

Requêtes sur un dictionnaire

On désire ranger les mots d'un dictionnaire dans un arbre binaire à raison d'un caractère et un booléen par nœud. Pour un nœud donné Nœud, Car(Nœud) fournit le caractère et Bool(Nœud) fournit le booléen. Le booléen est vrai si le nœud est final (dernier caractère d'un mot), autrement il est à faux.

L'arbre est défini comme suit:

(i) \forall le nœud N si Fd(N) \neq Nil, alors Car(N) < Car(Fd(N))

(ii) \forall la branche N_1, N_2, \dots, N_n , elle renferme les mots obtenus comme suit :

i := 1; Mot := Car(N₁); Si Bool (N_i), écrire (mot)

Pour i:=1, n-1

(a) Si N_{i+1} = fg(N_i)

Mot := Mot + Car(N_{i+1})

Si Bool (N_{i+1}), écrire(Mot)

(b) Si N_{i+1} = fd(N_i)

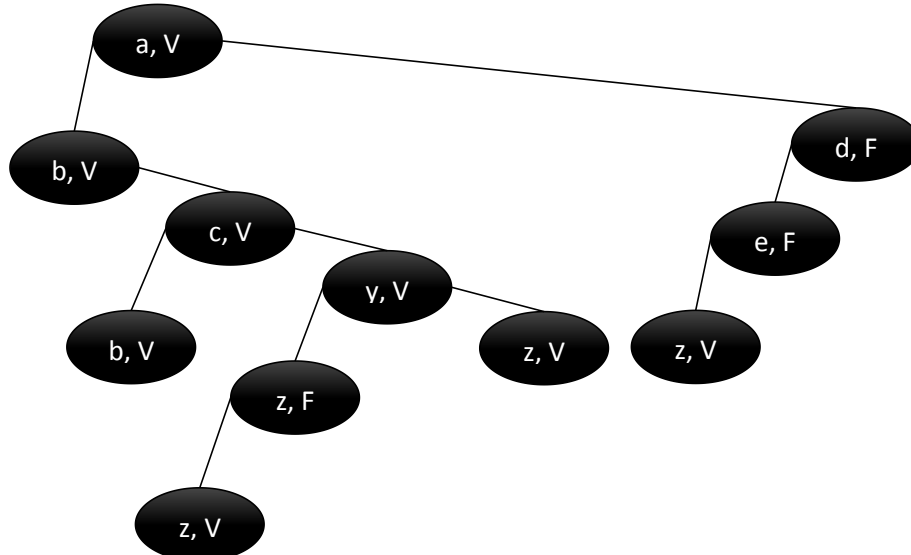
Mot := Mot_1 + Car(N_{i+1})

// Mot_1 désigne Mot sans le dernier caractère

Si Bool (N_{i+1}), écrire(Mot)

Finpour

La figure suivante montre un tel arbre avec les mots suivants insérés dans cet ordre : 'dez', 'ac', 'acb', 'ay', 'a', 'az', 'ab', 'ayzz'.



Dans l'arbre dressé Il y a 4 branches. La branche { (a, V), (b,V), (c,V), b, V } contient les mots 'a', 'ab', 'ac' et 'acb'. La branche { (a, V), (d,F), (e, F), (z, V) } contient le seul mot 'dez'.

Partie Khawarizm

Exprimez tous les modules nécessaires puis les mettre en œuvre sous l'environnement Khawarizm en considérant l'algorithme principal suivant:

Pour i :=1, n

- Générer un arbre avec m mots aléatoires de longueur entre 1 et L

- Vérifier la compatibilité de l'arbre (en affichant tous les mots)

- Trouvez les mots de la forme $CAAA...A$ où C désigne un caractère donné et A désigne soit un caractère donné, soit un caractère quelconque (noté ?) soit n caractères ($n \geq 0$) quelconques (notés *)

Pour

Prenez $m = 15$, $n=10$, $L = 10$.

Partie C

(a). Demander la traduction automatique vers C et tester le programme résultant.

Prenez $m = 1000$ et $n=100$. Vous pouvez les augmenter.

(b). Sur le programme C obtenu :

- * Rajouter un module qui construit l'arbre à partir d'un dictionnaire ou d'une encyclopédie

- * Formuler des requêtes de mots en exhibant des fenêtres glissantes montrant les résultats. Vérifiez préalablement leur syntaxe.

Note importante : Le dictionnaire ou l'encyclopédie est un fichier TXT qui doit être référencé dans le programme **sans le chemin**, ce qui implique qu'il doit figurer dans le répertoire du programme.

N.B

1. Sachez qu'une branche est un chemin de la racine vers la feuille.

2. L'arbre est déclaré comme suit : $ARB DE (CAR , BOOLEEN)$

3 On utilisera les fonctions $Longchaine(S)$ qui donne la longueur du mot (S) et $Caract(S, I)$ qui accède au $I^{ème}$ caractère du mot S.

4. $Aleachaine(Aleanombre(L) + 1)$ génère une chaîne de caractères entre 1 et L caractères.

5. Exemples de requête : $m?f^*$; $x*y$

Notation

Khawarizm : 15 pts

C : 5 pts

Passage au langage C

Il est peu probable que lors de la traduction, le programme obtenu ne tourne pas. Dans ce cas, l'étudiant devra faire l'effort pour chercher et corriger l'erreur. Les principales erreurs sont dues principalement aux initialisations implicites de variables dans l'environnement Khawarizm.

Toute erreur reconnue pertinente et corrigée par l'étudiant sera récompensée par un Bonus. Dans ce cas, nous envoyez le code Z pour lequel la traduction ne fonctionne pas et le code C corrigé en nous indiquant les changements. Si une erreur de traduction ne peut être corrigée, veuillez nous envoyer le code Z. L'erreur sera corrigée pour vous.

Attention au plagiat !!!

Date de remise : 31 Juillet 2019 avant Minuit

* Les TP seront remis selon un formulaire qui vous sera transmis par la suite

* Tout TP non remis selon l'échéancier fixé entraîne un retrait de 2 points par jour sur un total de 3 jours tolérés.

*Un TP non fait entraîne un retrait de 2 points sur la moyenne générale!