INFO0054 - Programmation fonctionnelle

Répétition 8

Jean-Michel Begon

07 Mai 2019

Récursivité forte de nombres

Exercice 1.
Ecire une fonction *efficace* permettant de calculer

$$f(n) = \sum_{i=0}^{n-1} f(n-i-1)^{f(i)}$$

Exercice 2.
Ecire une fonction *efficace* permettant de calculer

$$f(n) = \sum_{i=0}^{n-1} \big(((f(i)+2) \times (f(n-i-1)+3)) \mod \big(n^2+i+5\big)\big)$$

Inclusion/exclusion

Exercice 3.
Écrire une fonction `nbsum` qui prend comme argument un nombre n et qui renvoie le nombre de façons d'écrire une somme égale à n (on comptera une seule fois les commutations).

Exercice 4.
Écrire la fonction `lagrange`, prenant comme argument un entier naturel n et renvoyant la liste de tous les quadruplets d'entiers naturels (a, b, c, d) tels que $a^2 + b^2 + c^2 + d^2 = n$.

```
(lagrange 13) ==>
((0 0 2 3) (0 0 3 2) (0 2 0 3) (0 2 3 0) (0 3 0 2) (0 3 2 0)
 (1 2 2 2) (2 0 0 3) (2 0 3 0) (2 1 2 2) (2 2 1 2) (2 2 2 1)
 (2 3 0 0) (3 0 0 2) (3 0 2 0) (3 2 0 0))

(lagrange 18)
((0 0 3 3) (0 1 1 4) (0 1 4 1) (0 3 0 3) (0 3 3 0) (0 4 1 1)
 (1 0 1 4) (1 0 4 1) (1 1 0 4) (1 1 4 0) (1 2 2 3) (1 2 3 2)
 (1 3 2 2) (1 4 0 1) (1 4 1 0) (2 1 2 3) (2 1 3 2) (2 2 1 3)
 (2 2 3 1) (2 3 1 2) (2 3 2 1) (3 0 0 3) (3 0 3 0) (3 1 2 2)
 (3 2 1 2) (3 2 2 1) (3 3 0 0) (4 0 1 1) (4 1 0 1) (4 1 1 0))
```

Exercice 5.
Une *tricoupure* d'une liste ℓ est une liste de trois listes non vides dont la concaténation (dans l'ordre) vaut ℓ. Écrire une fonction `3-cuts` qui à toute liste ℓ associe la liste des tricoupures de ℓ.

Par exemple, si ℓ est `(a b)` la liste des tricoupures de ℓ est la liste vide ; si ℓ est `(a b c d)` la liste des tricoupures de ℓ comporte, dans un ordre quelconque, les trois listes `((a b) (c) (d))`, `((a) (b c) (d))` et `((a) (b) (c d))`.

Variante

Une *tricoupure* d'une liste ℓ est une liste de trois listes dont la concaténation (dans l'ordre) vaut ℓ. Écrire une fonction `tricoup-lv` qui à toute liste ℓ associe la liste des tricoupures de ℓ.

Par exemple, si ℓ est `(a)` la liste des tricoupures de ℓ comporte, dans un ordre quelconque, les trois listes `((a) () ())`, `(() (a) ())` et `(() () (a))`.

Arbres

Exercice 6.
Générer à partir d'une liste, tous les arbres binaires complets dont la lecture des feuilles de gauche à droite donne la liste.

Par exemple, si on représente un nœud par une liste et une feuille par l'atome étiquette.

```
(binary-trees '(a b c d)) ==>
((a (b (c d))) ((a b) (c d)) (a ((b c) d)) ((a (b c)) d)
 (((a b) c) d))
```