Solution - TD 3 - Constructions d'automates, déterminisation

Licence 3 - Université de Bordeaux

Solution de l'exercice 1 :

1. On rappelle le critère de divisibilité par 3 des nombres entiers (en base 10) : la somme des chiffres doit être divisible par 3.
   L'automate qui reconnait les entiers (décimaux) divisibles par 3 aura trois états : s_0, s_1, s_2 – un pour chaque reste à la division par 3. L'état s_0 est à la fois initial et final.
   De l'état s_i ($i \in \{0, 1, 2\}$) en lisant la lettre $d \in \{0, \ldots, 9\}$ on va à l'état s_j, où j est égal au reste de $s_i + d$ à la division par 3.
2. Nous allons construire l'automate qui reconnait l'ensemble des représentations binaires des entiers positifs divisibles par 3, en les lisant depuis le bit le plus fort. Par exemple, l'entier 6, écrit en binaire 110, sera lu en tant que 110.
   Imaginons que l'automate ait déjà lu le mot w. Notons e_w l'entier représenté par w. Lorsque l'automate lit une nouvelle lettre, le mot lu devient alors :
   $$\begin{cases} w0 & \text{si la lettre lue est 0;} \\ w1 & \text{si la lettre lue est 1.} \end{cases}$$
   Les entiers représentés par ces deux mots sont alors égaux à :
   $$\begin{cases} e_{w0} = 2e_w + 0; \\ e_{w1} = 2e_w + 1. \end{cases}$$
   On cherche à determiner la divisibilité par 3 de e_w, pour cela, on écrit e_x sous la forme $3k$, $3k + 1$ ou $3k + 2$.
   Lorsque l'on lit une lettre, la forme de l'entier représenté par le mot change et son évolution est décrite par le tableau suivant :

e_w	e_{w0}	e_{w1}
$3k + 0$	$2(3k + 0) + 0 = 3k'$	$2(3k + 0) + 1 = 3k' + 1$
$3k + 1$	$2(3k + 1) + 0 = 3k' + 2$	$2(3k + 1) + 1 = 3k' + 0$
$3k + 2$	$2(3k + 2) + 0 = 3k' + 1$	$2(3k + 2) + 1 = 3k' + 2$

   Si l'on code chaque forme par un état, on obtient alors l'automate suivant :

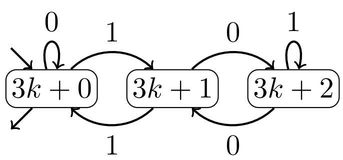


Figure 1 – Bit fort en tête.

Comme le mot de départ est le mot vide ϵ, il code l'entier 0 qui est de la forme $3k+0$. L'état initial de l'automate est donc l'état $3k+0$. Comme l'automate doit reconnaître les entiers divisibles par 3, il doit donc accepter les mots w dont l'entier e_w est de la forme $3k+0$. L'automate a donc un seul état final qui est l'état $3k+0$.

Nous allons maintenant construire l'automate qui reconnait l'ensemble des représentations binaires des entiers positifs divisibles par 3, avec le bit faible en tête. Par exemple, l'entier 6, écrit en binaire 110, sera lu en tant que 011.

Soit $w = a_1 \dots a_n$ le mot lu, $a_i \in \{0,1\}$. L'entier e_w représenté par $w = a_1 \dots a_n$ est $e_w = \sum_{i=1}^{n} a_i \cdot 2^{n-i}$.

Nous nous interessons donc à la divisibilité par 3 des puissances de 2. On peut vérifier que 2^i est de la forme $3k+1$ si i est *pair*, et de la forme $3k+2$ si i est *impair*.

Donc, e_w peut s'écrire comme $e_w = 3k + r$, avec :

$$r = \begin{cases} a_1 + 2a_2 + a_3 + \cdots + 2a_n & \text{si } |w| = n \text{ est pair} \\ a_1 + 2a_2 + \dots 2a_{n-1} + a_n & \text{si } |w| = n \text{ est impair} \end{cases}$$

On obtient alors l'automate suivant (chaque état code le reste à la division par 3 et la parité de la longueur du préfixe lu jusqu'à là) :

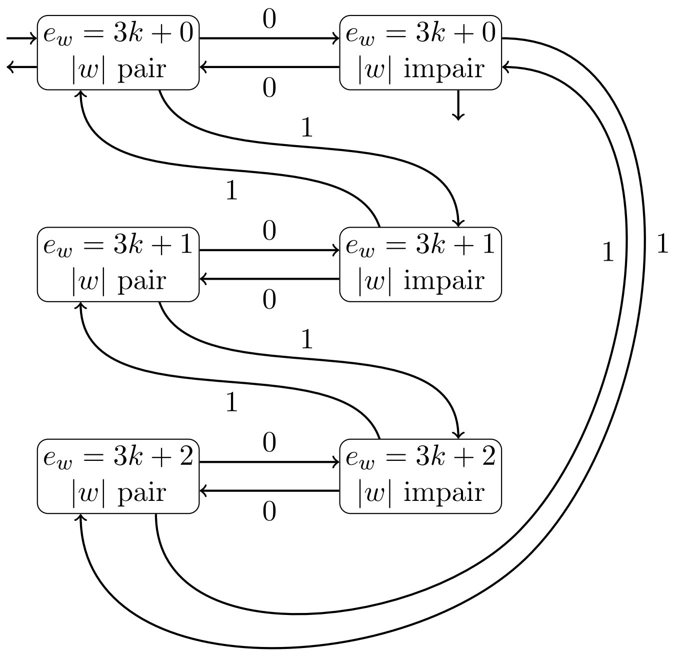
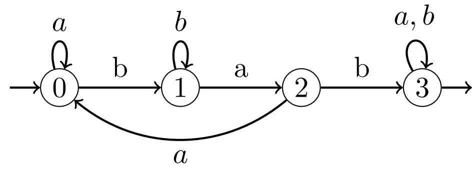


Figure 2 – Bit faible en tête.

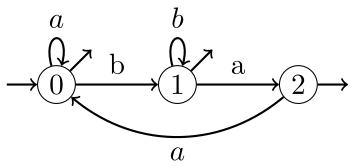
Solution de l'exercice 2 :

Pour tout l'exercice, on note $A = \{a, b\}$

1. Expression régulière : $A^*(bab)A^*$.
   Automate :



2. Expression régulière : $(a^*b^+aa)^*(a^* + a^*b^+ + a^*b^+a)$.
   Automate :

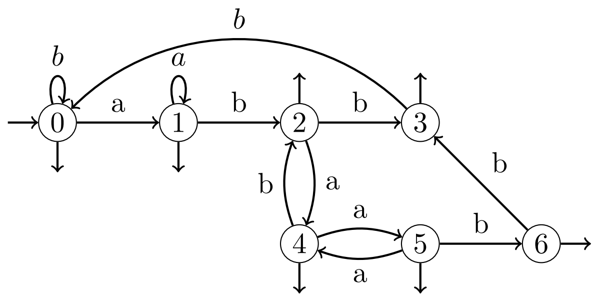
   

   Justification : On montre par induction sur la longueur $n = |u|$ du mot u l'invariant suivant

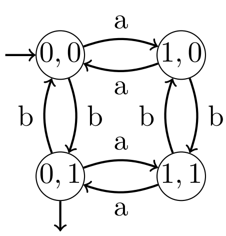
   (a) $0 \xrightarrow{u} 0$ si u ne contient pas bab comme facteur et soit finit par aa, ou $u = \epsilon$, ou $u = a$;
   (b) $0 \xrightarrow{u} 1$ si u ne contient pas bab comme facteur et soit finit par ab ou bb, ou $u = b$;
   (c) $0 \xrightarrow{u} 2$ si u ne contient pas bab comme facteur et finit par ba.

   (a) $n = 0, 1$: évident
   (b) Pour $n \geq 2$ on regarde les deux cas possibles : soit $u = va$ ou $u = vb$, avec $|v| \geq 1$.
   Si $u = va$ on verifie d'abord que notre invariant est satisfait pour $|v| = 1$. Ensuite, pour $|v| \geq 2$ on regarde les 2 dernières lettres de v. Si c'est aa alors par hypothèse d'induction, $0 \xrightarrow{v} 0$, donc $0 \xrightarrow{va} 0$, et notre invariant est satisfait pour va. Si c'est ab ou bb, alors $0 \xrightarrow{v} 1$ par hypothèse d'induction, donc $0 \xrightarrow{va} 2$, et notre invariant est satisfait pour va. Si c'est ba, alors $0 \xrightarrow{v} 2$ par hypothèse d'induction, donc $0 \xrightarrow{va} 0$, et notre invariant est satisfait pour va.
   Le cas $u = vb$ est similaire.

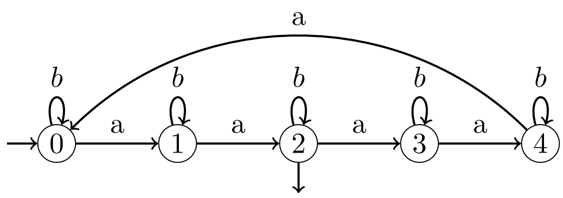
3. Automate :

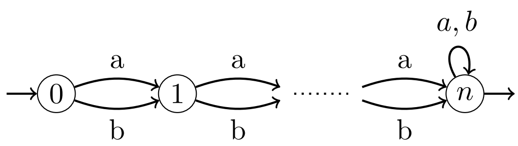
4. Automate :

5. Expression régulière : $b^*ab^*a(b^*ab^*ab^*ab^*ab^*a)^*b^*$.
   Automate :

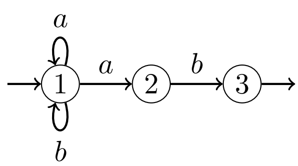


6. Expression régulière : $(a+b)^n(a+b)^*$.
   Automate :

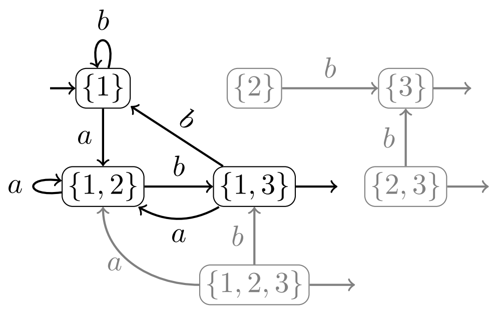


Solution de l'exercice 3 :

On commence par donner un automate non-déterministe :



En appliquant l'algorithme de déterminisation on obtient (on a mis en gris les états et transitions non nécessaires, car non accessibles depuis l'état initial) :



Solution de l'exercice 4 :

1. On sait que si un langage a un nombre infini de résiduels, alors il n'est pas régulier.
   Or le langage $L = \{a^n b^n \mid n \geq 0\}$ a un nombre infini de résiduels car

$$(a^k)^{-1}L = \{a^{n-k}b^n \mid n \geq k\},$$

$$(a^k)^{-1}L \neq (a^j)^{-1}L \qquad \text{si } k \neq j$$

   Donc, les résiduels $\epsilon^{-1}L, a^{-1}L, (aa)^{-1}L, \ldots$ sont tous différents.

2. Soit K l'ensemble des mots sur l'alphabet $\{a, b\}$ contenant autant de a que de b.
   On observe que $L = K \cap a^*b^*$.
   Preuve par contradition : Si K était régulier, alors L serait également régulier, car a^*b^* est régulier et l'intersection de deux réguliers est un régulier. Mais on a déjà vu que L n'est pas régulier, on obtient donc une contradiction.
   On aurait pu montrer aussi que les résiduels $(a^k)^{-1}K$ sont deux à deux différents, comme pour L.