

Nimi:
Rühm:

12. oktoober 2024. a.

Kontrolltöö 1 (näidis)

1 Substitutsioon (3/31)

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

Teosta järgnev substitutsioon:

$((\lambda y. y x)(K x))[x \rightarrow y]$

Lahendus: $(\lambda z. z y)(K y)$

2 Reduktsioon (6/31)

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

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

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

Lahendus:

$$\begin{aligned} \dots &\implies (\lambda x. (\lambda y. x) x y) ((\lambda y. (\lambda x. x) y) x) \\ &\implies (\lambda x. x y) ((\lambda y. (\lambda x. x) y) x) \\ &\implies (\lambda x. x y) ((\lambda y. y) x) \\ &\implies (\lambda x. x y) x \\ &\implies x y \end{aligned}$$

Vastus: 5 sammu

3 Andmestruktuurid (5/31)

Olgu antud makrodefiniitsioonid:

$(E_1, E_2) \equiv (\lambda s. s E_1 E_2)$

$\text{snd} \equiv (\lambda p. p \text{ false})$

$\text{fst} \equiv (\lambda p. p \text{ true})$

$\text{false} \equiv (\lambda t e. e)$

$\text{true} \equiv (\lambda t e. t)$

Mis on järgneva termi väärtus?

`snd (true, false) K I x y`

Vastus: `x y`

4 Fold (6/31)

Kirjuta funktsioon `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`

Vastus:

```
yl4 : List (Int, Bool) → Maybe Nat
yl4 = foldr f Nothing
  where f : (Int, Bool) → Maybe Nat → Maybe Nat
        f (1, False) xs = Just 0
        f (x, y) Nothing = Nothing
        f (x, y) (Just z) = (Just (z+1))
```

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]`

Vastus:

```
interface F a where
```

```
  f : a → a → a
```

```
F Bool where
```

```
f x y = not (x&& y)
```

```
F (List a) where
```

```
f xs ys = xs ++ ys
```

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 == []`

Vastus:

```
find_all : Eq a => a → Tree (a,b) → List b
```

```
find_all x (LeafJust (y, z)) = if (x==y) then [z] else []
```

```
find_all x LeafNothing = []
```

```
find_all x (Branch y z) = find_all x y ++ find_all x z
```