MIAS 1 et MASS 1 – Programmation fonctionnelle

TD n° 3 – Corrigé

**Exercice 1 :**

```
(cons 'a (cons 'b 'c))   ->   (a b . c)
(cons '() '())   ->   (())
(cons 'a (cons (cons 'b '()) '())   ->   (a (b))
(pair? '())   ->   #f
(pair? (cons 'a '()))   ->   #t
(list? (+ 2 3))   ->   #f
```

**Exercice 2 :**

```
(car (cdr '(a b c d)))   ->   b
(cdr '(a (b c d)))   ->   (b c d)
(car '((a b c)))   ->   (a b c)
(car (cdr '(a (a b c) d f)))   ->   (a b c)
(caddr '(a (a b c) d f))   ->   d
(caar '(a (b c)))   ->   erreur
```

**Exercice 3 :** l =(1 2 (3 (4) 5) 6 (7 8) 9)

```
(cons (car l) (caddr l))
(cons (cadddr l) (cons (car l) '()))
```

**Exercice 4 :** singleton? : $\mathit{Liste} \longrightarrow \mathit{Booléen}$

$$l \longmapsto \begin{cases} \text{vrai si l ne contient qu'un élément} \\ \text{faux sinon} \end{cases}$$

Une liste ne contient qu'un seul élément si elle est non vide et si son reste est vide.

```
(define singleton?
   (lambda (l)
       (and (pair? l)
            (null? (cdr l)))) )
```

**Exercice 5 :** somme : $\mathit{Liste}^* \longrightarrow \mathit{Réel}$

$$l \longmapsto \begin{cases} a \text{ si } l = (a) \\ a + b \text{ si } l = (a\ b) \\ a + b + c \text{ si } l = (a\ b\ c) \end{cases}$$

Remarque : on suppose ici que l'argument est une liste non vide de 3 nombres réels; dans les autres cas la fonction n'est pas définie.

```
(define somme
   (lambda (l)
       (cond ((null? (cdr l)) (car l))
             ((null? (cddr l)) (+ (car l) (cadr l)))
             (else (+ (car l) (cadr l) (caddr l))) )))
```