



(Séquence 5.4

Schéma général de récursion



méta-schéma de récursion



Le schéma de récursion usuel sur les listes est :

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

On peut l'abstraire en

```
(define (schema combinaison cas-liste-vide)  
  (define (frec L)  
    (if (pair? L)  
      (combinaison (car L) (frec (cdr L)))  
      cas-liste-vide ) )  
  frec )
```



Applications de schema



```
(define (c0 f)
  (define (c01 e r)
    (cons (f e) r) )
  c01 )
(map f L) ≡ ((schema (c0 f) (list)) L)
```

```
(define (c1 pred)
  (define (c11 e r)
    (if (pred e) (cons e r) r) ) )
(filtre pred L) ≡ ((schema (c1 pred) (list)) L)
```

```
(somme L) ≡ ((schema + 0) L)
```

```
(define (c2 e r) (+ 1 r))
(longueur L) ≡ ((schema c2 0) L)
```





Fin séquence)

