Nimi: 12. oktoober 2024. a.
Rühm:

Kontrolltöö 1 (näidis)

1 Substitutsioon (3/31)

Olgu antud makrodefinitsioonid: $I \equiv (\lambda x.\ x)$ ja $K \equiv (\lambda x\ y.\ x)$.

Teosta järgnev substitutsioon:
$((\lambda y.\ y\ x)(K\ x))[x \to y]$

2 Reduktsioon (6/31)

Olgu antud makrodefinitsioonid: $I \equiv (\lambda x.\ x)$ ja $K \equiv (\lambda x\ y.\ x)$.

Väärtusta järgnev term normaalkujule kasutades aplikatiivjärjekorda. Mitme β-reduktsiooni sammu pidite tegema?

$(\lambda x.\ K\ x\ x\ y)\ ((\lambda y.\ I\ y)\ x)$

3 Andmestruktuurid (5/31)

Olgu antud makrodefinitsioonid:

$$(E_1, E_2) \equiv (\lambda s.\ s\ E_1\ E_2) \qquad \mathsf{false} \equiv (\lambda t\ e.\ e)$$
$$\mathsf{snd} \equiv (\lambda p.\ p\ \mathsf{false}) \qquad \mathsf{true} \equiv (\lambda t\ e.\ t)$$
$$\mathsf{fst} \equiv (\lambda p.\ p\ \mathsf{true})$$

Mis on järgneva termi väärtus?

$\mathsf{snd}\ (\mathsf{true}, \mathsf{false})\ K\ I\ x\ y$

4 Fold (6/31)

Kirjuta funkstsioon yl4, mis otsib argumentlistis paari, kus paari esimene komponent on 1 ja teine False. Kui selline paar leidub, tagastatakse Just konstruktori all vähim sellise elemendi indeks – mitmes element listis on sellisel kujul. Kui sellist paari ei leidu, tuleb tagastada Nothing.

```
yl4 : List (Int, Bool) → Maybe Nat
yl4 = foldr f ?yl4_a
  where f : ?yl4_ty
        f = ?yl4_f
```

Näited:

- yl4 [] == Nothing
- yl4 [(1,False)] == Just 0
- yl4 [(1, True)] == Nothing
- yl4 [(0, False)] == Nothing
- yl4 [(1, True),(0,True),(0,False),(1,False),(1,False)] == Just 3

5 Liidesed (5/31)

Kirjuta liides F, mis sisaldab funktsiooni f, ja F-i instantsid nii, et järgmised võrdused kehtiksid.

- f True True == False
- f False False == True
- f [5] [6] == [5,6]

6 Puud (6/31)

Vaata puude definitsiooni ja funktsioonikutse näiteid. Kirjuta find_all-ile tüüp ja implementeeri funktsioon find_all x p nii, et see leiaks paaride puust p listina kõik paaride teised komponendid, kus esimene komponent on võrdne x-ga.

```
data Tree a = LeafJust a | LeafNothing | Branch (Tree a) (Tree a)

test_tree : Tree (Char, Int)
test_tree =
    Branch (LeafJust ('x', 1))
           (Branch (Branch LeafNothing (LeafJust ('y', 2))) (LeafJust ('x',10)))

find_all : ?rhs_find_all_type
find_all = ?rhs_find_all
```

Näited:

- find_all 'x' test_tree == [1, 10]
- find_all 'y' test_tree == [2]
- find_all 'z' test_tree == []