

Exercice 1 :

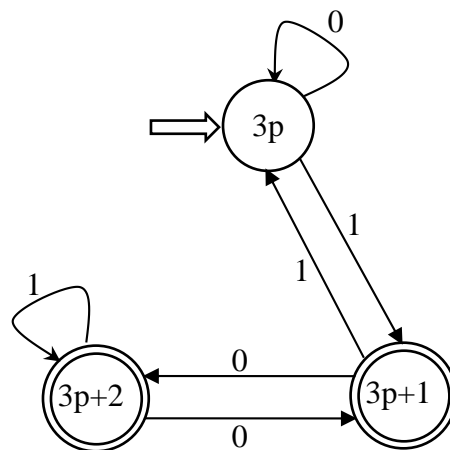
- 1) $L_4 = \{w \in \{0, 1\}^* / w \text{ n'est pas un multiple de 3}\}$

Exemple : $L_4 = \{1, 01, 10, 010, 0111, \dots\}$

Remarque : Les mots de L_4 sont les nombres binaires qui ne sont pas multiples de 3 donc multiples de 3 plus 1 ou multiples de 3 plus 2. Ainsi, l'automate comportera 3 états ($3p, 3p+1, 3p+2$).

Si w est un nombre binaire, alors $w0$ représente $2*w$ et $w1$ représente $2*w+1$ (voir question 10, exercice 6, série1). Si un nombre est un multiple de 3, et si on lit 0, il reste multiple de 3. Si on lit un 1, il devient multiple de 3 plus 1. Les états d'acceptation sont ceux correspondant aux multiples de 3 plus 1 et multiple de 3 plus 2.

D'où l'automate :



C'est un automate simple déterministe complet.

- 2) $L_5 = \{w / w \in \{a,b\}^*, |w|_a = 2*n \text{ et } |w|_b = 2*m, n,m \geq 0\}$

Exemple : $L_5 = \{\epsilon, aa, bb, abba, bbaa, aaaa, bbbb, abab, \dots\}$

Remarque : Les mots de L_5 sont les mots composés de **a** et **b** tels que le nombre de **a** est pair et le nombre de **b** est pair.

Dans un mot quelconque, le nombre de **a** et le nombre de **b** peuvent être l'une des combinaisons suivantes : (Pair, Pair), (Pair, Impair), (Impair, Pair), (Impair, Impair). La première composante de

chaque couple représente le nombre de **a** et la deuxième composante représente le nombre de **b**. Donc, l'automate contient quatre états : PP, PI, IP et II représentant dans l'ordre les quatre combinaisons possibles d'un mot données précédemment.

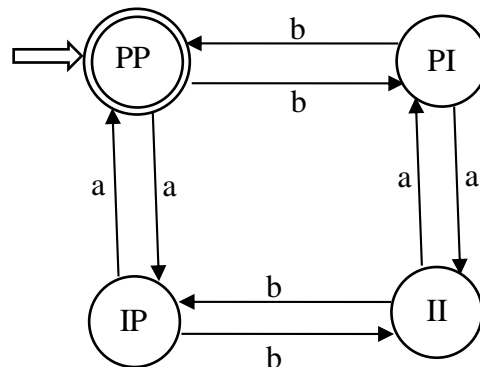
Si un mot est dans la configuration (Pair, Pair) alors l'automate se trouve dans l'état PP. Dans ce cas, si on lit un a alors le mot passe à la configuration (Impair, Pair) et l'automate passe à l'état IP. Si on lit un b, le mot devient (Pair, Impair) et l'automate transite vers l'état PI. Le raisonnement est

Théorie des Langages Série 2

LES AUTOMATES A ETATS FINIS & LANGAGES REGULIERS

similaire pour les autres états. L'état initial est PP car le mot vide correspond à la configuration (Pair, Pair). En effet, $|\varepsilon|_a = 0$ et $|\varepsilon|_b = 0$. L'état d'acceptation est (Pair, Pair).

D'où l'automate :



C'est un automate simple déterministe complet.

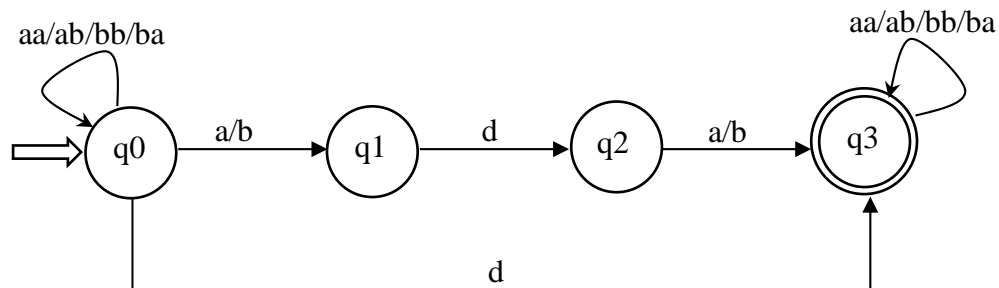
3) $L_6 = \{w_1dw_2 \mid w_1, w_2 \in \{a, b\}^* \text{ et } |w_1| + |w_2| \text{ est pair}\}$

Exemple : $L_6 = \{d, ada, adb, abdba, bbdaa, aaadabb, \dots\}$

Remarque : Les mots de L_6 sont composés de deux sous mots w_1 et w_2 séparés par un d . La somme des longueurs de w_1 et w_2 est paire.

Pour que la somme des longueurs de w_1 et w_2 soit paire, il faut que les deux longueurs de w_1 et w_2 soient paires (les deux en même temps), ou soient impaires (les deux en même temps). Le plus petit mot est d (le séparateur).

D'où l'automate :

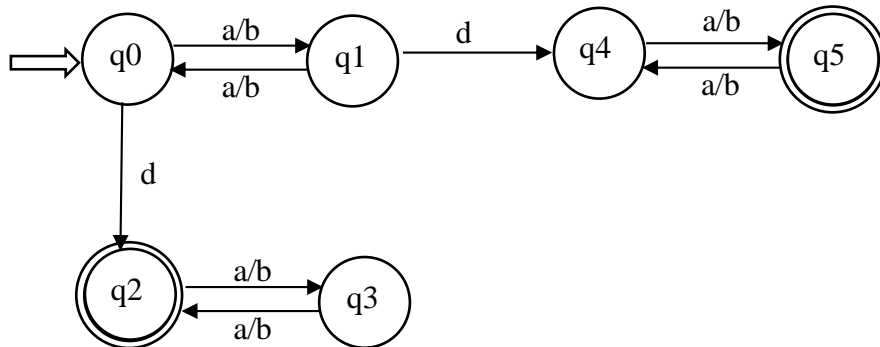


C'est un automate généralisé.

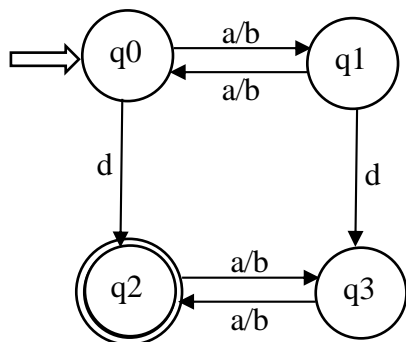
Explication de l'automate : Dans l'état q_0 , l'automate lit tout mot de $\{a, b\}^*$ de longueur paire. Soit, il transite par d directement à l'état q_3 (w_1 est de longueur paire), soit il transite par a ou b vers q_1 (w_1 est de longueur impaire). De l'état q_1 , il transite par d vers l'état q_2 afin de lire le mot w_2 qui doit être de longueur impaire. En effet dans l'état q_2 , il lit d'abord une lettre a ou b et transite vers q_3 . Dans l'état q_3 , il peut lire tout mot de $\{a, b\}^*$ de longueur paire.

LES AUTOMATES A ETATS FINIS & LANGAGES REGULIERS

L'automate suivant reconnaît aussi les mots du langage L_8 . Les transitions entre les états q_0 et q_1 permettent de lire w_1 et contrôler sa parité. La longueur de w_1 est paire (resp impaire) dans l'état q_0 (resp q_1). Les transitions entre q_2 et q_3 permettent de lire w_2 de longueur paire (q_2 est final) alors que celles entre q_4 et q_5 permettent de lire w_2 de longueur impaire (q_5 est final).



Un troisième automate équivalent est donné ci-dessous qui reconnaît L_6 . De l'état q_1 où la longueur de w_1 est impaire, l'automate transite par d vers q_3 . Partant de q_3 , le mot w_2 serait de longueur impaire.

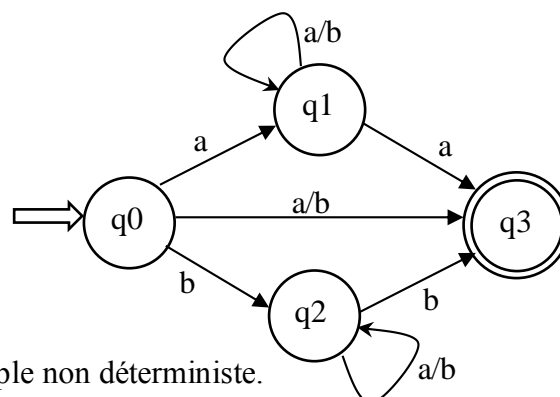


4) $L_7 = \{w \in \{a, b\}^* / w \text{ commence et se termine par le même symbole}\}$

Exemple : $L_7 = \{a, b, aa, bb, abba, bab, aababa, baabb, \dots\}$

Remarque : Les mots de L_7 commencent par **a** (respectivement **b**) et se terminent par **a** (respectivement **b**) et entre la première et la dernière lettre on peut trouver n'importe quelle combinaison de lettres. Les mots **a** et **b** sont des cas particuliers (ils appartiennent au langage).

D'où l'automate :



C'est un automate simple non déterministe.

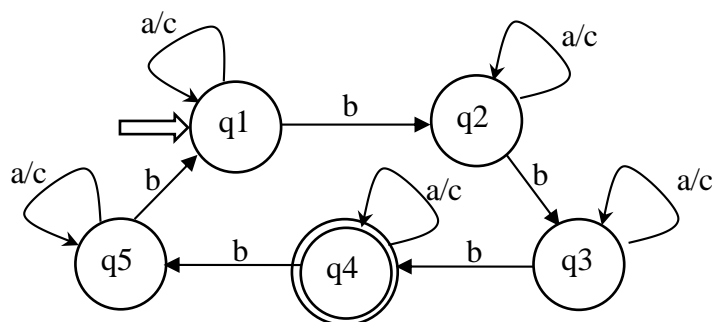
LES AUTOMATES A ETATS FINIS & LANGAGES REGULIERS

5) $L_8 = \{w \in \{a, b, c\}^* / w \text{ commence et se termine par } b \text{ et } |w|_b \equiv 3[5]\}$

Exemple : $L_8 = \{bbb, baabab, bbbbbb, baababaababaab, \dots\}$

Remarque : Les mots de L_8 commencent par **b** et se terminent par **b** et contiennent un nombre de **b** total multiple de 5 plus 3. Donc le nombre de **b** peut être 3 ou 8 ou 13, Après le premier **b**, avant le dernier **b** ou entre les **b** du milieu on peut trouver n'importe quelle suite aléatoire de **a** et **c**.

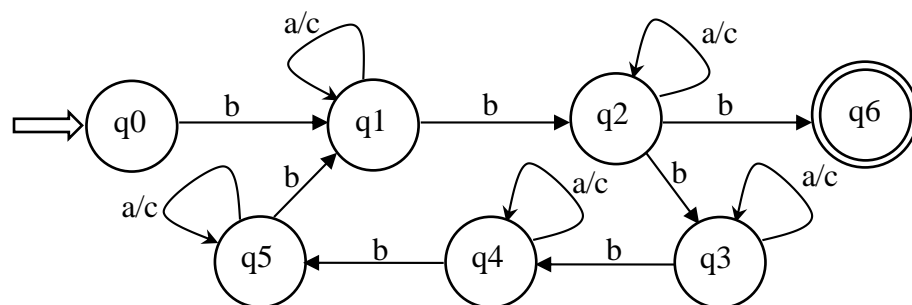
Dans un premier temps, on donne un automate d'états finis ci-dessus qui **reconnait** les mots de $\{a, b, c\}^*$ avec $|w|_b \equiv 3[5]$. Cet automate contient cinq états correspondants aux différentes possibilités du nombre de **c** modulo 5 : $5p, 5p+1, \dots, 5p+4$. Dans ce cas, l'état **q4** serait final.



Notons que dans cet automate, les mots peuvent commencer et se terminer par **a** ou par **c**.

Mais, les mots de L_8 doivent commencer et se terminer par la lettre **b**. On ajoute un état **q0** pour lire le premier **b** vers **q1** et dans ce cas le nombre de **b** devient $5p+1$. Donc, l'état **q1** va correspondre aux mots $5p+1$, l'état **q2** aux mots $5p+2$, l'état **q3** aux mots $5p+3$, etc.... L'état **q3** ne peut pas être final car il permet de lire des **a** et des **c** et le mot peut se terminer par **a** ou par **c**. Donc, on ajoute un état final **q6** et une transition de **q2** (qui correspond aux mots $5p+2$) par **b** vers **q6**. Et ainsi, les mots se terminent par **b**.

D'où l'automate :



C'est un automate simple non déterministe.