Devoir maison n°4

Corrigé

⋆ ⋆ ⋆

Minimisation d'automate

1 Algorithme de Brzozowski

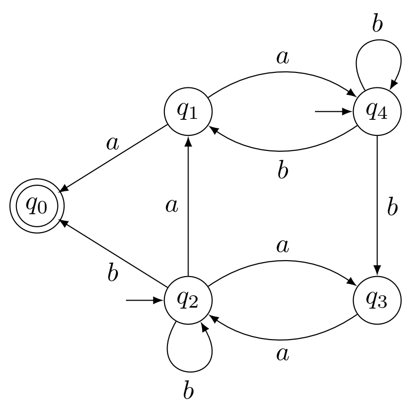
1.1 Langage miroir

Question 1 Montrons que $L(A_0)$ est l'ensemble des mots non vides dont le plus long suffixe de la forme a^k est de taille paire, c'est-à-dire l'interprétation de l'expression régulière $(\Sigma^* b \mid aa)(aa)^*$.

- si $u = \varepsilon$, alors $\delta^*(q_0, u) = q_0 \notin F$;
- si $|u| = 1$, alors $u = a$ ou $u = b$. On a $\delta^*(q_0, a) = q_1 \notin F$ et $\delta^*(q_0, b) = q_2 \in F$;
- si $|u| = n \geqslant 2$, posons $u = a_1 \dots a_n$. L'automate étant standard et complet, on remarque que pour $v \neq \varepsilon$, $\delta^*(q_0, v) \in \{q_1, q_2, q_3, q_4\}$. Distinguons selon les cas :
  - ⋆ si $a_n = b$, en remarquant que $\delta(q_1, b) = \delta(q_3, b) = \delta(q_4, b) = q_4 \in F$ et $\delta(q_2, b) = q_2 \in F$, on en déduit que $\delta^*(q_0, u) \in F$;
  - ⋆ sinon, $u = va^k$, avec v terminant par b ou $v = \varepsilon$. En remarquant que , distinguons :
    - — si k est pair, en remarquant que $\delta^*(q_0, aa) = \delta^*(q_2, aa) = \delta^*(q_4, aa) = q_2$, on en déduit par une récurrence rapide que $\delta^*(q_0, u) = q_2 \in F$;
    - — si k est impair, comme $\delta^*(q_0, v) \in \{q_0, q_2, q_4\}$ (par le raisonnement précédent), on a $\delta^*(q_0, va) \in \{q_1, q_3\}$. En remarquant que $\delta^*(q_1, aa) = \delta^*(q_3, aa) = q_3$, on en déduit que $\delta^*(q_0, u) = q_3$ sinon. Dans les deux cas, l'état est non final.

Les mots acceptés par l'automate sont bien ceux de la forme voulue.

Question 2 On inverse les transitions et les états initiaux et finaux. On obtient :



Question 3 On a bien $L(\overline{A}) = \overline{L(A)}$. En effet, soit $u \in \Sigma^*$, $u = a_1 \dots a_n$. Alors :

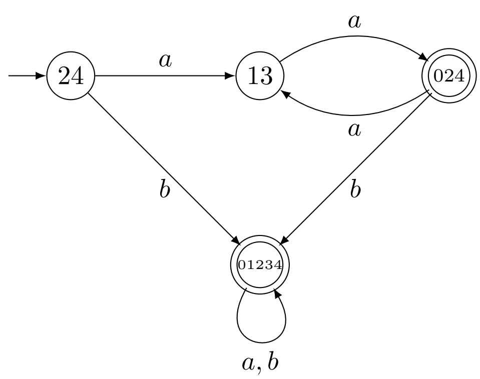
$$\begin{aligned}
u \in \overline{L(A)} &\Leftrightarrow \overline{u} \in L(A) \\
&\Leftrightarrow \text{il existe une suite de transitions dans } A \ : \ q_0 \xrightarrow{a_n} q_1 \xrightarrow{a_{n-1}} \dots \xrightarrow{a_1} q_n \in F \\
&\Leftrightarrow \text{il existe une suite de transitions dans } \overline{A} \ : \ q_n \in F \xrightarrow{a_1} q_{n-1} \xrightarrow{a_2} \dots \xrightarrow{a_n} q_0 \\
&\Leftrightarrow u \in L(\overline{A})
\end{aligned}$$

1.2 Déterminisation

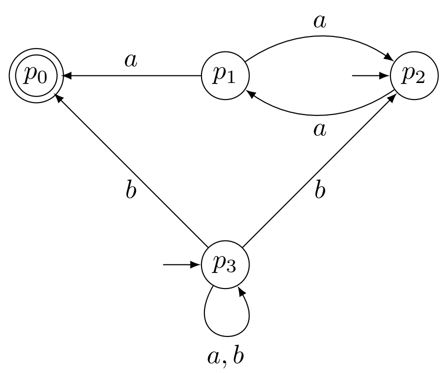
Question 4 On obtient, en indiquant uniquement les indices des états dans une partie :

	a	b
$\rightarrow$ 24	13	01234
13	024	$-$
01234 $\rightarrow$	01234	01234
024 $\rightarrow$	13	01234

L'automate A_1 est donc :



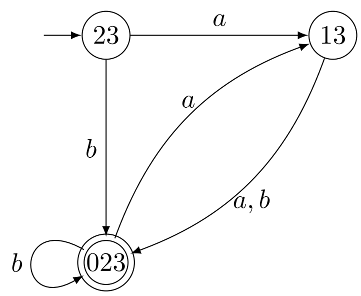
Question 5 On obtient, en renommant les états :



Question 6 On obtient :

	a	b
$\rightarrow$ 23	13	023
13	023	023
023 $\rightarrow$	13	023

L'automate A_2 est donc :



1.3 Algorithme de Brzozowski

Question 7 Pour un automate non déterministe C, on a, par principe de l'algorithme des parties, $L(C) = L(\det(C))$. On en déduit :

$$\begin{array}{rcl} L(B) & = & L(\det(\overline{\det(\overline{A})})) \\ & = & L(\overline{\det(\overline{A})}) \\ & = & \overline{L(\det(\overline{A}))} \\ & = & \overline{L(\overline{A})} \\ & = & \overline{\overline{L(A)}} \\ & = & L(A) \end{array}$$

Question 8 On commence par remarquer qu'avec les hypothèses, on peut écrire $F = \{f\}$.

Par principe de l'algorithme des parties, on a $q \in \delta^*(I, u)$ si et seulement si $q \in \Delta^*(I, u)$. Or, comme on a supposé que tous les états de A sont accessibles, cela est équivalent avec le fait que tous les états de $\overline{A}$ sont co-accessibles. On en déduit qu'il existe $v \in \Sigma^*$ tel que $f \in \Delta^*(q, v)$. Cela implique que $f \in \Delta^*(I, uv)$, soit que $uv \in L(\overline{A})$. On a bien $u^{-1}L(\overline{A}) \neq \emptyset$.

Question 9 On a $u^{-1}L(\overline{A}) = v^{-1}L(\overline{A})$ si et seulement si pour $w \in \Sigma^*$, $uw \in L(\overline{A}) \Leftrightarrow vw \in L(\overline{A})$.

Dès lors, soit $q \in \delta^*(I, u)$ et $w \in u^{-1}L(\overline{A})$, tel que $f \in \Delta^*(q, w)$ (comme à la question précédente), alors en notant δ_A la fonction de transition de A, on a $q = \delta_A(f, \overline{w})$. Comme $vw \in L(\overline{A})$ (car $uw \in L(\overline{A})$), alors $\overline{wv} \in L(A)$, on en déduit que $\delta_A^*(q, \overline{v}) \in I$ (les états initiaux de $\overline{A}$ sont les états finaux de A). Cela est bien équivalent avec le fait que $q \in \Delta^*(I, v) = \delta^*(I, v)$.

Question 10 Soit $B = (Q_B, \Sigma, \delta_B, q_B, F_B)$ un automate fini déterministe complet reconnaissant $L(\overline{A})$.

Supposons par l'absurde que $|Q_B| < |X|$. Notons, pour $x \in X$, $u_x \in \Sigma^*$ un mot tel que $\delta^*(I, u_x) = x$ (existe car tous les états de $\det(\overline{A})$ sont accessibles.

Par le principe des tiroirs, B étant complet, il existe $x, y \in X^2, x \neq y$ tels que $\delta_B^*(q_B, u_x) = \delta_B^*(q_B, u_y) = q$. Dès lors, pour $w \in \Sigma^*$, $u_x w \in L(B) \Leftrightarrow u_y w \in L(B)$ (car $\delta_B^*(q_B, u_x w) \in F_B \Leftrightarrow \delta_B^*(q, w) \in F_B \Leftrightarrow \delta_B^*(q_B, u_y w) \in F_B$), donc $u_x^{-1}L(\overline{A}) = u_y^{-1}L(\overline{A})$.

Par la question précédente, on en déduit que $x = \delta^*(I, u_x) = \delta^*(I, u_y) = y$, ce qui est absurde, car $x \neq y$. On en déduit que $|Q_B| \geqslant |X|$.

Question 11 Soit A un automate fini déterministe. L'automate $\det(\overline{A})$ est un automate fini déterministe dont tous les états sont accessibles. Par les questions précédentes, on en déduit que tous les automates finis déterministes complets reconnaissant $L(\overline{\det(\overline{A})}) = L(A)$ possèdent au moins autant d'états que $\det(\overline{\det(\overline{A})})$. L'algorithme de Brzozowski permet donc bien de construire un automate B reconnaissant $L(A)$ avec un nombre minimal d'états.

Par ailleurs, B est bien complet. En effet, s'il ne l'était pas, il possèderait un blocage, ce qui signifie que l'état $\emptyset$ est accessible et aurait dû être rajouté à l'automate.

Question 12 Soit A un automate déterministe à n états. La construction de $\overline{A}$ se fait en complexité $\mathcal{O}(n+\text{ nombre de transitions}) = \mathcal{O}(n+n|\Sigma|) = \mathcal{O}(n)$ pour un alphabet de taille fixe. C'est la complexité du calcul d'un graphe transposé, en nombre d'arêtes + nombre de sommets.

La déterminisation de $\overline{A}$ se fait en complexité $\mathcal{O}(2^n \times n)$ (pour chaque partie, on doit calculer l'image par la fonction de transition non déterministe). L'automate ainsi obtenu possède au plus 2^n états.

Le calcul de $\overline{\det(\overline{A})}$ se fait en temps $\mathcal{O}(2^n)$ de manière similaire à $\overline{A}$ (l'automate a au plus 2^n états et $|\Sigma|2^n$ transitions).

La déterminisation de $\overline{\det(\overline{A})}$ se fait en complexité $\mathcal{O}(2^{2^n})$ en première approche, mais on sait que l'automate obtenu est minimal. Il possède donc moins de n états (qui était le nombre d'états de A). On obtient alors une complexité en $\mathcal{O}(n \times 2^n)$ (n parties dont on doit calculer les images, avec de l'ordre d'au plus 2^n transitions sortantes par état).

La complexité totale est donc en $\mathcal{O}(n2^n)$. Ce n'est pas utilisable dès lors que l'automate possède plus que ~ 30 états.

2 Algorithme de Hopcroft

2.1 Équivalence de Nerode

Question 13 La paire $\{q_1, q_4\}$ est séparée par le mot vide (car q_1 n'est pas final mais q_4 l'est). La paire $\{q_0, q_1\}$ est séparée par a. La paire $\{q_2, q_4\}$ est une paire équivalente : soit $u \in \Sigma^*$, on distingue :

- si $u = \varepsilon$, alors $\delta^*(q_2, u) = q_2$ et $\delta^*(q_4, u) = q_4$ qui sont tous les deux finaux ;
- si u termine par b ou par aa, alors $\delta^*(q_2, u)$ et $\delta^*(q_4, u)$ valent q_2 ou q_4 qui sont tous les deux finaux ;
- si u termine par a mais pas par aa, alors $\delta^*(q_2, u)$ et $\delta^*(q_4, u)$ valent q_1 ou q_3 qui sont tous les deux non finaux.

On en déduit qu'aucun mot ne sépare $\{q_2, q_4\}$.

Question 14 $\bigcup_q a^{-1}q$ représente l'ensemble des états qui ont une transition sortante étiquetée par a, c'est-à-dire Q tout entier (car A est complet). L'automate étant déterministe, pour $q \neq q'$, $a^{-1}q \cap a^{-1}q' = \emptyset$ (une transition étiquetée par a depuis un état ne peut pas mener vers deux états distincts). On en déduit l'égalité voulue.

Question 15 La file contient initialement $(\{q_0, q_2\}, \{q_0, q_4\}, \{q_1, q_2\}, \{q_1, q_4\}, \{q_2, q_3\}, \{q_3, q_4\})$. Le déroulement de l'algorithme est le suivant :

- on défile $\{q_0, q_2\}$ et $\{q_0, q_4\}$ sans rien enfiler, car q_0 n'a pas d'antécédent ;
- on défile $\{q_1, q_2\}$, on enfile $\{q_0, q_1\}$ et $\{q_0, q_3\}$ en regardant les antécédents par a ;
- on défile $\{q_1, q_4\}$ sans rien enfiler, car q_1 n'a pas d'antécédent par b et q_4 n'a pas d'antécédent par a ;
- on défile le reste de la file sans rien enfiler, soit car on ne trouve pas d'antécédents, soit car on retombe sur des paires déjà dans D.

On obtient le tableau suivant (on met une croix pour les paires dans D) :

	q_0	q_1	q_2	q_3
q_1	X		X	
q_2	X	X		X
q_3	X		X	
q_4	X	X		X

Question 16 Il s'agit d'une double implication :

$\Rightarrow$ Soit p, q deux états séparables. Alors, il existe $u \in \Sigma^*$ tel que $\delta^*(p, u) \in F$ et $\delta^*(q, u) \notin F$. Or, $\delta^*(p, u)$ et $\delta^*(q, u)$ sont distinguables par l'algorithme, d'après l'initialisation. Si on pose $u = a_1 a_2 \ldots a_n$, $u_i = a_1 \ldots a_i$, $p_i = \delta^*(p, u_i)$ et $q_i = \delta^*(q, u_i)$, alors par une récurrence décroissante simple, on montre que p_i et q_i sont dans D de par l'implémentation de l'algorithme.

$\Leftarrow$ Soit $\{p, q\} \in D$. Alors par construction, il existe une suite de lettres a_1, a_2, ..., a_n et deux suites d'états p_1, p_2, ..., p_n, q_1, q_2, ..., q_n telles que $p_1 = p$ et pour $i \in [\![1, n-1]\!]$, $p_{i+1} = \delta(p_i, a_i)$ (de même pour (q_i)). De plus, p_i et q_i sont dans D. De par l'initialisation, on peut de plus choisir $p_n \in F$ et $q_n \in F$, ce qui montre que p et q sont séparables.

Question 17 On remarque qu'une paire $\{p, q\}$ ne passe qu'au plus une fois par la file. La création de D est en $O(|Q|^2)$ si on utilise une matrice de booléens. On utilise les structures suivantes :

- Φ est une file, toutes les opérations peuvent s'implémenter en temps constant ;
- D est une matrice de booléens. Sa création est en temps $\mathcal{O}(|Q|^2)$, mais les opérations d'ajouts et de tests d'appartenance sont en $\mathcal{O}(1)$;
- le calcul des $a^{-1}q$ peut se faire en calculant « l'automate transposé » : c'est un calcul de graphe transposé, qui peut se faire en temps $\mathcal{O}(|Q||\Sigma|)$ (nombre de sommets + nombre d'arêtes).

On en déduit que seule la taille de toutes les boucles **Pour** compte. Or en utilisant la question 2, on a :

$$\sum_{\{p,q\} \in \mathcal{P}_2(Q)} \sum_{a \in \Sigma} |a^{-1}p||a^{-1}q| \leqslant \sum_{p} \sum_{a} |a^{-1}p| \sum_{q} |a^{-1}q| = |\Sigma||Q|^2$$

La complexité totale est en $\mathcal{O}(|\Sigma||Q|^2)$. Elle est atteinte lorsque toutes les paires sont séparables.

2.2 Automate minimal

Question 18 On a, pour tout $q, q' \in Q_i$ et $a \in \Sigma$, $\delta(q, a)$ et $\delta(q', a)$ sont équivalents. En effet, si ce n'était pas le cas, alors $\{q, q'\}$ serait séparable, ce qui est absurde. On en déduit que le calcul de $\delta_B(Q_i, a)$ ne dépend pas du choix du représentant.

Question 19 Soient q, q' deux états non co-accessibles de A. Alors $\forall u \in \Sigma^*$, $\delta^*(q, u) \notin F$ et $\delta^*(q', u) \notin F$. On en déduit que q et q' sont équivalents. Pour conclure, il suffit de remarquer que Q_i co-accessible $\Leftrightarrow$ tous les états de Q_i sont co-accessibles.

Question 20 On remarque que $\delta_B(\overline{q}, a) = \overline{\delta(q, a)}$. Par une récurrence rapide, on montre que pour tout $u \in \Sigma^*$, $\delta_B^*(\overline{q_0}, u) = \overline{\delta^*(q_0, u)}$. On en déduit :

$$u \in L(A) \Leftrightarrow \delta^*(q_0, u) \in F \Leftrightarrow \overline{\delta^*(q_0, u)} \in F_B \Leftrightarrow \delta^*(Q_0, u) \in F_B \Leftrightarrow u \in L(B)$$

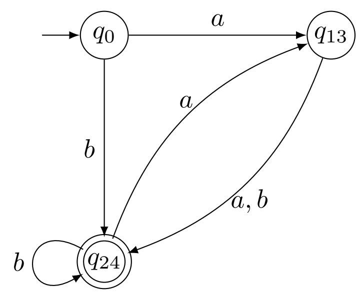
Question 21 Soit C un tel automate tel que $|Q_C| < |Q_B|$.

Par le principe des tiroirs, et par accessibilité, il existe deux états distincts $Q_i, Q_j \in Q_B$ et deux mots $u, v \in \Sigma^*$ tels que :

- $\delta_B(Q_0, u) = Q_i$,
- $\delta_B(Q_0, v) = Q_j$,
- $\delta_C(q_C, u) = \delta_C(q_c, v) = q \in Q_C$.

Or, par construction, Q_i et Q_j sont séparables. Il existe donc un mot w tel que $uw \in L(B)$ et $vw \notin L(B)$ (quitte à inverser i et j). C'est absurde, car alors $\delta_C(q, w)$ doit être un état à la fois final et non final de C.

Question 22 L'algorithme de séparation a permis de trouver que q_1 et q_3 sont équivalents, ainsi que q_2 et q_4. L'automate minimal est donc :



On retrouve bien l'automate minimal trouvé à la partie précédente.

Question 23 En appliquant l'algorithme de séparation, on obtient que les paires $\{q_2, q_4\}$ et $\{q_3, q_5\}$ sont équivalentes. On en déduit l'automate :