Structures de données et Algorithmes

Projet : Dictionnaire des synonymes

1. Présentation du problème

Dans ce TP, on se propose de mettre en œuvre un petit dictionnaire des synonymes. A un mot donné, on veut associer un ensemble de mots représentant ses différents synonymes. Par exemple, les synonymes du mot "voiture" pourront être représentés par un ensemble contenant (entre autres) les mots "automobile", "bagnole", "fiacre", "guimbarde", "véhicule", "wagon".
Le programme final devra être capable notamment de :

- lister les synonymes d'un mot donné;
- tester si un mot donné est synonyme d'un autre;
- déterminer quels mots sont synonymes de plusieurs mots à la fois.

2. Choix techniques et contraintes

- L'ensemble des synonymes d'un mot donné sera représenté sous la forme d'un arbre AVL, de même que le dictionnaire lui-même. La relation d'ordre utilisée sera l'ordre lexicographique.
  Un nœud du dictionnaire correspond à un mot, une valeur d'équilibre, l'ensemble des synonymes du mot, un pointeur sur le sous-arbre droit, un pointeur sur le sous-arbre gauche.
- Le h-équilibre sera maintenu à chaque intervention.
- Un ensemble de synonymes ne doit pas contenir pas de doubles.

3. Organisation générale du programme

Comme pour les TP précédents, il est naturel de structurer le programme en plusieurs modules. On définira deux modules dits "génériques" :

- un module pour la structure de donnés "ensemble de synonymes" :

```
unité interface sdd_Synonymes c'est
    type T_Syn;
    procédure S_init_vide (var T_Syn s);
    procédure ajout_synonyme (var T_Syn s; T_Mot m);
    fonction appartient_à (T_Mot m; T_Syn s) résultat bool;
fin;
```

- un module pour la structure de données "dictionnaire de synonymes" :

```
unité interface sdd_Dico c'est
    type T_Dico;
    procédure D_init_vide (var T_Dico d);
    procédure D_ajout_entrée (var T_Dico d; T_Mot mot; T_Syn s);
    procédure charger_dico (chaîne_de_caractères nom_fichier; var T_Dico d);
    fonction liste_syn (T_Dico d; T_Mot mot) résultat T_Syn;
    fonction est_synonyme_de (T_Dico d; T_Mot mot1, mot2) résultat bool;
    procédure synonymes_communs (T_Dico d; T_Mot mot1, mot2) résultat T_Syn;
fin;
```

4. Travail à réaliser

1. Programmer en C les unités “sdd_Synonymes” et “sdd_Dico” en fonction des choix techniques et contraintes exprimés.

2. A l’aide de ces structures de données, écrire le programme principal de l’application, dont le but est de tester la mise en œuvre de ces deux modules sous forme de menu permettant à l’utilisateur de manipuler facilement le dictionnaire. Des ensembles de synonymes vous sont fournis pour effectuer vos tests (cf. annexes B.1 et B.2).

A. Annexe technique

A.1. Ensembles fournis

Un fichier de synonymes vous est fourni sous le répertoire /users/etc/tmp/pivert/sdd/TP6, pour effectuer la mise au point de votre réalisation, et répondre à la séquence de tests. Son nom est base_synonymes.txt. Il contient la liste des synonymes de plusieurs mots (une centaine d’entrées), et est divisé en autant de blocs. Chaque bloc est introduit par le mot clé N_ENT, la ligne suivante contenant l’intitulé de la nouvelle entrée, et les lignes suivantes la liste des synonymes de ce mot, jusqu’au bloc suivant.

A.2. Séquence de test

Selon la base de synonymes fournie :

1. Quels sont les synonymes :
   (a) du mot “savon”?
   (b) du mot “crêpe”?
   (c) du mot “voiture”?
2. Les mots suivants sont-ils des synonymes du mot “dragon”?
   (a) “argoulet”?
   (b) “diablesse”?
   (c) “dinosaure”?
   (d) “monstre”?
3. Quels sont les synonymes communs à :
   (a) “rouge” et “jaune”?
   (b) “homme” et “femme”?
   (c) “ventripotent” et “espagnol”?
   (d) “bidule” et “machin”?
   (e) “chat” et “chien”?