Arbre lexical fils gauche- frère droit

Dans toute la suite, on appelle mot une suite d'au plus 50 caractères ne contenant pas de séparateurs (chaine au sens scanf("%51s"...)). Les accents sont donc significatifs. ("Interne" et "Interné" n'ont pas le même statut).

On appelle **lexique** d'un texte l'ensemble des mots qui apparaissent dans le texte.

1 Représentation du lexique par arbre binaire sous forme fils gauche, frère droit

On représentera le lexique à l'aide d'un arbre dans lequel chaque nœud possède 3 champs :

- un caractère
- 2 pointeurs filsg, frered

avec la structure C

```
typedef struct noeud{
unsigned char lettre;
struct noeud *filsg, *frered;
}Noeud, *Arbre;
```

Exemple:

l'arbre lexical de la figure 1 reconnait l'ensemble de mots {ce,ces,des,le,les,lettre,mes,mettre}.

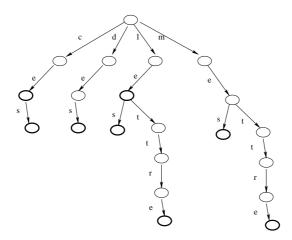


Figure 1: Arbre lexical

L'arbre sera construit sous la forme d'un arbre binaire fils gauche, frère droit, les nœuds frères étant ordonnés en ordre croissant voir figure 2. Un nœud pointe vers le premier de la liste de ses fils et vers son frère suivant.

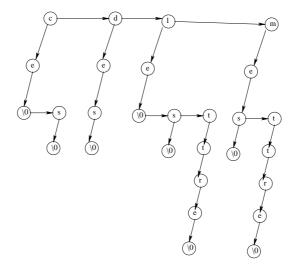


Figure 2: Arbre lexical forme fils gauche-frère droit

2 Fonctionnalités

Avec cette structure, construire le lexique d'un texte contenu dans un fichier fournit comme argument de la ligne de commande. L'exécutable aura pour nom Lexique et nécessitera toujours un argument : le nom du fichier à traiter.

Les fonctionnalités suivantes devront être implantées et accessibles par un menu:

- 1. afficher les mots du lexique en ordre alphabétique;
- 2. sauvegarder les mots du lexique en ordre alphabétique. Le nom du fichier de sauvegarde est le nom du fichier d'entrée suivi du suffixe(.L);
- 3. indiquer si le mot est présent dans le texte sous forme present ou absent;
- 4. sauvegarder l'arbre dans un fichier, le nom du fichier de sauvegarde est le nom du fichier d'entrée suivi du suffixe(.DICO)(format décrit plus loin);

Ces fonctionnalités seront également accessibles via une option de la ligne de commande (les options sont mutuellement exclusives):

- 1. Lexique -l nom affiche les mots du lexique du fichier nom en ordre alphabétique;
- 2. Lexique -s nom sauvegarde les mots du lexique du fichier nom en ordre alphabétique. Le nom du fichier de sauvegarde est le nom du fichier d'entrée suivi du suffixe(.L);
- 3. Lexique -r Mot nom indique si Mot apparaît dans le fichier nom

4. Lexique -S nom sauvegarde l'arbre. Le nom du fichier de sauvegarde est le nom du fichier d'entrée suivi du suffixe(.DICO)(format décrit plus loin).

L'option -S force la reconstruction à partir du fichier contenant le texte et la création d'un nouveau fichier de sauvegarde .DICO.

Pour les autres options, si un fichier de sauvegarde .DICO existe, le lexique sera reconstruit à partir de cette sauvegarde et non en traitant à nouveau le fichier contenant le texte. La cohérence des deux fichiers ne sera pas vérifiée.

En l'absence d'option sur la ligne de commande ces fonctionnalités seront accessibles via un menu.

Format de sauvegarde

Le fichier de sauvegarde sera construit par un affichage préfixe des nœuds de l'arbre. Puisque les caractères séparateurs ne sont pas utilisés comme étiquette des nœuds, on indique un sous arbre vide avec '\n' et on remplace le caractère '\0' (non imprimable) par un espace ''. Puisque le nœud d'étiquette '\0' n'a jamais de fils, seul le frère droit est codé.

L'arbre 2 est sauvegardé par la suite:

où '\n' désigne le passage à la ligne.

3 Rapport et date de remise

Le devoir est à réaliser en binome.

Chaque binome doit remettre une archive de nom nomauteur1nomauteur2.tar.

Le fichier contenant le code source aura pour nom Lexique.c. La première ligne du fichier Lexique.c est un commentaire rappelant le nom des auteurs.

Il sera compilable sur les machines de l'université par la commande:

```
gcc -ansi -Wall Lexique.c -o Lexique.
```

La remise devra être effectuée sur le site de elearning.univ-mlv.fr/ avant le lundi 26 mars, 23H59. Un rapport indiquant les choix de programmation, au format pdf, devra être joint.

Les sources devront être compilables sous Linux, sur les machines de l'université.

Une soutenance pourra être organisée.

4 Améliorations

Vous pouvez proposer des améliorations mais elles ne seront prises en compte que si la totalité des fonctionnalités précédentes a été correctement implantée.