

TP N°3 Automates d'états finis (2 séances)

Le but de ce TP est de réaliser un programme de simulation d'automates finis **déterministes** (AEFD) en utilisant la représentation des automates présentée en cours. L'alphabet d'entrée de ces automates sera limité aux caractères suivants : $V_t = \{ 'a', 'b', 'c', 'd' \}$. Par convention, l'état initial sera toujours l'état 0.

Votre programme devra permettre :

1- de saisir un automate au clavier, selon le format suivant :

Nombre d'états : **4**
Nombre de symboles de V_t : **3**
Nombre d'états terminaux : **2**
États terminaux : **0 3**

Nombre de transitions : **7**
Liste des transitions (état caractère état) :

0 a 1
0 b 2
1 a 1
1 b 3
1 c 2
2 a 3
3 c 0

2- D'afficher le tableau (matrice) de transitions de l'automate saisi

3- De simuler le fonctionnement d'un automate : l'utilisateur entre une chaîne de caractères, le programme affiche la trace du fonctionnement de l'automate et la réponse "OUI" ou "NON" selon que le mot est reconnu ou pas par l'automate. On supposera que l'utilisateur respecte les consignes et n'utilise que des caractères présents dans l'alphabet.

Avec l'automate saisi précédemment, nous aurons par exemple :

chaîne à analyser : **aabca**

trace : 0 -- a -> 1 -- a -> 1 -- b -> 3 -- c -> 0 -- a -> 1 résultat :

NON

bac ==> OUI

aaab ==> OUI

4- Simuler la reconnaissance en utilisant d'autres automates.

TP N°4 Automates d'états finis (2 séances)

Le but de ce TP est de réaliser un programme qui transforme un automate d'états finis non déterministe en un automate d'états finis déterministe, en appliquant la méthode vue en cours.

- 1- Trouver la bonne structure de données pour représenter un automate d'états finis non déterministe.
- 2- Ecrire un programme qui permet :
 - a) De lire (saisir) un automate non déterministe.
 - b) De transformer cet automate en un automate déterministe équivalent.