

**RAPPORT DU DEUXIEME TP\_THP**

**Thème**

**Intersection de deux automates**

**Réalisé par : Proposé par** :

BEY AHMED KHERNACHE Mohammed Mme BENATCHBA Karima

1CS/Groupe : 02

**Promotion** 2014/2015

Table des matières

[Introduction : 3](#_Toc407351037)

[Comment utiliser : 3](#_Toc407351038)

[Contrôles effectués par ce logiciel : 3](#_Toc407351039)

[Quelques captures d’écran : 4](#_Toc407351040)

[Conclusion : 6](#_Toc407351041)

# Introduction :

Aujourd’hui, les automates d’états finis sont parmi les outils les plus performants pour résoudre des problèmes du monde réel.

Ce TP consiste à construire un automate qui reconnaît l’intersection de deux langages engendrés par deux automates donnés en entrée.

Analyse :

* L’hypothèse de ce TP est que les automates en entrée sont simples et déterministes.

# Comment utiliser :

* Ce logiciel possède une interface, simple à utiliser, divisée en quatre parties qui comporte les fonctionnalités suivantes :

1. Remplir les champs de l’automate.
2. Contrôler les éventuelles erreurs qu’on peut commettre lors de la saisie.
3. Afficher l’automate créé graphiquement avec :

* La couleur jaune : état initial.
* La couleur verte : états finaux.

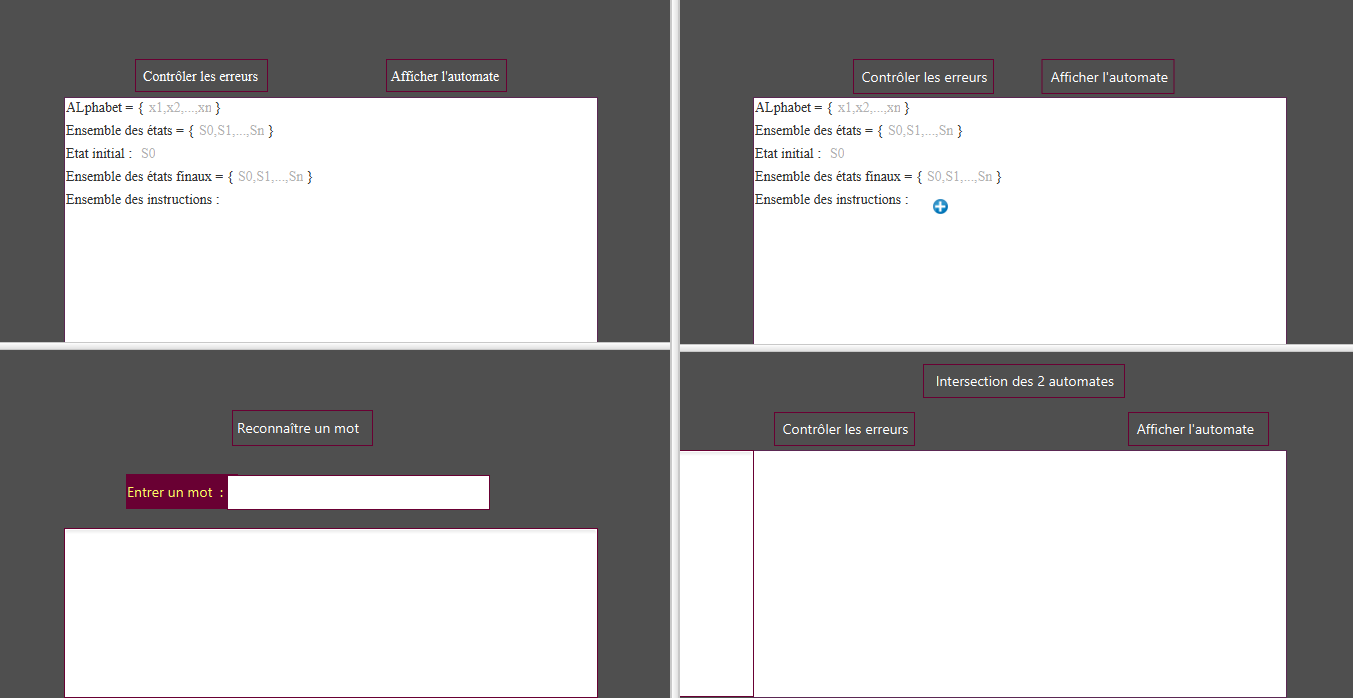
1. Reconnaître un mot de l’automate résultant de l’intersection des deux automates en entrée en écrivant le mot dans le champ de saisie et puis cliquant sur l’icône «Reconnaître un mot», ou taper la touche ENTRER.

* Pour ajouter une instruction ou la supprimer, il faut rapprocher la souris de l’instruction précédente et puis une petite icône apparaît avec laquelle on effectue l’opération souhaitée.

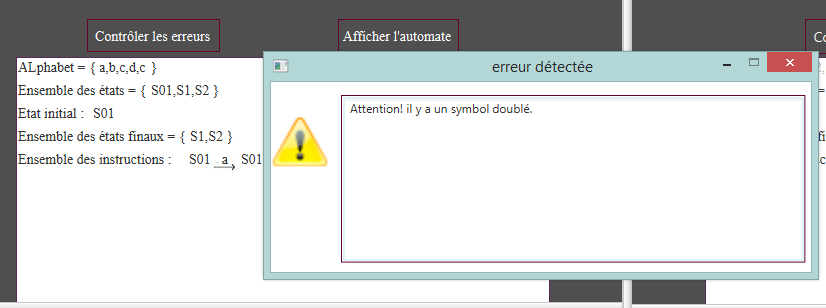
# Contrôles effectués par ce logiciel :

* Affichage graphique de l’automate construit par l’utilisateur.
* Affichage de l’automate formellement.
* Contrôle des états et des symboles s’ils sont utilisés sans les définir.
* Contrôler si l’utilisateur veut reconnaître un mot alors qu’il n’a pas introduit le mot.
* Afficher les différentes transitions pour atteindre un mot s’il est reconnu par l’automate.
* Puisqu’on travaille avec un automate simple déterministe, ce logiciel permet de :
* Détecter une erreur si l’utilisateur a introduit plus d’un état initial.
* Détecter une erreur si l’utilisateur a défini un symbole de longueur > 1.

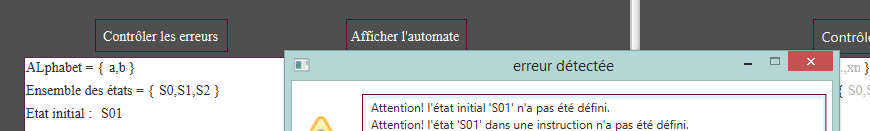
# Quelques captures d’écran :



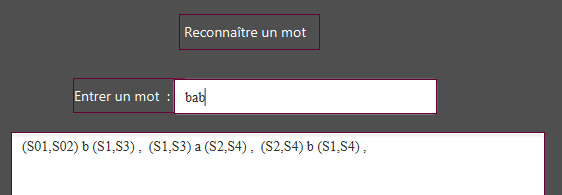
«Interface générale du TP»

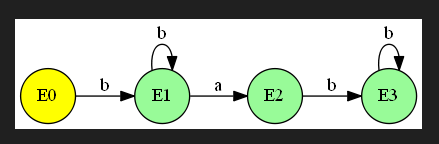


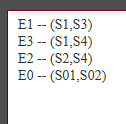
«Contrôle d’un symbole déclaré plus d’un fois»



«Utilisation d’un état non défini dans l’ensemble des états»



«Reconnaître un mot dans l’automate Intersection»



«Renommer les états pour alléger la lecture»

# Conclusion :

D’après ce TP, on a constaté que les opérations sur les automates (intersection, union, complément, miroir,…) ont permis à la résolution de plusieurs problèmes donc un certain nombre d’opérations de manipulation se font aisément sur les automates.