

1er juin 2012 - 2 heures

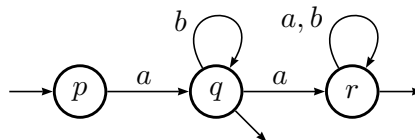
Les documents sont interdits. Les exercices sont indépendants. On pourra admettre la réponse à une question pour passer à la question suivante.

► Exercice 1.

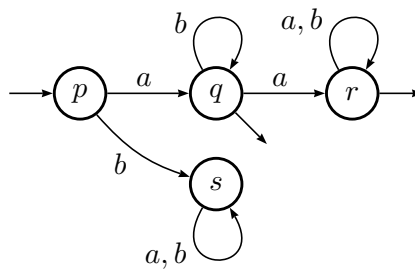
1. Calculer l'automate minimal du langage complémentaire de $ab^*(\varepsilon + a(a + b)^*)$.
2. Lorsqu'on calcule l'automate minimal du langage complémentaire reconnu par un automate $\mathcal{A}$, faut-il :
 - a) calculer d'abord l'automate minimal puis compléter ;
 - b) calculer d'abord le complémentaire puis l'automate minimal ;
 - c) l'ordre n'a pas d'importance.
 Justifier.



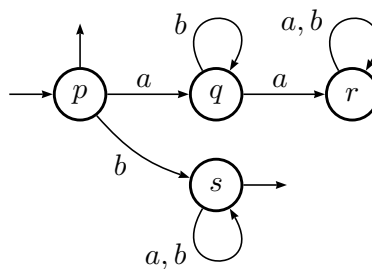
1. Pour cette expression, on peut directement trouver un automate reconnaissant le langage :



L'automate est déterministe, pour compléter, il suffit de le compléter



puis de rendre les états terminaux non-terminaux et vice-versa



On minimise le résultat. Au départ on a les états terminaux d'un côté et les états non terminaux de l'autre.

$$\equiv_0: p, s \mid q, r$$

Si on regarde les transitions étiquetées par a , on obtient

$$\begin{array}{ll} p \longrightarrow q & q \longrightarrow r \\ s \longrightarrow s & r \longrightarrow r \end{array}$$

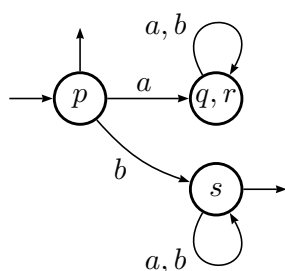
p et s vont donc dans des états qui appartiennent à des classes distinctes. Si on regarde les transitions étiquetées par b ,

$$\begin{array}{ll} p \longrightarrow s & q \longrightarrow q \\ s \longrightarrow s & r \longrightarrow r \end{array}$$

q et r vont dans des états qui appartiennent à la même classe et vont donc rester groupés. Finalement,

$$\equiv_1: p \mid s \mid q, r$$

Pour calculer $\equiv_2$, il suffit de regarder si q et r restent groupés, ce qui est le cas, donc $\equiv_1 = \equiv_2$. Finalement, l'automate minimal du complémentaire est donc

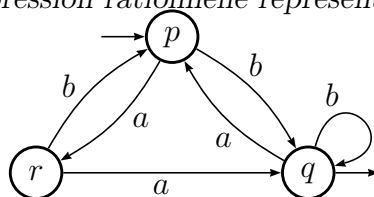


On peut émonder, mais l'automate minimal est normalement complet.

2. Il est à peu près évident que si on complémente puis on minimise on obtient le bon résultat. Dans l'autre ordre, il faut vérifier que si on complémente un automate minimal, on obtient encore un automate minimal. Si on applique la minimisation à un automate minimal, tous les états se trouvent séparés. Si on passe au complémentaire, la détermination va se dérouler exactement de la même manière, puisque la première étape consiste à séparer les états terminaux et non-terminaux et que donc les deux groupes sont les mêmes dans l'automate et dans son complémentaire; la suite de l'algorithme dépend uniquement des transitions et celles-ci n'ont pas changé. Donc on peut commencer par minimiser puis passer au complémentaire : les deux ordres sont possibles.

..... ✂

► **Exercice 2.** Calculer une expression rationnelle représentant le langage reconnu par





On peut poser un système décrivant les langages associés à chaque état :

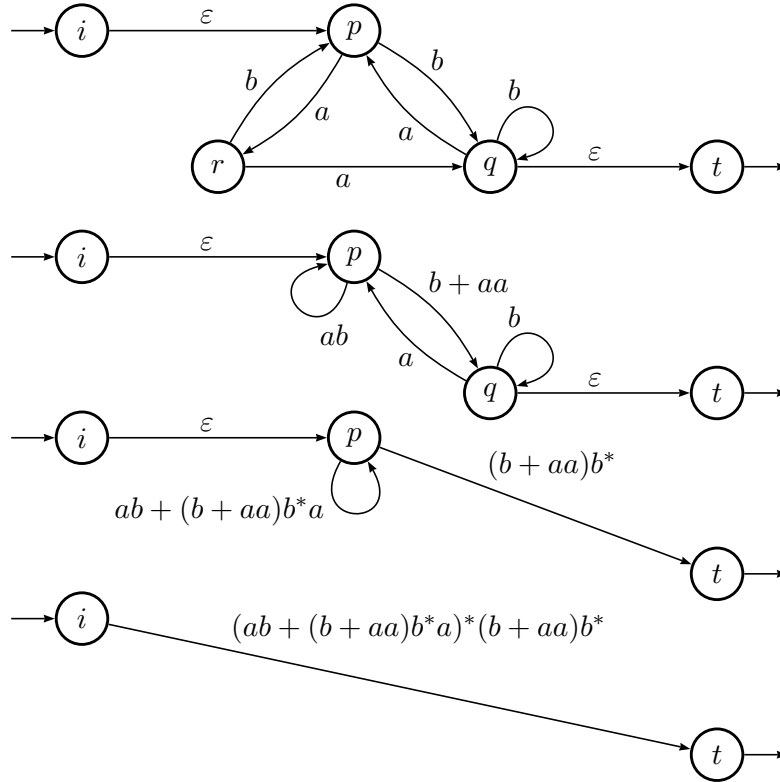
$$\begin{cases} P = bQ + aR \\ Q = bQ + aP + \varepsilon \\ R = bP + aQ \end{cases}$$

On peut remplacer R par sa valeur dans P et calculer Q en fonction de P grâce au lemme d'Arden :

$$\begin{cases} P = bQ + a(bP + aQ) = abP + (b + aa)Q \\ Q = b^*(aP + \varepsilon) \\ R = bP + aQ \end{cases}$$

Finalement, on obtient $P = abP + (b + aa)b^*(aP + \varepsilon) = (ab + (b + aa)b^*a)P + (b + aa)b^*$ donc, par le lemme d'Arden, $P = (ab + (b + aa)b^*a)^*(b + aa)b^*$. Comme p est l'unique état initial, l'expression de P représente le langage reconnu.

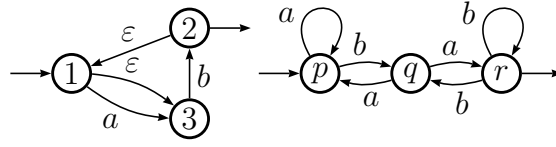
On peut sinon utiliser l'élimination d'états :



On trouve l'expression $(ab + (b + aa)b^*a)^*(b + aa)b^*$.



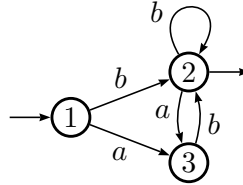
► **Exercice 3.** Calculer un automate émondé reconnaissant l'intersection des langages reconnus par les deux automates ci-dessous :



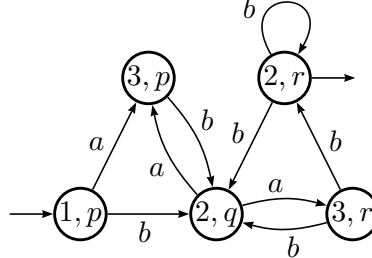
Il faut tout d'abord supprimer les ε -transitions du premier automate.

$$\begin{aligned}\text{Succ}_\varepsilon(1) &= \{3\} \\ \text{Succ}_\varepsilon(2) &= \{1, 3\} \\ \text{Succ}_\varepsilon(3) &= \emptyset\end{aligned}$$

Pour tout état p , pour tout ε -successeur q de p , pour toute flèche qui part de q , on ajoute une flèche qui part de p et qui va au même endroit.



On peut maintenant calculer l'intersection (inutile de déterminer).

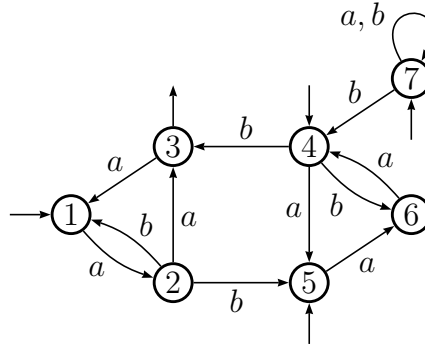


L'automate est émondé.

► Exercice 4.

1. Un automate est co-déterministe, s'il a au plus un état terminal et, pour toute lettre a , il y a au plus une transition étiquetée par a qui arrive dans chaque état.
Le miroir d'un automate est l'automate obtenu en retournant toutes les transitions et en remplaçant les flèches finales par des flèches initiales et vice-versa.
Que peut-on dire du miroir d'un automate co-déterministe?
2. Un mot w est dans le futur d'un état p s'il étiquète un chemin de p à un état final.
Montrer que dans un automate co-déterministe, un mot ne peut pas être dans le futur de deux états différents.

3. Soit $\mathcal{A}$ un automate et $\det(\mathcal{A})$ l'automate déterminisé de $\mathcal{A}$. Expliquer pourquoi si un mot w est dans le futur d'un état p de $\mathcal{A}$, il est aussi dans le futur de tout état de $\det(\mathcal{A})$ qui contient p .
4. Montrer que si $\mathcal{A}$ est un automate co-déterministe, et si X et Y sont deux états différents de $\det(\mathcal{A})$, il existe au moins un mot qui est dans le futur de l'un des deux états et pas dans le futur de l'autre.
En déduire que le déterminisé d'un automate co-déterministe est un automate minimal.
5. Calculer l'automate minimal du langage reconnu par l'automate ci-dessous :



6. Proposer un algorithme qui permet à partir d'un automate quelconque de calculer l'automate minimal sans chercher les états similaires.



1. Si on prend le miroir d'un automate co-déterministe, on obtient un automate avec au plus un état terminal et, pour toute lettre a , au plus une transition étiquetée par a qui sort de chaque état. C'est la définition d'un automate déterministe. Le miroir d'un automate co-déterministe est donc un automate déterministe.
2. Prenons un mot w quelconque. Si on lit w de droite à gauche, et que dans l'automate co-déterministe on part de l'état final et qu'on remonte les transitions, on ne peut arriver (au plus) que dans un seul état ; w n'est donc que dans le futur de cet état.

Autre preuve. Supposons qu'il existe deux états p et q distincts d'un automate co-déterministe qui ont le même mot w dans leur futur. Comme il n'y a qu'un seul état terminal, Les chemins étiquetés par w qui partent de p et q arrivent dans le même état final ; il se rejoignent donc quelque part, dans un état r . Il y a donc un chemin de p à r étiqueté par un certain mot u et un chemin de q à r étiqueté par le même mot u , ainsi qu'un mot v dans le futur de r tel que $w = uv$. Comme p et q sont distincts, u n'est pas le mot vide et comme r est le point de jonction, les chemins de p à r et de q à r étiquetés par u arrivent sur r par des transitions différentes étiquetées par la même lettre (la dernière lettre de u). C'est impossible car l'automate est co-déterministe, d'où contradiction.

3. Supposons que w est dans le futur d'un état p de $\mathcal{A}$, on a alors

$$p \xrightarrow{w_1} p_1 \xrightarrow{w_2} \dots \xrightarrow{w_n} p_{|w|} \longrightarrow$$

Si P est un état du déterminisé de $\mathcal{A}$, alors par définition de la déterminisation, p_1 est un état du successeur P_1 de P par w_1 , p_2 est un état du successeur P_2 de P_1 par w_2 , etc. On

a donc

$$P \xrightarrow{w_1} P_1 \xrightarrow{w_2} \dots \xrightarrow{w_n} P_{|w|}$$

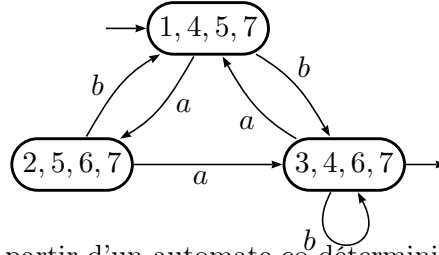
Par ailleurs, $P_{|w|}$ contient $p_{|w|}$ qui est final et est donc un état final. Donc w appartient au futur de P .

4. Réciproquement, si w est dans le futur d'un état P du déterminisé, alors il est dans le futur d'un des états de P . On peut le montrer par récurrence sur la longueur de w . Si $w = \varepsilon$, P est final et contient donc un état final; ε est dans le futur de tout état final. Si $w = aw'$. Considérons P' le successeur de P par a . Par hypothèse de récurrence, P' possède un état p' dans le futur duquel se trouve w' . Par définition de la déterminisation, il existe au moins un état p de P dont p' est un successeur par a ; w est donc dans le futur de cet état p .

Si $\mathcal{A}$ est un automate co-déterministe, les futurs de ses états sont disjoints. Si X et Y sont des états du déterminisé, il existe un état p qui est par exemple dans X et pas dans Y (ou le contraire). Tout mot w qui est dans le futur de p sera alors dans le futur de X mais pas dans celui de Y (ou le contraire).

Finalement, tous les états du déterminisé d'automate co-déterministe ont des futurs différents, ce qui est une caractéristique de l'automate minimal.

5. Comme l'automate est co-déterministe, il suffit de le déterminer pour obtenir un automate minimal.



6. Pour calculer un automate à partir d'un automate co-déterminisé, il suffit de déterminer. A partir d'un automate quelconque, on peut
 - calculer le miroir
 - déterminer
 - calculer le miroir
 - déterminer

Comme le miroir d'un automate déterministe est co-déterministe, après les trois premières étapes, on obtient un automate co-déterministe qui reconnaît le miroir du miroir, c'est-à-dire le langage de départ. Ces quatre étapes permettent donc de calculer l'automate minimal.

----- ✍