Solutions - TD Feuille 1 - Expressions rationnelles et automates finis

18 septembre 2023

Solution de l'exercice 1 :

1. Mots de longueurs 0 : ϵ ;
   Mots de longueurs 1 : a ;
   Mots de longueurs 2 : aa, ba ;
   Mots de longueurs 3 : aaa, aba, baa ;
   Mots de longueurs 4 : $aaaa$, $aaba$, $abaa$, $baba$, $baaa$.
2. Mots de longueurs 0 : aucun ;
   Mots de longueurs 1 : aucun ;
   Mots de longueurs 2 : aa ;
   Mots de longueurs 3 : aucun ;
   Mots de longueurs 4 : $aaaa$, $abaa$.

L'expression $(a + ba)^*$ décrit seulement des mots qui ne contiennent pas le facteur bb, mais pas tous : par exemple, $b \notin L((a + ba)^*)$.

Solution de l'exercice 2 :

1. a^*b^*
2. $(b + ab^*a)^*$
3. $b^*(a(bb)^*)^*b^*$
4. $(a + b)^*ba$
5. $(b + ab)^*(a + \epsilon)$

Solution de l'exercice 3 :

Automate à gauche : $\mathcal{A}_1$, automate à droite $\mathcal{A}_2$.

Mots de longueur 0 reconnus par l'automate $\mathcal{A}_1$: aucun ;
Mots de longueur 1 reconnus par l'automate $\mathcal{A}_1$: b ;
Mots de longueur 2 reconnus par l'automate $\mathcal{A}_1$: ba, aa, ab ;
Mots de longueur 3 reconnus par l'automate $\mathcal{A}_1$: baa, aaa, aba, abb ;
Mots de longueur 4 reconnus par l'automate $\mathcal{A}_1$: $baaa$, $aaaa$, $abaa$, $abab$, $abba$, $abbb$.

Il est possible de répondre à cette question de manière systématique en utilisant les matrices. Pour cela, on représente l'automate (que l'on peut voir comme un graphe) par la matrice d'adjacence suivante :

$$M = \begin{pmatrix} 0 & a & 0 & b \\ 0 & 0 & b & a \\ 0 & 0 & a+b & 0 \\ 0 & 0 & 0 & a \end{pmatrix}.$$

Ainsi, le coefficient d'indice i, j de la matrice M^k correspond aux mots de longueur k reconnus par l'automate, si l'état initial était l'état i et l'état final, l'état j. Si l'on souahite obtenir les mots de longueur k reconnus par notre automate, il suffit d'évaluer $M^k_{1,4} + M^k_{1,3}$. Voici les matrices M^0, M^1, M^2, M^3 et M^4 :

$$M^0 = \begin{pmatrix} \epsilon & 0 & 0 & 0 \\ 0 & \epsilon & 0 & 0 \\ 0 & 0 & \epsilon & 0 \\ 0 & 0 & 0 & \epsilon \end{pmatrix} \quad M^1 = \begin{pmatrix} 0 & a & 0 & b \\ 0 & 0 & b & a \\ 0 & 0 & a+b & 0 \\ 0 & 0 & 0 & a \end{pmatrix} \quad M^2 = \begin{pmatrix} 0 & 0 & ab & aa+ba \\ 0 & 0 & b(a+b) & a^2 \\ 0 & 0 & (a+b)^2 & 0 \\ 0 & 0 & 0 & a^2 \end{pmatrix}$$

$$M^3 = \begin{pmatrix} 0 & 0 & ab(a+b) & a^3+ba^2 \\ 0 & 0 & b(a+b)^2 & a^3 \\ 0 & 0 & (a+b)^3 & 0 \\ 0 & 0 & 0 & a^3 \end{pmatrix} \qquad M^4 = \begin{pmatrix} 0 & 0 & ab(a+b)^2 & a^4+ba^3 \\ 0 & 0 & b(a+b)^3 & a^4 \\ 0 & 0 & (a+b)^4 & 0 \\ 0 & 0 & 0 & a^4 \end{pmatrix}$$

Mots de longeurs 0 : $M^0_{1,2} + M^0_{1,4} = 0$;
Mots de longueurs 1 : $M^1_{1,3} + M^1_{1,4} = b$;
Mots de longueurs 2 : $M^2_{1,3} + M^2_{1,4} = ab + aa + ba$;
Mots de longueurs 3 : $M^3_{1,3} + M^3_{1,4} = aba + abb + aaa + baa$;
Mots de longueurs 4 : $M^4_{1,3} + M^4_{1,4} = abaa + abab + abba + abbb + aaaa + baaa$;

Pour l'automate $\mathcal{A}_2$, on a les résultats suivants :

$$M^0 = \begin{pmatrix} \epsilon & 0 & 0 & 0 \\ 0 & \epsilon & 0 & 0 \\ 0 & 0 & \epsilon & 0 \\ 0 & 0 & 0 & \epsilon \end{pmatrix} \quad M = \begin{pmatrix} a & 0 & 0 & b \\ a & 0 & a & 0 \\ b & 0 & 0 & 0 \\ 0 & b & 0 & 0 \end{pmatrix} \quad M^2 = \begin{pmatrix} a^2 & b^2 & 0 & ab \\ a^2+ab & 0 & 0 & ab \\ ba & 0 & 0 & b^2 \\ ba & 0 & ba & 0 \end{pmatrix}$$

$$M^3 = \begin{pmatrix} a^3+b^2a & ab^2 & b^2a & a^2b \\ a^3+aba & ab^2 & 0 & a^2b+ab^2 \\ ba^2 & b^3 & 0 & bab \\ b(a^2+ab) & 0 & 0 & bab \end{pmatrix} \quad M^4 = \begin{pmatrix} a(a^3+b^2a) & b^2(a^2+ab)+a^2b^2 & \dots \\ & & \\ & & \\ & & \end{pmatrix}$$

Mots de longueur 0 : $M^0_{1,1} + M^0_{1,2} = \epsilon$;
Mots de longueur 1 : $M^1_{1,1} + M^1_{1,2} = a$;
Mots de longueur 2 : $M^2_{1,1} + M^2_{1,2} = aa + bb$;
Mots de longueur 3 : $M^3_{1,1} + M^3_{1,2} = aaa + bba + abb$;
Mots de longueur 4 : $M^4_{1,1} + M^4_{1,2} = aaaa + abba + aabb + bbaa + bbab$;

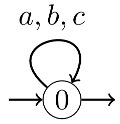
Solution de l'exercice 4 :

1. Vrai.
2. Faux. Prenons $L = \{\epsilon\}$, $LL^* = \{\epsilon\} \neq \emptyset = L^* \setminus \{\epsilon\}$.
3. Vrai.
4. Vrai.
   Montrons l'égalité :

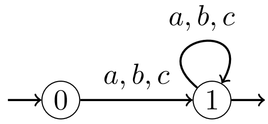
$$\begin{gathered} w \in (KL)^*K \\ \Leftrightarrow \\ w = (k_1 l_1 \cdots k_n l_n) k_{n+1} \text{ avec } k_1, \ldots, k_{n+1} \in K \text{ et } l_1, \ldots, l_n \in L \\ \Leftrightarrow \\ w = k_1 (l_1 k_2 \cdots l_n k_{n+1}) \\ \Leftrightarrow \\ w \in K(LK)^* \end{gathered}$$

Solution de l'exercice 5 :

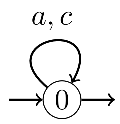
1. Expression régulière : $(a+b+c)^*$.
   Automate :



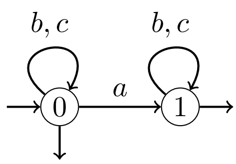
2. Expression régulière : $(a+b+c)^+ = (a+b+c)A^*$.
   Automate :



3. Expression régulière : $(a+c)^*$.
   Automate :



4. Expression régulière : $(b+c)^*(a+\epsilon)(b+c)^*$.
   Automate :



   On prouve maintenant que cet automate reconnaît bien le bon langage. On note :
   — L_0 le langage des mots ne contenant pas de a.

— K_0 le langage des mots pour lequels le calcul finit dans l'état 0.
— L_1 le langage des mots contenant exactement un a.
— K_1 le langage des mots pour lequels le calcul finit dans l'état 1.

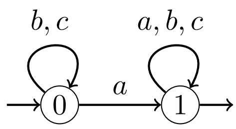
Observons que le langage de la question est $L_1 \cup L_0$ et le langage reconnu par l'automate $K_1 \cup K_0$. On va montrer $L_0 = K_0$ et $L_1 = K_1$. Considérons la propriété suivante, $\mathcal{P}_n$, paramétrée par un entier n :

$$\mathcal{P}_n : \text{Pour tout mot } w \text{ de taille } n : \begin{array}{l} w \in L_0 \Leftrightarrow w \in K_0 \\ w \in L_1 \Leftrightarrow w \in K_1 \end{array}$$

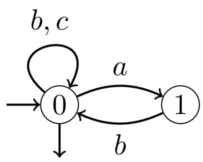
Il est clair que si $\mathcal{P}_n$ est vraie pour tout entier n, on a bien $K_0 = L_0$ et $K_1 = L_1$. Il nous reste à prouver $\mathcal{P}_n$ pour tout n. On procède par induction sur n.

— Si $n = 0$, on a $w = \epsilon$. Par définition, on a $w \in L_0$, $w \in K_0$, $w \notin L_1$ et $w \notin K_1$. Donc $\mathcal{P}_0$ est vérifiée.
— Soit $n > 0$, on suppose $\mathcal{P}_{n-1}$ et on prouve $\mathcal{P}_n$. Soit w un mot de taille n, puisque $n > 0$, on a $w = w'x$ où w' est un mot de taille $n-1 \geq 0$ et x une lettre dans $\{a, b, c\}$. Supposons que $w \in K_0$, par définition des transitions de l'automate on a $w' \in K_0$ et $x \in \{b, c\}$. Par induction on en déduit que $w' \in L_0$ donc $w \in L_0(b + c) \subseteq L_0$. Réciproquement, si $w \in L_0$, on a $w' \in L_0$ donc par induction $w' \in K_0$, par définition des transitions de l'automate on obtient $w \in K_0$.
  Maintenant, supposons que $w \in K_1$, par définition de l'automate, il y a deux cas possibles : soit $w' \in K_0$ et $x = a$, soit $w' \in K_1$ et $x \in \{b, c\}$. Dans le premier cas on conclut en utilisant l'induction que $w' \in L_0$ et $w \in L_0 a \subseteq L_1$. Dans le second cas on a par induction $w' \in L_1$ et $w \in L_1(b + c) \subseteq L_1$. Réciproquement, si $w \in L_1$, soit $x = a$ auquel cas $w' \in L_0$ et donc $w' \in K_0$ par induction ce qui implique que $w \in K_1$ par définition des transitions. Sinon $x \in \{b, c\}$, dans ce cas $w' \in L_1$ et $w \in L_1$ par définition des transitions.

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



6. Expression régulière : $(b + c)^*(ab(b + c)^*)^*$ ou $(b + c + ab)^*$.
   Automate :



On prouve maintenant que cet automate reconnaît bien le bon langage. On note :
— L_0 le langage des mots soit ne contenant pas de a soit tels que chaque a est immédiatement suivi d'un b
— K_0 le langage des mots pour lequels le calcul finit dans l'état 0.
— L_1 le langage des mots soit contenant un unique a comme dernière lettre, soit tels que chaque a est immédiatement suivi d'un b et la dernière lettre est un a.

— K_1 le langage des mots pour lequels le calcul finit dans l'état 1.
Observons que le langage de la question est L_0 et le langage reconnu par l'automate K_0.

On va montrer $L_0 = K_0$ et $L_1 = K_1$. Considérons la propriété suivante, $\mathcal{P}_n$, paramétrée par un entier n :

$$\mathcal{P}_n : \text{Pour tout mot } w \text{ de taille } n : \begin{array}{l} w \in L_0 \Leftrightarrow w \in K_0 \\ w \in L_1 \Leftrightarrow w \in K_1 \end{array}$$

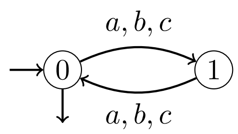
Il est clair que si $\mathcal{P}_n$ est vraie pour tout entier n, on a bien $K_0 = L_0$ et $K_1 = L_1$. Il nous reste à prouver $\mathcal{P}_n$ pour tout n. On procède par induction sur n.

— Si $n = 0$, on a $w = \epsilon$. Par définition, on a $w \in L_0$, $w \in K_0$, $w \notin L_1$ et $w \notin K_1$. Donc $\mathcal{P}_0$ est vérifiée.
— Soit $n > 0$, on suppose $\mathcal{P}_{n-1}$ vérifiée et on prouve $\mathcal{P}_n$.
  Soit w un mot de taille n, puisque $n > 0$, on a $w = w'x$ où w' est un mot de taille $n - 1 \geq 0$ et x une lettre dans $\{a, b, c\}$.

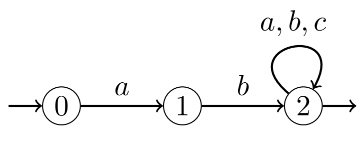
  Supposons que $w \in K_0$, par définition des transitions de l'automate on a soit $w' \in K_0$ et $x \in \{b, c\}$, soit $w' \in K_1$ et $x = b$. Dans les deux cas, $x \neq a$. Dans le cas $w' \in K_0$ par induction on déduit que $w' \in L_0$ et donc $w \in L_0(b+c) \subseteq L_0$. Dans le cas $w' \in K_1$ par induction on déduit que $w' \in L_1$ et donc $w \in L_1 b \subseteq L_0$. On vient de montrer que $K_0 \subseteq L_0$.
  Réciproquement, si $w \in L_0$, on a soit $w' \in L_0$, soit $w' \in L_1$. Si $w' \in L_0$, par induction $w' \in K_0$ et par définition des transitions de l'automate $x \in \{b, c\}$, donc $w \in K_0$. Si $w' \in L_1$, par induction $w' \in K_1$ et par définition des transitions de l'automate $x = b$, donc $w \in K_0$. On vient de montrer que $L_0 \subseteq K_0$, et puisque on a montré que $K_0 \subseteq L_0$, on a $w \in L_0 \Leftrightarrow w \in K_0$.

  Maintenant, supposons que $w \in K_1$, par définition de l'automate, on a $w' \in K_0$ et $x = a$. Par induction $w' \in L_0$ et $w \in L_0 a \subseteq L_1$. On a donc $K_1 \subseteq L_1$.
  Réciproquement, si $w \in L_1$, $x = a$ auquel cas $w' \in L_0$ et donc $w' \in K_0$ par induction ce qui implique que $w \in K_1$ par définition des transitions. On vient de montrer que $L_1 \subseteq K_1$, et puisque on a montré que $K_1 \subseteq L_1$, on a $w \in L_1 \Leftrightarrow w \in K_1$.

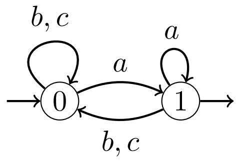
7. Expression régulière : $((a + b + c)(a + b + c))^*$
   Automate :



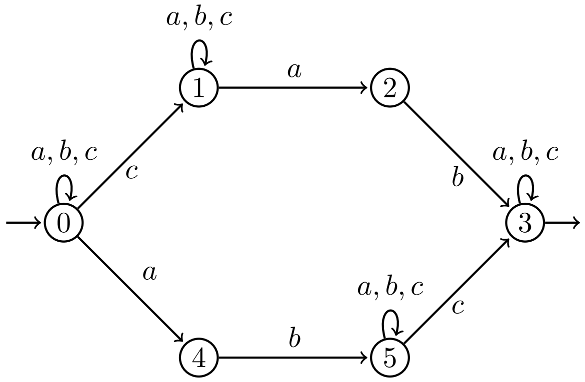
8. Expression régulière : $ab(a + b + c)^*$.
   Automate :



9. Expression régulière : $(a+b+c)^*a$.
   Automate :



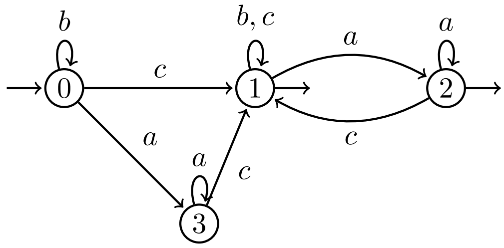
10. Expression régulière : $A^*cA^*abA^* + A^*abA^*cA^*$.
    Automate :



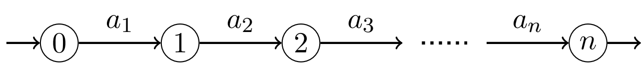
11. Expression régulière $b^*a^*c\,K$, où K est l'expression décrivant les mots qui ne contiennent pas le facteur ab :

$$K = (b+c)^*((a^+c)(b+c)^*)^*a^*$$

    Automate :



12. Soit $w = a_1 \cdots a_n$, $a_i \in A$.
    Expression régulière : $a_1 \cdot a_2 \cdots a_n$.
    Automate :



Solution de l'exercice 6 :

On fixe $x = x_1 \cdots x_n$ où $x_1, \ldots, x_n$ sont des lettres dans A. On va construire un automate déterministe à $n+1$ états qui reconnaît $L = A^*x$.

On commence par donner l'ensemble d'états de l'automate, puis on va construire petit à petit les transitions. On fixe $Q = \{q_0, \ldots, q_n\}$, $I = \{q_0\}$ et $F = \{q_n\}$. De plus pour tout entier $i \in \{1, \ldots, n\}$ on ajoute une transition $(q_{i-1}, x_i) \to q_i$. Pour l'instant notre automate est bien deterministe et reconnaît le langage $\{x\}$. On peut le dessiner de la façon suivante :

$$\rightarrow q_0 \xrightarrow{x_1} q_1 \xrightarrow{x_2} q_2 \xrightarrow{x_3} \cdots\cdots \xrightarrow{x_n} q_n \rightarrow$$

On va maintenant ajouter des transitions pour que l'automate reconnaisse le langage $L = A^*x$. Considérons les langages $K_0, \ldots, K_n$ définis de la façon suivante : $K_i = A^*x_1 \cdots x_i$ pour tout entier $i \in \{0, \ldots, n\}$. Notons que K_n est le langage L. Enfin on fixe $L_0, \ldots, L_n$ les langages définis comme suit : $L_n = K_n = L$ et pour tout $i \in \{0, n-1\}$, $L_i = K_i \setminus (K_{i+1} \cup \cdots \cup K_n)$.

Commençons par faire deux observations : 1) pour tout $i \neq j$, $L_i \cap L_j = \emptyset$, 2) $L_0 \cup \cdots \cup L_n = A^*$. Il en découle que tout mot w appartient à un et un seul langage L_i : par définition il s'agit du L_i pour le plus grand i tel que $x_1 \cdots x_i$ est un suffixe de w.

Notre but est de compléter les transitions de notre automate pour que le langage reconnu dans l'état q_i soit L_i pour tout $i \in \{0, \ldots, n\}$. On commence par donner la liste des transitions à ajouter puis on prouve que l'automate construit ainsi vérifie bien cette propriété.

Pour tout $i, j \in \{0, \ldots, n\}$ et $a \in A$ tel que $a \neq x_{i+1}$, on ajoute la transition $(q_i, a) \to q_j$ à l'ensemble de transitions de notre automate si et seulement si le mot $x_1 \cdots x_i a$ appartient au langage L_j. Observons qu'il existe toujours un unique langage L_j tel que $x_1 \cdots x_i a \in L_j$. Donc on n'ajoute qu'une unique transition partant de q_i et étiquetée par a pour chaque couple (q_i, a).

La construction de l'automate $\mathcal{A}$ est maintenant finie. Par construction, $\mathcal{A}$ est bien déterministe, il est même complet (c'est à dire que pour tout mot w le calcul de $\mathcal{A}$ sur w arrive dans un et un seul état). Il nous reste à montrer que $\mathcal{A}$ accepte bien le langage L.

Soit w un mot, puisque $\mathcal{A}$ est déterministe et complet, il existe un unique état q_i tel que le calcul sur w arrive en q_i. On prouve par induction sur la longueur de w que $w \in L_i$.

Si $w = \epsilon$, alors $q_i = q_0$ et $\epsilon \in L_0$. On suppose le résultat vrai pour les mots de longueur k. Soit w de longueur $k+1$. On a $w = w'a$ avec w' un mot de longueur k et $a \in A$. Soit $q_{i'}$ l'état dans lequel arrive le calcul sur w', par définition des transitions on en déduit que $x_1 \cdots x_{i'} a \in L_i \subseteq K_i$. Par hypothèse d'induction, on a $w' \in L_{i'}$, donc w' a pour suffixe $x_1 \cdots x_{i'}$ et w a pour suffixe $x_1 \cdots x_{i'} a$: $w \in K_i$. On veut montrer que $w \in L_i$, pour cela on doit montrer que $w \notin K_j$ pour $j > i$. Procédons par l'absurde et supposons qu'il existe $j > i$ tel que $w \in K_j$, donc $w' \in K_{j-1}$. Puisque $w' \in L_{i'}$, on a $i' \geq j-1$, donc $x_1 \cdots x_{i'} \in K_{j-1}$ et $x_1 \cdots x_{i'} a \in K_j$ ce qui est impossible étant donné que $x_1 \cdots x_{i'} a \in L_i$ et par définition, puisque $j > i$, $L_i \cap K_j = \emptyset$. On en déduit donc que $w \in L_i$.

En appliquant le résultat à l'etat final q_n on obtient que les mots acceptés par l'automate sont exactement les mots de L.