
Programmation fonctionnelle

Répétition 3

5 Mars 2015

Exercices sur les listes

Exercice 1.

Définir la fonction `count` à deux arguments qui renvoie le nombre de fois que l'on a rencontré le premier argument dans la liste en second argument.

Exercice 2.

Définir la fonction `make-list` qui prend comme arguments un nombre `n` et une expression quelconque `x` et qui construit une liste composée de `n` fois l'élément `x`.

Exercice 3.

Les nombres de Gribomont sont définis comme suit :

$$\begin{aligned} \text{si } n < 3, f(n) &= n \\ \text{si } n \geq 3, f(n) &= f(n-1) + f(n-2) + f(n-3). \end{aligned}$$

Écrire une fonction qui permet de calculer les nombres de Gribomont.

Correction exercices proposés

Exercice 4.

Écrire une fonction `big` qui prend comme arguments un nombre entier `n` et une liste `l` de nombres entiers, telle que `(big n l)` est la liste des éléments de `l` plus grands que `n`.

$$(\text{big } 5 '(7 -3 6 1 -2 5 6)) \Rightarrow (7 6 6)$$

Exercice 5.

Écrire une fonction `removeAll` à deux arguments `ls` et `n` dont les valeurs sont respectivement une liste de nombres entiers et un nombre entier, qui retourne la liste privée de toutes les occurrences de `n`.

$$(\text{removeAll } '(2 7 1 7 3 1) 1) \Rightarrow (2 7 7 3)$$

Exercice 6.

Définir la fonction `suffix` à deux arguments `l1` et `l2` dont les valeurs sont des listes et qui retourne `#t` si `l1` est suffixe de `l2` et `#f` sinon.

```
(suffix '(1 2) '(3 x 1 2)) ⇒ #t  
(suffix '(1 2) '(3 x 2)) ⇒ #f
```

Exercice proposé

Exercice 7.

Écrire une fonction `frequency` prenant en argument une liste d'atomes `ls` et renvoyant une table d'apparition de chacun des atomes dans la liste `ls`. Cette table sera représentée par une liste de paires pointées, dont le car est un atome et le cdr la fréquence d'apparition de cet atome. Tous les atomes de `ls` apparaissent une et une seule fois dans la table.

```
(frequency '(a b c b a b d a c b b))  
⇒ ((a . 3) (b . 5) (c . 2) (d . 1))
```