

Püsipunktid

- Termi M nimetatakse *püsipunktikombinaatoriks* kui

$$\forall F. MF = F(MF)$$

- Curry “paradoksaalne” kombinaator

$$Y \equiv \lambda f. (\lambda x. f(x x)) (\lambda x. f(x x))$$

- Kombinaator Y on püsipunktikombinaator

$$\begin{aligned} Y e &\rightarrow_{\beta} (\lambda x. e(x x)) (\lambda x. e(x x)) \\ &\rightarrow_{\beta} e((\lambda x. e(x x)) (\lambda x. e(x x))) \\ &=_{\beta} e(Y e) \end{aligned}$$

Püsipunktid

- “Tugev” püsipunkti kombinaator

$$\Theta \equiv (\lambda xy. y(xxy)) (\lambda xy. y(xxy))$$

- Püsipunktikombinaatoreid saab kasutada rekursiivsete funktsioonide defineerimiseks.
- Näide:

$$\text{add} = \lambda x y. \text{cond} (\text{iszero } x) y (\text{add}(\text{pred } x)(\text{succ } y))$$

$$\text{add} \equiv Y (\lambda f x y. \text{cond} (\text{iszero } x) y (f (\text{pred } x)(\text{succ } y)))$$

IO näide

```
main : IO ()
main = do
  putStrLn "Kirjuta midagi!"
  xs ← getLine
  if (xs=="")
    then putStr "Sõnakuulmatu!"
    else do
      putStrLn "Tänan!"
      putStrLn ("Kirjutasid: " ++ xs)
```

on sama mis

```
main : IO ()
main =
  putStrLn "Kirjuta midagi!" >>
  getLine >>= (\ xs =>
    if (xs=="")
      then putStr "Sõnakuulmatu!"
      else
        putStrLn "Tänan!" >>
        putStrLn ("Kirjutasid: " ++ xs)
  )
```

Lihtsustatult:

- `getLine` : IO String
- `putStrLn` : String → IO ()

Maybe näide

```
data Expr = Const Int | Add Expr Expr | Div Expr Expr

evalExpr : Expr → Maybe Int
evalExpr (Const x) = pure x
evalExpr (Add x y) = do
  xv ← evalExpr x
  yv ← evalExpr y
  pure (xv+yv)
evalExpr (Div x y) = do
  xv ← evalExpr x
  yv ← evalExpr y
  if yv == 0
    then Nothing
    else pure (xv `div` yv)
```

Tüübid:

- `pure` : $\text{Applicative } f \Rightarrow a \rightarrow f a$
- `(>>=)` : $\text{Monad } m \Rightarrow m a \rightarrow (a \rightarrow m b) \rightarrow m b$

Seisundimonaadi näide

```
import StateT

arvutus : St Maybe Int String
arvutus = do x ← get
            update (λ y ⇒ y+1)
            pure (show x)
```

Tüübid:

- `pure` : $\text{Applicative } f \Rightarrow a \rightarrow f\ a$
- `(>>=)` : $\text{Monad } m \Rightarrow m\ a \rightarrow (a \rightarrow m\ b) \rightarrow m\ b$
- `get` : $\text{Monad } m \Rightarrow \text{St } m\ s\ s$
- `set` : $\text{Monad } m \Rightarrow s \rightarrow \text{St } m\ s\ ()$