

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

Čas: 70 minut

Ustni izpiti: četrtek, 27. 8. 2015 ob 13h

---

### 1. [25%]

Podan je naslednji predikat  $p/3$ .

$p([], L, [L])$ .

$p([H|T], PL, A) :-$

$p(T, PL, AP),$

$p(T, [H|PL], APH),$

$\text{conc}(AP, APH, A)$ .

- Kaj odgovori prolog na vprašanje:  
?-  $p([], [], La)$ .
- Kaj odgovori prolog na vprašanje:  
?-  $p([3], [], Lb)$ .
- Kaj odgovori prolog na spodnje vprašanje? Kako to imenujemo matematično?  
?-  $p([3, 2, 1], [], Lc)$ .
- Kaj bi prolog odgovoril na vprašanje  $p([2, 1], [3], Ld)$ , če dodamo rez (!) tik pred  $\text{conc}$ ? Kako rez (!) spremeni rešitev?
- Napiši predikat  $\text{subset}(L, S)$ , kjer seznam  $S$  predstavlja podmnožico elementov seznama  $L$ . Vrstni red elementov v  $S$  ni pomemben. Pri definiciji  $\text{subset}/2$  **obvezno** uporabi predikat  $p/3$ .

### 2. [25%]

- Napiši DCG z začetnim simbolom  $s$ , ki sprejema jezik vseh palindromov. Terminali so lahko poljubni simboli. Gramatiko definiraj **brez uporabe argumentov** (pomena).

Primeri stavkov, ki jih naj gramatika sprejme:

- [a, b, c, b, a]

- [1, 2, 2, 1]

- [dva, dva, dva]

- Napiši predikat  $\text{pal}(L, P)$ , ki dan seznam  $L$  razširi v palindrom  $P$ , tako da se seznam  $P$  začne z elementi iz  $L$  in konča z istimi elementi v obratnem vrstnem redu. Pri definiciji predikata **obvezno uporabi gramatiko iz prejšnje točke** (tudi, če je nisi rešil).

Primer klica predikata  $\text{pal}/2$ :

?-  $\text{pal}([a, b, c], P)$ .

$P = [a, b, c, b, a]$  ;

$P = [a, b, c, c, b, a]$ .

Rešitev *lahko* vrača tudi dodatne odgovore, ki niso popolnoma instancirani; npr.

$P = [a, b, c, X, Y, X, c, b, a]$ .

### 3. [25%]

Podana sta naslednja predikata.

```
c1(L):-
  \+ ( member(X, L),
        \+ member(X, [1,2,5,10,20,50,100,200])) .
c2([H], S):- !,
  { H = S }.
c2([H1,H2|T], S):-
  { H = H1 + H2 },
  c2([H|T], S).
```

a) Kaj prolog odgovori na naslednja vprašanja?

```
?- c1([5, 200, 5]).
?- c1([5, 20, 35, 50, 100]).
?- c1([20, X, 50, 5, 10]).
```

b) Kaj pa prolog odgovori na spodnja vprašanja?

```
?- c2([5, 200, 5], S).
?- c2([5, 20, 35, 50, 100], 210).
?- c2([20, X, 50, 5, 10], 210).
```

c) Predikat `c2/2` bi se seveda dalo na precej podoben način implementirati tudi brez `CLP(R)`, torej z uporabo prologove aritmetike. Kakšno konkretno prednost vidiš z uporabo `CLP(R)`, npr. glede na točko b)?

d) Z uporabo `CLP(FD)` dopolni predikat `c1/1`, da bo obenem še preverjal, da so vsi elementi v podanem seznamu `L` različni med seboj!

### 4. [25%]

a) Kot običajno v dokazovanju pravilnosti programov naj bo "komentirani" `while` stavek (annotated while statement) zapisan v sintaksi

```
INV while B do S
```

kjer je `INV` zanjča invariants. Zapiši, katere logične formule morajo biti resnične, da pogoj `INV` zagotavlja, da je po izvedbi gornje `while` zanke resničen izstopni pogoj `PostCond`. Odgovoriš lahko tudi v obliki definicije `wp( PreCond, INV while B do S, PostCond )` v prologu.

b) Zdaj pogledajmo vgnezdene `while` zanko:

```
INV1 while B1 do ( INV2 while B2 do S2 )
```

Formuliraj najšibkejši predpogoj za tako zanko. Torej, navedi eksplicitno vse logične relacije oz. formule, ki morajo veljati, da bo po izvedbi vgnezdene zanke veljal pogoj `PostCond1`.