FEUILLE D'EXERCICES N°9 DE L'OPTION D'INFORMATIQUE.

Thème : Recherche de motifs.

0. Introduction.

Dans tout le problème, X désigne un alphabet fini, M,T des mots (respectivement appelés «motif» et «texte») de longueur respective m, n, la $k^{ème}$ lettre des mots M, K étant respectivement M_k, T_k.
On dira que M apparaît dans T en position s si $M = T_sT_{s+1}\dots T_{s+m-1}$.
Dans les questions où sont demandées des simulations d'algorithmes on prendra

$$X = \{a, b, c\}; \qquad M = abbaabb.$$

Les termes préfixe, suffixe sont des synonymes pour facteur gauche, facteur droit.

I Méthode itérative

Cette méthode consiste simplement à comparer les facteurs formés par m lettres consécutives de T au mot M.

I.1. En représentant M, T par des variables M, T de type `string` en `Caml` écrire `chercher(M,T)`, fonction `Caml` retournant la première position de M dans T (si elle existe), -1 si elle n'existe pas.

I.2. Évaluer la complexité de votre programme en supposant que seul compte le temps passé à comparer des lettres.

II Utilisation d'un automate

II.1. Proposer, pour le cas particulier du préambule, un automate fini déterministe reconnaissant le langage X^*M et montrer comment on peut utiliser cet automate pour déterminer l'existence de M dans chacun des textes

$$T = cabcabbaabbcba; \qquad T = abbcaabbca.$$

II.2. Revenant au cas général, en supposant que $\mathcal{A} = (X, Q, p_0, \{p_1\}, \delta)$ est un automate fini déterministe à m' états reconnaissant le langage X^*M, proposer un algorithme (ou programme) permettant la reconnaissance du motif M dans T.

Quelle est la complexité de votre programme?

III Utilisation d'une fonction de saut

Pour $k \in \{1, \dots, m\}$ on définit l'entier $f(k)$ - f est dite *fonction de saut* - tel que

- s'il existe un entier s tel que $s < k$ et $M_1 \dots M_s$, suffixe de $M_1 \dots M_k$, $f(k)$ est le plus grand de ces entiers,
- sinon, $f(k) = 0$.

En particulier, $f(1) = 0$.

III.1. Pour l'exemple du préambule, donner la table de la fonction f.

III.2. En examinant la situation où un préfixe de M de longueur $k < m$ a été identifié dans T, expliquer comment utiliser la fonction de saut pour un nouvel essai.

III.3. On considère l'algorithme `fonction-de-saut`

$g(1) \leftarrow 0$
  POUR j allant de 2 jusque m FAIRE
  $i \leftarrow g(j-1)$
  TANTQUE $(M_j \neq M_{i+1})$ ET $(i > 0)$ FAIRE $i \leftarrow g(i)$ FINFAIRE
  SI $(M_j \neq M_{i+1})$ ET $(i = 0)$
    ALORS $g(j) \leftarrow 0$
    SINON $g(j) \leftarrow i + 1$ FIN de SI
FINFAIRE

III.3.1. Montrer que cet algorithme est valide et que la fonction g calculée est égale à f. Écrire la fonction `Caml` associée.

III.3.2. Montrer qu'il est de complexité $O(m^2)$.

En utilisant N_j, nombre d'exécutions de la boucle `TANTQUE` lors de la $j^{ème}$ itération de la boucle `POUR`, montrer qu'en fait la complexité est $O(m)$.

III.4. Soit $\mathcal{A} = (X, Q, 0, \{m\}, \delta)$ avec $Q = \{0, \dots, m\}$ et δ défini par l'algorithme

POUR i allant de 1 jusque m FAIRE $\delta(i-1, M_i) \leftarrow i$ FINFAIRE
POUR chaque lettre x de $X \setminus \{M_1\}$ FAIRE $\delta(0, x) \leftarrow 0$ FINFAIRE
POUR i allant de 1 jusque m,
    POUR chaque lettre x de $X \setminus \{M_{i+1}\}$ FAIRE $\delta(i, x) \leftarrow \delta(f(i), x)$

III-4.1. Proposer une légère modification de cet algorithme permettant de l'utiliser sans provoquer une erreur d'exécution.

III-4.2. Faire le diagramme de $\mathcal{A}$ pour l'exemple du préambule et comparer à votre réponse donnée en (**II.1.**).

III-4-3. Prouver - $\overline{\delta}$ désignant l'extension classique de δ aux mots de X^* - que $\overline{\delta}(0, u)$ est la longueur du plus grand préfixe de M qui soit suffixe de u.

Que peut-on conclure concernant le langage reconnu par $\mathcal{A}$?

III-4-4. Proposer, en utilisant uniquement[1] la fonction f, un algorithme ou un programme simulant l'action de $\mathcal{A}$ sur le texte T.

III-4.5. Quelle est la complexité de l'algorithme de reconnaissance d'un motif utilisant la méthode de cette partie?

III-4-6. En utilisant les routines `Caml : print_string, print_int, print_newline,` écrire un programme qui recherche dans un texte T toutes les apparitions d'un motif M en imprimant les diverses positions obtenues.

1. En particulier, cet algorithme ne devra pas utiliser une fonction δ précalculée.