

Exercice 1 : Concatenation et étoile

Construire un automate pour chacun des langages suivants :

- le langage \mathcal{L}_1 formé des mots terminant soit par le suffixe bb , soit par le suffixe aa ;
- le langage \mathcal{L}_2 formé des mots ne contenant pas les facteurs bbb et aaa ;
- le langage $\mathcal{L}_3 = \mathcal{L}_1 \cdot \mathcal{L}_2$
- le langage $\mathcal{L}_4 = \mathcal{L}_2 \cdot \mathcal{L}_1$
- le langage $\mathcal{L}_5 = \mathcal{L}_1^*$
- le langage $\mathcal{L}_6 = \mathcal{L}_2^*$

Exercice 2 : Applications du cours

Trouver un automate reconnaissant les langages décrits par les expressions rationnelles suivantes :

- $E_1 = (a + bc)^*d(e + fg)$
- $E_2 = ((ab + c)(d + e))^*$
- $E_3 = (a + ab)^*b(a + ba)$
- $E_4 = ((ab + a)(a + b))^*$

Exercice 3 : Miroir

Le langage miroir d'un langage \mathcal{L} est le langage $\tilde{\mathcal{L}} = \{\tilde{u}, u \in \mathcal{L}\}$, où $\tilde{u} = x_n \dots x_1$ si $u = x_1 \dots x_n$.

Décrire un procédé permettant de construire l'automate reconnaissant $\tilde{\mathcal{L}}$ connaissant celui de \mathcal{L} .

Calculer ainsi le langage miroir du langage \mathcal{L} formé des mots commençant par baa

Exercice 4 : Construction d'automates

1. Utiliser les constructions vues en cours pour les opérateurs '+', '.', et '*' pour trouver des automates reconnaissant les langages décrits par les expressions rationnelles suivantes.
 - $E_1 = (a + ba + bba)^*$
 - $E_2 = (a + ba + bba)^*(1 + b + bb)$
 - $E_3 = (aa + b)^*$
 - $E_4 = (aa + b)^*(a + bb)^*$
 - $E_5 = (aa + bb + (ab + ba)(aa + bb)^*(ab + ba))^*$
 - $E_6 = (a^*b^*)^*$
 - $E_7 = b(ab)^* + (ba)^*b$
 - $E_8 = (a + bb)^*(b + aa)^*$
2. Faire de même avec l'algorithme de Thompson.
3. Faire de même avec l'algorithme de Glushkov.

Exercice 5 : Algorithmes

1. Donner un algorithme qui prend en entrée deux expressions rationnelles et qui renvoie vrai si elles contiennent au moins un mot commun.
2. Même question pour décider si deux expressions rationnelles représentent exactement le même langage.
3. Donner un algorithme prenant en entrée deux expressions rationnelles E_1 et E_2 , et qui renvoie vrai s'il existe un mot u tel que u est à la fois un suffixe d'un mot reconnu par E_1 et un préfixe d'un mot reconnu par E_2 , et *NON* sinon.