

# Laboratorul 1

## Limbaajul unui mini calculator

Definim în Haskell limbaajul unui mini calculator:

```
data Prog = On Instr
data Instr = Off | Expr :> Instr
data Expr = Mem | V Int | Expr :+ Expr
type Env = Int    -- valoarea celulei de memorie
type DomProg = [Int]
type DomInstr = Env -> [Int]
type DomExpr = Env -> Int
```

Dorim ca un program să afișeze lista valorilor corespunzătoare expresiilor, unde Mem reprezintă ultima valoare calculată. Valoarea inițială a lui Mem este 0.

De exemplu, programul

```
On ((V 3) :> ((Mem :+ (V 5)) :> Off))
```

va afișa lista [3,8]

## Exercițiu

Să se definească funcțiile de evaluare a expresiilor de tip Prog, Instr și Expr.

```
prog :: Prog -> DomProg
stmt :: Instr -> DomInstr
expr :: Expr -> DomExpr
```

## Mini-Haskell

Vom defini folosind Haskell un mini limbaj funcțional și semantica lui denotațională.

Limbaajul Mini-Haskell conține:

- expresii de tip 'Bool' și expresii de tip 'Int'
- expresii de tip funcție ( $\lambda$ -expresii)

- expresii provenite din aplicarea functiilor

```

type Name = String

data Hask =
    | HTrue
    | HFalse
    | HLit Int
    | HIf Hask Hask Hask
    | Hask ==: Hask
    | Hask :+: Hask
    | HVar Name
    | HLam Name Hask
    | Hask $: Hask
    deriving (Read, Show)

infix 4 ==:
infixl 6 :+:
infixl 9 $:

```

Pentru a evalua (interpreta) expresiile vom defini un mediu de evaluare (memoria) in care vom retine variabilele si valorile curente asociate.

```

data Value =
    | VBool Bool
    | VInt Int
    | VFun (Value -> Value)
    | VError -- pentru reprezentarea erorilor

type HEnv = [(Name, Value)]

```

Deci domeniul de evaluare al expresiilor este

```

type DomHask = HEnv -> Value

```

## Exercițiu

- Afișarea expresiilor din Hask: să se instanțieze clasa `Show` pentru tipul de date `Value`.
- Egalitate pentru valori: să se instanțieze clasa `Eq` pentru tipul de date `Value`. Pentru cazurile în care nu se poate verifica egalitatea (funcții sau erori) se va întoarce o eroare folosind funcția `error`.
- Evaluarea expresiilor Mini-Haskell în Haskell: să se implementeze funcția de evaluare a unei expresii de tip `Hask`.

```

hEval :: Hask -> DomHask

```