TD 02

Configuration d'un automate

Définition : On appelle *configuration d'un automate* en fonctionnement les valeurs de ses différents composants, à savoir la position de la tête L/E, l'état de l'automate et éventuellement le contenu de la mémoire auxiliaire (lorsqu'elle existe).

Il existe deux configurations spéciales :

1. La configuration ***initiale*** est celle qui correspond à l'état initial q0 et où la tête de L/E est positionnée sur le premier symbole du mot à lire.
2. Une configuration ***finale*** est celle qui correspond à un des états finaux qf et où le mot a été entièrement lu.

Mot reconnu par un automate

On dit qu'un mot **est reconnu par un automate** si, à partir d'une configuration initiale, on arrive à une configuration finale à travers une *succession de configurations* intermédiaires.

Un mot w est ***reconnu par l'automate*** A s'il existe une *configuration successive*

Configuration-initiale (w) $\vdash^*$ configuration-final(w)

$$(q_0, w) \vdash^* (q_f, \varepsilon)$$

La relation $\vdash$ permet de formaliser la notion d'étape élémentaire de calcul d'un automate.
Ainsi on écrira, pour a dans A et v dans A* : $(q, \boldsymbol{a}v) \vdash (\delta(q, a); v)$

Langage reconnu par un automate

On dit qu'un ***langage est reconnu par un automate*** X lorsque tous les mots de ce langage sont reconnus par l'automate on note L(X)

L(X)={$w \in A^*$ / Configuration-initiale (w) $\vdash^*$ configuration-final(w)}

L(X)={ $w \in A^*$ / $(q_0, w) \vdash^* (q, \varepsilon)$, avec $q \in QF$ }

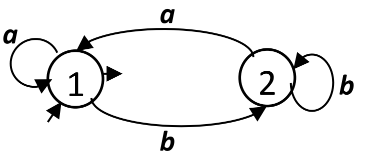
Passage de l'automate vers l'expression régulière

Soit X = (A, Q, q_0, Q_F, δ) un automate à états fini quelconque.
On note par $\mathbf{L_i}$ le langage reconnu par l'automate si son état initial était $\mathbf{q_i}$.
Par conséquent, trouver le langage reconnu par l'automate revient à trouver $\mathbf{L_0}$ étant donné que la reconnaissance commence à partir de l'état initial $\mathbf{q_0}$. L'automate permet d'établir un système d'équations aux langages de la manière suivante :

– si $\delta(q_i, a) = q_j$ alors on écrit : $L_i = aL_j$;
– si $q_i \in Q_F$, alors on écrit : $L_i = \varepsilon$
– si $L_i = \alpha$ et $L_i = \beta$ alors on écrit : $L_i = \alpha|\beta$;

Il suffit ensuite de résoudre le système précédant à des substitutions et en utilisant la règle suivante :
la solution de l'équation $L = \alpha L|\beta$ ($\varepsilon \notin \alpha$) est le langage $L = \alpha^*\beta$ (Le lemme d'Arden)

Exemple :
On cherche à déterminer le langage de l'automate suivant.



On s'intéresse au langage $\boldsymbol{L_1}$ des mots qui passent par l'état 1, et à $\boldsymbol{L_2}$ celui des mots qui passent par 2.
On a les équations suivantes.

- $L_1 = \varepsilon + aL_1 + bL_2$
- $L_2 = aL_1 + bL_2$

Ici ε apparaît puisque l'état 1 est initial.
Par le lemme d'Arden sur la seconde équation, il vient $L_2 = b^*(aL_1)$.
En récrivant la première, on a

- $L_1 = \varepsilon + aL_1 + b(b^*aL_1)$
  $= \varepsilon + aL_1 + b^+aL_1$
  $= b^*aL_1 + \varepsilon$

Par le lemme d'Arden sur cette équation, on obtient finalement que :
$L_1 = (b^*a)^*\varepsilon = (b^*a)^*$

Automate fini déterministe AFD

Définition : Un AEF (A, Q, q_0, Q_F, δ) est dit **déterministe** si les deux conditions sont vérifiées :
- $\forall q_i \in Q$, $\forall a \in A$, il existe ***au plus*** un état q_j tel que $\delta(q_i, a) = q_j$;
- L'automate ne comporte pas de ε-*transitions*.

Algorithme : Déterminiser un AEF sans les ε-transitions

Principe : considérer des ensembles d'états plutôt que des états (dans l'algorithme suivant, chaque ensemble d'états représente un état du futur automate).

1- Partir de l'état initial $E^{(0)} = \{q_0\}$ (c'est l'état initial du nouvel automate) ;
2- Construire $E^{(1)}$ l'ensemble des états obtenus à partir de $E^{(0)}$ par la transition a :

$$E^{(1)} = \bigcup_{q' \in E(0)} \delta(q', a)$$

3- Recommencer l'étape 2 pour toutes les transitions possibles et pour chaque nouvel ensemble $E^{(i)}$;

$$E^{(i)} = \bigcup_{q' \in E(i-1)} \delta(q', a)$$

4- Tous les ensembles contenant au moins un état final du premier automate deviennent finaux ;
5- Renuméroter les états en tant qu'états simples.

Exercice 01 :

I. Trouver, ***intuitivement***, des automates qui acceptent les langages dénotés par les expressions régulières :

1. $(a + bc)^* abc (a + b + c)^*$;
2. $aa(a + b)^* (ab)^* bb$
3. $a(ab)^* (bab)^* bb$
4. $(0+1)^*10 +10(010+1)^*$

II. En utilisant la méthode de ***Glushkov***, construire des automates correspondants aux expressions régulières :

1. $(ab^*a + b(a + b))^*$
2. $0 + 1(0 + 1)^*00$
3. $(ab + b(a + b))^*$
4. $a(b + ab)^* + b^* (a + bb)$.

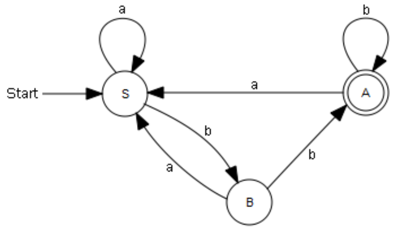
III. En utilisant les ***dérivées***, construire des automates correspondants aux expressions régulières :

1. $(bb^*aa^*b)^*ab^*$
2. $(a + ba)^*bb.b^*a$
3. $a(b + ab)^* + b^* (a + bb)$

Exercice 02 :
Utiliser les dérivées pour vérifier si les langages suivants sont réguliers :

a) $\{ a^{2n} / n \geq 0 \}$; **b)** $\{ a^n b^m / n, m \geq 1 \}$ **c)** ; $\{ a^n b^n / n \geq 0 \}$; **d)** $\{ ww^R / w \in \{a, b\}^* \}$.

Exercice 03 :
On considère l'automate A = (V, Q, Q_F, q_0, δ) suivant.



1. Expliciter V, Q, Q_F, q_0 et δ (on représentera δ par sa table de transition).
2. Donner 3 mots ***acceptés*** par A et 3 mots ***refusés*** par *A* .
3. Donner une *expression régulière* a dénotant L(*A*).
4. Calculer l'expression régulière en résolvant le système d'équations associé

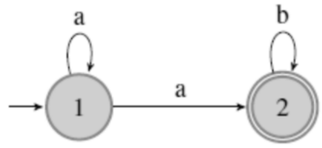
Exercice 04 :

Pour chacun des langages suivants, construire un automate d'états finis qui l'accepte :

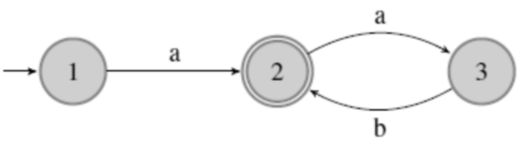
1. $L_1 = \{ w \in \{a, b, c\}^* / w$ contient au plus deux occurrences de la lettre '*a*' } ;
2. L_2 = l'ensemble des représentations binaires des nombres pairs
3. L_3 = l'ensemble des représentations décimales des multiples de 2
4. $L_4 = \{ w \in \{a, b\}^* / w = a^n b^m a$ ou $w = aba^n$; n, m ≥ 1 } ;
5. $L_5 = \{ w \in \{a, b\}^* / |w|$ est divisible par 3 } ;
6. $L_6 = \{w \in \{a, b\}^*$ t.q. dans ***w***, tout ***a*** est suivi de deux ***b*** exactement}
7. $L_7 = \{ w \in \{0, 1\}^* / w$ ne contient pas la sous chaîne «010» } ;
8. $L_8 = \{w \in \{a, b\}^*$ t.q. ***w*** ne contient pas le facteur ***aba***}

Exercice 05 :

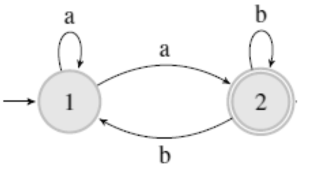
Déterminer une expression régulière équivalente à chacun de ces automates



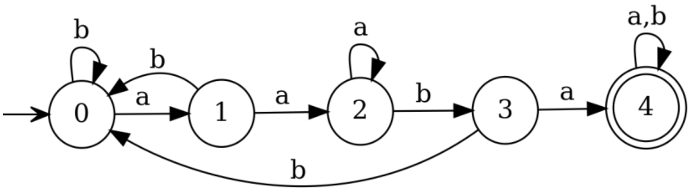
Automate M1



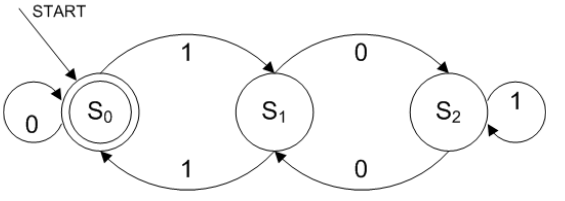
Automate M2



Automate M3



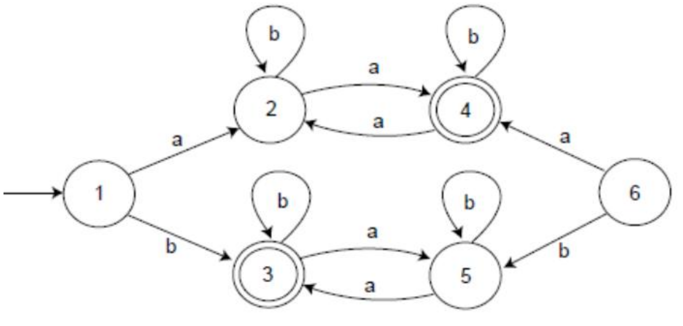
Automate M4



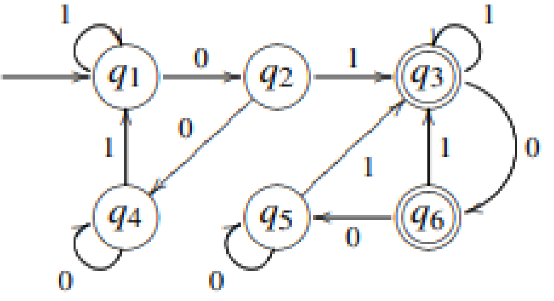
Automate M5

Exercice 06 :

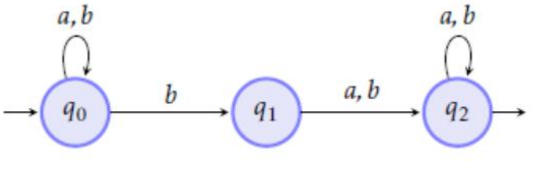
Minimiser les automates suivants



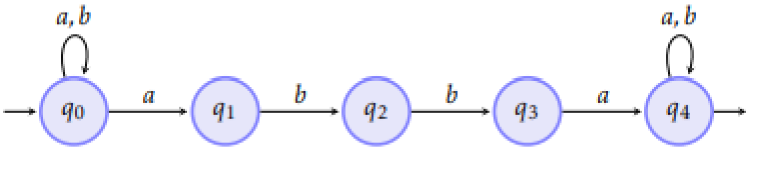
Automate A1



Automate A2



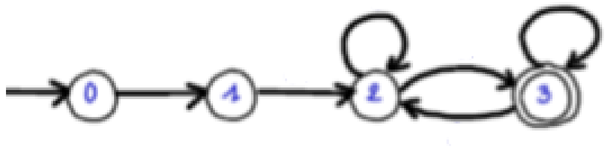
Automate A3

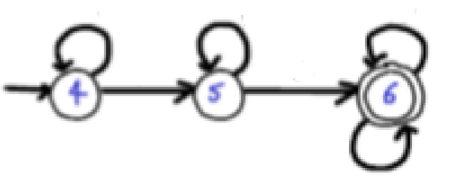


Automate A4

Exercice 09 :

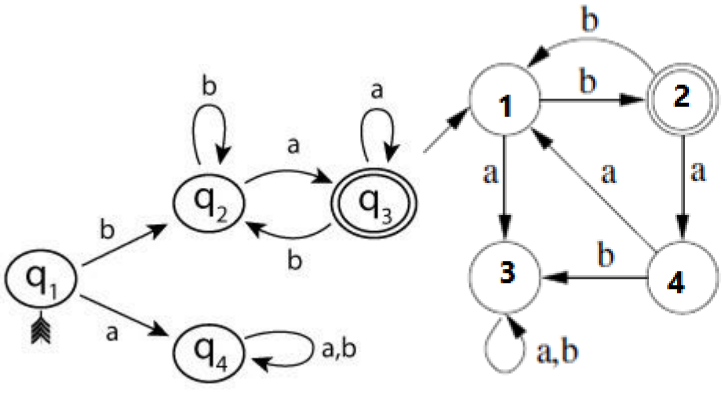
Soient A_1 et A_2 deux automates, L_1 et L_2 leurs langages respectifs.





a. Construire l'automate de L_1L_2.
b. Même question mais sans ε-transition.
c. Construire l'automate de $L_1 + L_2$
d. Construire l'automate de $(L_1)^*L_2$
e. Construire l'automate de $L_1 \setminus L_2$

Exercice 10 :
Déterminer une expression régulière du langage accepté par l'automate suivant (en utilise la méthode d'élimination d'états) :



Exercice 11:

Parmi les expressions rationnelles et les automates suivants dire quels sont les automates et les expressions rationnelles qui représentent le même langage.

$(ab^*a + b(a + b))^*$ $(ab + b(a + b))^*$