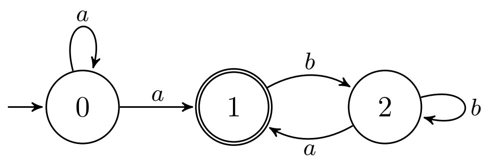
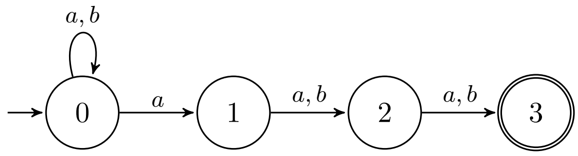
TD 1 : Langages rationnels

Exercice 1 (Déterminisation).

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



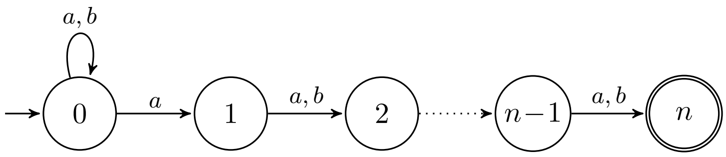
2. Donner une automate fini déterministe équivalent à l'automate suivant :



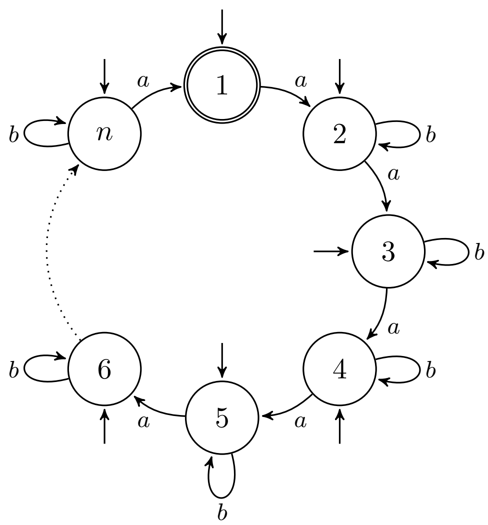
3. Montrer que l'automate des parties déterministe obtenu à partir de l'automate suivant a 2^n états. On pourra utiliser l'ensemble

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

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



4. Quelle est la taille de l'automate des parties pour l'automate suivant ?



Exercice 2 (Fermeture par substitution rationnelle inverse)**.** 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^*.

Exercice 3 (Critères de rationnalité)**.**

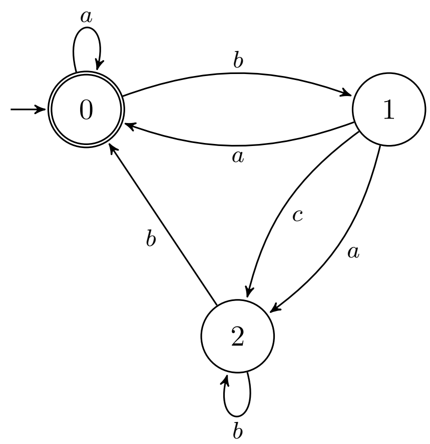
1. Soit L un langage quelconque sur Σ et $\#$ un symbole qui n'est pas dans Σ.
   (a) Montrer que le langage $L_\# = (\#^+ L) \cup \Sigma^*$ satisfait les conditions de la première version du lemme de l'étoile.
   (b) Montrer à l'aide de propriétés de clôture que si L n'est pas rationnel, alors $L_\#$ n'est pas rationnel.
   (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.
2. Soit L un langage quelconque sur Σ et \$ un symbole qui n'est pas dans Σ. On définit

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

   (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 rationnel, 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.
3. Montrer que l'ensemble des résiduels du langage $\{a^n b^n \mid n \geq 0\}$ n'est pas fini.

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 au plus une transition entre deux 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

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

   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 fini équivalent à une expression rationnelle.

4. On note l'ensemble des suffixes propres d'un mot w par

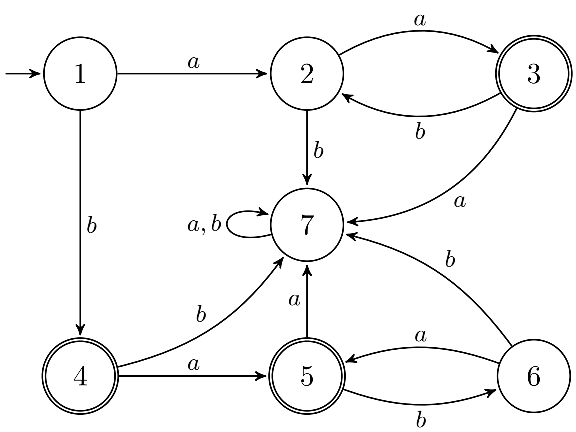
$$\mathrm{Suf}(w) = \{v \in \Sigma^+ \mid \exists u \in \Sigma^*, w = uv\} \ .$$

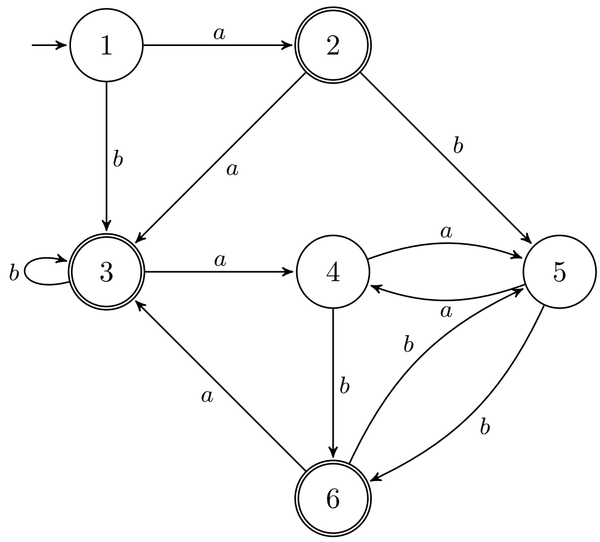
   Montrer les égalités et inégalités suivantes pour tout mot w de Σ^* et expressions E et F sur Σ :

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

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.

Exercice 7 (Minimisations)**.** Minimiser les automates suivants :





Exercice 8 (Minimisation par renversement)**.** Montrer que l'automate déterminisé d'un automate co-déterministe et co-accessible est minimal. Quelle est la complexité de cette méthode de minimisation ?

Exercice 9 (Algorithme d'Hopcroft)**.**

1. On considère l'algorithme de raffinement de partition suivant :
   Algorithme 1.
   Partitionne $(A = \langle Q, \delta, i, F \rangle)$
   $Q/\!\sim\ \leftarrow \{F, Q \backslash F\}$
   tant que $\exists B, C \in Q/\!\sim, a \in \Sigma.(\delta(B,a) \cap C \neq \emptyset \wedge \delta(B,a) \not\subseteq C)$ **faire**
       Raffine $(Q/\!\sim, B, C, a)$
   fin tant que
   retourner $Q/\!\sim$

   Raffine $(Q/\!\sim, B, C, a)$
   $B_1 \leftarrow B \cap \delta^{-1}(C, a)$
   $B_2 \leftarrow B \setminus B_1$
   $Q/\!\sim\ \leftarrow (Q/\!\sim \setminus \{B\}) \cup \{B_1, B_2\}$
   retourner (B_1, B_2)

   Dans cet algorithme, l'ensemble C sert à raffiner le bloc B par rapport à la lettre a.

   (a) Montrer que la procédure de l'algorithme 1 termine après au plus $|Q|$ itérations.

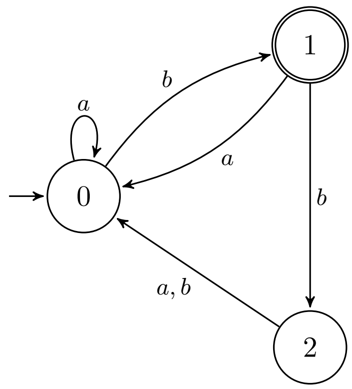
   (b) Montrer que l'algorithme 1 calcule la congruence de Nérode sur Q.

   (c) Donner une famille d'automates avec n états telle que cet algorithme effectue $\Omega(n^2)$ opérations.

2. Une première étape en direction de l'algorithme d'Hopcroft consiste à itérer sur un ensemble L de *raffineurs potentiels* (C, a) dans $2^Q \times \Sigma$.
   (a) Soit (C, a) un raffineur potentiel. Comment calculer l'ensemble des blocs B qui seraient *effectivement* raffinés par (C, a) ? On note $R_\sim(C, a)$ cet ensemble. Montrer qu'il peut être calculé en temps $O(\max(|C|, |\delta^{-1}(C, a)|)$.
   (b) Supposons que B soit raffiné par (C, a) en B_1 et B_2, et que D soit un sous-ensemble de B_1 ou de B_2. Le bloc D peut-il être effectivement raffiné par (C, a) ?
   (c) En déduire une reformulation simple de la procédure Partitionne qui itère sur un ensemble de raffineurs potentiels (et met cet ensemble à jour au fur et à mesure).
3. Pour obtenir une complexité en $O(n \log n)$, l'algorithme d'Hopcroft évite l'ajout de certains raffineurs redondants dans L.
   (a) Soit B un bloc raffiné en $B_1 \uplus B_2$. Montrer que, pour tout a de Σ, raffiner par rapport à (B, a) et (B_1, a) revient au même que raffiner par rapport à (B, a) et (B_2, a).
   (b) Donner une nouvelle version de l'algorithme qui travaille en $O(n \log n)$.

Exercice 10 (Reconnaissance par monoïde)**.**

1. Donner un monoïde fini M, un morphisme φ et une partie P de M qui permettent de reconnaître le langage accepté par l'automate suivant :



2. En utilisant la représentation par monoïdes, montrer que les langages reconnaissables sont clos par union, intersection et complémentaire.
3. En utilisant la représentation par monoïdes, montrer que les langages reconnaissables sont clos par concaténation. Si L_1 et L_2 sont reconnus par $\varphi_1 : \Sigma^* \to M_1$ et $\varphi_2 : \Sigma^* \to M_2$ respectivement, on pourra considérer

$$\psi : \quad \Sigma^* \to 2^{M_1 \times M_2}$$
$$w \mapsto \{(\varphi_1(u), \varphi_2(v)) \mid w = uv\} \, .$$

Exercice 11 (Langages apériodiques)**.**

1. Un automate fini déterministe complet a un *compteur* s'il existe un entier $n > 1$, une séquence d'états distincts $q_0, \ldots, q_{n-1}$ et un mot w de Σ^* tels que $\delta(q_i, w) = q_{i+1 \mod n}$ pour tout i de 0 à $n-1$.
   Montrer qu'un langage de Σ^* est apériodique si et seulement si son automate minimal n'a pas de compteur.
2. Montrer que si un langage est sans étoile, alors il est apériodique (utiliser les réponses de l'exercice précédent sur la reconnaissance par monoïde).
3. Pour chacun des langages suivants, montrer s'il est apériodique ou non :
   (a) $(ab)^*$,
   (b) $(aa)^*$,
   (c) $(a(ab)^*b)^*$,
   (d) $(ab + ba)^*$.