Bases formelles du TAL
Correction du partiel

Pierre-Léo Bégay

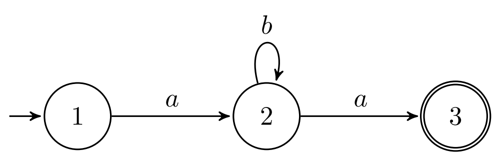
1 mars 2019

Dans tout ce partiel, on utilise l'alphabet $\Sigma = \{a, b\}$. Quand on parle d'automate non-déterministe, on autorise à la fois la définition "classique" et celle avec ϵ-transitions.

Exercice 1 [4 points]

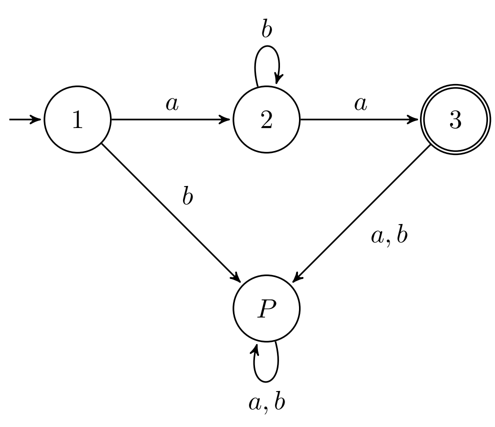
Soit $L = [\![ab^*a]\!]$

Question 1 [1] Donnez un automate fini qui reconnaît L

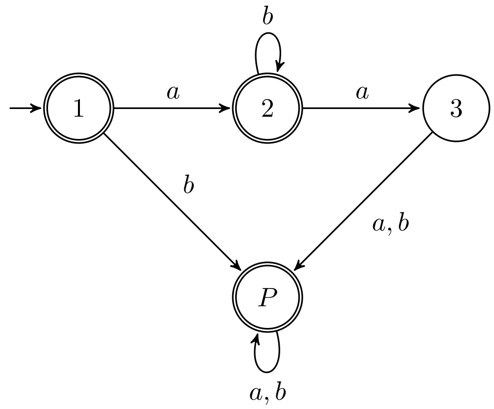


Question 2 [1] Donnez un automate fini qui reconnaît $\overline{L}$

Correction L'algorithme de complémentation nécessite un automate complet. Puisque celui de la question 1 ne l'est pas, on ajoute un état poubelle P et les transitions manquantes :



On peut maintenant inverser les états terminaux et non-terminaux :



Question 3 [2] Donnez une expression régulière qui décrit $\overline{L}$ (c'est sans doute plus facile en s'aidant de l'automate de la question précédente)

Correction Un mot est accepté s'il mène à l'état 1, 2 ou P. On analyse les 3 cas :

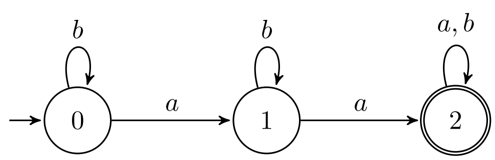
- Il n'y a aucun moyen de revenir à l'état 1. Le seul mot qui y "emmène" est donc le mot vide, ϵ
- La seule façon d'aller en 2 est de prendre la première transition en a et de lire uniquement des b. Les mots qui vont en 2 sont donc ceux de la forme ab^*
- Pour aller en P, il y a deux méthodes :
  - Prendre la première transition en b, puis lire n'importe quoi. L'expression est donc $b(a+b)^*$
  - passer par 2 puis 3. On a déjà vu que pour aller en 2 on doit lire ab^*. On ajoute ensuite un a pour aller en 3, puis un $(a+b)$ pour aller en P. D'ici, on lit ce qu'on veut. On a donc $ab^*a(a+b)(a+b)^*$, ou $ab^*a(a+b)^+$

Puisqu'on a le choix entre tous ces mots, l'expression qui les représente est une disjonction. L'expression régulière[1] de $\overline{L}$ est donc $\epsilon + ab^* + b(a+b)^* + ab^*a(a+b)^+$.

Exercice 2 [10 points]

Soit $L_1 = \{u \in \Sigma^* |\ aa \text{ est un sous-mot de } u\}$.

Question 1 [1] Donnez un automate fini déterministe complet reconnaissant L_1



[1] Notez qu'il y a toujours une infinité d'expressions régulières pour décrire le même langage, et donc équivalentes. Vous pouvez donc avoir trouvé autre chose et quand même avoir raison, je trouve juste celle-ci plus "naturelle".

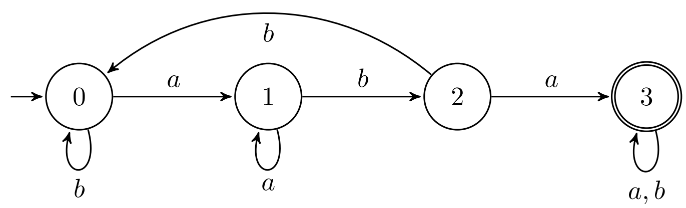
Les états servent à compter le nombre de a lus. Le dernier état est à comprendre comme "2 ou plus, mais de toute façon c'est pareil à partir de 2 on a gagné donc pas besoin de compter plus loin".

Question 2 [1] Donnez une expression rationnelle décrivant L_1

Correction Ici on peut exploiter le non-déterministe à l'envie : $\Sigma^* a \Sigma^* a \Sigma^*$.

Soit $L_2 = \{u \in \Sigma^* | \ aba \text{ est un facteur de } u\}$.

Question 3 [1] Donnez un automate fini déterministe complet reconnaissant L_2

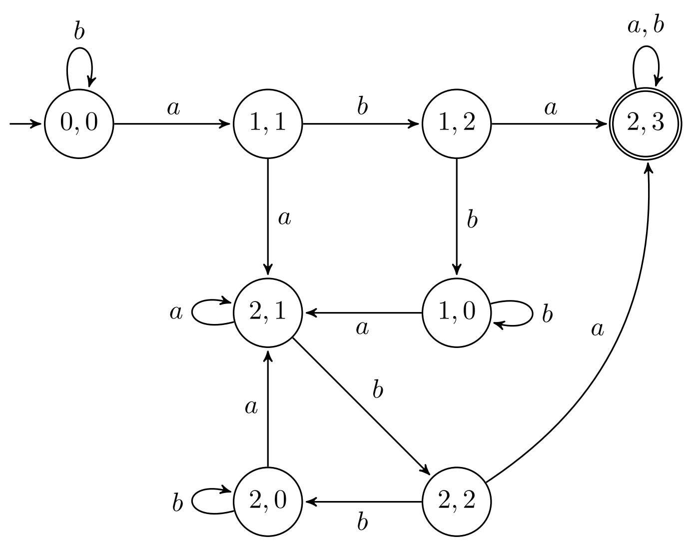


On a clairement un "chemin" aba de l'état initial à l'état gagnant 3. Si on est en 1, on vient de lire un a. Si on lit à nouveau un a, ça veut dire que le premier n'était pas le "bon" (le aba), mais qu'on peut retenter sa chance avec le nouveau. Si on lit un b en 2, ça fait deux b à la suite, ce qui n'est pas possible dans la fin du mot. On doit alors à nouveau attendre un a.

Soit $L_3 = L_1 \cap L_2$

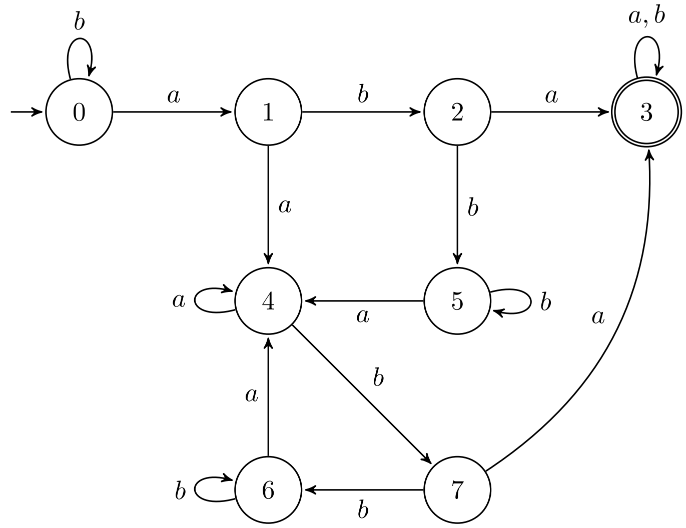
Question 4 [2] Donnez un automate déterministe complet reconnaissant L_3

Correction En appliquant l'algorithme de produit d'automates :



Question 5 [2 + 1 bonus] Minimisez l'automate obtenu.
Bonus : Qu'observez-vous ? Comment l'expliquez-vous ?

Correction Pour simplifier les notations, commence par renommer les états de l'automate précédent :

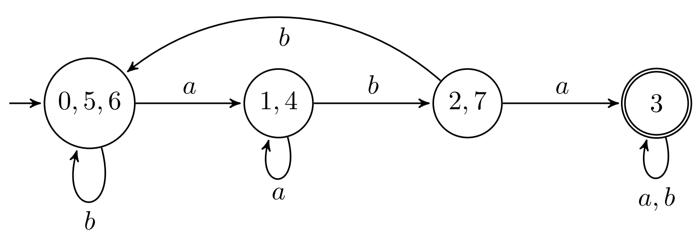


Au départ, on a deux classes : $C_T = \{3\}$ et $C_{nT} = \{0, 1, 2, 4, 5, 6, 7\}$.

Lors de la première itération, on remarque que via a, 2 et 7 envoient en C_T alors que les autres non. C'est la seule incompatibilité que l'on trouve. On sépare alors C_{nT} en $C_\alpha = \{2, 7\}$ et $C_\beta = \{0, 1, 4, 5, 6\}$.

On analyse alors C_β, et on observe que 1 et 4, contrairement aux autres, envoient sur C_α via b. C'est encore une fois la seule incompatibilité trouvée. On divise alors C_β en $C_\gamma = \{1, 4\}$ et $C_\zeta = \{0, 5, 6\}$.

A partir de maintenant, on ne trouvera plus d'incompatibilité dans les classes[2]. Il nous reste alors $C_T = \{3\}$, $C_\alpha = \{2, 7\}$, $C_\gamma = \{1, 4\}$ et $C_\zeta = \{0, 5, 6\}$. On fusionne donc les états équivalents (cad appartenant à la même classe), ce qui nous donne l'automate suivant :



[2]Attention à bien refaire le calcul pour toutes avant de dire que c'est fini ! En cassant une classe en deux, vous pouvez changer le résultat des calculs de compatibilité sur une autre

Bonus On observe que, modulo le nom des états, c'est exactement l'automate de L_2 ! C'est en fait normal, car $L_2 \subset L_1$, dans le sens où tout mot qui appartient à L_2 appartient aussi à L_1[3]. En effet, si un mot contient le facteur aba, alors il contient bien le sous-mot aa.

Pour vérifier qu'un mot appartient à L_3, il suffit donc de vérifier qu'il appartient à L_2. On peut donc se contenter d'utiliser l'automate de la question 2 (qui était déjà minimal).

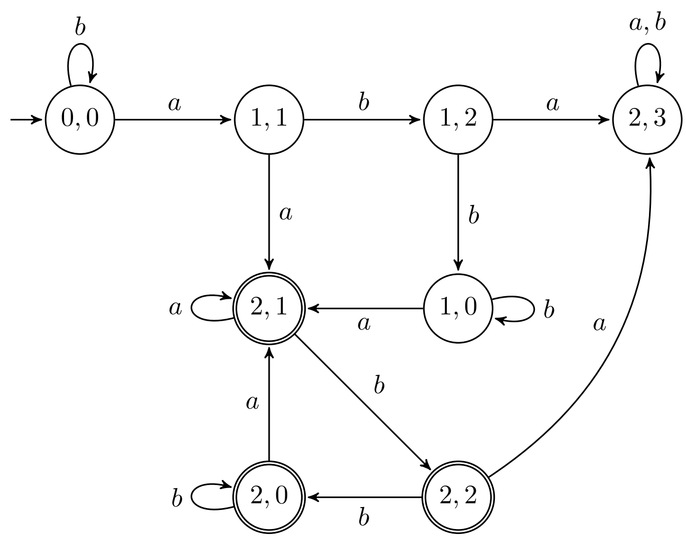
Question 6 (bonus) [1] Donnez une expression rationnelle décrivant L_3

Correction Dans le même état d'esprit, on peut se contenter d'utiliser une expression rationnelle décrivant L_2, comme $\Sigma^* aba \Sigma^*$.

Soit $L_4 = L_1 \backslash L_2$

Question 7 [1] Donnez un automate déterministe complet reconnaissant L_4.

Correction On peut réutiliser l'automate produit fait pour l'intersection, en changeant uniquement les états terminaux. En effet, au lieu de prendre comme terminaux les états dans $F_{L_1} \times F_{L_2}$, on prend ceux dans $F_{L_1} \times \overline{F_{L_2}}$[4]. On obtient donc

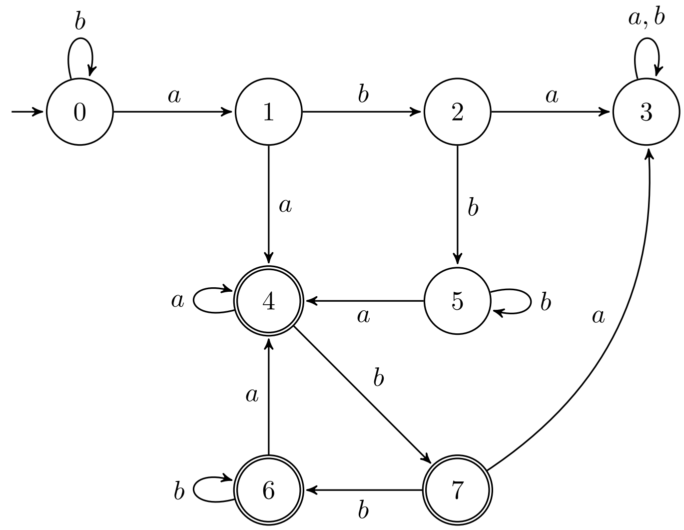


Question 8 [2] Minimisez l'automate obtenu.

Correction On renomme à nouveau les états :

[3] Attention, la réciproque n'est pas vraie. Cf. aa par exemple

[4] Informellement, puisqu'on veut être dans L_1 mais pas dans L_2, on veut à gauche un état terminal, et à droite un état non-terminal



On a au début 3 classes : $C_T = \{4, 6, 7\}$ et $C_{nT} = \{0, 1, 2, 3, 5\}$. On s'intéresse d'abord à la classe C_T, en remarquant que 7 permet d'aller en C_{nT} via a, alors que les autres restent en C_T. 7 est donc incompatible.

Or, 4 permet d'aller en 7 via b, alors que 6 boucle avec b. 4 et 6 sont donc également incompatibles. On a donc les classes $C_{nT} = \{0, 1, 2, 3, 5\}$, $C_4 = \{4\}$, $C_6 = \{6\}$ et $C_7 = \{7\}$.

On regarde maintenant C_{nT}. 1 et 5 permettent d'aller en C_4 via a, ce qui n'est pas le cas des autres, ils sont donc séparés. On a alors $C_\alpha = \{1, 5\}$ et $C_\beta = \{0, 2, 3\}$.

2 va en C_α via b, ce qui n'est pas le cas de 0 et 3. On sépare alors C_β en $C_\gamma = \{0, 3\}$ et $C_2 = \{2\}$.

0 va en C_α via a, ce qui n'est pas le cas de 3 (qui reste en C_γ. On les sépare alors en C_0 et C_3.

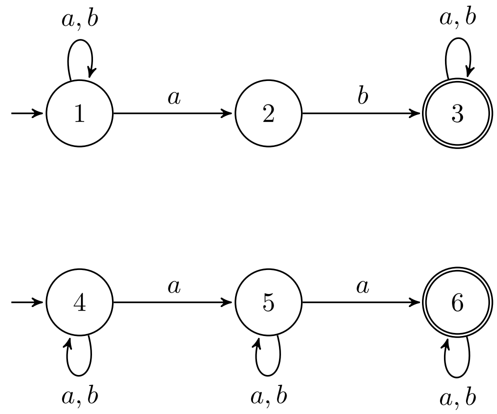
La seule classe non-atomique[5] restante est donc C_α. Or, 1 va en C_2 via b, alors que 5 reste en C_5.

Au final, toutes les classes sont atomiques, ce qui veut dire que l'automate était déjà minimal.

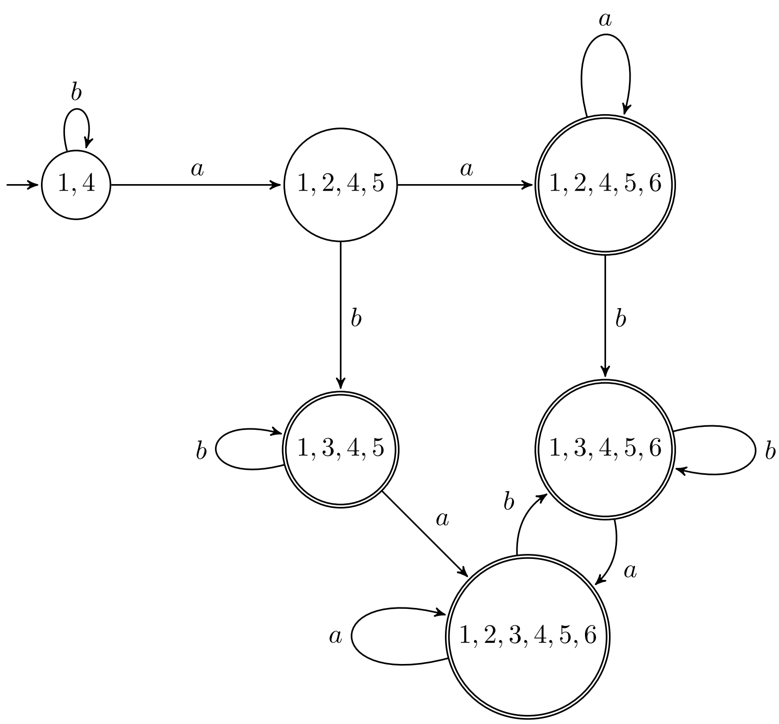
Exercice 3 [3 points]

Déterminisez l'automate suivant :

[5] Contenant plusieurs éléments



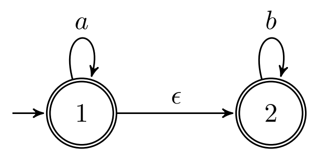
Correction En appliquant l'algorithme de déterminisation :



Exercice 4 [3 points]

Question 1 [1] Donnez un automate qui reconnaît le langage décrit par l'expression rationnelle a^*b^*.

Correction On veut lire une série (potentiellement vide) de a, puis passer à un mode où on lit uniquement des b. On peut le faire très facilement comme ceci par exemple :



Question 2 [2] Montrez que le langage $\{u \in \Sigma^* \mid |u|_a = |u|_b\}$ n'est pas reconnaissable par automate.

Correction On procède par l'absurde. On vient de montrer que le langage $[\![a^*b^*]\!]$ est reconnaissable par automate. Or, on sait également que l'intersection de deux langages reconnaissables est reconnaissable (cf. les propriétés de clôture). Donc, si $\{u \in \Sigma^* \mid |u|_a = |u|_b\}$ est reconnaissable, alors $[\![a^*b^*]\!] \cap \{u \in \Sigma^* \mid |u|_a = |u|_b\}$ l'est aussi.

Or, $[\![a^*b^*]\!] \cap \{u \in \Sigma^* \mid |u|_a = |u|_b\}$ = l'ensemble des mots composés uniquement de a puis uniquement de b, avec autant de a que de b, c'est à dire $\{a^n b^n \mid n \in \mathbb{N}\}$, dont on a vu en cours qu'il n'était pas reconnaissable.

Puisqu'on déduit quelque chose de faux, une de nos hypothèses doit l'être également. La seule hypothèse faite était que $\{u \in \Sigma^* \mid |u|_a = |u|_b\}$ est reconnaissable par automate fini. C'est donc faux.