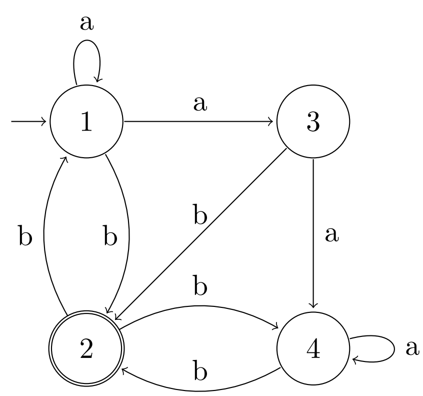
Université Paris 7-Denis Diderot | Examen du 21 mai 2012
L2
Automates finis
AF4

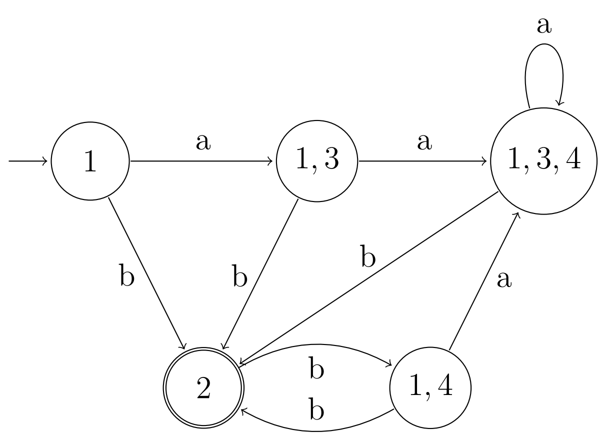
Corrigé

Exercice 1

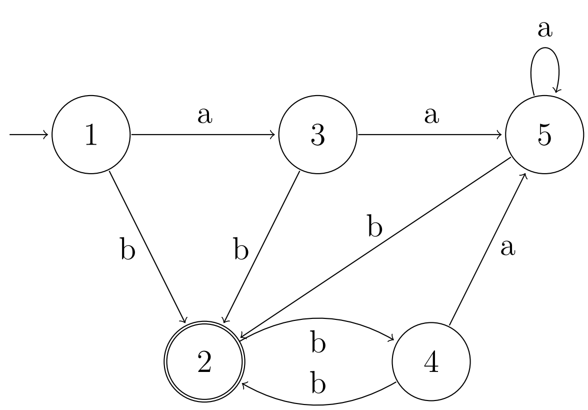


On considère l'automate fini :

Question 1 : Déterminiser cet automate.



Voici le déterminisé :



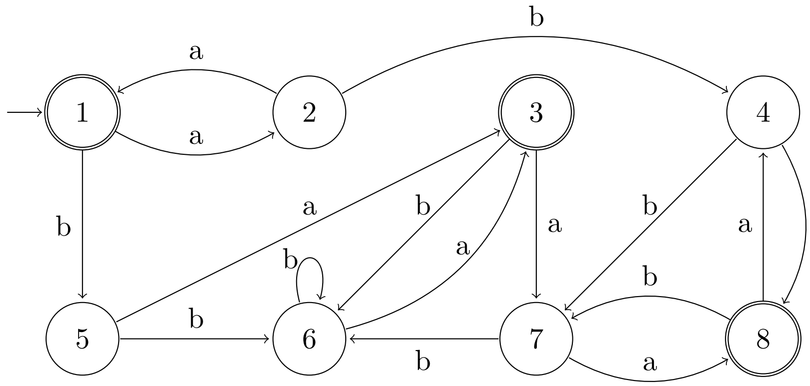
En renumérotant les états :

Question 2 : Si L le est le langage reconnu par cet automate, calculer par la méthode de Mac Naughton et Yamada une expression rationnelle de L.

Soit $Q = \{1, 2, 3, 4, 5\}$, il faut calculer $L = L^{Q}_{1\,2}$. Posons $P = Q \backslash \{2\}$. D'après Mac Naughton et Yamada, $L^{Q}_{1\,2} = L^{P}_{1\,2}.(L^{P}_{2\,2})^{*}$.
Clairement, les chemins qui vont de l'état 1 à l'état 2 sans passer par l'état 2 en intermédiaire ne passent pas par l'état 4, et sont étiquetés par $\{b, ab, aaa^{*}b\} = a^{*}b$.
Clairement, les chemins qui vont de l'état 2 à l'état 2 sans passer par l'état 2 en intermédiaire ne passent pas par les états 1 et 3, et sont étiquetés par $\{bb, baa^{*}b\} = ba^{*}b$.
On a donc $L = a^{*}b.(ba^{*}b)^{*}$.

Exercice 2

On considère l'automate fini :

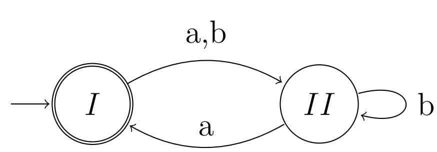


Question : Calculer l'équivalence de Nérode associée à cet automate. En déduire l'automate minimal qui reconnaît le même langage.

L'équivalence à l'ordre 0 a deux classes : les états d'acceptations (classe notée ici I), et ceux qui ne le sont pas (classe notée ici II). Calculons l'équivalence d'ordre 1 :

	$\underline{1}$	2	$\underline{3}$	4	5	6	7	$\underline{8}$
a	2	$\underline{1}$	7	$\underline{8}$	$\underline{3}$	$\underline{3}$	$\underline{8}$	4
b	5	4	6	7	6	6	6	7

L'équivalence à l'ordre 1 étant la même que l'équivalence à l'ordre 0, c'est donc l'équivalence de Nérode. On obtient l'automate minimal en confondant les états équivalents :



Exercice 3

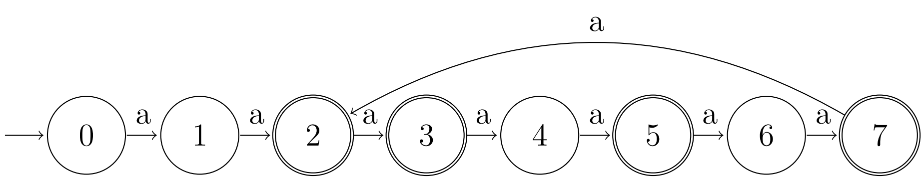
Soit l'alphabet $A = \{a, b\}$. Montrer que le langage $L = \{(a^n b)^n \mid n \in I\!N\}$ n'est pas reconnaissable.

Supposons que L soit reconnaissable. Soit alors N le nombre augmenté de 1 des états d'un automate fini (que l'on peut supposer déterministe) qui le reconnaît, et considérons le mot $w = (a^N b)^N$ qui, appartenant à L, est reconnu par cet automate. Le chemin qui mène à l'acceptation de ce mot passe lors de la lecture des N premiers a au moins 2 fois par le même état, et donc il existe une factorisation de w en $w = \alpha.u.\beta$ où $\alpha = a^i$, $u = a^q$ et $\beta = a^{N-i-q}b.(a^N b)^{N-1}$ avec $q > 0$, telle que la lecture de α et celle de $\alpha.u$ mènent au même état. Par conséquent la lecture de $\alpha\beta$ amène au même état que celle de $\alpha.u.\beta$, c'est à dire w, et donc à un état d'acceptation. Le mot $\alpha\beta$ est donc reconnu par cet automate, et devrait donc appartenir à L. Or $\alpha\beta = a^i.a^{N-i-q}b.(a^N b)^{N-1} = a^{N-q}b.(a^N b)^{N-1}$ n'appartient clairement pas à L. Contradiction.

Exercice 4

Si $L \subset A^*$, on note $L_{<n>} = L \cap A^n = \{w \in L \mid |w| = n\}$, et $\Theta(L) = \{n \in I\!N \mid L_{<n>} \neq \emptyset\}$. On dira que L est complet (pour les longueurs) si $\Theta(L) = I\!N$, et qu'il est presque complet si $\Theta(L)$ et $I\!N$ ne diffèrent que par un nombre fini d'entiers.

Question 1 : Sur l'alphabet $B = \{a\}$, soit le langage $K = a^3(a^2)^* + a^2(a^3)^*$. Donner la représentation sagittale (*i.e.* le dessin du graphe) d'un automate fini déterministe qui reconnaît K.



Question 2 : Calculer $\Theta(K)$. K est-il complet ? est-il presque complet ?

$\Theta(K) = \{3 + 2i \mid i \in I\!N\} \bigcup \{2 + 3i \mid i \in I\!N\} = \{3 + 6i, 5 + 6i, 7 + 6i, \mid i \in I\!N\} \bigcup \{2 + 6i, 5 + 6i \mid i \in I\!N\} = \{2 + 6i, 3 + 6i, 5 + 6i, 7 + 6i, \mid i \in I\!N\}$.
K n'est pas complet car $1 \notin \Theta(K)$. Si $n = 0 \ (mod\ 6)$ alors $n \notin \Theta(K)$, et il existe une infinité d'entiers congrus à 0 modulo 6, donc K n'est pas presque complet.

Question 3 : Montrer que si $H \in Rat(B^*)$, alors soit il est fini, soit il existe un entier $i \geq 0$ et un entier $k > 0$ tels que $H = F \cup \bigcup_{j \in J} a^i (a^k)^* a^j$ où F est un ensemble fini de mots tel que $f \in F \Longrightarrow |f| < i$ et J est un ensemble (fini) d'entiers inclus dans $\{0, 1, \ldots, k-1\}$.
Aide : on pourra s'inspirer de l'exemple de la question 1, et donner la forme générale d'un automate fini déterministe qui reconnaît un langage rationnel sur un alphabet à une lettre, pour en déduire la réponse à la question.

Dans un automate déterministe reconnaissant un langage sur un alphabet à une lettre ne part d'un état qu'une seule transition. L'automate se développe donc de manière linéaire, et on peut numéroter les états avec le nombre de fois la lettre lue depuis l'état de départ (celui-ci étant numéroté 0, donc). Au bout d'un certain temps, du fait de la finitude du nombre d'états, on finit par revenir sur un état déjà rencontré. Il n'existe donc qu'un seul type d'automate fini déterministe sur un alphabet à un lettre : la lecture de $i \geq 0$ occurrences de la lettre amène au premier état se trouvant dans une boucle de longueur $k > 0$.

En posant $F = H \bigcap \{f \mid |f| < i\}$, et si J est l'ensemble des entiers j tels que $i+j$ est le numéro d'un état d'acceptation, J est donc bien fini et l'on a $H = F \cup \bigcup_{j \in J} a^i (a^k)^* a^j$.

Question 4 : Donner un algorithme qui résout le problème :
Donnée : Une expression rationnelle d'un langage $K \in Rat(B^*)$
Question : k est-il complet ?
et un algorithme qui résout le problème :
Donnée : Une expression rationnelle d'un langage $K \in Rat(B^*)$
Question : k est-il presque complet ?

Pour que K soit complet il faut que tous les états de l'automate déterministe soient des états d'acceptation, et pour qu'il soit presque complet, il faut que tous les états de la boucle de l'automate déterministe soient des états d'acceptation. Ces deux conditions se testent de façon finie.
Pour résoudre le premier problème, il suffit donc de construire à partir de l'expression rationnelle un automate fini (par exemple grâce à l'algorithme de Glushkov) reconnaissant le langage décrit, de déterminiser cet automate (par l'algorithme de déterminisation), et de vérifier si l'automate obtenu a tous ses états qui sont des états d'acceptation, ou non.
Pour résoudre le second problème, on construit de même à partir de l'expression rationnelle un automate fini déterministe reconnaissant le langage décrit, et il suffit de regarder à quel état i arrive une transition et de vérifier si l'automate obtenu a tous ses états à partir de cet état i qui sont des états d'acceptation, ou non.

Question 5 : Donner un algorithme qui résout le problème :
Donnée : Deux expressions rationnelles sur l'alphabet B représentant deux langages H_1 et H_2
Question : A-t-on $\Theta(H_1) = \Theta(H_2)$?
et un algorithme qui résout le problème :
Donnée : Deux expressions rationnelles sur l'alphabet B représentant deux langages H_1 et H_2
Question : $\Theta(H_1)$ et $\Theta(H_2)$ diffèrent-ils d'un nombre fini d'entiers ?

On a $\Theta(H_1) = \Theta(H_2)$ si et seulement si $H_1 = H_2$, et on a $\Theta(H_1)$ et $\Theta(H_2)$ diffèrent d'un nombre fini d'entiers si et seulement si H_1 et H_2 diffèrent d'un nombre fini de mots.
Pour résoudre le premier problème, on construit à partir des expressions rationnelles des automates finis déterministes reconnaissant H_1 et H_2, et par l'algorithme du produit cartésien à partir de ceux-ci un automate fini qui reconnaît $(H_1 \bigcap \overline{H_2}) \bigcup (\overline{H_1} \bigcap H_2)$. Il suffit de vérifier si l'automate obtenu a tous ses états qui sont des états d'acceptation, ou non.
Pour résoudre le second problème, il suffit de construire le même automate fini déterministe que ci-dessus, et de regarder à quel état i arrive une transition et vérifier si l'automate obtenu a tous ses états à partir de cet état i qui sont des états d'acceptation, ou non.

Question 6 : Soit A un alphabet quelconque. On considère le morphisme de monoïdes $\pi : A^* \to B^*$ défini par : $\forall x \in A, \pi(x) = a$. Vérifier que $L \in Rat(A^*) \Longrightarrow \pi(L) \in Rat(B^*)$, et montrer que $n \in \Theta(L) \Longleftrightarrow n \in \Theta(\pi(L))$. En déduire des algorithmes qui résolvent les problèmes :
Donnée : Deux expressions rationnelles sur un alphabet A représentant deux langages K_1 et K_2
Question : A-t-on $\Theta(K_1) = \Theta(K_2)$?
et
Donnée : Deux expressions rationnelles sur un alphabet A représentant deux langages K_1 et K_2
Question : $\Theta(K_1)$ et $\Theta(K_2)$ diffèrent-ils d'un nombre fini d'entiers ?

La rationnalité étant conservée par morphisme, si $L \in Rat(A^*)$ alors $\pi(L) \in Rat(B^*)$. $n \in \Theta(L) \implies \exists w \in L\ : \Theta(w) = n$. Si $f = \pi(w)$, alors, on a $n = \Theta(f) = \Theta(\pi(w))$, et donc $n \in \Theta(\pi(L))$. Réciproquement, si $n \in \Theta(\pi(L))$, alors $\exists w \in L\ \Theta(\pi(w)) = n$, mais $\Theta(\pi(w)) = \Theta(w)$, donc $n \in \Theta(L)$.
En appliquant à une expression rationnelle sur un alphabet A le morphisme φ (qui consiste à confondre toutes les lettres en leur substituant l'unique lettre a), on obtient une expression rationnelle sur B du langage obtenu par le morphisme à partir du langage décrit.
En faisant précéder cette construction aux algorithmes de la question précédente, on obtient les algorithmes qui résolvent respectivement les problèmes pour cette question.

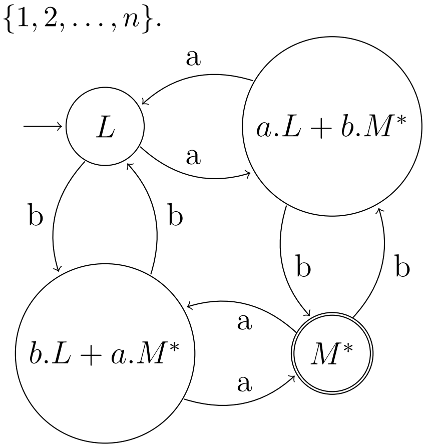
Exercice 5

On considère sur l'alphabet $A = \{a, b\}$ le langage $M = (aa+bb)+(ab+ba).(aa+bb)^*(ab+ba)$ et le langage $L = (aa + bb)^*(ab + ba).M^*$.

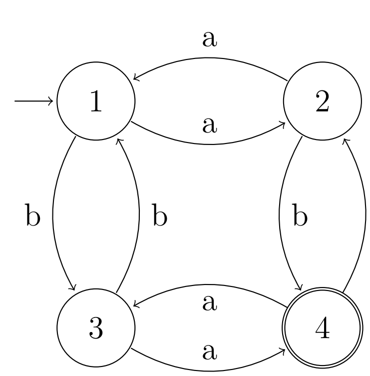
Question 1 : Calculer les résiduels de L (les exprimer en fonction de L et M).

On remarque que si l'on échange a et b, le langage M reste identique à lui-même, et qu'il en est de même pour L. Les calculs par rapport à s et par rapport à b seront donc identiques à cet échange près.
• Le résiduel de L par rapport au mot vide est le langage L lui-même.
• Le résiduel de L par rapport à a vaut :
$a^{-1}.[(aa + bb)^*(ab + ba).M^*] = (a^{-1}.[(aa + bb)^*]).(ab + ba).M^* + a^{-1}.[(ab + ba).M^*]$, car $1 \in (aa+bb)^*$, et comme $a^{-1}.[(aa+bb)^*] = (a^{-1}.[(aa+bb)]).(aa+bb)^*$, et $a^{-1}.[(aa+bb)] = a$ et $a^{-1}.[(ab + ba)] = b$, on a $a^{-1}.L = a.(aa + bb)^*(ab + ba).M^* + b.M^*$, soit
$a^{-1}.L = a.L + b.M^*$
• Le résiduel de L par rapport à b vaut : $b^{-1}.L = b.L + a.M^*$ (par symétrie)
• Le résiduel de L par rapport à aa vaut : $a^{-1}.[a.L + b.M^*] = L$, déjà trouvé
• Le résiduel de L par rapport à ab vaut : $b^{-1}.[a.L + b.M^*] = M^*$
• Le résiduel de L par rapport à ba vaut : M^* (par symétrie), déjà trouvé
• Le résiduel de L par rapport à bb vaut : L (par symétrie), déjà trouvé
• Le résiduel de L par rapport à aba vaut : $a^{-1}.[M^*] = (a^{-1}.[M]).M^*$, or $a^{-1}.[M] = a^{-1}.[(aa+bb) + (ab + ba).(aa + bb)^*(ab + ba)] = a + b.(aa + bb)^*(ab + ba)$, donc $a^{-1}.[M^*] = a.M^* + b.(aa + bb)^*(ab + ba).M^* = a.M^* + b.L$, déjà trouvé
• Le résiduel de L par rapport à abb vaut : $b.M^* + a.L$ (par symétrie), déjà trouvé
On a donc calculé tous les résiduels de L, qui sont au nombre de 4 : L, $a.L+b.M^*$, $b.L+a.M^*$ et M^*.

Question 2 : En déduire l'automate minimal $\mathcal{A} =< A, Q, \delta, \{1\}, F >$ reconnaissant L, où $Q = \{1, 2, \ldots, n\}$.



en renumérotant les états :



Question 3 : Calculer le monoïde de transitions T de cet automate. *(rappel : le monoïde de transitions d'un automate est le pus petit monoïde qui contient les applications $\overline{x}$ de Q dans Q définies pour chaque lettre $x \in A$ par : $\forall q \in Q, \bar{x}(q) = \delta(q, x)$.*

$\overline{1}$	$\bar{a}$	$\bar{b}$	$\overline{aa}$	$\overline{ab}$	$\overline{ba}$	$\overline{bb}$
1	2	3	1	4	4	1
2	1	4	2	3	3	2
3	4	1	3	2	2	3
4	3	2	4	1	1	4

On a $\overline{aa} = \overline{bb} = \overline{1}$ et $\overline{ba} = \overline{ab}$. Cette dernière égalité montre que les applications $\bar{a}$ et $\bar{b}$ commutent, et toute application $\bar{f}$ pour un mot f d'au moins trois lettres par commutation est équivalente à une application $\overline{f'}$ pour un mot f' ayant 2 lettres consécutives identiques, et par la première égalité, équivalente à une application $\overline{f''}$ pour un mot f'' strictement plus court. On a donc calculé toutes les applications du monoïde T : $T = \{\overline{1}, \bar{a}, \bar{b}, \overline{ab}\}$.

Question 4 : Soit $H = \{\nu \in T \mid \nu(1) \in F\}$ et soit $G = \{\tau \in T \mid \tau \circ \tau \circ \tau(1) \in F\}$. Que valent H et G ?

On lit sur la première ligne du tableau ci-dessus que l'application $\overline{ab}$ est la seule application de T qui envoie l'état de départ 1 sur un état d'acceptation, donc $H = \{\overline{ab}\}$.
Calculons le cube des applications de T : $\overline{1}^3 = \overline{1}$, $\bar{a}^3 = \bar{a}^2.\bar{a} = \bar{a}$, $\bar{b}^3 = \bar{b}^2.\bar{b} = \bar{b}$ et $\overline{ab}^3 = \overline{ab}.\overline{ab}^2 = \overline{ba}.\overline{ab}^2 = \bar{b}.\bar{a}.\bar{a}.\bar{b}.\overline{ab} = \bar{b}.\bar{b}.\overline{ab} = \overline{ab}$.
Parmi les applications $\overline{1}^3$, $\bar{a}^3$, $\bar{b}^3$ et $\overline{ab}^3$, seule cette dernière envoie l'état de départ 1 sur un état d'acceptation, donc $G = \{\overline{ab}^3\} = \{\overline{ab}\}$.

On rappelle que si $\varphi : A^* \to T$ est l'homorphisme défini par $\varphi(f) = \bar{f}$, (où $\overline{u.v} = \bar{u}.\bar{v} = \bar{v} \circ \bar{u}$) alors $L = \varphi^{-1}(H)$.

Question 5 : Vérifier que $\varphi^{-1}(G) = \{w \in A^* \mid w^3 \in L\}$, et en déduire que le langage $\mathcal{T}(L) = \{w \in A^* \mid w^3 \in L\}$ est reconnaissable.

$\varphi^{-1}(G) = \{w \in A^* \mid \varphi(w) \in G\} = \{w \in A^* \mid \varphi(w) = \overline{ab}\} = \{w \in A^* \mid \varphi(w^3) = \overline{ab}^3\} = \{w \in A^* \mid \varphi(w^3) = \overline{ab}\} = \{w \in A^* \mid w^3 \in L\}$.
$\{w \in A^* \mid w^3 \in L\}$ s'écrit donc comme l'image par un morphisme inverse d'une partie G d'un monoïde fini T. C'est donc par définition un langage reconnaissable.