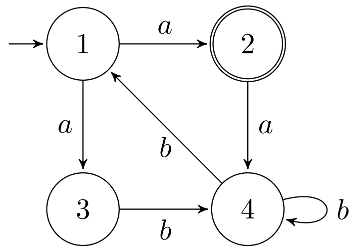
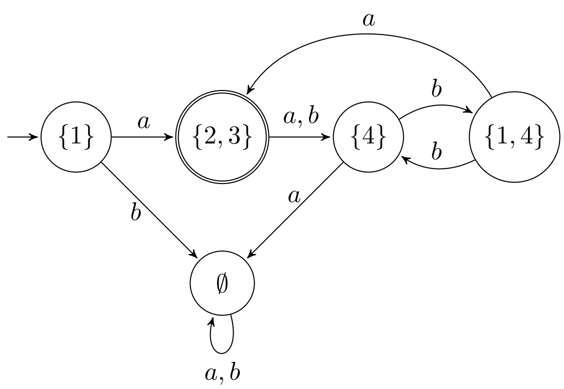
I Algorithme de déterminisation

Déterminiser l'automate suivant en utilisant l'algorithme du cours :



Solution :



II Clôture des langages reconnaissables

Si $m = m_1 ... m_n$ est un mot, on définit son miroir $\widetilde{m} = m_n ... m_1$. Si L est un langage, on définit son miroir $\widetilde{L} = \{\widetilde{m} \mid m \in L\}$.

1. Montrer que le miroir d'un langage reconnaissable est reconnaissable.

   Solution : Soit $A = (\Sigma, Q, I, F, E)$ un langage reconnaissant L. Alors $\widetilde{A} = (\Sigma, Q, F, I, \widetilde{E})$ reconnaît $\widetilde{A}$, où on a inversé toutes les transitions ($\widetilde{E} = \{(q_1, a, q_2) \mid (q_2, a, q_1) \in E\}$). En effet il existe un chemin $q_1 \in I \xrightarrow{m_1} q_2 \xrightarrow{m_2} ... \xrightarrow{m_k} q_{k+1} \in F$ dans A si et seulement si il existe un chemin $q_{k+1} \in F \xrightarrow{m_k} q_k \xrightarrow{m_{k-1}} ... \xrightarrow{m_1} q_1 \in I$ dans $\widetilde{A}$.

Si L est un langage sur Σ, on définit :

- $Pref(L) = \{u \in \Sigma^* \mid \exists v \in \Sigma^*,\ uv \in L\}$: ensemble des préfixes des mots de L.
- $Suff(L) = \{u \in \Sigma^* \mid \exists v \in \Sigma^*,\ vu \in L\}$: ensemble des suffixes des mots de L.
- $Fact(L) = \{u \in \Sigma^* \mid \exists v, w \in \Sigma^*,\ vuw \in L\}$: ensemble des facteurs des mots de L.

2. Montrer que si L est reconnaissable alors $Pref(L)$, $Suff(L)$, $Fact(L)$ le sont aussi.

   Solution : Soit $A = (\Sigma, Q, I, F, E)$ un langage reconnaissant L. Soit Q' l'ensemble des états co-accessibles et Q'' l'ensemble des états accessibles. Un mot m appartient à $Pref(L)$ si et seulement si il existe un chemin étiqueté par m d'un état de I vers un état de Q'. Donc (Σ, Q, I, Q', E) reconnaît $Pref(L)$. De même, (Σ, Q, Q'', F, E) reconnaît $Suff(L)$ et (Σ, Q, Q'', Q', E) reconnaît $Fact(L)$.

   Autre solution : après avoir démontré que $Pref(L)$ est reconnaissable, on peut remarquer que $Suff(L) = \widetilde{Pref(\widetilde{L})}$ ($m \in \widetilde{Pref(\widetilde{L})} \iff \widetilde{m} \in Pref(\widetilde{L}) \iff \exists v \in \Sigma^*,\ \widetilde{m}v \in \widetilde{L} \iff \exists v \in \Sigma^*,\ \widetilde{v}m \in L \iff m \in Suff(L)$, où on a utilisé le fait que $\widetilde{xy} = \widetilde{y}\,\widetilde{x}$).
   On peut aussi en déduire que $Fact(L)$ est reconnaissable en remarquant que $Fact(L) = Suff(Pref(L))$ $(= Pref(Suff(L)))$. En effet $m \in Suff(Pref(L)) \iff \exists u \in \Sigma^*,\ um \in Pref(L) \iff \exists u \in \Sigma^*,\ \exists v \in \Sigma^*,\ umv \in L \iff m \in Fact(L)$.

3. Montrer que si L est rationnel alors $Pref(L)$, $Suff(L)$, $Fact(L)$ le sont aussi (puisqu'on va montrer que rationnel = reconnaissable, c'est une preuve alternative à la précédente).

Solution : Soit e est une expression rationnelle dont le langage est L. On définit par induction une expression rationnelle $P(e)$ de langage $Pref(L)$:

- Si $e = a \in \Sigma$: $P(e) = \varepsilon + a$.
- Si $e = e_1 + e_2$: $P(e) = P(e_1) + P(e_2)$.
- Si $e = e_1 e_2$: $P(e) = P(e_1) + e_1 P(e_2)$.
- Si $e = e_1^*$: $P(e) = e_1^* P(e_1)$.

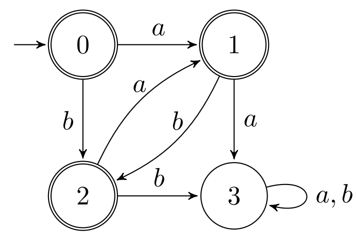
De même pour $Suff(L)$ et $Fact(L)$.

III Reconnaissable ou non ?

Pour chacun de ces langages, dire s'il est reconnaissable ou non. Justifier.

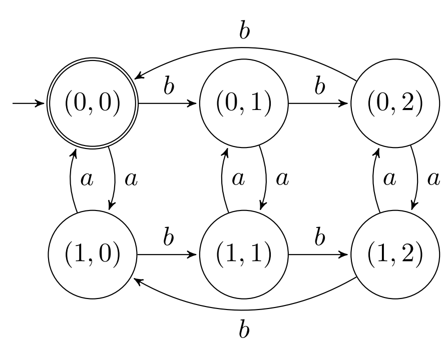
1. L_1 = mots sur $\{a, b\}$ sans lettres consécutives égales.

Solution : Idée : on se retrouve dans l'état 1 (respectivement 2) si la dernière lettre lue est un a (resp. b).



2. L_2 = mots sur $\{a, b\}$ ayant un nombre pair de a et dont le nombre de b est multiple de 3.

Solution : On peut construire un automate reconnaissant les mots ayant un nombre pair de a en utilisant 2 états (suivant la parité du nombre de a lus jusqu'à présent). De façon similaire, les mots ayant un nombre de b multiple de 3 peuvent être reconnus par un automate avec 3 états, pour chaque reste modulo 3 du nombre de b déjà lus.
Pour vérifier les deux en même temps, on peut utiliser l'automate « produit » vu en cours, pour reconnaître l'intersection des deux langages précédents. L'état nommé (i, j) correspond à la lecture d'un mot dont le nombre de a est i modulo 2 et le nombre de b est j modulo 3 :



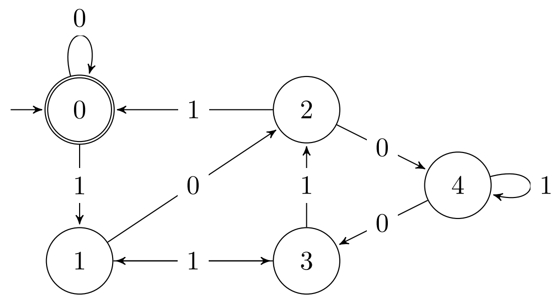
3. $L_3 = \{m \in \{a, b\}^* \mid |m|_a = |m|_b\}$ (où $|m|_a$ est le nombre de a du mot m).

Solution : Supposons que L_3 soit reconnaissable. Soit n l'entier donné par le lemme de l'étoile et $u = a^n b^n$.
$u \in L$ et $|u| \geq n$ donc, par le lemme de l'étoile, il existe des mots x, y, z tels que $|xy| \leq n$, $y \neq \varepsilon$ et $\forall k \in \mathbb{N}$, $xy^k z \in L$. Comme $|xy| \leq n$, y ne contient que des a. En prenant $k = 0$, on trouve que $xy^0 z = xz \in L_3$. Mais xz contient strictement plus de b que de a, ce qui est absurde.

4. L_4 = écritures en base 2 des multiples de 5.

Solution : On va construire un automate dont les états sont les restes possibles modulo 5. Si l'automate est dans un état q a déjà lu $n_1...n_p$ (c'est à dire $\overline{n_1...n_p}^2 \equiv q[5]$) et qu'il lit n_{p+1} alors il doit aller dans l'état $2q + n_{p+1}$ car $\overline{n_1...n_p n_{p+1}}^2 = 2 \times \overline{n_1...n_p}^2 + n_{p+1} \equiv 2q + n_{p+1}[5]$.

Ce langage est donc reconnu par l'automate $(\Sigma, Q, I, F, \delta) = (\{0,1\}, \{0,1,2,3,4\}, \{0\}, \{0\}, \delta)$ où $\delta(q,a) = 2q + a$ [5] :



5. $L_5 = \{a^p \mid p \text{ est un nombre premier}\}$.

Solution : Supposons que L_5 soit reconnaissable. Soit n l'entier donné par le lemme de l'étoile.
Soit $p \geq n+2$ un nombre premier et $u = a^p$.
$u \in L$ et $|u| \geq n$ donc, par le lemme de l'étoile, il existe i_1, i_2, i_3 tels que $i_1 + i_2 \leq n$, $i_2 > 0$ et $\forall k \in \mathbb{N}$, $a^{i_1}(a^{i_2})^k a^{i_3} = a^{i_1 + ki_2 + i_3} \in L$ (on a utilisé le fait que u ne contient que des a).
Soit $k = i_1 + i_3$ et $q = i_1 + ki_2 + i_3$. Alors, d'après le lemme de l'étoile, $a^q \in L$. Mais $q = (i_1 + i_3)(1 + i_2)$ avec $1 + i_2 > 1$ et $i_1 + i_3 \geq i_3 > 1$ (car $p \geq n+2$) donc q n'est pas premier, ce qui est absurde.

IV Longueur discriminante

1. Soit A un automate. Décrire un algorithme pour déterminer la plus petite longueur d'un mot reconnu par A et préciser sa complexité.

2. Soit A un automate à n états et de langage $L(A)$. Montrer que $L(A) = \emptyset$ si et seulement si $L(A)$ ne contient aucun mot de longueur strictement inférieure à n.

3. Soit $A_1 = (Q_1, i_1, F_1, \delta_1)$ et $A_2 = (Q_2, i_2, F_2, \delta_2)$ deux automates déterministes complets à n_1 et n_2 états et de langages L_1 et L_2. On suppose que $L_1 \neq L_2$. Soit $l(L_1, L_2)$ la plus petite longueur d'un mot u appartenant à l'un des deux langages mais pas à l'autre.
Montrer que $l(L_1, L_2) < n_1 n_2$.

V Ensemble distingant

Soient L un langage sur un alphabet Σ et $u, v \in \Sigma^*$. On dit que $w \in \Sigma^*$ est un *suffixe distingant* pour u et v si exactement l'un des mots uw ou vw appartient à L.

Un ensemble de mots D est *distingant* pour L si toute paire de mots de D a un suffixe distingant.

1. Soit L_1 le langage dénoté par l'expression régulière $(ab)^*$. Montrer que $\{\varepsilon, a, b\}$ est un ensemble distingant pour L_1.

2. On note $ind(L)$ le nombre minimum d'états d'un automate déterministe complet reconnaissant L. Montrer que si L a un ensemble distingant de taille n alors $ind(L) \geqslant n$.

3. Que vaut $ind(L_1)$?

4. On suppose que L a un ensemble distingant infini. Montrer que L n'est pas un langage régulier.

5. En déduire que $\{a^n b^n \mid n \in \mathbb{N}\}$ n'est pas un langage régulier.

6. Soit L_2 l'ensemble des mots de $\{a, b\}^*$ qui contiennent un nombre pair de a et un nombre pair de b. Déterminer $ind(L_2)$.

VI Oral ENS info

On fixe un alphabet Σ avec $|\Sigma| > 1$. Un mot $w \in \Sigma^*$ est un palindrome s'il s'écrit $w = a_1 \cdots a_n$ et qu'on a $a_i = a_{n-i+1}$ pour tout $1 \leq i \leq n$. On note $\Pi \subseteq \Sigma^*$ le langage des palindromes. Pour un automate fini A sur Σ, on note $L(A)$ le langage reconnu par A.

1. Soit $\Pi_n := \Pi \cap \Sigma^n$. Montrer que pour tout automate fini déterministe complet A, pour tout $n \in \mathbb{N}$, si $L(A) \cap \Sigma^{2n} = \Pi_{2n}$, alors A a au moins $|\Sigma|^n$ états.

2. En déduire que le langage Π n'est pas régulier.

3. Étant donné un automate fini A sur Σ, peut-on calculer un automate A_Π qui reconnaisse $L(A) \cap \Pi$?

4. Pour tout mot $u = b_1 \cdots b_m$ de Σ^*, on note $\overline{u} := b_m \cdots b_1$ son miroir. Étant donné A, peut-on calculer un automate A'_Π qui reconnaisse $\{u \in \Sigma^* \mid u\overline{u} \in L(A)\}$?

5. On appelle Π_{pair} l'ensemble des palindromes de longueur paire, i.e., $\Pi_{\text{pair}} = \bigcup_{n \in \mathbb{N}} \Pi_{2n}$. Proposer un algorithme qui, étant donné un automate fini A sur Σ, détermine si $L(A) \cap \Pi_{\text{pair}}$ est vide, fini, ou infini. Discuter de sa complexité en temps et en espace.

6. Modifier l'algorithme de la question 4 pour calculer la cardinalité de $L(A) \cap \Pi_{\text{pair}}$ quand cet ensemble est fini, en faisant l'hypothèse que l'automate d'entrée A est déterministe. Comment la complexité est-elle affectée?

7. Modifier l'algorithme des questions 4 et 5 pour qu'il s'applique à $L(A) \cap \Pi$.