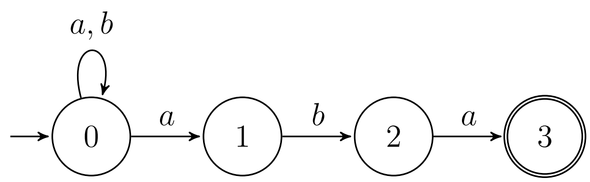
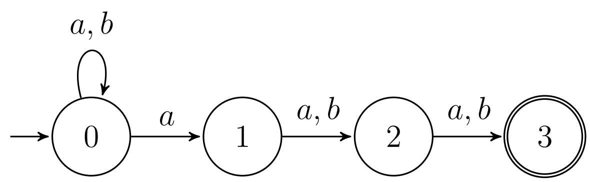
TD1 — 24 janvier 2019

Langages Formels

Exercice 1 (Déterminisation)

1. Pour chacun des automates suivants, déterminer le langage reconnu et donner un automate déterministe équivalent :





2. Si $\Sigma = \{a, b\}$, montrer que le langage $\Sigma^* a \Sigma^{n-1}$ ne peut être reconnu par un automate déterministe de moins de 2^n états.
3. Conclure que tout automate non-déterministe à n états admet un automate déterministe équivalent à au plus 2^n états, et que cette borne est "précise".

Exercice 2 (Construction d'automates)

On note $\Sigma = \{0, 1\}$.

1. Donner un automate qui reconnaît les langages :
   (a) $L_1 = \{u \in \Sigma^* :$ toute occurence de 1 est suivie de deux occurences de $0\}$
   (b) $L_2 = \{u \in \Sigma^* :$ u ne contient pas deux occurences successives de $0\}$
   (c) $L_3 = \{u \in \Sigma^* :$ le nombre d'occurences de 1 est pair$\}$

2. Donner un automate qui reconnaît le langage des multiples de 3 en base 2 où la représentation des entiers est "big-endian" (i.e. les bits sont rangés du plus fort au plus faible).
3. Donner un automate qui reconnaît les mots de Σ^* qui représentent les entiers non divisibles par 3 en notation "little-endian".

Exercice 3 (Des étoiles)
On considère les énoncés suivants, de plus en plus forts. Soit $L \in \mathrm{Rec}(\Sigma^*)$, il existe $N \geq 0$ tel que pour tout $x \in L$:

(a) Si $|x| \geq N$ alors il peut s'écrire $x = u_1u_2u_3$ avec $u_2 \neq \varepsilon$ et $u_1u_2^*u_3 \subseteq L$.

(b) Si $x = w_1w_2w_3$ avec $|w_2| \geq N$ alors $w_2 = u_1u_2u_3$ avec $u_2 \neq \varepsilon$ et $w_1u_1u_2^*u_3w_3 \subseteq L$.

(c) Si $x = uv_1v_2 \dots v_Nw$ avec $|v_i| \geq 1$ alors il existe $0 \leq j < k \leq N$ tel que $uv_1 \dots v_j(v_{j+1} \dots v_k)^*v_{k+1} \dots v_Nw \subseteq L$.

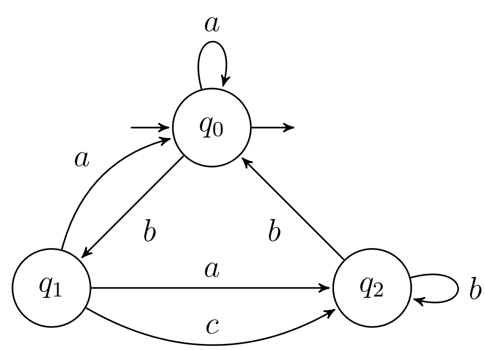
On sait que les langages réguliers satisfont ces trois propriétés.

1. Montrer que $\{\ u\ :\ u = \widetilde{u}\ \}$, c'est à dire le langage des palindromes, n'est pas régulier.
2. De même pour $\{\ (ab)^n(cd)^n\ \} \cup (\Sigma^*\{\ aa, bb, cc, dd, ac\ \}\Sigma^*)$.
3. Montrer que le langage suivant satisfait (c) mais n'est pas régulier : $\{\ udv\ :\ u, v \in \{a, b, c\}^*,\ u \neq v$ ou bien l'un des deux contient un carré $\}$. On pourra admettre l'existence d'une infinité de mots sans carrés sur l'alphabet $\{a, b, c\}$.

Exercice 4 (Algorithme de Brzozowski-McCluskey)
L'objectif de cet exercice est de traduire un automate fini en une expression rationnelle. Nous allons procéder par transformations successives de l'automate, en utilisant une généralisation du type d'automate considéré : la relation de transition sera un sous-ensemble de $Q \times 2^{\Sigma^*} \times Q$. Une exécution d'un tel automate reconnait la concaténation des langages des transitions, et le langage reconnu par l'automate est l'union de ces exécutions.

1. Montrer que tout automate généralisé est équivalent à un automate généralisé pour lequel il existe exactement une transition entre chaque paire d'états : $q' \in \delta(q, L)$ et $q' \in \delta(q, L')$ implique $L = L'$.
2. Soit un automate généralisé $\mathcal{A}$ avec un unique état initial i, un unique état final f. Soit $q \in Q_{\mathcal{A}}$, $q \notin \{i, f\}$. Montrer qu'il existe un automate équivalent à $\mathcal{A}$ avec pour ensemble d'états $Q_{\mathcal{A}} \setminus \{q\}$.

3. En déduire que si L est reconnu par un automate généralisé $\mathcal{A}$, alors L appartient à la clôture rationnelle des étiquettes des transitions de $\mathcal{A}$.
4. Appliquer la construction pour calculer une expression rationnelle correspondant à l'automate suivant :



5. On considère l'alphabet $\Sigma_n = \{1, \ldots, n\} \times \{1, \ldots, n\}$, et on définit :

$$L = \{ \ (a_1, a_2)(a_2, a_3) \ldots (a_m, a_{m+1}) \ : \ m \geq 1, \ a_1 = 1, \ a_{m+1} = n \ \}$$

   (a) Donner un automate de taille quadratique reconnaissant L.
   (b) Quelle est la taille de l'expression obtenue par la construction précédente sur cet automate ?

Exercice 5 (Dérivées partielles, automate d'Antimirov)
La dérivée partielle est aux expressions rationnelles ce que le résiduel est aux langages : on va définir une opération $\partial_a(E)$ qui va correspondre (modulo interprétation) à $a^{-1}\mathcal{L}(E)$.

Formellement, pour une expression E et une lettre a, on définit l'*ensemble d'expressions* $\partial_a(E)$ comme suit, où l'opération de concaténation est naturellement étendue sur les ensembles d'expressions :

$$\begin{array}{rcl}
\partial_a(\underline{\emptyset}) & = & \partial_a(\underline{\varepsilon}) \ = \ \emptyset \\
\partial_a(\underline{b}) & = & \left\{ \begin{array}{ll} \emptyset & \text{si } a \neq b \\ \{\underline{\varepsilon}\} & \text{sinon} \end{array} \right. \\
\partial_a(E + E') & = & \partial_a(E) \cup \partial_a(E') \\
\partial_a(E^*) & = & \partial_a(E) \cdot \{E^*\} \\
\partial_a(E \cdot E') & = & \left\{ \begin{array}{ll} \partial_a(E) \cdot \{E'\} & \text{si } \varepsilon \notin E \\ (\partial_a(E) \cdot \{E'\}) \cup \partial_a(E') & \text{sinon} \end{array} \right.
\end{array}$$

On définit ensuite $\partial_w(E)$ pour un mot w quelconque en posant $\partial_\varepsilon(E) = \{E\}$ et $\partial_{wa}(E) = \partial_a(\partial_w(E))$, où $\partial_w(S) = \bigcup_{E \in S} \partial_w(E)$ quand S est un ensemble d'expressions.

Étant donné un ensemble d'expression régulières S, on note $\mathcal{L}(S) = \bigcup_{E \in S} \mathcal{L}(E)$.

1. Calculer les dérivées partielles de $(ab + b)^*ba$ par a et b.
2. On note $w^{-1}L = \{u \in \Sigma^* \mid wu \in L\}$ le quotient à gauche de $L \subseteq \Sigma^*$ par $w \in \Sigma^*$.

   Montrer que pour tous $L, L' \subseteq \Sigma^*$ et $a \in \Sigma$,

$$
\begin{aligned}
a^{-1}(L \cup L') &= (a^{-1}L) \cup (a^{-1}L') \\
a^{-1}L^* &= (a^{-1}L) \cdot L^* \\
a^{-1}(L \cdot L') &= \begin{cases} a^{-1}L \cdot L' & \text{si } \varepsilon \notin L \\ (a^{-1}L \cdot L') \cup (a^{-1}L') & \text{sinon} \end{cases}
\end{aligned}
$$

3. Montrer que $\mathcal{L}(\partial_w(E)) = w^{-1}\mathcal{L}(E)$.
4. Utiliser cette construction pour traduire une expression rationnelle en un automate acceptant le même langage, en ne s'inquiétant pas de la finitude de cet automate pour l'instant.

   Appliquer cette construction à l'expression $(ab + b)^*ba$.
5. On définit l'ensemble des suffixes non vides d'un mot :

$$
\mathrm{Suf}(w) = \{\ v \in \Sigma^+ \ :\ \exists u, w = uv\ \}
$$

   Montrer les résultats suivants pour $w \in \Sigma^+$:

$$
\begin{aligned}
\partial_w(E + E') &= \partial_w(E) \cup \partial_w(E') \\
\partial_w(E \cdot E') &\subseteq (\partial_w(E) \cdot E') \cup \bigcup_{v \in \mathrm{Suf}(w)} \partial_v(E') \\
\partial_w(E^*) &\subseteq \bigcup_{v \in \mathrm{Suf}(w)} \partial_v(E) \cdot E^*
\end{aligned}
$$

6. Soit $\|E\|$ le nombre d'occurrences de lettres de Σ dans E. Montrer que l'ensemble des dérivées partielles différentes de E contient au plus $\|E\| + 1$ éléments.