

Série d'exercices #3

IFT-2035

September 16, 2024

3.1 Classes d'équivalences

Écrire une fonction `equiv_classes` qui prend une liste de valeurs, une fonction de comparaison (d'équivalence) et qui groupe les éléments de la liste d'entrée en sous-listes contenant des valeurs équivalentes (d'après la fonction fournie). E.g.:

```
equiv_classes [1, 2, 3, 5, 6]
              (\x y → (x - y) `mod` 2 == 0)
⇒
[[6,2],[5,3,1]]
(ou [[5,3,1],[2,6]], ou ...)
```

Indiquer aussi le type de cette fonction et décrire quelles sont les propriétés que la fonction passée en argument doit obéir pour que le résultat soit bien défini.

3.2 Micro optimizer

Soit le code Haskell ci-dessous:

```
data Exp = Enum Int          -- Une constante
         | Evar String       -- Une référence à une variable
         | Eplus Exp Exp     -- e1 + e2
         | Etimes Exp Exp    -- e1 * e2

optimize :: Exp → Exp
```

`Exp` est un type Haskell (utilisé ici comme un métalangage) qui représente des expressions simples d'un *langage objet* incluant uniquement des opérateurs arithmétiques. Écrire la fonction `optimize` qui devrait simplifier une expression en éliminant *toutes* les multiplications par 1 et 0, ainsi que les additions à 0.

3.3 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 Haskell (utilisé ici comme un *métalangage*) qui représente des expressions simples d'un *langage objet* 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. Identifier les redondances qui apparaissent dans votre code.
3. Proposer des changements qui permettraient de réduire la redondance dans votre code.