TD Théorie des Langages 1 — Feuille 1
Induction structurelle, Notions de langage

Exercice 7 On considère dans cet exercice des langages définis sur $V = \{a, b\}$.

▷ Question 1 Définir **par concaténation et itération** le langage L_1 des mots constitués d'une séquence de a suivie d'une séquence de b. Les séquences peuvent éventuellement être vides.

▷ Question 2 Définir **par induction** le langage L_2 des mots de la forme $a^n b^n$ avec $n > 0$.

▷ Question 3 Définir **à l'aide des opérations ensemblistes classiques** et des langages précédents le langage L_3 des mots de la forme $a^i b^j$ avec $i \neq j$.

Solution de l'Exercice 7.

▷ Question 1 $L_1 = \{a\}^*\{b\}^*$.

▷ Question 2

Base : $ab \in L_2$.
Induction : Si $w \in L_2$ alors $awb \in L_2$.

▷ Question 3 $L_3 = L_1 \setminus (\{\varepsilon\} \cup L_2)$.

Exercice 9 Pour chacune des égalités suivantes identifier celles qui sont vraies et celles qui sont fausses. Pour la première catégorie on donnera une intuition de la preuve. Pour la seconde catégorie on donnera des contre-exemples et on identifiera les inclusions uni-directionnelles.

1. $L^* = (L^*)^*$
2. $L^* \cup M^* = (L \cup M)^*$
3. $(L^* \cup M^*)^* = (L \cup M)^*$
4. $L^+ = L^* - \{\epsilon\}$
5. $(LM)^* = L^*M^*$
6. $(L \cup M)^* = (L^*M^*)^*$

Solution de l'Exercice 9. **Note :** Les preuves de cet exercice sont nettement plus lisibles en utilisant la propriété $L \subseteq M \implies L^* \subseteq M^*$ qui est démontrée à l'exercice 10. C'est ce qu'on fait ici.

1. L'égalité $L^* = (L^*)^*$ est vraie.
   L'inclusion $L^* \subseteq (L^*)^*$ est facile à démontrer car pour tout langage M, on a $M \subseteq M^*$, soit en prenant une suite de longueur 1 de mots de M, soit en utilisant la définition de $M^* = \bigcup_{i \geqslant 0} M^i$.
   Réciproquement, considérons un mot de $(L^*)^*$. Ce mot est de la forme $w_1 \cdots w_n$, où pour tout $i \in \{1, \ldots, n\}$, $w_i \in L^*$. Par définition, chaque w_i est de la forme $x_{i,1} \cdots x_{i,n_i}$, où pour tout $j = 1 \ldots, n_i$, $x_{i,j} \in L$. On en déduit que w est une concaténation finie d'éléments de L, les $x_{i,j}$, et est donc dans L^*.

2. Comme $L \subseteq (L \cup M)$, on a $L^* \subseteq (L \cup M)^*$, et de même, $M^* \subseteq (L \cup M)^*$. Donc $L^* \cup M^* \subseteq (L \cup M)^*$.
L'inclusion réciproque est fausse : si $L = \{a\}$ et $M = \{b\}$, alors $ab \in (L \cup M)^*$, mais $ab \notin L^* \cup M^*$.
3. L'égalité $(L \cup M)^* = (L^* \cup M^*)^*$ est vraie.
On a $(L \cup M)^* \subseteq (L^* \cup M^*)^*$. En effet, comme $L \subseteq L^*$ et $M \subseteq M^*$, on a $(L \cup M) \subseteq (L^* \cup M^*)$, d'où $(L \cup M)^* \subseteq (L^* \cup M^*)^*$.
Réciproquement, comme $L^* \subseteq (L \cup M)^*$ et $M^* \subseteq (L \cup M)^*$, on en déduit que $L^* \cup M^* \subseteq (L \cup M)^*$ puis que $(L^* \cup M^*)^* \subseteq ((L \cup M)^*)^* = (L \cup M)^*$ (question 1).
4. On a $L^* \setminus \{\varepsilon\} \subseteq L^+$ car $L^* = \bigcup_{i \geqslant 0} L^i = L^0 \cup \bigcup_{i>0} L^i = \{\varepsilon\} \cup L^+$.
L'inclusion réciproque est fausse : par exemple si $L = \{\varepsilon\}$, alors $L^+ = \{\varepsilon\}$ mais $L^* \setminus \{\varepsilon\} = \emptyset$. Plus généralement, elle est fausse ssi $\varepsilon \in L$.
5. Aucune des deux inclusions n'est vraie.
Posons $L = \{a\}$ et $M = \{b\}$. Alors $abab \in (LM)^*$, mais ce mot n'est pas dans L^*M^*. L'inclusion réciproque est également fausse : $aa \in L^*M^*$ mais $aa \notin (LM)^*$.
6. L'égalité $(L \cup M)^* = (L^*M^*)^*$ est vraie.
Comme $L \subseteq L^*M^*$ et $M \subseteq L^*M^*$, on a $(L \cup M) \subseteq L^*M^*$ et donc $(L \cup M)^* \subseteq (L^*M^*)^*$.
Réciproquement, comme $L^* \subseteq (L \cup M)^*$ et $M^* \subseteq (L \cup M)^*$, on a $L^*M^* \subseteq ((L \cup M)^*)^* = (L \cup M)^*$, et $(L^*M^*)^* \subseteq ((L \cup M)^*)^* = (L \cup M)^*$

Exercice 10 [Avancé] Soient L et M des langages sur un vocabulaire V. Montrer que si $L \subseteq M$ alors $L^* \subseteq M^*$. La réciproque est-elle vraie ? Justifier.

Solution de l'Exercice 10. Soit $w \in L^*$. Par définition w est de la forme $w_1 \cdots w_n$, où pour tout $i = 1, \ldots, n$, $w_i \in L$. Comme $L \subseteq M$, on a pour tout i, $w_i \in M$, donc $w \in M^*$. D'où $L^* \subseteq M^*$.
L'inclusion réciproque est fausse : si $L = \{aa\}$ et $M = \{a\}$, alors $L^* \subseteq M^*$, mais $L \not\subseteq M$.

TD Théorie des langages 1 — Feuille 2

Langages réguliers – Automates finis, epsilon-transitions

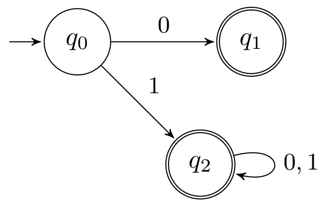
Exercice 1 Construire des automates reconnaissant les langages suivants :

1. L'ensemble des nombres binaires sans zéro inutile en tête.
2. Les mots sur $\{a, b\}$ contenant deux $'a'$ et/ou deux $'b'$ consécutifs.
3. **[Avancé]** Les mots sur $\{a, b\}$ contenant un nombre pair de $'a'$ et un nombre pair de $'b'$.

Solution de l'Exercice 1.

1. **Première méthode :**

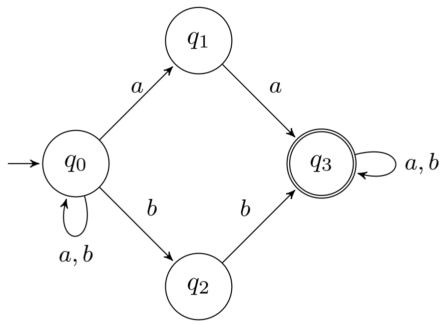
Q	0	1	initial/final
q_0	q_1	q_2	I
q_1	$\times$	$\times$	F
a_2	q_2	q_2	F



Deuxième méthode : On peut procéder par union : « zéro », un automate à 2 états, et « autres », un automate à 2 états. L'union fait le reste (2 états initiaux, 2 états acceptants, union des transitions...).

2.

Q	a	b	initial/final
q_0	q_0, q_1	q_0, q_2	I
q_1	q_3	$\times$	
q_2	$\times$	q_3	
q_3	q_3	q_3	F



Comme précédemment, on peut procéder par union :

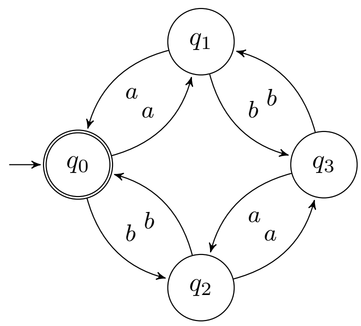
(a) construire un automate pour reconnaître deux $'a'$ consécutifs est facile (automate non-déterministe à 3 états) ;

(b) idem pour deux $'b'$ consécutifs ;

(c) faire l'union de ces deux automates (le « et » dans « et/ou » de l'énoncé est juste là pour perturber...)

3.

Q	a	b	initial/final
q_0	q_1	q_2	I, F
q_1	q_0	q_3	
q_2	q_3	q_0	
q_3	q_2	q_1	

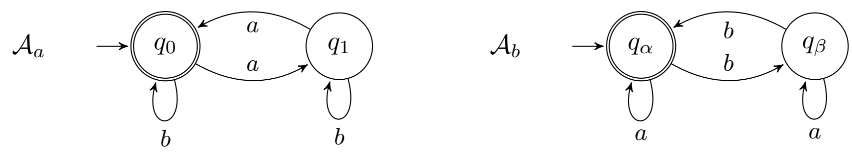


Remarque : On peut voir cet automate comme le produit cartésien deux deux automates déterministes complets $\mathcal{A}_a$ et $\mathcal{A}_b$ qui reconnaissent respectivement :

(a) pour $\mathcal{A}_a$, les mots contenant un nombre pair de 'a',

(b) pour $\mathcal{A}_b$, les mots contenant un nombre pair de 'b'.

En effet, si on prend ces deux automates :



L'automate pour les mots contenant un nombre pair de 'a' et un nombre pair de 'b' est alors l'automate produit $\mathcal{A}_a \times \mathcal{A}_b$ de $\mathcal{A}_a$ et $\mathcal{A}_b$ défini par

$$(Q_{\mathcal{A}_a} \times Q_{\mathcal{A}_b}, \quad V, \quad \delta_{\mathcal{A}_a} \times \delta_{\mathcal{A}_b}, \quad (i_{\mathcal{A}_a}, i_{\mathcal{A}_b}), \quad F_{\mathcal{A}_a} \times F_{\mathcal{A}_b})$$

avec

$$\delta_{\mathcal{A}_a} \times \delta_{\mathcal{A}_b} = \{((p_1, p_2), x, (q_1, q_2)) \mid (p_1, x, q_1) \in \delta_{\mathcal{A}_a} \text{ et } (p_2, x, q_2) \in \delta_{\mathcal{A}_b}\}$$

Autrement dit, les états sont des paires d'un état de chaque automate. L'automate produit permet d'exécuter deux automates déterministes complets en parallèle. Choisir $F_{\mathcal{A}_a} \times F_{\mathcal{A}_b}$ permet de reconnaître l'intersection des langage $\mathcal{L}(\mathcal{A}_a)$ et $\mathcal{L}(\mathcal{A}_b)$ mais on peut faire d'autre choix pour reconnaître l'union ou la différence.

Exercice 2 L'ensemble des littéraux numériques en Python forme un langage, formellement défini dans `https://docs.python.org/3/reference/lexical_analysis.html#numeric-literals`. Dans cet exercice, on considère un sous-ensemble des littéraux entiers et flottants écrits en base 10. Ils sont composés d'une partie entière, d'une partie décimale optionnelle et d'un exposant optionnel ; ils sont définis sur le vocabulaire

$$V = \{\mathbf{0}, \ldots, \mathbf{9}, \mathbf{e}, \mathbf{E}, \mathbf{.}, +, -\}.$$

On définit les ensembles suivants :

$$\begin{aligned}
\texttt{nonzerodigit} &\stackrel{\text{def}}{=} \{\mathbf{1}, \ldots, \mathbf{9}\} \\
\texttt{digit} &\stackrel{\text{def}}{=} \{\mathbf{0}\} \cup \texttt{nonzerodigit} \\
\texttt{integer} &\stackrel{\text{def}}{=} \texttt{nonzerodigit}\,(\texttt{digit}^*) \cup \{\mathbf{0}\}^+ \\
\texttt{dot} &\stackrel{\text{def}}{=} \{\mathbf{.}\} \\
\texttt{pointfloat} &\stackrel{\text{def}}{=} (\texttt{digit}^*)\,\texttt{dot}\,(\texttt{digit}^+) \cup (\texttt{digit}^+)\,\texttt{dot} \\
\texttt{exponent} &\stackrel{\text{def}}{=} \{\mathbf{e}, \mathbf{E}\}\,\{\varepsilon, +, -\}\,\texttt{digit}^+ \\
\texttt{exponentfloat} &\stackrel{\text{def}}{=} \big(\texttt{digit}^+ \cup \texttt{pointfloat}\big)\,\texttt{exponent} \\
\texttt{number} &\stackrel{\text{def}}{=} \texttt{integer} \cup \texttt{pointfloat} \cup \texttt{exponentfloat}
\end{aligned}$$

▷ QUESTION 1 Parmi les mots suivants, lesquels appartiennent à `number` ? Lesquelles n'y appartiennent pas ?

.314, **.3E+4**, **0.5E−2**, **0000**, **E67**, **1E7e3**, **6E+1234**, **2E++3.4**

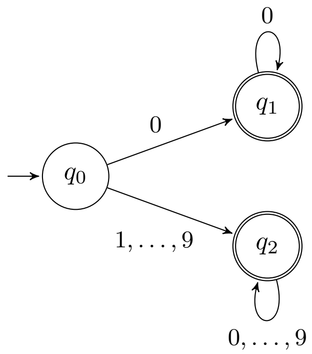
▷ Question 2 Donner un automate qui reconnaît le langage `integer`.

▷ Question 3 Donner un automate qui reconnaît le langage `number`.

Solution de l'Exercice 2.

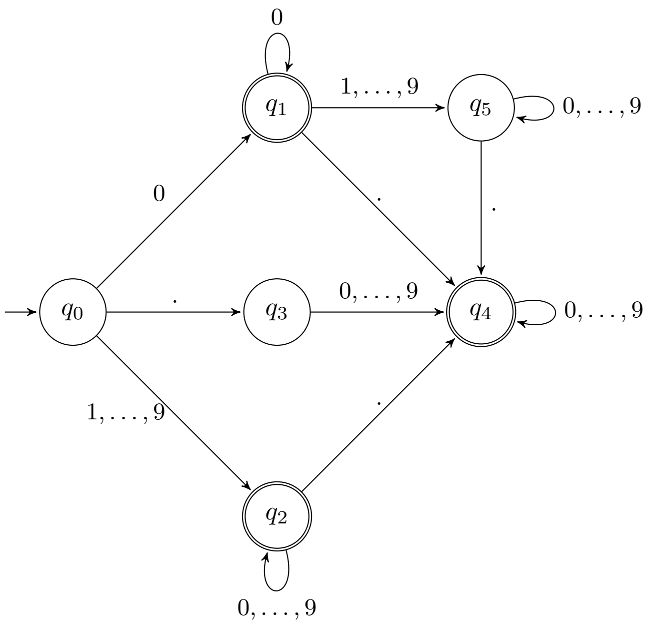
▷ Question 1 Eléments dans `number` : **.314**, **.3E+4**, **0.5E−2**, **0000**, **6E+1234**

▷ Question 2 L'automate suivant reconnaît `integer` :

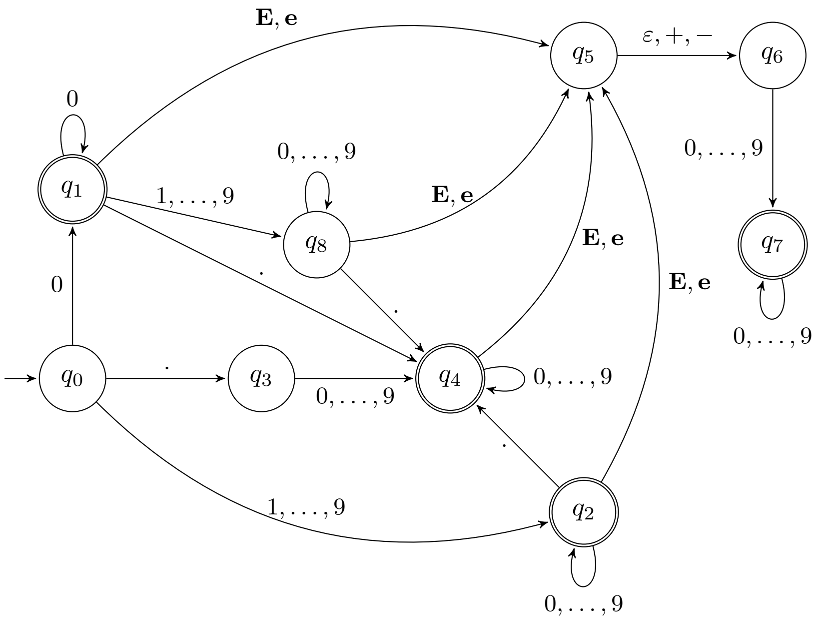


▷ Question 3 Le plus simple pour résoudre cet exercice est de procéder par étapes.

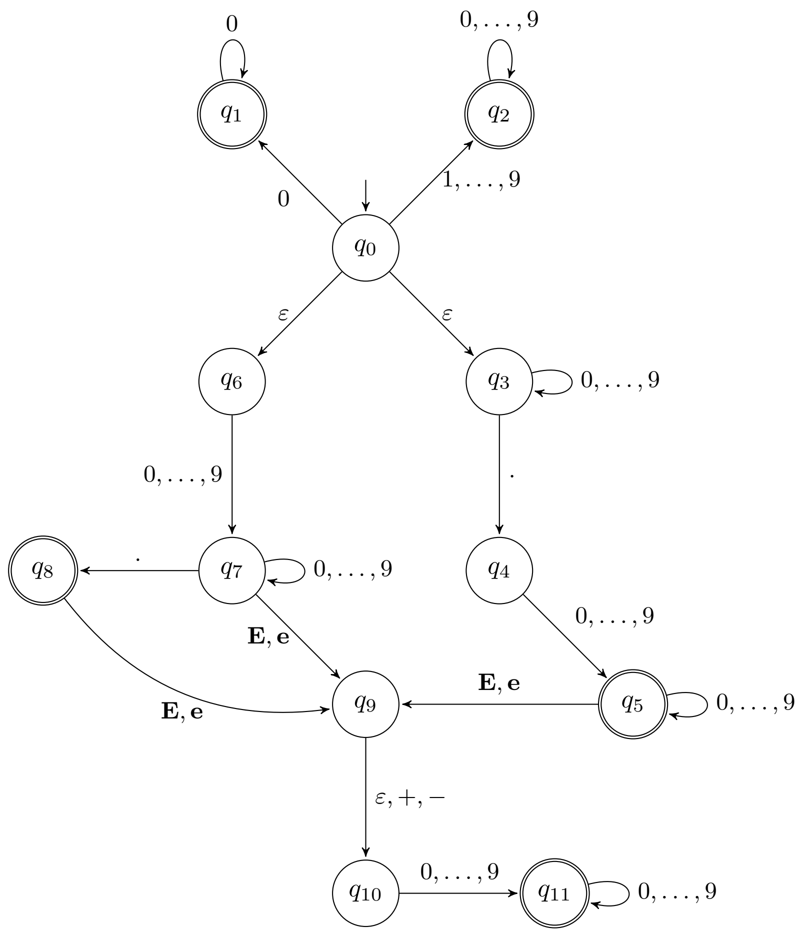
— L'automate suivant reconnaît les nombres décimaux :



— L'automate suivant reconnaît les littéraux décimaux en Python :



Autre solution moins compacte :



On peut aussi procéder facilement avec des unions et concaténations : un automate pour `pointfloat` (déjà l'union de 2 automates) ; un automate pour `exponent`, un automate pour `digit`$^+$, puis des unions et concaténations avec des ε-transitions, et enfin une union (en dupliquant `pointfloat`).