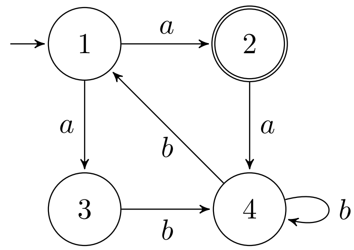
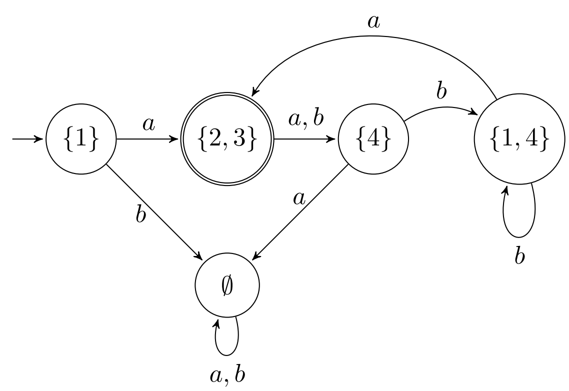
I Algorithme de déterminisation

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



Solution : En appliquant l'algorithme du cours, on obtient :



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 $\widetilde{L}$, 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 s'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. Donner des expressions régulières pour $Pref(a^*b)$ et $Pref((ab)^*)$.

   Solution : $Pref(a^*b) = a^*|a^*b$, $Pref((ab)^*) = (ab)^*|(ba)^*|(ab)^*a|(ba)^*b$.

3. 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 $m \in Suff(Pref(L)) \iff \exists u \in \Sigma^*,\ um \in Pref(L) \iff \exists u \in \Sigma^*,\ \exists v \in \Sigma^*,\ umv \in L \iff m \in Fact(L)$.

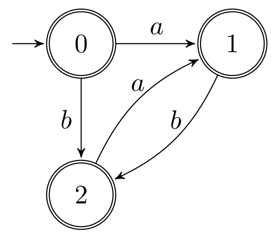
4. Montrer que si L est régulier alors $Pref(L)$, $Suff(L)$, $Fact(L)$ le sont aussi (puisqu'on va montrer que régulier = reconnaissable, c'est une preuve alternative à la précédente).

> Solution : Soit e est une expression régulière dont le langage est L. On définit par induction une expression régulière $P(e)$ de langage $Pref(L)$:
>
> - Si $a \in \Sigma$: $P(a) = \varepsilon | a$.
> - $P(\varepsilon) = \varepsilon$.
> - $P(\emptyset) = \emptyset$.
> - $P(e_1 | e_2) = P(e_1) | P(e_2)$.
> - $P(e_1 e_2) = P(e_1) | e_1 P(e_2)$.
> - $P(e_1^*) = e_1^* P(e_1)$.
>
> De même pour $Suff(L)$ et $Fact(L)$.

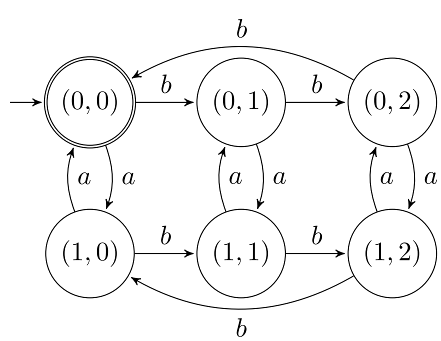
III Reconnaissable ou non ?

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 : 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. $L_3 = \{u \in \{a, b\}^* \mid |u|_a \mod 2 = |u|_b \mod 3\}$ (où $|m|_a$ est le nombre de a du mot m).

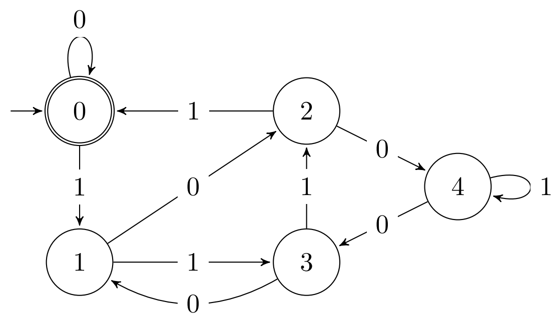
> Solution : Reprendre l'automate précédent en choisissant $(0, 0)$ et $(1, 1)$ comme états finaux.

4. $L_4 = \{m \in \{a, b\}^* \mid |m|_a = |m|_b\}$.

Solution : Supposons que L_4 soit reconnaissable. Soit n l'entier donné par le lemme de l'étoile et $u = a^n b^n$.
$u \in L_4$ 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.

5. L_5 = 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 $\overline{n_1 ... n_p n_{p+1}}^2 = 2 \times \overline{n_1 ... n_p}^2 + n_{p+1} \equiv 2q + n_{p+1}[5]$.
L_5 est donc reconnu par l'automate $(\Sigma, Q, I, F, \delta) = (\{0, 1\}, \{0, 1, 2, 3, 4\}, \{0\}, \{0\}, \delta)$ où $\delta(q, a) = 2q + a \ [5]$:



6. $L_6 = \{a^p \mid p$ est un nombre premier$\}$.

Solution : Supposons que L_6 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 $\forall k \in \mathbb{N}$, $a^{i_1}(a^{i_2})^k a^{i_3} = a^{i_1 + k i_2 + i_3} \in L$ (on a utilisé le fait que u ne contient que des a).
Soit $k = i_1 + i_3$ et $q = i_1 + k i_2 + i_3$. 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 Algorithmes sur les automates

1. À quelle condition nécessaire et suffisante simple le langage reconnu par un automate est vide ? Décrire un algorithme pour le savoir.

Solution : Il est vide si et seulement si il n'existe pas de chemin d'un état initial vers un état final. On peut le décider en effectuant un parcours du graphe de l'automate.

2. À quelle condition nécessaire et suffisante simple le langage reconnu par un automate est fini ? Décrire un algorithme pour le savoir.

Solution : Il est fini si et seulement si son graphe n'a pas de cycle. Nous avons vu un algorithme pour cela dans le cours sur les graphes.

3. Décrire un algorithme pour déterminer si deux automates admettent le même langage.

Solution : Soient $A_1 = (\Sigma, Q_1, i_1, F_1, \delta_1)$ et $A_2 = (\Sigma, Q_2, i_2, F_2, \delta_2)$. On peut construire des automates A et A' reconnaissant $L(A_1) \backslash L(A_2)$ et $L(A_2) \backslash L(A_1)$, en utilisant « l'automate produit » décrit dans le cours. Il suffit alors de déterminer si $L(A) = \emptyset$ et $L(A') = \emptyset$, en utilisant la question 3.