

INFO0054 - Programmation fonctionnelle

Répétition 5

Jean-Michel BEGON

19 Mars 2019

Polynômes

Exercice 1.

Écrire une fonction qui à tout polynôme $P(x)$ et naturel n associe la dérivée n -ième du polynôme $\frac{d^n P(x)}{d^n x}$.

On représentera un polynôme par la liste de ses coefficients, par ordre croissant des degrés.
 $\Rightarrow a + bx + cx^2 + \dots$ sera représenté par la liste (a b c ...).

Exercice 2.

Écrire une fonction qui à deux polynômes P et Q associe le polynôme composé $P \circ Q$.

On représentera un polynôme par la liste de ses coefficients, par ordre croissant des degrés.
 $\Rightarrow a + bx + cx^2 + \dots$ sera représenté par la liste (a b c ...).

Listes de listes

Exercice 3.

Définir la procédure `count-all` à deux arguments, un élément et une liste, et qui compte, en profondeur, le nombre de de fois que l'élément est contenu dans la liste.

(count-all 1 '(0 (1 2 (3 4 (1)) (3 (2 1) 1) 1) 0 (1 2 (1 2 3)))) \Rightarrow 7

Exercice 4.

Écrire une fonction `prod-cart` qui prend en entrée deux ensembles et qui retourne l'ensemble des paires pointées dont le `car` est un élément du premier ensemble et le `cdr` un élément du second :

(prod-cart '(1 2 3) '(a b)) \Rightarrow '((1 . a) (1 . b) (2 . a) (2 . b) (3 . a) (3 . b))

Exercice 5.

Écrire une fonction `insert-a` qui prend une liste `l` en argument et qui renvoie la liste de toutes les listes `l` auxquelles 'a a été ajouté à une position donnée.

Exercice 6.

Écrire une fonction `lpref` prenant comme argument une liste `u` et retournant la liste des préfixes de `u`. (La liste vide et la liste `u` elle-même sont des préfixes de `u`.)

Exercice 7.

Écrire une fonction qui renvoie la liste des sous-ensembles d'un ensemble donné.

```
(lset '(a b c)) ⇒ '(() (c) (b) (b c) (a) (a c) (a b) (a b c))
```

Exercice 8.

Écrire une fonction qui renvoie la liste des permutations d'une liste donnée.

Exercice 9.

Soit un alphabet décrit par une liste de symboles. Écrire une fonction `words` qui engendre la liste (dans un ordre quelconque) des mots de longueur `n` écrits dans cet alphabet.

```
(words 2 '(a b c)) ⇒ ((a a) (b a) (c a) (a b) (b b) (c b) (a c) (b c) (c c))
```