Projet C, ING1

Un éditeur textuel ou graphique pour les automates d'états finis (AEF)

Zaouche Djaouida

Notion d'automates d'états finis

Un **automate fini** ou **automate avec un nombre fini d'états** (en anglais *finite-state automaton* ou *finite state machine* ou *FSM*) est un modèle mathématique de calcul, utilisé dans de nombreuses circonstances, allant de la conception de programmes informatiques et de circuits en logique séquentielle aux applications dans des protocoles de communication, en passant par le contrôle des processus, la linguistique et même la biologie [1].

Exemple :

Un portillon.

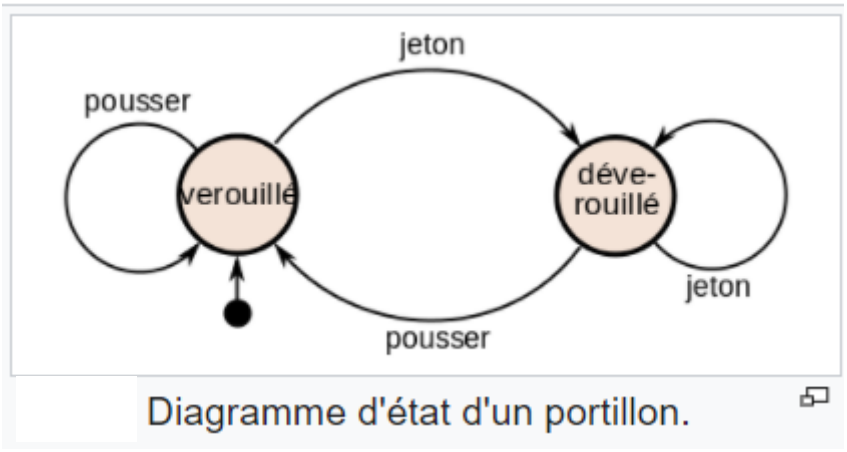


Diagramme d'état d'un portillon.

Travail demandé

Il est vous est demandé de réaliser un éditeur dédié aux automates finis. Dans ce qui suit, X est un alphabet (un ensemble de symbole). L'éditeur permet de :

1. Manipuler des mots d'un alphabet :
   - vérifier si un mot appartient à X*.

     appartient(string,alphabet)
     1. appartient("aaab",{a,b,c})=true
     2. appartient("aaabd",{a,b,c})=false

   - calculer la puissance d'un mot.

     puis(string,int)
     1. puis("ab",3)=ababab
     2. puis("ab",0)= ""

   - vérifier si un mot est vide (ε).

     vide(string)
     1. vide(*"aaab")=false*
     2. vide(*"")=true*

   - concaténer deux mots.

     concat(string,strig)
     1. concat(*"aaab","aa")=aaabaa*

   - calculer le miroir d'un mot.

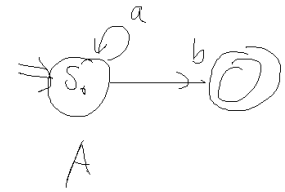
     miroir(string)
     1. miroir(*"aaab")=baaa*

   - vérifier si une chaine de caractères représente un mot fini.

     estFini(string)
     1. estFini(*"aaab")=true*
     2. estFini(*"a*aa*bd*a")=false*

2. Manipuler un AEF :
   - saisir un AEF,
   - importer un AEF à partir d'un fichier,
   - modifier un AEF,
   - sauvegarder un AEF dans un fichier.

3. Vérifier si un mot est reconnu par un AEF.

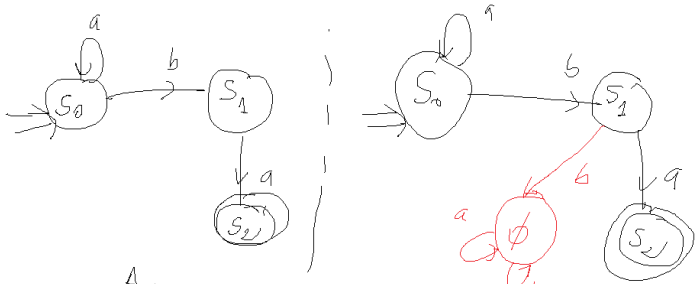


A

$ab \in L(A)$

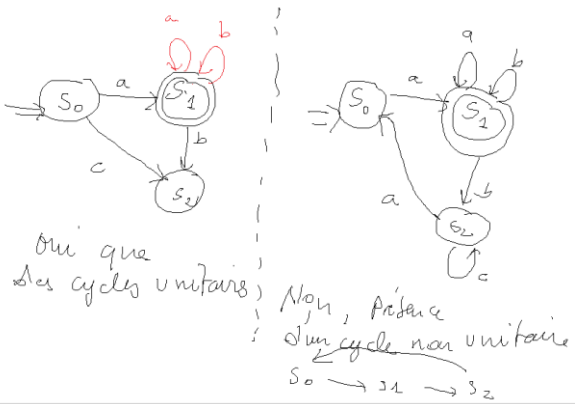
$ba \notin L(A)$

4. Vérifier si un automate est complet.

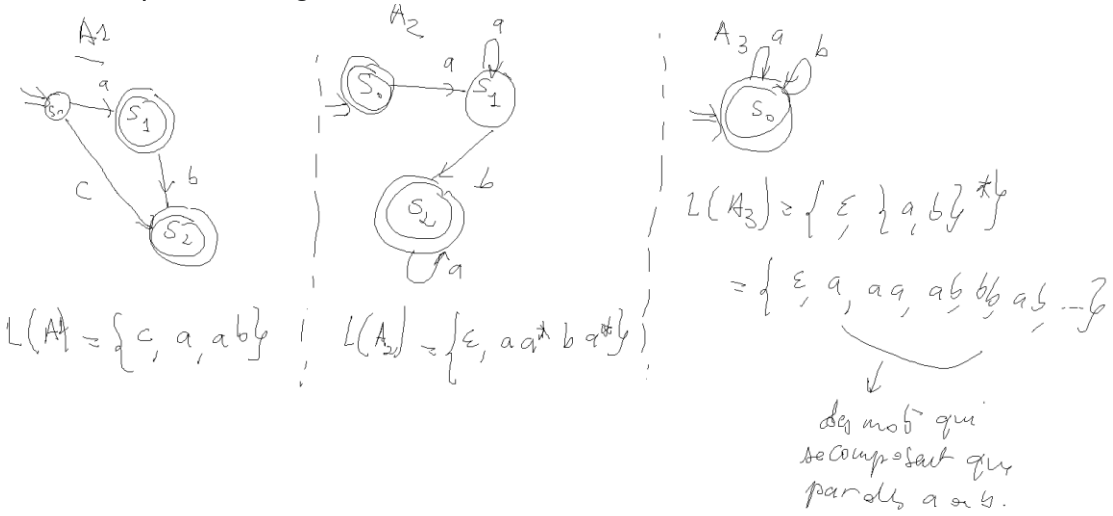


$L(A_1) = L(A_2)$

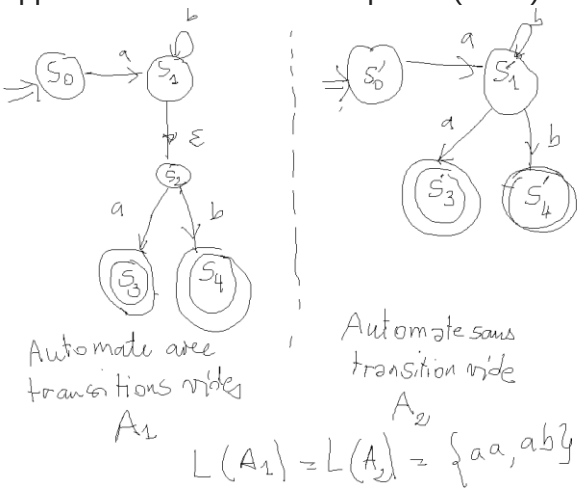
5. Rendre un automate complet.
6. Vérifier si un automate est déterministe.
7. Rendre un AEF déterministe.
8. Vérifier que tous les cycles d'un AEF sont unitaires. Répondre par vrai pour tout AEF sans cycle. Un cycle est un chemin fermé (une boucle).



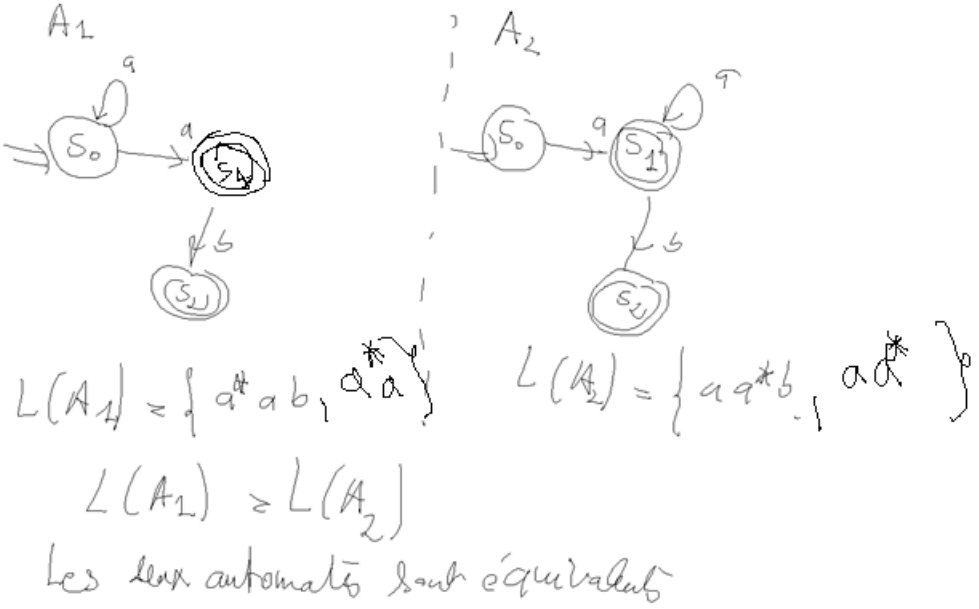
9. Extraire le langage reconnu par un AEF. Vous supposez que seuls les cycles unitaires sont pris en charge.



10. Suppression des transitions epsilon (vides).



11. Réaliser des opérations suivantes sur les AEFs :
    - complément d'un AEF,
    - miroir d'un AEF,
    - produit de deux AEFs,
    - concaténation de deux AEFs.
12. Vérifier si deux automates sont équivalents Vous supposez que seuls les cycles unitaires sont pris en charge.

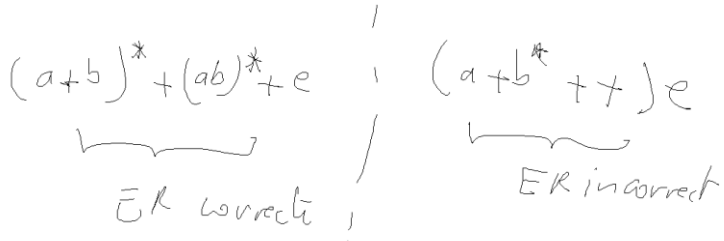


13. Rendre minimal un EDF déterministe (voir https://fr.wikipedia.org/wiki/Minimisation_d%27un_automate_fini_d%C3%A9terministe).

C. Expressions régulières (Ers)

1. Manipuler une ER :
   - saisir une ER,
   - importer une ER à partir d'un fichier,
   - modifier une ER,
   - sauvegarder une ER dans un fichier

2. Vérifier si une ER est correcte



3. Vérifier si un mot est reconnu par une ER.

ER = aba* + ba

(1) Le mot abaaa est reconnu par l'expression ER. Il est compatible avec le motif "aba*" qui représente tous les mots qui commencent par ab suivi par une série de a, éventuellement vide.

(2) C'est l'union donc le mot ba est reconnu par ER.

(3) Le mot abab n'est pas reconnu par ER.

4. Traduire une ER en un AEF équivalent.

ER = $ab^* + ab + \varepsilon$

A équivalut à ER