

(Séquence 9.3

Évaluation dans un environnement



Évaluation dans un environnement

L'environnement est représenté par une liste d'associations

```
Environnement ≡ LISTE[COUPLE[Symbole Nombre]]
```

```
(define (MonEnv) ' ((x 3) (y 4) (z 5)))
```

```
(valeur-de 'y (MonEnv)) → 4
```

La fonction d'évaluation doit donc avoir 2 arguments :

- ▶ l'expression elle même
- ▶ et la liste d'associations représentant l'environnement.



Spécification et application

```
;;; evaluation-expr: S-Expr  
;;;      * LISTE[COUPLE[Symbole Nombre]] -> Nombre  
;;; (evaluation-expr E env) rend la valeur de  
;;; l'expression arithmétique E dans  
;;; l'environnement env
```

```
(define (MonEnv) ' ((x 3) (y 4) (z 5)))  
  
(evaluation-expr '(+ x (* 3 y 5) (+ 1 2 y (* z 10)))  
  (MonEnv))  
  
→ 120
```

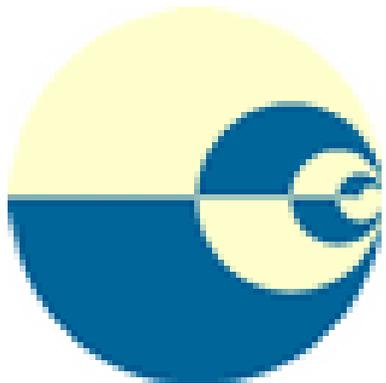


Définition de `evaluation-expr`

```
(define (evaluation-expr E env)
  ;; evaluation-expr-env: S-Expr -> Nombre
  ;; (evaluation-expr-env E): même sémantique
  ;; que evaluation-expr mais fonction rendue
  ;; unaire pour utilisation dans un map
  (define (evaluation-expr-env E)
    (evaluation-expr E env))

  (if (pair? E)
    (let ((args (map evaluation-expr-env
                      (cdr E) )))
      (application (operation (car E)) args))
    (if (symbol? E)
      (valeur-de E env)
      E)))
```





Fin séquence)

