Solutions - TD Feuille 2 - Automates finis

4TIN604U Modèles de la Programmation et du Calcul
Licence 3 - Université de Bordeaux

Solution de l'exercice 1 :

L'automate $\mathcal{A}_1$ a pour définition :

$$\mathcal{A}_1 = \left(Q = \{1,2,3,4\}, \quad q_i = 1, \quad F = \{3,4\}, \quad \delta : \left\{ \begin{array}{l} (1,a) \to 2 \\ (1,b) \to 4 \\ (2,a) \to 4 \\ (2,b) \to 3 \\ (3,a) \to 3 \\ (3,b) \to b \end{array} \right. \right)$$

L'automate $\mathcal{A}_2$ a pour définition :

$$\mathcal{A}_2 = \left(Q = \{1,2,3,4\}, \quad q_i = 1, \quad F = \{1,2\}, \quad \delta : \left\{ \begin{array}{l} (1,a) \to 1 \\ (1,b) \to 4 \\ (2,a) \to 1 \\ (2,a) \to 3 \\ (3,b) \to 1 \\ (4,b) \to 2 \end{array} \right. \right)$$

Solution de l'exercice 2 :

Les automates $\mathcal{A}_3$ et $\mathcal{A}_4$ sont dessinés ci-dessous :

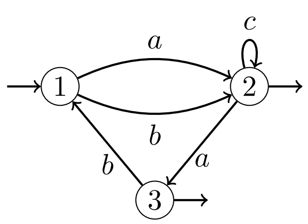


FIGURE 1 – Automate $\mathcal{A}_3$

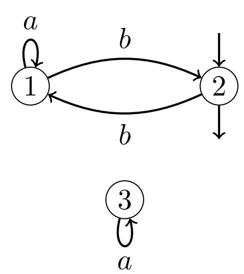


FIGURE 2 – Automate $\mathcal{A}_4$

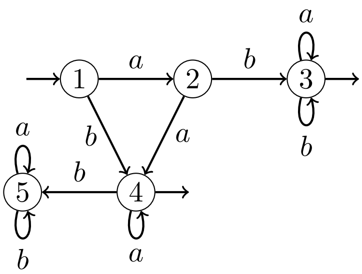
Solution de l'exercice 3 :

1. (a) Automate $\mathcal{A}_1$:

   i. Il n'est pas complet, pour le compléter on ajoute un état 5 (état "puits", non terminal !) et les transitions :

   $$\begin{aligned}(4, b) &\to 5\\ (5, a) &\to 5\\ (5, b) &\to 5\end{aligned}$$

   On obtient ainsi l'automate :

   ii. Il est déterministe car il a un seul état initial et pour tout état $q \in Q$, pour toute lettre $a \in A$, il y a au plus une transition étiquetée par a ayant origine en q.

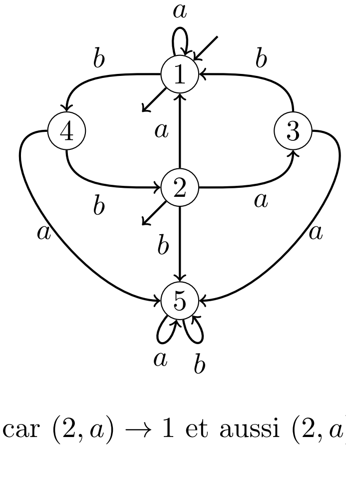
   iii. Plus court mot accepté : b, ayant pour calcul acceptant $1 \xrightarrow{b} 4$.

   (b) Automate $\mathcal{A}_2$:

   i. Il n'est pas complet, pour le compléter on ajoute un état 5 (état "puits", non terminal !) et les transitions :

   $$\begin{aligned}(4, a) &\to 5\\ (2, b) &\to 5\\ (3, a) &\to 5\\ (5, a) &\to 5\\ (5, b) &\to 5\end{aligned}$$

   On obtient ainsi l'automate :

   ii. Il n'est pas déterministe car $(2, a) \to 1$ et aussi $(2, a) \to 3$.

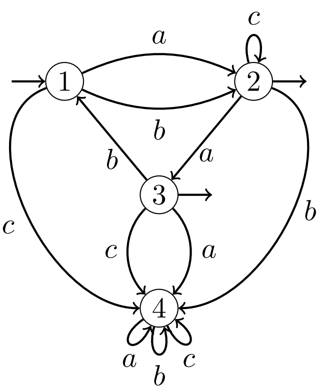
iii. Plus court mot accepté : ϵ, ayant pour calcul acceptant : 1.

(c) Automate $\mathcal{A}_3$:

i. Il n'est pas complet, pour le compléter on ajoute un état 4 (état "puits", non terminal !) et les transitions :

$$\begin{array}{c} (1,c) \to 4 \\ (2,b) \to 4 \\ (3,a) \to 4 \\ (3,c) \to 4 \\ (4,a) \to 4 \\ (4,b) \to 4 \\ (4,c) \to 4 \end{array}$$

On obtient ainsi l'automate :



ii. Il est déterministe (voir définition ci-dessus, pour $\mathcal{A}_1$).

iii. Plus courts mots acceptés : a, ayant pour calcul accaptant $1 \xrightarrow{a} 2$, et b ayant pour calcul acceptant $1 \xrightarrow{b} 2$.

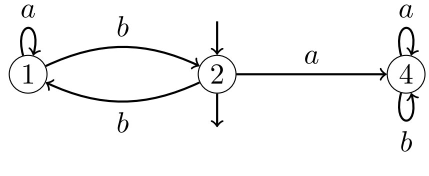
(d) Automate $\mathcal{A}_4$: on peut remarquer qu'il possède un état qui n'est pas accessible depuis l'état initial, cet état 3 ne donne aucune contribution. L'automate obtenu en l'éliminant (ainsi qu'en éliminant la transition $(3,a) \to 3$), reconnaît le même langage.

i. Il est déterministe (voir définition ci-dessus, pour $\mathcal{A}_1$).

ii. Il n'est pas complet, pour le compléter on ajoute un état 4 (état "puits", non terminal !) et les transitions :

$$\begin{array}{c} (2,a) \to 4 \\ (4,a) \to 4 \\ (4,b) \to 4 \end{array}$$

On obtient ainsi l'automate :



iii. Plus court mot accepté : ϵ.

(e) Automate $\mathcal{A}_5$:

i. Il est complet.

ii. Il est déterministe.

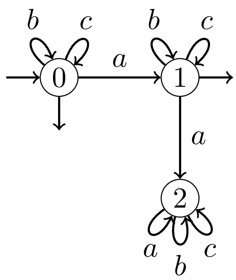
iii. Plus court mot accepté : ϵ.

(f) Automate $\mathcal{A}_6$:

i. Il n'est pas complet, pour le compléter on ajoute un état 2 (état "puits", non terminal !) et les transitions :

$$\begin{array}{l}(1,a)\to 2\\(2,a)\to 2\\(2,b)\to 2\\(2,c)\to 2\end{array}$$

On obtient ainsi l'automate :

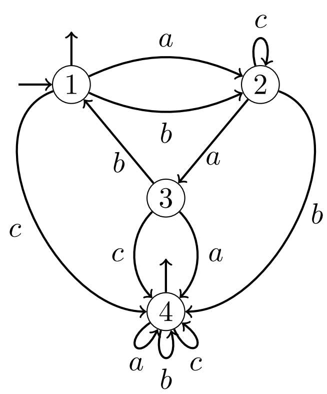


ii. Il est déterministe.

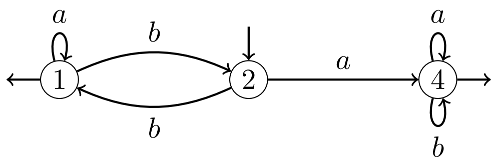
iii. Plus court mot accepté : ϵ.

2. L'automate $\mathcal{A}_2$ est tel que, pour le mot *bbab*, il y a deux chemins :
   - un chemin $\to 1 \xrightarrow{b} 4 \xrightarrow{b} 2 \xrightarrow{a} 1 \xrightarrow{b} 4 \to$ qui se termine dans un état non terminal
   - un chemin $\to 1 \xrightarrow{b} 4 \xrightarrow{b} 2 \xrightarrow{a} 3 \xrightarrow{b} 1 \to$ qui se termine dans un état terminal.
3. Pour chacun de ces automates, on a déjà construit un automate **déterministe complet** qui accepte le même langage. Il suffit donc d'inverser les états terminaux avec les états non-terminaux dans ces automates déterministes complets.

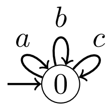
(a) Automate qui accepte le complémentaire du langage de l'automate $\mathcal{A}_3$:



(b) Automate qui accepte le complémentaire du langage de l'automate $\mathcal{A}_4$:

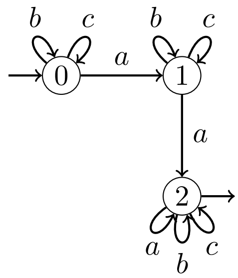


(c) Automate qui accepte le complémentaire du langage de l'automate $\mathcal{A}_5$:



Cet automate n'a pas d'état terminal : il accepte le langage vide : $\emptyset$

(d) Automate qui accepte le complémentaire du langage de l'automate $\mathcal{A}_6$:



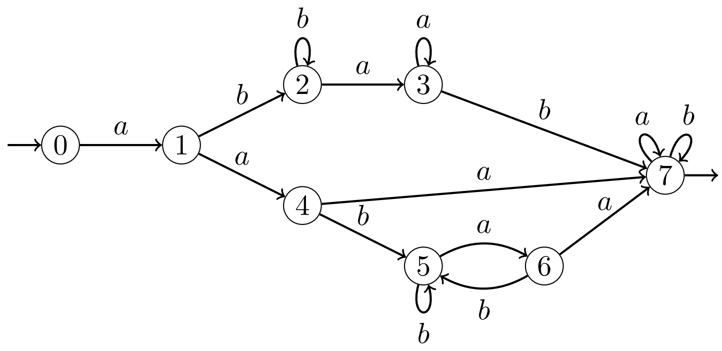
Solution de l'exercice 4 :

1. Vrai, pour tout $i \geq 0$ et pour tout $w \in L$ on a $w^i \in L^i$. Donc $\{w^i \mid i \geq 0 \text{ et } w \in L\} \subset \bigcup_{i \geq 0} L^i$. Or la définition de L^* est $L^* = \bigcup_{i \geq 0} L^i$.
2. Faux. Contrexemple : si $L = \{a, b\}$ on a $ab \in L^*$ mais il n'existe pas d'entier $i \geq 0$ tel que $ab = a^i$ ou $ab = b^i$.
3. Vrai. On prouve la double inclusion. Commençons par supposer que $u \in \{w^i \mid i \geq 0 \text{ et } w \in L^*\}$, on doit montrer que $u \in L^*$. On sait qu'il existe $i \geq 0$ et $w \in L^*$ tels que $u = w^i$. Puisque $w \in L^*$, il existe des mots $w_1, \ldots, w_k \in L$ tels que $w = w_1 \ldots w_k$. Mais alors $w^i = \underbrace{w_1 \ldots w_k}_{1} \underbrace{w_1 \ldots w_k}_{2} \ \ldots \ \underbrace{w_1 \ldots w_k}_{i} \in L^{ki}$, donc $w^i \in L^*$.

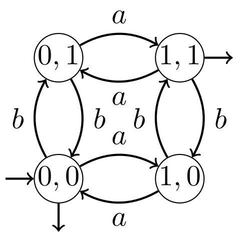
   Supposons maintenant que $u \in L^*$, on doit prouver que $u \in \{w^i \mid i \geq 0 \text{ et } w \in L^*\}$. Prenons $i = 1$ et $w = u \in L^*$, on a bien $u = w^i$ donc $u \in \{w^i \mid i \geq 0 \text{ et } w \in L^*\}$ ce qui termine la preuve.

Solution de l'exercice 5 :

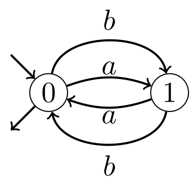
1. Expression régulière : $a(b(a+b)^*ab + a(a + (a+b)^*aa))(a+b)^*$
   Automate :



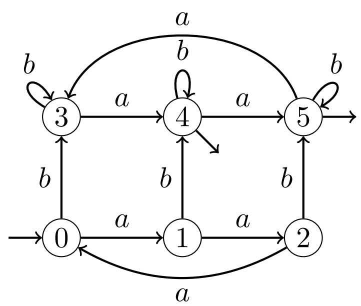
2. L'automate suivant prend en compte la parité des 'a' et des 'b' de chaque préfixe du mot à analyser :



Mais si les mots reconnus doivent avoir les nombres de 'a' et de 'b' de même parité, ils ont forcement une longueur paire, car aussi bien la somme de deux nombres pairs que de deux nombres impairs est un nombre pair. L'expression régulière est donc $((a+b)(a+b))^*$. L'automate suivant reconnait le même langage que l'automate à quatre états précédent :



3. Expression : $(aaa)^*(b^+(ab^* + ab^*ab^*) + ab^+(ab^* + \epsilon) + aab^+(ab^*ab^* + \epsilon))(ab^*ab^*ab^*)^*$
Automate :

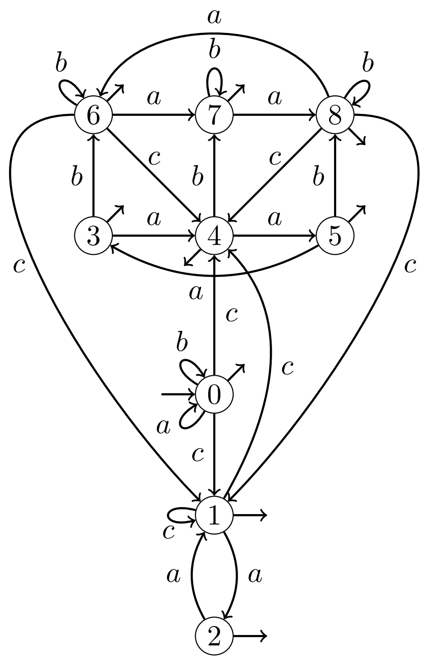


Solution de l'exercice 6 :

On commence par donner l'expression régulière. Pour simplifier, on réutilise le langage L_3 de l'exercice précédent :

$$(a + b)^*(c(L_3c + (aa)^*c)^*(a + b)^* + \epsilon)$$

On donne maintenant l'automate. Pour simplifier le schéma on donne une version non-déterministe :



Solution de l'exercice 7 :

1. On a $(ab)^* \sqcup\!\sqcup \{c\} = (ab)^*c(ab)^* + (ab)^*acb(ab)^*$.
2. Par hypothèse, on dispose d'un automate $\mathcal{A}_K = (A, Q, Q_0, F_K, \delta_K)$ qui accepte K et d'un automate $\mathcal{A}_L = (A, R, R_0, F_L, \delta_L)$ qui accepte L. En utilisant ces deux automates, on construit l'automate $\mathcal{A}_{K \sqcup\!\sqcup L} = (A, S, S_0, F, \delta)$ de la façon suivante :
   — $S = Q \times R$.
   — $S_0 = Q_0 \times R_0$.
   — $F = F_K \times F_L$.
   — $\delta = \begin{array}{c} \{(q,r) \xrightarrow{a} (q',r) \mid (q \xrightarrow{a} q') \in \delta_K \text{ et } r \in R\} \cup \\ \{(q,r) \xrightarrow{a} (q,r') \mid (r \xrightarrow{a} r') \in \delta_L \text{ et } q \in Q\} \end{array}$.
   On peut vérifier $\mathcal{A}_{K \sqcup\!\sqcup L}$ accepte bien le langage $K \sqcup\!\sqcup L$.

Solution de l'exercice 8 :

1. On a $C(abab) = \{abab, baba\}$, $C(a^*b^*) = a^*b^*a^* + b^*a^*b^*$ et $C(\{a^nb^n \mid n \geq 0\}) = \{a^ib^na^j \mid n = i + j\} \cup \{b^ia^nb^j \mid n = i + j\}$.
2. Par hypothèse, on dispose d'un automate $\mathcal{A}_L = (A, Q, Q_0, Q_f, \delta)$ qui accepte L. On veut construire un automate qui accepte $C(L)$.
   Pour tout $q \in Q$, on construit les deux automates suivants :
   — $\mathcal{A}_{q,0} = (A, Q, \{q\}, Q_f, \delta)$ dont on note le langage $L_{q,i}$.
   — $\mathcal{A}_{q,f} = (A, Q, Q_0, \{q\}, \delta)$ dont on note le langage $L_{q,f}$.
   On peut vérifier que $C(L) = \cup_{q \in Q} L_{q,0} \cdot L_{q,f}$. Puisque les langages réguliers sont clos par concaténation et union, on peut donc construire un automate qui accepte $C(L)$ à partir des automates $\mathcal{A}_{q,0}$ at $\mathcal{A}_{q,f}$.