

Descriptif des algorithmes

Machine n°1)

Cette machine permet de reconnaître si un mot est bien un mot de Dyck. Avec cette machine on ne s'occupe pas de la complexité de l'algorithme. Puisque la complexité est négligé, Savoir si un mot appartient à Dyck revient à compter son nombre de parenthèses ouvrantes et fermantes et vérifier que ces 2 nombres sont bien les même.

La machine va donc parser le mot qu'on lui donnera. Pour chaque lettre 'O' qu'il va trouver, une lettre 'c' va être rajouté à la fin du mot. Dans le cas où la lettre est 'F' est trouvé, un 'c' sera enlevé de la fin du mot. Le mot sera bien parenthésé si à la fin de l'algorithme aucun 'c' n'est trouvé (preuve que le nombre de parenthèses ouvrantes et fermantes sont bien égales). Dans le cas où l'on trouve ne serait que 1 'c' à la fin du mot ou alors que durant l'application de l'algorithme on cherche à effacer un 'c' qui n'existe pas, Alors le mot n'est pas bien parenthésé.

On se retrouve donc avec une machine de complexité $O(n^2)$ puisqu'on doit parser le mot en entier, puis rajouter/enlever un 'c' à la fin du mot avant de reprendre le parsing à la lettre suivante.

Machine n°2)

Dans le cas de cette machine on essaye de trouver la complexité la plus faible possible. Plutôt que de chercher à compter le nombres de 'O' ou 'F' on va s'intéressait à compter les paires de 'O' et 'F'.

On va commencer à parser le mot tout en cherchant les 'O'. Dès qu'un 'O' est trouvé il sera remplacé par 'o' puis on va chercher dans la suite des lettres du mots un 'F' que l'on va lui aussi remplacé par 'f' avant de retourner à l'endroit où on s'était arrêté. On recommence l'opération jusqu'à ce que l'on tombe sur un caractère '#', puis on parcourra le mot dans le sens inverse pour vérifier si toutes les lettres ont bien été appariés (vérifier qu'il n'y a que des 'o' et des 'f').

On se retrouve donc avec une machine de complexité $O(n \log n)$

Machine n°3)

Dans ce cas-ci, le problème n'est pas de trouver si le mot est bien parenthésé mais de trouver quelles lettres il faut enlever de manière à trouver un mot bien parenthésé. Le principe de base de la machine 2 est repris. Nous allons tenter d'apparier le maximum de 'O' 'F' possible et une fois arrivé à la fin du mot il suffira de parcourir le mot pour enlever ceux qui n'ont pas de pair.

On se retrouve donc avec une machine de complexité $O(n^2)$