

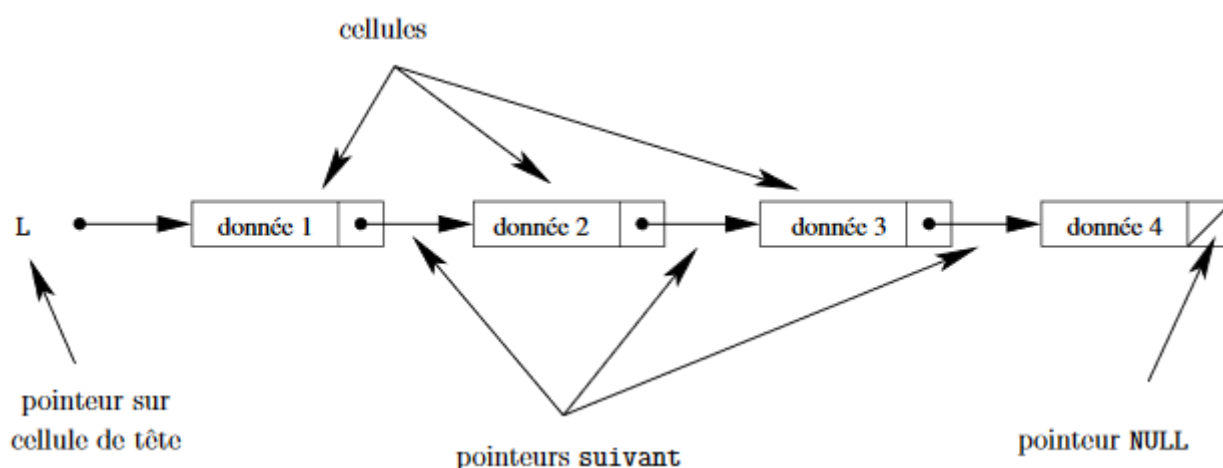
LISTES CHAÎNÉES

1. Introduction

Une liste chaînée est un ensemble de cellules liées entre elles par des pointeurs. Chaque cellule est une structure contenant les champs suivants :

- une ou plusieurs données comme dans n'importe quelle structure ;
- un pointeur suivant sur la cellule suivante.

On accède à la liste par un pointeur L sur la première cellule, puis en parcourant la liste d'une cellule à l'autre en suivant les pointeurs suivant . Le dernier pointeur suivant vaut NIL , ce qui indique la fin de la liste.



Exemple de liste chaînée avec 4 cellules

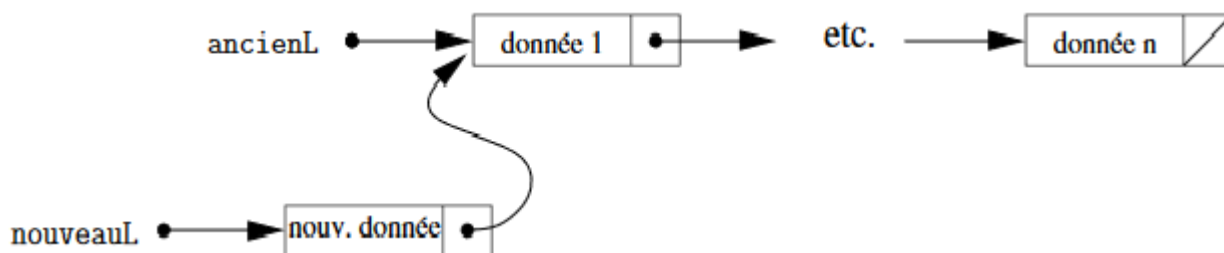
2. Déclarer une Liste Chaînée

Pour créer une liste chaînée, il faut déclarer une nouvelle structure de données : la structure qui représentera une cellule.

```
{ définition du type cellule }
Cellule : enregistrement
debut
    donnee : type_donnée    { définition des données }
    suivant : *Cellule      { pointeur sur élément suivant }
fin
```

3. Insertion en Tête de Liste

La fonction suivante prend en paramètre une liste et une donnée, et ajoute la donnée en tête de liste. La fonction renvoie la nouvelle adresse de la tête de liste.



Fonction InsereEnTete(ancienL : *Cellule, donnee : TypeDonnee) : *Cellule

variables :

nouveauL : *Cellule { nouvelle tête de liste }

debut

nouveauL->donnee := donnee { on met la donnée à ajouter }

nouveauL->suivant := ancienL { chaînage */

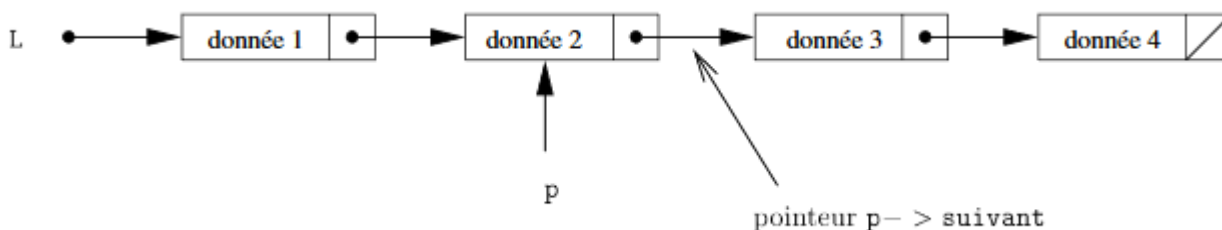
InsereEnTete := nouveauL { on retourne la nouvelle tête de liste }

fin

Ne pas confondre l'utilisation du point . et l'utilisation de la flèche -> pour accéder aux champs d'un enregistrement. On utilise le point pour une variable de type enregistrement, et une flèche pour une variable de type pointeur sur enregistrement.

4. Parcours de Liste

L'idée du parcours de liste chaînée est de prendre un pointeur auxiliaire p. On fait pointer p sur la première cellule, puis le pointeur p passe à la cellule suivante (par une affectation p = p->suivant). Le parcours s'arrête lorsque p vaut le suivant de la dernière cellule, c'est-à-dire lorsque p vaut NULL.



procedure Affichage(liste : *Cellule)

debut

p : *Cellule := liste { on pointe sur la première cellule }

tanque p <> NULL faire

debut

Afficher(p->donnee)

p := p->suivant { on passe à la cellule suivante }

fin

fin

Lors du parcours, on s'arrête lorsque p vaut NULL, et non pas lorsque p->suivant vaut NULL. En effet, p->suivant vaut NULL lorsque p pointe sur la dernière cellule. Il faut traiter cette

dernière.