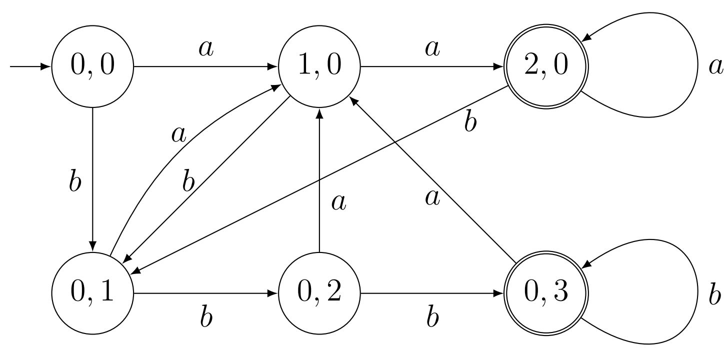
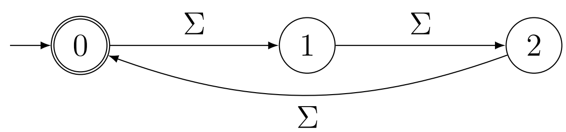
1 Automates finis déterministes

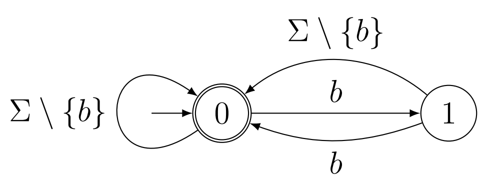
1) a) mots terminant par a^2 ou b^3, pour s'aider, on code les dernières lettres lues (nbre de a,nbre de b) :



b) mots de longueur un multiple de 3 :



c) mots terminant par un nombre pair de b :



2) a)

```
let a = {alphabet = ['a';'b'] ;
         nbre_etats = 3;
         transitions = (let t i x = match i,x with
                                  |0,'a' -> 1
                                  |0, 'b' -> 2
                                  |1, 'a' -> -1
                                  |1, 'b' -> 0
                                  |2, 'a' -> 1
                                  |2, 'b' -> -1
                                  |_ -> -1
                             in t);
         etats_acceptation = [1;2]};;
```

b) Pour lire un mot dans l'automate on parcourt le mot et on regarde l'état courant. Si on peut lire le mot sans blocage, on regarde alors si l'état est dans la liste des états acceptants.

```
let rec appartient l x =
 match l with
  | [] -> false
  | t :: q -> (t = x) || appartient q x;;


let lecture a l =
 let rec avance li et = match li with
   | [] -> appartient a.etats_acceptation et
   | x :: q -> let newet = a.transitions et x
     in (newet <> (- 1)) && (avance q newet)
 in
 avance l 0;;
```

c) Pour compléter un automate on ajoute un état puit.

```
let completude a =
let n = a.nbre_etats in
let newtrans i x =
 if (i = n) || (a.transitions i x = - 1) then n else a.transitions i x
in
{alphabet = a.alphabet; nbre_etats = a.nbre_etats + 1;
 transitions = newtrans; etats_acceptation = a.etats_acceptation};;
```

d) Pour déterminer le complémentaire, **on complete d'abord l'automate**. ll suffit ensuite de créer la liste des états qui n'était pas acceptants.

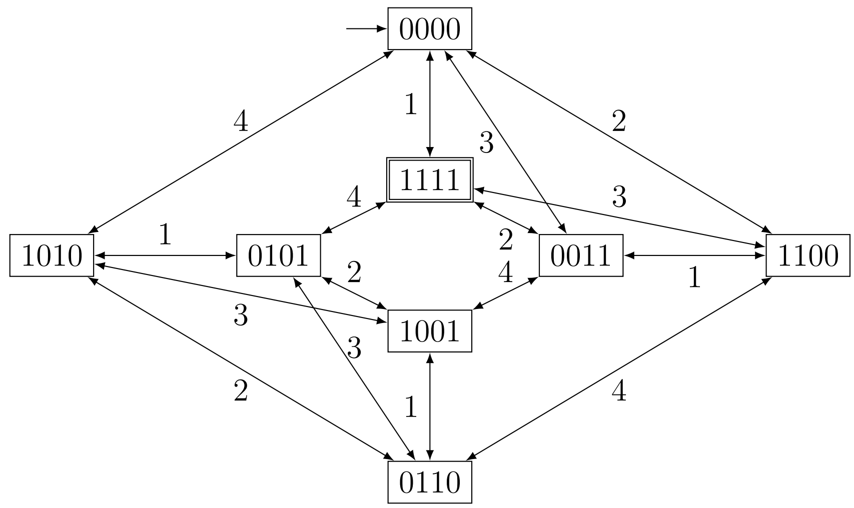
```
let complementaire a =
 let ac = completude a in
 let l = ref [] in
 for i = 0 to (ac.nbre_etats - 1) do
  if not (appartient ac.etats_acceptation i) then l := i :: !l
 done;
 {alphabet = ac.alphabet; nbre_etats = ac.nbre_etats;
  transitions = ac.transitions; etats_acceptation = !l};;
```

e) Pour savoir s'il existe un mot dans le langage, on parcourt l'automate et on regarde si un état acceptant est accessible.

```
let estNonVide a =
 let ac = completude a in
 let b = ref false in
 let visit = Array.make (ac.nbre_etats) false in
 let rec parcours etat =
  if not (visit.(etat)) then
   (visit.(etat) <- true;
    if appartient (ac.etats_acceptation) etat then b := true;
    parcours (ac.transitions etat 'a');
    parcours (ac.transitions etat 'b')
   )
 in
 parcours 0;
 !b;;
```

f) $(a|ba)((ba)^*(bba)^*)^*$

3) On dessine le graphe des états possibles reliés par les arêtes étiquetées par les interrupteurs qui permettent de passer d'un état à l'autre. On peut coder les configurations avec 4 bits : le premier représente l'état des projecteurs dont le numéro est de la forme $3k+1$ et pair, le second bit représente l'état des projecteurs dont le numéro n'est pas de la forme $3k+1$ mais est pair, le troisième bit représente l'état des projecteurs dont le numéro est de la forme $3k+1$ et impair, et le dernier bit représente l'état des projecteurs dont le numéro n'est pas de la forme $3k+1$ et est impair.

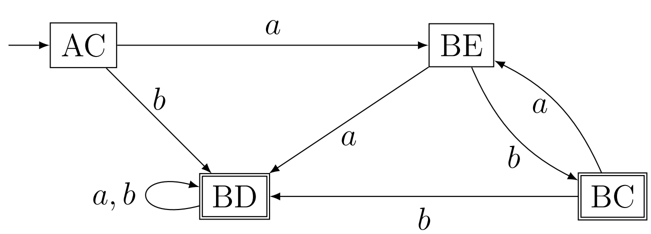


L'ensemble des combinaisons permettant d'éclairer tous les projecteurs en même temps, est exactement le langage reconnu par l'automate que l'on a défini, où l'état initial est l'état 0000 et l'état acceptant est l'état où tout est allumé, c'est-à-dire 1111.

4) a) i) $L_1 = (a|b).(a|b)^*$

ii) $L_2 = (ab)^*a$

b) i)



ii) $(L_1 \setminus L_2) \cup (L_2 \setminus L_1)$

c) Soit (s,t) un état de $\mathcal{A}_1 \oplus \mathcal{A}_2$, alors $(s,t) \in Q_1 \times Q_2$. Soit $a \in X$, alors comme $\mathcal{A}_1$ est complet, il existe $s' \in Q_1$ tq $\delta_1(s,a) = s'$. De même, comme $\mathcal{A}_2$ est complet, il existe $t' \in Q_2$ tq $\delta_2(t,a) = t'$. Par définition de $\mathcal{A}_1 \oplus \mathcal{A}_2$, on a alors $\delta_{\mathcal{A}_1 \oplus \mathcal{A}_2}((s,t),a) = (s',t')$. Il y a donc bien une arête sortante de l'état (s,t) pour la lettre a, et cela pour tout état (s,t) de $\mathcal{A}_1 \oplus \mathcal{A}_2$ et pour toute lettre de l'alphabet X. L'automate $\mathcal{A}_1 \oplus \mathcal{A}_2$ est donc bien complet.

d) Prouvons cette propriété par récurrence sur la longueur du mot m.

(I) Par définition, $\forall q_1 \in Q_1, \delta_1^*(q_1, \varepsilon) = q_1$, $\forall q_2 \in Q_2, \delta_2^*(q_2, \varepsilon) = q_2$ et $\forall q_1 \in Q_1, \forall q_2 \in Q_2, \delta^*((q_1,q_2),\varepsilon) = (q_1,q_2)$. Donc, $\forall q_1 \in Q_1, \forall q_2 \in Q_2$, on a bien $\delta^*((q_1,q_2),\varepsilon) = (q_1,q_2) = \delta_1^*(q_1,\varepsilon), \delta_2^*(q_2,\varepsilon)$.

(H) Soit $n \in \mathbb{N}$, supposons que la propriété est vraie pour tout mot de longueur n. Soit m un mot de longueur $n+1$ alors m est de la forme $m'.x$ où m' est un mot de longueur n et x une lettre de l'alphabet X.
Par définition, $\forall q_1 \in Q_1, \delta_1^*(q_1,\, m'.x) = \delta_1(\delta_1^*(q_1, m'),\, x)$, $\forall q_2 \in Q_2, \delta_2^*(q_2,\, m'.x) = \delta_2(\delta_2^*(q_2, m'),\, x)$ et $\forall q_1 \in Q_1, \forall q_2 \in Q_2, \delta^*((q_1,q_2),\, m'.x) = \delta(\delta^*((q_1,q_2), m'),\, x)$.

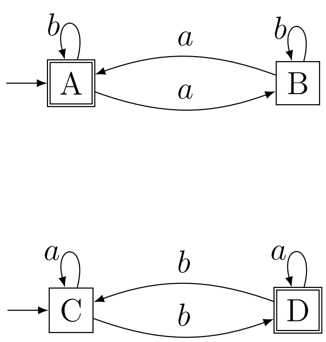
Or, par hypothèse de récurrence, $\delta^*((q_1,q_2),m') = \delta_1^*(q_1,m'),\delta_2^*(q_2,m')$. Ainsi, $\delta(\delta^*((q_1,q_2),m'),\ x) = \delta((\delta_1^*(q_1,m'),\delta_2^*(q_2,m')),\ x)$, et enfin par définition de δ, $\delta((\delta_1^*(q_1,m'),\delta_2^*(q_2,m')),\ x) = (\delta_1(\delta_1^*(q_1,m'),\ x),\delta_2(\delta_2^*(q_2,m'),\ x))$. En conséquence, on a bien $\delta^*((q_1,q_2),m') = (\delta_1^*(q_1,m'.x),\delta_2^*(q_1,m'.x))$.

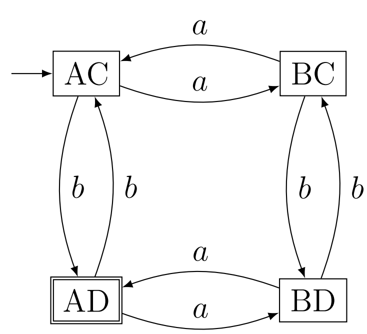
(C) La propriété est initialisée et héréditaire, donc par le principe de récurrence, elle est vraie pour tout mot m.

e) Un mot m est reconnu par $\mathcal{A}_1$ si et seulement si $\delta_1^*(i_1,m) \in T_1$, de même m est reconnu par $\mathcal{A}_2$ si et seulement si $\delta_2^*(i_2,m) \in T_2$. De plus, m est reconnu par $\mathcal{A}_1 \oplus \mathcal{A}_2$, si et seulement si $\delta^*((i_1,i_2),m) in (T_1 \times (Q_2 \setminus T_2)) \cup ((Q_1 \setminus T_1) \times T_2)$.

Or $\delta^*((i_1,i_2),m) = (\delta_1^*(i_1,m),\delta_2^*(i_2,m))$. Ainsi, m est reconnu par $\mathcal{A}_1 \oplus \mathcal{A}_2$ si et seulement s'il est "reconnu par $\mathcal{A}_1$ mais pas par $\mathcal{A}_2$" ou "reconnu par $\mathcal{A}_2$ mais pas par $\mathcal{A}_1$".

f) On peut construire l'automate $\mathcal{A}_1$ qui reconnaît l'ensemble des mots contenant un nombre pair de a et l'automatque $\mathcal{A}_2$ qui reconnaît l'ensemble des mots contenant un nombre impair de b. On peut alors construire le produit $\mathcal{A}_1 \oplus \mathcal{A}_2$. Pour obtenir un automate reconnaissant l'ensemble des mots souhaité, il suffit de modifier l'ensemble des états acceptants en prenant comme seul état acceptant le couple des deux états acceptants de $\mathcal{A}_1$ et $\mathcal{A}_2$. On a ainsi construit l'intersection des deux ensembles de mots.





5) Appuyons nous sur les notations de l'exercice précédent. Soient $\tilde{\mathcal{A}}_1$ et $\tilde{\mathcal{A}}_2$ les complétés de respectivement $\mathcal{A}_1$ et $\mathcal{A}_2$. Soit $\mathcal{A}$, l'automate $\tilde{\mathcal{A}}_1 \oplus \tilde{\mathcal{A}}_2$ dont les états acceptants sont définis comme suit : $(T_1 \times \tilde{Q}_2) \cup (\tilde{Q}_1 \times T_2)$ où $\tilde{Q}_1$ et $\tilde{Q}_2$ sont les ensembles des états de $\tilde{\mathcal{A}}_1$ et $\tilde{\mathcal{A}}_2$

Remarque : on n'introduit pas fait de nouvelle notation pour les ensembles des états acceptants de $\tilde{\mathcal{A}}_1$ et $\tilde{\mathcal{A}}_2$ car l'ensemble des états acceptants n'est pas modifié par l'opération de complétion.

Le nombre d'états est égal à $(n_1+1)*(n_2+1)$, ce qui est bien polynomial par rapport à n_1 et n_2. Les "+1" correspond aux puits ajoutés lors de la complétion.

6) a) Pour décider si $L_1 \subset L_2$, on peut construire à nouveau le produit des automates M_1 et M_2, cette fois, on définit les états acceptants comme suite : $T_1 \times (Q_2 \setminus T_2)$. Le langage de cet automate est vide si et seulement si $L_1 \subset L_2$.

b) $L_1 = L_2$ si et seulement si $L_1 \subset L_2$ et $L_2 \subset L_1$. Mais on peut également tester simplement la vacuité du langage de l'automate $M_1 \oplus M_2$.

Ces deux algorithmes sont bien polynomiaux car l'automate produit est de taille $n_1 * n_2$ où n_1 et n_2 sont le nombre d'états de M_1 et M_2.

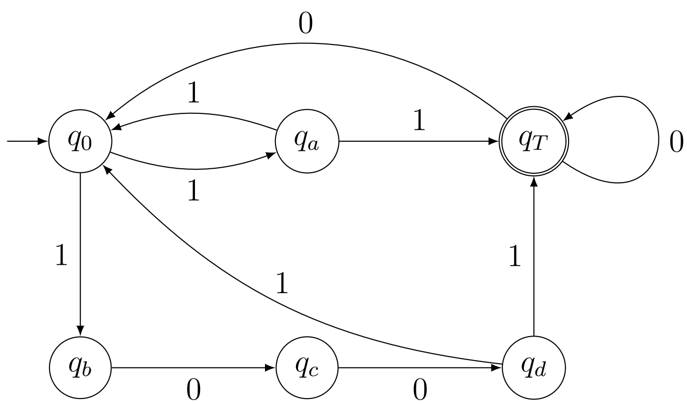
2 Automates non déterministes

7) Écriture d'un nombre en base 2

On définit sur l'alphabet $A = \{0, 1\}$ le langage R par les règles :

- $11 \in R$ et $1001 \in R$
- si $m \in R$ alors $m.0 \in R$
- si $m, m' \in R$ alors $m.m' \in R$

a) Considérons l'automate A suivant :



Justifions que le langage L reconnu par cet automate est le langage R qui est le plus petit langage satisfaisant les règles de l'énoncé.

- $R \subset L$. Montrons par récurrence le prédicat suivant :
  $\mathcal{P}_n$: "Tout mot m de R de longueur n appartient à L"
  - $\boxed{\text{I}}$ Pour $n = 0$ et $n = 1$, $\mathcal{P}_n$ est vrai car il n'y a pas de mots de longueur 0 ou 1 dans R. De même pour $n = 2$, $n = 3$ ou $n = 4$, le prédicat est vrai. Il suffit de vérifier tous les mots concernés : $11, 110, 1001, 1111$.
  - $\boxed{\text{H}}$ On procède par récurrence forte. Soit $n \geq 5$ on suppose $\mathcal{P}_k$ pour tout $k < n$. Montrons $\mathcal{P}_n$. Soit $m \in R$ tel que $|m| = n$. Comme $|m| > 4$, il y a deux cas : soit $m = w0$ où $w \in R$ soit $m = ww'$ où w et w' appartiennent à R. Dans le premier cas, en appliquant $\mathcal{P}_{n-1}$, $w \in L$ donc il existe un calcul réussi

$$q_0 \xrightarrow{w_1} \cdots \xrightarrow{w_{n-2}} q \xrightarrow{w_{n-1}} q_T$$

dans l'automate A sur w. On en déduit que l le calcul suivant dans A sur m

$$q_0 \xrightarrow{w_1} \cdots \xrightarrow{w_{n-2}} q \xrightarrow{w_{n-1}} q_T \xrightarrow{0} q_T$$

est aussi réussi. Cela montre que $m \in L$.
Dans le deuxième cas, on peut écrire $m = ww'$ où $|w|$ et $|w'|$ sont strictement inférieurs à n. De ce fait, par récurrence forte, ce sont des mots de L. Il existe donc des calculs sur ces deux mots réussis dans A

$$q_0 \xrightarrow{w_1} \cdots \xrightarrow{w_{k-1}} q \xrightarrow{w_k} q_T$$

et

$$q_0 \xrightarrow{w'_1} \cdots \xrightarrow{w'_{k'-1}} q' \xrightarrow{w'_{k'}} q_T$$

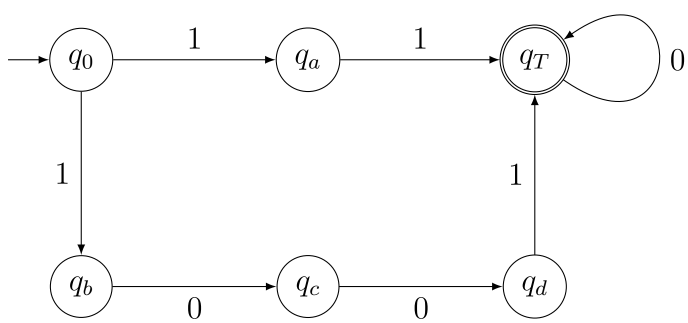
On remarque que l'avant dernier état q du calcul sur w est donc q_a, q_d ou q_T. On peut donc "remplacer" la transition $q \xrightarrow{w_k} q_T$ par $q \xrightarrow{w_k} q_0$. De ce fait, le calcul

$$q_0 \xrightarrow{w_1} \cdots \xrightarrow{w_{n-2}} q \xrightarrow{w_k} q_0 \xrightarrow{w'_1} \cdots \xrightarrow{w'_{n-2}} q' \xrightarrow{w'_{k'}} q_T$$

est un calcul réussi dans A sur $m = ww'$ et donc $m \in R$.

  – $\boxed{\text{C}}$ On a bien montré que pour tout entier naturel n, $\mathcal{P}_n$ et donc $R \subset L$.

- $L \subset R$. On considère l'ensemble Z des trois transitions qui aboutissent à q_0 : $q_a \xrightarrow{1} q_0$; $q_d \xrightarrow{1} q_0$ et $q_T \xrightarrow{0} q_0$. Montrons par récurrence le prédicat suivant : $\mathcal{P}_n$: "Tout mot m de L tel qu'il existe un calcul réussi sur m dans A empruntant n transitions dans Z est dans R"
  – $\boxed{\text{I}}$ Pour $n = 0$, un mot m de L tel qu'il existe un calcul réussi sur m dans A n'empruntant aucune transitions dans Z est dans R est donc reconnu par l'automate obtenu en enlevant les transitions de l'ensemble Z :



Le langage des mots reconnu par cet automate est $L((11|1001)0^*)$ qui est clairement inclus dans R.

  – $\boxed{\text{H}}$ Soit $n \geq 1$. On suppose $\mathcal{P}_{n-1}$. Soit m un mot de L tel qu'il existe un calcul réussi sur m dans A empruntant n transitions dans Z. Considérons $q \xrightarrow{m_r} q_0$ la première de ces n transitions. Le calcul total s'écrit alors :

$$q_0 \xrightarrow{m_1} \cdots q \xrightarrow{m_r} q_0 \xrightarrow{m_{r+1}} \cdots \xrightarrow{m_p} q_T$$

On peut alors remplacer $q \xrightarrow{m_r} q_0$ par $q \xrightarrow{m_r} q_T$ de telle sorte que

$$q_0 \xrightarrow{m_1} \cdots q \xrightarrow{m_r} q_T$$

est un calcul réussi dans A sans transitions dans Z. D'après $\mathcal{P}_0$, $m_1 \cdots m_r \in R$. De même,

$$q_0 \xrightarrow{m_{r+1}} \cdots \xrightarrow{m_p} q_T$$

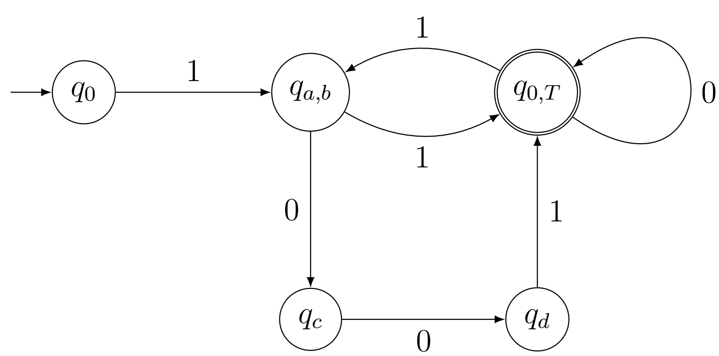
est un calcul réussi dans A avec $n-1$ transitions dans Z. D'après $\mathcal{P}_{n-1}$, $m_{r+1} \cdots m_p \in R$.
Finalement, $m = ww'$ où $w = m_1 \cdots m_r \in R$ et $w' = m_{r+1} \cdots m_p \in R$. Cela prouve que $m \in R$.

– $\boxed{\text{C}}$ On a bien montré que pour tout entier naturel n, $\mathcal{P}_n$ et donc $L \subset R$.

b) Déterminisons cet automate :

état	0	1
$\{q_0\}$	$\emptyset$	$\{q_a, q_b\}$
$\emptyset$	$\emptyset$	$\emptyset$
$\{q_a, q_b\}$	$\{q_c\}$	$\{q_0, q_T\}$
$\{q_c\}$	$\{q_d\}$	$\emptyset$
$\{q_0, q_T\}$	$\{q_0, q_T\}$	$\{q_a, q_b\}$
$\{q_d\}$	$\emptyset$	$\{q_0, q_T\}$

Cela donne (en enlevant l'état $\emptyset$ qui n'est pas coaccessible) l'automate



c) Soit $m = m_1 \cdots m_n$ on lui associe $\overline{m} = \sum_{i=1}^{n} m_i 2^{n-i}$.

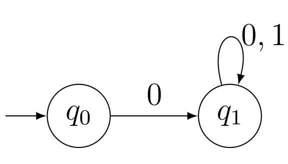
On procède par induction (ou par récurrence forte sur la longueur du mot).

- Cas de base. Si $m = 11$, $\overline{m} = 3$ qui est divisible par 3. Si $m = 1001$, $\overline{m} = 9$ qui est divisible par 3.
- Induction.
  Si $m = w0$ où $3|\overline{w}$ alors $\overline{m} = 2 \times \overline{w}$ et $3|\overline{m}$.
  Si $m = ww'$ où $3|\overline{w}$ et $3|\overline{w'}$ alors $\overline{m} = 2^{|w'|}\overline{w} + \overline{w'}$ et $3|\overline{m}$.

Donc pour tout $m \in R$, $3|\overline{m}$.

d) L'écriture binaire de 21 et 10101 qui n'appartient pas à R.

8) a)



b) i)

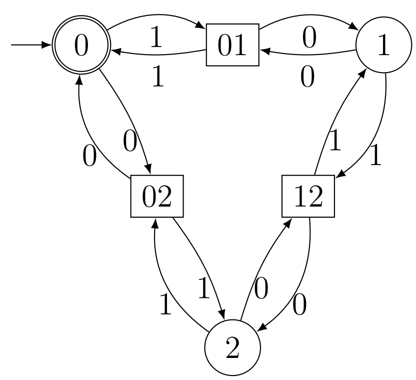
$$\begin{aligned} v(m) &= \sum_{k=0}^{2n-1} d_k 2^k = \sum_{k=0}^{n-1} d_{2k} 2^{2k} + \sum_{k=0}^{n-1} d_{2k+1} 2^{2k+1} \\ &= \sum_{k=0}^{n-1} d_{2k} 2^{2k} + 2 d_{2k+1} 2^{2k} = \sum_{k=0}^{n-1} (d_{2k} + 2 d_{2k+1}) 2^{2k} \\ &= \sum_{k=0}^{n-1} e_k 2^{2k} = \sum_{k=0}^{n-1} e_k 4^k \end{aligned}$$

ii) Tout d'abord, $4 \equiv 1 \mod 3$, ainsi pour tout $k \in \mathbb{N}^*, 4^k \equiv 1^k \mod 3$, c'est-à-dire $4^k \equiv 1 \mod 3$. De cette façon, pour tout $a \in \mathbb{N}$, pour tout $k \in \mathbb{N}$, $a * 4^k \equiv a \mod 3$. En conséquence, on obtient que $v(m) = \sum_{k=0}^{n-1} e_k 4^k \equiv \sum_{k=0}^{n-1} e_k \mod 3$.

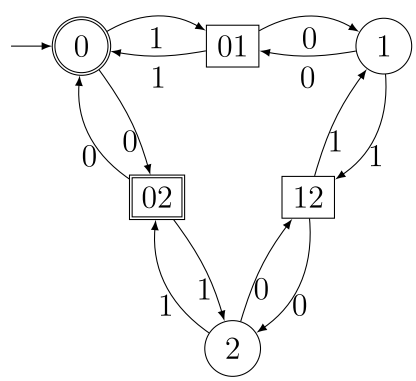
c) L'idée est de construire 3 états représentants le modulo 3 des mots de longueur paire.

- On passe respectivement des états 0, 1 et 2 aux états 1, 2 et 0 avec le mot $e_k = 10 (= 1)$.
- On passe respectivement des états 0, 1 et 2 aux états 2, 0 et 1 avec le mot $e_k = 01 (= 2)$.
- On passe ainsi d'un état 0 à lui même avec les mots $e_k = 00 (= 0)$ et $e_k = 11 (= 3)$.

Les transitions étant étiquetées par un seul chiffre, il faut construire des états intermédiaires. On remarque rapidement qu'il suffit d'utiliser 3 états intermédiaires car les transitions sont compatibles. Sur le dessin, on représente ces états intermédiaires par des rectangles.



d) Pour obtenir l'automate qui reconnaît également les mots de longueur impaire, il suffit d'accepter les mots de longueur impaire qui seraient acceptés en ajoutant un 0. Il suffit donc de reprendre le même automate en rendant acceptant l'état intermédiaire 02.



9) a) Supposons que L est reconnu par l'automate $M = (A, Q, E, I, T)$. Explicitons la construction d'un automate normalisé $\tilde{M} = (A, \tilde{Q}, \tilde{E}, \{i\}, \{t\})$ qui reconnaît $L \setminus \{1_{A^*}\}$.

- $\tilde{Q} = Q \cup \{i, t\}$
- i et t sont des états distincts des états de Q
- $\tilde{E} = E \cup \{(i, x, q) | \exists i_0 \in I, (i_0, x, q) \in E\} \cup \{(q, x, t) | \exists t_0 \in I, (q, x, t_0) \in E\}$

Montrons que $\tilde{M}$ reconnaît $L \setminus \{1_{A^*}\}$.

- $\tilde{M}$ n'accepte pas 1_{A^*} par construction.
- Soit $w \in A^* \setminus \{1_{A^*}\}$, montrons que w admet une lecture acceptante dans M si et seulement si w admet une lecture acceptante dans $\tilde{M}$.
  - ($\Rightarrow$) Supposons que $i_0 \xrightarrow{w_1} q_1 \xrightarrow{w_2} ...q_{|w|-1} \xrightarrow{w_{|w|}} t_0$ est une lecture acceptante de w dans M.
    Par construction, $i \xrightarrow{w_1} q_1 \xrightarrow{w_2} ...q_{|w|-1} \xrightarrow{w_{|w|}} t$ est alors une lecture acceptante dans $\tilde{M}$.
  - ($\Leftarrow$) Supposons que $i \xrightarrow{w_1} q_1 \xrightarrow{w_2} ...q_{|w|-1} \xrightarrow{w_{|w|}} t$ est une lecture acceptante de w dans $\tilde{M}$.
    Par construction de $\tilde{M}$, il existe respectivement i_0 et t_0 tels que $i_0 \to q_1$ et $q_{|w|-1} \xrightarrow{w_{|w|}} t_0$. Ainsi, $i_0 \xrightarrow{w_1} q_1 \xrightarrow{w_2} ...q_{|w|-1} \xrightarrow{w_{|w|}} t_0$ est une lecture acceptante de w dans M.

On a donc bien prouvé que $\tilde{M}$ reconnaît $L \setminus \{1_{A^*}\}$.

b) Définissons un automate qui va reconnaître G_q : $M_c = (A, Q \times \{1,2\}, E_c, i_c, t_c)$ où

- $E_c = \{((p, ind), x, (p', ind)) \in (Q \times \{1,2\}) \times A \times (Q \times \{1,2\}) | (p, x, p') \in E\} \cup \{((t,1), x, (p,2)) | (i, x, p) \in E\}$
- $i_c = (q, 1)$
- $t_c = (q, 2)$

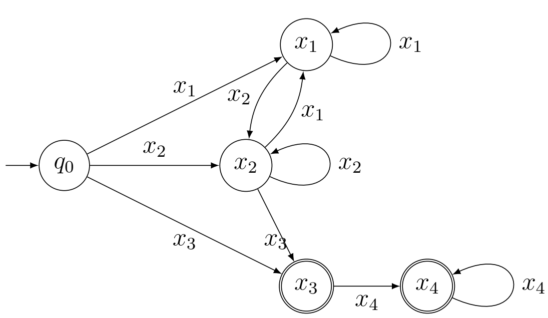
Alors, on a $i \xrightarrow{f_1} q \xrightarrow{f_2} t$ dans M si et seulement si $(q,1) \xrightarrow{f_2} (t,1) \xrightarrow{f_1} (q,2)$. Donc M_c reconnaît G_q.

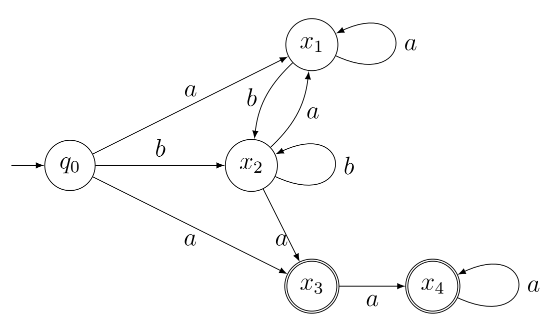
c) Le langage $Conj(L)$ est une union finie de langages rationnels, il est donc rationnel.

3 Algorithme de Glushkov

10) a) $(a^*b)^*aa^* \to (x_1^*x_2)^*x_3x_4^*$

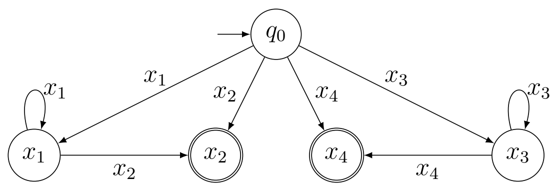
$$(\{x_1, x_2, x_3\}, \{x_1x_1, x_1x_2, x_2x_2, x_2x_1, x_2x_3, x_3x_4, x_4x_4\}, \{x_3, x_4\})$$

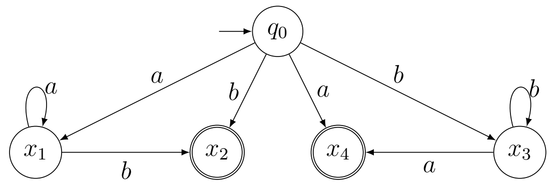




b) $a^*b + b^*a \to x_1^*x_2 + x_3^*x_4$

$$(\{x_1, x_2, x_3, x_4\}, \{x_1x_1, x_1x_2, x_3x_3, x_3x_4\}, \{x_2, x_4\})$$

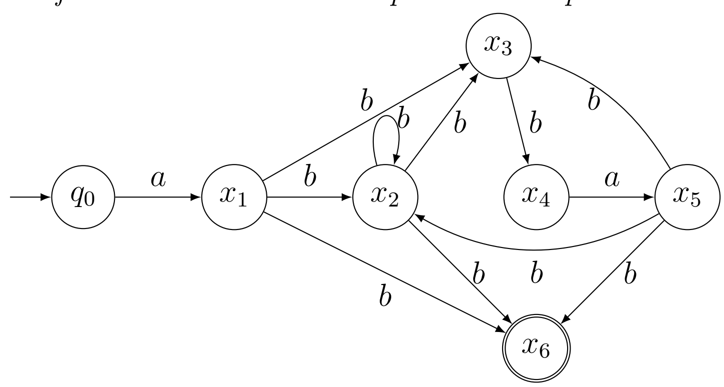




c) $a(b^* + bba)^*b \to x_1(x_2^* + x_3x_4x_5)^*x_6$

$$(\{x_1\}, \{x_1x_2, x_1x_3, x_1x_6, x_2x_2, x_2x_3, x_2x_6, x_3x_4, x_4x_5, x_5x_2, x_5x_3, x_5x_6\}, \{x_6\})$$

Pour cette question, l'automate étant plus gros, on ne représente que la version finale. Pour les trois questions suivantes, on ne dessine pas du tout les automates, mais ils sont facilement déductibles à partir des triplets.



d) $a^2b^*(ab)^* + ab^* + a^*b^2a^* \to x_1x_2x_3^*(x_4x_5)^* + x_6x_7^* + x_8^*x_9x_{10}x_{11}^*$
Noter qu'il faut traiter les carrés avec attention. En effet, il faut deux symboles distincts dans la linéarisation (par exemple $a^2 \to x_1x_2$), sinon on risque d'obtenir le langage correspondant à l'expression rationnelle où a^2 est remplacé par a^* ou un simple a, selon si on autorise le x_ix_i correspondant ou non dans les facteurs de longueur 2.

$$(\{x_1, x_6, x_8, x_9\}, \{x_1x_2, x_2x_3, x_2x_4, x_3x_3, x_3x_4, x_4x_5, x_5x_4, x_6x_7,$$
$$x_7x_7, x_8x_8, x_8x_9, x_9x_{10}, x_{10}x_{11}, x_{11}x_11\}, \{x_2, x_3, x_5, x_6, x_7, x_{10}, x_{11}\})$$

e) $((a^*bc^*)^*acbb^*b)^* \to ((x_1^*x_2x_3^*)^*x_4x_5x_6x_7^*x_8)^*$
Noter que le mot vide appartient à ce langage. Lors de la construction de l'automate, il faudra donc définir l'état initial comme acceptant.

$$(\{x_1, x_2, x_4\}, \{x_1x_1, x_1x_2, x_2x_3, x_2x_1, x_2x_2, x_2x_4, x_3x_3, x_3x_1, x_3x_1,$$
$$x_3x_2, x_3x_4, x_4x_5, x_5x_6, x_{-}x_7, x_6x_8, x_7x_7, x_7x_8, x_8x_1, x_8x_2, x_8x_4\}, \{x_8\})$$

f) $((ab)^*cb^*)^* + ((ab)^2cb^*)^*a^* \to ((x_1x_2)^*x_3x_4^*)^* + (x_5x_6x_7x_8x_9x_10^*)^*x_11^*$
Noter que comme précédemment, on linéarise le carré avec précaution : $(ab)^2 \to x_5x_6x_7x_8$. De plus, le mot vide appartient au langage, l'état initial devra donc être acceptant.

$$(\{x_1, x_3, x_5, x_{11}\}, \{x_1x_2, x_2x_1, x_2x_3, x_3x_4, x_3x_1, x_3x_3, x_4x_4, x_4x_1, x_4x_3, x_5x_6, x_6x_7,$$
$$x_7x_8, x_8x_9, x_9x_{10}, x_9x_5x_9x_{11}, x_{10}x_{10}, x_{10}x_5, x_{10}x_{11}, x_{11}x_{11}\}, \{x_3, x_4, x_9, x_{10}, x_{11}\})$$

4 Langages non rationnels

11) a) Supposons par l'absurde qu'il existe $i \neq j$ tq $E_i = E_j$. Alors, comme $\mathcal{A}$ est déterministe, le mot ab^j est accepté depuis l'état $E_i = E_j$. Le mot ab^iab^j est donc accepté par $\mathcal{A}$, ce qui est absurde, car ce mot n'appartient pas à R.

b) Supposons par l'absurde que R est rationnel. D'après le théorème de Kleene, il existe un automate fini qui reconnaît R. Soit $\mathcal{A}$ un tel automate et $\{q_1, ..., q_n\}$ l'ensemble des états de $\mathcal{A}$. Comme $\{ab^iab^i | i \in [\![0, n]\!]\} \subset R$, pour tout $i \in [\![0, n]\!]$, il existe $s \in \{q_1, ..., q_n\}$ tq $E_i = s$. Comme $\#[\![0, n]\!] = n+1 > n = \{q_1, ..., q_n\}$, il existe $i \neq j \in [\![0, n]\!]$ tq $E_i = E_j$. D'après la question précédente, cela est contradictoire avec le fait que $\mathcal{A}$ reconnaisse R. Le langage R n'est donc pas rationnel.

12) a) $L \cap B = \{a^pb^p | p \in \mathbb{N}\}$

b) L'intersection de deux langages rationnels est rationnelle (voir fin du cours sur les automates). On a vu en cours que $L \cap B$ n'est pas rationnel. Or le langage $B = a * b*$ est rationnel par définition, donc L n'est pas rationnel.

13) a) $\mathcal{A} = (\{q_0, ..., q_{N+1}, p\}, q_0, \{q_{N+1}\}, \delta)$ où $\forall x \in A, \forall i \in [\![0, N-1]\!]$, $\delta(q_i, x) = q_{i+1}$, $\delta(q_{N+1}, x) = q_{N+1}$, $\delta(p, x) = p$, $\delta(q_N, b) = p$ et $\delta(q_N, a) = q_{N+1}$.

b) Comme l'indique l'énoncé, les mots $\varepsilon, a, a^2, ..., a^{N+1}, b^{N+1}$ doivent mener à des états tous distincts. En effet, nous allons dresser une liste de mots telle que chacun est reconnu par l'automate après un unique mot de la liste initiale.

$$b^Na, b^{N-1}a, ..., a, \varepsilon$$

Il y a donc au moins $N+2$ états. De plus, en lisant b^{N+1}, on ne peut plus rien accepter ensuite. Il y a donc un état depuis lequel l'état acceptant n'est pas accessible. On a donc bien au moins $N+3$ états.

c) $\mathcal{A} = (\{q_0, ..., q_{N+1}\}, q_0, \{q_{N+1}\}, \delta)$ où $\forall x \in A, \forall i \in [\![1, N]\!]$, $\delta(q_i, x) = q_{i+1}$, $\delta(q_0, x) = q_0$ et $\delta(q_0, a) = q_1$.

d) Pour $N = 1$, $\mathcal{A} = (\{q_0, q_1, q_2\}, q_0, \{q_2\}, \delta)$ où $\forall x \in A$, $\delta(q_1, x) = q_2$, $\delta(q_0, x) = q_0$ et $\delta(q_0, a) = q_1$.

Lorsque je déterminise, j'obtiens $\mathcal{A} = (\{q_0, q_{0,1}, q_{0,2}, q_{0,1,2}\}, q_0, \{q_{0,2}, q_{0,1,2}\}, \delta)$ avec :

- $\delta(q_0, a) = q_{0,1}$ et $\delta(q_0, b) = q_0$
- $\delta(q_{0,1}, a) = q_{0,1,2}$ et $\delta(q_{0,1}, b) = q_{0,2}$
- $\delta(q_{0,2}, a) = q_{0,1}$ et $\delta(q_{0,2}, b) = q_0$
- $\delta(q_{0,1,2}, a) = q_{0,1,2}$ et $\delta(q_{0,1,2}, b) = q_{0,2}$.

e) Soit $\mathcal{A}$ un automate déterministe complet reconnaissant D_N.

Soit $w, w' \in A^{N+1}$ tels que $w \neq w'$, montrons que $\delta(q0, w) \neq \delta(q0, w')$.

Soit i_0 le plus grand indice tel que $w_{i_0} \neq w'_{i_0}$. Sans perte de généralité, je suppose que $w_{i_0} = a$ et $w'_{i_0} = b$.

Notons :

- q l'état tel que $\delta(q_0, w_{0,...,i_0}) = q$

- q' l'état tel que $\delta(q_0, w'_{0,\dots,i_0}) = q'$
- q_f l'état tel que $\delta(q_0, w) = q_f$
- q'_f l'état tel que $\delta(q_0, w') = q'_f$

Notre objectif est donc de prouver que $q_f \neq q'_f$.
Remarquons d'abord que $\delta(q, w_{i_0+1,\dots,N}.b^{N-i_0})$ est final alors que $\delta(q', w'_{i_0+1,\dots,N}.b^{N-i_0})$ ne l'est pas.
Or :

$$\delta(q, w_{i_0+1,\dots,N}.b^{N-i_0}) = \delta(q_f, b^{N-i_0})$$

et

$$\delta(q', w'_{i_0+1,\dots,N}.b^{N-i_0}) = \delta(q'_f, b^{N-i_0})$$

Donc $q_f \neq q'_f$.
On a donc au moins un état distinct pour chaque mot de longueur $N+1$. Ainsi, il y a au moins 2^{N+1} états dans l'automate $\mathcal{A}$.

f) Les langages D_N et G_N ne doivent pas accepter de mots de longueur inférieure à $N+1$. Il faut donc que le plus court chemin d'un état initial à un état acceptant soit de longueur $N+1$. Il y a donc au moins N états distincts à traverser depuis un état initial pour accéder à un état final. Cela fait au minimum $N+2$ états (car ces langages sont non-vides).

14) a) Soit $\mathcal{A} = (Q, q_0, T, \delta)$ un AFD complet reconnaissant $\mathcal{L}$. Soit l'automate non-déterministe $\tilde{\mathcal{A}} = (Q, I, \{q_0\}, \tilde{\delta})$ tel que :

- $I = T$
- $\forall x \in A, \forall q \in Q, \tilde{\delta}(q, x) = \{q' \in Q | \delta(q', x) = q\}$

Par récurrence (je vous laisse l'écrire), la lecture d'un mot w dans $\mathcal{A}$ est acceptante si et seulement si il existe une lecture acceptante de $\tilde{w}$ dans $\tilde{\mathcal{A}}$.

b) $|\mathcal{P}_n| = p^{\lceil \frac{n}{2} \rceil}$. Le langage $\mathcal{P}_n$ est fini, donc rationnel.

c) Fait en cours.

15) a) Notons $\mathcal{L}_2$ le langage des mots de longueur une puissance de 2. Supposons que $\mathcal{L}_2$ est rationnel. Alors, il existe un automate fini déterministe qui reconnaît ce langage. Cet automate ayant un nombre fini d'états, il existe $p < q$ tels que $\delta(q_0, a^{2^p}) = \delta(q_0, a^{2^q})$. Alors $a^{2^p+2^q} \in \mathcal{L}_2$. Or, $2^q < 2^q + 2^p < 2^q + 2^q = 2^{q+1}$. Ainsi $2^q + 2^p$ n'est pas une puissance de 2, donc $a^{2^p+2^q} \notin \mathcal{L}_2$. Absurde !

b) Notons $\mathcal{L}$ le langage de l'énoncé. Supposons que $\mathcal{L}$ est rationnel. Alors, il existe un automate fini déterministe qui reconnaît ce langage. Cet automate ayant un nombre fini d'états, il existe $p+1 < q$ tels que $\delta(q_0, ab^p) = \delta(q_0, ab^q)$. Alors $ab^q ab^{p+1} \in \mathcal{L}$. Absurde !