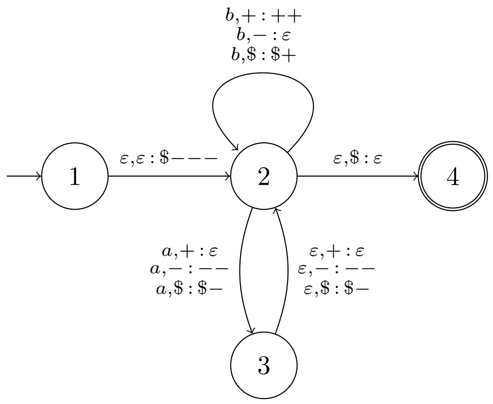
1. Langages Algébriques

1.1. Le langage $L_1 = \{w \in \{a,b\}^* \ : \ |w|_b = 2\,|w|_a + 3\}$ est algébrique.

1.1.1. *Avec des automates.* L'automate à pile suivant reconnait L_1. L'alphabet de pile est $\{\$, +, -\}$. On se permet d'empiler plusieurs lettres durant une transition, cela peut facilement être décomposé comme une suite de plusieurs transitions.



On vérifie d'abord que dans une configuration accessible, la pile est toujours de la forme $\$+^*$ ou $\$-^*$, sauf dans les états $1, 4$ où la pile est nécessairement vide. C'est une récurrence immédiate : toutes les transitions conservent cette propriété. On interprète alors la pile comme un entier : $\$+^n$ représente n, et $\$-^n$ représente $-n$. Les transitions sur les états 2 et 3 manipulent la pile comme un compteur qui peut être incrémenté (en lisant b) ou décrémenté (en lisant a).

Précisément, pour tout mot w on vérifie par récurrence sur w qu'il existe une unique exécution lisant w et arrivant dans l'état 2, et à la fin de cette exécution, la pile représente le nombre $|w|_b - 2\,|w|_a - 3$. Enfin, la transition menant à l'état final vérifie que la pile représente 0, et on accepte donc exactement les mots w tels que $|w|_b = 2\,|w|_a + 3$.

1.1.2. *Preuve alternative, avec des grammaires.* Il est aussi possible de reconnaitre L_1 avec une variante de la grammaire

$$T \to aTbTb \mid bSaSb \mid bSbSa \mid SS \mid \varepsilon$$

vue en TD, qui reconnait les mots avec deux fois plus de b que de a. En admettant la correction de cette grammaire, on l'étend par

$$S \to TbTbTbT,$$

en prenant S comme nouvel axiome.

Notons $L(T) \overset{\text{def}}{=} \{w \ : \ T \Rightarrow^* w\}$, et de même $L(S)$. De plus, pour tout mot w, notons $f(w) \overset{\text{def}}{=} |w|_b - |w|_a$. Si $w \in L(S)$, alors il existe $w_1, w_2, w_3, w_4 \in L(T)$ tels que $w = w_1 b w_2 b w_3 b w_4$. Puisque $w_i \in L(T)$, on a $f(w_i) = 0$, donc $f(w) = 3 + \sum_{i=1}^{4} w_i = 3$, et $w \in L_1$.

Réciproquement, soit $w \in L_1$. Soit w_1 le plus grand préfixe de w tel que $f(w_1) \leq 0$. Alors w_1 est nécessairement suivit par un b dans w (ou il ne serait pas maximal), et donc $f(w_1) = 0$ (sinon $f(w_1 b) \leq 0$). Donc $w = w_1 b w'$ avec $f(w') = 2$. On trouve alors de même w_2, w_3, w_4 tels que $w = w_1 b w_2 b w_3 b w_4$, et $f(w_i) = 0$. On a alors $w_i \in L(T)$, donc $w \in L(S)$.

1.2. Le langage $L_2 = \{w\#x \ : \ w, x \in \{a,b\}^*$ et w est un sous-mot de $x\}$ n'est pas algébrique.

Supposons par contradiction que L soit algébrique. Soit $p \in \mathbb{N}$ la longueur de pompage obtenue par le lemme de l'étoile. Considérons le mot $s = a^p b^p \# a^p b^p \in L$. Alors, s se factorise en $s = xuyvz$ telle que les conditions du lemme de l'étoile soient satisfaites. Il y a 4 cas :

— Si u ou v contient le $\#$, alors on a directement une absurdité car les mots de L contiennent tous exactement un seul $\#$.

— Si z contient le $\#$, on obtient alors une contradiction en considérant xu^2yv^2z car le mot à gauche du $\#$ est plus long que celui à droite, donc ce premier n'est pas un sous-mot de ce dernier.

— Si x contient le $\#$, on obtient de même une contradiction en considérant $xu^0yv^0z = xyz$.

— Si enfin y contient le $\#$, remarquons que la condition $|uyv|$ du lemme de l'étoile garantie que u ne contient que des b, et v ne contient que des a. Si u est non vide, xu^2yv^2z contient plus de b à gauche du $\#$ qu'à droite, donc n'est pas dans L_2. De même, si v est non vide, xu^0yv^0z contient plus de a à gauche du $\#$ qu'à droite. Dans les deux cas, on obtient une contradiction.

1.3. Le langage $L_3 = \{a^p \ : \ p$ est premier$\}$ n'est pas algébrique.

Supposons par contradiction que L_3 soit algébrique. Soit $m \in \mathbb{N}$ la longueur de pompage obtenue par le lemme de l'étoile. Considérons le mot $s = a^p \in L$, où p est un nombre premier plus grand que m. Alors, s se factorise en $s = xuyvz$ telle que les conditions du lemme de l'étoile soient satisfaites. Soit $t = |u| + |v| > 0$. En considérant $xu^{p+1}yv^{p+1}z$, nous avons

$$|xu^{p+1}yv^{p+1}z| = p - t + t(p+1) = p(t+1),$$

où $t + 1 > 1$. La longueur du mot $xu^{p+1}yv^{p+1}z$ n'est pas un nombre premier, et donc $xu^{p+1}yv^{p+1}z \notin L$, une contradiction.

1.4. Le langage $L_4 = \{a^{n_0}ba^{n_1}b \cdots a^{n_k}b \ : \ \exists j \geq 0, n_j \neq j\}$ est algébrique. Il est engendré par la grammaire $\mathcal{G}$ suivante.

$$\begin{aligned}
S &\to T_3 b T_2 \\
T_0 &\to aT_0 \mid \varepsilon \\
T_1 &\to T_0 b \\
T_2 &\to T_1 T_2 \mid \varepsilon \\
T_3 &\to T_1 T_3 a \mid T_1 T_2 \mid aT_0
\end{aligned}$$

En notant $L(T) \overset{\text{def}}{=} \{w \ : \ T \Rightarrow^* w\}$ pour un non-terminal T, on vérifie par des récurrences immédiates sur la longueur des dérivations :

$$L(T_0) = a^*, \ L(T_1) = a^*b, \text{ et } L(T_2) = (a^*b)^*$$

On vérifie ensuite que

$$L(T_3) = \{a^{n_0}ba^{n_1}b \dots a^{n_{k-1}}ba^{n_k} \ : \ n_k \neq k\}$$

En effet, si $w \in L(T_3)$, alors on a l'un des cas suivants :

— $w \in L(T_1)L(T_2) = (a^*b)^+$, qui est de la forme désirée avec $n_k > k = 0$.

— $w \in aL(T_0) = a^+$, qui est de la forme désirée avec $0 = n_k < k$.

— $w \in L(T_1)L(T_3)a$, auquel cas $w = a^nbw'a$ pour un $w' \in L(T_3)$. Par récurrence sur w, on peut supposer que $w' = a^{n_0}ba^{n_1}b \dots a^{n_k}$ avec $k \neq n_k$. Alors $w = a^nba^{n_0}ba^{n_1}b \dots a^{n_k+1}$ qui est de la forme désirée puisque $k+1 \neq n_k+1$.

Réciproquement, si $w = a^{n_0}ba^{n_1}b \dots a^{n_k}$ avec $n_k \neq k$, on peut distinguer 3 cas.

— Si $k = 0$, on a $w \in (a^*b)^+ \subset L(T_3)$ par la règle $T_3 \to T_1T_2$.

— Si $n_k = 0$, on a $w \in a^+ \subset L(T_3)$ par la règle $T_3 \to aT_0$.

— Sinon on a $k > 0$ et $n_k > 0$, et par récurrence sur w on peut supposer que $w' \stackrel{\text{def}}{=} a^{n_1}ba^{n_2}b \dots a^{n_{k-1}}ba^{n_k-1} \in L(T_3)$. Puisque $w = a^*bw'a$, on en déduit $w \in L(T_3)$ via la règle $T_3 \to T_1T_3a$.

Montrons enfin $L(S) = L_4$. Si $w = a^{n_0}ba^{n_1}b \dots a^{n_k}b$ et $j \neq n_j$, on peut décomposer $w = ubv$ avec $u = a^{n_0}b \dots a^{n_j}$, et $v = a^{n_{j+1}}b \dots a^{n_k}b$. On a bien $u \in L(T_3)$ et $v \in (a^*b)^*$, donc $w \in L(S)$. Réciproquement, tout $w \in L(S)$ se décompose en ubv avec $u \in L(T_3)$ et $v \in L(T_2)$. On peut alors écrire $u = a^{n_0}b \dots a^{n_j}$ avec $j \neq n_j$, et $v = a^{n_{j+1}}b \dots a^{n_k}b$. On a donc bien $w = a^{n_0}b \dots a^{n_k}b$ vérifiant $j \neq n_j$.

2. Mélange

2.1. On vérifie que $\mathrm{Mel}(a^n, b^m)$ est l'ensemble des mots constitués de n lettres a et m lettres b, par une récurrence facile sur $n + m$. Donc, $\mathrm{Mel}\left((aa)^*, (bbb)^*\right)$ est l'ensemble des mots w tels que $|w|_a$ soit pair, et $|w|_b$ soit divisible par 3. L'ensemble des w tels que $|w|_a$ soit pair est bien sûr rationnel, et de même l'ensemble des w tels que $|w|_b$ soit divisible par 3. Donc leur intersection $\mathrm{Mel}\left((aa)^*, (bbb)^*\right)$ est aussi rationnelle.

2.2. Soient L_1, L_2 des langages tels que L_i soit reconnu par un automate $\mathcal{A}_i = (Q_i, \Sigma, \delta_i, I_i, F_i)$. On supposera que $\mathcal{A}_i$ n'a pas de transition sur ε. On construit une variante de l'automate produit, $\mathcal{A} = (Q, \Sigma, \delta, I, F)$ tel que

— $Q \stackrel{\text{def}}{=} Q_1 \times Q_2$,

— $I \stackrel{\text{def}}{=} I_1 \times I_2$,

— $F \stackrel{\text{def}}{=} F_1 \times F_2$, et

— pour $q = (q_1, q_2) \in Q$, $a \in \Sigma$, on a

$$\delta(q, a) \stackrel{\text{def}}{=} (\delta_1(q_1, a) \times \{q_2\}) \cup (\{q_1\} \times \delta_2(q_2, a))$$

L'idée est que là où l'automate produit habituel fait avancer $\mathcal{A}_1$ et $\mathcal{A}_2$ en parallèle, on choisit ici de façon non déterministe de faire avancer $\mathcal{A}_1$ ou $\mathcal{A}_2$. La correction de cette construction est une conséquence de résultat suivant.

Lemme 1. *Pour tous $w \in \Sigma^*$, $q_1, q_1' \in Q_1$ et $q_2, q_2' \in Q_2$, on a $(q_1, q_2) \xrightarrow{w}_{\mathcal{A}} (q_1', q_2')$ si et seulement s'il existe w_1, w_2 tels que $q_i \xrightarrow{w_i}_{\mathcal{A}_i} q_i'$ et $w \in \mathrm{Mel}(w_1, w_2)$.*

Preuve. Par récurrence sur w. Dans le cas de base $w = \varepsilon$, on a $\varepsilon \in \mathrm{Mel}(w_1, w_2)$ si et seulement si $w_1 = w_2 = \varepsilon$, et dans chaque automate, on vérifie $q \xrightarrow{\varepsilon} q'$ si et seulement si $q = q'$, puisqu'il n'y a pas d'ε-transitions. On en déduit l'équivalence demandée.

Soit maintenant $w = av$. Supposons $(q_1, q_2) \xrightarrow{w}_{\mathcal{A}} (q_1', q_2')$, et considérons la première étape :

$$(q_1, q_2) \xrightarrow{a}_{\mathcal{A}} (q_1'', q_2'') \xrightarrow{v}_{\mathcal{A}} (q_1', q_2')$$

La définition de δ donne

— $q_1 \xrightarrow{a}_{\mathcal{A}_1} q_1''$ et $q_2 = q_2''$, ou

— $q_1 = q_1''$ et $q_2 \xrightarrow{a}_{\mathcal{A}_2} q_2''$.

Sans perte de généralité, on considère le premier cas. Par hypothèse de récurrence appliquée à v, il existe donc v_1, v_2 tels que $q_i'' \xrightarrow{v_i}_{\mathcal{A}_i} q_i'$ et $v \in \mathrm{Mel}(v_1, v_2)$. On fixe alors $w_1 = av_1$, $w_2 = v_2$. On a bien $w \in \mathrm{Mel}(w_1, w_2)$, et de plus

$$q_1 \xrightarrow{a}_{\mathcal{A}_1} q_1'' \xrightarrow{v_1}_{\mathcal{A}_1} q_1'$$
$$q_2 = q_2'' \xrightarrow{w_2=v_2}_{\mathcal{A}_2} q_2'$$

d'où le résultat.

Réciproquement, soient w_1, w_2 tels que $q_i \xrightarrow{w_i}_{\mathcal{A}_i} q_i'$ et $w \in \mathrm{Mel}(w_1, w_2)$. La définition de mélange donne alors que

- $w_1 = av_1$, et $v \in \mathrm{Mel}(v_1, w_2)$, ou
- $w_2 = av_2$, et $v \in \mathrm{Mel}(w_1, v_2)$.

Sans perte de généralité, on considère le premier cas. Il existe alors un q_1'' tel que

$$q_1 \xrightarrow{a}_{\mathcal{A}_1} q_1'' \xrightarrow{v_1}_{\mathcal{A}_1} q_1'$$

Par hypothèse de récurrence appliquée à $v \in \mathrm{Mel}(v_1, w_2)$, on a $(q_1'', q_2) \xrightarrow{v}_{\mathcal{A}} (q_1', q_2')$. Alors,

$$(q_1, q_2) \xrightarrow{a}_{\mathcal{A}} (q_1'', q_2) \xrightarrow{v}_{\mathcal{A}} (q_1', q_2').$$

□

2.3. Notons $f : \{a, b, c\}^* \to \{a, b\}^*$ la fonction qui supprime tous les c d'un mots, et garde les autres lettres. Pour tout $L \subseteq \{a, b\}^*$, on vérifie par une récurrence facile sur la longueur d'un mot que

$$\mathrm{Mel}(L, c^*) = f^{-1}(L)$$

Si L est algébrique (comme par exemple $\{a^n b^n \ : \ n \in \mathbb{N}\}$) et $\mathcal{A}$ est un automate à pile reconnaissant L, on peut modifier $\mathcal{A}$ en ajoutant sur chaque état q une transition $q \xrightarrow{c, \varepsilon \to \varepsilon} q$, qui ne modifie pas la pile. Après cette transformation, le langage reconnu est $f^{-1}(L)$, qui est donc algébrique.

2.4. Soient L_1 algébrique reconnu par un automate à pile $\mathcal{A}_1 = (Q_1, \Sigma, \Gamma, \delta_1, I_1, F_1)$ et L_2 rationnel reconnu par un automate fini $\mathcal{A}_2 = (Q_2, \Sigma, \delta_2, I_2, F_2)$. Comme à la question 2, on construit une variante de l'automate produit, $\mathcal{A} = (Q, \Sigma, \Gamma, \delta, I, F)$ avec

- $Q = Q_1 \times Q_2$,
- $I = I_1 \times I_2$,
- $F = F_1 \times F_2$, et
- pour $q = (q_1, q_2) \in Q$, $a \in \Sigma$, $s \in \Gamma$ on a

$$\delta(q, a, s) \stackrel{\mathrm{def}}{=} \{(q_1', q_2, s') \ : \ (q_1', s') \in \delta_1(q_1, a, s)\} \cup \{(q_1, q_2', s) \ : \ q_2' \in \delta_2(q_2, a)\}$$

Une configuration de $\mathcal{A}$ est une paire (q, π) où $q = (q_1, q_2) \in Q$ est l'état, et $\pi \in \Gamma^*$ est le contenu de la pile. On se permettra de réécrire cette configuration comme (γ_1, q_2) où $\gamma_1 \stackrel{\mathrm{def}}{=} (q_1, \pi)$ est une configuration de $\mathcal{A}_1$. Avec cette notation, la correction de l'automate s'énonce comme suit.

Lemme 2. *Pour tous $w \in \Sigma^*$, γ_1, γ_1' configurations de $\mathcal{A}_1$, et $q_2, q_2' \in Q_2$, on a $(\gamma_1, q_2) \xrightarrow{w}_{\mathcal{A}} (\gamma_1', q_2')$ si et seulement s'il existe w_1, w_2 tels que $\gamma_1 \xrightarrow{w_1}_{\mathcal{A}_1} \gamma_1'$, $q_2 \xrightarrow{w_2}_{\mathcal{A}_2} q_2'$, et $w \in \mathrm{Mel}(w_1, w_2)$.*

La preuve est la même qu'à la question 2, en remplaçant les états de l'automate fini $\mathcal{A}_1$ par les configurations de l'automate à pile.

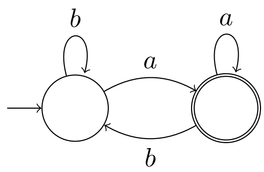
2.5. Le mélange de deux langages algébriques n'est pas nécessairement algébrique. Considérons $L_1 = \{w \ : \ |w|_a = |w|_b\}$ et $L_2 = \{w \ : \ |w|_c = |w|_d\}$, qui sont algébriques. On vérifie facilement que le mélange est alors

$$L \overset{\text{def}}{=} \text{Mel}(L_1, L_2) = \{w \ : \ |w|_a = |w|_b \text{ et } |w|_c = |w|_d\}$$

Supposons par contradiction que L soit algébrique. Son intersection avec $a^*(bc)^*d^*$, qui est $\{a^n(bc)^n d^n \ : \ n \in \mathbb{N}\}$, serait alors algébrique. Ce n'est pas le cas, la preuve est la même que pour $\{a^n b^n c^n \ : \ n \in \mathbb{N}\}$, par pompage.

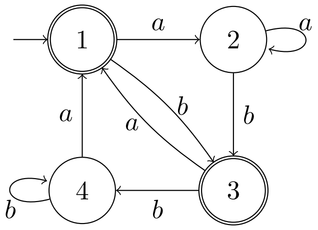
3. Automates de Büchi

3.1. Soit $L = (b^*a)^\omega$, c'est à dire l'ensemble des mots contenant une infinité de a. L'automate suivant reconnait L.



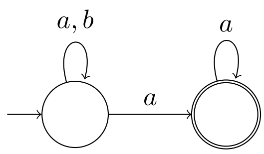
En effet, dans (l'unique) exécution de cet automate sur un mot w, le $n$ième état est acceptant ssi la $n$ième lettre de w est un a, donc w est accepté ssi il contient une infinité de a.

3.2.



Le langage reconnu par cet automate est l'ensemble des mots avec une infinité de a et une infinité de b, que l'on peut écrire $b^*(a^+b^+)^\omega$. En effet, on vérifie que dans (l'unique) exécution de l'automate sur un mot w, le $n$ième état (avec $n \geq 2$) est acceptant ssi les $(n-1)$ième et $n$ième lettres sont distinctes. Donc w est accepté ssi il contient une infinité d'alternance (i.e. un facteur ab ou ba), ce qui est équivalent à contenir une infinité de a et de b.

3.3. L'automate suivant accepte l'ensemble des mots avec un nombre fini de b.



En effet, si w n'a qu'un nombre fini de b, on peut écrire $w = va^\omega$ pour un certain mot fini v. L'automate peut alors lire v sur l'état initial, puis lire a^ω sur l'état acceptant, et accepte donc v. Inversement, une exécution acceptante sur un mot w doit à un moment atteindre l'état final, disons à la $n$ième lettre de w. Alors toutes les lettres de w après la $n$ième sont des a.

3.4. Soit $\mathcal{A} = (Q, \Sigma, \delta, i, F)$ un automate déterministe acceptant tous les mots de $\Sigma^* a^\omega$. Pour tout $w \in \Sigma^*$, wa^ω est accepté par $\mathcal{A}$. Donc pour tout w, il existe un $n \in \mathbb{N}$ tel que $\delta(i, wa^n) \in F$. Une récurrence immédiate donne alors une suite $(n_i)_{i\in\mathbb{N}}$ telle que pour tout i,

$$\delta(i, a^{n_0} b a^{n_1} \dots b a^{n_i}) \in F$$

Alors, le mot infini $a^{n_0} b a^{n_1} b a^{n_2} \dots$ est accepté par $\mathcal{A}$, mais n'est pas dans $\Sigma^* a^\omega$.

Ce langage est donc reconnaissable par NFA, mais pas par DFA (q. 3). De plus, son complémentaire est reconnaissable par DFA (q. 1).

3.5. Soient L_1, L_2 langages de mots infinis, et $\mathcal{A}_k = (Q_k, \Sigma, \delta_k, i_k, F_k)$ un DFA reconnaissant L_k. On considère $\mathcal{A}$ l'automate produit habituel, mais avec pour ensemble d'état finals

$$F = \{(q_1, q_2) \in Q_1 \times Q_2 \ : \ q_1 \in F_1 \text{ ou } q_2 \in F_2\}$$

On remarquera que $\mathcal{A}$ est bien un DFA.

Soit $w \in \Sigma^\omega$, et notons $(q_n^{(k)})_{n\in\mathbb{N}}$ la suite d'états de l'unique exécution de $\mathcal{A}_k$ sur w commençant en i_k. Alors $\left(q_n^{(1)}, q_n^{(2)}\right)_{n\in\mathbb{N}}$ est l'unique exécution de $\mathcal{A}$ sur w commençant en (i_1, i_2). On a alors

$$\begin{aligned}
w \in L(\mathcal{A}) &\iff \left\{n \ : \ (q_n^{(1)}, q_n^{(2)}) \in F\right\} \text{ est infini} \\
&\iff \left\{n \ : \ q_n^{(1)} \in F_1 \text{ ou } q_n^{(2)} \in F_2\right\} \text{ est infini} \\
&\iff \left\{n \ : \ q_n^{(1)} \in F_1\right\} \text{ est infini} \quad \text{ou} \quad \left\{n \ : \ q_n^{(2)} \in F_2\right\} \text{ est infini} \\
&\iff w \in L_1 \cup L_2
\end{aligned}$$

3.6. On considère un automate $\mathcal{A}'$ avec états $Q \times \{1, 2\}$. La fonction de transition alterne entre les deux copies de $\mathcal{A}$ dès qu'un état acceptant est rencontré, c'est à dire

$$\delta'((q,1), a) = \begin{cases} \delta(q,a) \times \{1\} & \text{si } q \notin F_1 \\ \delta(q,a) \times \{2\} & \text{si } q \in F_1 \end{cases}$$

$$\delta'((q,2), a) = \begin{cases} \delta(q,a) \times \{2\} & \text{si } q \notin F_2 \\ \delta(q,a) \times \{1\} & \text{si } q \in F_2 \end{cases}$$

L'ensemble d'états initiaux de $\mathcal{A}'$ est $I \times \{1\}$, et les états acceptants sont $F' = F_1 \times \{1\} \cup F_2 \times \{2\}$. Si $\mathcal{A}$ est déterministe, cette construction est déterministe.

Soit $w \in \Sigma^\omega$ un mot reconnu par $\mathcal{A}'$, et $(q_n, i_n)_{n\in\mathbb{N}}$ la suite d'états d'une exécution acceptant w, avec $q_n \in Q, i_n \in \{1, 2\}$. Remarquons que $(q_n)_{n\in\mathbb{N}}$ est une exécution de $\mathcal{A}$ sur w. De plus, on a $i_n \neq i_{n+1}$ si est seulement si $(q_n, i_n) \in F'$. La sous-suite constituées d'états dans F' (qui est infinie puisque l'exécution est acceptante) est donc de la forme

$$(q_{n_0}, 1), (q_{n_1}, 2), (q_{n_2}, 1), (q_{n_3}, 2), \dots$$

et vérifie donc $q_{n_k} \in F_1$ pour k pair et $q_{n_k} \in F_2$ pour k impair. Donc $(q_n)_{n\in\mathbb{N}}$ visite infiniment souvent F_1 et F_2, c'est à dire que $\mathcal{A}$ accepte w.

Réciproquement, soit $w \in \Sigma^\omega$ un mot reconnu par $\mathcal{A}$, et $(q_n)_{n\in\mathbb{N}}$ la suite d'états d'une exécution acceptant w. Il existe une unique suite $(i_n)_{n\in\mathbb{N}}$ telle que $(q_n, i_n)_{n\in\mathbb{N}}$ soit une exécution valide de $\mathcal{A}'$ sur w. Elle peut être définie par récurrence : $i_0 = 1$, et $i_{n+1} \neq i_n$ ssi $q_n \in F_{i_n}$. Montrons que $(q_n, i_n)_{n\in\mathbb{N}}$ est une exécution acceptante. Soit $n_0 \in \mathbb{N}$ arbitraire, et supposons sans perte de généralité que $i_{n_0} = 1$. Soit $n \geq n_0$ minimal tel que $q_n \in F_1$. Alors on a aussi $i_n = 1$ (par minimalité de n), et donc $(q_n, i_n) \in F'$. On en déduit que $(q_n, i_n)_{n \in \mathbb{N}}$ visite infiniment souvent F'. C'est donc une exécution acceptante sur w.

3.7. Si $\mathcal{A}_1$, $\mathcal{A}_2$ sont deux NFA (resp. DFA), on considère l'automate produit habituel, sauf pour les états acceptants : on munit le produit de deux conditions d'acceptations qui correspondent aux états acceptants de $\mathcal{A}_1$ et à ceux de $\mathcal{A}_2$ respectivement. Avec la sémantique définie à la question précédente, cet automate reconnait $L(\mathcal{A}_1) \cap L(\mathcal{A}_2)$. Il suffit d'appliquer la transformation de la question précédente pour obtenir un NFA (resp. DFA).