EFREI 2021/2022

**Automates Finis et Expressions Rationnelles**

L3APP RI

Enseignant : H.Barbot

**Travail pratique**

Rappel : note finale = 0,75 \* note DE + 0,25 \* note TP

Historique:

| 02/11 08:40 | Version “initiale”. |
| --- | --- |
| 02/11 (séance TP) | Ajout fichiers “txt” en résultat à rendre.  Modification de la liste des automates de test.  Indication sur les suites de symboles à utiliser pour la reconnaissance de mots. |
| 03/11 | Mise à jour des équipes :  C.Schiavo déplacé en équipe 8 ;  E.Moulin a priori démissionnaire ;  Z.Bencheikh n’a pas réagi et est sans équipe. |

# **Travail à effectuer**

Votre programme doit mettre en oeuvre les fonctions suivantes :

1. lecture d’un automate décrit dans un fichier “.txt” et représentation en mémoire
2. affichage de l’automate sous forme d’une table des transitions, en indiquant les états initiaux et terminaux
3. vérification des propriétés “déterministe” ou non, “complet” ou non de l’automate
4. reconnaissance de mots
5. standardisation et ajout / élimination du mot vide
6. passage au langage complémentaire

Les automates à prendre en compte auront les caractéristiques suivantes :

* les états sont représentés par des numéros, à partir de ‘0’, sans rupture de séquence (ex. ‘0’ à ‘4’ pour 5 états) ;
* les lettres sont les symboles de l’alphabet sans rupture de séquence (ex. ‘a’ à ‘c’ pour 3 symboles).

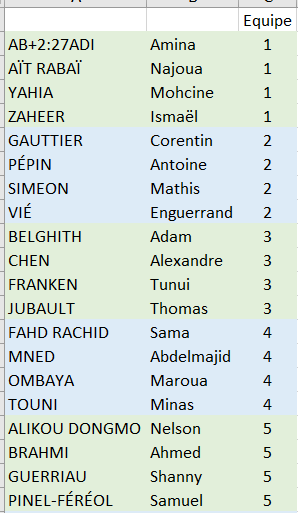
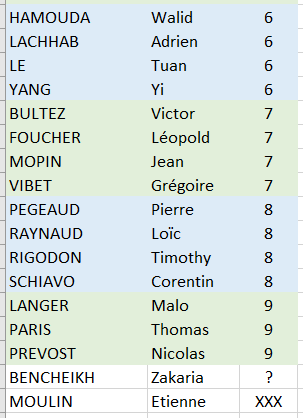
La reconnaissance de mots doit être mise en œuvre sur un AF quelconque (déterministe ou non, complet ou non). Vous pouvez mettre en œuvre plusieurs fonctions (pour automate déterministe et complet, déterministe mais non complet, non déterministe).

Les traitements de standardisation et de passage au langage complémentaire sont indépendants l’un de l’autre (le passage au complémentaire se fait sur l’automate lu à l’étape 1).

Le passage au langage complémentaire doit pouvoir fonctionner sur un automate non complet (votre programme doit bien entendu le compléter).

# Conditions de travail

## **Equipes**

** 

## **Langage de programmation**

Au choix : C C++ Python Java

## Environnements de développement

Vous pouvez avoir la liste des IDE installés sur l’ordinateur de votre enseignant pour tester vos programmes en suivant [ce lien](https://docs.google.com/document/d/1mdXW5dWujufWTt1ZRqRSiJoYGPNEifwR8WZpK4bSoW4/edit?usp=sharing).

# Résultat

## Rendu du travail

Programme + traces d’exécution sur les automates de test qui vous seront fournis.

Date limite de rendu du travail : **16 janvier 2022**

A fournir :

programme (code source uniquement)[[1]](#footnote-0)

graphes des états des automates de test (fichier pdf)

fichiers txt représentant les automates de test (point 1 du travail à effectuer)

traces d’exécution sur les automates de test

le tout dans un fichier compressé au format .zip (pas de fichier .rar !) nommé par votre numéro d’équipe (par exemple “8.zip”)[[2]](#footnote-1)

A envoyer par email :

À : [herve.barbot@intervenants.efrei.net](mailto:herve.barbot@intervenants.efrei.net)

Sujet : L3.APP.RI MPI TP[[3]](#footnote-2)

Message :

...*a priori rien*...

PJ : ...*votre fichier .zip*...

## Automates de test

Vous devez créer vous-même vos automates de tests.

Caractéristiques communes pour tous les automates choisis :

* alphabet = { a , b , c }
* nombre d’états : au moins 5

Caractéristiques particulières :

Vous devez créer 9 automates de test en respectant les propriétés ci-dessous.

Respectez les numéros des graphes en fonction du tableau !

|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9\* |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| déterministe | non | non | non | non | oui | oui | oui | oui | oui |
| complet | non | non | non | non | oui | non | oui | non | oui |
| standard | non | non | oui | oui | non | non | oui | oui |  |
| mot vide accepté | non | oui | oui | non | oui | non | oui | oui | non |

\* Automate 9 : déterministe et complet,standard ou non, mais aucun état terminal (donc mot vide non reconnu).

Si vous le jugez intéressant, plusieurs équipes peuvent coopérer pour construire les mêmes automates de test et comparer les résultats.

## Mots à utiliser pour la fonction de reconnaissance

Vous devez, pour chaque automate de test, fournir un nombre “suffisant” de mots de test permettant de juger de la fiabilité de votre fonction de reconnaissance.

Pour chaque automate testé, et selon ses propriétés, vous devez fournir des traces de reconnaissance de mots pour au moins les suites de symboles ayant les caractéristiques suivantes :

pour tous les automates de test :

un mot reconnu (sauf automate 9 !),

un mot non reconnu,

le mot vide,

un mot (non reconnu) pour lequel en fin de mot, l’état courant n’est pas un état terminal,

un mot qui contient un symbole qui n’appartient pas à l’alphabet ;

pour un automate non complet :

un mot menant à une situation pour laquelle il n’y a pas de transition applicable pour l’état courant et le symbole courant.

La liste de ces mots doit être donnée, pour chaque automate, dans le fichier contenant les graphes des états des automates de test.

*\_\_\_ fin du document \_\_\_*

1. Fournir un fichier archive du répertoire associé au projet créé dans votre IDE, avec tous les fichiers créés par l’IDE lui-même, est donc interdit. [↑](#footnote-ref-0)
2. Seul les fichiers .zip seront acceptés. Les autres seront ignorés. [↑](#footnote-ref-1)
3. Je mets en place des filtres dans mon client de messagerie. Tout manquement à cette consigne entraînera un mauvais traitement de votre rendu de travail. [↑](#footnote-ref-2)