

**SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
VARAŽDIN**

Antun Tkalčec

BAZA ZNANJA O POSLOVNIM PRAVILIMA

PROJEKT

DEKLARATIVNO PROGRAMIRANJE

Varaždin, 2022.

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
V A R A Ž D I N

Antun Tkalčec

Matični broj: 0016136241

Studij: Baze podataka i baze znanja

BAZA ZNANJA O POSLOVNIM PRAVILIMA

PROJEKT

Mentor:

dr. sc. Bogdan Okreša Đurić

Varaždin, siječanj 2022.

Antun Tkalčec

Izjava o izvornosti

Izjavljujem da je moj projekt izvorni rezultat mojeg rada te da se u izradi istoga nisam koristio drugim izvorima osim onima koji su u njemu navedeni. Za izradu rada su korištene etički prikladne i prihvatljive metode i tehnike rada.

Autor potvrdio prihvatanjem odredbi u sustavu FOI Radovi

Sažetak

Opsega od 100 do 300 riječi. Sažetak upućuje na temu rada, ukratko se iznosi čime se rad bavi, teorijsko-metodološka polazišta, glavne teze i smjer rada te zaključci.

Ključne riječi: Flora-2, baza znanja, deduktivne baze podataka, transakcijska logika, F-logika, F-molekule

Sadržaj

1. Uvod	1
2. Formalizam deklarativnog programa	2
2.1. F-logika, HiLog, TL	2
2.2. F-molekule i semantika F-logike	3
3. Opis implementacije	4
3.1. Cilj aplikacije	4
3.2. Model baze znanja	4
3.3. Sustav Flora-2	5
4. Prikaz rada aplikacije	9
5. Kritički osvrt	12
6. Zaključak	13
7. Literatura	14
Popis literature	14
Popis slika	15
Popis isječaka koda	16

1. Uvod

Ovaj projektni rad bavit će se logikom temeljenom na okvirima, odnosno F-logikom koja je implementacija objektno-orijentiranog programiranja u logičkom programiranju. Implementacija baze znanja o poslovnim pravilima bit će izrađena u *Flora-2*, semantičkom sustavu temeljenom na pravilima čiji jezik potječe od F-logike, logike višeg reda i transakcijske logike [1]. Implementacija aplikacije ovog projekta sastojat će se od nekolicine takozvanih F-molekula koje će opisivati jedno poslovno okruženje školskog sustava, a prikazivat će poslovna pravila. Poslovna pravila mogu biti, između ostalog, hijerarhijska struktura nekog poslovanja te pravila po kojima se radnicima određuju plaće [6], a baza znanja sadržavat će upravo ta dva primjera poslovnih pravila.

Moja motivacija za odabir ove teme leži u činjenici da pohađam diplomski studij "Baze podataka i baze znanja", a kako već neko vrijeme volim objektno-orijentirano programiranje, odlučio sam uzeti ovu temu koja spaja baze znanja i objektno orijentirano programiranje.

U sljedećem poglavlju će pobliže biti objašnjeni pojmovi F-logike, F-molekula, baze znanja i ostalih već navedenih pojmova.

2. Formalizam deklarativnog programa

U ovom poglavlju će biti opisani prethodno spomenuti pojmovi, postavljeni teorijski temelji za izradu aplikacije te napisani predikati od kojih će se sastojati baza znanja.

2.1. F-logika, HiLog, TL

Kao što je već spomenuto, aplikacija i implementacija baze znanja bit će izrađena u Flori-2, čiji jezik potječe od F-logike, logike višeg reda (HiLog) i transakcijske logike (TL).

Logika višeg reda, ili HiLog je logika koja pruža čišću deklarativnu semantiku u usporedbi sa logikom prvog reda. Nastoji poboljšati i sintaksu i semantiku predikatne logike. Što se tiče sintakse, logika višeg reda dopušta pojavu varijabli u mjestima gdje se inače pojavljuju predikati ili simboli funkcija. HiLog semantika, za razliku od semantike prvog reda, se očituje u strukturama u kojima varijable mogu biti u rasponu domena relacija i funkcija izgrađenih iz domena pojedinaca. Logika višeg reda sadrži beskonačan raspon varijabli i konačan raspon simbola parametara [2].

F-logika, ili logika temeljena na okvirima, svojevrsna je implementacija objektno-orijentiranog programiranja u logičkom programiranju, stoga u F-logici postoje koncepti OOP-a [3]:

- objekti
- klase
- atributi
- metode
- nasljeđivanje itd.

Formule logike temeljene na okvirima grade se od [3]:

- konstruktori objekata: $F = \{a, b, c, f, k, a1, \dots\}$
- varijable: $V = \{x, y, z, x1, \dots\}$
- pomoćni simboli: $(,), [,], \rightarrow, \rightarrow\rightarrow, \bullet\rightarrow, \bullet\rightarrow\rightarrow, \dots$
- logički veznici i kvantifikatori: simboli disjunkcije, konjunkcije, negacije, "za svaki", "postoji barem jedan" itd.

Semantika F-logike je određena korištenjem F-struktura [7].

Transakcijska logika je logika u kojoj procedure mogu biti deklarativno specificirane, kao logički programi. Te procedure se zovu transakcije, a one su promjene u bazi znanja. Transakcijska logika osmišljena je prvenstveno za primjenu u bazama podataka, logičkom programiranju i umjetnoj inteligenciji. Logičko programiranje proširuje na način da uvodi operaciju

update te upite. U objektno-orijentiranim bazama podataka se može koristiti uz F-logiku kako bi pružala pristup metodama, koje su u suštini funkcije unutar nekog objekta [4]. Sintaksa transakcijske logike gradi se od [3]:

- konstanti (a,b,c,a1,...),
- varijabli (x,y,z,x1,...),
- funkcija (f,g,h,f1,...),
- predikata/relacija (P,Q,R,P1,...),
- logičkih i transakcijskih veznika (disjunkcija, konjunkcija, negacija, implikacija, ekvivalencija, ...)
- kvantifikatora ("za svaki", "postoji barem jedan")

2.2. F-molekule i semantika F-logike

F-molekula je svaki izraz oblika [3]:

Isječak kôda 1: F-molekula

```
1 objekt : klasa[
2   atribut_1 -> vrijednost_atributa_1,
3   ... ,
4   atribut_n -> vrijednost_atributa_n,
5   metoda_1( parametri_1 ) -> rezultat_1,
6   ... ,
7   metoda_m( parametri_m ) -> rezultat_m
8 ].
```

Semantički, *objekt* je instance klase *klasa* te ima za atribut *atribut_1* s vrijednosti *vrijednost_atributa_1*. Pozivom metode *metoda_1* s parametrima *parametri_1* dobiva se rezultat *rezultat_1*.

Primjer F-molekule koja će biti dio aplikacije izgleda ovako:

Isječak kôda 2: Primjer F-molekule iz aplikacije

```
1 ivica : ravnatelj [
2   ime -> Ivica,
3   prezime -> Ivic,
4   godiste -> 1964,
5   satnica -> 65,
6   radno_vrijeme -> od_do(7, 19),
7   podređeni -> { ivona, hrvoje, joza, stjepan }
8 ].
```


3. Opis implementacije

U ovom će poglavlju biti prikazan opis komponenti aplikacije, što uključuje model baze znanja, opis sustava u kojem će aplikacija biti napravljena, kao i cilj koji se aplikacijom želi postići.

3.1. Cilj aplikacije

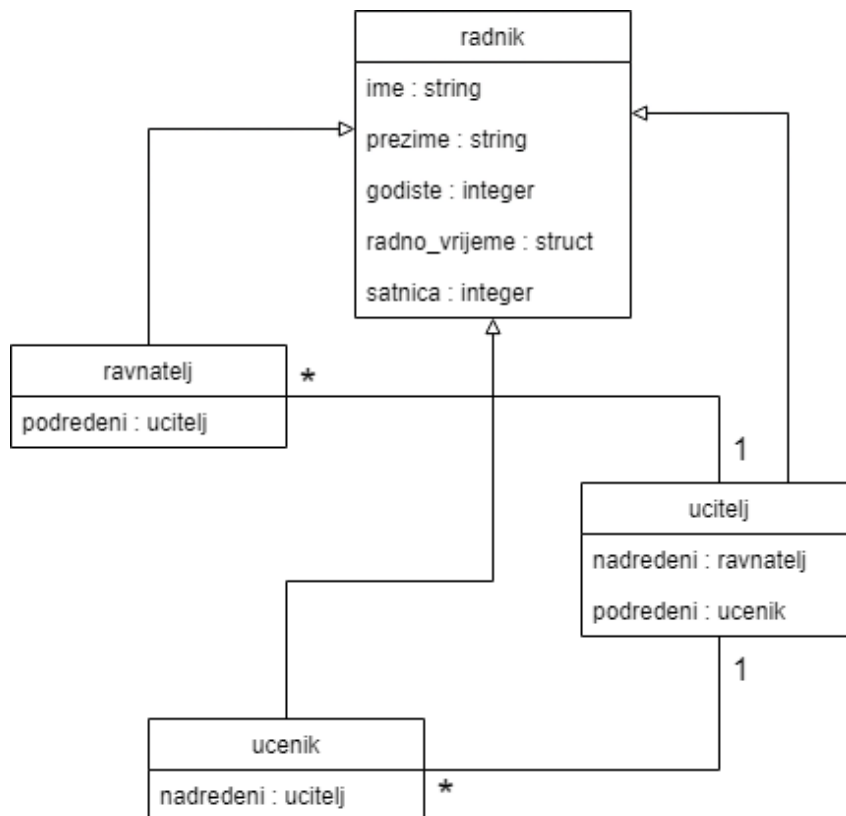
Aplikacija za cilj ima implementirati poslovnu hijerarhiju neke škole. Sastojat će se od baze znanja u Flori-2, gdje će F-molekule biti razine poslovanja u školi, npr. učenici, učitelji i ravnatelj.

3.2. Model baze znanja

Model baze znanja se sastoji od klase *radnici* koja je nadređena svim ostalim klasama. Ostale klase su:

- ravnatelj - sadrži podređenog učitelja
- ucitelj - sadrži nadređenog ravnatelja i podređene učenike
- ucenik - sadrži nadređenog učenika

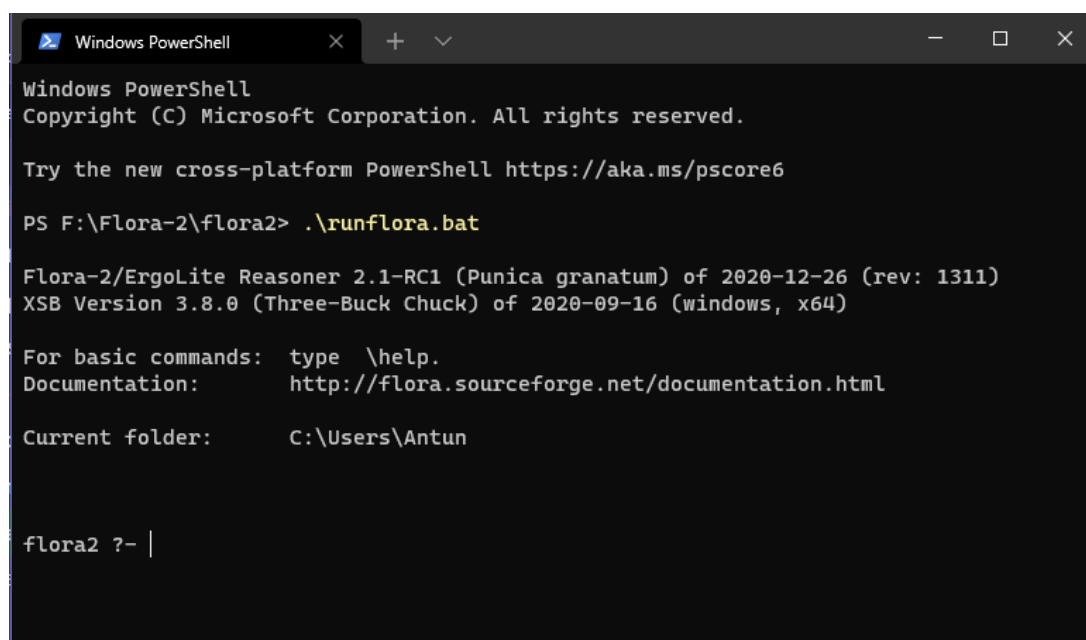
Klasa *radnici* sadrži imena, prezimena, godišta, radno vrijeme te satnicu svih sudionika ovog hipotetskog školskog sustava, s time da učenici, naravno, imaju satnicu u iznosu 0. Poslovno pravilo računanja plaća radnika ostvarit će se na način da se plaća računa prema satnici danog radnika i njihovom radnom vremenu.



Slika 1: Model baze znanja; autorov uradak

3.3. Sustav Flora-2

Kako bi se na računalo (Windows) instalirao sustav Flora-2, prvo ga je potrebno skinuti i instalirati s <http://flora.sourceforge.net/> [5]. Nakon instalacije, u Windows Terminalu se potrebno pozicionirati u direktorij gdje je instaliran, te upisati `.runflora.bat`.



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS F:\Flora-2\flora2> .\runflora.bat

Flora-2/ErgoLite Reasoner 2.1-RC1 (Punica granatum) of 2020-12-26 (rev: 1311)
XSB Version 3.8.0 (Three-Buck Chuck) of 2020-09-16 (windows, x64)

For basic commands: type \help.
Documentation:      http://flora.sourceforge.net/documentation.html

Current folder:    C:\Users\Antun

flora2 ?- |
```

Slika 2: Flora-2 u Windows Terminalu; autorov uradak

Nakon ovog koraka, za početak rada sa bazom znanja potrebno je napraviti datoteku sa nazivom npr. *flora.flr* i u nju upisati F-molekule aplikacije. Datoteku *flora.flr* odlučio sam *editati* u *Visual Studio Code*, a napisane F-molekule izgledaju ovako:

```

flora.flr
C: > Users > Antun > Desktop > DP > Projekt > flora.flr
1  radnik[
2      ime => \string,
3      prezime => \string,
4      godiste => \integer,
5      radno_vrijeme => struct,
6      satnica => \integer
7  ].
8
9  ravnatelj :: radnik[
10     podređeni => ucitelj
11 ].
12
13 ucitelj :: radnik[
14     nadređeni => ravnatelj,
15     podređeni => ucenik
16 ].
17
18 ucenik :: radnik[
19     nadređeni => ucitelj
20 ].
21
22
23 ivica : ravnatelj[
24     ime -> Ivica,
25     prezime -> Ivic,
26     godiste -> 1964,
27     radno_vrijeme -> od_do(7, 19),
28     satnica -> 65,
29     podređeni -> { ivona, hrvoje, joza, stjepan }
30 ].
31
32 ivona : ucitelj[
33     ime -> Ivona,
34     prezime -> Ivonkic,
35     godiste -> 1986,
36     radno_vrijeme -> od_do(8, 14),
37     satnica -> 45,
38     nadređeni -> ivica,
39     podređeni -> { hrvoje, joza, stjepan }
40 ].
41
42 hrvoje : ucenik[
43     ime -> Hrvoje,
44     prezime -> Hrvic,
45     godiste -> 2010,
46     radno_vrijeme -> od_do(8, 14),
47     satnica -> 0,
48     nadređeni -> ivona
49 ].
50
51 joza : ucenik[
52     ime -> Joza,
53     prezime -> Josefinski,
54     godiste -> 2009,
55     radno_vrijeme -> od_do(8, 14),
56     satnica -> 0,
57     nadređeni -> ivona
58 ].
59
60 stjepan : ucenik[
61     ime -> Stjepan,
62     prezime -> Stipic,
63     godiste -> 2011,
64     radno_vrijeme -> od_do(8, 14),
65     satnica -> 0,
66     nadređeni -> ivona
67 ].

```

Slika 3: Upis F-molekula; autorov uradak

Učitavanje ove baze znanja u sustav Flora-2 vrši se naredbom *load*:

```
flora2 ?- load{'C:/Users/Antun/Desktop/DP/Projekt/flora.flr'}
.
Times (in seconds): elapsed = 1.347; pure CPU = 1.234
Yes
flora2 ?- |
```

Slika 4: Učitavanje baze znanja u sustav; autorov uradak

Baza znanja je sada učitana u sustav, što možemo provjeriti na sljedeći način:

```
flora2 ?- radnik[
    ime => \string,
    prezime => \string,
    godiste => \integer,
    radno_vrijeme => struct,
    satnica => \integer
].
Times (in seconds): elapsed = 0.021; pure CPU = 0.016
Yes
flora2 ?- |
```

Slika 5: Provjera baze znanja; autorov uradak

Odgovorom "Yes" vidimo da je baza znanja učitana i da klasa *radnik* postoji.

4. Prikaz rada aplikacije

Budući da su F-molekule sada upisane, baza znanja je stvorena te je učitana u sustav Flora-2, možemo početi s prikazivanjem nekih primjera rada s aplikacijom. Zamislit ću da sam netko tko upravlja sustavom ove škole, te ću smisliti neke upite koji bi mi dali korisne informacije o ljudima i poslovnim pravilima unutar iste.

Prvu stvar koju bih volio saznati je tko sve radi ovdje i pohađa ovu školu. Upit će izgledati ovako:

```
flora2 ?- ?_ : radnik[

    ime -> ?ime,
    prezime -> ?prezime ].

?ime = Hrvoje
?prezime = Hrvic

?ime = Ivica
?prezime = Ivic

?ime = Ivona
?prezime = Ivonkic

?ime = Joza
?prezime = Josefinski

?ime = Stjepan
?prezime = Stipic

5 solution(s) in 0.000 seconds; elapsed time = 0.000
Yes
```

Slika 6: Upit 1; autorov uradak

Sličnim principom će biti stvoreni i prikazani ostali upiti nad bazom znanja.

Radno vrijeme svakog od radnika se može dohvatiti sljedećim upitom:

```
flora2 ?- ivica.radno_vrijeme = od_do(?od, ?do).

?od = 7
?do = 19

1 solution(s) in 0.000 seconds; elapsed time = 0.000
Yes
```

Slika 7: Upit 2; autorov uradak

Radno vrijeme radnika bit će važno za računanje njihove plaće.

Poslovno pravilo hijerarhije u školi može se prikazati na ovaj način:

```
flora2 ?- ?_ : radnik[ prezime -> ?prezime, podređeni -> ?podređeni ].  
  
?prezime = Ivic  
?podređeni = hrvoje  
  
?prezime = Ivic  
?podređeni = ivona  
  
?prezime = Ivic  
?podređeni = joza  
  
?prezime = Ivic  
?podređeni = stjepan  
  
?prezime = Ivonkic  
?podređeni = hrvoje  
  
?prezime = Ivonkic  
?podređeni = joza  
  
?prezime = Ivonkic  
?podređeni = stjepan  
  
7 solution(s) in 0.000 seconds; elapsed time = 0.000  
Yes
```

Slika 8: Upit 3; autorov uradak

Ovim upitom se dobiju prezimena nadređene osobe i prezimena svih osoba njima podređenima. Vidimo da su učenici Hrvoje, Joža i Stjepan podređeni Ivici Iviću, kao i učiteljica Ivona. Učiteljici Ivoni Ivonkić su podređeni učenici.

U suprotnom smjeru se hijerarhija dobiva ovako:

```
flora2 ?- ?_ : radnik[ prezime -> ?prezime, nadređeni -> ?nadređeni ].  
  
?prezime = Hrvic  
?nadređeni = ivona  
  
?prezime = Ivonkic  
?nadređeni = ivica  
  
?prezime = Josefinski  
?nadređeni = ivona  
  
?prezime = Stipic  
?nadređeni = ivona  
  
4 solution(s) in 0.000 seconds; elapsed time = 0.000  
Yes
```

Slika 9: Upit 4; autorov uradak

Kako bi dobili članove sustava koji primaju plaću, možemo postaviti sljedeći upit:

```
flora2 ?- ?_ : radnik [ ime -> ?ime, prezime -> ?prezime, satnica -> ?satnica ], ?satnica > 0.

?ime = Ivica
?prezime = Ivic
?satnica = 65

?ime = Ivona
?prezime = Ivonkic
?satnica = 45

2 solution(s) in 0.000 seconds; elapsed time = 0.000
Yes
```

Slika 10: Upit 5; autorov uradak

Za dobivanje prosječne satnice u ovoj školi možemo koristiti sljedeći upit:

```
flora2 ?- ?x = avg{?_g | ?_:radnik [ satnica -> ?_g ], ?_g > 0}.

?x = 55.0

1 solution(s) in 0.000 seconds; elapsed time = 0.000
Yes
```

Slika 11: Upit 6; autorov uradak

Za dobivanje mjesečne plaće radnika koji ju primaju, možemo koristiti sljedeći upit:

```
flora2 ?- ?_ : radnik [ ime -> ?ime, prezime -> ?prezime, satnica -> ?satnica, radno_vrijeme -> od_do(?od, ?do) ], ?satnica > 0, ?placa \is ?satnica * (?do - ?od) * 20.

?ime = Ivica
?prezime = Ivic
?satnica = 65
?od = 7
?do = 19
?placa = 15600

?ime = Ivona
?prezime = Ivonkic
?satnica = 45
?od = 8
?do = 14
?placa = 5400

2 solution(s) in 0.000 seconds; elapsed time = 0.000
Yes
```

Slika 12: Upit 7; autorov uradak

Ovime smo ostvarili i prikazivanje drugog poslovnog pravila, računanja plaća radnika. Plaća se računa na način da se satnica pomnoži sa brojem odrađenih sati u danu, pomnoženo sa 20 (broj radnih dana u mjesecu).

5. Kritički osvrt

Izradom ovog projekta dobio sam bolju sliku rada sa deduktivnim bazama podataka i bazama znanja te radom sa Flora-2 sustavom. Nakon dubljeg prolaska kroz ovu tematiku dobivam dojam da je ovakav način pohrane podataka vrlo inferioran običnim relacijskim bazama podataka, pa i onim polustrukturiranim. Razlog tome je što mi se ovaj način čini daleko manje skalabilan.

Za potrebe ovog projekta imao sam tek nekoliko jednostavnih komponenata baze znanja, a već i to mi je zadavalo glavobolju. Manjak grafičkog sučelja, nedostatak jednostavnosti izrade baze znanja koji se očituje u potrebi da se ona napravi u nečemu što je u biti tekstualna datoteka te *loada* u sustav Flora-2 i, meni osobno, apstraktan i nejasan jezik kojim se radi u Flori-2 čine savršen razlog da se više ne vraćam ovakvim načinima pohrane i obrade podataka.

Zbog svih spomenutih čimbenika, jasno mi je zašto u struci gotovo nema spomena o Prologu i Flori-2, kao niti *tutoriala* na YouTube-u (bar u slučaju Flore-2) ili neke korisnije, lako razumljive dokumentacije na internetu.

6. Zaključak

U ovom je projektnom radu, s ciljem prikaza rada aplikacije u sustavu *Flora-2*, postavljen teorijski temelj logike višeg reda, logike temeljene na okvirima te transakcijske logike. Navedene su osnovne značajke svake od tih triju vrsti logike, te su prikazani izgledi F-molekula unutar F-logike ili logike temeljena na okvirima. Izrađena je baza znanja o poslovnim pravilima čiji je model napravljen u online alatu i prikazan u poglavlju 3.2. Za implementaciju baze znanje korišten je sustav *Flora-2*, koji je usko vezan uz F-logiku, HiLog i transakcijsku logiku. Na kraju, prikazan je rad aplikacije u *Windows Terminal*, gdje su postavljani upiti uzlazne kompleksnosti i ostvaren cilj prikaza dvaju primjera poslovnih pravila koja su služila kao temelj i motivacija aplikacije.

Kako je već spomenuto u kritičkom osvrtu, autorovo mišljenje je da postoje mnogo jednostavniji i skalabilniji načini pohrane i obrade podataka. U svakom slučaju, može se reći da su deduktivne baze podataka i baze znanja vjerojatno neprikladne za širu struku, no zasigurno postoje oni koji su u sustavima poput *Prolog* i *Flora-2* pronašli svoju *nišu*.

7. Literatura

- [1] „Flora-2 - Wikipedia“. <https://en.wikipedia.org/wiki/Flora-2> (pristupljeno sij. 12, 2022).
- [2] W. Chen, M. Kifer, i D. S. Warren, „HiLog: A foundation for higher-order logic programming“, The Journal of Logic Programming, sv. 15, izd. 3, str. 187–230, velj. 1993, doi: 10.1016/0743-1066(93)90039-J.
- [3] Schatten, M., Fakultet organizacije i informatike, Deklarativno programiranje, Materijali dostupni na sustavu za e-učenje
- [4] Logic Programming. Pristupljeno: 12. siječanj 2022. [Na internetu]. Dostupno na: https://books.google.com/books/about/Logic_Programming.html?hl=hr&id=w9oabJA5J90C
- [5] „Flora-2“. <http://flora.sourceforge.net/> (pristupljeno sij. 15, 2022).
- [6] „What Are Business Rules and What Is Their Importance?“, Workflow Management Software by Integrify. <https://www.integrify.com/blog/posts/business-rules-explained/> (pristupljeno sij. 15, 2022).
- [7] M. Ledvinka i P. Křemen, „Formalizing Object-Ontological Mapping Using F-logic“, u Rules and Reasoning, sv. 11784, P. Fodor, M. Montali, D. Calvanese, i D. Roman, Ur. Cham: Springer International Publishing, 2019, str. 97–112. doi: 10.1007/978-3-030-31095-0_7.

Poveznica na Overleaf projekt: <https://www.overleaf.com/read/vpvpvrmmsgkwd>

Popis slika

1.	Model baze znanja; autorov uradak	5
2.	Flora-2 u Windows Terminalu; autorov uradak	6
3.	Upis F-molekula; autorov uradak	7
4.	Učitavanje baze znanja u sustav; autorov uradak	8
5.	Provjera baze znanja; autorov uradak	8
6.	Upit 1; autorov uradak	9
7.	Upit 2; autorov uradak	9
8.	Upit 3; autorov uradak	10
9.	Upit 4; autorov uradak	10
10.	Upit 5; autorov uradak	11
11.	Upit 6; autorov uradak	11
12.	Upit 7; autorov uradak	11

Popis isječka koda

1.	F-molekula	3
2.	Primjer F-molekule iz aplikacije	3