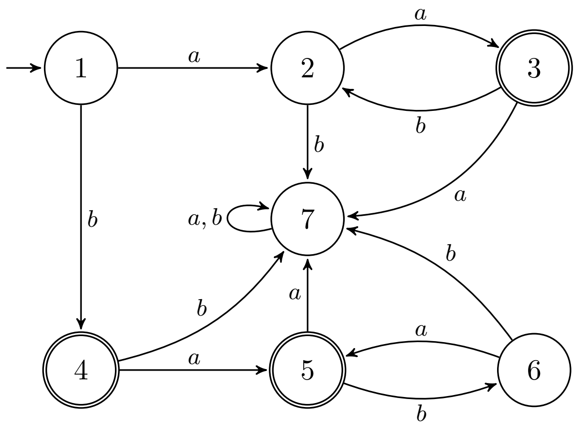
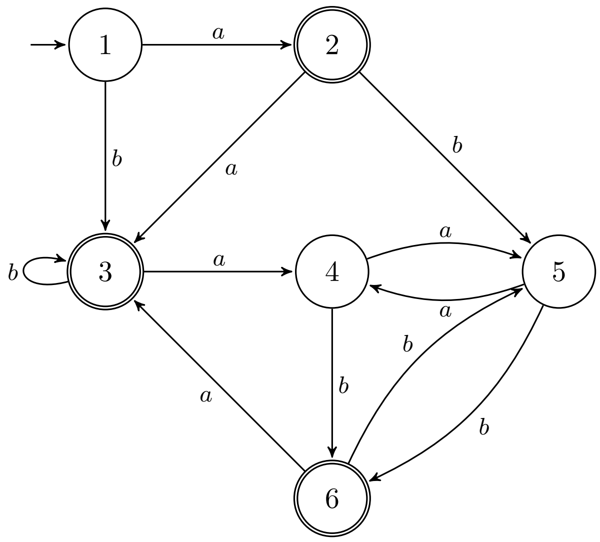
TD 2 : Minimisation, expressions rationnelles, MSO

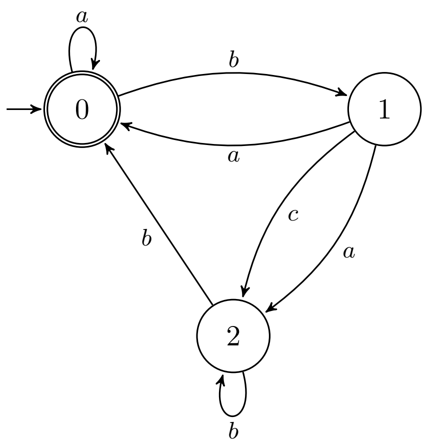
Exercice 1. (**À préparer** - Minimisations)
Minimiser les deux automates suivants :





Exercice 2. (**À préparer** - Automates $\leftrightarrow$ MSO)

1. Écrire une formule de la logique monadique du second ordre (MSO) pour l'automate ci-dessous.
2. Donner des automates équivalents à chacune des formules MSO suivantes :
   (a) $(P_a(x) \to (x \in Z \leftrightarrow y \in Z)) \vee (P_b(y) \to (y = x + 2 \wedge P_c(x)))$
   (b) $\exists x\,((P_a(x) \to (x \in Z \leftrightarrow y \in Z)) \vee (P_b(y) \to (y = x + 2 \wedge P_c(x))))$
   (c) $\forall x\,((P_a(x) \to (x \in Z \leftrightarrow y \in Z)) \vee (P_b(y) \to (y = x + 2 \wedge P_c(x))))$



Exercice 3. (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 4. (MSO et clôture transitive) Soit R une relation définissable en MSO sur les positions d'un mot, c'est-à-dire telle qu'il existe une formule $\varphi(x_1, x_2)$ de MSO avec deux variables libres x_1 et x_2 vérifiant : toute paire de positions (i, j) d'un mot w appartient à la relation R si, et seulement si, $w, x_1 \mapsto i, x_2 \mapsto j \models \varphi(x_1, x_2)$.

1. Écrire une formule de MSO pour la relation R_1 définie par : si la lettre courante est un a, sauter deux positions à droite ; si la lettre courante est un b, sauter une position à gauche ; si la lettre courante est un c, sauter trois positions à droite. Autrement dit, pour un mot w fixé, $R_1 = \{(i, j) \mid w_i = a \text{ et } j = i + 2\} \cup \{(i, j) \mid w_i = b \text{ et } j = i - 1\} \cup \{(i, j) \mid w_i = c \text{ et } j = i + 3\}$.
2. Écrire une formule de MSO pour la relation R_2 définie par : si la lettre courante est un a, sauter soit deux positions à gauche, soit une position à droite ; si la lettre courante est un b, sauter soit une position à gauche, soit trois positions à droite. Autrement dit, pour un mot w fixé, $R_2 = \{(i, j) \mid w_i = a \text{ et } j = i - 2 \text{ ou } j = i + 1\} \cup \{(i, j) \mid w_i = b \text{ et } j = i - 1 \text{ ou } j = i + 3\}$.
3. Écrire une formule de MSO $\psi_1(x_3, x_4)$ avec deux variables libres x_3 et x_4 qui définit une relation binaire R_1^* qui est la clôture réflexive et transitive de R_1.
4. Plus généralement, construire une formule de MSO définissant la clôture réflexive et transitive d'une relation définie en MSO.
5. Pour une relation R définissable en MSO, écrire une formule de MSO sans variable libre telle que la formule s'évalue à vrai sur un mot si, et seulement si, la paire donnée par la première position et la dernière position du mot appartient à la clôture réflexive et transitive de R.

Exercice 5. (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 de l'exercice 2.
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 6. (Langages locaux, algorithme 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

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

   Soit E une expression rationnelle telle que $L(E) = L$. 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

$$L \setminus \{\epsilon\} = (P(L)\Sigma^* \cap \Sigma^* S(L)) \setminus \Sigma^* N(L) \Sigma^*$$

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.