



Logičko i funkcijsko programiranje

ŠKOLSKA 2025/26

Organizacija predmeta

Dve velike oblasti:

- Logičko programiranje – Prolog
- Funkcijsko programiranje – Haskell (na vežbama i Scala)

Predispitne obaveze:

- Dolasci – 4 poena
- Kolokvijumi – 23 + 23 poena
 - Minimum po kolokvijumu 10 poena
 - Obavezan bar jedan zadatak u Scala programskom jeziku
- Seminarski rad – 20 poena
 - Do kraja oktobra
 - Bilo koji logički ili funkcijski jezik
 - Odbrana u ispitnim rokovima

Završni ispit – 30 poena

- Uslov – 26 poena na kolokvijumima i dolascima

Komunikacija – Teams

- Logicko i funkcijsko programiranja 2025-26

Prolog

IVAN BRATKO, *PROLOG PROGRAMMING
FOR ARTIFICIAL INTELLIGENCE.*
PEARSON EDUCATION, 2001.

Logičko programiranja

Stil programiranja koji se bitno razlikuje od imperativnog programiranja

Programming in Logic – Prolog je najrasporstranjeniji jezik za ovu namenu

Prva implementacija – Univerzitet u Marseju početkom 1970tih

Dalji razvoj na Univerzitetu u Edinburgu je doveo do standardizovane verzije - Edinburški prolog

Prološki programi su deklarativni

Postoje samo dve komponente programa

- Činjenice i
- Pravila

Prološki sistem učitava program, a korisnik postavlja serije upita

Answer Set Programming (**ASP**) – čest pristup u stručnoj literaturi

Primeri primene

Ekspertski sistemi, sistemi za zaključivanje, reprezentacija znanja, automatsko zaključivanje

- Medicinska dijagnostika, pravni ekspertski sistemi, sistemi za planiranje
- Detekcija finansijskih prevara, analiza rizika, dijagnostika kvarova (NASA)
- Datalog – Souffle (Google)

Gramatike i parsiranje jezika

- Parsiranje rečenica, gramatički modeli, lingvistička istraživanja

Kombinatorni problemi i optimizacija

- Rasporedi, planiranje ruta, konfiguracioni sistemi
- Kompatibilnost komponenti, ograničenja konfiguracija (IBM, Siemens)

Prolog

Programiranje u Prologu se sastoji od :

- Definisanja **činjenica** o objektima i odnosima među njima
- Definisanja **pravila** o objektima i odnosima među njima
- Formiranju **upita** o objektima i odnosima među njima

Programi mogu biti “proverni” i/ili “generatorni”

Osnovni mehanizmi zaključivanja su:

- Pravilo rezolucije
- Backtracking

Imena **relacija** i **objekata** moraju početi MALIM SLOVOM

Imena **promenljivih** moraju početi VELIKIM SLOVOM ili “_”

Svaka činjenica ili pravilo mora se **završiti** TAČKOM “.”

SWI Prolog

<https://www.swi-prolog.org/download/stable>

Imenovanje izvornih datoteka – ime.pl

Instalacija: sudo apt-get install swi-prolog

Pokretanje: prolog

Odzivni znak: ?-

Učitavanje

- U okviru SWI-Prolog interpretera – navođenjem imena datoteka između znakova [] – [ime].
- Nakon izmena izvornih datoteka, ponovno učitavanje se obavlja komandom **make**.

Listanje

- svih učitanih predikata (relacija) – **listing**.
- određenih predikata (relacija) – **listing (ime_predikata)**.

Izlazak – **halt**.

SWI Prolog



```
SWI-Prolog (AMD64, Multi-threaded, version 9.0.2)
File Edit Settings Run Debug Help
Welcome to SWI-Prolog (threaded, 64 bits, version 9.0.2)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?-
```

```
SWI-Prolog (AMD64, Multi-threaded, version 9.0.2)
File Edit Settings Run Debug Help
Consult ...
Edit ...
New ...
Reload modified files
Navigator ...
Exit
```

Create new Prolog source

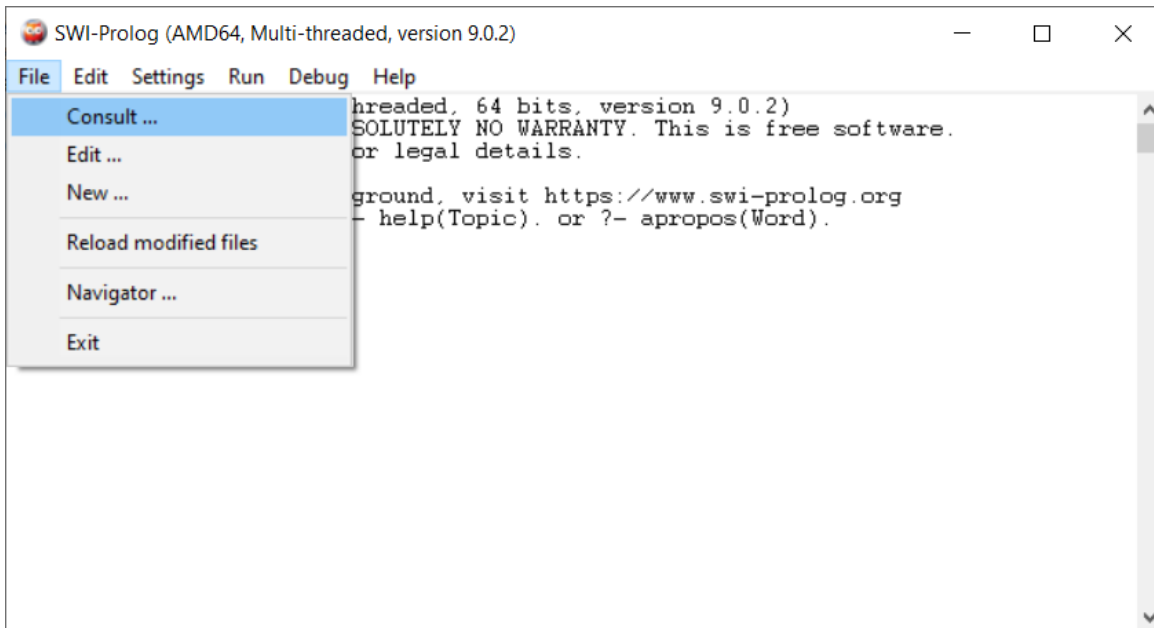
VI-MAS Univerzitet > Materijal > Prolog primeri

Name	Status	Date modified	Type
Primer1	🟢	12/8/2022 10:09 AM	Perl Source File

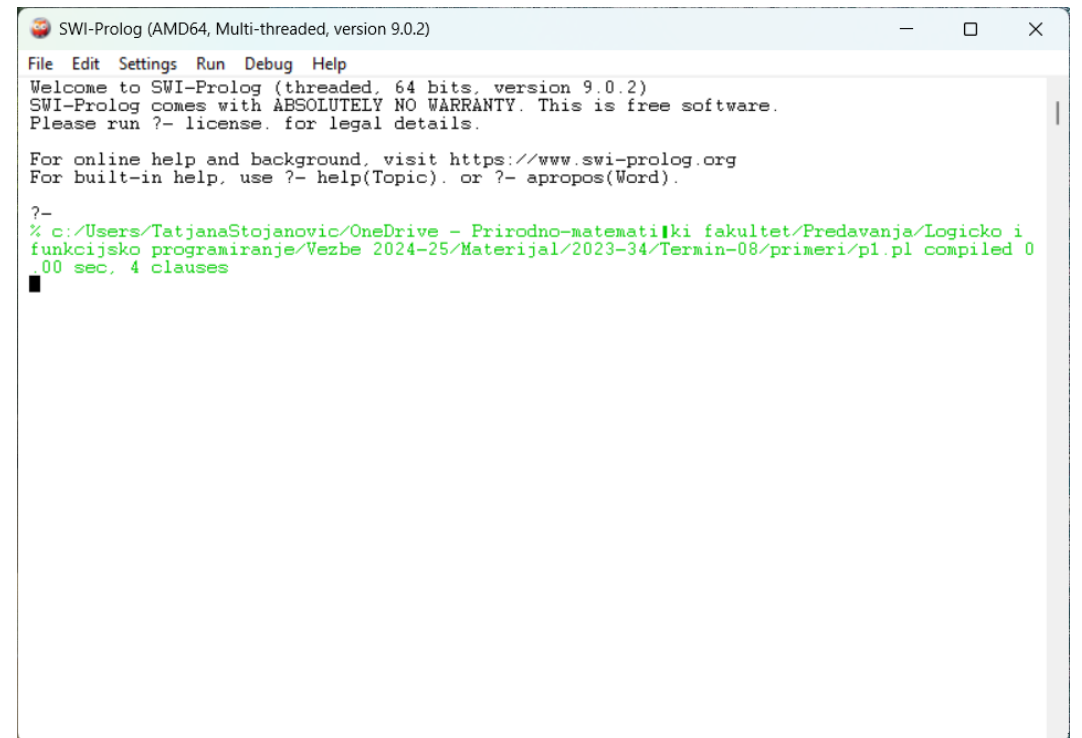
Primer1.pl

```
File Edit Browse Compile Prolog Pce Help
Primer1.pl
dog(fido).
dog(rover).
dog(henry).
cat(felix).
cat(michael).
cat(jane).
animal(X):-dog(X).
```

SWI Prolog



Može se integrisati sa Visual Studio Code



Prolog okruženje

?- -- prompt
?- [ime]. -- učitavanje programa
?-halt. -- izlazak
?-make. -- ponovno učitavanje

```
11 ?- write('Hello World'),nl,write('Welcome to Prolog'),nl.  
Hello World  
Welcome to Prolog  
true.
```

Primer

činjenice

```
dog(fido).  
dog(rover).  
dog(henry).  
cat(felix).  
cat(michael).  
cat(jane).  
animal(X):-dog(X).
```

pravilo

upiti

```
4 ?- [pr0501].  
true.  
5 ?- dog(fido).  
true.  
6 ?- dog(jane).  
false.  
7 ?- animal(fido).  
true.  
8 ?- dog(X).  
X = fido ;  
X = rover ;  
X = henry.  
9 ?- animal(felix).  
false.
```

Podaci u Prologu - termi

Brojevi

- Celi i razlomljeni, sa opcionim znakom + ili – ispred
 - 623
 - -47
 - +5
 - 025
 - 6.43
 - -.245
 - +256.
 - 0.56

Podaci u Prologu - termi

Atomi

- Konstante koje nemaju numeričku vrednost
- Niz slova, cifara i donje crte koji počinje malim slovom
 - john
 - today_is_Tuesday
 - fred_jones
 - a32_BCD
- Ali ne
 - Today
 - today-is-Tuesday
 - 32abc

Podaci u Prologu - termi

Atomi

- Konstante koje nemaju numeričku vrednost
- Niz karaktera ograničenih jednostrukim navodima
 - 'Today is Tuesday'
 - 'today-is-Tuesday'
 - '32abc'
- Niz specijalnih karaktera + - * / > < = & # @
 - +++
 - >=
 - >
 - +-

Podaci u Prologu - **termi**

Promenljive

- Niz slova, cifara i donje crte koji počinje velikim slovom
 - X
 - Author
 - Person_A
 - _123A
- Ali ne
 - 45_ABC
 - Person-A
 - authorToday is Tuesday'
 - 'today-is-Tuesday'
 - '32abc'

Podaci u Prologu - termi

Složeni termi

- Strukturirani podaci – predikati (ili funktori).
- Jedan ili više argumenata razdvojenih zarezom
- Opšti oblik $functor(t_1, t_2, \dots, t_n)$ $n \geq 1$, gde su t_i termi
 - likes(paul,prolog)
 - read(X)
 - dog(henry)
 - cat(X)
 - >(3,2)
 - person('john smith',32,doctor,london)

 - likes(dog(henry),Y)
 - pred3(alpha,beta,gamma,Q)
 - pred(A,B,likes(X,Y),-4,pred2(3,pred3(alpha,beta,gamma,Q)))

Podaci u Prologu - **termi**

Liste

- Specijalan tip strukturiranih podataka
- Neograničen broj argumenata (termova) razdvojenih zarezima i oivičeni srednjim zagradama []
 - `[dog,cat,y,mypred(A,b,c),[p,q,R],z]`
 - `[[john,28],[mary,56,teacher],robert,parent(victoria,albert),[a,b,[c,d,e],f],29]`
 - `[[portsmouth,edinburgh,london,dover],[portsmouth,london,edinburgh],[glasgow]]`
 - `[]`

Matematička logika

Iskazni račun

- Iskazna slova p, q, r, \dots
- Operatori $\neg, \wedge, \vee, \Rightarrow$
- Istinitosne vrednosti formula \top, \perp

Predikatski račun

- Skup funkcijskih simbola – f, g, h, f_1, f_2, \dots
- Skup relacijskih simbola – P, Q, R, P_1, P_2, \dots
- Operatori $\neg, \wedge, \vee, \Rightarrow, \exists, \forall$

- Koriste se i
 - Simboli promenljivih (zamena) – X, Y, Z, X_1, \dots
 - Simboli konstanti - a, b, c, \dots - funkcijski simboli arnosti 0

Reprezentacija zasnovana na logici

Kao i u predikatskom računu, predikati su osnovna reprezentaciona struktura u prologu

Prolog može da se posmatra kao dokazivač teorema koji prihvata upite korisnika i ispituje da li su tačni li ne

Srpski	Predikatski računa	Prolog
i	\wedge	,
ili	\vee	;
samo ako	\Leftarrow	:-
ne	\neg	not

Zarez koji razdvaja predikate, a ne elemente liste ili argumente

Reprezentacija zasnovana na logici

Engleski	Predikatski računa	Prolog
Everyone likes Susie	$\forall x Likes(x, Susie)$	likes(X, susie). likes(Everyone, susie).
George likes Kate and George likes Susie.	$Likes(George, Kate)$ $\wedge Likes(George, Susie)$	likes(george, kate), likes(george, susie).
George likes Susie if George does not like Kate	$Likes(George, Susie)$ $\Leftarrow \neg Likes(George, Kate)$	likes(george, susie) :- not(likes(george, kate)).

Hornove formule (kaluze)

Disjunkcija sa najviše jednim pozitivnim (nenegiranim) literalom

$$\neg p \vee \neg q \vee \dots \vee \neg t \vee u \quad \rightarrow \quad (p \wedge q \wedge \dots \wedge t) \Rightarrow u \quad \rightarrow \quad u: -p, q, \dots, t.$$

Specijalno

$$\neg p \vee \neg q \vee \dots \vee \neg t \quad \rightarrow \quad (p \wedge q \wedge \dots \wedge t) \Rightarrow \perp \quad \rightarrow \quad :-p, q, \dots, t.$$

Pravilo rezolucije je osnovni mehanizam zaključivanja

$$\frac{A \vee p, B \vee \neg p}{A \vee B}$$

$$\frac{A \vee p, B \vee \neg p}{A \vee B}$$

Zaključivanje

a.
b.
c: $\neg a, b$.
d: $\neg b, c$.
?-d.

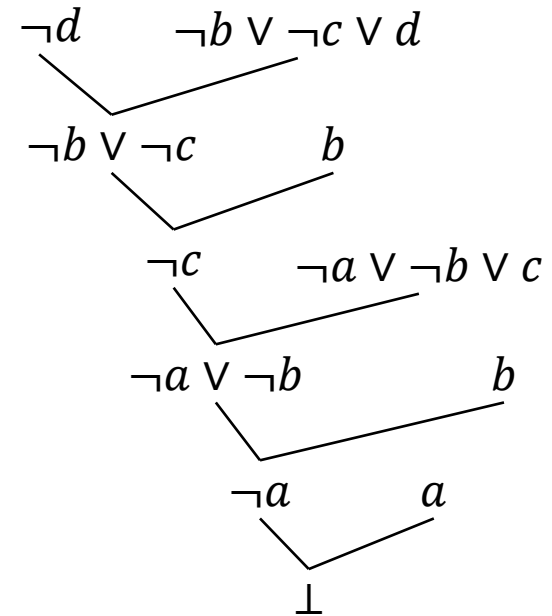
$\{a, b, a \wedge b \Rightarrow c, b \wedge c \Rightarrow d\} \vdash d$

tj.

$\{a, b, \neg a \vee \neg b \vee c, \neg b \vee \neg c \vee d\} \vdash d$

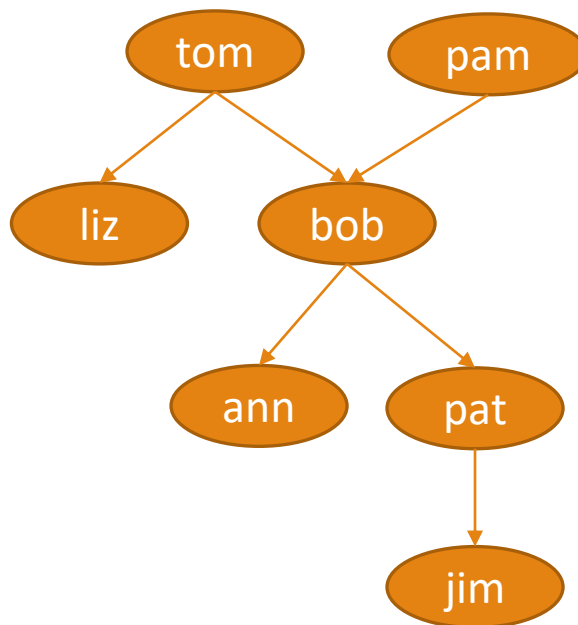
tj.

$\{a, b, \neg a \vee \neg b \vee c, \neg b \vee \neg c \vee d, \neg d\} \vdash \perp$



Definisanje činjenica

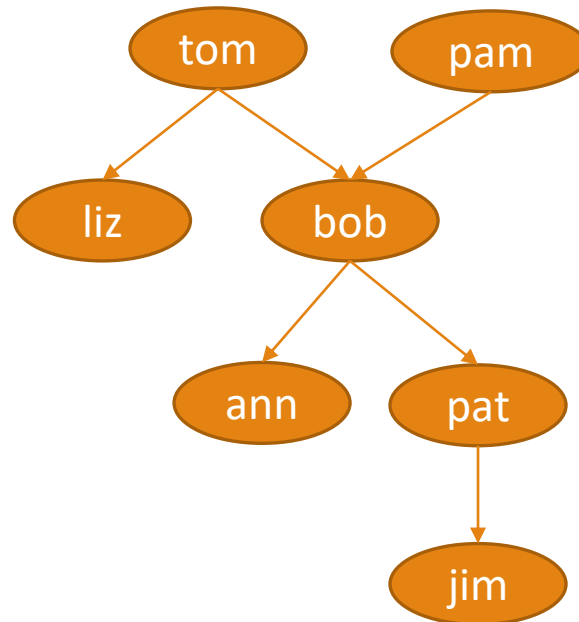
```
parent(tom,bob).  
parent(pam,bob).  
parent(tom,liz).  
parent(bob,ann).  
parent(bob,pat).  
parent(pat,jim).
```



```
14 ?- parent(bob,pat).  
true.  
  
14 ?- parent(liz,pat).  
false.  
  
15 ?- parent(X,liz).  
X = tom.  
  
16 ?- parent(bob,X).  
X = ann ;  
X = pat.  
  
17 ?- parent(X,Y).  
X = tom,  
Y = bob ;  
X = pam,  
Y = bob ;  
X = tom
```

Definisanje činjenica

```
parent(tom,bob).  
parent(pam,bob).  
parent(tom,liz).  
parent(bob,ann).  
parent(bob,pat).  
parent(pat,jim).
```



```
18 ?- parent(Y,jim),parent(X,Y).  
Y = pat,  
X = bob.  
  
19 ?- parent(tom,X),parent(X,Y).  
X = bob,  
Y = ann ;  
X = bob,  
Y = pat ;  
false.  
  
20 ?- parent(X,ann),parent(X,pat).  
X = bob.
```

Definisanje pravila

```
parent(tom,bob).  
parent(pam,bob).  
parent(tom,liz).  
parent(bob,ann).  
parent(bob,pat).  
parent(pat,jim).
```

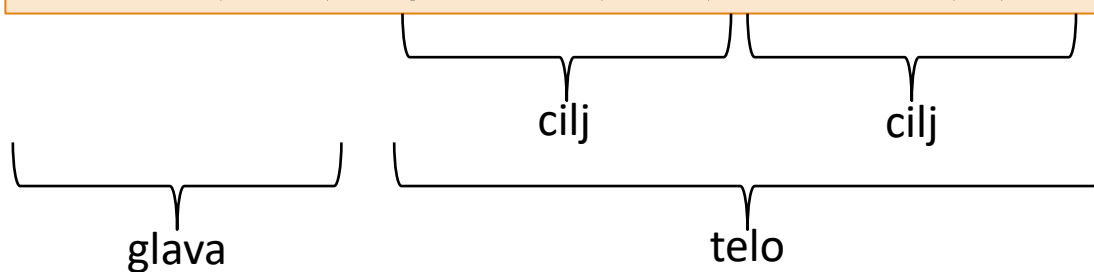
```
female(pam).  
male(tom).  
male(bob).  
female(liz).  
female(pat).  
female(ann).  
male(jim).
```

```
27 ?- mother(pam,bob).  
true.
```

X = pam i Y = bob

```
mother(pam,bob):-parent(pam,bob),  
                female(pam).
```

```
mother(X,Y):-parent(X,Y),female(X).
```



```
28 ?- mother(X,Y).  
X = pam,  
Y = bob ;  
X = pat,  
Y = jim.
```

Definisanje pravila

```
parent(tom,bob).  
parent(pam,bob).  
parent(tom,liz).  
parent(bob,ann).  
parent(bob,pat).  
parent(pat,jim).
```

```
female(pam).  
male(tom).  
male(bob).  
female(liz).  
female(pat).  
female(ann).  
male(jim).
```

```
mother(X,Y):-parent(X,Y),female(X).
```

```
grandparent(X,Z):-parent(X,Y),parent(Y,Z).
```

```
sister(X,Y):-parent(Z,X),parent(Z,Y),female(X).
```

```
sister(X,Y):-parent(Z,X),parent(Z,Y),female(X), X\=Y.
```

```
33 ?- sister(X,pat).  
X = ann ;  
X = pat ;  
false.
```

```
35 ?- sister(X,pat).  
X = ann ;  
false.
```