Exercice 1 : langages vers expressions régulières et automates

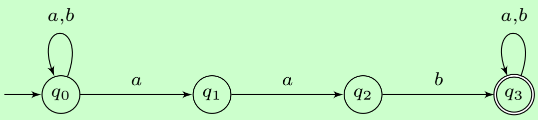
Considérons l'alphabet $\Sigma = \{a, b\}$. Pour chacun des langages suivants sur Σ, donner une expression régulière qui le dénote et un automate qui le reconnait.

1. L'ensemble des mots qui contiennent le facteur aab.

Réponse.

Expression régulière : $(a+b)^*aab(a+b)^*$

Automate (non-déterministe, non-complet) :

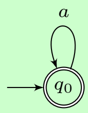


2. L'ensemble des mots qui ne contiennent pas de b.

Réponse.

Expression régulière : a^*

Automate (déterministe, non-complet) :

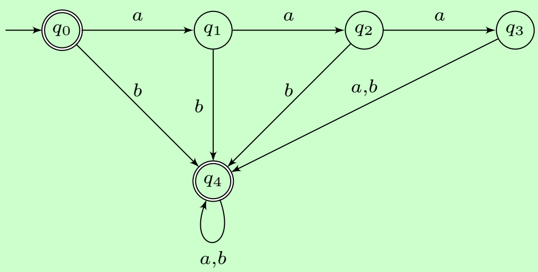


3. L'ensemble des mots qui ne sont pas $\{a, aa, aaa\}$.

Réponse.

Expression régulière : $\varepsilon + b(a+b)^* + ab(a+b)^* + a^2b(a+b)^* + a^3(a+b)^+$

Automate (déterministe complet) :

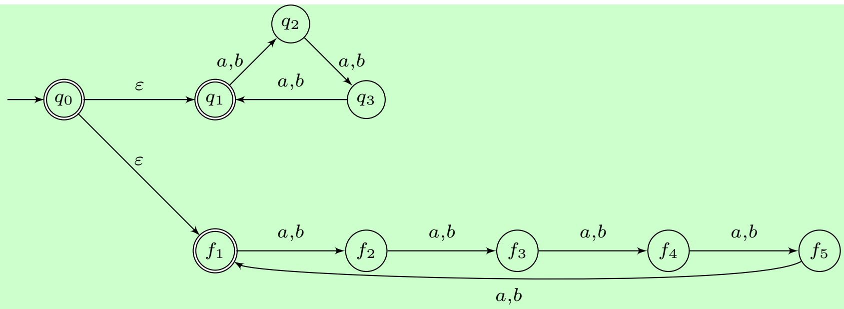


4. L'ensemble des mots dont la longueur est soit multiple de 3, soit multiple de 5.

Réponse.

Expression régulière : $((a+b)^3)^* + ((a+b)^5)^*$

Automate (non-déterministe complet) :

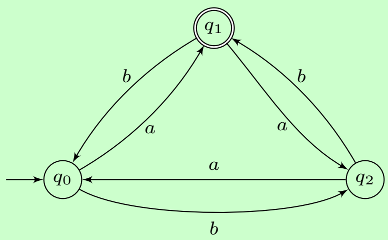


5. L'ensemble des mots w tels que $|w|_a - |w|_b \equiv 1[3]$ (*i.e.* le nombre de a moins le nombre de b est congru à 1 modulo 3).

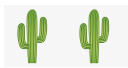
Réponse.

Expression régulière : $(ba)^*(a+b^2)(ab+(a^2+b)(ba)^*(b^2+a))^*$

Automate (déterministe complet) :



Ce dernier langage est dur à deviner directement. Il est préférable ici de construire l'automate d'abord, puis d'utiliser la méthode du cours pour construire le langage associé à cet automate. Notez également qu'il existe plusieurs écritures différentes pour ce langage.

Exercice 2 : Régularité d'un langage 

Prouver que les langages suivants ne sont pas réguliers.

1. $\{a^i b^j \mid i,j \in \mathbb{N} \wedge i \geq j\}$

Réponse.

Supposons que ce langage, dénoté L *est régulier. Selon le lemme de l'étoile, il existe* $N \in \mathbb{N}$ *tel que pour tout mot* $w \in L$*, si* $|w| \geq N$*, il existe une décomposition* $x, y, z \in \{a,b\}^*$ *de* w *(*i.e., $xyz = w$*) tel que* $|xy| \leq N$*,* $|y| > 0$ *et pour tout* $k \in \mathbb{N}$*,* $xy^kz \in L$*.*

Fixons $w = a^N b^N$*. On a que* $|w| \geq N$*, il existe donc une décomposition* $xyz = w$ *satisfaisant le lemme de l'étoile. En particulier, comme* $|xy| \leq N$ *et que les* N *premières lettres de* w *sont 'a',* y *est de la forme* a^m *avec* $1 \leq m \leq N$*.*

Par conséquent, $xy^0z = a^{N-m}b^N \notin L$ *(car* $N - m < N$*). Ceci contredit que pour tout* $k \in \mathbb{N}$*,* $xy^kz \in L$*,* L *n'est donc pas régulier.*

2. $\{w \cdot w \mid w \in \{a,b\}^*\}$

Réponse.

Supposons que ce langage, dénoté L est régulier. Soit N la constante associée à L par le lemme de l'étoile.

Fixons $w = a^N ba^N b$. On a que $|w| \geq N$, il existe donc une décomposition $xyz = w$ satisfaisant le lemme de l'étoile. En particulier, comme $|xy| \leq N$ et que les N premières lettres de w sont 'a', y est de la forme a^m avec $1 \leq m \leq N$.

Par conséquent, $xy^2z = a^{N+m}ba^N b \notin L$ (car $a^{N+m}b \neq a^N b$). Ceci contredit que pour tout $k \in \mathbb{N}$, $xy^kz \in L$, L n'est donc pas régulier.

3. $\{a^{2^n} \mid n \in \mathbb{N}\}$

Réponse.

Supposons que ce langage, dénoté L est régulier. Soit N la constante associée à L par le lemme de l'étoile. Soit N' tel que $2^{N'} > N$.

Fixons $w = a^{2^{N'}}$. On a que $|w| \geq N$, il existe donc une décomposition $xyz = w$ satisfaisant le lemme de l'étoile. En particulier, comme $|xy| \leq N$ et que les N premières lettres de w sont 'a', y est de la forme a^m avec $1 \leq m \leq N$.

Par conséquent, $xy^2z = a^{2^{N'}+m}$. $xy^2z \in L$ impliquerait qu'il existe t tel que $2^{N'} + m = 2^t$. En particulier, $t > N'$. Donc $2^t \geq 2^{N'+1} \geq 2^{N'} + 2^{N'} > 2^{N'} + N \geq 2^{N'} + m$. Ce qui contredit que $2^{N'} + m$ est une puissance de 2. Par conséquent, $xy^2z \notin L$, L n'est donc pas régulier.

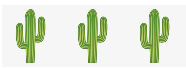
Montrer qu'aucun sous ensemble infini de $\{a^n b^n \mid n \in \mathbb{N}\}$ n'est régulier.

Réponse.

Soit L un sous-ensemble infini de $\{a^n b^n \mid n \in \mathbb{N}\}$. Supposons que L est régulier. Soit N la constante associée à L par le lemme de l'étoile.

Comme L est infini, il existe un mot $w = a^n b^n \in L$ avec $n \geq N$. On a donc que $|w| \geq N$, il existe donc une décomposition $xyz = w$ satisfaisant le lemme de l'étoile. En particulier, comme $|xy| \leq N$ et que les N premières lettres de w sont 'a', y est de la forme a^m avec $1 \leq m \leq N$.

Par conséquent, $xy^2z = a^{n+m}b^n \notin L$ (car $n+m > n$). Ceci contredit que pour tout $k \in \mathbb{N}$, $xy^kz \in L$, L n'est donc pas régulier.

Exercice 3 : langages réguliers et automates 

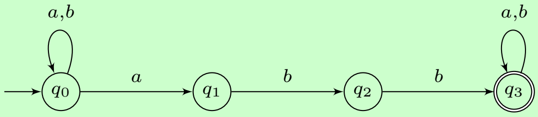
Produisez une expression régulière pour les langages suivants. Donner les étapes vous ayant permis de la construire.

— $L_1 = mirroir(\overline{(a+b)^*abb(a+b)^*})$

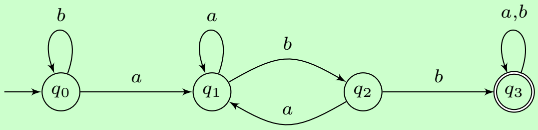
Réponse.

L'objectif de cet exercice est d'appliquer les constructions du cours sur les automates, principalement le mirroir et le complément, en n'oubliant pas d'utiliser des automates déterministes *et complet. Puis de reconstruire le langage de l'automate en expression régulière.*

Tout d'abord, l'expression $(a+b)^*abb(a+b)^*$ *est reconnu par l'automate suivant.*

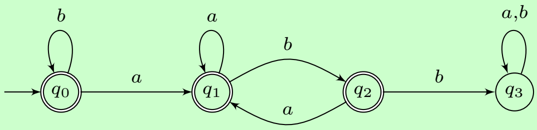


Celui-ci se déterminise et complète en l'automate :

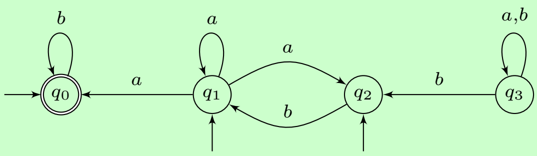


Notez que cet automate a été simplifié par rapport à celui construit en appliquant directement la déterminisation du cours en fusionnant tous les états finaux (ceux-ci permettant de lire n'importe quoi une fois atteint de toute façon).

On prend ensuite le complément en inversant états finaux et non-finaux.



Enfin on construit le mirroir en inversant états finaux et états initiaux et en inversant les transitions.



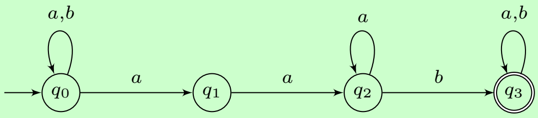
On construit ensuite le langage associé (en ignorant q_3 *qui n'est pas accessible).*

On obtient : $b^* + (\varepsilon + b)(a+ab)^*ab^*$.

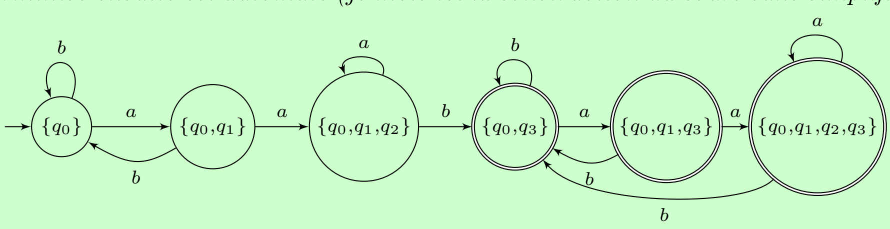
— $L_2 = \overline{(a+b)^*a^2\overline{a^*}}$

Réponse.

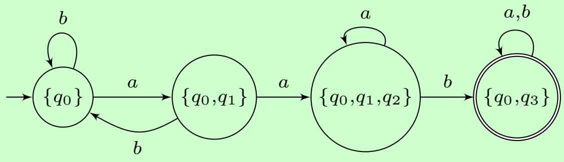
On construit d'abord l'automate reconnaissant $(a+b)^*a^2\overline{a^*}$ *(remarquez que* $\overline{a^*}$ *peut être construit au préalable, soit en prenant le complément de l'automate déterministe complet reconnaissant* a^**, soit en remarquant qu'il s'agit des mots contenant au moins une occurrence de 'b').*



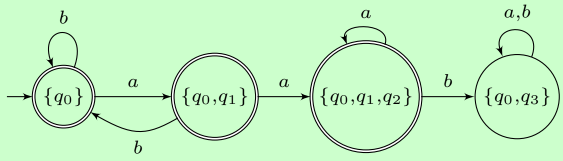
On déterminise ensuite cet automate (je mets ici la construction du cours sans simplification).



À nouveau, cet automate peut se simplifier en remarquant qu'une fois q_3 atteint, tous les mots peuvent être lu. Donnant l'automate équivalent suivant.



Son complément est

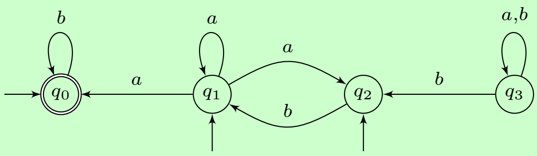


On peut maintenant construire son langage, notant que le dernier état ($\{q_0, q_3\}$) n'est pas co-accessible. L'expression régulière associée est : $(b + ab)^*a^*$

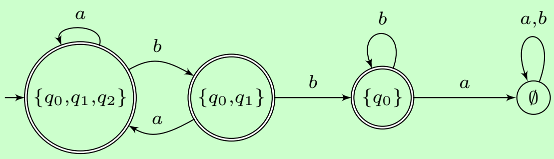
— Je donne un troisième exemple pour le langage $L_3 = mirroir(\overline{L_1} \cdot L_2)$. Ceci n'était pas attendu dans le DM bien entendu.

Réponse.

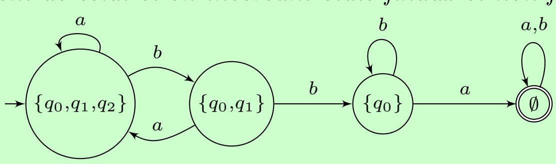
On commence par construire les automates déterministes complet représentant L_1 et L_2. L'automate que l'on a construit précédemment pour L_2 est déterministe complet, mais pas celui pour L_1, que je remets ici.



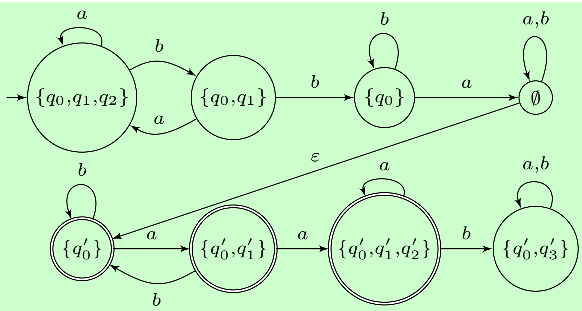
On doit donc le déterminiser et le compléter.



On prend le complément de celui-ci en inversant états finaux et non-finaux.

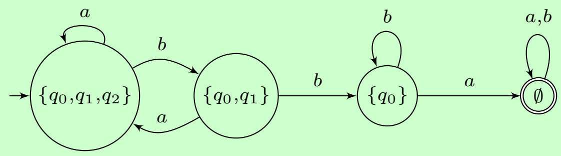


On construit ensuite l'automate concaténant cet automate et celui reconnaissant L_2. Pour rappel, ceci est fait en reliant les états finaux du premier automate aux états initiaux du second par une transition ε et en retirant le caractère final ou initial de ces automates. Je reproduit ci-dessous l'automate obtenu en ajoutant des primes aux états de l'automate produisant L_2.

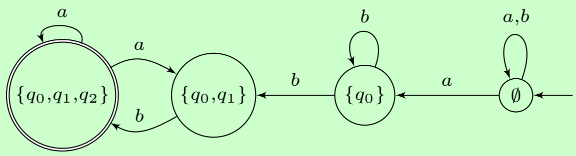


On pourrait ici déterminiser à nouveau avant de prendre le mirroir. Mais une simplification importante est à nouveau possible. En effet, cet automate est équivalent à l'automate du langage $\overline{L_1}$: En effet, tout mot accepté doit passer par l'état $\emptyset$. De plus, tout mot peut être lu dans cet état grâce à la boucle situé dessus. Finalement, on peut ensuite lire ε por atteindre un état final. En d'autre mots, tout mot atteignant q_3 est accepté, quoi qu'il advienne dans la seconde partie de l'automate.

On a donc simplement l'automate suivant :



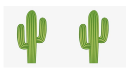
On en prend le mirroir.



Dont le langage est $L_3 = (a+b)^*abb(a+b)^*$.

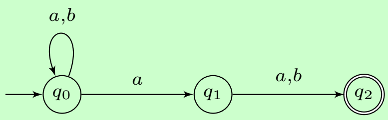
Rappel : le mirroir d'un langage est le langage obtenu en inversant l'ordre des lettres ($mirroir(abcd) = dcba$) et $\overline{L}$ est le langage complément de L ($\Sigma^* \setminus L$).

On attend ici que vous utilisiez le lien entre automate et langage régulier vu lundi 22 Janvier.

Exercice 4 : Complexité de la déterminisation 

— Construisez un automate non-déterministe simple acceptant le langage $(a+b)^*a(a+b)$.

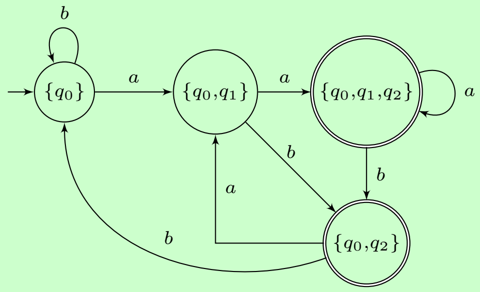
Réponse.



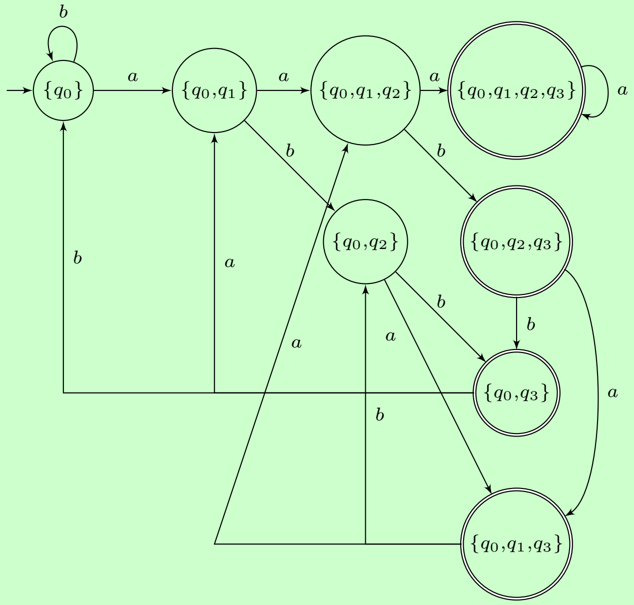
— Construisez ensuite un automate déterministe pour ce langage, ainsi que pour $(a+b)^*a(a+b)^2$.

Réponse.

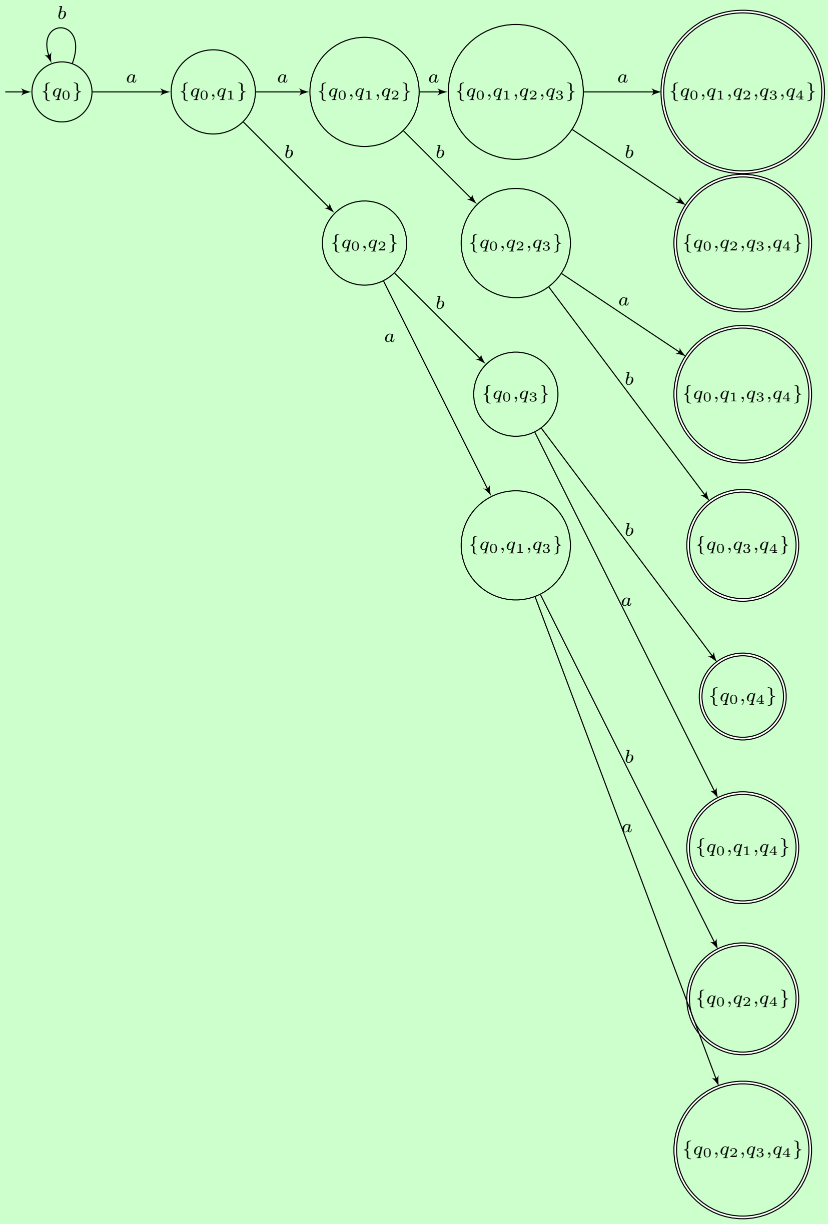
On commence par déterminiser l'automate précédent.



Ce qui va encore, mais ça va bientôt devenir douloureux. L'automate pour $(a+b)^*a(a+b)^2$ *est assez similaire, mais ajoutant un état* q_3 *qui devient l'état acceptant et que l'on peut atteindre depuis* q_2 *avec* a *ou* b*, ceci double le nombre d'états de l'automate déterministe.*



Maintenant, qui de $(a+b)^*a(a+b)^3$ *que je ne vous avais pas demandé ? On rajoute donc similairement un état* q_4 *dans la version non-déterministe, et voici une partie de l'automate déterministe correspondant (Je ne représente pas les transitions sortants des états finaux ici car ceux-ci enlèverait toute lisibilité à la représentation. Notamment car le point important de cet exercice est de montrer que l'automate déterministe nécessite un nombre d'états important.)*



— Combien d'états possèderait votre automate non-déterministe s'il était étendu au langage $(a+b)^*a(a+b)^n$ (avec $n \in \mathbb{N}$). Combien d'états possèderait sa version déterministe ?

Réponse.

L'automate non-déterministe possèderait $n+2$ états (q_0 et q_1 ne dépendent pas de n, et un état est nécessaire pour chaque $(a+b)$ que l'on souhaite ajouter au langage).

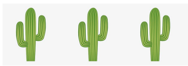
L'automate déterministe correspondant au même langage possède par construction au plus 2^{n+2} états (c'est le nombre d'ensemble d'états de l'automate non-déterministe). Cependant comme q_0 est toujours présent, on a besoin "seulement" de 2^{n+1} états.

— Bonus : prouvez qu'il n'existe pas d'automate déterministe acceptant $(a+b)^*a(a+b)^n$ ayant moins de 2^n états.

Réponse.

Soit $n \in \mathbb{N}$. Supposons qu'il existe un automate déterministe A de moins de 2^n états acceptant le langage $(a+b)^*a(a+b)^n$.

Comme il existe 2^n mots de longueur n et que A a moins de 2^n états, il existe deux mots différents $w_1 = x_1 \dots x_n$ et $w_2 = y_1 \dots y_n$ tels que ces deux mots mènent au même état q dans A (A étant déterministe, chaque mot mène à un unique état). Comme $w_1 \neq w_2$, il existe $i \leq n$ tel que $x_i \neq y_i$. Supposons sans perte de généralité que $x_i = a$ et $y_i = b$. Alors le mot $w_1 b^{n-i} \in (a+b)^*a(a+b)^n$ alors que $w_1 b^{n-i} \notin (a+b)^*a(a+b)^n$. Cependant, depuis q il existe un unique chemin dans A pour le mot b^{n-i}. Donc soit A accepte $w_1 b^{n-i}$ et $w_2 b^{n-i}$, soit il refuse les deux. A n'accepte donc pas le langage $(a+b)^*a(a+b)^n$. Ce qui contredit l'hypothèse initiale.

Exercice 5 : Minimisation d'un automate 

Soit L un langage et w un mot. On dénote par $w^{-1}L$ le langage des suffixes u tel que $wu \in L$. Par exemple, $(ab)^{-1}(a^*b^*) = b^*$.

— Calculez les expressions rationnelles des langages suivants : $\varepsilon^{-1}(ab^*(ab)^*)$, $a^{-1}(ab^*(ab)^*)$, $(aab)^{-1}(ab^*(ab)^*)$.

Réponse.

$\varepsilon^{-1}(ab^*(ab)^*) = ab^*(ab)^*, a^{-1}(ab^*(ab)^*) = b^*(ab)^*, (aab)^{-1}(ab^*(ab)^*) = (ab)^*$.

— Montrez que pour $u, v \in \Sigma^*$ et $L \subseteq \Sigma^*$, $u^{-1}(v^{-1}L) = (vu)^{-1}L$.

Réponse.

$$w \in (vu)^{-1}(L) \Leftrightarrow vuw \in L \Leftrightarrow uw \in v^{-1}(L) \Leftrightarrow w \in u^{-1}(v^{-1}(L))$$

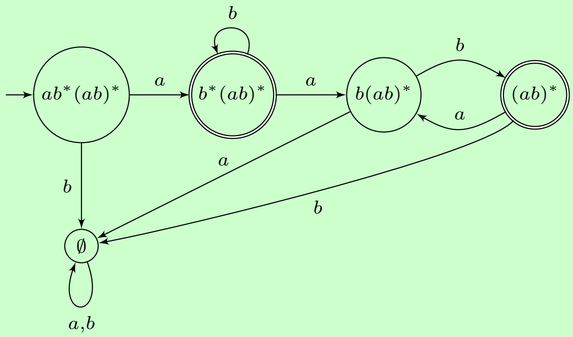
— Utilisant le résultat ci-dessus, listez l'ensemble des expressions rationnelles possibles représentant des langages de la forme $w^{-1}(ab^*(ab)^*)$.

Réponse.

Les résiduels possibles sont : $\varepsilon^{-1}(ab^*(ab)^*) = ab^*(ab)^*$, $a^{-1}(ab^*(ab)^*) = b^*(ab)^* =, b^{-1}(ab^*(ab)^*) = \emptyset$, $(aa)^{-1}(ab^*(ab)^*) = b(ab)^*$ *et* $(aab)^{-1}(ab^*(ab)^*) = (ab)^*$.

— Construisez l'automate A dont les états Q sont les éléments de la liste que vous avez construite. Dont l'état initial est associé à L. Dont les états finaux sont ceux associés à un langage contenant ε. Et qui possède une transition (L, z, L') ssi $L' = z^{-1}L$ (pour $z \in \{a, b\}$).

Réponse.



— Quel langage est produit par cet automate ?

Réponse.
Il reconnait le langage $ab^*(ab)^*$.

— Peut-on diminuer le nombre d'états de cet automate (tout en restant complet) ?

Réponse.
Non. Cette construction est appellée l'automate des résiduels. Il s'agit du plus petit automate déterministe complet pour $ab^*(ab)^*$. *Cette technique peut être répétée pour n'importe quelle langage.*

Exercice 6 : Langages continuables et mots primitifs

On fixe un alphabet fini Σ et on suppose $|\Sigma| > 1$. Dans cet exercice, on considérera des automates sur l'alphabet Σ qui seront toujours supposés déterministes complets. Un mot non-vide $w \in \Sigma^*$ est dit primitif s'il n'existe pas de mot $u \in \Sigma^*$ et d'entier $p > 1$ tel que $w = u^p$.

1. Le mot *abaaabaa* est-il primitif ? Le mot *ababbaabbbababbab* (de longueur 17) est-il primitif ?

   Réponse.
   $abaaabaa = (abaa)^2$, *il n'est donc pas primitif. Le mot ababbaabbbababbab est primitif, notamment car* 17 *est un nombre premier.*

2. Proposer un algorithme naïf qui, étant donné un mot, détermine s'il est primitif ?

   Réponse.
   Soit w *un mot. On fixe* $n = |w|$. *Pour tout* $i \in \mathbb{N}$ *tel que* $1 < i < n$ *et* i *divise* n, *on décompose* w *en* $w = u_i v_i$ *où* u_i *est le préfixe de* w *de longueur* i *(*i.e. $|u_i| = i$*). On vérifie ensuite que* $v_i = u_i^{n/i}$ *en lisant successivement les lettres de* v_i *: on vérifie que la* k*-ième lettre de* v_i *est la même que la* $((k \mod i) + 1)$*-ième lettre de* u_i.

3. Donner un exemple d'un langage régulier infini qui ne contienne aucun mot primitif.

   Réponse.
   aa^+

4. Donner un exemple d'un langage régulier infini qui ne contient que des mots primitifs.

Réponse.

a^*b

5. Un langage régulier L est dit continuable s'il a la propriété suivante : pour tout $u \in \Sigma^*$, il existe $v \in \Sigma^*$ tel que $uv \in L$. Donner un exemple de langage régulier infini non continuable. Existe-t-il des langages réguliers continuables dont le complémentaire soit infini ?

Réponse.

Le langage a^* est infini est régulier mais pas continuable car étant donné $u = b$ il n'existe pas de mot $v \in \Sigma^*$ tel que $uv \in a^*$.

Le langage $((a+b)^2)^*$ des mots de longueur pair est continuable (car tout mot peut être étendu pour être de longueur pair). Son complémentaire est infini car il contient tous les mots de longueur impair.

6. Étant donné un automate A, proposer un algorithme pour déterminer si le langage $L(A)$ qu'il reconnaît est continuable. Justifier sa correction.

Réponse.

Soit A un automate. En suivant les méthodes du cours, on construit A' l'automate reconnaissant le même langage que A mais étant déterministe et complet. $L(A)$ est continuable si tous les états accessibles de A' sont co-accessible.

En effet, si tous les états accessibles de A' sont co-accessible, alors pour tout mot $u \in \Sigma^*$, u mène à un unique état q de A' (par déterminisme et complétion de A'). q état co-accessible, il existe un chemin depuis q menant à un état final et étiqueté par un mot v. Donc $uv \in L(A)$.

Inversement, s'il existe un état accessible q de A' qui n'est pas co-accessible, alors pour u un mot dont la lecture mène à l'état q, il n'existe aucun mot v tel que $uv \in^L (A)$.

7. (a) Montrer que tout langage régulier continuable contient une infinité de mots primitifs.

Réponse.

Soit L un langage régulier continuable contenant une infinité de mots primitifs. Soit N l'entier donné par le lemme de l'étoile pour L.

Considérons le mot $w = a^N b \in \Sigma^*$. L étant continuable, il existe un mot $w' \in \Sigma^*$ tel que $ww' \in L$. D'après le lemme de l'étoile, il existe $x, y, z \in \Sigma^*$ tel que $xyz = ww'$, $|xy| \leq N$, $y \neq \varepsilon$ et pour tout $k \in \mathbb{N}, xy^kz \in L$. En particulier, de par le choix de w, $x = a^{k_1}$ et $y = a^{k_2}$.

On choisit $k = |w'| + 1$. On a donc $xy^kz = a^C bw' \in L$ où $C > |w'|$. Si xy^kz n'était pas primitif, alors w' contiendrait le facteur $a^C b$, ce qui est impossible car $C > |w'|$. Donc pour tout $k \in \mathbb{N}, xy^kz$ est primitif. L contient donc une infinité de mots primitifs.

(b) Étant donné un langage régulier continuable L, donner une borne supérieure sur la taille du plus petit mot primitif de L.

Réponse.

Soit A l'automate reconnaissant L. Soit n le nombre d'états de A. Comme vu dans le cours, on peut sélectionner $N = n + 1$ pour le choix de constante du lemme de l'étoile. Par ailleurs, de par la caractérisation de la question (6), la continuation d'un mot u correspond à un chemin depuis un état atteint par le mot u jusqu'à un état final. En l'absence de cycle, ce chemin peut être choisi de longueur inférieure à n.

En suivant la construction d'un mot primitif de la question précédente, on construit le mot $a^C bw'$ où $|w'| \leq n$ (car w' est la continuation de $a^N b$) et $C \leq N \times (|w'| + 1) \leq (n + 1)^2$. On a donc une borne supérieure sur la taille du plus petit mot primitif de $(n+1)^2 + 1 + n$.

(c) La réciproque de la question (a) est-elle vraie ?

Réponse.

Non : a^*b contient une infinité de mots primitifs et n'est pas continuable.