Faculté des Sciences
Aix*Marseille Université

Année Universitaire 2017/2018

Licence d'Informatique (L3)
SIN5U1TL – Algorithmique avancée

Devoir n°1 – Algorithme d'Aho et Corasick

Description rapide de l'algorithme d'Aho et Corasick

L'algorithme d'Aho et Corasick permet d'identifier toutes les occurrences de p mots-clés dans un texte de N caractères, et ce en parcourant « simplement » ce texte une fois (détection « à la volée »). Pour cela, une phase de « prétraitement » est nécessaire : cette phase va produire un automate à nombre fini d'états qui permettra, lors de la lecture du texte, de trouver à partir de chaque état (« chaîne(s) en cours de détection ») et de la transition « détectée » (« caractère courant »), ce que sera le prochain état. Cet algorithme a été publié en juin 1975 et vous trouverez une copie de la publication originale sur la page associée à ce devoir. Il est toujours d'actualité puisqu'il est à la base de la recherche de mots-clés dans de nombreux logiciels comme, par exemple, le grep d'UNIX.

L'essentiel de l'algorithme consiste en la création de l'automate à nombre fini d'états. Ensuite, il suffira de partir de l'état 0 (« aucune chaîne n'est en cours de détection ») et de lire séquentiellement chaque caractère pour trouver le nouvel état, voir si des détections de mots-clés lui sont associées et afficher les résultats correspondants.

L'automate à nombre fini d'états est associé à un graphe orienté dont les sommets sont les états (chaîne(s) en cours de détection) et les arcs sont les transitions (nouveau caractère). Dans un premier temps, on construit une arborescence dont l'état 0 (aucune chaîne détectée) est la racine et où chaque parcours depuis la racine jusqu'à un sommet donné correspond à une sous-chaîne de mot(s)-clé(s) partant du début de ces derniers. Dans un deuxième temps, on complète cette arborescence avec de nouveaux arcs pour former un graphe orienté : les arcs ajoutés correspondent aux possibilités d'autres sous-chaînes (plus courtes) détectées lorsque l'on ne peut plus évoluer directement dans l'arborescence ; pratiquement, on recherche le plus grand suffixe de la chaîne associée à l'état courant qui soit aussi un préfixe de mot(s)-clé(s), ce qui nous donne un tableau des « échecs » (en cas d'échec dans le parcours de l'arborescence, nous pouvons éventuellement repartir sur une autre branche) ; les transitions à partir des sommets correspondant à ces « positions d'échec » nous fournissent alors les arcs complémentaires.

Travail demandé

On demande de mettre en place la structure de données adéquate (en pensant à tous les éléments qui seront nécessaires à la mise en œuvre de l'algorithme) et de programmer cet algorithme. La programmation de la phase de prétraitement constituera l'essentiel du travail à effectuer. La lecture du fichier à analyser et l'identification des mots-clés fait partie aussi du travail à effectuer mais il pourra se limiter à une « ergonomie de base », ou bien, pour ceux qui sont plus à l'aise (et cela sera apprécié), au développement d'une interface plus conviviale (mots-clés soulignés ou colorés dans le texte, …). Des jeux de données se trouveront sur la page du devoir.

Conditions du devoir

Les devoirs sont effectués en binôme. Il est formellement interdit copier des « bouts de code » sur Internet ou sur ce qu'ont fait d'autres étudiants. Lors de la dernière séance de TP associée à ce devoir (4^{ième} séance, soit TP n°5 pour celui-ci), chaque binôme aura à présenter son travail sur son poste de travail (cela ne consistera pas à faire simplement tourner son programme … et s'il y a des « copies », cela sera flagrant à ce moment-là). Chaque binôme devra envoyer au responsable de l'U.E. (Jean Sequeira) un dossier dont le nom est la concaténation des deux noms avec une majuscule au début de chacun d'eux (e.g. SequeiraTartempion) et qui contiendra le programme et un rapport en pdf de quelques pages donnant des indications sur la structuration du programme, les raisons des choix effectués, les difficultés rencontrées, éventuellement des résultats (mais pas une description de l'algorithme ni des « bouts de code » ni des fichiers de résultats !) – ce dossier sera de type .zip (Mac) ou .rar (PC) (si vous avez un souci avec ça, envoyez les fichiers en documents attachés) et ne devra pas « peser » plus de 2Mo. Le choix du langage de programmation est libre.