

4. Les séquences

Définition

- Soient un ensemble de valeurs E et un ensemble fini totalement ordonné I d'indices, une séquence s est une application de I dans E

$$s : I \rightarrow E$$

A chaque i de I correspond un élément de $E : i \rightarrow s_i$

I est souvent \mathbb{N} ou une partie de \mathbb{N} , mais ce peut être un type scalaire discret.

L'énumération des éléments de la séquence induit un ordre total appelé ordre induit par la séquence (ordre des rangs) \neq ordre éventuel sur E .

- Séquence triée : soit E totalement ordonné par la relation \leq , une séquence est triée lorsque l'ordre induit par la séquence est compatible avec l'ordre sur E :

$$\forall (i, j) \in I, i \leq j \text{ ssi } s_i \leq s_j$$

Trier une séquence s , c'est trouver une séquence triée s' formée des mêmes éléments que s .

Cela n'est possible que s'il existe une relation d'ordre sur E .

TAD Seq

Sorte : Seq

Utilise : Entier

Opérations :

CréerSeq(Entier, Entier) \rightarrow Seq *[opération interne]*

Changerlème(Seq, Entier, Elem) \rightarrow Seq *[opération interne]*

BorneInf(Seq) \rightarrow Entier

BorneSup(Seq) \rightarrow Entier

lème(Seq, Entier) \rightarrow Elem

Axiomes :

BorneInf (CréerSeq(x,y)) = x

BorneInf (Changerlème (s,n,v)) = BorneInf (s)

BorneSup (CréerSeq (x,y)) = y

BorneSup (Changerlème (s,n,v)) = BorneSup (s)

lème(CréerSeq (x,y),n) = Indéfini

lème(Changerlème (s,n,v),m) = **si** m < BorneInf (s) **ou** m > BorneSup (s)

alors Indéfini **sinon si** m=n **alors** v **sinon** lème(s,m)

Pour supprimer un élément, on met la valeur vide ε à la place de l'élément.

TAD SeqTriée

□ Même signature et mêmes opérations.

□ 2 Axiomes à ajouter :

$\forall i, j \in [\text{BorneInf}(s), \text{BorneSup}(s)],$

$i \leq j \Leftrightarrow \text{lème}(s,i) \leq \text{lème}(s,j)$

Changer $\text{lème}(s,i,e) = \text{erreur}$

si $e < \text{lème}(s,i-1)$ et $e > \text{lème}(s,i+1)$

Séquence de coordonnées

Sorte : SeqCoord = Seq [Elem :: Coord]

Opérations : (on ajoute une opération)

Compte(SeqCoord) \rightarrow Entier

Axiomes :

Compte(CréerSeq(i,j)) = 0 (toutes les cases sont à ε)

Compte(Changerlème(s,i,c)) =

si (lème(s,i) = ε) et (c \neq ε)

alors Compte(s) + 1

sinon si lème(s,i) \neq ε et c = ε

alors Compte(s) - 1

sinon Compte(s)

où ε symbolise la valeur vide

Implémentation d'une séquence

- Un tableau permet l'implantation d'une séquence. C'est le contenant.
- Un tableau à une dimension (= vecteur) est une structure de données formées de cellules contiguës et à accès direct.
- Pour définir un tableau, il faut préciser son identificateur, les types de I et E.

```
Const max = 30
```

```
Type tab-entier : tableau [1..max] d'entier
```

```
Var t : tab-entier
```

CréerSeq(i,j) : fait par la déclaration du type et de la variable ;
les initialisations restent à faire

BorneInf(t)	1
BorneSup(t)	30
lème(t,i)	t[i]
Changerlème(t,i,e)	t[i] ← e

Exemples

□ Lecture d'une séquence

```
lire(n)
Pour i←1 à n inc +1 Faire
    lire(t[i])
FinPour
```

□ Ecriture d'une séquence

```
Pour i←1 à n inc +1 Faire
    écrire(t[i])
FinPour
```

Séquence doublement indicée

- Définition

Soient un ensemble de valeurs E et deux ensembles finis I et J totalement ordonnés,

une séquence sur E est une application de $I \times J$ dans E

$$s : I \times J \rightarrow E$$

- A chaque couple (i, j) dans I et J , correspond un élément de E :
 $(i, j) \rightarrow s_{i,j}$

- Sert à modéliser une table à double entrée, souvent représentée par une matrice.

TAD SeqDouble

Sorte : SeqDouble

Utilise : Entier

Opérations :

CréerSeqDouble(Entier, Entier, Entier, Entier) → SeqDouble

Changerlème(SeqDouble, Entier, Entier, Elem) → SeqDouble

BorneInf(SeqDouble, Entier) → Entier

BorneSup(SeqDouble, Entier) → Entier

lème(SeqDouble, Entier, Entier) → Elem

Implémentation d'une séquence doublement indicée

- Un tableau à deux dimensions est une structure de données permettant l'implantation d'une séquence doublement indicée.
- Pour définir un tel tableau, il faut préciser son identificateur, le type de I, J et E.

```
Const maxi = 30
```

```
maxj = 50
```

```
Type mat-entier : tableau [1..maxi, 1..maxj] d'entier
```

```
Var m : mat-entier
```

CréerSeq(i,j,k,l) : fait par la déclaration du type et de la variable ;
les initialisations restent à faire

```
BorneInf(m, 1)      1
```

```
BorneSup(m, 1)     30
```

```
BorneInf(m, 2)     1
```

```
BorneSup(m, 2)     50
```

```
lème(m,i,j)        m[i,j]
```

```
Changerlème(m,i,j,e)  m[i,j] ← e
```

Exemples

- Lecture d'une séquence doublement indicée

```
lire(n)
lire(m)
Pour i←1 à n inc +1 Faire
    Pour j←1 à m inc +1 Faire
        lire(m[i,j])
    FinPour
FinPour
```

- Ecriture d'une séquence doublement indicée

```
Pour i←1 à n inc +1 Faire
    Pour j←1 à m inc +1 Faire
        écrire(m[i,j])
    FinPour
FinPour
```