

## TP : Simulation d'un automate

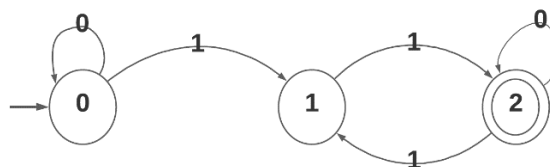
### Objective de TD :

- Implémentation d'un automate particulier
- Simulation d'un automate quelconque

### Délai de remis :

**00h00 le 1 avril 2021**

### Exercice 1 (**Obligatoire**) : Simulation d'un automate particulier



1. Quel est le langage reconnu par l'automate ci-dessus ?
2. Donner l'expression régulière de ce langage.

La simulation d'un automate consiste à parcourir ces états en consommant les alphabets d'un mot binaire donné (composé des 0s et 1s), pour vérifier si l'automate peut l'accepter ou non.

3. Ecrire un algorithme qui simule l'automate ci-dessus.
4. Traduire l'algorithme de simulation en langage C.

Exemples :

Entrée :

0111100

Sortie :

Etat 0 → Etat 0 → Etat 1 → Etat 2 → Etat 1 → Etat 2 → Etat 2 → Etat 2

Mot accepté

Entrée :

011100

Sortie :

Etat 0 → Etat 0 → Etat 1 → Etat 2 → Etat 1 → Erreur

Mot non accepté

### Indications :

1. L'algorithme/programme est destiné à cet automate en particulier.
2. Vous pouvez utiliser une boucle « **while** » pour parcourir l'alphabet du mot.
3. Vous pouvez utiliser une variable « **Etat** » pour la gestion de l'automate.
4. Utiliser les structures de contrôle conditionnels (**If** ou **switch case**) pour changer l'état en fonction de l'alphabet, si le changement est possible.
5. Dans le cas où le changement de l'état n'est pas possible, déclencher une erreur.
6. Vérifier à la fin si le dernier état après l'analyse est final pour accepter le mot.

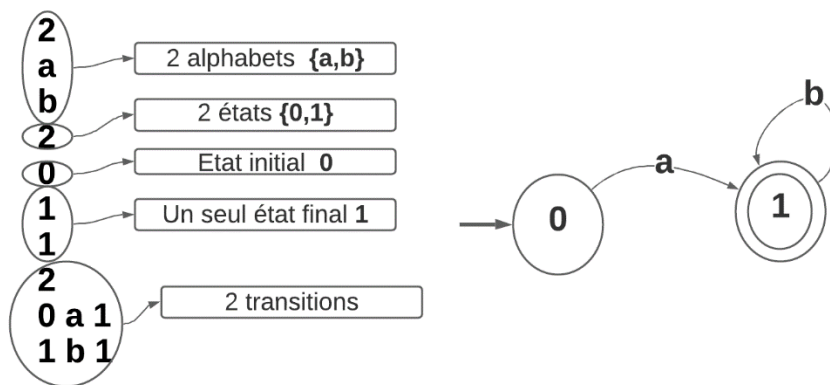
## Exercice 2 (Facultative) : Simulation d'un automate quelconque

Cet exercice est facultatif et l'étudiant peut l'ignorer. Cependant, les étudiants qui proposent une solution correcte pour cet exercice bénéficieront d'un bonus dans la note finale du TP.

L'objectif est d'écrire un programme C capable de simuler n'importe quel automate défini dans un fichier texte préparé par l'utilisateur. La structure de ce fichier d'entrée est la suivante :

<b>Nombre d'alphabet</b>
Alphabet 1
...
Alphabet N
<b>Nombre d'états</b>
<b>Etat initial</b>
<b>Nombre d'état finaux</b>
Etat final 1
...
Etat final M
<b>Nombre de transitions</b>
Transition 1
...
Transition K

Exemple :



La structure de l'automate est définie dans le fichier header **automate.h**. L'automate est défini comme une structure qui contient les champs suivants :

1. **char alphabets[100]** : l'alphabet de l'automate.
2. **int nb\_alphabets** : nombre d'alphabets.
3. **int nb\_etats** : nombre d'états.
4. **int etat\_initial** : état initial.
5. **int etats\_finaux[100]** : les états finaux de l'automate.
6. **int nb\_etats\_finaux** : nombre d'états finaux.
7. **int table\_transition[100][100]** : table deux dimensions des transitions de l'automate.

Pour faciliter l'écriture du programme final, un ensemble de fonctions d'aide sont mises dans le fichier header **helper\_funtions.h** et le fichier du code **helper\_funtions.c**. Les fonctions disponibles sont les suivantes :

1. **void print\_string(char str[],int size,int align)**: Fonction pour afficher une chaine de caractères dans un tableau.
2. **void display\_transition\_table(Automate automate)**: Fonction pour l'affichage de la table de transition.

3. **int index\_alphabet(char alphabet, Automate automate):** Fonction qui détermine l'indice d'un alphabet pour accéder à la table de transition de l'automate. Cette fonction retourne **-1** si l'alphabet n'existe pas.
4. **int read\_automate\_from\_file(char file\_name[100],Automate \*automate):** Fonction pour la lecture de l'automate à partir d'un fichier text qui respecte le format expliqué ci-dessus.
5. **int etat\_finale(int etat, Automate automate) :** Fonction pour vérifier si un état est un état final de l'automate ou non.

#### Travail demandé :

Ecrire la fonction **reconnaitre** présente dans le fichier **main.c** qui permet de vérifier si un mot donné par l'utilisateur est accepté par l'automate ou non.

**int reconnaitre(char mot[100],Automate automate)**

Cette fonction doit afficher les états de l'automate durant l'analyse du mot comme suit :

Reconnaissance du mot 0101 :

Etat : 0 (Initial)

Etat : 1

Etat : 0 (Final)

Etat : 1

Etat : 0 (Final)

Le mot (0101) est reconnu par l'automate

Modifier directement le fichier **main.c**

#### Indications :

1. Essayer de comprendre la structure du projet C avant de commencer l'implémentation.
2. Pour comprendre comment un programme C est organisé en plusieurs fichiers .h et .c, vous pouvez voir le lien suivant : <https://flaviocopes.com/c-header-files/>
3. Essayer de comprendre globalement les fonctions implémentées dans **helper\_funtions.c**.
4. Focaliser vos efforts sur l'implémentation de la fonction **reconnaitre**.
5. Considérer le cas où le mot contient des alphabets qui n'appartiennent pas à l'alphabet de l'automate.
6. Utiliser la table de transition de l'automate pour déterminer l'état suivant en fonction de l'état actuel et l'alphabet du mot.

#### Note finale :

- Envoyer les fichiers de code C (chaque exercice dans un fichier.c séparé) dans un seul email à : [bba.compilation.tp@gmail.com](mailto:bba.compilation.tp@gmail.com)
- Joindre des captures d'écran de l'exécution si possible.
- Objet de l'email : **Nom Prénom (Groupe)**