), escaped ("who?" he))
11.	He felt nothing but contempt for her.	felt ("who?" he, "what?" nothing, "what?" but ("what?" contempt), "whom?" her ("whom?" for))
12.	I loved maths when I was at school.	"when?" when (loved ("who?" I, "what?" math), was ("who?" I, "where?" school ("where?" at)))
13.	I had a great time when I went to the coast.	"when?" when (had ("who?" I, "what?" time ("what?" great ("art?" a))), went ("who?" I, "where?" coast ("art?" the ("where?" to))))
14.	I wonder if she noticed.	"what?" if (wonder ("who?" I), noticed ("who?" she))
15.	You can go here if you get the point.	"what?" if (can (go ("who?" you, "where?" here)), get ("who?" you, "what?" point ("art?" the)))
16.	You will feel cold if you don't wear a coat.	"what?" if (will (pay ("who?" I, "how?" double, "whom?" you)), get ("who?" you, "what?" work ("art?" the), "what" finished, "when?" Friday ("when?" by)))
17.	Monika is glad that workers aren't losing their jobs.	"what?" that (is ("who?" Monika, "what?" glad), aren't (losing ("who?" workers, "what?" jobs ("whose?" their))))
18.	It was still painful, so I went to see a specialist.	"why?" so (was ("what?" it, "what?" still, "how?" painful), went ("who?" I, "what?" see ("what?" to), "who?" specialist ("art?" a)))
19.	You can ask for help whenever you need it.	"when?" whenever (can (ask ("who?" you, "what?" help ("what?" for))), need ("who?" you, "what?" it))
20.	He minded the dog while I was away.	"when?" while (minded ("who?" he, "what?" dog ("art?" the)), was ("who?" I, "where?" away))

## Privitak 6. Pitanja korištena u inicijalnom testiranju i pripadajući QFNOK-ovi

R.B.	QTENG	QFNOK
1.	What does Tom and Paul talking about?	talking ( & are) ( _ and ( _ Paul, _ Tom), "what?" X ( _ about))
2.	Who talks about the solutions?	talks ("who?" X, _ solution ( _ about, _ the))
3.	Who talks to Tom?	talks ( "who?" X, _ Tom ( _ to))
4.	Who talks to Paul?	talks ( "who?" X, _ Paul ( _ to))
5.	What does Peter train?	train ( & does) ( _ Peter, "what?" X)
6.	Who trains basketball?	trains ("who?" X, _ basketball)
7.	Who trains basketball, volleyball or tennis?	trains ( "who?" X, _ or ( _ basketball , _ volleyball, _ tennis))
8.	What were we?	were ( _ we, "what?" X )
9.	Who were wet?	were ("who?" X, _ wet)
10.	Who were tired?	were ("who?" X, _ tired)
11.	Who were wet and tired?	were ("who?" X, _ and ( _ wet, _ tired))
12.	What did he draw and colour?	_ and (& did) (draw ( _ he, "what?" X)), colour)
13.	What did he draw?	draw ( & did) ( _ he, "what?" X)
14.	What did he colour?	colour ( & did) ( _ he, "what?" X)
15.	How did he colour the heart?	colour ( & did) ( ( _ he ( _ heart ( _ the), "how?" X)
16.	Who drew the heart?	drew ("who?" X, _ heart ( _ a))
17.	Where is Lucija from?	is ( _ Lucija , "where?" X ( _ from))
18.	Who is from Zagreb?	is ("who?" X, _ from ( _ Zagreb))
19.	Who is from Rijeka?	is ("who?" X, _ from ( _ Rijeka))
20.	Who is from Zagreb or Rijeka?	is ("who?" X, _ from ( _ or ( _ Zagreb, _ Rijeka)))
21.	Who should eat more?	eat ( & should) ("who?" X, _ more)
22.	Who will make yourself ill?	make ( & will) ( "who?" X, _ yourself, _ ill)
23.	Why should you eat more?	why?" X (should (eat ( _ you, _ more)))
24.	Who is happy?	is ("who?" X, _ happy)
25.	Who loves her hobby?	loves ("who?" X, _ hobby ( _ her))
26.	What is Sanja?	is ( _ Sanja, "what?" X)
27.	Why is Sanja happy?	is ( _ Sanja, _ happy, "why?" X)
28.	Who didn't go out?	didn't (go ("who?" X, _ out))
29.	Who had no money?	had ("who?" X, _ money ( _ no))
30.	What did not I had?	had ( & dont) ( _ I, "what?" X)
31.	Why did not I go out?	didn't (go ( _ I, _ out), "why?" X)
32.	Who is sleepy?	is ("who?" X, _ sleepy)
33.	What is Vanessa?	is ( _ Vanessa, "what?" X)
34.	Who has been studying all night?	studying ( & has, & been) ("who?" X, _ night ( _ all))
35.	Why is Vanessa sleepy?	is ( _ sleepy, _ Vanessa, "why?" X)
36.	Who is running after him?	running ( & is) ( "who?" X, _ after, _ him )



37.	Who is running after him?	running ( & is) ( "who?" X, _ after, _ him )
38.	Who escaped?	escaped ("who?" X)
39.	What did he felt for her?	felt ( & did) ( _ he, "what?" X, _ her ( _ for))
40.	Who felt nothing?	felt ("who?" X, _ nothing)
41.	Who felt contempt?	felt ("who?" X, _ contempt)
42.	What did I love?	love ( & did) ( _ I, "what?" X)
43.	Who loved maths?	loved ( "who?" X, _ math)
44.	When did I love maths?	love ( & did) ( _ I, _ math, "when?" X)
45.	What did I love when I was in school?	_ when (loved ( & did) ( _ I, "what?" X), was ( _ I, _ school ( _ at)))
46.	When did I have great time?	have ( & did) ( _ I, _ time ( _ great ( _ a)), "when?" X)
47.	Who had a great time?	had ("who?" X, _ time ( _ great ( _ a)))
48.	Who went to the coast?	went ( "who?" X, _ coast ( _ the ( _ to)))
49.	Where did I go?	go ( & did) ( _ I, "where?" X)
50.	What do I wonder?	wonder ( & do) ( _ I, "what?" X)
51.	Who noticed?	noticed ("who?" X)
52.	Who can go?	can (go ( "who?" X))
53.	Where can you go?	can (go ( _ you, "where?" X))
54.	When can you go here?	can (go ( _ you, _ here), "when?" X))
55.	Who get the point?	get ("who?" X, _ point ( _ the))
56.	What can you do if you get the point?	_ if (do (& can) ( _ you, "what?" X), get ( _ you, _ point ( _ the)))
57.	Who will feel cold?	feel ( & will) ( "who?" X, _ cold )
58.	When will you feel cold?	feel ( & will) ( _ you, _ cold, "when?" X)
59.	What will you feel if you don't wear a coat?	_ if (feel (& will) ( _ you, "what?" X), wear (& don't) ( _ you, _ coat ( _ a)))
60.	Who is glad?	is ("who?" X, _ glad)
61.	Who is glad that workers aren't losing their jobs?	_ that (is ("who?" X, _ glad), aren't (losing ( _ workers, _ jobs ( _ their ))))
62.	Why is Monika glad?	is ( _ Monika, _ glad, "why?" X)
63.	Who aren't losing their jobs?	aren't (losing ( "who?" X, _ jobs ( _ their))
64.	What was still painful?	was ( "what?" X, _ still, _ painful)
65.	Why do I went to see a specialist?	went ( & do) ( _ I, _ see ( _ to), _ specialist ( _ a), "why?" X)
66.	Who went to see a specialist?	went ( "who?" X, _ see ( _ to), _ specialist ( _ a))
67.	Who can ask for help?	ask ( & can) ("who?" X, _ help ( _ for))
68.	When can you ask for help?	can (ask ( _ you, _ help ( _ for), "when?" X)
69.	What can you whenever you need it?	_ whenever (can ( _ you, "what?" X), need ( _ you, _ it))
70.	Who minded the dog?	minded ("who?" X, _ dog ( _ the))
71.	When does he minded the dog?	minded ( & does) ( _ he, _ dog ( _ the), "when?" X)
72.	Who minded the dog while I was away?	_ while (minded ("who?" X, _ dog ( _ the)), was ( _ I, _ away))

## Privitak 7. Testirane rečenice engleskog jezika i pripadajući FNOK-ovi

R.B.	TENG	FNOK
1.	Tom and Paul talk about the solution.	talk ("who?" and ("who?" Tom, "who?" Paul), "what?" solution ("art?" the, "what?" about))
2.	Ana and John play the violin.	play ("who?" and ("who?" Ana, "who?" John), "what?" violin ("art?" the))
3.	Martina and Josip play the violin extremely well.	play ("who?" and ("who?" Martina, "who?" Josip), "what?" violin ("art?" the), "how?" well ("how?" extremely))
4.	Sanja is training volleyball and tennis.	is (training ("who?" Sanja), "what?" and ("what?" volleyball, "what?" tennis))
5.	Peter trains basketball, volleyball and tennis.	trains ("who?" Peter, "what?" and ("what?" basketball, "what?" volleyball, "what?" tennis))
6.	Sanja and Franjo are programmers.	are ("who?" and ("who?" Sanja, "who?" Franjo), "what?" programmers)
7.	Marko and Ana are from Zagreb.	are ("who?" and ("who?" Marko, "who?" Ana), "where?" from, "what?" Zagreb)
8.	We were wet and tired.	were ("who?" we, "what?" and ("what?" wet, "what?" tired))
9.	We kissed and hugged each other.	"what?" and (kissed ("who?" we), hugged ("who?" other ("who?" each)))
10.	Take your clothes off and get in the bath.	"what?" and (take ("what?" clothes ("whose?" your)), get ("where?" bath ("where?" in ("art?" the))))
11.	He drew a heart and coloured it red.	"what?" and (drew ("who?" he, ("what?" heart ("art?" a), coloured ("what?" it, "how?" red))))
12.	A young girl came up to me and asked for money.	"what?" and (came ("who?" girl ("which?" young ("art?" a)), "where?" up ("where?" me ("where?" to))), asked ("what?" money ("what?" for)))
13.	Monika has become a very patient and attentive dentist.	has (become ("who?" Monika, "what?" dentist ("art?" a, "what?" and ("what?" patient ("how?" very), "what?" attentive))))
14.	It costs a lot to feed and clothe five children.	costs ("what?" it, "how_much?" lot ("art?" a), "what?" to, "what?" and ("what?" feed, "what?" clothe), "who?" children ("how_many?" five))
15.	Three and two are five.	are ("how_much?" and ("how_much?" three, "how_much?" two), "how_much?" five)
16.	I made dinner and watched the television	"what?" and (made ("who?" I, "what?" dinner), watched ("what?" television ("art?" the)))
17.	I got dressed and had my breakfast.	"what?" and (got (dressed ("who?" I)), had ("what?" breakfast ("whose?" my)))
18.	The water was lovely and cold.	was ("what?" water ("art?" the), "what?" and ("what?" lovely, "what?" cold))
19.	You can come and see me tomorrow.	"what?" and (come ( can ("who?" you), see ("who?" me, "when?" tomorrow)))
20.	Snakes and crocodiles are dangerous reptiles.	are (what?" and ("what?" snakes, "what?" crocodiles), "what?" reptiles ("what?" dangerous))
21.	Turtles and lizards are reptiles.	are ("what?" and ("what?" turtles, "what?" lizards), "what?" reptiles)
22.	I opened the door and looked around.	"what?" and (opened ("who?" I, ("what?" door ("art?" the)), looked ("where?" around)))

23.	Tom is happy and healthy boy.	is ("who?" Tom, "what?" boy ("what?" and ("what?" happy, "what?" healthy)))
24.	Lucija is from Zagreb or Rijeka.	is ("who?" Lucija, "where?" or ("where?" Zagreb, "where?" Rijeka) ("where?" from))
25.	Ana will come on Tuesday or Wednesday.	will (come ("who?" Ana), "when?" or ("when?" Tuesday, "when?" Wednesday) ("when?" on))
27.	Monika or Sanja will receive the main prize.	will (receive ("who?" or ("who?" Monika, "who?" Sanja), "what?" prize ("what?" main ("art?" the))))
28.	We don't tell them it's good or bad.	don't (tell ("who?" we, "whom?" them, is ("what?" it, "what?" or ("what? Good, "what?", bad))))
29.	Hurry up, or you will miss it all.	"what?" or (hurry ("what?" up), will (miss ("who?" you ("what?" it ("what?" all))))
30.	I'd better tell him myself or I'll get in even more trouble.	"what?" or (would (tell ("how?" better ("who?" I, "whom?" him ("how?" myself)), will (get ("who?" I, "what?" trouble ("what?" more ("what?" even ("what?" in))))))
31.	I either take taxis or walk everywhere.	"what?" or (take ("who?" I, "how?" either, "what?" taxis), walk ("where?" everywhere))
32.	She never learned to read or write.	learned ("who?" she, "when?" never, "what?" or ("what?" read, "what?" write))
33.	You should eat more, or you'll make yourself ill.	"what?" or (should (eat ("who?" you, "what?" more)), will (make ("who?" you, "whom?" yourself, "what?" ill )))
34.	Marko is happy because he love his job.	"why?" because (is ("who?" Marko, "what?" happy), love ("who?" he, "what?" job ("whose?" his)))
35.	Sanja is happy because she love her interesting hobby.	"why?" because (is ("who?" Sanja, "what?" happy), loves ("who?" she, "what?" hobby ("whose?" her, "what?" interesting)))
36.	The journey was quite quick because the road was clear.	"why?" because (was ("what?" journey ("art?" the), "how?" quick ("how?" quite)), was ("what?" road ("art?" the), "what?" clear))
37.	Patients were discharged from hospital because the beds were needed by other people.	"why?" because (were (discharged ("who?" patients), "where?" hospital ("where?" from)), were (needed ("what?" beds ("art?" the)), "who?" people ("which?" other ("which?" by))))
38.	I almost missed my flight because there was a long queue in the duty-free shop.	"why?" because (missed ("who?" I, "how?" almost, "what?" flight ("whose?" my)), was ("where?" there, "where?" shop ("what?" duty-free ("art?" the ("where?" in))), "what?" queue ("what?" long ("art?" a))))
39.	I didn't go out because I had no money.	"why?" because (didn't (go ("who?" I, "where?" out)), had ("who?" I, "what?" money ("how_much?" no)))
40.	She was delighted because it was such a surprise.	"why?" because (was ("who?" she, "what?" delighted), was ("what?" it, "what?" surprise ("art?" a ("what?" such))))
41.	Vanessa is sleepy because she has been studying all night.	"why?" because (is ("who?" Vanessa, "what?" sleepy), has (been (studying ("who?" she, "when?" night ("what?" all))))
42.	Daniel loves summer because he loves swimming.	"why?" because (loves ("who?" Daniel, "what?" summer), loves ("who?" he, "what?" swimming))
43.	We were worried because of the storm.	"why?" because (were ("who?" we, "what?" worried), "what?" storm ("art?" the ("how?" of)))
44.	We did it because we felt it our duty.	"why?" because (did ("who?" we, "what?" it), felt ("who?" we, "what?" duty ("whose?" our ("what?" it))))

45.	The trip was cancelled because of bad weather.	"why?" because (was ("what?" trip ("art?" the), "who?" cancelled), "what?" weather ("how?" bad ("how?" of)))
46.	There was nothing they could do but swallow their pride.	"what?" but (was (could (do ("where?" there, "what?" nothing, "who?" they))), swallow ("what?" pride ("whose?" their)))
47.	There was nothing to do but wait.	"what?" but (was (do ("where?" there, "where?" to ("what?" nothing))), wait)
48.	This is not caused by evil, but by simple ignorance.	"what?" but (is (caused ("what?" this, "what?" not, "what?" evil ("what?" by))), "what?" ignorance ("what?" simple ("what?" by)))
49.	We ran after him, but he escaped.	"what?" but (ran ("who?" we, "how?" after, "who?" him ), escaped ("who?" he))
50.	It's a nice dress, but it creases very easily.	"what?" but (is ("what?" it, "what?" dress ("what?" nice ("art?" a))), creases ("what?" it, "how?" easily ("how?" very)))
51.	I've received replies from everybody but Jane.	have (received ("who?" I, "what?" replies, "who?" everybody ("who?" from), who?" but ("whom?" Jane)))
52.	He felt nothing but contempt for her.	felt ("who?" he, "what?" nothing, "what?" but ("what?" contempt, "whom?" her ("whom?" for)))
53.	They had no alternative but to follow.	had ("who?" they, "what?" no, "what?" alternative, "what?" but ("what?" follow ("what?" to)))
54.	She's worried but hopefully she'll be fine.	"what?" but (is ("who?" she, "what?" worried), will (be ("who?" she, "what?" hopefully, "what?" fine)))
55.	I loved math when I went to school.	"when?" when (loved ("who?" I, "what?" math), was ("who?" I, "where?" school ("where?" to)))
56.	When I was young, there were no houses here.	"when?" when (was ("who?" I, "what?" young, were ("where?" there, "what?" houses ("what?" no), ("where?" here)))
57.	When the new park opens, I will go there every day.	"when?" when (opens ("what?" park ("what?" new ("art?" the))), will (go ( "who?" I, "where?" there, "when?" day ("what?" every))))
58.	I had a great time when I went to the coast.	"when?" when (had ("who?" I, "what?" time ("what?" great ("art?" a))), went ("who?" I, "where?" coast ("art?" the ("where?" to)))
59.	Saturday is the day when I get my hair done.	"when?" when (is ("what?" saturday, "what?" day ("art?" the)), get (done ("who?" I, "what?" hair ("whose?" my))))
60.	You have to know when to look for help.	"when?" when (have (know ("who?" you, "what?" to)), look ("where?" to, "what?" help ("what?" for)))
61.	I hate it when there's nobody in the office.	"when?" when (hate ("who?" I, "what?" it), is ("where?" there, "who?" nobody, "where?" office ("art?" the ("where?" in))))
62.	I was just getting into the bath when the phone rang.	"when?" when (was (getting ("who?" I, "when?" just, "where?" bath ("art?" the ("where?" into))), rang ("what?" phone ("art?" the)))
63.	I went there when I was a child.	"when?" when (went ("who?" I, "where?" there), was ("who?" I, "what?" child ("art?" a)))
64.	He was shocked when I told him.	"when?" when (was ("who?" he, "what?" shocked), told ("who?" I, "whom?" him))

65.	She was only 20 when she had her first baby.	"when?" when (was ("who?" she, "how_many?" 20 ("what?" only)), had ("who?" she, "what?" baby ("what?" first ("whose?" her))))
66.	When I start college, I'll miss my old school friends.	"what?" when (start ("who?" I, "what?" college), will (miss ("who?" I, "whom?" friends ("what?" school ("what?" old ("whose?" my))))))
67.	If you have a complaint, write to the director.	"what?" if (have ("who?" you, "what?" complaint ("art?" a)), write ("who?" director ("art?" the ("whom?" to))))
68.	Let me know now if you are not sure about this.	"what?" if (let (know ("who?" me ("when?" now))), are ("who?" you, "what?" sure ("what?" not), "what?" about ("what?" this)))
69.	You can go only if you're 18 or over.	"what?" if (can (go ("who?" you, "how?" only)), are ("who?" you, "how_many?" or ("how_many?" 18, "how_many?" over)))
70.	If I could just use the phone, I'll get a taxi.	"what?" if (could (use ("who?" I, "what?" just, "what?" phone ("art?" the))), will (get ("who?" I, "what?" taxi ("art?" a))))
71.	I wonder if she noticed.	"what?" if (wonder ("who?" I), noticed ("who?" she))
72.	I wonder if it can get any worse.	"what?" if (wonder ("who?" I), can (get ("what?" it, "what?" worse ("how?" any))))
73.	She was honest, if a little brutal.	"what?" if (was ("who?" she, "what?" honest), "what?" brutal ("what?" little ("art?" a)))
74.	He asked if we would like some coffee.	"what?" if (asked ("who?" he), would (like ("who?" we, "what?" coffee ("what?" some))))
75.	You can go here if you get the point.	"when?" if (can (go ("who?" you, "where?" here)), get ("who?" you, "what?" point ("art?" the)))
76.	I get really tired if I stay up late.	"what?" if (get ("who?" I, "what?" tired ("how?" really)), stay ("who?" I, "what?" up ("what?" late)))
77.	I'll pay you double if you get the work finished by Friday.	"when?" if (will (pay ("who?" I, "how?" double, "whom?" you)), get ("who?" you, "what?" work ("art?" the), "what?" finished, "when?" Friday ("when?" by)))
78.	If she hadn't called, I wouldn't have known.	"what?" if (hadn't (called ("who?" she)), wouldn't (have (know ("who?" I))))
79.	If I don't get enough sleep I get a headache.	"what?" if (don't (get ("who?" I, "what?" sleep ("how_much?" enough))), get ("who?" I, "what?" headache ("art?" a)))
80.	I'm very sorry if I've offended you.	"what?" if (am ("who?" I, "what?" sorry ("how?" very)), have (offended ("who?" I, "whom?" you)))
81.	You will feel cold if you don't wear a coat.	"what?" if (feel ((& will) ("who?" you, "what?" cold)), wear ((& don't) ("who?" you, "what?" coat ("art?" a))))
82.	Biscuits will stay crisp if you keep them in an airtight container.	"when?" if (will (stay ("what?" biscuits, "what?" crisp)), keep ("who?" you, "whom?" them, "where?" container ("what?" airtight? ("art?" an ("where?" in))))))
83.	You will stay hungry if you don't eat wright away.	"what?" if (will (stay ("who?" you, "what?" hungry)), don't (eat ("who?" you, "when?" away ("when?" wright))))
84.	You have to go to college for a lot of years if you want to be a doctor.	"why?" if (have (go ("who?" you, "where?" college ("where?" to), "what?" years ("what?" of ("how_much?" lot ("art?" a))))), want (be ("who?" you, "what?" to, "who?" doctor ("art?" a))))

85.	We have so many questions and yet so few answers.	have ("who?" we, "what?" and ("what?" questions ("how_much?" many ("how_much?" so)), answers ("how_much?" few ("how_much?" so),"when?" yet)))
86.	You don't know what you can achieve until you try.	"what?" what (don't (know ("who?" you)), can (achive ("who?" you, "what?" try, "when?" until)))
87.	Monika is glad that workers aren't losing their jobs.	"what?" that (is ("who?" Monika, "what?" glad), aren't (losing ("who?" workers, "what?" jobs ("whose?" their )))
88.	It's inevitable that it will happen.	"what?" that (is ("what?" it, "what?" inevitable), will (happen ("what?" it)))
89.	It was still painful, so I went to see a specialist.	"what?" so (was ("what?" it, "what?" still, "how?" painful), went ("who?" I, "what?" see("what?" to), "who?" specialist ("art?" a)))
90.	She does not read articles about her, unless they are sent to her by her publishers.	"when?" unless ( does (not (read ("who?" she, "what?" articles ("what?" about ("whom?" her))))), (are (sent ("what?" they, "whom?" her ("whom?" to), ""who?" publishers ("whose?" her ("whose?" by))))))
91.	I wouldn't wear those shoes unless I was trying to break my ankle	"when?" unless (wouldn't (wear ("who?" I, "what?" shoes ("what?"? Those))), was (trying (break ("who?" I, "what?" to, "what?" ankle ("whose?" my))))
92.	You can ask for help whenever you need it.	"when?" whenever (can (ask ("who?" you, "what?" help ("what?" for))), need ("who?" you, "what?" it))
93.	The springs in the armchair creak whenever I change position.	"when?" whenever (creak ("what?" springs ("art?" the), "where?" armchair ("art?" the ("where?" in))), change ("who?" I, "what?" position))
94.	Spain is increasing the size of its fishing fleet while ours is contracting.	"when?" while (is (increasing ("who?" Spain, "what?" size ("art?" the), "what?" fleet ("what?" fishing ("whose?" its))), is (contracting ("whose?" ours)))
95.	He minded the dog while I was away.	"when?" while (minded ("who?" he, "what?" dog ("art?" the)), was ("who?" I, "where?" away))
96.	You can't be on the phone while driving.	"when?" while (can't (be ("who?" you), "what?" phone ("art?" the ("what?" on))), driving)
97.	Nothing much changed while he was away.	"when?" while (changed ("what?" nothing, "howmuch?" much), was ("who?" he, "where?" away))
98.	Lucija and Ana discuss about the weather.	discuss ("who?" and ("who?" Lucija, "who?" Ana), "what?" weather ("art?" the, "what?" about))
99.	She was glad when the meeting was over.	"when?" when (was ("who?" she, "what?" glad), was ("what?" meeting ("art"" the), "what?" over))
100.	Marko practise service every morning and every evening.	practise ("who?" Marko, "what?" service. "when?" and ("when?" morning ("what?" every), "when?" evening ("what?" every)))

## Privitak 8. Testirana pitanja engleskog jezika i pripadajući QFNOK-ovi

R.B.	QTENG	QFNOK
1.	What are Tom and Paul talking about?	talking ( & are) ( _ and ( _ Paul, _ Tom), "what?" X ( _ about))
2.	Who talks about the solutions?	talks ("who?" X, _ solution ( _ about, _ the))
3.	Who discuss about the solutions?	discuss ("who?" X, _ solution ( _ about, _ the))
4.	Who talks to Tom?	talks ( "who?" X, _ Tom ( _ to))
5.	Who talks to Paul?	talks ( "who?" X, _ Paul ( _ to))
6.	Who is talking?	talking ( & is) ( "who?" X)
7.	Who is playing?	playing ( & is) ( "who?" X)
8.	Who plays the violin?	plays ("who?" X, _ violin ( _ the))
9.	What are Ana and John playing?	playing ( & are) ( _ and ( _ Ana, _ John), "what?" X)
10.	Who is Ana playing violin with?	playing ( & is) ("who?" X ( _ with), _ Ana, _ violin)
11.	Who does John play the violin with?	play ( & does) ("who?" X ( _ with), _ John, _ violin ( _ the))
12.	Who plays the violin extremely well?	plays ("who?" X, _ violin ( _ the), _ well ( _ extremely))
13.	What does Sanja train?	train ( & does) ( _ Sanja, "what?" X)
14.	What does Sanja practise?	practise ( & does) ( _ Sanja, "what?" X)
15.	Who trains volleyball?	trains ( "who?" X, _ volleyball)
16.	Who trains tennis?	trains ( "who?" X, _ tennis)
17.	Who trains volleyball and tennis?	trains ( "who?" X, _ and ( _ volleyball, _ tennis))
18.	What does Peter train?	train ( & does) ( _ Peter, "what?" X)
19.	Who trains basketball?	trains ("who?" X, _ basketball)
20.	Who trains basketball, volleyball or tennis?	trains ( "who?" X, _ or ( _ basketball , _ volleyball, _ tennis))
21.	Who are the programmers?	are ("who?" X, _ programmers ( _ the))
22.	What are Sanja and Franjo?	are ( _ and ( _ Sanja, _ Franjo), "what?" X)
23.	What is Sanja?	is ( _ Sanja, "what?" X)
24.	What is Franjo?	is ( _ Franjo, "what?" X)
25.	Who are from Zagreb?	are ("who?" X, _ from ( _ Zagreb))
26.	Where are Ana and Marko from?	are ( _ and ( _ Ana, _ Marko), "where?" X ( _ from))
27.	Where is Marko from?	is ( _ Marko, "where?" X ( _ from))

28.	Where is Ana from?	is ( _ Ana, "where?" X ( _ from))
29.	What were we?	were ( _ we, "what?" X )
30.	Who is wet?	is ("who?" X, _ wet)
31.	Who is tired?	is ("who?" X, _ tired)
32.	Who is wet and tired?	is ("who?" X, _and ( _ wet, _ tired))
33.	What did we do to each other?	what? X (do ( & did) ( _ we, _ other ( _ each)))
34.	Who did we kiss and hug?	_ and (kiss ( _ we), hugg ("who?" X))
35.	What did he draw and colour?	_ and (& did) (draw ( _ he, "what?" X), colour)
36.	What did he draw?	draw ( & did) ( _ he, "what?" X)
37.	What did he paint?	paint ( & did) ( _ he, "what?" X)
38.	What did he colour?	colour ( & did) ( _ he, "what?" X)
39.	How did he colour the heart?	colour ( & did) ( ( _ he ( _ heart ( _ the), "how?" X)
40.	Who drew a heart?	drew ("who?" X, _ heart ( _ a))
41.	Who came up to me?	came ("who?" X, _ up ( _ me ( _ to)))
42.	What did the young girl ask?	ask (& did) ( _ girl ( _ young ( _ a)), "what?" X)
43.	Which girl came?	came ( _ girl ("which?" X))
44.	Who came up to me and asked for money?	_ and (came ("who?" X, _ up ( _ me ( _ to))), asked ( _ money ( _ for)))
45.	Who asked me for money?	asked ("who?" X, _ me, _ money ( _ for))
46.	Who became a dentist?	became ("who?" X, _ dentist ( _ a))
47.	Who became a patient dentist?	became ("who?" X, _ dentist ( _ a, _ patient))
48.	Who became a attentive dentist?	became ("who?" X, _ dentist ( _ a, _ attentive))
49.	Who became a very patient and attentive dentist?	became ("who?" X, _ dentist ( _ a, _ and ( _ patient ( _ very), _ attentive)))
50.	What dentist did Monika become?	become ( & did) ( _ Monika, _ dentist ("what?" X))
51.	What did Monika become?	become ( & did) ( _ Monika, "what?" X)
52.	How many is three and two?	is ( _ and ( _ three, _ two), "how_many?" X )
53.	How much is five?	is ("how_many?" X, _ five)
54.	What did I made?	made ( & did) ( _ I, "what?" X)
55.	What was I watched?	watched ( & was) ( _ I, "what?" X)
56.	Who made dinner?	made ("who?" X, _ dinner)
57.	Who watched television?	watched ("who?" X, _ television)
58.	Who made dinner and watched television?	_ and (made ("who?" X, _ dinner), watched ( _ television))
59.	Who got dressed?	got (dressed ("who?" X))
60.	Who had breakfast?	had ("who?" X, _ breakfast)



61.	What did I get?	get (& did) ( _ I, "what?" X)
62.	What was the water?	was ( _ water ( _ the), "what?" X)
63.	What was lovely and cold?	was ("what?" X, _ and ( _ lovely, _ cold))
64.	What was lovely?	was ("what?" X, _ lovely)
65.	What was cold?	was ("what?" X, _ cold)
66.	Who can come and see me tomorrow?	_ and (can (come ("who?" X)), see ( _ me, _ tomorrow))
67.	Who can come?	(can ( come ("who?" X))
68.	Who can see me?	see (& can) ("who?" X, _ me)
69.	Who can see me tomorrow?	see (& can) ("who?" X, _ me, _ tomorrow)
70.	What are snakes and crocodiles?	are ( _ and ( _ snakes, _ crocodiles), what?" X)
71.	What are reptiles?	are ("what?" X, _ reptiles)
72.	What are dangerous reptiles?	are ("what?" X, _ reptiles ( _ dangerous))
73.	What are turtles and lizards?	are ( _ and ( _ turtles, _ lizards), "what?" X)
74.	What did I open?	open ( & did) ( _ I, "what?" X)
75.	Who opened the door?	opened ("who?" X, _ door ( _ the))
76.	Where I looked?	looked ( _ I, "where?" X)
77.	Who looked around?	looked ("who?" X, _ around)
78.	Who is happy and healthy boy?	is ("who?" X, _ boy ( _ and ( _ happy, _ healthy)))
79.	Who is happy boy?	is ("who?" X, _ boy ( _ happy))
80.	Who is healthy boy?	is ("who?" X, _ boy ( _ healthy))
81.	What is Tom?	is ( _ Tom, "what?" X)
82.	Where is Lucija from?	is ( _ Lucija , "where?" X ( _ from))
83.	Who is from Zagreb?	is ("who?" X, _ from ( _ Zagreb))
84.	Who is from Rijeka?	is ("who?" X, _ from ( _ Rijeka))
85.	Who is from Zagreb or Rijeka?	is ("who?" X, _ from ( _ or ( _ Zagreb, _ Rijeka)))
86.	Who is coming on Tuesday or Wednesday?	coming ( & is) ("who?" X, _ on, _ or ( _ Tuesday, _ Wednesday))
87.	When will Ana come?	will (come ( _ Ana, "when?" X))
88.	Who will receive the main prize?	will (receive ("who?" X, _ prize ( _ main ( _ the))))
89.	What will Monika or Sanja receive?	will (receive ( _ or ( _ Monika, _ Sanja), "what?" X))
90.	What is good or bad?	is ("what?" X, _ or ( _ good, _ bad))
91.	Who will miss it all?	will (miss ("who?" X, _ all ( _ it)))
92.	What you will miss?	will (miss ( _ you, "what?" X)
93.	What would better?	would ("what?" X, _ better)
94.	Who will get in even more trouble?	get ( & will) ("who?" X, _ trouble ( _ more ( _ even ( _ in))))

95.	Who either take taxis or walk everywhere?	_ or (take ( _ either, "who?" X, _ taxis)), walk ( _ everywhere))
96.	Who take taxis?	take ("who?" X, _ taxis)
97.	Who walk everywhere?	walk ("who?" X, _ everywhere)
98.	What I either take?	take ( _ I, _ either, "what?" X)
99.	What she never learned?	learned ( _ she, _ never, "what?" X)
100.	Who never learned to read or write?	learned ("who?" X, _ never, _ to, _ or ( _ read, _ write))
101.	Who should eat more?	eat ( & should) ("who?" X, _ more)
102.	Who will make yourself ill?	make ( & will) ( "who?" X, _ yourself, _ ill)
103.	Why you should eat more?	why?" X (should (eat ( _ you, _ more)))
104.	Who love his job?	love ("who?" X, _ job ( _ his))
105.	What is Marko?	is ( _ Marko, "what?" X)
106.	Why is Marko happy?	is ( _ Marko, _ happy, "why?" X)
107.	Who loves her hobby?	loves ("who?" X, _ hobby ( _ her))
108.	Who is happy?	is ("who?" X, _ happy)
109.	Why is Sanja happy?	is ( _ Sanja, _ happy, "why?" X)
110.	What was quite quick?	was ("what?" X, _ quick ( _ quite))
111.	What was the journey?	was ( _ journey ( _ the), "what?" X)
112.	Why was the journey quite quick?	was ( _ journey ( _ the), _ quick ( _ quite), "why?" X)
113.	What was clear?	was ("what?" X, _ clear)
114.	What was the road?	was ( _ road ( _ the), "what?" X)
115.	Who is discharged from hospital?	discharged ( & is) ("who?" X, _ hospital ( _ from))
116.	Who needed the beds?	needed ("who?" X, _ beds ( _ the ))
117.	What was needed by other people?	needed ( & was) ("what?" X, _ people ( _ other))
118.	Why were patients discharged from hospital?	discharged were (discharged ( _ patients, _ hospital ( _ from)), "why?" X)
119.	What I almost missed?	missed ( _ I, _ almost, "what?" X)
120.	Who almost missed the flight?	missed ( "who?" X, _ almost, _ flight ( _ the))
121.	Where was a long queue?	was ( _ queue ( _ a, _ long), "where?" X)
122.	What was in the duty-free shop?	was ("what?" X, _ shop ( _ duty-free ( _ the)))
123.	Why I almost missed my flight?	missed ( _ I, _ almost, _ flight ( _ my), "why?" X)
124.	Who didn't go out?	didn't (go ("who?" X, _ out))
125.	Where I didn't go?	didn't ( go ( _ I, "where?" X))
126.	Who had no many?	had ("who?" X, _ money ( _ no))
127.	What I dont had?	had ( & dont) ( _ I, "what?" X)
128.	Who didn't go out?	go (& didn't) ("who?" X, _ out)

129.	Why didn't I go out?	didn't (go ( _ I, _ out), "why?" X)
130.	Why was she delighted?	was ( _ she, _ delighted, "why?" X)
131.	Who is sleepy?	is ("who?" X, _ sleepy)
132.	What is Vanessa?	is ( _ Vanessa, "what?" X)
133.	Who has been studying all night?	studying ( & has, & been) ("who?" X, _ night ( _ all))
134.	Why is Vanessa sleepy?	is ( _ sleepy, _ Vanessa, "why?" X)
135.	Who loves summer?	loves ("who?" X, _ summer)
136.	Who loves swimming?	loves ("who?" X, _ swimming)
137.	What does Daniel love?	love ( & does) ( _ Daniel, "what?" X)
138.	Why Daniel loves summer?	loves ( _ Daniel, _ summer, "why?" X)
139.	Who was worried?	was ("who?" X, _ worried)
140.	What we were?	were ( _ we, "what?" X)
141.	Why we were worried?	were ( _ we, _ worried, "why?" X)
142.	What we did?	did ( _ we, "what?" X)
143.	Why we did it?	did ( _ we, _ it, "why?" X)
144.	What we felt?	felt ( _ we, "what?" X)
145.	Who felt it was their duty?	felt ( & is) ("who?" X, _ it, _ duty ( _ their))
146.	What was cancelled?	was ("what?" X, _ canceled)
147.	What was the trip?	was ("what?" X, _ trip ( _ the))
148.	Why the trip was cancelled?	was ( _ trip ( _ the), _ canceled, "why?" X)
149.	What they could do?	do ( & could) ( _ they, "what?" X)
150.	What they swallowed?	swallowed ( _ they, "what?" X)
151.	Who swallows their pride?	swallows ("who?" X, _ pride ( _ their))
152.	Where was nothing they could do?	was ( could (do ( _ they, "where?" X, _ nothing)))
153.	What is not caused by evil?	is ( not (caused ("what?" X, _ evil ( _ by))))
154.	What is not caused this?	is (not (caused ( _ this, "what?" X)))
155.	What is caused by simple ignorance?	caused ( & is) ("what?" X, _ ignorance ( _ simple ( _ by)))
156.	Who is running after him?	running ( & is) ( "who?" X, _ after, _ him )
157.	Who are we running after?	running ( & are) ( _ we, _ after, "who?" X)
158.	Who escaped?	escaped ("who?" X)
159.	What is nice?	is ("what?" X ( _ nice))
160.	What creases very easily?	creases ("what?" X, _ easily ( _ very))
161.	Who has received replies?	received ( & has) ("who?" X, _ replies)
162.	Who received replies from everybody?	received ("who?" X, _ replies, _ everybody ( _ from))
163.	Who I received replies from?	received ( _ I, _ replies, "who?" X ( _ from))
164.	Who did I not received replies from?	received ( & did, & not) ( _ I, _ replies, "who?" X ( _ from))
165.	What did he felt for her?	felt ( & did) ( _ he, "what?" X, _ her ( _ for))

166.	Who felt nothing?	felt ("who?" X, _ nothing)
167.	Who felt contempt?	felt ("who?" X, _ contempt)
168.	What they did not have?	have ( & did) ( _ they, _ not, "what?" X)
169.	Who had no alternative?	had ("who?" X, _ no, _ alternative)
170.	Who had to follow?	had ("who?" X, _ follow ( _ to))
171.	Who is worried?	is ("who?" X, _ worried)
172.	Who will be hopefully fine?	will ( be ("who?" X, _ hopefully, _ fine))
173.	What is she?	is ( _ she, "what?" X)
174.	What did I love?	love ( & did) ( _ I, "what?" X)
175.	Who loved math?	loved ( "who?" X, _ math)
176.	When did I love math?	love ( & did) ( _ I, _ math, "when?" X)
177.	What did I love when I was at school?	_ when (loved ( & did) ( _ I, "what?" X), was ( _ I, _ school ( _ at)))
178.	Who was young?	was ("who?" X, _ young)
179.	When were there no houses here?	were ( _ there, _ houses ( _ no), _ here, "when?" X)
180.	Who will go there?	go ( & will) ("who?" X, _ there)
181.	Who will go there every day?	go ( & will) ("who?" X, _ there, _ day ( _ every))
182.	Where I will go?	go ( & will) ( _ I, "where?" X)
183.	When I go there?	go ( & will) ( _ I, "when?" X)
184.	When I go to the new park?	go ( _ I, _ to, _ park ( _ new ( _ the)), "when?" X)
185.	What will open?	open ( & will) ("what?" X)
186.	When did I have a great time?	have ( & did) ( _ I, _ time ( _ great ( _ a)), "when?" X)
187.	Who had a great time?	had ("who?" X, _ time ( _ great ( _ a)))
188.	Who went to the coast?	went ( "who?" X, _ coast ( _ the ( _ to)))
189.	Where did I go?	go ( & did) ( _ I, "where?" X)
190.	What is Saturday?	is ( _ Saturday, "what?" X)
191.	When I get my hair done?	get (done ( _ I, _ hair ( _ my)), "when?" X)
192.	When I hate it?	hate ( _ I, _ it, "when?" X)
193.	Where I was just getting into?	was (getting ( _ I, _ just, "where?" X ( _ into)))
194.	When the phone rang?	rang ( _ phone ( _ the), "when?" X)
195.	What rang?	rang ("what?" X)
196.	Who was just getting into the bath?	was (getting ("who?" X, _ just, _ bath ( _ the ( _ into))))
197.	Where did I go when I was a child?	_ when (go ( & did) ( _ I, "where?" X), was ( _ I, _ child ( _ a)))
198.	When he was shocked?	was ( _ he, _ shocked, "when?" X)
199.	Who was shocked?	was ("who?" X, _ shocked)
200.	Who told him?	told ("who?" X, _ him)
201.	Who had her first baby?	had ("who?" X, _ baby ( _ first ( _ her)))
202.	When she had her first baby?	had ( _ she, _ baby ( _ first ( _ her)), "when?" X)

203.	Whom I will miss when I start college?	_ when (start ( _ I, _ college), will (miss ( _ I, "whom?" X))))
204.	When I will miss my old school friends?	will (miss ( _ I, _ friends ( _ school ( _ old ( _ my)))),"when?" X))
205.	Who have a complaint?	have ("who?" X, _ complaint)
206.	When you have to write to director?	write ( & have) ( _ you, _ director ( _ the ( _ to)), "when?" X)
207.	Who let me know?	let (know ("who?" X, _ me))
208.	What you are not sure about?	are ( _ you, _ sure ( _ not), _ about ("what?" X))
209.	When you let me know?	let (know ( _ me ("when?" X)))
210.	Who can go?	can (go ( "who?" X))
211.	What can if you are 18 or over?	_ if (can ("what?" X, are ( _ you, _ or ( _ 18, _ over)))
212.	What I could just use?	use ( & could) ( _ I, _ just, "what?" X)
213.	What I will get if I could just use the phone?	_ if (could (use ( _ I, _ just, _ phone ( _ the))), will (get ( _ I, "what?" X)))
214.	Who will get a taxi?	get ( & will) ("who?" X, _ taxi ( _ a))
215.	What do I wonder?	wonder ( & do) ( _ I, "what?" X)
216.	Who noticed?	noticed ("who?" X)
217.	What can get any worse?	can (get ("what?" X, _ worse ( _ any)))
218.	What she was?	was ( _ she, "what?" X)
219.	Who was honest?	was ("who?" X, _ honest)
220.	Who was little brutal?	was ("who?" X, _ brutal ( _ little))
221.	Who asked?	asked ("who?" X)
222.	What he asked?	asked ( _ he, "what?" X)
223.	Who asked if we would like some coffee?	_ if (asked ("who?" X), would (like ( _ we, _ coffee ( _ some))))
224.	What we would like?	would (like ( _ we, "what?" X))
225.	Who can go?	can (go ( "who?" X))
226.	Where can you go?	can (go ( _ you, "where?" X))
227.	When can you go here?	can (go ( _ you, _ here), "when?" X))
228.	Who get the point?	get ("who?" X, _ point ( _ the))
229.	What can you do if you get the point?	_ if (do (& can) ( _ you, "what?" X), get ( _ you, _ point ( _ the)))
230.	Who get really tired?	get ("who?" X, _ tired ( _ really))
231.	When I get really tired?	"when?" X (get ( _ I, _ tired ( _ really)))
232.	What if I stay up late?	_ if ("what?" X, stay ( _ I, _ up ( _ late)))
233.	Who will pay double?	will (pay ("who?" X, _ double))
234.	When will I pay you double?	will (pay ( _ I, _ double, _ you), "when?" X)
235.	How I will pay you if you get the work finished by Friday?	_ if (will (pay ( _ I, "how?" X, _ you)), get ( _ you, _ work ( _ the), _ finished, _ Friday ( _ by)))

236.	Who had called?	called ( & had) ("who?" X)
237.	When I wouldn't have known?	wouldn't (have (know ( _ I)), "when?" X)
238.	When I get a headache?	get ( _ I, _ headache ( _ a), "when?" X)
239.	What I get if I don't get enough sleep?	_ if (don't ( _ I, _ sleep ( _ enough)), get ( _ I, "what?" X))
240.	Who is very sorry?	is ("who?" X, _ sorry ( _ very))
241.	Why I am very sorry?	am ( _ I, _ sorry ( _ very), "why?" X)
242.	Who will feel cold?	feel ( & will) ( "who?" X, _ cold )
243.	When will you feel cold?	feel ( & will) ( _ you, _ cold, "when?" X)
244.	What will you feel if you don't wear a coat?	_ if (feel (& will) ( _ you, "what?" X), wear (& don't) ( _ you, _ coat ( _ a)))
245.	What will stay crisp?	stay ( & will) ("what?" X, _ crisp)
246.	When biscuits will stay crisp?	will (stay ( _ biscuits, _ crisp), "when?" X)
247.	What will stay crisp if you keep them in an airtight container?	_ if (will (stay ("what?" X, _ crisp)), keep ( _ you, _ them, _ container ( _ airtight?( _ an ( _ in))))
248.	Who will stay hungry?	stay ( & will) ("who?" X, _ hungry)
249.	When you will stay hungry?	will (stay ( _ you, _ hungry), "when?" X)
250.	What you will stay if you don't eat right away?	_ if (will (stay ( _ you, "what?" X)), don't (eat ( _ you, _ away ( _ right))))
251.	Where you have to go?	have (go ( _ you, _ to, "where?" X))
252.	Where you have to go if you want to be a doctor?	_ if (have (go ( _ you, _ to, "where?" X)), want (be ( _ you, _ to, _ doctor ( _ a))))
253.	Why you have to go to college for a lot of years?	have (go ( _ you, _ college ( _ to), _ years ( _ of ( _ lot ( _ a))), "why?" X))
254.	What we have so many?	have ( _ we, "what?" X ( _ many ( _ so)))
255.	What we have yet so few?	have ( _ we, "what?" X ( _ few ( _ so ( _ yet))))
256.	What you don't know?	don't (know ( _ you), "what?" X)
257.	Who is glad?	is ("who?" X, _ glad)
258.	Who is glad that workers aren't losing their jobs?	_ that (is ("who?" X, _ glad), aren't (losing ( _ workers, _ jobs ( _ their ))))
259.	Why is Monika happy?	is ( _ Monika, _ happy, "why?" X)
260.	Why is Monika glad?	is ( _ Monika, _ glad, "why?" X)
261.	Who aren't losing their jobs?	aren't (losing ( "who?" X, _ jobs ( _ their))
262.	What was still painful?	was ( "what?" X, _ still, _ painful)
263.	Why do I went to see a specialist?	went ( & do) ( _ I, _ see ( _ to), _ specialist ( _ a), "why?" X)
264.	Who went to see a specialist?	went ( "who?" X, _ see ( _ to), _ specialist ( _ a))
265.	What she does not read?	does (not (read ( _ she, "what?" X)))
266.	What I wouldn't wear?	wouldn't (wear ( _ I, "what?" X))

267.	When I would wear those shoes?	would (wear ( _ I, _ shoes ( _ those), "when?" X)
268.	Who can ask for help?	ask ( & can) ("who?" X, _ help ( _ for))
269.	When can you ask for help?	can (ask ( _ you, _ help ( _ for), "when?" X)
270.	What can you whenever you need it?	_ whenever (can ( _ you, "what?" X), need ( _ you, _ it))
271.	What creak in the armchair?	creak ("what?" X, _ armchair ( _ the ( _ in)))
272.	When the springs in the armchair creak?	creak ( _ springs ( _ the), _ armchair ( _ the ( _ in)), "when?" X)
273.	Who is increasing the size of its fishing fleet while ours is contracting?	_ while (is (increasing ("who?" X, _ size ( _ the), _ fleet ( _ fishing ( _ its))), is (contracting ( _ ours)))
274.	What Spain is increasing?	is (increasing ( _ Spain, "what?" X)
275.	Who minded the dog?	minded ("who?" X, _ dog ( _ the))
276.	When does he minded the dog?	minded ( & does) ( _ he, _ dog ( _ the), "when?" X)
277.	Who minded the dog while I was away?	_ while (minded ("who?" X, _ dog ( _ the)), was ( _ I, _ away))
278.	When you can't be on the phone?	can't (be ( _ you, _ phone ( _ the ( _ on))), "when?" X)
279.	What changed while he was away?	_ while (changed ("what?" X), was ( _ he, _ away)
280.	Who is discussing?	discussing ( & is) ("who?" X)

## Privitak 9. Odgovori sustava na pitanja

R.B.	QTENG	Očekivani odgovor	Odgovor sustava	Analiza odgovora
1.	What are Tom and Paul talking about?	about the solution	about the solution	Točno
2.	Who talks about the solutions?	Tom and Paul	Tom and Paul	Točno
3.	Who discuss about the solutions?	Tom and Paul	Tom and Paul	Točno
4.	Who talks to Tom?	Paul		
5.	Who talks to Paul?	Tom		
6.	Who is talking?	Tom and Paul	Tom and Paul	Točno
7.	Who is playing?	Ana and John, Martina and Josip	Ana and John, Martina and Josip	Točno
8.	Who plays the violin?	Ana and John, Martina i Josip	Ana and John, Martina i Josip	Točno
9.	What are Ana and John playing?	the violin	the violin	Točno
10.	Who is Ana playing violin with?	John		Nema odgovora
11.	Who does John play the violin with?	Ana		Nema odgovora
12.	Who plays the violin extremely well?	Martina i Josip	Martina i Josip	Točno
13.	What does Sanja train?	volleyball and tennis	volleyball and tennis	Točno
14.	What does Sanja practise?	volleyball and tennis	volleyball and tennis	Točno
15.	Who trains volleyball?	Sanja, Peter	Sanja, Peter	Točno
16.	Who trains tennis?	Sanja, Peter	Sanja, Peter	Točno
17.	Who trains volleyball and tennis?	Sanja, Peter	Sanja, Peter	Točno
18.	What does Peter train?	basketball, volleyball and tennis	basketball and volleyball and tennis	Točno
19.	Who trains basketball?	Peter	Peter	Točno
20.	Who trains basketball, volleyball or tennis?	Peter, Sanja	Peter, Sanja	Točno
21.	Who are the programmers?	Sanja and Franjo	Sanja and Franjo	Točno
22.	What are Sanja and Franjo?	programmers	programmers	Točno
23.	What is Sanja?	programmer, happy	programmers, happy	Točno
24.	What is Franjo?	programmer	programmers	Točno
25.	Who are from Zagreb?	Marko and Ana, Lucija	Marko and Ana, Lucija	Točno
26.	Where are Ana and Marko from?	Zagreb/ from Zagreb	from Zagreb	Točno
27.	Where is Marko from?	Zagreb/ from Zagreb	from Zagreb	Točno



28.	Where is Ana from?	Zagreb/ from Zagreb	from Zagreb	Točno
29.	What were we?	worried, wet and tired	worried, wet and tired	Točno
30.	Who is wet?	we	we	Točno
31.	Who is tired?	we	we	Točno
32.	Who is wet and tired?	we	we	Točno
33.	What did we do to each other?	kissed and hugged		Nema odgovora
34.	Who did we kiss and hug?	each other	each other	Točno
35.	What did he draw and colour?	heart, a heart	a heart	Točno
36.	What did he draw?	heart, a heart	a heart	Točno
37.	What did he paint?	heart, a heart	a heart	Točno
38.	What did he colour?	heart, a heart, it	it	Djelomično točno
39.	How did he colour the heart?	red		
40.	Who drew a heart?	he	he	Točno
41.	Who came up to me?	girl, young girl, a young girl	a young girl	Točno
42.	What did the young girl ask?	for money, money		
43.	Which girl came?	young, a young	a young	Točno
44.	Who came up to me and asked for money?	girl, young girl, a young girl	a young girl	Točno
45.	Who asked me for money?	girl, young girl, a young girl		Nema odgovora
46.	Who became a dentist?	Monika	Monika	Točno
47.	Who became a patient dentist?	Monika	Monika	Točno
48.	Who became a attentive dentist?	Monika	Monika	Točno
49.	Who became a very patient and attentive dentist?	Monika	Monika	Točno
50.	What dentist did Monika become?	a very patient and attentive	very patient and attentive	Točno
51.	What did Monika become?	a very patient and attentive dentist	a very patient and attentive dentist	Točno
52.	How many is three and two?	five	five	Točno
53.	How much is five?	three and two	three and two	Točno
54.	What did I made?	dinner	dinner	Točno
55.	What was I watched?	television, the television	the television	Točno
56.	Who made dinner?	I	I	Točno
57.	Who watched television?	I		
58.	Who made dinner and watched television?	I	I	Točno
59.	Who got dressed?	I	I	Točno
60.	Who had breakfast?	I		Nema odgovora

61.	What did I get?	in even more trouble, a taxi, really tired, a headache	in even more trouble, a taxi, really tired, a headache	Točno
62.	What was the water?	lovely and cold	lovely and cold	Točno
63.	What was lovely and cold?	the water, water	the water	Točno
64.	What was lovely?	the water, water	water, and	Gramatički neispravno
65.	What was cold?	the water, water	water, and	Gramatički neispravno
66.	Who can come and see me tomorrow?	you	you	Točno
67.	Who can come?	you	you	Točno
68.	Who can see me?	you		Nema odgovora
69.	Who can see me tomorrow?	you		Nema odgovora
70.	What are snakes and crocodiles?	dangerous reptiles	dangerous reptiles	Točno
71.	What are reptiles?	snakes and crocodiles, turtles and lizards	snakes and crocodiles, turtles and lizards	Točno
72.	What are dangerous reptiles?	snakes and crocodiles	snakes and crocodiles	Točno
73.	What are turtles and lizards?	reptiles	reptiles	Točno
74.	What did I open?	the door	the door	Točno
75.	Who opened the door?	I	I	Točno
76.	Where I looked?	around	around	Točno
77.	Who looked around?	I		Nema odgovora
78.	Who is happy and healthy boy?	Tom	Tom	Točno
79.	Who is happy boy?	Tom	Tom	Točno
80.	Who is healthy boy?	Tom	Tom	Točno
81.	What is Tom?	happy and healthy boy	happy and healthy boy	Točno
82.	Where is Lucija from?	Zagreb or Rijeka	Zagreb or Rijeka from; is Zagreb or Rijeka from	Gramatički neispravno
83.	Who is from Zagreb?	Lucija	Lucija	Točno
84.	Who is from Rijeka?	Lucija	Lucija	Točno
85.	Who is from Zagreb or Rijeka?	Lucija	Lucija	Točno
86.	Who is coming on Tuesday or Wednesday?	Ana	Ana	Točno
87.	When will Ana come?	Tuesday or Wednesday	on, Tuesday or Wednesday	Gramatički neispravno
88.	Who will receive the main prize?	Monika or Sanja	Monika or Sanja	Točno
89.	What will Monika or Sanja receive?	the main prize, main prize	the main prize	Točno
90.	What is good or bad?	it	it	Točno

91.	Who will miss it all?	you	you	Točno
92.	What you will miss?	it all	it all	Točno
93.	What would better?	tell him miself	myself him tell	Gramatički neispravno
94.	Who will get in even more trouble?	I	I	Točno
95.	Who either take taxis or walk everywhere?	I	I	Točno
96.	Who take taxis?	I	I	Točno
97.	Who walk everywhere?	I		
98.	What I either take?	taxis	taxis	Točno
99.	What she never learned?	read or write, to read or write	read or write to	Gramatički neispravno
100.	Who never learned to read or write?	she	she	Točno
101.	Who should eat more?	you	you	Točno
102.	Who will make yourself ill?	you	you	Točno
103.	Why you should eat more?	you will make yourself ill	should or will make yourself ill	Gramatički neispravno
104.	Who love his job?	Marko, he	he	Djelomično točno
105.	What is Marko?	happy	happy	Točno
106.	Why is Marko happy?	because he love his job	is because love he his job	Gramatički neispravno
107.	Who loves her hobby?	she, Sanja	she	Djelomično točno
108.	Who is happy?	Sanja, Marko	Sanja, Marko	Točno
109.	Why is Sanja happy?	because she loves her interesting hobby	because loves she her interesting hobby	Gramatički neispravno
110.	What was quite quick?	the journy	the journy	Točno
111.	What was the journey?	quick, quite quick	quite quick	Točno
112.	Why was the yourney quite quick?	because the road was clear	because was the road clear	Gramatički neispravno
113.	What was clear?	road, the road	the road	Točno
114.	What was the road?	clear	clear	Točno
115.	Who is discharged from hospital?	patients	patients	Točno
116.	Who needed the beds?	other people	by other people	Gramatički neispravno
117.	What was needed by other people?	the beds	the beds	Točno
118.	Why were patients discharged from hospital?	because the beds were needed by other people	because were needed by other people the beds	Gramatički neispravno
119.	What I almost missed?	my flight	my flight	Točno
120.	Who almost missed the flight?	I		Nema odgovora
121.	Where was a long queue?	in the duty-free shop	there, in the duty-free shop	Gramatički neispravno

122.	What was in the duty-free shop?	a long queue	long a queue	Gramatički neispravno
123.	Why I almost missed my flight?	because there was a long queue in the duty-free shop	because was there long a queue in the duty-free shop	Gramatički neispravno
124.	Who didn't go out?	I	I	Točno
125.	Where I didn't go?	out	out	Točno
126.	Who had no many?	I	I	Točno
127.	What I dont had?	no money	no money	Točno
128.	Who didn't go out?	I	I	Točno
129.	Why didn't I go out?	because I had no money	because had I no money	Gramatički neispravno
130.	Why was she delighted?	because it was such surprise	because was it a surprise	Gramatički neispravno
131.	Who is sleepy?	Vanessa	Vanessa	Točno
132.	What is Vanessa?	sleepy	sleepy	Točno
133.	Who has been studying all night?	she, Vanessa	she	Djelomično točno
134.	Why is Vanessa sleepy?	because she has been studying all night	because has been studying she all night	Gramatički neispravno
135.	Who loves summer?	Daniel	Daniel	Točno
136.	Who loves swimming?	Daniel	he	Djelomično točno
137.	What does Daniel love?	summer, swimming	summer, swimming	Točno
138.	Why Daniel loves summer?	because he loves swimming	because loves he swimming	Gramatički neispravno
139.	Who was worried?	we	we	Točno
140.	What we were?	worried	worried	Točno
141.	Why we were worried?	because of the storm	the storm	Točno
142.	What we did?	it	it	Točno
143.	Why we did it?	because we felt it our duty	because felt our duty	Gramatički neispravno
144.	What we felt?	it our duty	it our duty	Točno
145.	Who felt it was their duty?	we		Nema odgovora
146.	What was cancelled?	the trip	the trip	Točno
147.	What was the trip?	canceled	canceled	Točno
148.	Why the trip was cancelled?	because of bad weather	of bad weather	Gramatički neispravno
149.	What they could do?	nothing, but swallow their pride	nothing	Djelomično točno
150.	What they swallowed?	their pride, pride		Nema odgovora
151.	Who swallows their pride?	they		Nema odgovora
152.	Where was nothing they could do?	there	there	Točno
153.	What is not caused by evil?	this	this	Točno

154.	What is not caused this?	by evil	by evil	Točno
155.	What is caused by simple ignorance?	this		Nema odgovora
156.	Who is running after him?	we	we	Točno
157.	Who are we running after?	him	him	Točno
158.	Who escaped?	he	he	Točno
159.	What is nice?	it, dress	it, a nice dress	Gramatički neispravno
160.	What creases very easily?	it, dress	it	Djelomično točno
161.	Who has received replies?	I	I	Točno
162.	Who received replies from everybody?	I	I	Točno
163.	Who I received replies from?	everybody	Jane ; from everybody	Djelomično točno
164.	Who did I not received replies from?	Jane	from everybody but Jane	Djelomično točno
165.	What did he felt for her?	nothing but contempt	nothing, but	Djelomično točno
166.	Who felt nothing?	he	he	Točno
167.	Who felt contempt?	he	he	Točno
168.	What they did not have?	alternative	alternative, but	Gramatički neispravno
169.	Who had no alternative?	they	they	Točno
170.	Who had to follow?	they	they	Točno
171.	Who is worried?	we, she	we, she	Točno
172.	Who will be hopefully fine?	she	she	Točno
173.	What is she?	worried, delighted, honest	worried, hopefully; delighted; honest	Gramatički neispravno
174.	What did I love?	math	math	Točno
175.	Who loved math?	I	I	Točno
176.	When did I love math?	when I was at school	when was at school	Točno
177.	What did I love when I was at school?	math	math	Točno
178.	Who was young?	I	I	Točno
179.	When were there no houses here?	when I was young	when was I young	Točno
180.	Who will go there?	I	I	Točno
181.	Who will go there every day?	I	I	Točno
182.	Where I will go?	there, new park	there	Djelomično točno
183.	When I go there?	every day	every day	Točno
184.	When I go to the new park?	every day		
185.	What will open?	the new park	the new park	Točno
186.	When did I have a great time?	when I went to the coast	when went to the coast	Točno
187.	Who had a great time?	I	I	Točno
188.	Who went to the coast?	I	I	Točno
189.	Where did I go?	the coast, there	the coast, there	Točno

190.	What is Saturday?	the day	the day	Točno
191.	When I get my hair done?	Saturday; Saturday is the day	when is saturday the day	Gramatički neispravno
192.	When I hate it?	when there's nobody in the office.	when there is nobody the office.	Točno
193.	Where I was just getting into?	into the bath	the into bath	Gramatički neispravno
194.	When the phone rang?	I was just getting into the bath	when was getting I just into bath	Gramatički neispravno
195.	What rang?	the phone	the phone	Točno
196.	Who was just getting into the bath?	I	I	Točno
197.	Where did I go when I was a child?	there	there	Točno
198.	When he was shocked?	when I told him	when told I him	Gramatički neispravno
199.	Who was shocked?	he	he	Točno
200.	Who told him?	I	I	Točno
201.	Who had her first baby?	she	she	Točno
202.	When she had her first baby?	when she was only 20	when was only 20	Točno
203.	Whom I will miss when I start college?	my old school friends	my old school friends	Točno
204.	When I will miss my old school friends?	When I start college	When start college	Točno
205.	Who have a complaint?	you	you	Točno
206.	When you have to write to director?	If you have complaint	If have you complaint	Točno
207.	Who let me know?	you		Nema odgovora
208.	What you are not sure about?	this	this	Točno
209.	When you let me know?	now	now	Točno
210.	Who can go?	you	you	Točno
211.	What can if you are 18 or over?	you can go		
212.	What I could just use?	the phone	the phone	Točno
213.	What I will get if I could just use the phone?	a taxi	a taxi	Točno
214.	Who will get a taxi?	I	I	Točno
215.	What do I wonder?	if she notice; if it can get any worse	if noticed she, if it get it any worse	Gramatički neispravno
216.	Who noticed?	she	she	Točno
217.	What can get any worse?	it	it	Točno
218.	What she was?	delighted, honest	delighted, honest	Točno
219.	Who was honest?	she	she	Točno
220.	Who was little brutal?	she		Nema odgovora
221.	Who asked?	he, you	you; he	Točno

222.	What he asked?	if we would like some coffee	if would like we some coffee	Gramatički neispravno
223.	Who asked if we would like some coffee?	he	he	Točno
224.	What we would like?	some coffee	some coffee	Točno
225.	Who can go?	you	you	Točno
226.	Where can you go?	here	here	Točno
227.	When can you go here?	if you get the point	if get the point	Točno
228.	Who get the point?	you	you	Točno
229.	What can you do if you get the point?	go here		Nema odgovora
230.	Who get really tired?	I	I	Točno
231.	When I get really tired?	if I stay up late	get if stay late up	Gramatički neispravno
232.	What if I stay up late?	I get really tired	get really tired	Točno
233.	Who will pay double?	I	I	Točno
234.	When will I pay you double?	if you get the work finished by Friday	if get the work finished by Friday	Točno
235.	How I will pay you if you get the work finished by Friday?	double	double	Točno
236.	Who had called?	she	she	Točno
237.	When I wouldn't have known?	If she hadn't called	if hadn't called she	Gramatički neispravno
238.	When I get a headache?	If I don't get enough sleep	If don't enough sleep	Gramatički neispravno
239.	What I get if I don't get enough sleep?	a headache	a headache	Točno
240.	Who is very sorry?	I	I	Točno
241.	Why I am very sorry?	if I've offended you	if have offended you	Točno
242.	Who will feel cold?	you	you	Točno
243.	When will you feel cold?	if you don't wear a coat	if wear don't a coat	Gramatički neispravno
244.	What will you feel if you don't wear a coat?	cold	cold	Točno
245.	What will stay crisp?	biscuits	biscuits	Točno
246.	When biscuits will stay crisp?	if you keep them in an airtight container	if keep you them airtight an container	Točno
247.	What will stay crisp if you keep them in an airtight container?	biscuits	biscuits	Točno
248.	Who will stay hungry?	you	you	Točno
249.	When you will stay hungry?	if you don't eat wright away	if don't eat wright away	Točno
250.	What you will stay if you don't eat right away?	hungry	hungry	Točno
251.	Where you have to go?	college	college	Točno
252.	Where you have to go if you want to be a doctor?	college	college	Točno

253.	Why you have to go to college for a lot of years?	if you want to be a doctor	if want to a doctor	Gramatički neispravno
254.	What we have so many?	question	many questions and yet few answers ; many questions	Gramatički neispravno
255.	What we have yet so few?	answers		Nema odgovora
256.	What you don't know?	what you can achieve until you try		Nema odgovora
257.	Who is glad?	Monika	Monika	Točno
258.	Who is glad that workers aren't losing their jobs?	Monika	Monika	Točno
259.	Why is Monika happy?	that workers aren't losing their jobs	that aren't losing workers their jobs	Gramatički neispravno
260.	Why is Monika glad?	that workers aren't losing their jobs	that aren't losing workers their jobs	Gramatički neispravno
261.	Who aren't losing their jobs?	workers	workers	Točno
262.	What was still painful?	it	it	Točno
263.	Why do I went to see a specialist?	It was still painful	so was it still painful	Gramatički neispravno
264.	Who went to see a specialist?	I	I	Točno
265.	What she does not read?	articles about her	about her articles	Gramatički neispravno
266.	What I wouldn't wear?	those shoes	those shoes	Točno
267.	When I would wear those shoes?	unless I was trying to break my ankle		Nema odgovora
268.	Who can ask for help?	you	you	Točno
269.	When can you ask for help?	whenever you need it	whenever need it	Točno
270.	What can you whenever you need it?	ask for help		Nema odgovora
271.	What creak in the armchair?	the springs, springs	springs	Točno
272.	When the springs in the armchair creak?	whenever I change position	whenever change I position	Gramatički neispravno
273.	Who is increasing the size of its fishing fleet while ours is contracting?	Spain	Spain	Točno
274.	What Spain is increasing?	the size of its fishing fleet	the size; its fishing fleet	Gramatički neispravno
275.	Who minded the dog?	He	He	Točno
276.	When does he minded the dog?	while I was away	while was I away	Točno
277.	Who minded the dog while I was away?	He	He	Točno
278.	When you can't be on the phone?	while driving a car	while driving a car	Točno
279.	What changed while he was away?	nothing much	nothing much	Točno
280.	Who is discussing?	Lucija and Ana	Lucija and Ana	Točno



## Životopis

Marina Rauker Koch rođena je 29. lipnja 1976. godine u Ogulinu. Od rođenja živi u Vrbovskom gdje je završila osnovnu školu. Od 1990. do 1994. pohađala je prirodoslovno – matematičku gimnaziju u ondašnjoj Srednjoj školi Ogulin, a današnjoj Gimnaziji Bernardina Frankopana.

Od 1998. do 2002. studira na Filozofskom fakultetu u Rijeci, gdje je završila studij pedagogije i informatike. Od prosinca 2002. do rujna 2015. radi u Željezničkoj tehničkoj školi Moravice kao nastavnik informatike i računalstva. Godine 2012. počela je raditi kao vanjski suradnik na Veleučilištu u Rijeci gdje je od 2015. stalno zaposlena kao predavač, a od 2020. godine kao viši predavač. U akademskoj godini 2023./2024. izvodi nastavu (predavanja, vježbe i seminare) i nositelj je kolegija Sučelje poslovnih i procesnih sustava, Modeliranje i simulacije, Izgradnja multimedijских sustava, Multimedijски sustavi, Računalna grafika, Informacijski sustavi, Informacijski sustavi u prometu, te Informacijski i komunikacijski sustavi u prometu.

Poslijediplomski doktorski studij Informatika upisala je 2013. na Odjelu za informatiku, danas Fakultetu informatike i digitalnih tehnologija Sveučilišta u Rijeci.

Od 2019. do 2022. godine sudjeluje kao suradnik na znanstvenom projektu (Sveučilišne potpore) „Razvoj NOK platforme za transformaciju rečenica prirodnog jezika u relacijsku bazu podataka“, a od 2024. godine sudjeluje kao suradnik na projektu „Razvoj modela sustava za provjeru znanja temeljenog na konceptualnom okviru Node of knowledge“.

U radu s učenicima, a danas studentima, a i kao članica Odbora društva Crvenog križa Primorsko – goranske županije, redovito je poticala i potiče volonterizam, društvenu odgovornost i humanitarni rad.

Znanstveni interesi obuhvaćaju područje metodologije razvoja informacijskih sustava i prikaza znanja. Do sada je objavila sljedeće znanstvene i stručne radove:

1. Guštin Nemanič, T., Rauker Koch, M. (2024). Analiza isplativosti uporabe sustava praćenja kretanja i rada kamiona. Zbornik radova Međimorskog veleučilišta u Čakovcu, 15(1), 8-15.
2. Rauker Koch, M., Tomljenović, Lj., Stilin, A. (2024.). Integracija normi ESG i ISO 9001 u visokom obrazovanju uz uporabu digitalnih alata. In 25. međunarodni simpozij KVALITETA - JUČER, DANAS, SUTRA.
3. Jerković, G., Rauker Koch, M., & Jauk, M. (2023). Multimedijske streaming platforme. Zbornik radova Međimorskog veleučilišta u Čakovcu, 14(1), 67-75.
4. Ašenbrener Katić, M., Čandrlić, S., & Rauker Koch, M. (2022). Adverb and Preposition Representation in Croatian and English using the Node of Knowledge Method. In 33rd Central European Conference on Information and Intelligent Systems (CECIIS 2022) (pp. 41-48).
5. Rauker Koch, Marina; Čandrlić, Sanja; Ašenbrener Katić, Martina, AUTOMATION OF THE CONVERSION OF NATURAL LANGUAGE TO FORMALIZED NODE OF KNOWLEDGE

RECORD // Zbornik Veleučilišta u Rijeci / Journal of the Polytechnic of Rijeka (2022)  
doi:10.31784/zvr.10.

6. Rauker Koch, Marina; Tomljenović, Ljerka; Brusić, Antonio, Društveno odgovorno učenje kao element nastave // Proceedings of the 7th International Conference "Vallis Aurea" Focus on: Research & Innovation / Katalinić, Branko (ur.). Požega, Hrvatska, 2020. str. 549-555 (poster, međunarodna recenzija, cjeloviti rad (in extenso), stručni)
7. Rauker Koch, Marina; Božičević, Matea; Dobrača, Igor, Prijedlog Infrared© zaštite pratećih dokumenata u prometu opasnim teretom // PRINTING & DESIGN 2018 / Žiljak Gršić, Jana (ur.). Zagreb: Fotosoft d.o.o., 2018. str. 46-56.
8. Rauker Koch, Marina; Ašenbrener Katić, Martina; Pavlić, Mile, Fable representation in FNOK and DNOK formalisms using the NOK conceptual framework // Proceedings of the 28th DAAAM International Symposium / Katalinic, Branko (ur.). Vienna: DAAAM International, Vienna, Austria, EU, 2017, 2017. str. 439-445  
doi:10.2507/28th.daaam.proceedings.061
9. Rauker Koch, Marina; Pavlić, Mile; Ašenbrener Katić, Martina, Homonyms and Synonyms in NOK Method // Procedia Engineering, 100 (2015), 1055-1061  
doi:10.1016/j.proeng.2015.01.466
10. Pavlić, Mile; Zdravko Dovedan Han; Jakupović, Alen; Čandrić, Sanja; Ašenbrener Katić, Martina; Tomljanović, Jasminka; Rauker Koch, Marina, Sustav za odgovaranje na pitanja prirodnim jezikom // Razvoj poslovnih i informatičkih sustava CASE 27 / Polonijo, Mislav (ur.). Rijeka: CASE d.o.o., 2015. str. 5-16
11. Rauker Koch, Marina; Pavlić, Mile; Jakupović, Alen, Application of the NOK method in sentence modelling // Proceedings of the 37th International Convention MIPRO 2014 / Biljanović, P. (ur.). Opatija: Croatian Society for Information and Communication Technology, Electronics and Microelectronics - MIPRO, 2014. str. 1426-1431  
doi:10.1109/MIPRO.2014.6859746
12. Rauker Koch, Marina; Pavlić, Mile; Ašenbrener Katić, Martina, Homonyms and Synonyms in NOK Method // Annals of DAAAM International for 2014., Volume 25, No 1 / Branko Katalinić (ur.). Beč, Austrija: DAAAM International Vienna, 2014. str. 1055-1061, doi:10.1016/j.proeng.2015.01.466
13. Gligora Marković, Maja; Antić, Mario; Rauker Koch, Marina, Proces razvoja multimedijske računalne igre // Zbornik Veleučilišta u Rijeci, 1 (2013), 1; 151-163
14. Rauker Koch, Marina; Brajdić, Iva, Bonton i autorsko pravo u digitalnom okruženju // 15. CARNetova korisnička konferencija – CUC 2013., Dohvati znanje! Zagreb: Carnet, 2013. str. 124-131
15. Gligora Marković, Maja; Rauker Koch, Marina; Frančić, Miro, Use of Web 2.0 tools in teaching // Međunarodni ICT skup - MIPRO 2012 / Marina Čičin-Šain, Jadranka Sunde, Nina Lipljin, Ivan Uroda, Ivana Turčić Prstačić, Ivanka Slugaonović (ur.). Opatija: MIPRO, 2012. str. 1465-1469