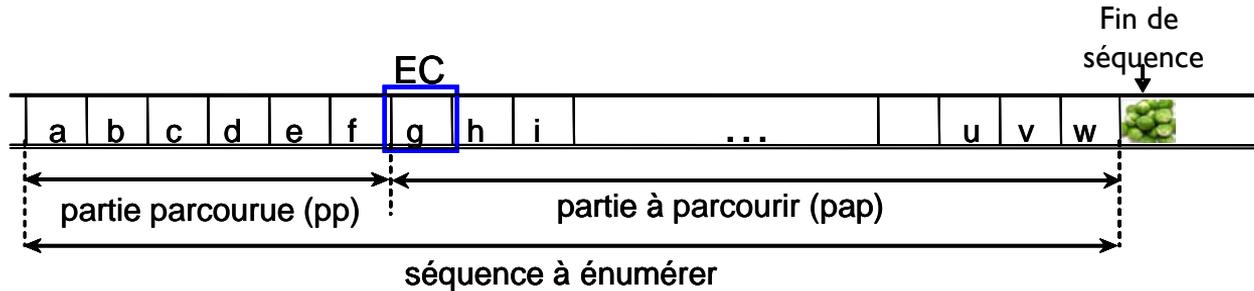




Propriétés générales de l'itération

Jean-Michel Adam - UGA - UFR SHS

Parcours d'une séquence



- Dans un algorithme de parcours, le lexique est composé
 - des **variables de parcours** servant à énumérer les éléments de la séquence,
 - des **variables de traitement** servant à réaliser le traitement des éléments de la séquence.

Nous nous intéressons ici à la spécification des variables de traitement.

Traitement d'une séquence

- Les variables de traitement apparaissent dans:
 - Initialisation du traitement
 - Traitement de l'élément courant
 - Terminaison du traitement
 - Traitement de la séquence vide (si nécessaire)
- Les variables de traitement
 - Sont en général définies **par rapport à la partie parcourue** de la séquence
 - On utilise une fonction pour caractériser cette définition
 - Cette fonction est définie par récurrence, la récurrence est basée sur la définition récurrente d'une séquence.

Définition d'une séquence par récurrence

- Base :

[] la séquence vide est une séquence

- Récurrence :

soit \mathbf{s}_0 une séquence, soit \mathbf{e} un élément

on obtient une nouvelle séquence s_1 en plaçant \mathbf{e} à la suite de \mathbf{s}_0 :

$$s_1 = s_0 \mid \mathbf{e}$$

Fonction récursive f caractérisant une variable de traitement

- Cas de base:
 - $f([]) = v_0$ valeur de f pour une séquence vide
- Cas général
 - Soient s_0 une séquence et e un élément, on a :
 $f(s_0|e) = g(f(s_0), e)$
calcul de f pour une séquence comportant un élément de plus

Rappel: Schéma de parcours d'une séquence S

Démarrer // $pp_S = []$

InitialisationDuTraitement

tantque non **findeSéquence** faire

// $pp_S = s_0$, **élémentCourant** = e

Traitement élémentCourant

Avancer

// $pp_S = s_0|e$

ftantque

// **findeSéquence** : $pp_S = S$

TerminaisonDuTraitement

Définition de la variable de traitement et des instructions de traitement et d'initialisation

Rappel : $f([]) = v_0$ et $f(s_0|e) = g(f(s_0), e)$

Soit v la variable calculant f pour une séquence S

Lexique

$v : \text{type_de_v} \quad // v = f(pp_S)$

Initialisation du traitement :

$v \leftarrow v_0$

Traitement de l'élément courant :

$v \leftarrow g(v, \text{élémentCourant})$

Parcours d'une séquence S

Lexique

v : type_de_v // $v = f(pp_S)$

Algorithme

Démarrer // $pp_S = []$

$v \leftarrow v_0$ // $v = f([])$

tantque non **findeSéquence** faire

// $pp_S = s_0$, **ElementCourant** = e , $v = f(s_0)$

$v \leftarrow g(v, \mathbf{élémentCourant})$ // $v = f(s_0|e)$

Avancer

// $pp_S = s_0|e$, $v = f(s_0|e)$ donc : $v = f(pp_S)$

ftantque

// **FindeSéquence** : $pp_S = S \Rightarrow v = f(S)$

TerminaisonDuTraitement

Exemple : moyenne d'une séquence d'entiers

- Pour calculer la moyenne il faut :
 - La somme des éléments de la séquence
 - Le nombre d'éléments de la séquence (sa longueur)
- Définissons somme et longueur par récurrence:

Somme :

- Base : $\text{somme}([]) = 0$
- Récurrence : $\text{somme}(s_0|e) = \text{somme}(s_0) + e$

Longueur:

- Base : $\text{longueur}([]) = 0$
- Récurrence : $\text{longueur}(s_0|e) = \text{longueur}(s_0) + 1$

Exemple : moyenne d'une séquence d'entiers (fichier)

Algorithme

```
f.lirePremier           // ppS = []
s ← 0 ; n ← 0          // s = somme([]), n = longueur([]),
tantque non f.fdf faire
    // ppS = s0, f.ec = e , s = somme(s0), n = longueur(s0)
    s ← s + f.ec ; n ← n + 1
    // s = somme(s0|e), n = longueur(s0|e)
    f.lireSuivant
    // ppS = s0|e ⇒ { s = somme(s0|e) = somme(ppS)
    //                  n = longueur(s0|e) = longueur(ppS)
ftantque
// f.fdf : ppS = S ⇒ s = somme(S), n = longueur(S)
selon n
    n = 0 : e.afficher("séquence vide")
    n ≠ 0 : e.afficher("moyenne =", s/n)
fselon
```

Lexique

f: fichier d'entiers // séquence S parcourue
n : entier ≥ 0 // n = longueur(pp_S)
s : entier // s = somme(pp_S)
e: écran // pour affichage résultats

Exercice : calcul de la factorielle

Base : $\text{fact}(0) = 1$

Récurrance : $\text{fact}(n) = \text{fact}(n-1) * n$

Quelle est la séquence énumérée ?

A quoi correspond s_0 ?

A quoi correspond e ?

Base : $\text{fact}(0) = 1$

Récurrance : $\text{fact}(e) = \text{fact}(s_0) * e$

Calcul de la factorielle

Fonction **fact**(n : entier ≥ 0) \rightarrow entier > 0

// fact(n) renvoie factorielle de n : $n!$

Lexique

n : entier ≥ 0

// paramètre : valeur dont on calcule la factorielle

k : entier ≥ 0

// élément courant de la séquence d'entiers énumérée

f : entier > 0

// f = factorielle($k-1$)

Algorithme

$k \leftarrow 1$

$f \leftarrow 1$

// f = fact(0)

tantque $k \leq n$ faire

// f = factorielle($k-1$) (invariant)

 // $k = k_0$, f = factorielle(k_0-1)

$f \leftarrow f * k$;

 // f = factorielle(k_0)

$k \leftarrow k + 1$

 // $k = k_0+1 \Rightarrow f$ = factorielle(k_0) = factorielle($k-1$)

ftantque

// $k = n+1 \Rightarrow f$ = factorielle($k-1$) = factorielle(n)

renvoyer(f)

Fonction mystérieuse

Fonction mystère($n, q : \text{entier} \geq 0$) \rightarrow $\text{entier} \geq 0$

Lexique

$n, q : \text{entier} \geq 0$

$k : \text{entier} \geq 0$

$s : \text{entier}$

Algorithme

$k \leftarrow 0$

$s \leftarrow 0$

tantque $k < n$ faire

$s \leftarrow s + q ;$

$k \leftarrow k + 1$

ftantque

renvoyer(s)

Fonction mystérieuse

Fonction mystère($n, q : \text{entier} \geq 0$) \rightarrow entier ≥ 0

Lexique

// $n, q : \text{entier} \geq 0$ // paramètres

$k : \text{entier} \geq 0$

$s : \text{entier}$ // $s = k * q$

Algorithme

$k \leftarrow 0$

$s \leftarrow 0$

tantque $k < n$ faire

// $k = k_0, s = s_0 = k_0 * q$

$s \leftarrow s + q ;$ // $s = s_0 + q = k_0 * q + q = (k_0 + 1) * q$

$k \leftarrow k + 1$ // $k = k_0 + 1$

// $k = k_0 + 1, s = (k_0 + 1) * q \Rightarrow s = k * q$

ftantque

// $k = n \Rightarrow s = n * q$

renvoyer(s) // $n * q$

