



(Séquence 4.5

Schéma de récursion



Schéma usuel de récursion

Une liste est :

- ▶ soit vide
- ▶ soit constituée d'un premier *élément* (`car L`) et d'une *liste* restante (`cdr L`)

```
;;; fRec: LISTE[alpha] -> ...  
(define (fRec L)  
  (if (pair? L)  
    (combinaison  
      (car L)  
      (fRec (cdr L)) )  
    cas-liste-vide ) )
```



Méthode d'écriture

Pour écrire une fonction récursive, il est très fortement recommandé de suivre la méthode suivante :

1. décrire le calcul à partir du calcul sur des éléments « plus petits » (relation de récurrence),
2. déterminer les valeurs dites « valeurs de base » pour lesquelles :
 - 2.1 la relation de récurrence n'est pas définie,
 - 2.2 les valeurs d'appel de la fonction récursive, pas **strictement** « plus petites » que la donnée.
3. déterminer les valeurs de la fonction pour les valeurs de base.



Arrêt de récursion

Pour que les récursions s'achèvent, il faut

1. une « diminution » dans l'argument de l'appel récursif
 - ▶ Si la récursivité se fait sur les entiers naturels, la valeur de l'argument de l'appel récursif doit diminuer (par exemple $n - 1$, ou $n/2$, ou ...)
 - ▶ Si la récursivité se fait sur une liste, la longueur de la liste passée en argument de l'appel récursif doit être plus courte (par exemple `(cdr L)`).
2. un test d'arrêt sur les valeurs de base





Fin séquence)

