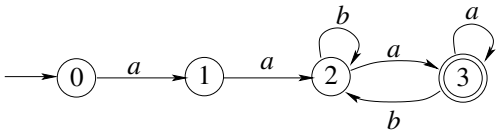
IUT Sénart-Fontainebleau, UPEC
MATH 4.2
IUT Info 2A

Florent Madelaine

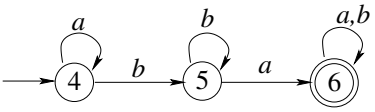
2022-23

Construction d'automates

Exercice 1. Soient A_1 et A_2 deux automates reconnaissant respectivement les langages L_1 et L_2.

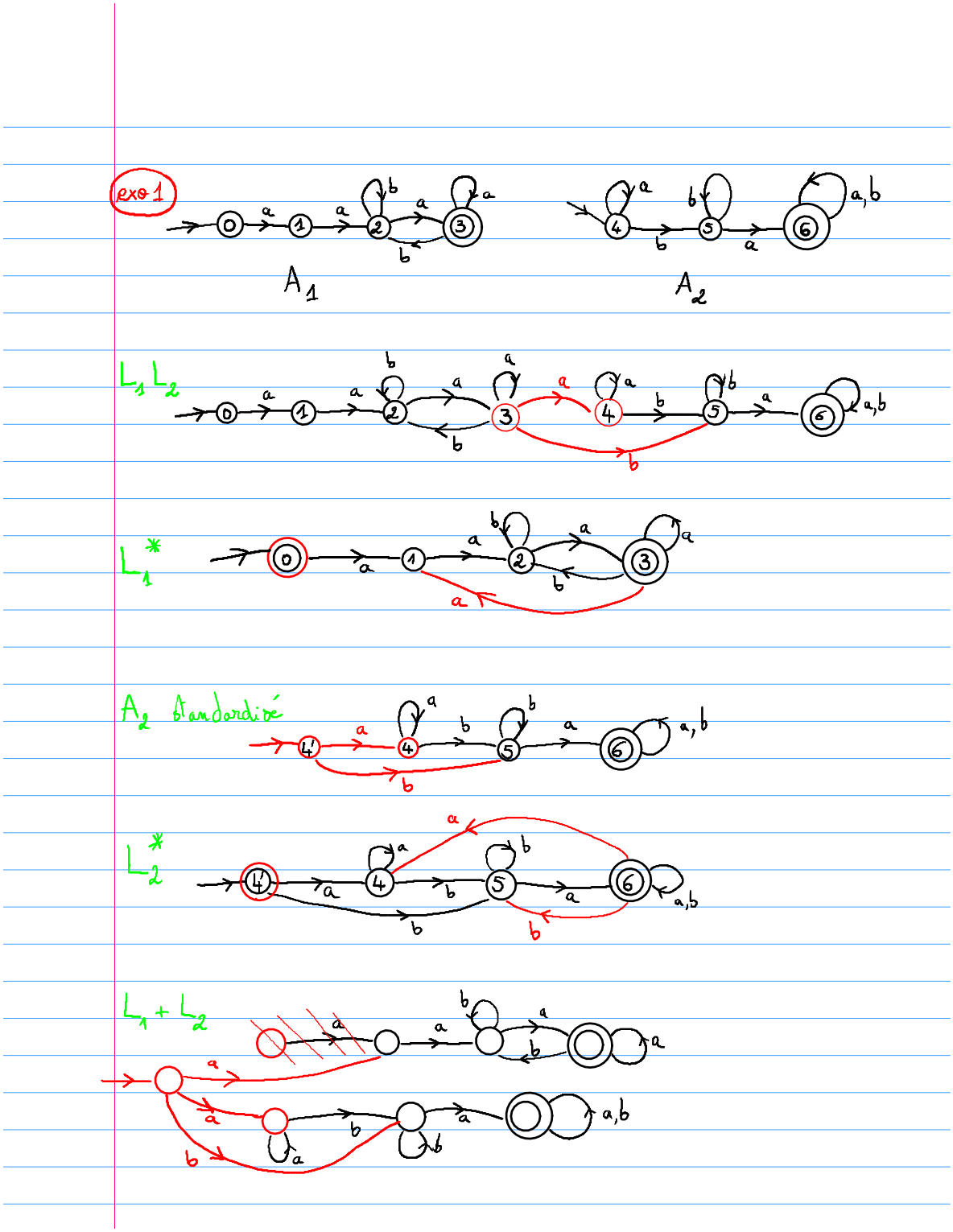


A_1

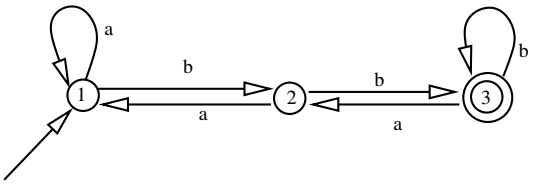


A_2

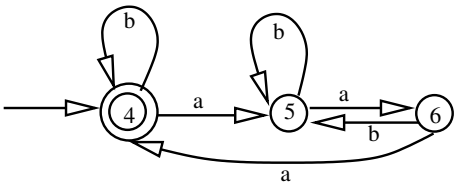
1. En utilisant la méthode vue en cours, construire un automate reconnaissant L_1L_2.
2. En utilisant la méthode vue en cours, construire un automate reconnaissant L_1^*.
3. En utilisant la méthode vue en cours, construire un automate reconnaissant L_2^*.
4. En utilisant la méthode vue en cours, construire un automate reconnaissant L_1+L_2.



Exercice 2. Soient A_1 et A_2 deux automates reconnaissant respectivement les langages L_1 et L_2.

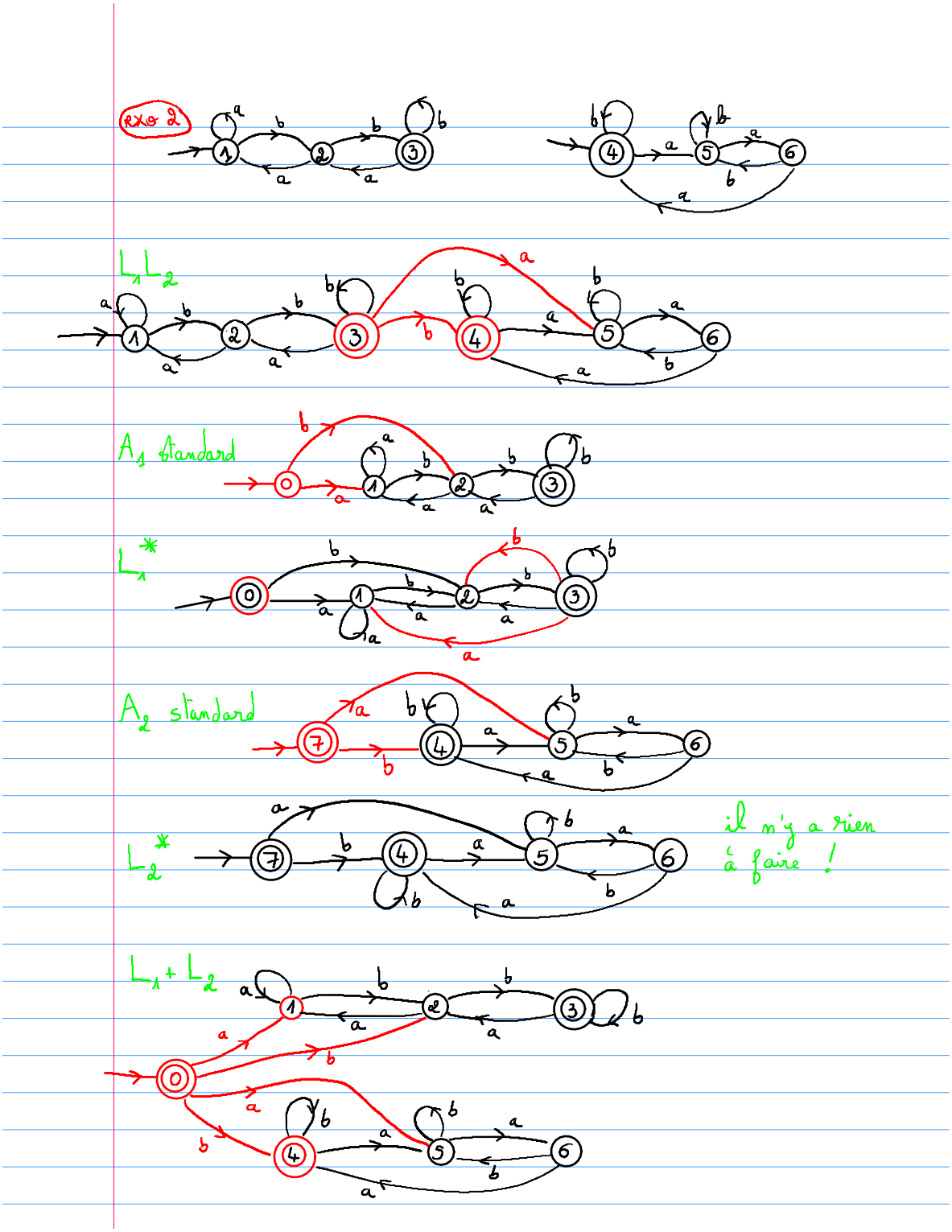


A_1



A_2

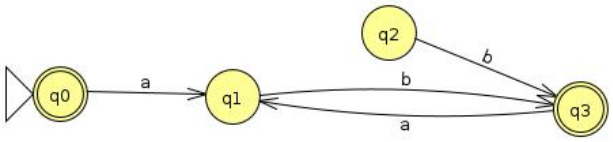
1. En utilisant la méthode vue en cours, construire un automate reconnaissant $L_1 L_2$.
2. En utilisant la méthode vue en cours, construire un automate reconnaissant L_1^*.
3. En utilisant la méthode vue en cours, construire un automate reconnaissant L_2^*.
4. En utilisant la méthode vue en cours, construire un automate reconnaissant $L_1 + L_2$.



Exercice 3. Soient A_a et A_b les automates suivants :

1. En utilisant la méthode vue en cours, construire à partir des automates A_a et A_b un automate $A_{(ab)^*}$ qui reconnaît le langage $(ab)^*$.
2. Sans refaire le calcul, donner un automate $A_{(ba)^*}$ qui reconnaît le langage $(ba)^*$.
3. En utilisant la méthode vue en cours, construire à partir des automates $A_{(ab)^*}$ et $A_{(ba)^*}$, un automate $A_{(ab)^*\cdot(ba)^*}$ qui reconnait le langage $(ab)^*\cdot(ba)^*$.
4. En utilisant la méthode vue en cours, construire à partir des automates $A_{(ab)^*}$ et $A_{(ba)^*}$, un automate $A_{(ab)^*+(ba)^*}$ qui reconnait le langage $(ab)^*+(ba)^*$.

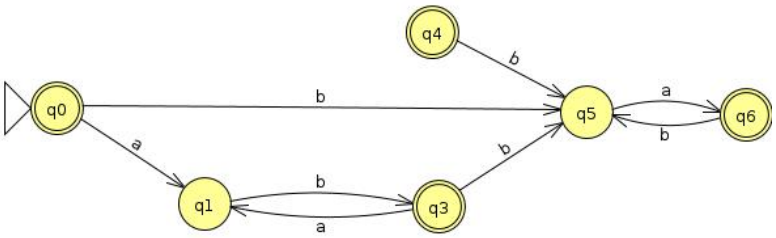
1. Automate qui reconnaît $(ab)^*$:



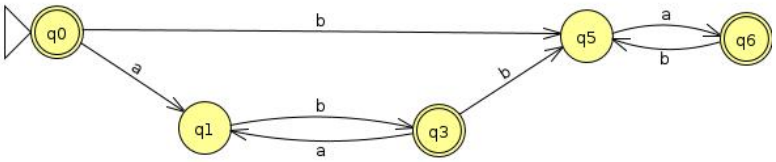
L'état q_2 est devenu inaccessible : on peut le supprimer.

2. Il suffit de permuter les rôle de a et b :

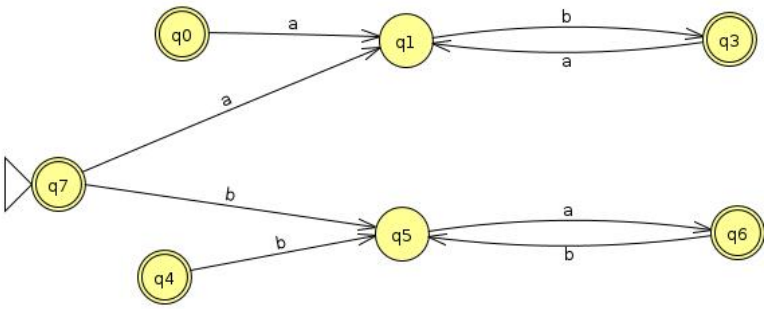
3. Automate qui reconnaît $(ab)^* \cdot (ba)^*$:



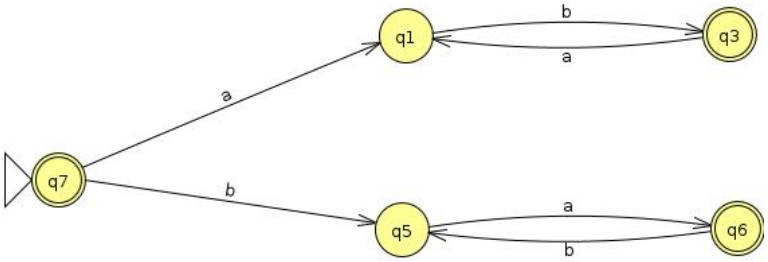
L'état q_4 est devenu inaccessible : on peut le supprimer.



4. Automate qui reconnaît $(ab)^* + (ba)^*$:



Les états q_0 et q_4 sont devenus inaccessible : on peut les supprimer.



Exercice 4. En utilisant la méthode vue en cours, construire un automate qui reconnaît le langage $(ba+a)^*$.

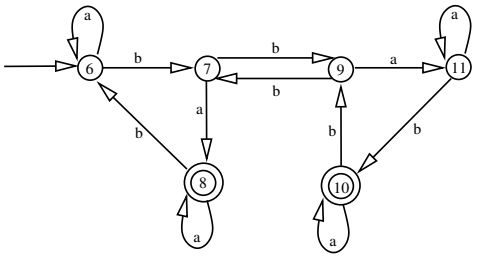
Exercice 5 (**)**.** On considère l'algorithme ci-dessous :

Soit A un automate **déterministe complet** [1]. On construit un nouvel automate A' de la manière suivante :

- A et A' ont les mêmes états et le même état initial ;
- A et A' ont mêmes transitions ;
- Les états acceptants de A' sont les états qui ne sont pas acceptants dans A.

1. Appliquer cet algorithme à l'automate ci-dessous.

1. Un automate est dit *complet* si de chaque état part une transition étiquetée par chaque lettre de l'alphabet. Dans le cas contraire, l'automate est dit *incomplet*. Pour *compléter* un automate incomplet, il suffit d'ajouter un état-rebut r (s'il n'y en a pas déjà un) et pour chaque état q et chaque lettre a, s'il n'y a pas de transition étiquetée par a qui part de q, d'ajouter une transition de q vers r étiquetée par a. Autrement dit, toutes les lettres qui bloquaient le calcul envoient maintenant au rebut.



2. Si on appelle L le langage reconnu par l'automate de départ A, que peut-on dire du langage reconnu par l'automate A' construit en suivant cet algorithme ? Justifiez votre réponse.
3. Construire un automate qui reconnaît les mots sur l'alphabet $\{a,b\}$ qui **ne contiennent pas** trois a consécutifs. Expliquez votre démarche.
4. Question subsidiaire : que se passe-t-il si on applique l'algorithme ci-dessus sur un automate A qui n'est pas complet ?

Exercice 6 (**)**.** En utilisant la méthode vue en cours, construire un automate qui reconnaît le langage $(((ab)^*bb^*)^*+a(a^*+b)b)^*$