Compilation
TD 2

1 Automates d'états finis

Exercice 1

1. Soit le langage $L_1 = \{w|w \text{ commence avec un 1 et se termine avec un 0}\}$. Construire un automate fini déterministe qui reconnaît L_1.
2. Soit le langage $L_2 = \{w|w \text{ contient au moins trois 1}\}$. Construire un automate fini déterministe qui reconnaît L_2.
3. Soit le langage $L_3 = \{w|w \text{ contient la sous-chaîne 0101}\}$. Construire un automate fini déterministe qui reconnaît L_3.
4. Soit le langage $L_4 = \{w|w \text{ a une taille d'au plus 5}\}$. Construire un automate fini déterministe qui reconnaît L_3.

Exercice 2

1. Construire un automate fini non déterministe de trois états qui reconnaît le langage $0^*1^*0^+$.
2. Construire un automate fini non déterministe de trois états qui reconnaît le langage $1^*(001^+)^*+$.

Exercice 3

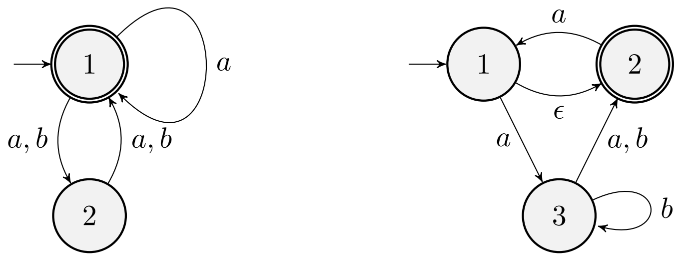
1. Soit le langage $L_1 = \{w|w \text{ a au moins 3 } a \text{ et au moins 2 } b\}$. Construire un automate fini déterministe qui reconnaît L_1.
2. Soit le langage $L_2 = \{w|w \text{ a exactement 2 } a \text{ et au moins 2 } b\}$. Construire un automate fini déterministe qui reconnaît L_2.

Exercice 4

1. Soit le langage $L_1 = \{w|w \text{ ne contient pas la sous-chaîne } ab\}$. Construire un automate fini déterministe qui reconnaît L_1.
2. Soit le langage $L_2 = \{w|w \text{ ne contient ni la sous-chaîne } ab \text{ ni la sous-chaîne } ba\}$. Construire un automate fini déterministe qui reconnaît L_2.

Exercice 5

Déterminiser les deux automates suivants.



(a) (b)

Figure 1 – Automates de l'exercice 5.

Exercice 6

1. Construire un automate fini non déterministe équivalent à l'expression régulière $(0+1)^*000(0+1)^*$.
2. Construire un automate fini non déterministe équivalent à l'expression régulière $\left(\left((00)^*\,(11)\right)+01\right)^*$.
3. Construire un automate fini non déterministe équivalent à l'expression régulière $\emptyset^*$.

Exercice 7

Pour chacun des languages suivants, donner deux chaînes appartenant et deux chaînes non appartenant. On suppose que l'alphabet $\Sigma = \{a, b\}$.

1. a^*b^*
2. $a\,(ba)^*\,b$
3. $a^* \cup b^*$
4. $(aaa)^*$
5. $\Sigma^*a\Sigma^*b\Sigma^*a\Sigma^*$
6. $aba \cup bab$
7. $(\epsilon \cup a)\,b$
8. $(a \cup ba \cup bb)\,\Sigma^*$

Exercice 8

Convertir les deux automates suivants en expressions régulières équivalentes.

(a) (b)

Figure 2 – Automates de l'exercice 8.

Exercice 9

Construire pour chacune des expressions régulières suivantes, un AFD équivalent :

1. $a(abb)^* + b$,
2. $a^+ + (ab)^+$,
3. $(a + b^+)a^+b^+$.

Exercice 10

Montrer que les langages suivants ne sont pas réguliers :

1. $A_1 = \{0^n 1^n 2^n | n \geq 0\}$,
2. $A_2 = \{www | w \in \{a, b\}^*\}$,
3. $A_3 = \{a^{2^n} | n \geq 0\}$.

Exercice 11

1. Soit $B_n = \{a^k | k \text{ est un multiple de } n\}$. Montrer que pour tout $n \geq 1$, B_n est régulier.
2. Soit $C_n = \{x | x \text{ est un nombre binaire multiple de } n\}$. Montrer que pour tout $n \geq 1$, C_n est régulier.

Exercice 12

Soit $\Sigma = \{0, 1, +, =\}$ et $ADD = \{x = y+z | x, y, z$ sont des entiers binaires et x est la somme de y et de $z\}$. Montrer que ADD n'est pas régulier.

Exercice 13

Dans certains langages de programmation, les commentaires apparaissent entre des délimiteurs e.g. `/*` et `*/`. Soit C le langage de tous le commentaires valides. Un membre de C doit commencer par `/*` et se terminer par `*/` et ne doit contenir entre les deux aucun `*/`. Pour simplifier, on suppose que $V = \{a, b, /, *\}$.

1. Construire un AFD qui reconnaît C.
2. Donner une expression régulière équivalente.