

SIMON BOLDUC
Mathématiques pour informaticiens II
PIF1006, gr.00

DEVOIR NO. 1

Travail présenté à
Adam Joly

Département de l'informatique
Université du Québec à Trois-Rivières
Le 20 octobre 2023

TABLE DES MATIERES

Rôles des membres de l'équipe.....	3
Problèmes et difficultés.....	3
Guide utilisateur.....	3
Charger un automate par fichier.....	3
Instructions acceptées.....	3
Afficher l'automate	4
Valider un « input » par l'automate	4
Quitter.....	4
Interfaces utilisateurs.....	5
Menu principal.....	5
Option 1 : charger un automate par fichier.....	5
Option 2 : afficher l'automate	6
Option 3 : valider un input.....	7
Tests fonctionnels	8
Un exemple de fichier d'automate conforme et son chargement initial	8
Un exemple de fichier d'automate non conforme (transition pour état non existant) et le succès du chargement en ignorant l'action	9
L'affichage de l'automate en console/UI (le faire pour 2 automates différents avec au moins 4 états chacun)	9
La soumission d'un input qui retourne une acceptation (3 cas de tests).....	11
La soumission d'un input qui retourne un rejet (3 cas de tests).....	12
La soumission d'un input composée d'au moins un caractère qui n'est pas un 0 ou un 1, et qui retourne un rejet sans plantage.....	13

Rôles des membres de l'équipe

Étant une équipe d'une personne, la responsabilité de l'analyse, du développement et des tests est attribuée à moi-même. J'ai aussi fait le choix de technologie selon mes connaissances et mon expérience afin de simplifier la tâche de développement (meilleure expérience développeur) et pour faciliter les tests fonctionnels du logiciel. Finalement, le rapport a été complètement détaillé par moi-même permettant d'expliquer et de démontrer le résultat du travail pratique sur le sujet des automates.

Problèmes et difficultés

Évidemment, lorsqu'un développeur reçoit une tâche, le premier problème est de comprendre la demande afin de savoir exactement ce qu'il est demandé. La lecture de l'énoncé et l'analyse de la solution de base en C# a permis d'avoir une meilleure idée du problème. Maintenant que la tâche est claire, le deuxième problème est de définir par quel moyen nous allons y arriver. Le choix de technologie et l'architecture de base est un problème majeur dans le contexte du développement logiciel. Une mauvaise architecture ou choix technologique peut apporter un lot de conséquence à court, moyen et long terme. Une fois que l'on avait une base solide, il est facilement possible d'implémenter des fonctionnalités supplémentaires et de maintenir le code afin de l'améliorer.

Guide utilisateur

Charger un automate par fichier

Afin de charger une configuration d'automate, il est nécessaire d'utiliser l'option numéro 1. Cette option vous demandera par la suite d'entrer le chemin absolu du fichier texte contenant la configuration souhaitée. Il est important de noter que le fichier doit être un fichier de type .txt. Le contenu du fichier sera validé lors de son chargement et dans le cas d'erreur de configuration, un message d'erreur vous sera affiché.

Instructions acceptées

STATE <nom de l'état> <est final (1|0)>

TRANSITION <nom de l'état initial> <valeur de l'input transitoire > <nom de l'état final>

Toute autre instruction ou valeur d'argument sera refusé par le programme.

Afficher l'automate

Il est possible d'afficher la configuration qui a été chargée par l'automate. Pour ce faire, utiliser l'option numéro 2 à partir du menu principal. Il est important de s'assurer qu'une configuration a été chargée avant d'utiliser cette option. Dans le cas contraire, une erreur vous sera retournée. Dans le cas où une configuration est chargée, la liste des états chargés et la liste des transitions avec leur valeur vous sera retournée.

Valider un « input » par l'automate

Pour pouvoir valider un input selon les états et les transitions chargés dans l'automate, choisir l'option numéro 3 du menu principal. Cette option vous demandera d'entrer l'input final puis le validera par l'automate. La chaîne doit être composée de 0 et de 1 seulement pour être valide. Par la suite, si l'automate assume que l'input est valide, un succès sera affiché, contrairement, une erreur sera affichée.

Quitter

Lorsque vous êtes sur le menu principal, simplement utiliser l'option numéro 4 afin de quitter le programme et de le fermer.

Interfaces utilisateurs

Menu principal

```
Text Terminal
1. Charger un automate par fichier
2. Afficher l'automate
3. Valider un input par l'automate
4. Quitter
Option : |
```

Option 1 : charger un automate par fichier

Cas valide

```
1. Charger un automate par fichier
2. Afficher l'automate
3. Valider un input par l'automate
4. Quitter
Option : 1
Veuillez entrer le chemin absolu du fichier: |
```

```
1. Charger un automate par fichier
2. Afficher l'automate
3. Valider un input par l'automate
4. Quitter
Option : 1
Veuillez entrer le chemin absolu du fichier: C:
\Users\Sim\Desktop\tp1\automate-fonctionnel.txt
1. Charger un automate par fichier
```

Cas d'échec : chemin invalide

```
1. Charger un automate par fichier
2. Afficher l'automate
3. Valider un input par l'automate
4. Quitter
Option : 1
Veuillez entrer le chemin absolu du fichier: dwada

[ERREUR] Le chemin précisé n'est pas un fichier valide
```

Option 2 : afficher l'automate

Cas succès

```
1. Charger un automate par fichier
2. Afficher l'automate
3. Valider un input par l'automate
4. Quitter
Option : 2
```

[SUCCÈS] Automate :

États possibles :

s0(F), s1, s2(F), s3, s4(F)

Transitions possibles :

s0 -> ε, s1(0)

s1 -> s0(0), s2(1)

s2 -> s2(1), s3(0)

s3 -> s1(1), s4(0)

s4 -> s1(0)

Cas d'échec : l'automate n'a pas été configuré

```
1. Charger un automate par fichier
2. Afficher l'automate
3. Valider un input par l'automate
4. Quitter
Option : 2
```

[ERREUR] L'automate n'a pas été définie

Option 3 : valider un input

Cas d'échec : l'input comporte des caractères interdits

```
1. Charger un automate par fichier
2. Afficher l'automate
3. Valider un input par l'automate
4. Quitter
Option : 3
Veuillez entrer un input à valider par l'automate
La chaine doit être composée de 0 et de 1 seulement: dwada
Invalid value.
Expected format: ^$|(0|1)+$
```

Cas d'échec : input invalide

```
1. Charger un automate par fichier
2. Afficher l'automate
3. Valider un input par l'automate
4. Quitter
Option : 3
Veuillez entrer un input à valider par l'automate
La chaine doit être composée de 0 et de 1 seulement: 010101
[ERREUR] La chaine {010101} est invalide.
```

Cas d'échec : l'automate n'a pas été configuré

```
1. Charger un automate par fichier
2. Afficher l'automate
3. Valider un input par l'automate
4. Quitter
Option : 3
Veuillez entrer un input à valider par l'automate
La chaine doit être composée de 0 et de 1 seulement: 010101
[ERREUR] L'automate n'est pas complète. Veuillez configurer
l'automate en premier lieu
```

Cas succès

```
1. Charger un automate par fichier
2. Afficher l'automate
3. Valider un input par l'automate
4. Quitter
Option : 3
Veuillez entrer un input à valider par l'automate
La chaine doit être composée de 0 et de 1 seulement: 01

[SUCCÈS] La chaine {01} est bien valide.
```

Tests fonctionnels

Un exemple de fichier d'automate conforme et son chargement initial

```
state s0 1
state s1 0
state s2 1
state s3 0
state s4 1
transition s0 0 s1
transition s1 0 s0
transition s1 1 s2
transition s2 1 s2
transition s2 0 s3
transition s3 1 s1
transition s3 0 s4
transition s4 0 s1
```

```
1. Charger un automate par fichier
2. Afficher l'automate
3. Valider un input par l'automate
4. Quitter
Option : 1
Veuillez entrer le chemin absolu du fichier: |
```

```
1. Charger un automate par fichier
2. Afficher l'automate
3. Valider un input par l'automate
4. Quitter
Option : 1
Veuillez entrer le chemin absolu du fichier: C:
\Users\Sim\Desktop\tp1\automate-fonctionnel.txt
```


Un exemple de fichier d'automate non conforme (transition pour état non existant) et le succès du chargement en ignorant l'action

```
state s0 1
state s1 0
state s2 1
transition s9999 0 s1
transition s1 0 s0
transition s1 1 s2
```

```
4. Quitter
Option : 1
Veuillez entrer le chemin absolu du fichier: C:
\Users\Sim\Desktop\tp1\automate-err-state-unknown.txt

[ERREUR] L'état initiale {s9999} n'a pas été trouvée

1. Charger un automate par fichier
2. Afficher l'automate
3. Valider un input par l'automate
4. Quitter
Option : 2

[SUCCÈS] Automate :

États possibles :
s0(F), s1, s2(F)
Transitions possibles :
s0 -> ε,
s1 -> s0(0), s2(1)
s2 ->
```

L'affichage de l'automate en console/UI (le faire pour 2 automates différents avec au moins 4 états chacun)

```
state s0 1
state s1 0
state s2 1
state s3 0
state s4 1
transition s0 0 s1
transition s1 0 s0
transition s1 1 s2
transition s2 1 s2
transition s2 0 s3
transition s3 1 s1
transition s3 0 s4
transition s4 0 s1
```

```
3. Valider un input par l'automate
4. Quitter
Option : 1
Veuillez entrer le chemin absolu du fichier: C:
\Users\Sim\Desktop\tp1\automate-fonctionnel.txt
1. Charger un automate par fichier
2. Afficher l'automate
3. Valider un input par l'automate
4. Quitter
Option : 2

[SUCCÈS] Automate :

États possibles :
s0(F), s1, s2(F), s3, s4(F)
Transitions possibles :
s0 -> ε, s1(0)
s1 -> s0(0), s2(1)
s2 -> s2(1), s3(0)
s3 -> s1(1), s4(0)
s4 -> s1(0)
```

```
state s0 1
state s1 0
state s2 0
state s3 1
state s4 0
state s5 1
transition s0 1 s1
transition s1 1 s0
transition s1 0 s2
transition s2 1 s2
transition s2 1 s3
transition s3 0 s1
transition s3 0 s4
transition s4 0 s1
transition s4 1 s5
```

```

4. Quitter
Option : 1
Veuillez entrer le chemin absolu du fichier: C:
\Users\Sim\Desktop\tp1\automate-fonctionnel2.txt
1. Charger un automate par fichier
2. Afficher l'automate
3. Valider un input par l'automate
4. Quitter
Option : 2

[SUCCÈS] Automate :

États possibles :
s0(F), s1, s2, s3(F), s4, s5(F)
Transitions possibles :
s0 -> ε, s1(1)
s1 -> s0(1), s2(0)
s2 -> s2(1), s3(1)
s3 -> s1(0), s4(0)
s4 -> s1(0), s5(1)
s5 ->

```

La soumission d'un input qui retourne une acceptation (3 cas de tests)

```

state s0 1
state s1 0
state s2 1
state s3 0
state s4 1
transition s0 0 s1
transition s1 0 s0
transition s1 1 s2
transition s2 1 s2
transition s2 0 s3
transition s3 1 s1
transition s3 0 s4
transition s4 0 s1

```

```

1. Charger un automate par fichier
2. Afficher l'automate
3. Valider un input par l'automate
4. Quitter
Option : 3
Veuillez entrer un input à valider par l'automate
La chaîne doit être composée de 0 et de 1 seulement: 01

[SUCCÈS] La chaîne {01} est bien valide.

```

```
1. Charger un automate par fichier
2. Afficher l'automate
3. Valider un input par l'automate
4. Quitter
Option : 3
Veuillez entrer un input à valider par l'automate
La chaîne doit être composée de 0 et de 1 seulement: 011

[SUCCÈS] La chaîne {011} est bien valide.
```

```
1. Charger un automate par fichier
2. Afficher l'automate
3. Valider un input par l'automate
4. Quitter
Option : 3
Veuillez entrer un input à valider par l'automate
La chaîne doit être composée de 0 et de 1 seulement: 01100

[SUCCÈS] La chaîne {01100} est bien valide.
```

La soumission d'un input qui retourne un rejet (3 cas de tests)

```
state s0 1
state s1 0
state s2 1
state s3 0
state s4 1
transition s0 0 s1
transition s1 0 s0
transition s1 1 s2
transition s2 1 s2
transition s2 0 s3
transition s3 1 s1
transition s3 0 s4
transition s4 0 s1
```

```
1. Charger un automate par fichier
2. Afficher l'automate
3. Valider un input par l'automate
4. Quitter
Option : 3
Veuillez entrer un input à valider par l'automate
La chaîne doit être composée de 0 et de 1 seulement: 1

[ERREUR] La chaîne {1} est invalide.
```

```
\Users\Sim\Desktop\tp1\automate-fonctionnel.txt
1. Charger un automate par fichier
2. Afficher l'automate
3. Valider un input par l'automate
4. Quitter
Option : 3
Veuillez entrer un input à valider par l'automate
La chaine doit être composée de 0 et de 1 seulement: 11111
[ERREUR] La chaine {11111} est invalide.
```

```
1. Charger un automate par fichier
2. Afficher l'automate
3. Valider un input par l'automate
4. Quitter
Option : 3
Veuillez entrer un input à valider par l'automate
La chaine doit être composée de 0 et de 1 seulement: 10101001
[ERREUR] La chaine {10101001} est invalide.
```

La soumission d'un input composée d'au moins un caractère qui n'est pas un 0 ou un 1, et qui retourne un rejet sans plantage

```
1. Charger un automate par fichier
2. Afficher l'automate
3. Valider un input par l'automate
4. Quitter
Option : 3
Veuillez entrer un input à valider par l'automate
La chaine doit être composée de 0 et de 1 seulement: 10101e101
Invalid value.
Expected format: ^$|(0|1)+$

Veuillez entrer un input à valider par l'automate
La chaine doit être composée de 0 et de 1 seulement:
```