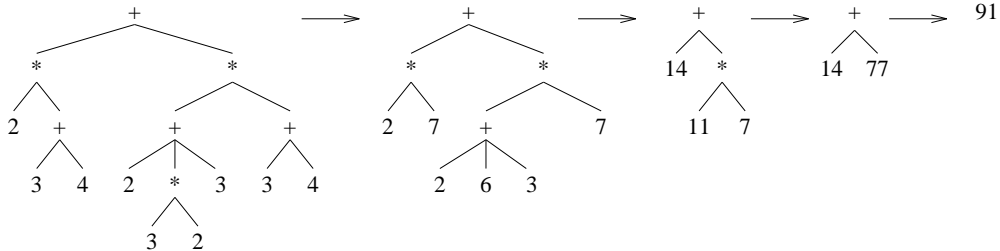


Exercice 1 :

a) $(* (+ (- 3) 4) (+ * 12))$

b) $(* (+ 4 2 (* 3 4) (+ (* 2 3) 4))$

c) $(+ (* 2 (+ 3 4)) (* (+ 2 (* 3 2) 3) (+ 3 4)))$



Exercice 2 :

carre: $R\acute{e}el \rightarrow R\acute{e}el$
 $x \mapsto x * x$

```
(define carre
  (lambda (x) (* x x)))
```

Exercice 3 :

volume-sphere: la lambda-expression est mal formée, le corps ne doit être formé que d’une expression (composée ou simple) or là il y a deux expressions. L’interprète ne signale pas d’erreur mais le résultat est celui de l’évaluation de la dernière expression: $(* r r r)$.

```
(define volume-sphere
  (lambda (r)
    (* (/ 4 3)
       3.1415927
       (* r r r) ) ) )
```

successeur: on utilise l’écriture préfixée en Scheme.

```
(define successeur
  (lambda (x)
    (+ x 1) ) )
```

double: le paramètre x n’est pas lié à une valeur. Soit on veut définir une variable double dont on lie la valeur à deux fois la valeur d’une variable x et dans ce cas il faut d’abord lier x à une valeur :

```
(define x 12)
(define double (* 2 x))
```

Soit on veut définir une fonction double: $R\acute{e}el \rightarrow R\acute{e}el$ et alors il faut utiliser une lambda-expression $x \mapsto 2 * x$

dont le paramètre formel est x :

```
(define double
  (lambda (x)
    (* 2 x)))
```

aire-triangle: les paramètres base et hauteur doivent être les paramètres formels de la lambda-expression, sinon il ne seront liés à aucune valeur lors de l’appel de la fonction. Le paramètre triangle est inutile.

```
(define aire-triangle
  (lambda (base hauteur)
    (* 0.5 base hauteur) ) )
```