

LOG2810 : mathématique discrètes

TP2 : AUTOMATE D'ARBRE

Bouchra Nour El yakin Dahamni 1909765

Mamoudou sacko 1924187

Fabrice Ndui 1914377

04/12/2018

Introduction :

Le but de ce TP est pratiquer les notions théoriques des automates finis qu'on a vu en cours, donc dans ce lab on va créer un automate fini qui représente un langage basé sur un lexique. Cet automate est capable de donner des suggestions de mots à l'utilisateur une fois il rentre une lettre précise, autrement dit la lettre représente l'arc de notre automate.

Explication de notre solution :

- 1) On a notre lexique qui contient plusieurs mots, au début on choisit le mot le plus long du lexique.
- 2) Une fois qu'on a le mot on commence une boucle de 0 jusqu'à la longueur de ce mot
- 3) On se fixe un tableau qui contient le prochain état à traiter, ce tableau est rempli à la fin de chaque itération.
- 4) Pour que chacun des états dans un tableau on crée les branches
La création de ces branches consiste, à prendre les lettres distinctes qui correspondent à notre position, on filtre cette branche en tenant compte de sa branche précédente.
- 5) Pour chaque branche résultante de l'étape précédente, on crée le prochain état, on associe le prochain état et l'état courant (comme un arbre) dans les deux sens.
- 6) On associe à l'état courant les indices du tableau de mots où on a trouvé cette lettre
- 7) On ajoute l'état courant dans notre liste d'états de lexique.
- 8) On de nouveau le tableau par chaque prochain état créé et on recommence l'itération se basant sur ce tableau.

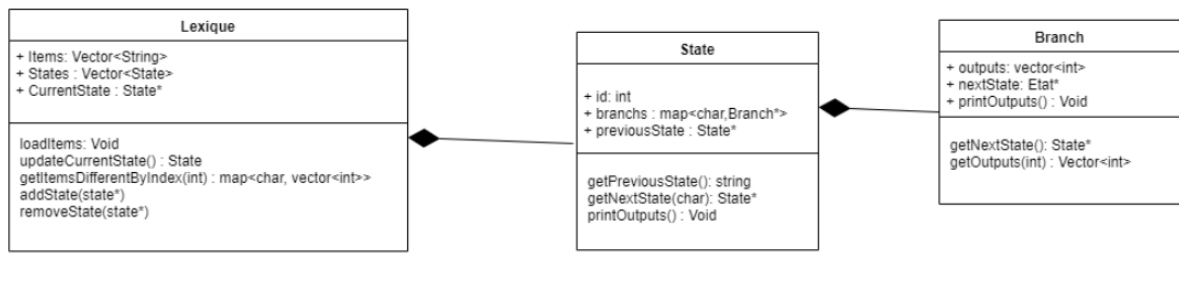
Utilisation de l'automate :

- 1) On met à jour notre état courant à l'état initial.
- 2) À partir de notre interface, on récupère ce que l'utilisateur a saisi.
- 3) On essaye de trouver une branche qui contient le mot saisi à partir de l'état courant.
 - 3.1) si on trouve une branche : Dès qu'on trouve une branche, on affiche le résultat associé à l'état suivant via cette branche.
On met à jour l'état courant.
 - 3.2) si on ne trouve pas de branche, on le met en prison et il resettera là-bas jusqu'à qu'il revient à ses séquences de lettres juste avant qu'il rentre en prison, là il sort c'est comme si on a créé un mot de passe (tant qu'il est en prison on affiche rien)
- 4) et on continuera à itérer tant qu'il rentre des lettres valides.

Difficultés rencontrées dans ce laboratoire :

- 1) Trouver un algorithme qui charge un automate fini pour chaque lexique dynamiquement.
- 2) Implémenter l'interface avec Qt Creator.
- 3) Finir le lab dans les délais avec la charge de travail de la fin de session et la date de remise qui est proche de notre examen final.

Diagrammes de classe :



Conclusion :

Dans ce laboratoire, on a pu écrire du code qui nous permet de pratiquer sur les automates finis et leur utilité. Il nous a permis de renforcer nos connaissances en langage de programmation C++ et en Qt Creator. De développer notre esprit de travail en équipe et de partager les tâches entre les membres avec équilibre. En ce qui concerne l'utilité de ce lab, nous avons d'abord compris comment implémenter un automate en C++ et des algorithmes pour les langages. De plus, nous avons appris à nous servir de quelques nouvelles fonctionnalités de Qt Creator.

Pour nos projets dans le futur, nous voudrions envisager d'implémenter des options qui permettraient de souligner en rouge les mots mal orthographiés comme en Word. Par la suite, nous voudrions tenter d'analyser la syntaxe correcte de la phrase écrite par l'utilisateur pour corriger certains erreurs d'orthographe.