TD 4 : Automates et Algorithmes de Transformation

Exercice :

On considère l'expression régulière suivante sur l'alphabet $\{a, b\}$:

$$(a|b)^*a(a|b)$$

1. Donner une description en langage naturel de l'ensemble des mots acceptés par cette expression régulière.
2. Appliquer l'algorithme de McNaughton-Yamada-Thompson pour construire l'automate fini non déterministe (AFN) avec ε-transitions correspondant à l'expression régulière donnée.
3. Fournir la table de transition de l'AFN obtenu.
4. Convertir l'AFN obtenu en un automate fini déterministe (AFD) en utilisant l'algorithme de déterminisation (méthode des sous-ensembles).
5. Construire la table de transition de l'AFD.
6. Minimiser le nombre d'états de l'AFD en appliquant l'algorithme de partitionnement (algorithme de Moore ou de Hopcroft).
7. Fournir la table de transition de l'AFD minimal obtenu.
8. Vérifier que l'AFD minimal reconnaît bien le même langage que l'expression régulière initiale en testant des mots comme : "a", "b", "aa", "ab", "ba", "bb", "aaa", "bab", "aba", etc.

Rappels sur les algorithmes :

Algorithme de Thompson (McNaughton-Yamada-Thompson) :

Permet de construire un AFN à partir d'une expression régulière en utilisant ε-transitions pour représenter l'union, la concaténation et l'étoile de Kleene.

Algorithme des sous-ensembles (Déterminisation d'un AFN en AFD) :

Transforme un AFN en AFD en regroupant des ensembles d'états atteignables dans un nouvel état unique de l'AFD.

Algorithme de minimisation (Moore ou Hopcroft) :

Réduit le nombre d'états d'un AFD en regroupant les états équivalents, c'est-à-dire ceux ayant les mêmes comportements pour toutes les transitions.

Objectif : Après avoir appliqué ces transformations, on obtient un automate minimal qui reconnaît le même langage avec le plus petit nombre d'états possible.

Bon travail.