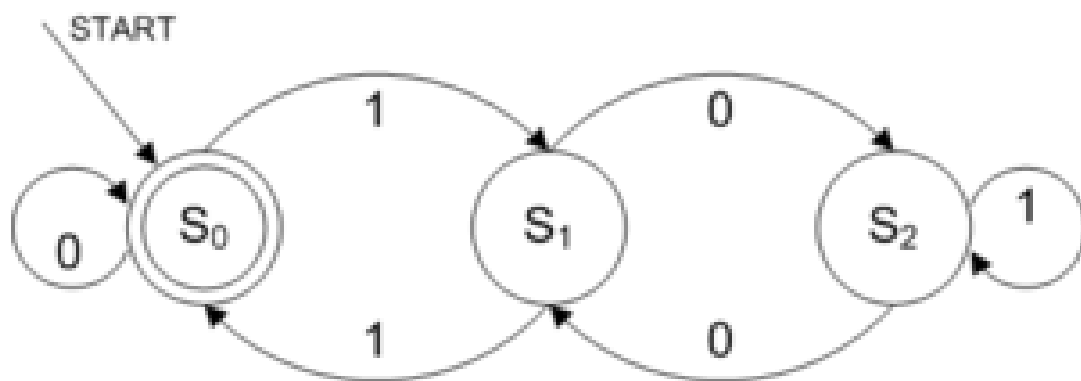


PROJET AF4 :

GESTIONNAIRE

D'AUTOMATES FINIS



Amadou SY
& Samy Meziane

SOMMAIRE

1) MODE D'EMPLOI

1.1) Installation

1.2) Démarrage

1.2) Navigation

2) LE PROJET

2.1) Les fonctionnalités demandées

2.2) Les fonctionnalités supplémentaires

2.3) Les difficultés rencontrées

2.4) Les erreurs, problèmes

2.5) Les fonctionnalités abandonnées

I) Mode d'emploi

A. Installation

L'installation se fait à partir du terminal, une fois dans le répertoire on utilise les commandes suivantes :

- « javac *.java » (Pour compiler les classes)
- « java Main » (pour exécuter les programmes)

B. Démarrage

Une fois le programme lancé l'utilisateur atteint la page suivante, où il est invité à effectuer une action, plusieurs fonctions lui sont proposées :

```
PS C:\Users\Amadou\Dropbox\PROJETS\AF4\29_04> javac *.java
PS C:\Users\Amadou\Dropbox\PROJETS\AF4\29_04> java Main

Bienvenu sur le gestionnaire d'automates

1. Comparer deux automates
2. Minimiser un automate
3. Generer un automate
4. Convertir une expression en automate
5. Afficher un automate
6. Quitter

Veuillez choisir une action:
```

C. La navigation

La navigation dans le programme se fait à l'aide d'entrée de l'utilisateur.

Lorsque l'utilisateur a le choix entre plusieurs options, il doit entrer le chiffre qui se trouve devant (dans la figure précédente par exemple pour comparer deux automates il faut entrer « 1 »).

L'utilisateur peut aussi se voir demander d'entrer un nom de fichier dans ce cas il doit écrire le chemin dans lequel se trouve le fichier.

```
Veuillez choisir une action: 1

##### COMPARAISON DE 2 AUTOMATES #####

Fichier 1 : A/A1
Fichier 2 : A/A2
```

II) LE PROJET

Dans l'énoncé du projet un certain nombre de **fonctionnalités** étaient demandé, le programme est censé :

- Construire un outil qui reconnaisse si deux automates ou expressions sont équivalents,
- Minimiser un automate à l'aide de l'algorithme de Moore,
- Convertir une expression rationnelle en automate avec l'algorithme de Glushkov ,
- Calculer les résiduels d'une expression puis construire l'automate des résiduels
- La lecture d'une expression rationnelle ou d'un automate dans un fichier,
- Générer un automate à l'aide d'un nombre d'états et d'un alphabet donnés,
- Un moyen de choisir les fonctions à appeler, j'ai choisis le menu textuel.

Toutes ces fonctionnalités ont été implémentées et fonctionnent parfaitement. En plus des **fonctions supplémentaires** permettent à l'utilisateur de :

- Sauvegarder l'automate construit par une des fonctions.
- De charger sans distinction un automate ou une expression rationnelle.

Cependant lors de la réalisation du projet, j'ai rencontré quelques **difficultés**, notamment :

- L'exploitation des codes utilisés en TP.
- La méthode de minimisation par Moore a posé problème.

Certains **problèmes** ont été **constatés** lors de la phase de test:

- La gestion des exceptions n'est pas prise en compte (notamment en cas de mauvaises entrées).
- La fonction de comparaison arrête le programme.

Certains **fonctionnalités** ont elle été **abandonnées** en cours de réalisation, car non pertinentes :

- Une interface graphique
- L'implantation des méthodes déjà codées en TP.