Chapitre IV

SÉQUENCES, ENSEMBLES ET LISTES

1 Séquences

Définition. On appelle **séquence** *(expression sequence)*[1] une collection ordonnée d'expressions séparées par des virgules, et éventuellement entre parenthèses.

Une séquence : *exp1, exp2, ..., expn*

Exemples. *Effectuer :*

```
>3,4,5;
>s1:=a,b,c; s2:=(d,e);
>sequence:=s1,s2,f,g,h;
>seqvide:=NULL; # La séquence vide
>sequence[1],sequence[8];sequence[9];
>sequence[2]:=x; # Ceci est interdit
>sequence:=sequence[1],x,sequence[3..8];
```

Il faut bien comprendre que les séquences sont des structures en lecture seule. On ne peut pas modifier uniquement le $k^{ème}$ terme de la séquence : il faut redéfinir entièrement la séquence.

Modes de définition des séquences. On a vu que l'on peut définir une séquence directement en écrivant une suite d'expressions séparées par des virgules. On peut aussi faire appel aux opérateurs `seq` et `$` de la façon suivante :
Effectuer :

```
>seq(2^i,i=1..10);
>2^i $ i=1..10;
>j $ 5;
>seq(k^2,k=1..n); # Ceci conduit à une erreur
>k^2 $ k=1..n; # C'est mieux ainsi
>subs(n=5,%);
>eval(%);
```

2 Ensembles

Définition. On appelle **ensemble** *(set)* une collection non ordonnée d'expressions toutes différentes séparées par des virgules, notées entre accolades

Un ensemble : {*exp1, exp2, ..., expn*}

Exemples. *Effectuer :*

```
>{3,4,5};
>{5,4,3}; # Bien noter l'ordre
>{a,d,d,b,f,b,e,c,c};
>{seq(1/k,k=1..10)};
>ensvide:={}; # L'ensemble vide.
```

Y a-t-il un autre moyen de définir l'ensemble vide ?

1. *lit.* suite d'expressions

Opérations sur les ensembles. On peut effectuer sur les ensembles les opérations de réunion (`union`), d'intersection (`intersect`), de différence ensembliste (`minus`).
Un ensemble est avant tout une expression, au même titre que `a*x+b`. On peut en extraire les opérandes (on obtient alors une séquence des éléments) ou le nombre des opérandes (on obtient le cardinal) à l'aide des fonctions `op` et `nops`.
Effectuer :

```
>ens1:={1,2,3,4,5};ens2:={seq(2*k,k=0..3)};
>"Réunion"=ens1 union ens2;
>Intersection:=ens1 intersect ens2;
>`Différence de ens1 et ens2`:=ens1 minus ens2;
>op(ens1);
>nops(ens2);
```

3 Listes

Définition. On appelle **liste** *(list)* une collection ordonnée d'expressions séparées par des virgules, notées entre crochets.

Une liste : `[exp1,exp2, ... ,expn]`

Opérations sur les listes. *Effectuer :*

```
>li1:=[Alain,Bernard,Céline];
>li1[3];
>li2:=[Daniel,Élise];
>seq3:=Frédérick,Guillaume;
>li3:=[seq3];
>`Mauvaise liste`:=li1,li2,li3;
>`Bonne liste`:=[op(li1),op(li2),op(li3)];
>`Pour la frime`:=[seq(op(li||k),k=1..3)];
>li1[2]:=Bertrand; # Cette méthode est à éviter
>li1:=subsop(2=Bruno,li1); # C'est la bonne méthode
(voir l'aide en ligne)
>li1;
```

Différence entre les trois structures. Une liste diffère d'un ensemble parce que ses éléments sont ordonnés et que la répétition y est autorisée.
Une liste diffère d'une séquence par le fait qu'elle n'est constituée que d'une seule expression, tandis qu'une séquence est la juxtaposition des expressions que sont ses opérandes. Ce point est important pour l'utilisation de certaines fonctions.
Effectuer :

```
>restart;
>subs(x=1,y=2,3*x+2,5*y+4);
>subs(x=1,y=2,[3*x+2,5*y+4]);
```

Plus tard. On parlera au moment de l'algèbre linéaire de la structure de tableau, qui permet la modification d'un des éléments sans avoir à redéfinir tous les éléments.