ITT9080 Funktsionaalprogrammeerimine
Koduülesanded

Lahenduste esitamise tähtaeg on 8.6.2004. Ülesannete lahendamine on eksamihinde saamiseks nõutav. Lahenduste kopeerimine ei ole lubatud, aga ülesannete ühine arutamine on aktsepteeritav. Küsimused on teretulnud meiliaadressil `tarmo@cs.ioc.ee`, samuti olen heal meelel nõus andma konsultatsiooni KübIs.

1. Defineeri funktsioon `fibonacci`, mis genereerib kõigi Fibonacci arvude listi.

   (Raskem teha efektiivselt:) Defineeri funktsioon `hamming`, mis genereerib kõigi Hammingi arvude listi. Hammingi arvud on positiivsed täisarvud kujul $2^i \cdot 3^j \cdot 5^k$, kus i, j, k on naturaalarvud.

   Näpunäide: Kasuta funktsioone `merge`, mis sorteerib kokku kaks eelnevalt sorteeritud listi ning funktsiooni `scale`, mis korrutab etteantud arvulisti kõiki elemente etteantud arvuga.

2. Defineeri funktsioon `vars`, mis listina etteantud hulgast genereerib etteantud arvu elementidega variatsioonid. Nt

   ```
   vars [1..3] 2 --> [[1,2], [1,3], [2,1], [2,3], [3,1], [3,2]]
   ```

3. Defineeri funktsioon `applyEach`, mis etteantud listi funktsioone rakendab igaühe etteantud väärtusele. Nt

   ```
   applyEach [(+3), (*2), const 7] 6 --> [9, 12, 7]
   ```

   Defineeri funktsioon `applyAll`, mis listist funktsioonidest $[f_1, \ldots, f_n]$ ja väärtusest v rehkendab väärtuse $f_1\,(\ldots\,(f_n\,v)\ldots)$. Nt

   ```
   applyAll [(+3), (*2), const 7] 6 --> 17
   ```

4. Defineeri binaarpuude andmetüüp (andmed paiknevad hargnemistippudes). Kirjuta funktsioon `sorted`, mis kontrollib, kas etteantud binaarpuu (üle järjestatud tüübi) on otsingupuu (st üheski hargnemistipus olev väärtus pole väiksem vasaku vahetu alampuu maksimumist ega suurem parema vahetu alampuu miinimumist).

   Kirjuta funktsioon `avl`, mis kontrollib, kas etteantud binaarpuu on AVL-puu. AVL-puu on otsingupuu, milles iga kahe naaberalampuu kõrguste vahe on ülimalt 1.

   Kirjuta funktsioon `find`, mis kontrollib, kas etteantud element leidub etteantud otsingupuus. Kirjuta funktsioonid `insert` ja `delete` elemendi lisamiseks otsingupuusse, kustutamiseks sealt.

   (Raskem:) Kirjuta funktsioonid `insertAVL` ja `deleteAVL` elemendi lisamiseks AVL puusse, kustutamiseks sealt.

5. Disaini While-keelele väike laiendus pointeritega. Näitena võiksid uuteks käskudeks olla kuhjamäluruumi allokeerimine-initsialiseerimine, muteerimine ja deallokeerimine ning uueks avaldiseks derefereerimine. Defineeri Haskellis laienduse abstraktne süntaks ja kirjuta interpretaator.

6. Kirjuta regulaator simbotile, mis tsükliliselt külastaks etteantud nimekirja punkte (eeldusel, et need ei ole okupeeritud kellegi teise simboti või mõne eseme poolt) ning igaühes emiteeriks rõõmuhüüde.

   Idee võiks olla alati püüda liikuda parajasti eesmärgiks oleva punkti suunas, kuid kui ees on takistus, siis seda mingil moel vältida, nt hoida vasakule, ja sattunud liialt lähedale takistusele (nii et pööramine juba võimatu), tagurdada lühidalt.