

Navodila: Uporaba zapiskov, literature in elektronskih naprav ni dovoljena.

Čas: 60 min

Ustni izpiti: Četrtek, 6. julija ob 14h

1. [20%]

Dano je zaporedje prologovih stavkov o klicih.

```
called( eva, tomi, 117). % Eva je poklicala Tomija, klic je trajal 117 sekund.  
called( tomi, miha, 28). % Tomi je poklical Miho, klic je trajal 28 sekund.  
called( tina, eva, 636).  
called( eva, tomi, 218).  
called( miha, tomi, 15).  
called( tina, eva, 436).
```

- Podaj vse prologove odgovore na vprašanje:
?- called(X, Y, D), D > 120.
- Zapiši vprašanje za prolog, ki vrne, koliko časa so trajali vsi pogovori Eve (tako dohodni kot odhodni klici).
- Zapiši vprašanje za prolog, ki vrne osebi, ki sta opravili najdaljši klic.

2. [20%]

Urejanje z zlivanjem.

- Napiši predikat `merge(SL1, SL2, SL)`, ki dva že urejena seznama `SL1` in `SL2` zlije v urejen seznam `SL`. Predpostavi, da sta seznama `SL1` in `SL2` že urejena. Primer klica:

```
?- merge([2,3,5], [1,4,6], SL).  
    SL = [1,2,3,4,5,6].
```

Namig: Prvi element seznama `SL` je manjši izmed obeh prvih elementov seznama `SL1` in `SL2`.

- Napiši predikat `merge_sort(L, SL)`, ki uredi seznama `L` in vrne urejen seznam `SL`. Uporabi predikata `merge/3` in `divide/3`. Predikat `divide(L, L1, L2)` smo že napisali na vajah in razdeli seznam `L` na dva seznama, `L1` in `L2`, ki sta čimbolj enake dolžine. Primer uporabe predikata `divide/3`:

```
?- divide([2,1,3,4,5,6], L1, L2).  
    L1 = [2,3,5], L2 = [1,4,6].
```

Namig: Najprej je potrebno seznam `L` razdeliti na dva seznama (uporabi predikat `divide/3` z vaj), nato vsakega izmed njiju rekurzivno uredi ter ju zlij v en urejen seznam.

3. [20%]

Podana je naslednja gramatika.

```
ex --> dg.  
ex --> op, ['('], ex, [' '], ex, [')'].  
op --> [X], {member(X, [+ , *])}.  
dg --> [X], {member(X, [0,1,2,3,4,5,6,7,8,9])}.
```

a) Kaj odgovori prolog na naslednja vprašanja?

- 1) ?- ex([9], []).
- 2) ?- ex([+], []).
- 3) ?- ex([9, '(', +, '(', 3, ' ', 2, ')', ' ', 6, ')'], []).
- 4) ?- ex([*, '(', 2, ' ', +, '(', 5, ' ', 3, ')', ')'], []).

b) Dopolni gramatiko še z operacijama minus (-) in deljeno (/).

c) Je ta gramatika dvoumna? Na kratko utemelji odgovor.

d) Dodaj pomen osnovni gramatiki (brez dopolnitve v točki b) tako, da kot pomen dobiš izračunan rezultat: npr. stavek $*(2, +(5, 3))$ ima pomen 16.

4. [20%]

Podan je naslednji program v prologu s CLP(FD).

```
p(L):-  
    length(L, N),  
    L ins 1..N,  
    all_different(L),  
    labeling([], L).
```

a) Kaj prolog odgovori na naslednja vprašanja?

- ?- p([1,2,3,2,1]).
- ?- p([1,1,1,1,1]).
- ?- p([1,2,3,4,5]).

b) Naštejte vse prologove odgovore na spodnje vprašanje v pravilnem vrstnem redu!

- ?- p([A,B]).

c) Kaj računa dani program z eno besedo?

5. [20%]

a) Kako je formalno definirana parcialna pravilnost programa?

b) Pri dokazovanju pravilnosti programov uporabljamo pojem najšibkejšega predpogoja. Kako je definiran ta pojem kot relacija med tremi stvarmi.

c) Zakaj se pri dokazovanju pravilnosti izogibamo uporabi močnejših predpogojev, do kakšnih težav bi lahko vodila uporaba močnejših predpogojev?

d) Zapiši definicijo najšibkejšega predpogoja za programski stavek oblike:

```
if C then S1 else S2
```