

Série d'exercices #2 $\frac{1}{2}$

IFT-2035

May 9, 2021

Mini Évaluateur

Soit le code Haskell ci-dessous:

```
data Exp = Enum Int          -- Une constante
        | Eplus Exp Exp     -- e1 + e2
        | Etimes Exp Exp    -- e1 * e2
        | Eneg Exp          -- (- e)
        | Egt Exp Exp       -- e1 > e2
        | Enot Exp          -- (not e)
        | Eif Exp Exp Exp   -- if e1 then e2 else e3

data Val = Vnum Int
        | Vbool Bool

eval :: Exp -> Val
```

Exp est un type qui représente des expressions simples incluant des opérateurs arithmétiques et booléens. *Val* est un type qui représente des valeurs entières ou booléennes. *eval* est une fonction qui doit évaluer une expression et retourner la valeur qui en résulte.

1. Écrire la fonction *eval*
2. Proposer des changements au type *Exp* qui permettraient de réduire la redondance dans votre code