

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
VARAŽDIN

Marko Škalić

INTERAKTIVNA EDUKATIVNA APLIKACIJA ZA UČENJE PROLOGA

PROJEKT

UVOD U UMJETNU INTELIGENCIJU

Varaždin, 2024.

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

Marko Škalić

Matični broj: 0016149028

Studij: Informacijski i poslovni sustavi 1.2

INTERAKTIVNA EDUKATIVNA APLIKACIJA ZA UČENJE PROLOGA

PROJEKT

Mentor:

Prof. dr. sc. Markus Schatten

Varaždin, siječanj 2024.

Marko Škalić

Izjava o izvornosti

Izjavljujem da je ovaj 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

U razvoju interaktivne edukativne aplikacije za učenje Prologa, cilj je bio stvoriti platformu koja studentima omogućuje učenje osnova ovog logičkog programskog jezika. Aplikacija je zamišljena kao alat koji nudi teorijske lekcije, interaktivne zadatke i kvizove za provjeru znanja, naime u prvotnoj verziji to trenutno nije implementirano. Ono što je implementirano ujedno je i glavna značajka, a to je integrirani Prolog interpreter koji korisnicima omogućava da u realnom vremenu isprobavaju svoje kodove i odmah dobivaju povratnu informaciju. Dakle s pomoću integriranog Tau Prolog interpretera korisniku se omogućava da testira svoj Prolog kod i provjerava ishod izvršavanja. Dodatno, implementiran je vrlo jednostavan oblik umjetne inteligencije koji analizira korisnički kod, odnosno prepoznaje ključne riječ koje korisnik koristi u kodu. Primarno provjerava da li se predikat koji je tražen u zadatku i nalazi u unesenom kodu. Iako je AI ovdje prvenstveno keyword-based i ne uključuje napredne NLP tehnike, pruža osnovnu analitičku podršku koja može biti korisna početnicima u razumijevanju strukture i logike Prologa. Cilj aplikacije je da korisniku osiguramo interaktivan okvir za učenje, što učenje Prologa čini pristupačnijim i angažiranijim.

Ključne riječi: Prolog, Flask, HTML, Tau, JavaScript, interaktivno, sučelje

Sadržaj

1.	Uvod	1
2.	Metode i tehnike rada	2
3.	Razrada teme	3
4.	Implementacija	4
	4.0.1. Kratak opis dijelova programa	4
5.	Prikaz rada aplikacije	7
6.	Zaključak	10
	Popis literature	11
	Popis slika	12

1. Uvod

Prolog je logički programski jezik. Ima važnu ulogu u umjetnoj inteligenciji. Za razliku od mnogih drugih programskih jezika, Prolog je namijenjen prvenstveno kao deklarativni programski jezik. U prologu se logika izražava kao relacije (nazvane Činjenice i Pravila). Srž prologa leži u primijenjenoj logici[1].

Odlučio sam se baš za ovu temu zato što smo se s prologom upoznali također u ovom semestru na predmetu "Uvod u modeliranje znanja", te sam mislio kako ću moći spojiti znanja s tog predmeta i iskoristiti ih u ovom projektu.

Osnovna ideja projekta je bila da aplikacija bude interaktivna, te da omogućava da korisniku da izravno u njoj isprobava svoj Prolog kod. Također je u planu bilo implementirati model umjetne inteligencije koji će moći pročitati korisnikov programski kod te nam reći što se točno s njime pokušava otkriti.

U nastavku ćemo proći kroz sve aspekte aplikacije, objasniti pojedine dijelove, te tehnologije koje su korištene, također ćemo na primjeru prikazati kako funkcionira interaktivno sučelje.

2. Metode i tehnike rada

Za pisanje i izvršavanje programskog koda korišten je Visual Studio Code. To je zapravo alat koji smo koristili kao integrirano razvojno okruženje (engl. *Integrated Development Environment*) za programski jezik Python.

Koristio sam Flask za izradu back-end dijela aplikacije. Flask je lagan i snažan web framework za Python, koji je poslužio kao temelj aplikacije. Flask je omogućio kreiranje web servera i definiranje ruta, upravljanje HTTP zahtjevima, učitavanje HTML predložaka, te obradu podataka poslanih kroz forme.[2]

HTML, CSS i JavaScript. Ove web tehnologije su korištene za izradu korisničkog sučelja (UI). HTML sam koristilo za strukturu web stranica, CSS za stiliziranje i dizajn, te JavaScript za klijentsku interaktivnost i asinkrone zahtjeve na server s pomoću AJAX-a.

Tau Prolog je JavaScript implementacija Prologa koja je omogućila da izvršavamo Prolog kod direktno unutar web preglednika, pružajući korisniku instantne povratne informacije o ishodima njegova Prolog upita[3]. Ovo je ključno za interaktivni dio aplikacije gdje korisnik može unijeti i testirati svoj Prolog kod.

Koristio sam JSON format za strukturiranje i pohranu podataka o zadacima i njihovim rješenjima. JSON datoteka je poslužila kao lagan i pristupačan način za učitavanje zadataka u aplikaciju i njihovu analizu.

U aplikaciji sam implementirao osnovnu logiku umjetne inteligencije za analizu Prolog koda. Koristio sam jednostavne if-else izjave da bi program detektirao ključne riječi i izrazilo namjere koda, što korisnicima omogućuje razumijevanje što njihov kod pokušava postići.

3. Razrada teme

Cilj projekta bio je omogućiti studentima i svima zainteresiranim za učenje Prologa pristup interaktivnom alatu koji potiče učenje kroz praktično iskustvo. Prolog je programski jezik temeljen na predikatnoj logici, što ga čini idealnim za predstavljanje i rješavanje problema u domeni umjetne inteligencije. Algoritmi u Prologu koriste unaprijed definirana pravila i činjenice za izvođenje zaključaka. **PrologBook** Interaktivni pristup u aplikaciji omogućava korisniku da odmah testira svoje kodove i vidi rezultate izvršenja, što je značajno poboljšava proces učenja.

Što se tiče umjetne inteligencije, u početnim fazama razvoja, primjena umjetne inteligencije bila je zamišljena u obliku analize koda i automatske provjere ispravnosti rješenja koje studenti unose. Formalizam koji bi se implementirao obuhvaća NLP (*Natural Language Processing*) tehnike za razumijevanje namjera koda i pravila koja su napisana u Prologu. To bi uključivalo semantičku analizu koda kako bi se izvukli ključni pojmovi i strukture koji su relevantni za zadani problem, te bi na osnovu toga AI mogao davati povratnu informaciju o točnosti ili netočnosti rješenja.

Prva faza implementacije umjetne inteligencije u aplikaciji uključuje jednostavan oblik analize koda koji se temelji na prepoznavanju ključnih riječi unutar Prolog koda. To je realizirano kroz funkciju `analyze_code_intent[4]` u Flask aplikaciji. Ova funkcija prima Prolog kod i provjerava prisutnost određenih ključnih riječi ili fraza, kao što su *'parent'*, *'factorial'*, *'voli_pizzu'*, i na temelju toga vraća osnovnu analizu namjere koda. Ovaj pristup, iako ograničen, omogućava korisnicima neku osnovnu razinu uvida u to što njihov kod može raditi, što je posebno korisno za početnike.

Iako trenutna implementacija ne uključuje napredne AI tehnike, projekt bi se mogao proširiti uključivanjem sofisticiranijih metoda, kao što su sintaksna i semantička analiza koda, što bi omogućilo dublje razumijevanje i interpretaciju Prolog koda. Primjerice, integracija s alatima za statičku analizu koda ili razvoj vlastitog parsera koji bi mogao interpretirati i analizirati strukturu Prolog programa omogućila bi detaljniju i precizniju povratnu informaciju korisnicima. Također, primjena tehnika strojnog učenja za analizu uzoraka u kodu mogla bi dodatno poboljšati sposobnost aplikacije da pruža relevantne savjete i sugestije.

Trenutna implementacija pruža temelj za buduće proširenje i integraciju naprednijih AI tehnika. Praktičnost aplikacije leži u njenoj sposobnosti da omogući korisnicima interaktivno i angažirano učenje. Aplikacija može poslužiti kao alat za podršku nastavi, samostalno učenje i eksperimentiranje s Prologom. Kritički gledano, aplikacija trenutno pruža osnovnu funkcionalnost i povratne informacije, ali postoji značajan potencijal za nadogradnju kako bi se unaprijedila korisnička iskustva i obrazovna vrijednost.

4. Implementacija

Backend je implementiran inicijalizacijom Flask Aplikacije: Kreiran je Flask app kao osnova za web aplikaciju. Flask sam odabrao zbog njegove jednostavnosti i fleksibilnosti. Definirane su rute za različite dijelove aplikacije, uključujući početnu stranicu, stranice za lekcije i kvizove, te glavnu stranicu za interakciju s Prolog kodom (/projekti). Učitavanje Zadataka: Funkcija *load_zadaci* učitava zadatke iz JSON datoteke, te tako omogućuje dinamičko upravljanje sadržajem.

Za frontend su korištene web tehnologije. HTML: Izrađena je struktura web stranice koristeći HTML, uključujući tekstualna polja za unos Prolog koda i naredbi, te gumbе za njihovu obradu. CSS: Stiliziranje stranice izvedeno je s pomoću CSS-a radi poboljšanja korisničkog iskustva i interaktivnosti. (Uz pomoć ChatGpta, stvoreni HTML predlošci te napisan CSS kod, kako bi se ubrzao projekt) JavaScript: Implementirana je logika na klijentskoj strani za interakciju s Prolog interpreterom i backend-om. Uključuje slanje zahtjeva za provjeru i analizu koda.

Integracija Tau Prolog Interpretera važan je bio korak u izradi projekta. Korištenje Tau Prologa, JavaScript implementacije Prologa, omogućuje izvršavanje Prolog koda unutar preglednika. To omogućava korisnicima da odmah vide rezultate svog koda.[5] Korisnici mogu unositi Prolog kod direktno u tekstualno polje i izvršavati ga pomoću Tau Prolog interpretera integriranog u web sučelje. Nakon izvršavanja koda, rezultati se prikazuju u realnom vremenu, što omogućava korisnicima trenutno razumijevanje učinka svog koda. Ova neposredna povratna informacija je iznimno korisna za učenje, jer korisnici mogu eksperimentirati s različitim pristupima i odmah vidjeti rezultate svojih promjena.

Implementacija AI za Analizu Koda. Kao što je već spomenuto prilično je jednostavan model odabran za prvobitnu verziju aplikacije. U Flask backendu, implementirana je funkcija *analyze_code_intent* koja prima Prolog kod i analizira ga pretraživanjem ključnih riječi. Ovisno o pronađenim ključnim riječima, vraća opis onoga što kod može raditi. Ovaj pristup je osnovan, ali pruža korisnicima smjernice i pomaže im u razumijevanju funkcionalnosti njihovog koda. Iako ova analiza nije duboka ili kompleksna, ona pruža koristan početni uvid i može biti posebno korisna za početnike u Prologu. Na frontendu, kada korisnici kliknu na gumb "Analiza programa", JavaScript šalje trenutni Prolog kod na server preko AJAX zahtjeva. Server vraća analizu, koja se zatim prikazuje korisniku.

4.0.1. Kratak opis dijelova programa

Na slici 1 prikazuje isječak koda gdje je vidljivo mapiranje URL-ova. *@app.route('/')* osnovna je ruta koja je odnosi na domenu stranice. Za pristup ostalim stranicama iza znaka *'/'* se pišu njihovi nazivi. [6]

Na slici 2 na početku vidljiva je funkcija *"load_zadaci"* koja otvara JSON datoteku te pristupa zadacima koji su spremljeni u JSON datoteci. Nadalje funkcija *"analyze_code_intent(prolog_code)"* prima string Prolog koda i provjerava prisutnost ključnih riječi. Ovisno o ključnoj riječi vraća raz-

```

app = Flask(__name__)

@app.route('/')
def home():
    return render_template('index.html')

@app.route('/lekcije')
def lessons():
    return render_template('lekcije.html')

@app.route('/kvizovi')
def quizzes():
    return render_template('kvizovi.html')

@app.route('/projekti')
def projects():
    return render_template('projekti.html', zadaci=zadaci)

```

Slika 1: Mapiranje ruta

```

def load_zadaci():
    with open('resursi/zadaci.json', 'r', encoding='utf-8') as file:
        zadaci = json.load(file)
    return zadaci

zadaci = load_zadaci()

@app.route('/analyze_prolog_code', methods=['POST'])
def analyze_prolog_code():
    data = request.get_json()
    prolog_code = data['code']
    analysis = analyze_code_intent(prolog_code)
    return jsonify({"analysis": analysis})

def analyze_code_intent(prolog_code):
    if 'djedbaka' in prolog_code:
        return "Ovaj kod pokušava otkriti unuke od osobe."
    elif 'factorial' in prolog_code:
        return "Ovaj kod računa faktoriijel broja."
    elif 'voli_pizzu' in prolog_code:
        return "Ovaj kod provjerava koja osoba voli pizzu."
    elif 'putanja' in prolog_code:
        return "Ovaj kod provjerava postoji li put između dva ruba na grafu."
    elif 'kombinacija' in prolog_code:
        return "Ovaj kod provjerava sve moguće kombinacije tri broja."
    else:
        return "Nije mi jasno što se u kodu traži."

if __name__ == '__main__':
    app.run(debug=True)

```

Slika 2: Jednostavna umjetna inteligencija

```

function runProlog() {
    var session = pl.create();
    var code = document.getElementById('prolog-code').value;
    var query = document.getElementById('prolog-query').value;
    var output = document.getElementById('prolog-output');
    output.textContent = "";

    session.consult(code, {
        success: function() {
            session.query(query, {
                success: function(goal) {
                    var getAnswer = function() {
                        session.answer({
                            success: function(answer) {
                                output.textContent += pl.format_answer(answer) + "\n";
                                getAnswer(); // Rekurzivno dohvaćanje sljedećeg odgovora
                            },
                            fail: function() {
                                // Kada nema više rješenja
                                output.textContent += "\n";
                            },
                            error: function(err) {
                                output.textContent += "Greška: " + err.toString();
                            },
                            limit: function() {
                                output.textContent += "Dosegnuto ograničenje.\n";
                            }
                        });
                    };
                    getAnswer(); // iteracija kroz sva rješenja
                }
            });
        }
    });
}

```

Slika 3: runProlog funkcija

ličite poruke, odnosno vraća grešku ako ne prepozna ključnu riječ.

Na slici 3 vidljiv je isječak JavaScript koda sa stranice projekti.html. Ovaj funkcija je možda i najvažnija. Ukratko, ovaj kod omogućuje korisniku da unese Prolog kod i upit, te pokrene interaktivnu sesiju u kojoj se Prolog kod izvršava, upiti se postavljaju, a odgovori prikazuju. Koristi se asinkroni pristup, gdje se za obradu uspjeha ili neuspjeha koriste callback funkcije. Također, koristi se rekurzija za prikupljanje i prikazivanje svih odgovora koji odgovaraju upitu, dok postoji posebna obrada za greške i ograničenja koja se mogu pojaviti tijekom izvođenja Prolog sesije.

5. Prikaz rada aplikacije

Nakon što smo pokrenuli server u terminalu možemo u web pregledniku otići na adresu `http://127.0.0.1:5000/`. Ondje bi nam se trebala prikazati početna stranica. (Vidljivo na slici 1)



Slika 4: Početna stranica aplikacije

S lijeve strane je sidebar i trenutno postoje tri destinacije gdje možete ići

- Lekcije
- Kvizovi
- Projekti i Primjeri koda

U planu je bilo da na stranici Lekcije budu postavljeni materijali za učenje prologa, ali trenutno je prazna. Slično i sa stranicom Kvizovi, na njoj je zamišljeno da se mogu rješavati kvizovi, ali trenutno nema dodanih kvizova.

Ono što nas najviše zanima je stranica Projekti i Primjeri koda. Na toj stranici se na vrhu nalazi forma u kojoj možete unositi i testirati vlastiti prolog kod. Ispod forme je linije preko koje se pokreću pravila. Pored linije je gumb "Provjeri rješenje". Klikom na gumb ispisuje se na ekran rješenje upita ili odgovarajuća poruka o grešci. Također postoji gumb "Analiza programa". Klikom na njega aplikacija pokušava povezati odgovarajući kod u formi s opisom i ključnim riječima nekog zadatka, te ako zna o čemu se radi javlja na ekranu što misli da se u kodu traži. Na kraju na dnu stranice imamo kratke opise zadataka. (vidljivo na slici 2)

Početna

Lekcije

Kvizovi

Projekti i Primjeri koda

Projekti i Primjeri Koda

Unesite svoj Prolog kod ovdje...

Upišite naredbu...

Provjeri Rješenje

Analiza programa

Zadatak 1
Napišite funkciju koja pronalazi unuke za zadane roditelje. (koristi predikat 'djedbaka')

Zadatak 2
Napišite funkciju koja izračun faktorijela broja. (koristi 'factorial')

Zadatak 3
Napišite funkciju koja pronalazi osobu koja voli jesti pizzu. (koristi predikat 'voli_pizzu')

Zadatak 4
Napišite funkciju koja pronalazi postoji li put između dva ruba na grafu. (koristi predikat 'putanja')

Zadatak 5
Napišite funkciju koja pronalazi sve moguće kombinacije tri broja. (koristi predikat 'kombinacija')

Slika 5: Izgled stranice Projekti i kodovi

Na slici 3 vidljiv je prikaz stranice kada korisnik upiše neki programski kod. U ovom slučaju odlučili smo se za rješavanje trećeg zadatka, u kojem pokušavamo otkriti koje sve osobe vole pizzu.

Zadatak je postavljen uz pomoć četiri činjenice oblika "*voli(osoba,hrana)*". Svaka od ovih činjenica koristi predikat *voli* sa dva argumenta.

Pravilo je izraz koji definira odnos između činjenica i dolazi u obliku glava:-tijelo. Konkretno u našem primjeru bi se pravilo "*voli_pizzu(X) :- voli(X, pizza)*." čitalo kao: Osoba X voli pizzu, ako je pizza napisana pod argumentom hrana.

U liniju ispod forme upisujemo pravilo (obavezno sa točkom na kraju) i pritiskom na gumb "Provjeri rješenje", ispisuje nam se traženo rješenje, odnosno X = sven, X = paula, što upravo znači da te dvije osobe vole pizzu.

Pritiskom na gumb "Analiza programa" vidljivo je da nam se na ekran ispiše poruka "Ovaj kod provjerava koja osoba voli pizzu", time možemo zaključiti da je aplikacija pravilno pročitala ključne riječi iz forme te ih povezala sa backendom aplikacije kako bi nam ispisala točnu poruku za ovaj programski kod.

Lekcije

Kvizovi

Projekti i Primjeri koda

```
voli(sven, pizza).
voli(domagoj, cips).
voli(paula, pizza).
voli(petra, steak).
voli_pizzu(X) :- voli(X, pizza).
```

Provjeri Rješenje

Analiza programa

X = sven X = paula

Ovaj kod provjerava koja osoba voli pizzu.

Zadatak 1
Napišite funkciju koja pronalazi unuke za zadane roditelje. (koristi predikat 'djedbaka'))

Zadatak 2
Napišite funkciju za izračun faktoriijela broja. (koristi 'factorial')

Zadatak 3
Napišite funkciju koja pronalazi osobu koja voli jesti pizzu. (koristi predikat 'voli_pizzu')

Slika 6: Slika sa unosom koda

6. Zaključak

U ovom radu predstavljena je interaktivna edukativna aplikacija za učenje Prologa, dizajnirana da pomogne studentima da se upoznaju s osnovama ovog logičkog programskog jezika. Aplikacija koristi Prolog zbog njegove jedinstvene deklarativne prirode koja omogućuje izražavanje logike kroz relacije i pravila. Glavna značajka aplikacije je njena sposobnost da bude interaktivna, dopuštajući korisnicima da direktno u aplikaciji isprobavaju svoj Prolog kod.

Korišteni alati u razvoju uključuju Flask za backend, Tau Prolog za izvođenje Prolog koda unutar preglednika, i HTML, CSS, te JavaScript za razvoj korisničkog sučelja. JSON se koristi za strukturiranje i pohranu podataka, dok se jednostavna AI logika koristi za analizu koda koristeći if-else izjave kako bi detektirala ključne riječi i izrazila namjere koda. U trenutnoj implementaciji, AI komponenta aplikacije je prilično osnovna, ali postoji potencijal za nadogradnju i uvođenje naprednijih tehnika poput strojnog učenja za analizu uzoraka u kodu.

Zaključno, aplikacija predstavljena u radu postavlja temelje za interaktivno učenje Prologa. Iako su neki planirani dijelovi još uvijek u razvoju, kao što su lekcije i kvizovi, ključna funkcionalnost interaktivnog testiranja i analize Prolog koda je uspješno implementirana. To korisniku pruža instantnu povratnu informaciju i potiče ih na eksperimentiranje, što je ključno za učenje. S obzirom na trenutnu funkcionalnost i potencijal za buduća poboljšanja, aplikacija ima sve preduvjete postati korisno sredstvo u obrazovanju i pružiti dublje razumijevanje Prologa i njegove primjene u umjetnoj inteligenciji.

Popis literature

- [1] „Prolog | An introduction,” GeeksForGeeks. (2020.), adresa: <https://www.geeksforgeeks.org/prolog-an-introduction/> (pogledano 11. 1. 2024.).
- [2] M. Grinberg, „Flask web development: Developing web applications with python,” O'Reilly Media, Inc., 2018.
- [3] J. A. RIAZA, „A prolog interpreter for the web,” *Theory and Practice of Logic Programming*, Cambridge University Press, 18. 9. 2023., str. 1–21.
- [4] C. Kulhan. „Artificial Intelligence with Prolog.” (31. 8. 2022.), adresa: <https://che-kulhan.medium.com/artificial-intelligence-with-prolog-93b572fd3ae6> (pogledano 11. 1. 2024.).
- [5] P. Brown. „Hello Tau Prolog.” (), adresa: <https://pbrown.me/blog/hello-tau-prolog/> (pogledano 11. 1. 2024.).
- [6] „Flask Routing,” Flask. (), adresa: <https://flask.palletsprojects.com/en/3.0.x/quickstart/#routing> (pogledano 5. 1. 2024.).

Popis slika

1.	Mapiranje ruta	5
2.	Jednostavna umjetna inteligencija	5
3.	runProlog funkcija	6
4.	Početna stranica aplikacije	7
5.	Izgled stranice Projekti i kodovi	8
6.	Slika sa unosom koda	9