

Programiranje I: 1. izpit

28. januar 2020

Čas reševanja je 150 minut. Veliko uspeha!

1. naloga

a) Napišite funkcijo, ki sešteje `int option` argumenta. Funkcija vrne vsoto argumentov, če oba argumenta vsebujeta število, in `None` sicer.

```
option_sum: int option -> int option -> int option
```

b) Napišite funkcijo

```
twostep_map: ('a -> 'b * 'c) -> ('b -> 'd) -> ('c -> 'e) -> 'a -> 'd * 'e
```

ki argument tipa `'a` najprej preslika s prvo funkcijo, nato pa na komponentah ustrezno uporabi drugo in tretjo funkcijo.

```
# twostep_map (fun x -> (x, x)) ((+)1) ((-)2) 3;;  
- : int * int = (4, -1)
```

c) Definirajte funkcijo, ki sprejme funkcijo `f` in seznam `xs` ter vrne nov seznam, kjer se vsak element `x` seznama `xs` ponovi `(f x)`-krat. Nepozitivno število ponovitev pomeni, da elementa ne vključimo v končni seznam. Za vse točke naj bo funkcija repno rekurzivna, kar tudi argumentirajte v komentarju.

```
function_repeat: ('a -> int) -> 'a list -> 'a list
```

d) Definirajte funkcijo `iterate`, ki sprejme funkcijo `f`, zaustavitveni pogoj in začetno vrednost. Nato funkcijo `f` zaporedoma uporablja, dokler za rezultat ne velja zaustavitveni pogoj, in vrne prvi rezultat, ki zadošča zaustavitvenemu pogoju.

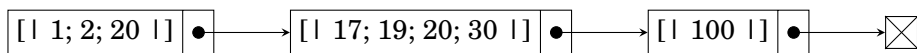
```
iterate: ('a -> 'a) -> ('a -> bool) -> 'a -> 'a
```

e) (za čast in slavo) Funkcijo `iterate` napišite brez uporabe `let rec` in zanke `while` (tudi v morebitnih pomožnih funkcijah).

2. naloga

Napreden seznam je podoben vgrajenemu seznamu v OCaml-u, le da v vozliščih namesto vrednosti hrani tabelo vrednosti (velikosti tabel niso nujno enake). Tako kot običajen seznam je sestavljen iz dveh različnih gradnikov: praznega seznama in vozlišča, ki vsebuje tabelo in preostanek naprednega seznama.

a) Definirajte polimorfen tip `'a improved_list` ter seznam `test : int improved_list`, ki predstavlja spodnji napredni seznam:



b) Definirajte funkcijo `count: 'a improved_list -> int`, ki vrne število vseh elementov v podanem seznamu.

c) Definirajte funkcijo, ki vrne i -ti element, če ga seznam vsebuje.

```
nth: int -> 'a improved_list -> 'a option
```

d) Definirajte funkcijo, ki preveri ali je napreden seznam urejen (predpostavimo, da vsebuje elemente, ki jih lahko primerjamo z $<$). Za vse točke mora funkcija imeti linearno časovno zahtevnost.

```
is_sorted: 'a improved_list -> bool
```

e) Napišite funkcijo `update: 'a improved_list -> int -> 'a -> 'a improved_list`, ki vrne nov napreden seznam, kjer vrednost na indeksu (drugi argument) nadomesti s podano vrednostjo (tretji argument). Pazite, da pri tem začetni seznam ostane nespremenjen.

```
# nth (update test 5 (-3)) 5;;  
- : int option = Some (-3)  
# nth test 5;;  
- : int option = Some 20
```

3. naloga

Nalogo lahko rešujete v Pythonu ali OCamlu.

a) Mama Franca želijo na balkon širine n postaviti m korit z nageljni širine l (korit, ne nageljnov). Zaradi lažjega zalivanja mora biti med dvema koritoma vsaj za 1 enoto prostora. Mama Franca želijo postaviti vsa korita, jih pa zaradi slabega vida med seboj ne razlikujejo. Primer vseh štirih možnih postavitev pri balkonu širine 9 s tremi koriti širine 2:



Sestavite funkcijo, ki sprejme n , m in l ter vrne število vseh različnih postavitev korit za rože na balkon.

b) Mama Franca so na razprodaji nabrali korita različnih širin. Sestavite funkcijo, ki sprejme širino balkona n in seznam celih števil, ki predstavljajo širine korit, in vrne število različnih postavitev. Pri tem je vrstni red korit določen z vrstnim redom širin v podanem seznamu.