



# (Séquence 8.2

## Récursion sur les arbres généraux



# Récursion sur les arbres généraux

Un arbre général est constitué :

- ▶ d'une étiquette
- ▶ et d'une forêt (liste) d'arbres généraux

Traitement d'un arbre :

- ▶ Pour traiter un arbre, il faut traiter la forêt de ses descendants directs.
- ▶ Pour traiter une forêt, il faut traiter chaque arbre de cette liste.

C'est une **récursion croisée**.

Les itérateurs `map` et `reduce` sont utiles.



# Schéma récursif sur les arbres généraux

```
;;; ArbreRec: ArbreGeneral[ $\alpha$ ] ->  $\beta$ 
(define (ArbreRec G)
  (combinaison1 (ag-etiquette G)
                (ForetRec (ag-foret G)) ) )

;;; ForetRec: LISTE[ArbreGeneral[ $\alpha$ ]] ->  $\beta$ 
(define (ForetRec F)
  (if (pair? F)
      (combinaison2 (ArbreRec (car F))
                    (ForetRec (cdr F)) )
      base ) )
```



# Schéma récursif sur les arbres généraux

Avec usage des itérateurs `map` et `reduce` :

```
;;; ForetRec: LISTE[ArbreGeneral[ $\alpha$ ]] ->  $\beta$   
(define (ForetRec F)  
  (reduce combinaison2 base (map ArbreRec F)) )
```

**et même en une seule fonction:**

```
;;; ArbreRec: ArbreGeneral[ $\alpha$ ] ->  $\beta$   
(define (ArbreRec G)  
  (combinaison1 (ag-etiquette G)  
    (reduce combinaison2 base  
      (map ArbreRec (ag-foret G)) ) ) )
```





**Fin séquence)**

