Travaux Dirigés de programmation fonctionnelle n°5

Licence d'informatique

Les listes en Caml

Nous allons voir dans ce TD comment manipuler très simplement des listes en Caml grâce à la définition récursive suivante : une liste est soit vide (`[]`) soit un élément suivi d'une liste (`element::liste`).

▸ Exercice 1. Opérations de base

1. Écrire une fonction qui teste si une liste est vide.
2. Écrire une fonction qui renvoie le nombre d'éléments d'une liste.
3. Écrire une fonction insérant un entier à la fin d'une liste.
4. Écrire une fonction qui teste si un entier appartient à une liste.
5. Écrire une fonction prenant 2 listes et qui renvoie leur concaténation.

▸ Exercice 2. Opérations sur les éléments

1. Écrire des fonctions qui calculent la somme et le produit d'une liste d'entiers.
2. Écrire une fonction qui calcule le minimum d'une liste d'entiers positifs.
3. Écrire une fonction qui retourne la concaténation de toutes les chaînes de caractères d'une liste.
4. Quelles ressemblance voit-on apparaître dans la structure de ces fonctions ? Nous allons voir comment écrire une fonction qui fait le travail en une fois.

Toutes ces fonctions retournent le résultat de l'application d'une fonction à une liste.

▸ Exercice 3. Fonctionnelles sur les listes

1. Écrire une fonction `do_list` qui appelle une fonction `f : ('a -> unit)` pour chacun des éléments d'une liste.
2. Écrire la fonction `map` qui applique une fonction `f` à tous les éléments `x` d'une liste, et renvoie la liste de tous les résultats `f(x)`.
3. Écrire une fonction `iterateur` qui permette d'unifier les écritures des fonctions de l'exercice précédent. Cette fonction prendra en paramètres une liste `l` et une fonction `f : 'a -> 'a -> 'a`, et retournera un résultat dont de type `'a`.

   Récrire les fonctions de l'exercice 2 en utilisant `iterateur`.