

Programiranje I: 2. izpit

6. julij 2018

Čas reševanja je 150 minut. Veliko uspeha!

1. naloga

a) Napišite funkcijo

```
uporabi : ('a -> 'b) -> 'a -> 'b
```

ki uporabi podano funkcijo na podanem argumentu.

b) Napišite funkcijo

```
ibaropu : 'a -> ('a -> 'b) -> 'b
```

ki izvede obrnjeno uporabo. Naprimer, `ibaropu (ibaropu x f)` g je enakovredno $g(f(x))$.

c) Napišite funkcijo

```
zacetnih : int -> 'a list -> ('a list) option
```

za katero zacetnih `n` `xs` vrne začetnih `n` elementov seznama `xs`, oziroma `None` v primeru, ko ima seznam `xs` manj kot `n` elementov. Za vse točke naj bo funkcija *repno-rekurzivna*.

2. naloga

Neprazne seznane lahko v OCamlu predstavimo s tipom

```
type 'a neprazen_sez = Konec of 'a | Sestavljen of 'a * 'a neprazen_sez
```

a) Napišite funkciji `prvi : 'a neprazen_sez -> 'a` in `zadnji : 'a neprazen_sez -> 'a`, ki vrneta prvi in zadnji element nepraznega seznama.

b) Napišite funkcijo `dolzina : 'a neprazen_sez -> int`, ki izračuna dolžino nepraznega seznama.

c) Napišite funkcijo `pretvori_v_seznam : 'a neprazen_sez -> 'a list`, ki pretvori neprazen seznam tipa `'a neprazen_sez` v navaden seznam tipa `'a list`.

d) Napišite funkcijo `zlozi : ('b -> 'a -> 'b) -> 'b -> 'a neprazen_sez -> 'b`, ki zloži podano funkcijo preko nepraznega seznama. Delovanje funkcije naj bo podobno delovanju funkcije `List.fold_left`.

3. naloga

Dr. Ana Kek je matematičarka, ki se navdušuje nad simetrijo. Po vrnitvi z izleta v Hajjah v Jemnu je začela iskati simetrije v črtastih črno-belih zapestnicah, ki jih je kupila na potovanju.

Zato si želi razdeliti zaporedje črnih in belih črt na zapestnicah na simetrične dele. Njen cilj je poiskati delitev z najmanjšim številom delov. Da si delo olajša, se je odločila zaporedje črt predstaviti z nizom ničel in enic.

a) Napišite funkcijo `simetricen`, ki preveri ali je nek del simetričen, torej palindrom.

```
# Primer:
>>> simetricen("01010")
True
```

b) Napišite funkcijo `stevilo_delov`, ki izračuna na najmanj koliko delov moramo razdeliti zaporedje, da so vsi deli simetrični.

```
# Primer:
>>> stevilo_delov("00101011")
3
```

c) Napišite funkcijo `razdeli`, ki vrne delitev, kjer zaporedje razdelimo na najmanjše možno število simetričnih delov. Če je takšnih delitev več, naj funkcija vrne poljubno izmed njih.

```
# Primer:
>>> razdeli("00101011")
["0", "01010", "11"]
```

d) Poleg simetričnih pa se dr. Ana Kek zanima tudi za *vsotno-simetrične* dele. To so tisti deli D dolžine n , pri katerih je vsota prvih $\lfloor n/2 \rfloor$ števk enaka vsoti preostalih števk. Napišite funkcijo `vsotno_simetricen`, ki preveri ali je del vsotno-simetričen.

```
# Primer:
>>> vsotno_simetricen("01001000")
True
>>> vsotno_simetricen("1011")
False
```

Namig: iz niza števk `b` lahko v Pythonu naredite seznam števil `[int(c) for c in b]`, v OCamlu isto dosežete tako, da naložite modul "Str" z ukazom `#load "str.cma" ;;` in uporabite

```
List.map int_of_string (Str.split (Str.regexp "") b)
```

e) Za primer, da se bo dr. Ana Kek kdaj začela zanimati tudi za druge vrste simetrij, funkciji `stevilo_delov` in `razdeli` napišite tako, da za drugi argument sprejemo funkcijo, ki preverja ali je nek del simetričen.

```
# Primer:
>>> razdeli("00101011", simetricen)
["0", "01010", "11"]
>>> razdeli("00101011", vsotno_simetricen)
["00", "101011"]
>>> razdeli("00101011", simetricen)
["0", "01010", "11"]
```