Matematička logika u računarstvu: Drugi praktični kolokvij (Haskell)

PMF MO, ak. god. 2013./2014.

2. lipnja, 2014.

Napomena: Trajanje kolokvija je 120 minuta. Rješenja zadataka pošaljite na ***jan.snajder@fer.hr****. Kolokvij nosi 20 bodova, od kojih je za prolaznu ocjenu potrebno ostvariti barem 10 bodova.*

1. *(2 boda)* Definirajte funkciju `wordHamming w1 w2` koja prebrojava u koliko se znakova razlikuju stringovi `w1` i `w2`.

   ```
   wordHamming "riba" "pita" ⇒ 2
   wordHamming "voda" "vatra" ⇒ 4
   ```

   Definirajte dvije varijante ove funkcije:

   (a) Rekurzivna definicija (`wordHamming1`);

   (b) Definicija pomoću ugrađenih funkcija višega reda (`wordHamming2`).

2. *(3 boda)* Napišite funkciju `vigenere key s` koja string `s` šifrira Vigenèreovom šifrom s ključem (stringom) `k`. Pretpostavite da je ključ sastavljen od malih slova `a-z`. Prije šifriranja poruke, poruku je potrebno pretvoriti u mala slova i iz nje izbaciti sve znakove koji nisu alfabetski, uključivo razmak (koristite funkciju `Data.Char.isAlpha`). Za pretvorbu znaka u ASCII-kôd i obrnuto koristite funkciju `chr` odnosno `ord`. Od koristi vam može biti funkcija `cycle`.

   ```
   vigenere "az" "abba" ⇒ "aabz"
   vigenere "charlie" "it takes a revolution to make a solution"
   ⇒ "katrvmwcyemztyvpoeewqcrerdwpwaify"
   ```

3. *(3 boda)*

   (a) Napišite rekurzivnu definiciju funkcije `nubRight` za izbacivanje duplikata iz liste. Duplikatom se smatraju svi elementi koji se pojavljuju nadesno od prvog pojavljivanja elementa.

   ```
   nubRight :: Eq a => [a] -> [a]
   nubRight "kikiriki" ⇒ "kir"
   ```

   (b) Napišite rekurzivnu definiciju za `nubLeft`:

   ```
   nubLeft :: Eq a => [a] -> [a]
   nubLeft "kikiriki" ⇒ "rki"
   ```

4. *(4 boda)* Napišite rekurzivnu definiciju funkcije za zbrajanje elemenata na parnim pozicijama u listi:

```
sumEven :: Num a => [a] -> a
sumEven [1,4,3,5,7] ⇒ 9
```

Definirajte sljedeće verzije ove funkije:

(a) Rekurzivna izvedba (`sumEven1`);
(b) Izvedba s akumulatorskim parom (`sumEven2`);
(c) Izvedba s funkcijama višega reda (`sumEven3`).

Objasnite:

(a) Vremenske i prostorne složenosti ovih funkcija.
(b) Koja se od ovih funkcija može koristiti nad beskonačnom listom i zašto.

5. *(4 boda)* Binarno stablo koje pohranjuje cijele brojeve u listovima možemo definirati na sljedeći način:

```
data IntTree1 = Leaf1 Int | Node1 IntTree1 IntTree1
```

Definirajte sljedeće funkcije nad ovakvom strukturom.

(a) `depth :: IntTree1 -> Int`
Funkcija koja izračunava dubinu stabla. Dubina stabla s jednim čvorom je 1.
(b) `findDeepest :: (Int -> Bool) -> IntTree1 -> [Int]`
Funkcija vraća najdublji element (cijeli broj pohranjen u listu stabla) koji zadovoljava zadani predikat. Ako postoji više takvih elemenata, funkcija vraća onaj najlijeviji. Ako ne postoji element koji zadovoljava zadani predikat, funkcija vraća praznu listu.
(c) `level :: Int -> IntTree1 -> [Int]`
Funkcija vraća listu brojeva koji se u stablu nalaze na zadanoj dubini.
(d) Neka je definirana struktura stabla koja pohrajuje vrijednosti i u listovima i u čvorovima:
`data IntTree2 = Leaf2 Int | Node2 Int IntTree2 IntTree2`
Definirajte funkciju koja pretvara `IntTree1` u `IntTree2`:
`pruneTree :: Int -> IntTree1 -> IntTree2`
na način da svaki čvor `n` pohranjuje vrijednost koja je jednaka zbroju vrijednosti svih čvorova u podstablu ukorijenjenom u `n`, a dodatno se svi čvorovi koji su ispod zadane razine odbacuju (međutim njihove se vrijednosti uzimaju u obzir pri zbrajanju).

6. *(4 boda)* Proučite podatkovnu strukturu `Data.Map` iz istoimenog modula. Binarna relacija nad cijelim brojevima može se definirati na sljedeći način:

```
type BinRel = Data.Map Int [Int]
```

(a) Definirajte funkciju
`symmetricize :: BinRel -> BinRel`
koja izračunava simetrično zatvaranje zadane binarne relacije.
(b) Definirajte funkciju
`isTransitive :: BinRel -> Bool`
koja provjerava je li zadana relacija tranzitivna.