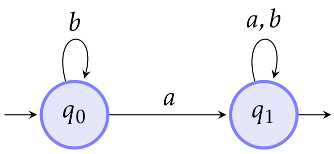
Chapitre 4

Corrigé des exercices

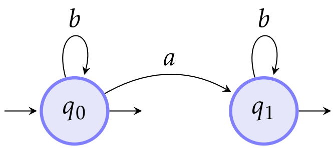
• Automates finis déterministes

Exercice 1

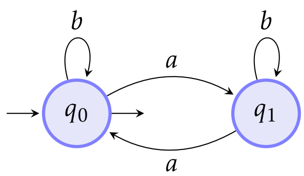
1. Le langage des mots contenant au moins une fois la lettre a :



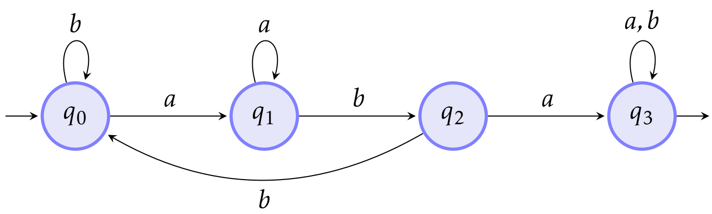
2. Le langage des mots contenant au plus une fois la lettre a :



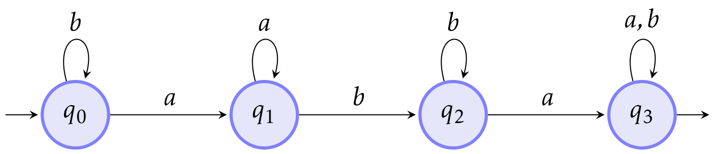
3. Le langage des mots contenant un nombre pair de fois la lettre a :



4. Le langage des mots admettant aba pour facteur :



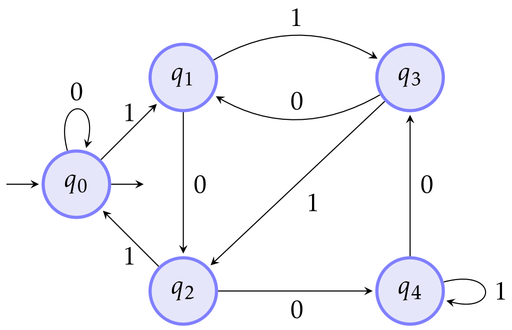
5. Le langage des mots admettant aba pour sous-mot :



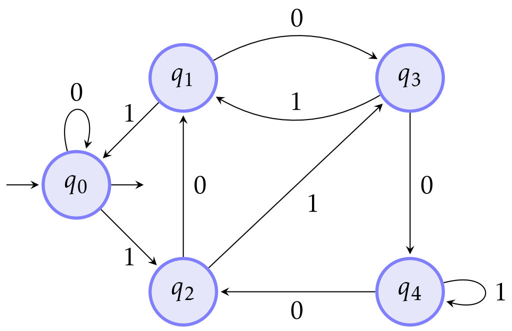
Exercice 2 Posons $n = 2p + r$ avec $r \in \{0, 1\}$; alors :

$$
\begin{aligned}
&p \equiv 0 \bmod 5 \Longrightarrow n \equiv r \bmod 5 &\qquad& p \equiv 3 \bmod 5 \Longrightarrow n \equiv 1 + r \bmod 5\\
&p \equiv 1 \bmod 5 \Longrightarrow n \equiv 2 + r \bmod 5 && p \equiv 4 \bmod 5 \Longrightarrow n \equiv 3 + r \bmod 5\\
&p \equiv 2 \bmod 5 \Longrightarrow n \equiv 4 + r \bmod 5
\end{aligned}
$$

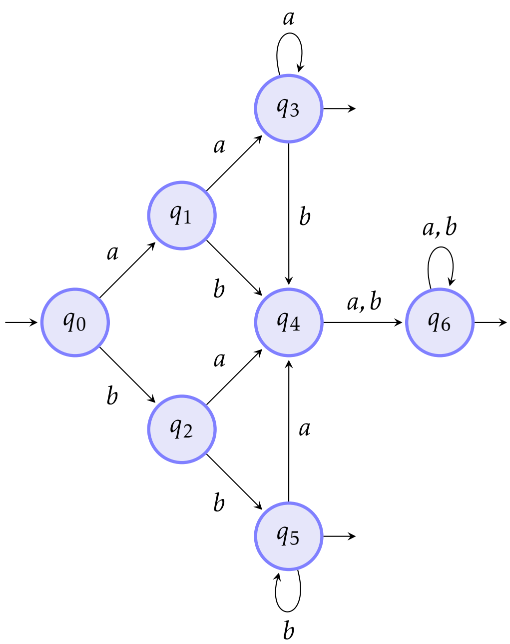
D'où l'automate :



Pour lire les entiers à partir du bit de poids le plus faible, il suffit de considérer l'automate transposé, c'est-à-dire celui obtenu en inversant l'ordre des transitions et en échangeant états initiaux et états finaux. En général, la transposée d'un automate déterministe est non déterministe, mais l'automate ci-dessus fait exception.

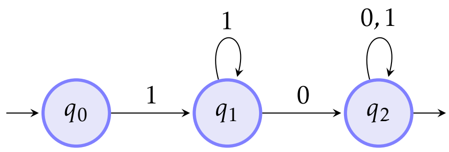


Exercice 3 Il y a trois types de mots dans ce langage : ceux qui contiennent au moins un a et un b avant le dernier caractère (état q_6), ceux qui ne contiennent que des a et qui sont de longueur au moins 2 (état q_3), ceux qui ne contiennent que des b et qui sont de longueur au moins 2 (état q_5).

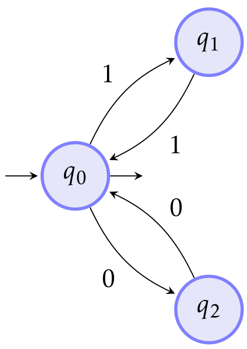


Exercice 4

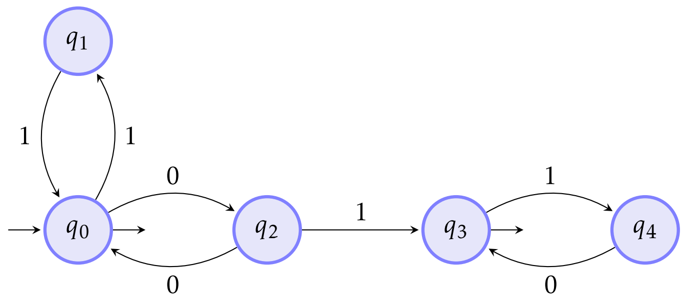
1. Les mots de L′ sont les mots qui commencent par 1 et qui comportent au moins un 0 dans leur écriture. D'où l'automate :



2. Les mots de E sont reconnus par l'automate :



3. Notons L_1 le langage dénoté par 01 et L_2 le langage dénoté par $(10)^*$. Alors $S = EL_1L_2$ donc S est reconnu par l'automate :



Exercice 5 On définit deux suites (R_i) et (S_i) de parties de Q en posant $R_0 = \{q_0\}$, $S_0 = F$ et pour tout $i \in \mathbb{N}$:

$$R_{i+1} = R_i \cup \left\{ q \in Q \;\middle|\; \text{il existe une transition d'un élément de } R_i \text{ à } q \right\}$$
$$S_{i+1} = S_i \cup \left\{ q \in Q \;\middle|\; \text{il existe une transition de } q \text{ à un élément de } S_i \right\}$$

Ces deux suites sont croissantes dans Q fini donc sont stationnaires. Leurs limites R et S sont atteintes dès lors que deux termes consécutifs sont égaux et dans ce cas R est l'ensemble des états accessibles et S l'ensemble des états co-accessibles. Il suffit dès lors de supprimer les états qui n'appartiennent pas à $R \cap S$ ainsi que les transitions dans lesquelles ces états interviennent pour obtenir l'automate émondé demandé.

Commençons par une fonction qui détermine les états accessibles en suivant l'algorithme exposé ci-dessus :

```
let accessible a =
  let rec aux1 b = function
    | []                                     -> b
    | ((i, _), j)::q when mem i b && not mem j b -> aux1 (j::b) q
    | _::q                                   -> aux1 b q
  and aux2 b = let bb = aux1 b a.Delta in
               if b = bb then b else aux2 bb
  in aux2 [a.Start] ;;
```

On détermine de même les états co-accessibles :

```
let coaccessible a =
  let rec aux1 c = function
    | []                                      -> c
    | ((i, _), j)::q when mem j c && not mem i c -> aux1 (i::c) q
    | _::q                                    -> aux1 c q
  and aux2 c = let cc = aux1 c a.Delta in
               if c = cc then c else aux2 cc
  in aux2 a.Accept ;;
```

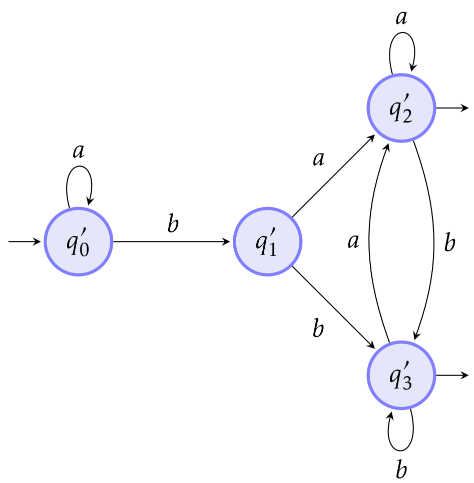
Il reste à émonder l'automate :

```
let emonde a =
  let e = intersect (accessible a) (coaccessible a) in
  let rec aux = function
    | []                                         -> []
    | ((i, _), j)::q when not mem i e || not mem j e -> aux q
    | t::q                                       -> t::(aux q)
  in {Start = a.Start; Accept = intersect a.Accept e; Delta = aux a.Delta} ;;
```

• Automates non déterministes

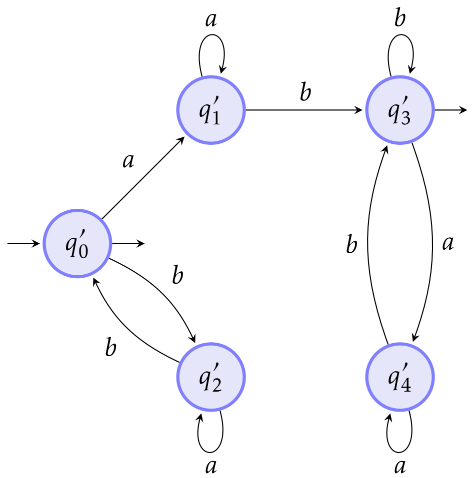
Exercice 6 La déterminisation du premier automate conduit au résultat :

δ'	a	b
$\{q_0\}$	$\{q_0\}$	$\{q_0, q_1\}$
$\{q_0, q_1\}$	$\{q_0, q_2\}$	$\{q_0, q_1, q_2\}$
$\{q_0, q_2\}$	$\{q_0, q_2\}$	$\{q_0, q_1, q_2\}$
$\{q_0, q_1, q_2\}$	$\{q_0, q_2\}$	$\{q_0, q_1, q_2\}$



La déterminisation du second automate conduit au résultat :

δ'	a	b
$\{q_0\}$	$\{q_1\}$	$\{q_2\}$
$\{q_1\}$	$\{q_1\}$	$\{q_0, q_2\}$
$\{q_2\}$	$\{q_2\}$	$\{q_0\}$
$\{q_0, q_2\}$	$\{q_1, q_2\}$	$\{q_0, q_2\}$
$\{q_1, q_2\}$	$\{q_1, q_2\}$	$\{q_0, q_2\}$



Exercice 7 Seules quatre configurations sont possibles :

– les quatre verres sont tous dans le même sens (configuration q_0) ;

– trois verres sont dans un sens et le quatrième dans l'autre sens (configuration q_1) ;

– deux verres voisins sont dans un sens et les deux autres dans l'autre sens (configuration q_2) ;

– deux verres opposés sont dans un sens et les deux autres dans l'autre sens (configuration q_3).

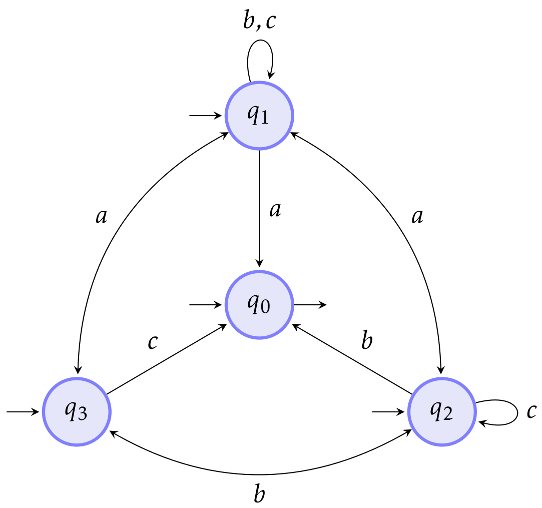
On désigne par la lettre :

– a le fait de changer l'orientation d'un des quatre verres ;

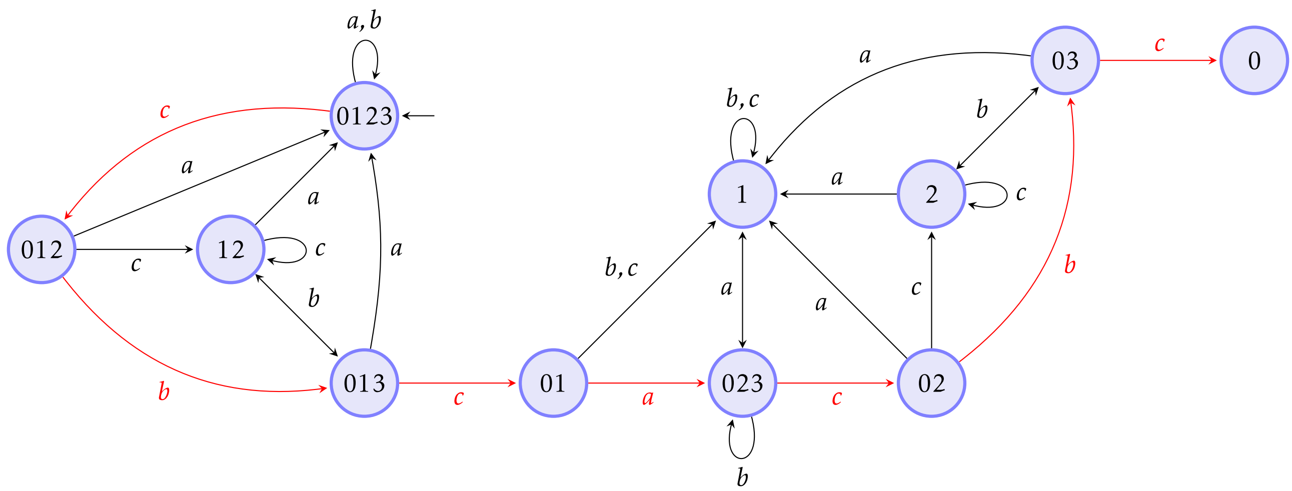
– b le fait de changer l'orientation de deux verres voisins ;

– c le fait de changer l'orientation de deux verres opposés.

Le jeu peut alors être représenté par l'automate non déterministe suivant :

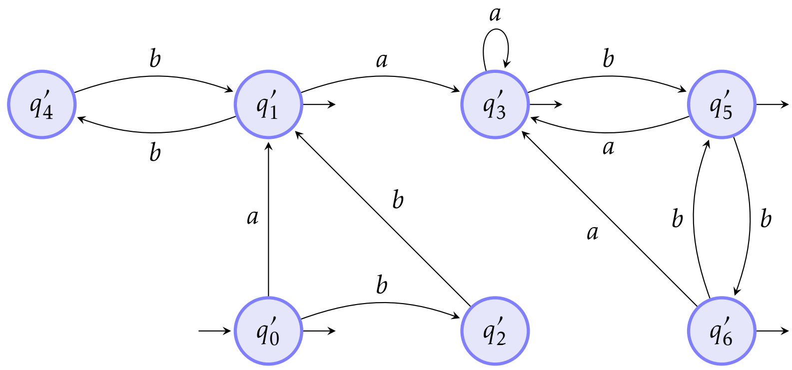


Sa déterminisation conduit à l'automate suivant :

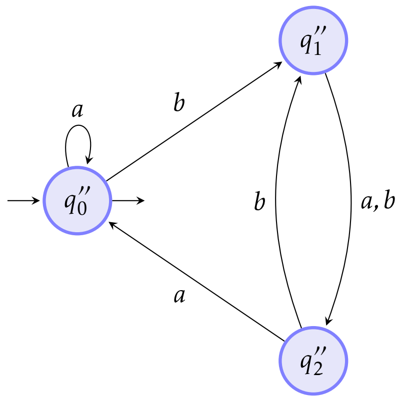


On constate que la succession de mouvement *cbcacbc* conduit nécessairement à une position gagnante pour le barman.

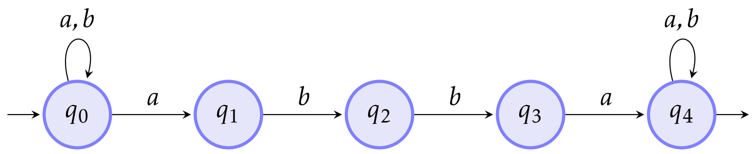
Exercice 8 $\mathcal{T}(A)$ et donc $A' = \mathcal{D}(\mathcal{T}(A))$ reconnait l'image miroir du langage reconnu par A, donc A″ reconnait le même langage que A ; il est équivalent à A.
On obtient pour A′ l'automate :



et pour A″ l'automate :

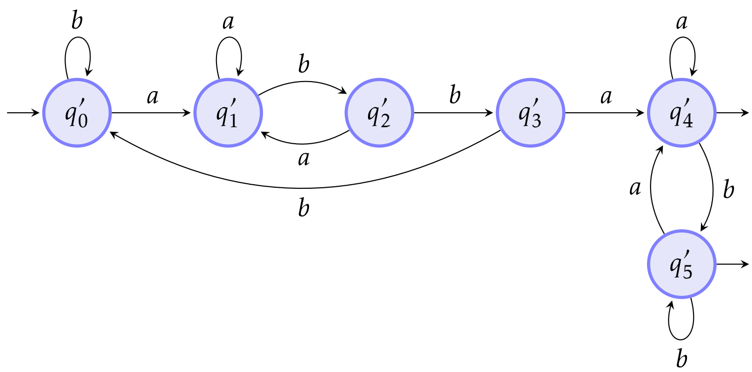


Exercice 9 L'automate non déterministe :



se déterminise en :

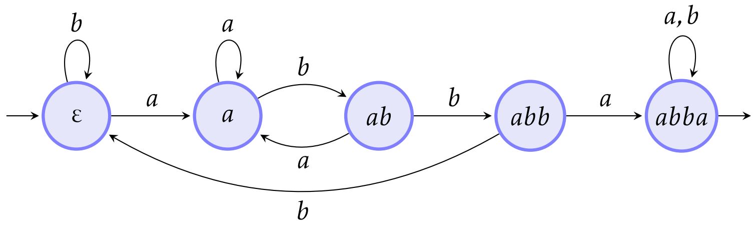
δ'	a	b
$\{q_0\}$	$\{q_0, q_1\}$	$\{q_0\}$
$\{q_0, q_1\}$	$\{q_0, q_1\}$	$\{q_0, q_2\}$
$\{q_0, q_2\}$	$\{q_0, q_1\}$	$\{q_0, q_3\}$
$\{q_0, q_3\}$	$\{q_0, q_1, q_4\}$	$\{q_0\}$
$\{q_0, q_1, q_4\}$	$\{q_0, q_1, q_4\}$	$\{q_0, q_4\}$
$\{q_0, q_4\}$	$\{q_0, q_1, q_4\}$	$\{q_0, q_4\}$



On notera que l'état q'_5 peut être supprimé sans changer le langage reconnu par cet automate.
L'algorithme KMP consiste à considérer les états et transitions suivants :

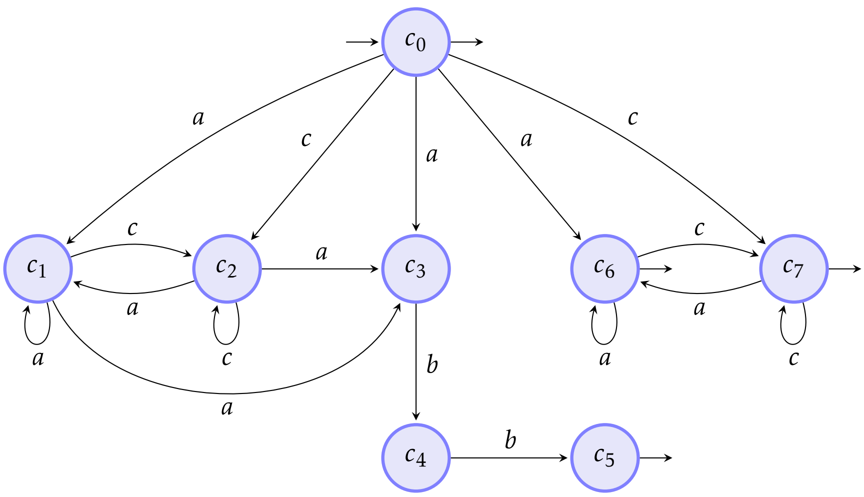
δ	a	b
ε	a	ε
a	a	ab
ab	a	abb
abb	$abba$	ε
$abba$	a	ab

puis à transformer l'état acceptant ($abba$) en puit :

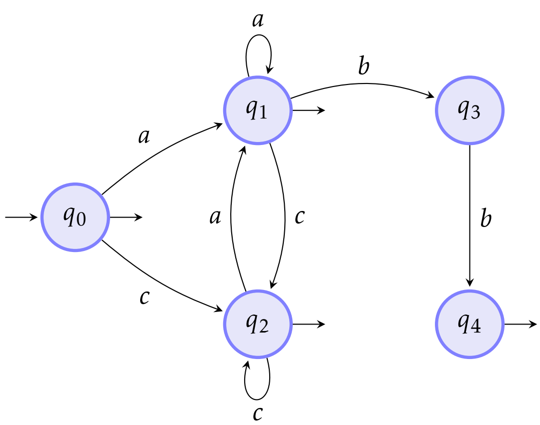


• Théorème de Kleene

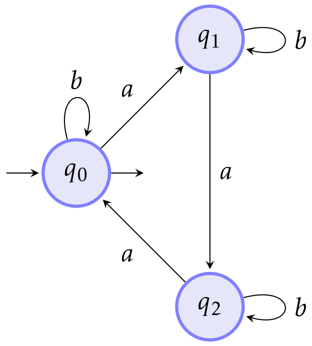
Exercice 10 On commence par se débarrasser du symbole ε : l'expression rationnelle est équivalente à $(a+c)^*abb+(a+c)^*$. On la linéarise pour obtenir un langage local : $(c_1+c_2)^*c_3c_4c_5+(c_6+c_7)^*$, avec $\mathrm{P}=\{c_1,c_2,c_3,c_6,c_7\}$, $\mathrm{S}=\{c_5,c_6,c_7\}$ et $\mathrm{F}=\{c_1^2,c_2^2,c_1c_2,c_2c_1,c_1c_3,c_2c_3,c_3c_4,c_4c_5,c_6^2,c_7^2,c_6c_7,c_7c_6\}$.
On déduit de l'automate local qui en résulte l'automate de Glushkov de l'expression rationnelle en supprimant le marquage des transitions :



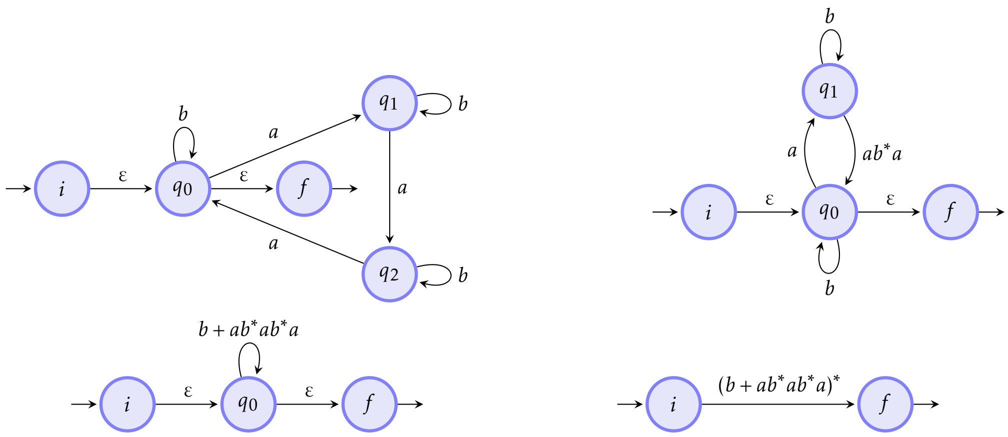
On notera que puisque ε appartient au langage c_0 est un état acceptant. Sa déterminisation fournit l'automate suivant :



Exercice 11 L'automate suivant reconnait le langage $\mathrm{L}=\left\{m\in\Sigma^*\ \middle|\ |m|_a\equiv 0 \bmod 3\right\}$:

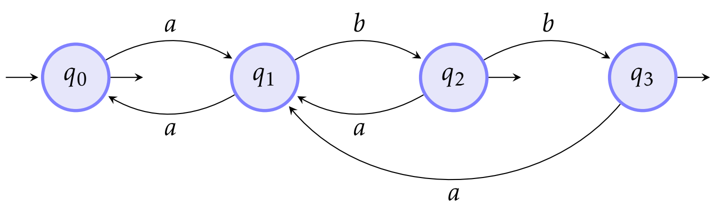


On lui applique l'algorithme d'élimination des états en éliminant successivement q_2, q_1 et q_0 :

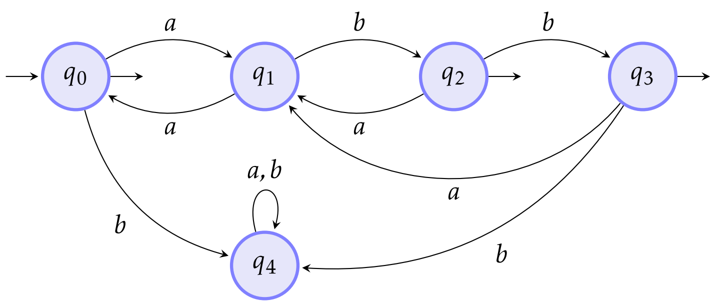


L est donc dénoté par $(b+ab^*ab^*a)^*$.

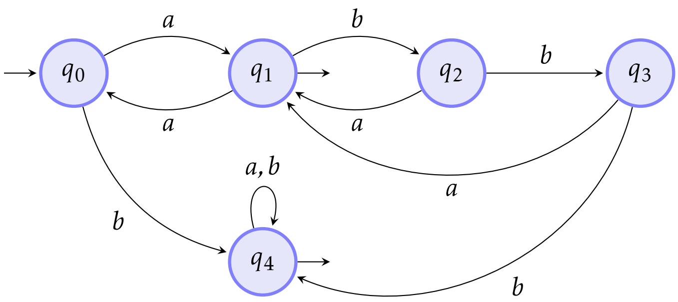
Exercice 12 L'automate suivant reconnait le langage L :



On le rend complet en ajoutant un état puit :



En inversant les états acceptants on obtient un automate qui reconnait $\overline{\mathrm{L}}$:



Exercice 13 Si L_1 et L_2 sont deux langages reconnus par des automates A_1 et A_2 nous savons calculer les automates reconnaissant l'intersection et le complémentaire donc des automates reconnaissant $L_1 \setminus L_2$ et $L_2 \setminus L_1$. Il reste à utiliser l'équivalence : $L_1 = L_2 \iff (L_1 \setminus L_2 = \emptyset \text{ et } L_2 \setminus L_1 = \emptyset)$ pour conclure.

Exercice 14 Considérons un automate fini déterministe $A = (\Sigma, Q, q_0, F, \delta)$ qui reconnait L, notons Ac (respectivement Co) l'ensemble des états accessibles (respectivement co-accessibles) et considérons les trois automates :

$$A_p = (\Sigma, Q, q_0, \mathrm{Co}, \delta), \qquad A_s = (\Sigma, Q, \mathrm{Ac}, F, \delta), \qquad A_f = (\Sigma, Q, \mathrm{Ac}, \mathrm{Co}, \delta)$$

(les deux derniers sont non déterministes).
Alors A_p reconnait pref(L), A_f reconnait suff(L), A_f reconnait fact(L) (on peut aussi observer que fact(L) = pref(suff(L)).

Exercice 15 Soit $A = (\Sigma, Q, q_0, F, \delta)$ un automate qui reconnait L. Pour tout $q \in Q$ on note I_q l'ensemble des mots qui étiquètent un chemin de q_0 à q et F_q l'ensemble des mots qui étiquètent un chemin de q à l'un des états finaux de F. Ces deux langages sont respectivement reconnus par $(\Sigma, Q, q_0, \{q\}, \delta)$ et $(\Sigma, Q, q, F, \delta)$ donc rationnels. L'égalité $\sqrt{L} = \bigcup_{q \in Q} I_q \cap F_q$ prouve alors que $\sqrt{L}$ est aussi rationnel.

Exercice 16 Soit $A = (\Sigma, Q, q_0, F, \delta)$ un automate qui reconnait le langage L. On note I l'ensemble des états accessibles à partir de q_0 en suivant un chemin étiqueté par un mot de K. On considère alors l'automate (non déterministe) $A' = (\Sigma, Q, I, F, \delta)$; nous allons montrer que A' reconnait $K^{-1}L$.
Considérons un mot $v \in K^{-1}L$ et $u \in K$ tel que $uv \in L$. Puisque $uv \in L$ le chemin $q_0 \overset{uv}{\rightsquigarrow} q_f$ mène à un état acceptant $q_f \in F$. Ce dernier se décompose en deux chemins $q_0 \overset{u}{\rightsquigarrow} q \overset{v}{\rightsquigarrow} q_f$ avec $q \in I$ ce qui montre que v étiquète un chemin menant d'un état $q \in I$ à un état $q_f \in F$. v est donc reconnu par A'.
Réciproquement, si v est reconnu par A' il existe $q \in I$, $q_f \in F$ et un chemin $q \overset{v}{\rightsquigarrow} q_f$ étiqueté par v. Par définition de I il existe $u \in K$ et un chemin $q_0 \overset{u}{\rightsquigarrow} q$ étiqueté par u ce qui prouve que uv est reconnu par A, donc que $uv \in L$.

• Lemme de l'étoile

Exercice 17 Supposons le lemme de l'étoile vérifié par le langage L_1 et posons $u = \varepsilon$, $v = a^k$ et $w = b^{k+1}$. Le mot uvw appartient à L_1 donc v se factorise en $v = a^{k_1} a^{k_2} a^{k_3}$ avec $k_2 \geqslant 1$ et pour tout $n \in \mathbb{N}$, $uv_1 v_2^{n+1} v_3 w = a^{k+nk_2} b^{k+1} \in L_1$, ce qui est absurde.
Supposons le lemme de l'étoile vérifié pour le langage L_2 et posons $u = a^k$, $v = b^k$, $w = \varepsilon$. Alors il existe $k_2 > 1$ tel que pour tout $n \in \mathbb{N}$, $a^k b^{k+nk_2} \in L_2$, ce qui est absurde.
Supposons le lemme de l'étoile vérifié pour le langage L_3 et considérons un entier premier p tel que $p \geqslant k$. Alors a^p se factorise sous la forme $a^{k_1} a^{k_2} a^{k_3}$ avec $k_2 \geqslant 1$ et pour tout $n \in \mathbb{N}$, $a^{p+nk_2} \in L_3$. En particulier, pour $n = p$ le nombre $p + pk_2$ doit être premier, ce qui est absurde.

Exercice 18 Si le langage de Dɪᴄᴋ était rationnel il existerait un automate reconnaissant les mots de la forme $a^n b^n$; or nous avons vu que dans ce cas il existe $k_2 \geqslant 1$ tel que le mot $a^n b^{n+k_2}$ soit reconnu, et ce dernier mot n'est pas un mot de Dɪᴄᴋ.

Pour tout $k \in \mathbb{N}$ le mot `(1+(1+(1+(...1))...)` (comportant k parenthèses ouvrantes et fermantes) appartient au langage Caml ; en posant $u =$ `(1+(1+(1+(...1`, $v =$ `))...)` et $w = \varepsilon$ le second lemme de l'étoile conduit à une absurdité.

Exercice 19 Supposons L rationnel ; d'après le second lemme de l'étoile il existe un entier k tel que pour tout palindrome $m = uvw$ tel que $|v| \geqslant k$ se factorise sous la forme $m = uv_1v_2v_3w$ avec $|v_2| \geqslant 1$, $|v_1v_2| \leqslant k$ et $\forall n \in \mathbb{N}$, $uv_1v_2^nv_3w$ est un palindrome. En considérant le mot $m = a^kba^k$ avec $u = \varepsilon$, $v = a^k$ et $w = ba^k$ on prouve l'existence d'un entier $p > 0$ tel que pour tout $n \in \mathbb{N}$, $a^{k+np}ba^k$ doive être un palindrome, ce qui est absurde.

Exercice 20 Supposons L rationnel et notons k l'entier qui intervient dans la première version du lemme de l'étoile. Si la conjecture est vraie il existe $p \geqslant k$ tel que $2^p - 1$ soit un nombre premier et dans ce cas le mot 1^p appartient à L. D'après le lemme de l'étoile ce mot se factorise sous la forme uvw avec $|v| \geqslant 1$ et pour tout $n \in \mathbb{N}$, $uv^nw \in$ L. En posant $v = 1^j$ on prouve que pour tout $n \in \mathbb{N}$ le mot 1^{p+nj} appartient à L, autrement dit que $2^{p+nj} - 1$ est premier. Mais en prenant $n = p$ on aboutit à une absurdité car $2^{p(j+1)} - 1$ est divisible par $2^{j+1} - 1 \geqslant 3$.