

Nimi:  
Rühm:

2. detsember 2024. a.

## Kontrolltöö 2 (näidis)

Tüübituletuse ülesande lahendus *tuleb esitada paberil*. Ülesanded 2 kuni 5 esitada Moodlesse.

### 1 Tüübituletus (10/31)

Kasuta tüübituletusalgoritmi, et proovida tuletada järgneva avaldise printsiipaalne tüüp. Selleks, a) annotaeri term, b) ehita tüübituletuse puu, c) kirjuta välja kõik kitsendused ja d) lahenda kitsenduste süsteem.

Lihtsustused:

- ei pea kirjutama " $x^\alpha \in \Gamma$ "-sid
- kitsendusi ei pea kirjutama puu sisse (mis oli slaididel roheline)
- kitsenduste lahendamise vahesamme ei pea välja kirjutama.

Term:  $(\lambda x. (\lambda y. (y(\lambda z. z))))$

## 2 Sisend-väljund (6/31)

Kirjuta protseduur `yl2`  $n$   $m$ , mis genereerib  $n$  juhuarvu nullist  $m$ -ni ja tagastab genereeritud arvude aritmeetilise keskmise.

```
yl2 : Nat → Int32 → IO Double
yl2 = ?yl2_rhs
```

Näited:

```
Main> :exec yl2 1 30 >>= println
12.0
Main> :exec yl2 1 30 >>= println
8.0
Main> :exec yl2 10000 30 >>= println
14.9808
Main> :exec yl2 10000 30 >>= println
15.0275
```

## 3 Sõltuvad tüübid (5 / 31)

Etteantud kood:

```
half : Nat → Nat
half 0 = 0
half 1 = 0
half (S (S k)) = 1 + half k
```

Kirjuta täielik funktsioon `pooledElem` : `Vect n a` → `Vect (half n) a` nii et testid (all) töötaksid.

total

```
pooledElem : Vect n a → Vect (half n) a
pooledElem = ?pooledElem_rhs
```

Testid:

- `pooledElem [1,2,3,4,5,6,7] == [1, 3, 5]`
- `pooledElem [1,2,3,4,5,6,7,8] == [1, 3, 5, 7]`
- `pooledElem [1,2,3,4,5,6,7,8,9] == [1, 3, 5, 7]`

## 4 Tõestamine (5 / 31)

Etteantud kood:

```

infixl 11 \\/
data (\\/) : Type → Type → Type where
  Disjll : a

```

```

-----
→ a \\/ b

```

```

Disjlr : b
-----
→ a \\/ b

```

```

DisjE : a\\b → (a→ c) → (b→ c)
-----
→ c

```

```

DisjE ( Disjll x) q w = q x
DisjE ( Disjlr x) q w = w x

```

Ülesanne: Tõesta idrises järgnev väide.

total

```

ex4 : (a → Void) /\ (b → a)
-----
→ (b → c)
ex4 = ?ex4_rhs

```

```

infixl 10 /\
data (/\\) : Type → Type → Type where
  Conl : a → b

```

```

-----
→ a /\ b

```

```

VoidE : Void
-----
→ b

```

```

VoidE q impossible

```

## 5 Tõestamine rewrite-ga (5 / 31)

Etteantud kood:

```

liida : Nat → Nat → Nat
liida 0 y = y
liida (S k) y = S (liida k y)

```

```

mymap : (a → b) → List a → List b
mymap f [] = []
mymap f (x :: xs) = f x :: mymap f xs

```

Ülesanne: Tõesta idrises järgnev võrdus.

total

```

mymapLiida0 : (xs: List Nat)
-----
→ mymap (liida 0) xs = xs
mymapLiida0 = ?mymapLiida0_rhs

```