

UE ALGO6 — TD2 — Séance 1 : Recherche de motif

L'objectif est de rechercher toutes les occurrences d'un motif, un mot ou un groupe de mots, dans un texte. On suppose que le texte est réalisé par un tableau $T[1..n]$, le motif par un tableau $M[1..m]$, formés de caractères pris dans un alphabet fini, par ex. $\{a, b, c, d, \dots, z\}$. Par exemple, le motif *abaab* est présent dans le texte *ababcabbaabaabbac* à la position 10.

Question 1. Soit l'algorithme de recherche suivant :

```
Recherche_Motif(T, M)
    { n est la longueur de T }
    { m est la longueur de M }
    pour i dans [0..n-m] faire
        si M[1..m]=T[i+1..i+m] alors afficher ("le motif apparaît à la position " s+1)
```

Si on mesure le coût en nombre de comparaisons entre caractères, quel est le coût de cet algorithme, en fonction de n et m ?

Question 2. Que pensez-vous de l'implémentation suivante, en Java, de cet algorithme ?

```
void rechercheMotif(String texte, String motif) {
    for (int i=0; i ≤ texte.length()-motif.length(); i++) {
        if (texte.substring(i,i+motif.length()).equals(motif)) {
            System.out.println("le motif apparaît à la position " + i);
        }
    }
}
```

Question 3. Essayant de diminuer le nombre d'opérations, on s'aperçoit qu'il suffirait pour chaque caractère de T lu de connaître le plus long suffixe lu qui est un préfixe de M (au lieu de recommencer à chaque fois toutes les comparaisons). Dans ce but, on construit un automate d'états finis, destiné à reconnaître les préfixes de M . On rappelle qu'un automate est défini par :

- un ensemble fini d'états Q
- un état initial q_0 élément de Q
- un sous-ensemble A de Q formé des états terminaux
- un alphabet fini S
- une fonction $d : Q \times S \rightarrow Q$ dite fonction de transition

Un automate qui reconnaît un motif M de longueur m possède $m + 1$ états, numérotés de 0 à m , 0 désignant l'état initial, m l'état terminal ; cet automate démarre à l'état $q_0 = 0$, et lit un par un les caractères du texte d'entrée ; lorsque l'automate est dans un état q , et reçoit en entrée un caractère x , il passe dans l'état $d(q, x)$; le motif est reconnu lorsque l'automate arrive à l'état terminal m .

La fonction de transition vérifie la propriété : Si $d(q, x) = k$ alors $M[1..k]$ est suffixe de $M[1..q].x$.

Dessiner un automate qui reconnaît l'exemple précédent, avec l'alphabet $\{a, b, c\}$; donner la table de la fonction de transition. Pour l'état $q = 3$, préciser pour chacun des caractères possibles x le plus grand k tel que $M[1..k]$ soit suffixe de $M[1..q].x$.

Question 4. Écrire une fonction qui calcule la table de la fonction de transition d'un motif $M[1..m]$. Écrire un algorithme qui recherche toutes les occurrences de ce motif dans un texte $T[1..n]$. Quel est le coût de ce nouvel algorithme ?