

TD numéro 3

1 Fonctions sur les listes en langage algorithmique

On dispose de trois fonctions prédéfinies sur les listes :

- Premier(liste) : retourne le premier élément de la liste
- Reste(liste) : retourne la liste privée du premier élément
- Vide(liste) : retourne VRAI si la liste est vide, FAUX sinon
- Donner une définition récursive de la fonction **Longueur**(liste), qui retourne le nombre d'éléments d'une liste. *À préparer avant la séance*
- Donner une définition récursive de la fonction **Somme**(liste), qui retourne la somme des éléments d'une liste de nombres. *À préparer avant la séance*
- Donner une définition récursive de la fonction **Appartient**(élément, liste), qui retourne VRAI si l'élément est dans la liste et FAUX sinon.
- Donner une définition récursive de la fonction **Dernier**(liste), qui retourne le dernier élément d'une liste non vide.

2 Évaluation d'expressions en Scheme

- (car (cdr '(a b c d)))
- (car (cdr '(abc d)))
- (cdr '(a (b c d)))
- (cdr '((a b c d)))
- (cdr (car (cdr '(a (b c) (d e))))))
- (cons '(a b) '(c d))
- (cons 'a (cons 'b '(c d)))
- (cons '(a b) 'c)
- (list? (+ 2 3))
- (list? '(+ 2 3))

3 Premières fonctions sur les listes en Scheme

Écrire une fonction:

- qui retourne le second élément d'une liste d'au moins deux éléments ; *À préparer avant la séance*
- qui retourne vrai si et seulement si une liste n'a qu'un seul élément.

4 Fonctions récursives sur les listes en Scheme

Écrire une fonction :

- qui calcule la longueur d'une liste ;
- qui prend une liste et un élément et retourne vrai si et seulement si l'élément appartient à la liste ;
- qui retourne le n^{ième} élément d'une liste ;
- qui insère un élément dans une liste après le i^{ème} élément.

Donner la spécification de la fonction mystère ci-dessous :

```
(define mystere
  (lambda (x l)
    (cond ((null? l) 0)
          ((eq? x (car l)) (+ 1 (mystere x (cdr l))))
          (else (mystere x (cdr l))))))
```