

# Série d'exercices #4 $\frac{1}{2}$

IFT-2035

May 12, 2025

## Mini-évaluateur en manque de fonctions

```
type Var = String
-- Expressions du code source en forme ASA.
data Exp = Enum Int          -- Une constante
         | Evar Var          -- Une variable
         | Elet Var Exp Exp  -- Une expr "let x = e1 in e2"
         | Ecall Exp Exp     -- Un appel de fonction

-- Valeurs renvoyées.
data Val = Vnum Int          -- Un nombre entier
         | Vprim (Val → Val) -- Une primitive

elookup x ((x1,v1):env) =
  if x == x1 then v1 else elookup x env

eval env (Enum n) = Vnum n
eval env (Evar x) = elookup env x
eval env (Elet x e1 e2) =
  let v = eval env e1 in eval ((x,v):env) e2
eval env (Ecall fun actual) =
  case eval env fun of
    Vprim f -> f (eval actual)
```

Soit les déclarations ci-dessus utilisées pour un interpréteur:

1. Donner le type des deux fonctions. Vérifier que le code est typé correctement et corriger les éventuelles erreurs.
2. Étendre ce petit langage avec la possibilité de définir de nouvelles fonctions. I.e. Ajouter un constructeur d'expression `Elambda`, un constructeur de valeur correspondante `Vlambda`, ainsi que les cas pour gérer ces constructeurs dans `eval`.