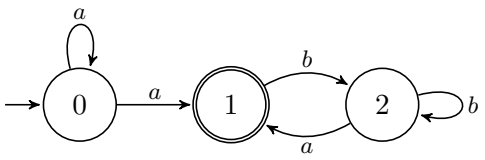
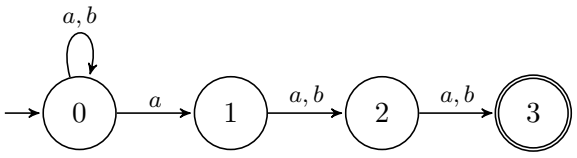
TD 1 : Langages reconnaissables / rationnels

Exercice 1 (Déterminisation)**.**

1. Donner un automate fini déterministe équivalent à l'automate suivant :



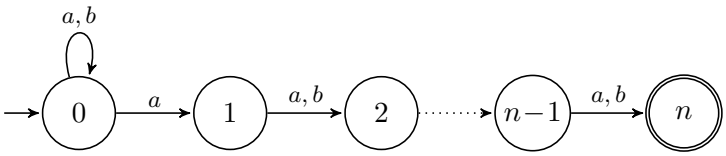
2. Donner une automate fini déterministe équivalent à l'automate suivant :



3. Montrer que l'automate des parties déterministe obtenu à partir de l'automate suivant a 2^n états. On pourra utiliser l'ensemble

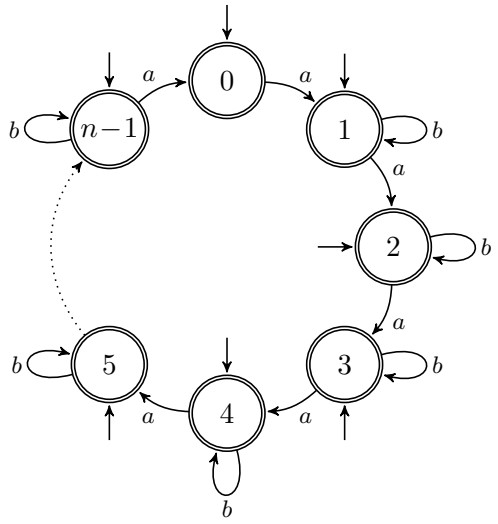
$$P-1 = \{i-1 \mid i \neq 1 \in P\}$$

où P est un sous-ensemble de $\{1, \ldots, n\}$.



Exercice 2 (Fermeture des langages reconnaissables)**.**

1. Montrer que les langages reconnaissables sont fermés par union, concaténation, étoile de Kleene, intersection, complément, et homomorphisme.
2. Montrer que si $L \subseteq \Sigma^*$ est reconnaissable et σ est une substitution de Δ^* dans les parties reconnaissables de Σ^*, alors $\sigma^{-1}(L)$ est reconnaissable sur Δ^*.
3. Quelle est la taille de l'automate des parties pour l'automate $\mathcal{A}_n$ suivant ? Montrer que tout automate reconnaissant $\overline{L(\mathcal{A}_n)}$ possède au moins $2^n - 2$ états.



Exercice 3 (Critères de reconnaissabilité)**.** Des conditions nécessaires pour qu'un langage soit reconnaissable sont connues sous le nom de « lemme de l'étoile » ou lemmes d'itération :

Lemme 1 (Lemme de l'étoile)**.** *Si L est un langage reconnaissable de Σ^*, alors il existe un entier n tel que*

1. *pour tout mot w de L avec $|w| \geq n$, il existe une factorisation $w = u_1u_2u_3$ avec u_2 non vide, telle que $u_1u_2^*u_3 \subseteq L$,*
2. *pour tout mot w de L et pour toute factorisation de $w = v_1v_2v_3$ avec $|v_2| \geq n$, il existe une factorisation $v_2 = u_1u_2u_3$ avec u_2 non vide, telle que $v_1u_1u_2^*u_3v_3 \subseteq L$,*
3. *pour tout mot w de L et pour toute factorisation $w = u_0u_1 \cdots u_nu_{n+1}$ avec u_i non vide pour i de 1 à n, il existe deux positions $0 \leq j < k \leq n$ telles que $u_0u_1 \cdots u_j(u_{j+1} \cdots u_k)^*u_{k+1} \cdots u_nu_{n+1} \subseteq L$.*

1. Démontrez le lemme de l'étoile.
2. Montrez que $\{a^nb^n \mid n \geq 0\}$ n'est pas reconnaissable sur $\{a, b\}^*$.
3. Soit L un langage quelconque sur Σ et $\#$ un symbole qui n'est pas dans Σ.
   (a) Montrer que le langage $L_\# = (\#^+L) \cup \Sigma^*$ satisfait les conditions de la première version du lemme de l'étoile.
   (b) Montrer à l'aide de propriétés de clôture que si L n'est pas reconnaissable, alors $L_\#$ n'est pas reconnaissable.
   (c) Montrer que si L ne satisfait pas les conditions de la première version du lemme de l'étoile, alors $L_\#$ ne satisfait pas les conditions de la seconde version du lemme.

4. Soit L un langage quelconque sur Σ et \$ un symbole qui n'est pas dans Σ. On définit

$$L_1 = \{\$^+ a_1 \$^+ a_2 \$^+ \dots \$^+ a_n \$^+ \mid n \geq 0, \forall i\ a_i \in \Sigma, \text{ et } w = a_1 \dots a_n \in L\}$$
$$L_2 = \{\$^+ v_1 \$^+ v_2 \$^+ \dots \$^+ v_n \$^+ \mid n \geq 0, \forall i\ v_i \in \Sigma^* \text{ et } \exists j\ |v_j| > 1\}$$
$$L_\$ = L_1 \cup L_2\ .$$

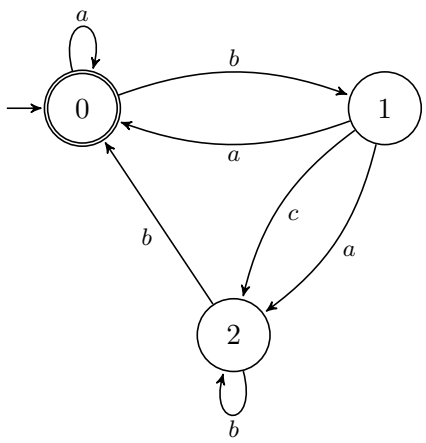
   (a) Montrer que $L_\$$ satisfait les conditions de la seconde version du lemme de l'étoile.

   (b) Montrer à l'aide de propriétés de clôture que si L n'est pas reconnaissable, alors $L_\$$ non plus.

   (c) Montrer que si L ne satisfait pas les conditions de la troisième version du lemme de l'étoile, alors $L_\$$ ne les vérifie pas non plus.

Exercice 4 (Algorithme de Brzozowski et McCluskey)**.** Un automate *généralisé* sur l'alphabet Σ utilise une relation de transition sur $Q \times 2^{\Sigma^*} \times Q$. Une exécution dans un tel automate reconnaît la concaténation des langages des transitions, et le langage reconnu par l'automate est l'union de ces exécutions.

1. Montrer que si $\mathcal{A}$ est un automate généralisé, alors on peut construire un automate généralisé $\mathcal{B}$ équivalent tel qu'il existe exactement une transition entre chaque paire d'états de $\mathcal{B}$.
2. Montrer que si $\mathcal{A} = \langle Q \uplus \{q, i, f\}, \delta, \{i\}, \{f\} \rangle$ est un automate généralisé sur Σ, alors il existe un automate généralisé équivalent avec pour ensemble d'états $Q \uplus \{i, f\}$, c'est-à-dire que l'on peut éliminer q de l'ensemble des états de $\mathcal{A}$.
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 cette construction au calcul d'une expression rationnelle équivalente à l'automate suivant :



5. On considère l'alphabet $\Sigma_n = \{1, \ldots, n\} \times \{1, \ldots, n\}$ des paires sur $\{1, \ldots, n\}$. Pour une paire (i, f) de Σ_n, on définit

$$L_{i,f} = \{(a_1, a_2)(a_2, a_3), \ldots, (a_m, a_{m+1}) \in \Sigma_n^m \mid m \geq 1, a_1 = i, a_{m+1} = f\}$$

l'ensemble des chemins de i à f dans le graphe complet défini par Σ_n.

   (a) Montrer que $L_{i,f}$ est reconnu par un automate fini de taille $O(n^2)$.

   (b) Quelle est la taille de l'expression rationnelle que vous obtenez à partir de cet automate ?

Exercice 5 (Langage local, automate de Glushkov)**.**

1. Une expression rationnelle E sur Σ est *linéaire* si chaque symbole de Σ apparaît au plus une fois dans E.

   Montrer que tout langage rationnel sur Σ est le résultat de l'application d'un morphisme alphabétique $\Delta \to \Sigma$ au langage d'une expression rationnelle linéaire sur Δ.

2. Soit L un langage sur Σ. On définit les ensembles

$$P(L) = \{a \in \Sigma \mid a\Sigma^* \cap L \neq \emptyset\} \qquad \text{(1-préfixes)}$$
$$S(L) = \{a \in \Sigma \mid \Sigma^* a \cap L \neq \emptyset\} \qquad \text{(1-suffixes)}$$
$$F(L) = \{u \in \Sigma^2 \mid \Sigma^* u \Sigma^* \cap L \neq \emptyset\} \qquad \text{(2-facteurs)}$$
$$N(L) = \Sigma^2 \backslash F(L) \qquad \text{(2-non facteurs)}$$

   Soit E une expression rationnelle telle que $\mathcal{L}(E) = L$. Donner un algorithme pour calculer $P(L)$, $S(L)$ et $F(L)$ à partir de E.

3. Un langage L sur Σ est *local* si

$$L \backslash \{\varepsilon\} = (P(L)\Sigma^* \cap \Sigma^* S(L)) \backslash \Sigma^* N(L) \Sigma^* \ .$$

   Un automate fini déterministe est *local* si, pour chaque symbole a de Σ, il y a au plus un état accessible par une transition sur a : $|\{q' \in Q \mid \exists q \in Q, (q, a, q') \in \delta\}| =\leq 1$. Il est *standard* si son état initial n'a aucune transition entrante.

   Montrer que les trois propositions suivantes sont équivalentes :

   (a) L est un langage local.

   (b) L est reconnu par un automate local standard.

   (c) L est reconnu par un automate local.

4. Les langages locaux sont clos par plusieurs opérations. Montrer que

   (a) si L est local, alors L^* est local.

   (b) si L_1 et L_2 sont des langages locaux sur des aphabets disjoints, alors $L_1 \cup L_2$ et $L_1 \cdot L_2$ sont des langages locaux.

Montrer qu'une expression linéaire représente un langage local.

5. En déduire un algorithme de contruction d'un automate équivalent à une expression rationnelle. Quelle est la taille de l'automate obtenu ?
6. Appliquer cet algorithme à l'expression rationnelle $(ab+b)^*ba$.
7. Soit $\Sigma_n = \{1, \ldots, n\}$ un alphabet et L l'ensemble des sous-mots de $1 \cdots n$. Donner une expression rationnelle pour L, et construire son automate de Glushkov.

Exercice 6 (Dérivées partielles, automate d'Antimirov)**.** La *dérivée partielle* $\partial_a(E)$ d'une expression rationnelle E sur Σ par une lettre a de Σ est l'ensemble d'expressions rationnelles sur Σ défini par

$$\begin{aligned}
\partial_a(\underline{\emptyset}) &= \emptyset \\
\partial_a(\underline{\varepsilon}) &= \emptyset \\
\partial_a(\underline{b}) &= \begin{cases} \{\underline{\varepsilon}\} & \text{si } a = b \\ \emptyset & \text{sinon} \end{cases} \\
\partial_a(E+F) &= \partial_a(E) \cup \partial_a(F) \\
\partial_a(E^*) &= \partial_a(E) \cdot E^* \\
\partial_a(EF) &= \begin{cases} \partial_a(E) \cdot F & \text{si } \varepsilon \notin \mathcal{L}(E) \\ \partial_a(E) \cdot F \cup \partial_a(F) & \text{sinon} \end{cases}
\end{aligned}$$

où l'opération de concaténation est étendue de manière évidente aux ensembles d'expressions rationnelles.

On étend cette définition à des mots w de Σ^* et des ensembles d'expressions rationnelles S par

$$\begin{aligned}
\partial_\varepsilon(E) &= \{E\} \\
\partial_{wa}(E) &= \partial_a(\partial_w(E)) \\
\partial_w(S) &= \bigcup_{E \in S} \partial_w(E) \ .
\end{aligned}$$

1. Calculer les dérivées partielles de $(ab+b)^*ba$ par a et b.
2. Montrer que pour toute expression rationnelle E sur Σ et tout mot w de Σ^*, $\mathcal{L}(\partial_w(E)) = w^{-1}\mathcal{L}(E)$.
3. Utiliser les dérivées partielles pour construire un automate (que l'on montrera fini plus tard) équivalent à une expression rationnelle.
4. On note l'ensemble des suffixes propres d'un mot w par

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

Montrer les égalités et inégalités suivantes pour tout mot w de Σ^* et expressions E et F sur Σ :

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

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