

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

Čas: 75 minut

Ustni izpiti: torek, 12. julij ob 14:15

1. [20%]

Dano je zaporedje prologovih stavkov o nakupih, npr.:

bought(eva, [bread/1, butter/0.25]). % Eva je kupila 1 kg kruha in 0.25 kg masla

bought(tomi, [beer/6]). % Tomi je kupil 6 litrov piva

bought(tim, [beer/1, beer/2]). % Tim je kupil 3 litre piva, ki so se
% mu na računu pojavili kot dve postavki

bought(tina, [salami/0.5, beer/2, coffee/0.5]).

bought(tina, [apples/2]). % Ob naslednjem nakupu je Tina kupila 2 kg jabolk

price(beer, 2.00). % En liter piva stane 2.00 EUR

price(butter, 6.00).

price(bread, 2.80).

price(salami, 16.00).

price(coffee, 12.00).

price(apples, 1.20).

a) Za gornjo bazo nakupov podaj vse prologove odgovore na vprašanje:

?- bought(X, [_, _ | _]).

b) Zapiši vprašanje za prolog, ki poišče z vračanjem vsako osebo X, ki je v enem nakupu kupila vsaj dva dejansko različna artikla (lahko seveda v različnih količinah).

c) Definiraj predikat

total(Product, Amount, Value)

ki pomeni: Amount je celotna prodana količina proizvoda Product v bazi nakupov, Value pa je celotna vrednost prodane količine proizvoda. Npr. za gornjo bazo:

?- total(beer, A, V).

A = 11, V = 22. % 11 litrov piva v skupni vrednosti 22 EUR

2. [20%]

a) Definiraj v prologu predikat all_equal(L), ki uspe, če so v seznamu L vsi elementi enaki. Primera klicev tega predikata:

?- all_equal([a, a, a, a]).

true.

?- all_equal([a, X, Y, a, a]).

X = a, Y = a.

b) Definiraj predikat p(L, SL), ki vrne najdaljše zaporedje enakih elementov (to je seznam SL) v seznamu L. Če je takih več, naj predikat p/2 vrne vse (glej primer).

Namig: Uporabi predikata all_equal(L) in sublist(List, SubList) ter zastavi definicijo predikata z uporabo negacije, torej da ne obstaja drugo zaporedje, ki bi bilo daljše kot iskano zaporedje. Predikat sublist(L, SL) smo spoznali na vajah in vrne vse možne podsezname seznama L, enega po enega. Primer klica predikata p:

?- p([a,b,c,c,c,b,b,a,a,a], SL).

SL = [c,c,c];

SL = [a,a,a].

3. [20%]

Zapiši pomen v logiki spodnjih stavkov:

- (a) Mary likes all animals but snakes.
- (b) Peter and a friend of his know all good restaurants in Piran.
- (c) A lady from the information office only helps tourists that travel with a dog.

Prevod gornjih stavkov v slovenščino:

Mary ima rada vse živali razen kač.

Peter in njegov prijatelj poznata vse dobre restavracije v Piranu.

Gospa iz informacijske pisarne pomaga le turistom, ki potujejo s psom.

4. [20%]

Dana je DCG gramatika, ki opisuje vertikalno gibanje enostavnega robota v smeri navzgor ali navzdol:

```
move --> step.  
move --> step, move.  
step --> [short, up].  
step --> [long, up].  
step --> [short, down].  
step --> [long, down].
```

Kaj odgovori ta gramatika na vprašanja a in b (podaj vse možne odgovore):

- a) ?- move([short, up, L, down], []).
- b) ?- move([long, X, Y], []).
- c) Gramatiko dopolni z atributom D tako, da bo izračunala D kot razdaljo med začetno in končno točko robotovega giba (move). Korak vrste »short« povzroči premik za eno enoto, korak vrste »long« pa za 2. Tako korak [long, down] povzroči premik -2. Npr.

```
?- move(D, [long, up, short, down, long, up], []).  
D = 3.
```

- d) Dopolni gramatiko iz vprašanja c tako, da bo dovoljevala tudi zanke v programih za robota. Npr.:

```
?- move(D, [long, up, repeat, long, up, short, down, long, up, three, times,  
short, up], []).  
D = 12.
```

Gramatika naj dopušča zanke z največ pet ponovitvami.

Namig: Dodaj pravilo za stavčno frazo "step", tako da bo veljalo:

```
?- step(D, [repeat, long, up, short, down, four, times], []).  
D = 4.
```

```
?- step(D, [repeat, long, up, repeat, short, down, five, times, two, times], []).  
% Vgnezdena zanka: Dvakrat ponovi "long up" in vgnezdeno zanko  
% "petkrat ponovi short down"  
D = -6.
```

5. [20%]

Naslednja vprašanja se nanašajo na matematično dokazovanje pravilnosti programov.

- a) Kako so definirani pojmi:
 - parcialna pravilnost programa;
 - iztekanje programa;
 - totalna pravilnost programa?
- b) Najšibkejše predpogoje za tipe stavkov programskega jezika smo definirali s predikatom wp/3. Pojasni, kaj so trije argumenti C1, S in C2 tega predikata. Kaj pomeni wp(C1, S, C2)?
- c) Zapiši definicijo v prologu predikata wp/3 za stavek **while**.