

Logičko i funkcijsko programiranje

II kolokvijum

2023/2024

Na *Desktop*-u u direktorijumu *Rad* kreirati direktorijum *Ime_Prezime_BrIndeksa_GodinaUpisa* i unutar njega sačuvati programe koji sadrže rešenja datih zadataka. Nazivi fajlova **moraju** biti oblika *Zadatak1.pl, Zadatak2.pl, Zadatak3.pl* i *Zadatak4.pl*.

1. (4 poena) Definisati prološki predikat **encode** koji enkodira datu listu tako što uzastopne pojave istog elementa *E* transformiše u listu $[N, E]$, gde *N* odgovara broju uzastopnih pojavljivanja i formira listu čiji su elementi ovakve liste.

```
?- encode([a,a,a,a,b,c,c,a,a,d,e,e,e,e],X).  
X = [[4,a],[1,b],[2,c],[2,a],[1,d],[4,e]]
```

2. Definisati prološki predikat

- a) (2 poena) **range** koja sadrži listu integera iz intervala $[M,N]$

```
?- range(4,9,L).  
L = [4,5,6,7,8,9]
```

- b) (3 poena) **rnd_select** koja iz date liste na slučajno bira zadati broj elementa.

```
?- rnd_select([a,b,c,d,e,f,g,h],3,L).  
L = [e,d,a]
```


- c) (1 poena) **lotto** koja formira listu od *N* različitih celih brojeva iz intervala $[1..M]$

```
?- lotto(6,49,L).  
L = [23,1,17,33,21,37]
```

3. (5 poena) Definisati prološki predikat **sudoku** koji rešava Sudoku problem dimenzije 2. Sudoku tabla se sastoji od 4 veća polja u formatu 2x2, pri čemu svako veće polje sadrži 4 manja polja u formatu 2x2. U manja polja se upisuju cifre 1, 2, 3 i 4, ali tako da u većem polju nema dve iste cifre, takođe kada se posmatra cela u jednoj vrsti, odnosno jednoj koloni ne smeju biti dve iste cifre. Sudoku tabla se zadaje navođenjem malih polja po vrstama velike table.

```
?- sudoku([[A,3,C,D],[1,F,G,H],[I,J,2,L],[M,N,O,4]]).
A = G, G = J, J = 4,
C = L, L = N, N = 1,
D = F, F = M, M = 2,
H = I, I = O, O = 3
```

A	3	C	D
1	F	G	H
I	J	2	L
N	N	O	4



4	3	1	2
1	2	4	3
3	4	2	1
2	1	3	4

Napomena: Dobijena rešenja prilikom poziva predikata u zadatku 2. i 3. ne moraju biti identična rešenjima iz primera, ali moraju zadovoljavati uslove zadatak.

4. (8 poena) U bazi znanja, činjenice o putevima između gradova su zabeležene na sledeći način:

road(GradA, GradB, Duzina)

Činjenica **road** predstavlja dužinu puta između grada GradA i GradB.

```
road(zuerich, baden, 22).
road(baden, basel, 83).
road(zuerich, rotkreuz, 39).
road(baden, olten, 43).
road(olten, bern, 67).
road(olten, luzern, 57).
road(olten, basel, 39).
road(rotkreuz, luzern, 18).
road(rotkreuz, schwyz, 26).
road(luzern, altdorf, 39).
road(schwyz, altdorf, 16).
```

Potrebno je:

a) Napisati predikat **route(GradA, GradB, Gradovi, Duzina)** koji određuju rutu Gradovi (niz gradova koji se redom obilaze) između gradova GradA i GradB čija je dužina Duzina.

```
?- route(zuerich, altdorf, Gradovi, Duzina).
Gradovi = [zuerich,baden,olten,luzern,altdorf],
Duzina = 161
```

b) Napisati predikat **bestRoute(GradA, GradB, Kriterijumi, Gradovi)** koji će naći putanju po zadatim kriterijumima između gradova GradA i GradB.

Kriterijumi mogu biti:

- **maxLen (X)** – maksimalna dužina segmenta rute
 - **exclude (Grad)** – grad Grad ne sme biti deo pronađene rute
- ?- bestRoute(zuerich, altdorf, [exclude(luzern)], Gradovi).
Gradovi = [zuerich,baden,olten,schwyz,altdorf].

Korisne ugrađene funkcije:

- Određivanje dužine liste – `length(X,L)`
- Generisanje slučajnog broja u intervalu $[0,N-1]$ – `random(N)`
- Generisanje permutacije date liste – `permutation(X,L)`
- Pripadnost listi – `member(X,L)`