

## TD 2 : Opérations sur les automates finis

### Exercice 1 - Opérations sur les automates

1.1 Soit  $A = \{a, b\}$ . Proposer des automates finis reconnaissant les langages suivants :

- (i)  $L_1 = \{u \in A^*, |u|_b > 1\}$
- (ii)  $L_2 = \{u \in A^*, ab \text{ est un facteur de } u\}$
- (iii)  $L_3 = \{u \in A^*, u \text{ commence par } ba\}$

1.2 En utilisant les opérations sur les automates vues en cours, déduire des automates finis pour reconnaître les langages suivants :

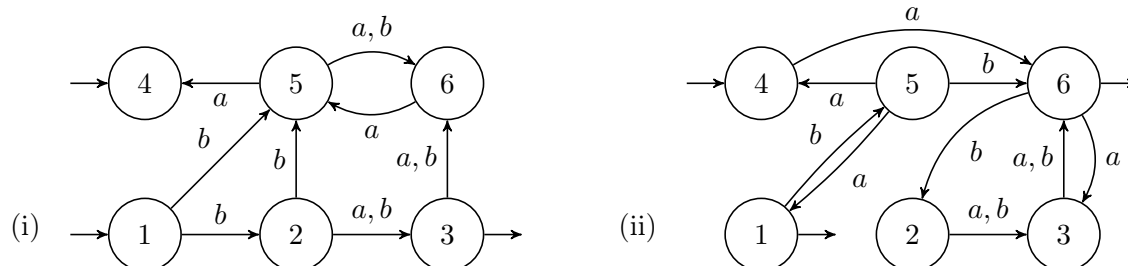
$$L'_1 := L_2 \cup L_3, \quad L'_2 := L_2^t, \quad L'_3 := L_3 \cap L_2^t, \quad L'_4 := L_1 \cap L_3.$$

1.3 Décrire les mots appartenant aux langages  $L'_1$  à  $L'_4$ , puis construire quand c'est possible un automate fini reconnaissant ces langages avec moins d'états qu'à la question précédente.

### Exercice 2 - Émondage d'automates

2.1 Expliquer comment adapter l'algorithme `accessibilite` vu en cours pour déterminer en une passe tous les états accessibles.

2.2 Pour chacun des automates suivants, déterminer les états accessibles, les états co-accessibles, puis calculer l'automate émondé équivalent :



### Exercice 3 - Opérations sur les automates

(examen 2021-2022)

Dans cet exercice, l'alphabet est  $A = \{a, b\}$ .

3.1 Donner un automate pour le langage  $L_1$  des mots se terminant par  $aa$ .

3.2 En déduire un automate pour  $L_1^*$ .

3.3 Existe-t-il :

- (i) un mot de  $L_1$  qui n'est pas dans  $L_1^*$  ?
- (ii) un mot de  $L_1^*$  qui n'est pas dans  $L_1$  ?

3.4 Donner un automate pour le langage  $L_2 = \{u \in A^*, |u|_b \equiv 1 \pmod{2}\}$  des mots avec un nombre impair de  $b$ .

3.5 Construire un automate pour le langage  $L_1 \cap L_2$ .

3.6 Quel est le plus petit mot de  $L_1 \cap L_2$  ?

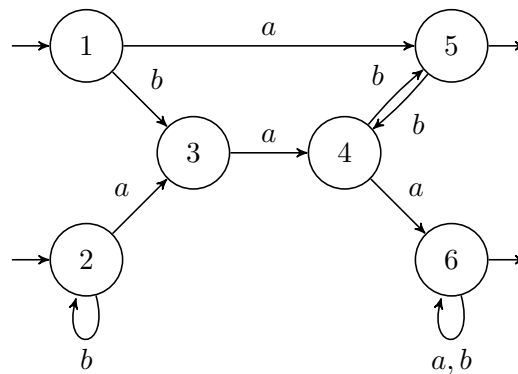
#### Exercice 4 - Normalisation d'un automate

Un automate normalisé est un automate fini qui :

- (i) n'a qu'un seul état initial  $i$ ,
- (ii) n'a qu'un seul état final  $f$ ,
- (iii) n'a aucune transition menant en  $i$ ,
- (iv) n'a aucune transition partant de  $f$ .

En particulier, on notera que tout automate normalisé est un automate standard.

4.1 Construire un automate normalisé équivalent à l'automate suivant :



4.2 Proposer un algorithme **normalisation** permettant de construire, étant donné un automate  $\mathcal{A} = (A, Q, I, F, E)$ , un automate normalisé équivalent à  $\mathcal{A}$ .

4.3 Vérifier à l'aide de l'exemple de la question 1 que cet algorithme semble correct.

4.4 Démontrer que cet algorithme est correct.

4.5 Comment utiliser cette normalisation pour reconnaître un produit de deux langages reconnaissables ? Comparer avec la méthode vue en cours.