

1. PROLEGOMENES

Le langage utilisé dans ce livre d'introduction à la programmation est LISP.

LISP est un acronyme de **LIS**t **P**rocessor.

Comme son nom l'indique, LISP traite des LISTES. Une liste est quelque chose qui commence par une parenthèse ouvrante "(" et qui se termine par une parenthèse fermante ")". C'est simple. Voici quelques exemples de listes :

(DO RE MI)
(OH LA LA)
(LE CUBE EST SUR LA TABLE)

Syntaxiquement une liste est définie comme :

liste ::= ({qqc₁ qqc₂ ... qqc_n })

et "qqc" est défini comme :

qqc ::= liste | atome

Le signe ::= veut dire *est défini comme*, et le signe | veut dire *ou*. Les choses entre accolades, { et }, sont optionnelles. Cela veut dire qu'il peut en avoir ou pas dépendant du cas où vous vous trouvez. La définition précédente se lit donc "qqc est défini comme une liste ou un atome".

Continuons nos définitions :

atome	::=	nombre nom
nombre	::=	0 1 2 3 ... 1024 ... -1 -2 ... -1024 ... (ce qui veut dire qu'il peut y avoir des nombres positifs ou négatifs)
nom	::=	suite de caractères alphanumériques contenant au moins une lettre et ne contenant aucun <i>séparateur</i>
séparateur	::=	. espace () [] ' tabulation ; retour chariot ce sont des caractères qui ont un rôle particulier

Quelques exemples d'atomes :

DO
RE
MI
CECI-EST-UN-ATOME-TRES-LONG
CUBE
CUBE1
1A

Voici donc quelques exemples d'objets LISP :

128 ; un nombre ;
-32600 ; un nombre négatif ;
HAHA ; un atome nom (atome alphanumérique) ;
() ; une liste à 0 élément ;
(HAHA) ; une liste à 1 élément ;
(UNE LISTE) ; une liste à 2 éléments ;
((UNE) LISTE) ; c'est une liste à 2 éléments, le premier ;
; élément est une liste à 1 élément, le ;
; deuxième élément est un atome ;
(UNE (LISTE)) ; encore une liste à deux éléments ;

Voici trois exemples de choses qui ne sont pas des objets possibles en LISP :

) ; rien ne peut commencer par une parenthèse fermante
(TIENS (TIENS (TIENS ; ici, il y'a un tas de parenthèses
; fermantes manquantes
(...) ; malheureusement, le caractère "point"
; n'est pas permis en LISP (pour l'instant)

Il est possible de compter les éléments d'une liste, on peut donc parler de la *longueur* d'une liste. La *longueur* d'une liste est le nombre de ses éléments.

Voici quelques listes de longueur 2 :

(1 2)
(TIENS TIENS)
((AHA) (ENCORE UNE LISTE))
(())

et voici encore quelques listes et leurs longueurs :

listes	longueur
()	0
(JOHN GIVES MARY A BOOK)	5
((SUJET)(VERBE)(OBJET))	3
((X + Y) + Z) --> (X + (Y + Z))	3
(QUELQU-UN AIME MARY)	3
(DU PASSE FAISONS TABLE RASE)	5
(1 2 3 A B C C B A 3 2 1)	12

La liste vide, (), s'appelle **NIL**. Pour LISP l'atome **NIL** et la liste () ont la même valeur, c'est-à-dire : (). Vous pouvez écrire soit sous la forme d'un atome : **NIL**, soit sous la forme d'une liste : ().

Visiblement, les parenthèses sont toujours bien équilibrées : une liste, et donc également chaque sous-liste, comporte le même nombre de parenthèses ouvrantes et fermantes. De plus, en lisant une liste, caractère par caractère, de gauche à droite, le nombre de parenthèses fermantes est toujours inférieur au nombre de parenthèses ouvrantes rencontrées. Excepté, naturellement, à la fin de la liste où le nombre de parenthèses ouvrantes est égal au nombre de parenthèses fermantes.

Si vous avez des problèmes pour déterminer si un objet représente une liste bien formée, voici un petit algorithme qui peut vous aider :

Si l'objet en question ne commence pas par une parenthèse ouvrante, bien évidemment, vous n'avez pas affaire à une liste et le tour est joué. Sinon, comptez les parenthèses de manière suivante : imaginez que vous ayez un compteur qui est initialement à zéro. Ensuite vous lisez l'objet de gauche à droite. A chaque rencontre d'une parenthèse ouvrante vous incrémentez votre compteur de 1 et vous marquez la parenthèse ouvrante par la valeur de votre compteur. Si vous rencontrez une parenthèse fermante, vous la marquez avec la valeur courante de votre compteur, ensuite vous le décrémentez de 1. Si, avant d'atteindre la fin de la liste vous obtenez (à un instant quelconque) un nombre inférieur ou égal à 0 dans votre compteur, alors l'objet n'est pas une liste bien formée. Sinon, si à la fin de l'objet votre compteur est égal à 0, alors l'objet est bel et bien une liste des plus correctes.

Voici quelques exemples d'application de cet algorithme :

- L'objet **((AHA) (ENCORE UNE LISTE))** peut être étiqueté comme suit :

$$({}_1({}_2({}_3\text{AHA})_3)_2}({}_2\text{ENCORE UNE LISTE})_2)_1$$

La dernière parenthèse est étiquetée '1' : cet objet est donc bien une liste.

- Essayons pour l'objet que voici : **(HEIN? (QUOI?))**

$$({}_1\text{HEIN?}({}_2\text{QUOI?})_2)_1)_0$$

Ici, il y a une parenthèse fermante en trop !

- finalement : **((X + Y) + Z) -> (X + (Y + Z))** donne

$$({}_1({}_2({}_3\text{X} + \text{Y})_3 + \text{Z})_2} \rightarrow ({}_2\text{X} + ({}_3\text{Y} + \text{Z})_3)_2)_1$$

Alors ? Est-ce une liste ?

Cette manière de numéroter les parenthèses permet également de reconnaître aisément les différents éléments d'une liste, les différents éléments des sous-listes, des sous-sous-listes, etc. Prenons, par exemple la dernière liste :

$$({}_1({}_2({}_3\text{X} + \text{Y})_3 + \text{Z})_2} \rightarrow ({}_2\text{X} + ({}_3\text{Y} + \text{Z})_3)_2)_1$$

C'est une liste à 3 éléments : les deux sous-listes entourées par les parenthèses étiquetées '2' et l'atome '->'. Pour être plus précis, le numérotage des parenthèses correspond à la *profondeur* de la liste : la liste **(X + Y)** se trouve à la profondeur 2, la liste **(X + (Y + Z))** se trouve à la profondeur 1, etc. Une liste entourée par des parenthèses numérotées *x* se trouve à la profondeur *x-1*, et tous les éléments à l'intérieur de cette liste se trouvent à la profondeur *x*.

1.1. EXERCICES

1. Déterminez pour chacun des objets ci-dessous s'il s'agit d'une liste, d'un atome ou d'un objet inadmissible en LISP :

123
(EIN (SCHO%NES) BUCH)
(AHA (ETONNANT . . . !))
(((1) 2) 3) 4) 5)
-3Aiii
T
((((ARRRGH))))

2. Déterminez le nombre d'éléments des listes suivantes :

(EIN (SCHO%NES) BUCH)
(((1) 2) 3) 4) 5)
(((ARRRGH))))
(UNIX (IS A) TRADEMARK (OF) BELL LABS)

3. Pour chacune des listes ci-dessus, donnez l'élément de profondeur maximale et sa profondeur.