heig-vd

Laboratoire 4

Fonctions d'ordre supérieur, Projection, Pliage, Filtrage

heig-vd

Exercice 4.1 - Projection

1. Définir la fonction check qui permet de construire la liste:
   `check f [(x1,y1) … (xn,yn)] → [(x1,y1, f x1 y1) … (xn,yn,f xn yn)]`
2. Écrire une fonction prenant une liste de fonctions et une valeur et qui applique chaque fonction à la valeur passée comme paramètre:
   `f [succ, pred] 2 → [3, 1]`
3. Ecrire une fonction qui transforme une liste d'éléments en liste de listes:
   `listes [x1,x2 .. xn] → [[x1],[x2] .. [xn]]`
4. Pour la fonction précédente vous avez probablement défini une fonction accessoire qui transforme un élément en liste; utilisez `flip` et `:` pour définir listes sans cette fonction accessoire.

heig-vd

Exercice 4.2 - Projection

1. Ecrire une fonction pour calculer la liste des carrés d'une liste de nombres:
   `carres [x1,x2 … xn] → [x1^2,x2^2 … xn^2]`

2. Ecrire la fonction précédente sans fonction accessoire
   (utiliser `(^)` et `flip`).

3. Ecrire une fonction qui transforme une liste de nombres en liste de listes:
   `[x1,x2 .. xn] → [[1..x1],[1..x2] .. [1..xn]]`

4. Ecrire une fonction qui remplace les nombres négatifs d'une liste par 0.

heig-vd

Exercice 4.3 - Filtrage

1. Filtrer les éléments pairs ou impairs d'une liste.
2. Filtrer les éléments d'une liste supérieurs ou inférieurs a une valeur donnée.
3. Ecrire une fonction qui prend une liste de chaines de caractères et ne retient que les 5 premiers caractères de chaque chaine.
4. Ecrire une fonction qui permet de chercher le premier élément dans une liste qui satisfait un prédicat:

```
cherche (>3) [1,2,3,4,5,6]
```

heig-vd

Exercice 4.4 - Pliages

- Utiliser un pliage pour écrire la fonction
  `existe :: … => [t] -> (t -> Bool) -> Bool`
  qui correspond au quantificateur **existentiel**.
  - `existe list prédicat` retourne `True` si au moins un élément de la liste satisfait le prédicat
- Utiliser un pliage pour écrire la fonction
  `tous :: … => [t] -> (t -> Bool) -> Bool`
  qui correspond au quantificateur **universel**.
  - `tous liste prédicat` retourne `True` si tous les éléments de la liste satisfont le prédicat.
- Donner une version alternative de ces fonctions sans définir de fonction auxiliaire pour le pliage, en utilisant `map` pour transformer la liste `[t]` en `[Bool]`.

heig-vd

Exercice 4.5 - Chaines de Collatz

- Une chaine de Collatz est la liste de nombres :
  $[x_i, x_{i+1}, x_{i+2}, \ldots, 1]$ tels que:
  $x_{i+1} = x_i = x_i*3+1$ si x_i impair
  $x_{i+1} = x_i = x_i/2$ si x_i pair
- Exemple : La chaine de Collatz de 3 est:
  `[3, 10, 5, 16, 8, 4, 2, 1]`
- Ecrire une fonction pour calculer ces chaines.
- Quelle est la longueur de la chaine de Collatz de 156159 ?