Un automate déterministe fini :

1. Explication de l'automate :

Nous avons réalisé un automate déterministe fini qui suit ces caractéristiques :

M = (K, Σ, δ, s, F) où :

– K : ensemble fini (non vide) d'états

– Σ : alphabet (ensemble non vide de lettres)

– δ : fonction de transition : K × Σ à K δ(q, σ) = q' (q' : état de l'automate après avoir lu la lettre σ dans l'état q)

– s : état initial : s ∈ K

– F : ensemble des états finaux : F ⊂ K

Exemple M = (K, Σ, δ, s, F) de fonctionnement de notre automate prenons :

– K = {q0, q1}

– Σ = {a, b}

– s = q0

– F = {q0}

δ :	q	σ	δ(q, σ)
	Q0	a	Q0
	Q0	b	Q1
	Q1	a	Q1
	Q1	b	Q0

⇔

	a	b
Q0	Q0	Q1
Q1	Q1	Q0

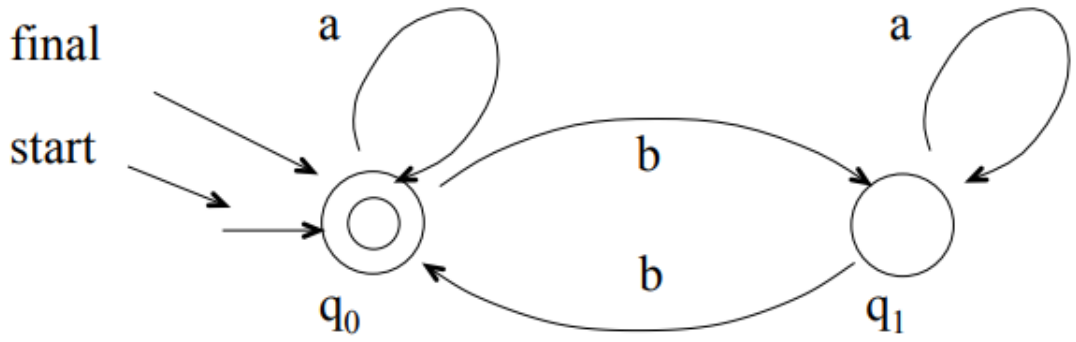

Figure 1 schéma de l'automate qu'on pourrait avoir

2. Programmation de l'automate fini :

Nous avons ensuite programmé cet automate en langage C++ dont voici le fonctionnement :

a. Construction de l'automate :

On commence d'abord par la construction de l'automate, à l'aide de deux structures (etat et trans) qui nous permettrons de définir les paramètres de notre automate. Puis grâce à la fonction automate() qui est de type TRANS, nous définissons le comportement de notre automate. Tout d'abord nous définissons les différentes transitions de l'automate avec leurs états initiaux et finaux qui nous permettrons de nous déplacer dans l'automate (ne pas oublier que les transitions et les états doivent être faits en fonction du mot que l'on veut écrire).

Nous vérifions ensuite le bon fonctionnement de celui-ci (bonnes transitions, pas de doublons, états valides), une fois cela effectuer on passe à la saisie du mot et de ces états initiaux et finaux et si cela semble cohérent on passe à la suite mais dans le cas contraire on doit ressaisir le mot.

b. Fonction Main :

Cette fonction nous permet de tester notre automate et de savoir si le mot est accepté ou non par l'automate fini déterministe. Pour exécuter le programme, commencez par :

- Les transitions sont des lettres (a..z & A..Z)
- Les états initiaux et finaux sont des entiers et doivent se suivre
- Pour taper un mot, il faut tout d'abord donner le nombre de lettres puis chaque lettre à part.
- Faire bien attention pour les messages que le programme donne (exemple Voulez-vous ajouter d'autres transitions ?) pour ne pas se tromper
- Le programme se termine alors pour nous dire si l'automate est accepté ou non par l'AFD que nous avons saisi.

c. Exemple de test de l'automate en console :

```
Bienvenue sur la creation de votre automate fini
Donnez une transition a votre automate
a
Donnez etat initial
0
Donnez etat final
1
Voulez vous ajouter d'autres etats pour cette transition ? (oui/non)
non
Voulez vous ajouter d'autres transitions a l'automate ? (oui/non)
oui
Donnez une transition a votre automate
b
Donnez etat initial
1
Donnez etat final
2
Voulez vous ajouter d'autres etats pour cette transition ? (oui/non)
non
Voulez vous ajouter d'autres transitions a l'automate ? (oui/non)
oui
Donnez une transition a votre automate
c
Donnez etat initial
2
Donnez etat final
3
Voulez vous ajouter d'autres etats pour cette transition ? (oui/non)
non
Voulez vous ajouter d'autres transitions a l'automate ? (oui/non)
oui
Donnez une transition a votre automate
d
Donnez etat initial
3
Donnez etat final
4
Voulez vous ajouter d'autres etats pour cette transition ? (oui/non)
non
Voulez vous ajouter d'autres transitions a l'automate ? (oui/non)
non
d       3       4
c       2       3
b       1       2
a       0       1
Donnez la taille de votre mot
3
Donner les differentes lettre(s) une a une en appuyant a chaque fois sur entrer 1
a
Donner les differentes lettre(s) une a une en appuyant a chaque fois sur entrer 2
b
Donner les differentes lettre(s) une a une en appuyant a chaque fois sur entrer 3
c
Votre mot est abc. Passons au test ce celui-ci
Donner etat de depart
0
Donnez le nombre des etats finaux
1
Donnez etat final 1
3
Mot Accepte
Voulez vous essayez un autre mot ? (oui/non)
```