INFO0054 - Programmation fonctionnelle

Répétition 4: Récursion sur les nombres et sur les listes

Jean-Michel Begon

09 Mars 2017

Représentation en mémoire

Exercice 1.
Représenter le résultat en mémoire des opérations suivantes :

```
'(2)
(cons 1 2)
(list 1 2)
'(3 4 5)
```

```
'((3 4 5))
(list '(1 2) '(3 4 5))
(cons '(1 2) '(3 4 5))
(append '(1 2) '(3 4 5))
```

Listes

Exercice 2.
Ecrire une fonction `insert` qui prend en entrée `ls`, une liste de nombres par ordre croissant, ainsi que `x`, un nombre, et qui renvoie une liste contenant `x` et tous les éléments de `ls` par ordre croissant.

Exercice 3.
Soit la fonction f définie sur $\mathbb{N}$ par :

$$f(n) = \left(\sum_{i=0}^{n-1} ([2 + f(i)] \times [3 + f(n - i - 1)]) \right) \mod (2n + 3)$$

Implémentée par le programme suivant :

```
(define f (lambda (n) (car (fx n 0 '()))))
(define fx
  (lambda (k i u)
    (if (zero? k) u
        (let ((next
               (modulo (apply + (map (lambda (x y) (* (+ 2 x) (+ 3 y)))
                                     u
                                     (reverse u)))
                       (+ i i 3))))
          (fx (- k 1) (+ i 1) (cons next u))))))
```

Spécifier la fonction `fx`.

Exercice 4.

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 5.
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` le nombre d'occurrences 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))
```

Les diviseurs

Exercice 6.
Un entier naturel est *parfait* s'il est égal à la somme de ses diviseurs positifs propres. Ecrire un prédicat `perfect?` déterminant si son argument est un nombre parfait.

Le nombre 28 est parfait parce que $28 = 1 + 2 + 4 + 7 + 14$;
le nombre 32 n'est pas parfait parce que $32 \neq 1 + 2 + 4 + 8 + 16$.

Exercice 7.
Écrire un prédicat `prime?` qui détermine si un entier strictement positif est premier ou non.

Exercices proporés

Exercice 8.
Ecrire la fonction `conway` qui calcule le $n$ieme terme de la suite de Conway :

$$1$$
$$1, 1$$
$$2, 1$$
$$1, 2, 1, 1$$
$$1, 1, 1, 2, 2, 1$$

Exercice 9.
Implémenter le quicksort, spécifier les fonctions auxiliaires et discuter sa complexité.

Exercice 10.
Implémenter le tri par fusion, spécifier les fonctions auxiliaires et discuter sa complexité.