

**Compilation**  
**SMI – Semestre 5**  
**Série n° 1**

**Exercice 1 :**

On considère l'alphabet  $A = \{a, b, c\}$ . Soient deux mots  $u = ababc$  et  $v = caba$ .

1. Calculer  $u^0$ ,  $u^1$ ,  $u^2$ ,  $u^3$ ,  $uv^2u$ ,  $|u|_{ab}$ ,  $|(ab)^4|$  et  $|(ab)^4|_{aba}$ .
2. Donner les préfixes, les suffixes et les sous-mots du mot  $v$ .
3. Donner le miroir du mot  $uv$ .
4. Le mot  $uvvvu$  est-il un palindrome sur  $A$  ?

**Exercice 2 :**

Donner les expressions régulières qui décrivent :

1. Le langage de mots sur  $\{a, b\}$  qui commencent par  $a$  et se terminent par  $b$ .
2. Le langage formé des mots sur  $\{a, b, c\}$  qui débutent par  $a$ , contiennent exactement deux  $b$  et se terminent par  $cc$ .
3. Le langage de tous les mots sur  $\{a, b\}$  concaténés avec les mots sur  $\{c, d\}$ .
4. L'ensemble des mots sur  $\{0, 1\}$  composés d'un nombre arbitraire de  $1$ , suivis de  $01$ , suivis d'un nombre arbitraire de  $0$ .

**Exercice 3 :**

Simplifier les expressions régulières suivantes:

1.  $\varepsilon + ab + ab + abab(ab)^*$
2.  $aa(b^* + a) + a(ab^* + aa)$
3.  $a(a+b)^* + aa(a+b)^* + aaa(a+b)^*$

**Exercice 4 :**

Montrer les égalités suivantes :

1.  $b + ab^* + aa^*b + aa^*ab^* = a^*(b + ab^*)$
2.  $a^*(b + ab^*) = b + aa^*b^*$

### **Exercice 5 :**

Soit l'AFD  $M = (E, A, q_0=1, \delta, F)$  où  $E = \{1; 2; 3\}$ ,  $A = \{a; b\}$ ,  $F = \{3\}$  et où la fonction de transition  $\delta$  est donnée par :

$\delta$	$a$	$b$
1	1	2
2	3	2
3	3	1

1)

- 1) Tracer le diagramme de transition de A.
- 2) Donner l'exécution de A sur les mots abba, bbbabb, bababa,

### **Exercice 6 :**

1. Définir un type mot comme étant une chaîne de caractères en utilisant un pointeur.
2. Ecrire une fonction qui teste si un mot donné est bien défini sur l'alphabet binaire  $A = \{a, b\}$ .
3. Ecrire une fonction itérative qui calcule la longueur d'un mot.
4. Ecrire une fonction récursive qui calcule la longueur d'un mot.
5. Ecrire une fonction qui calcule la puissance nième d'un mot donné.
6. Ecrire une fonction qui calcule le miroir d'un mot donné.
7. Ecrire une fonction qui teste si un mot donné est un facteur d'un autre mot.

### **Exercice 7 :**

Un Automate Fini Déterministe (AFD) sera stocké dans un fichier texte selon le format suivant :

- sur la première ligne, on écrit le nombre des états.
- sur la deuxième ligne, on écrit la chaîne représentant l'alphabet.
- sur la troisième ligne, on écrit l'état initial.
- sur la quatrième ligne, on écrit le nombre des états finaux.
- sur la cinquième ligne, on écrit les états finaux séparés par des espaces.
- sur les lignes suivantes, on représente les transitions en écrivant sur chaque ligne, l'état de départ, le symbole lu et l'état d'arrivée, séparés par des espaces.

- 1- Proposer une structure de données **afd** qui représente votre automate.
- 2- Ecrire en C /C++ les fonctions suivantes :

a- `afd read(char * nameFile)` qui ouvre le fichier texte représentant un AFD et retourne la structure qui l'implémente en mémoire.

b- `void print(afd M)` qui prend un AFD M et affiche sa description à l'écran selon le format suivant :

**E = {0, 1, 2} A = {a, b, c}**  
**Transitions :**  
**t(0, a) = 1**  
 .....  
**I = 0**  
**F = {2}**

c- `int accept(afd M, char * w)` qui prend un AFD M et un mot w et qui teste si ce mot est reconnu par l'automate.

**4- Le programme principal qui :**

- a. Demande à l'utilisateur d'entrer **le nom du fichier contenant la description de l'automate qui décrit notre mini-langage.**
- b. Charge l'automate décrit dans ce fichier en mémoire.
- c. Affiche cet automate.
- d. Demande à l'utilisateur de saisir **le nom du fichier à analyser.**
- f. Affiche le résultat de la reconnaissance.