

Programiranje I: 2. Izpit

10. februar 2020

Čas reševanja je 150 minut. Veliko uspeha!

1. naloga

a) Napišite funkcijo `dot_prod: float * float * float -> float * float * float -> float`, ki sprejme dva vektorja iz \mathbb{R}^3 in vrne njun skalarni produkt.

b) Napišite funkcijo `curry_second : ('a -> 'b -> 'c) -> 'b -> 'a -> 'c`. Ki sprejme funkcijo dveh argumentov in vrednost drugega argumenta ter vrne prvotno funkcijo, kjer

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

c) Napišite funkcijo `combine_and_filter : ('a -> 'b -> 'c option) -> 'a list -> 'b list -> 'c list`, ki podano funkcijo aplicira na posameznih elementih podanih seznamov in iz rezultatov zgradi nov seznam, pri čemer zavrže vse rezultate, ki so `None` in konča, ko se izteče prvi katerikoli od podanih seznamov. Za vse točke mora biti repno rekurzivna in imeti linearno časovno zahtevnost.

Primer

d) Napišite funkcijo `conditional_print : (string -> bool) -> string list -> unit`, ki sprejme predikat in seznam nizov. Funkcija naj izpiše z vejico ločene elemente, za katere predikat velja. Funkcija naj bo repno rekurzivna in ima linearno časovno zahtevnost. Pazite, da bo tip funkcije pravilen.

Primer

2. naloga

AB drevesa definiramo s tipom

```
type ('a, 'b) tree =  
  | Empty  
  | ANode of ('a, 'b) tree * 'a * ('a, 'b) tree  
  | BNode of ('a, 'b) tree * 'b * ('a, 'b) tree
```

a) Definirajte AB drevo `test : (int, bool) tree`, ki predstavlja spodnje drevo.
TODO SLIKICA PROSIM :)

b) Definirajte funkciji `adepth: ('a, 'b) tree -> int`, ki vrne globino najglobljega A vozlišča in funkcijo `bdepth`, ki vrne globino najglobljega B vozlišča.

c) Definirajte funkcijo `count: ('a, 'b) tree -> result`, ki prešteje število posameznih vozlišč v AB drevesu.

```
# count test;;  
- : result = {aNode= 2; bNode = 2}
```

d) Definirajte funkcijo `is_typemirror : ('a, 'b) tree -> ('b, 'a) tree -> bool`, ki preveri, ali je prvo drevo enako drugemu, kjer v drugem drevesu A in B vozlišča med seboj zamenjamo.

e) Definirajte funkcijo `foldmap fa fb acc tr`, kjer imajo argumenti tipe: `fa: 'a -> 'b -> 'a * 'c`, `fb: 'a -> 'd -> 'a * 'e`, `acc: 'a in tr: ('b, 'd) tree`, ki vrne rezultat tipa `'a * ('c * 'e) tree`. Funkcija se sprehodi po podanem drevesu in vsako vozlišče pretvori z ustrezno funkcijo, hkrati pa pridobljene rezultate zлага kot običajna funkcija `fold`. Vrstni red sprehoda po drevesu je poljuben, v komentarju pa argumentirajte, zakaj doseže vsa vozlišča.

3. naloga

Napišite funkcijo `f(k, n)`, ki vrne število vseh zaporedij naravnih števil (naravna števila vsebujejo 0) dolžine n , ki se začnejo z 0 in je razlika med zaporednima členoma manjša ali enaka k .