

Série d'exercices #6

IFT-2035

May 26, 2025

6.1 Une classe de conteneurs

Soit la classe de types suivante:

```
class Conteneur a where
  size    :: a → Int
  subset  :: a → a → Bool
  reverse :: a → a
```

Où `reverse` renvoie un conteneur dont les éléments sont dans l'ordre inverse.

1. Donner une définition pour l'instance `Integer` (on n'utilisera pas les nombres négatifs), où chaque bit représente la présence du nombre correspondant, donc $11 = 2^0 + 2^1 + 2^3$ représente l'ensemble $\{0, 1, 3\}$.
2. Donner une définition pour l'instance `[Int]`¹. Pour vous aider vous pouvez utiliser la fonction prédéfinie `elem :: Eq a ⇒ [a] → [a] → Bool`.
3. Donner une définition pour l'instance `Maybe a`, qui ne peut contenir qu'un élément au maximum.
Rappel: `data Maybe a = Nothing | Just a`
4. Donner le type des fonctions suivantes:

```
sizes xs = map size xs
size2 x y z = if x == z
              then let s = size x in (s, s)
              else (size x, size y)
minsize x y = if subset x y then size x else size y
```

6.2 Expression conditionnelle à multiple branches

Écrire une macro `mif` en `ELisp` qui peut s'utiliser comme suit:

¹Les classes de type de Haskell n'acceptent pas ce genre d'instances à moins d'activer l'option "FlexibleInstances", par exemple en ajoutant `{-# LANGUAGE FlexibleInstances #-}`

```
(mif
  (cond1 exp1)
  (cond2 exp2)
  ...
  (_ expn))
```

et dont l'expansion devrait être

```
(if cond1 exp1
  (if cond2 exp2
    ...
    expn) ..)
```

Ensuite étendre cette macro pour que si exp_n manque, la macro renvoie la valeur de $cond_n$ à la place.

6.3 RPN en Lisp

Définir en ELisp la macro `postfix` qui prend une expression sous forme postfixée:

```
(let ((x 5))
  (postfix 1 x + 3 * 2 /))  $\implies$  9
```

Il suffira d'accepter les opérateurs `+`, `-`, `*`, `/`, `not`, `≥`, et `if`.

Montrer les étapes de l'évaluation de `(postfix 1 x + 3 * 2 /)` ci-dessus, jusqu'à l'obtention du résultat 9, en indiquant clairement quelles parties ont lieu lors de la compilation et quelles parties ont lieu à l'exécution.