Compilation
TD 2

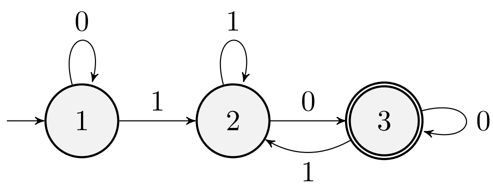
1 Automates d'états finis

Exercice 1

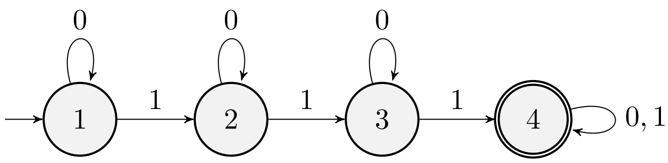
1. Soit le langage $L_1 = \{w|w \text{ commence avec un 1 et se termine avec un 0}\}$. Construire un automate fini déterministe qui reconnaît L_1.
2. Soit le langage $L_2 = \{w|w \text{ contient au moins trois 1}\}$. Construire un automate fini déterministe qui reconnaît L_2.
3. Soit le langage $L_3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

Eléments de solution de l'exercice 1

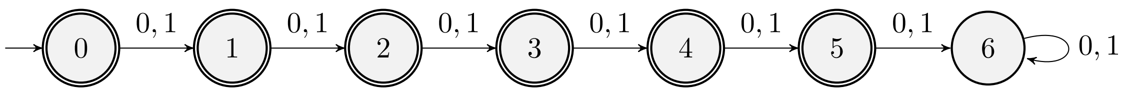
1.



2.



3.

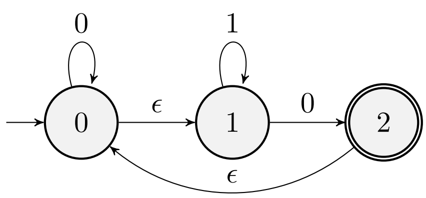


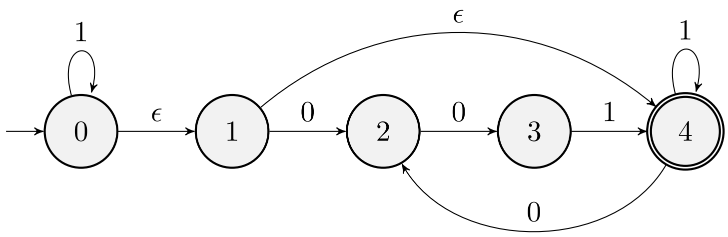
Exercice 2

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 $1^*(001^+)^*+$.

Eléments de solution de l'exercice 2

1.



2. 

Exercice 3

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

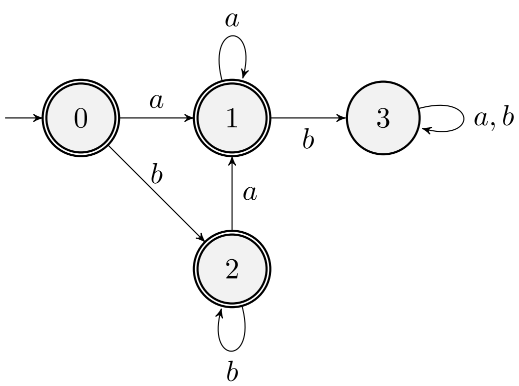
Eléments de solution de l'exercice 3

Exercice 4

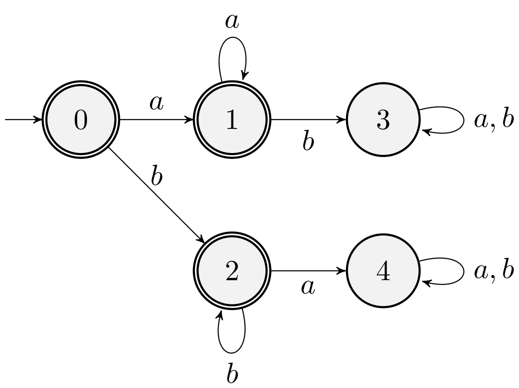
1. Soit le langage $L_1 = \{w|w$ ne contient pas la sous-chaîne $ab\}$. Construire un automate fini déterministe qui reconnaît L_1.
2. Soit le langage $L_2 = \{w|w$ ne contient ni la sous-chaîne ab ni la sous-chaîne $ba\}$. Construire un automate fini déterministe qui reconnaît L_2.

Eléments de solution de l'exercice 4

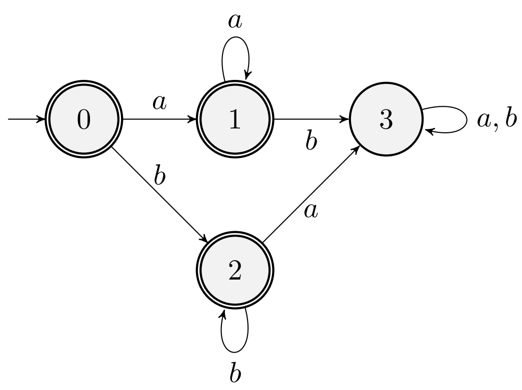
1. On construit d'abord l'automate qui reconnaît les mots contenant la sous-chaîne ab puis on prend son complémentaire : l'état final devient le seul état non final et tous les autres états non finaux deviennent finaux.



2. On construit les deux automates qui reconnaissent resp. les mots contenant ab resp. ba. on les connecte à partir de l'état initial et on fait comme pour la question précédente.

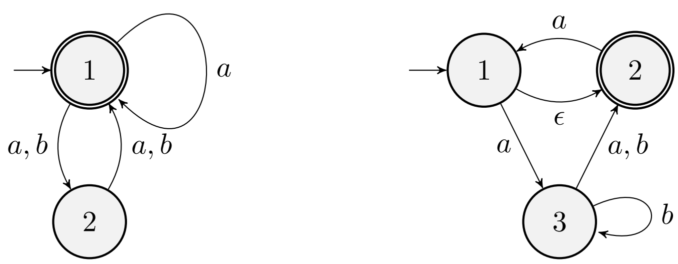


Remarque : on aurait pu fusionner les états 3 et 4 de l'automate précédent :



Exercice 5

Déterminiser les deux automates suivants.

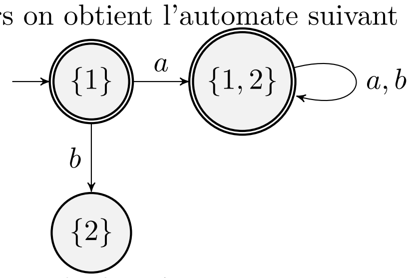


(a) (b)

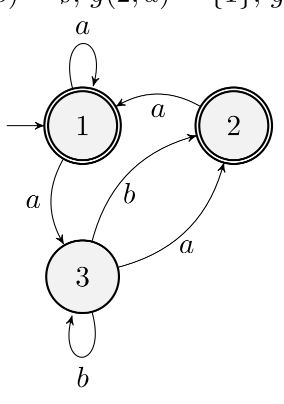
Figure 1 – Automates de l'exercice 5.

Eléments de solution de l'exercice 5

Le premier automate ne contient pas de ϵ-transition, on le déterminise donc directement. On considère l'ensemble des parties des états : les états $\{\{1\},\{2\},\{1,2\}\}$. En appliquant la méthode du cours on obtient l'automate suivant :

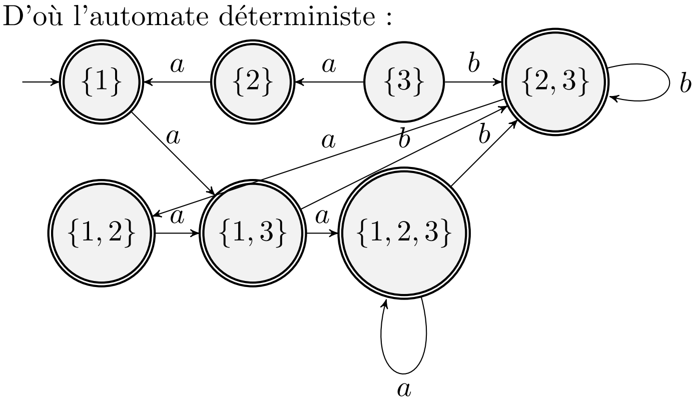


Pour le deuxième automate, on a une ϵ-transition qu'il faudra éliminer. On calcule (en suivant les notation du cours) les élément : $f(1)=\{1,2\}$, $f(2)=\{2\}$, $f(3)=\{3\}$. et on calcule $g(1,a)=\{3,1\}$, $g(1,b)=\emptyset$, $g(2,a)=\{1\}$, $g(2,b)=\emptyset$, $g(3,a)=\{2\}$, $g(3,b)=\{2,3\}$. On obtient donc l'automate :



La deuxième étape consiste à déterminiser l'automate. Les états sont :
$\{\{1\},\{2\},\{3\},\{1,2\},\{1,3\},\{2,3\},\{1,2,3\}\}$.
$\eta(\{1\},a)=\{1,3\}$, $\eta(\{1\},b)=\emptyset$, $\eta(\{2\},a)=\{1\}$, $\eta(\{2\},b)=\emptyset$, $\eta(\{3\},a)=\{2\}$, $\eta(\{3\},b)=\{2,3\}$, $\eta(\{1,2\},a)=\{1,3\}$, $\eta(\{1,2\},b)=\emptyset$, $\eta(\{1,3\},a)=\{1,2,3\}$, $\eta(\{1,3\},b)=\{2,3\}$, $\eta(\{2,3\},a)=\{1,2\}$, $\eta(\{2,3\},b)=\{2,3\}$, $\eta(\{1,2,3\},a)=\{1,2,3\}$, $\eta(\{1,2,3\},b)=\{2,3\}$.

D'où l'automate déterministe :

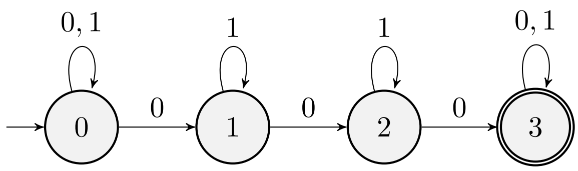


Exercice 6

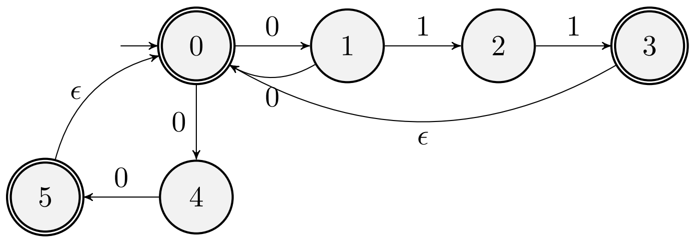
1. Construire un automate fini non déterministe équivalent à l'expression régulière $(0+1)^*000(0+1)^*$.
2. Construire un automate fini non déterministe équivalent à l'expression régulière $(((00)^*(11))+01)^*$.
3. Construire un automate fini non déterministe équivalent à l'expression régulière $\emptyset^*$.

Eléments de solution de l'exercice 6

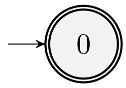
1.



2.



3. $\emptyset^*=\{\epsilon\}$ donc on obtient l'automate :



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. $(a\cup ba\cup bb)\,\Sigma^*$

Eléments de solution de l'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^* : des chaînes appartenants : a, b, ab, a^2b, a^k, b^m. Des chaînes non appartenant : $ba, aba, \ldots$
2. $a\,(ba)^*\,b$: appartenant : $ab, abab, ababab, \ldots$. Non appartenant : $bbaa, aabb, \ldots$
3. $a^* \cup b^*$: appartenant : $a, b, aa, bbb, \ldots$. Non appartenant : $ab, ba, \ldots$
4. $(aaa)^*$: appartenant : $\epsilon, aaa, aaaaaa, \ldots$. Non appartenant : $a, aa, b, aaaaa, \ldots$
5. $\Sigma^*a\Sigma^*b\Sigma^*a\Sigma^*$: appartenant : $aba, aaaabaabaab, \ldots$. Non appartement : $aaa, bbbb$.
6. $aba \cup bab$: appartenant : aba, bab. Non appartenant : $a, b, \ldots$.
7. $(\epsilon \cup a)\,b$: le seul mot appartenant est ab et aucun autre mot n'appartien. $ba, bb, \ldots$ n'appartiennent pas.
8. $(a \cup ba \cup bb)\,\Sigma^*$: appartenant : a, ba, bb (tous les mot dont le préfixe est a, ba, bb). Non appartenant : b. C'est le seul mot sur V qui n'appartient pas.

Exercice 8

Convertir les deux automates suivants en expressions régulières équivalentes.

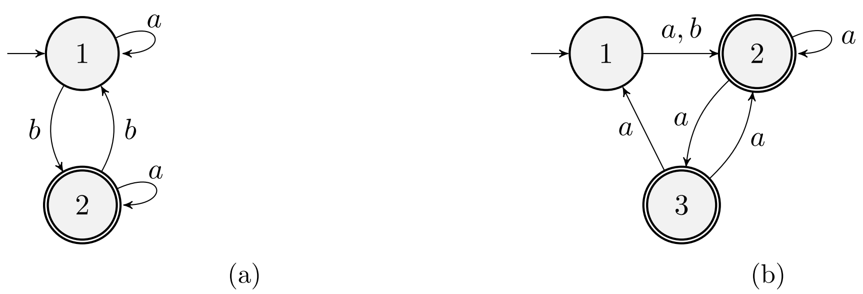


Figure 2 – Automates de l'exercice 8.

Eléments de solution de l'exercice 8

1. A partir de l'automate on déduit le système suivant : $\begin{cases} L_1 &= aL_1 + bL_2 \\ L_2 &= aL_2 + bL_1 \end{cases}$

   On peut utiliser le lemme d'Arden pour le résoudre : de l'équation 2 on déduit que $L_2 = a^*bL_1$ puis en remplaçant dans la première équation, on obtient que $L_1 = (a + ba^*b)L_1$ et donc que $L_1 = (a + ba^*b)^*$ et $L_2 = (a^*b)(a + ba^*b)^*$
2. On utilise la même méthode : on déduit les équations à partir de l'automate : $\begin{cases} L_1 &= (a+b)L_2 \\ L_2 &= aL_2 + aL_3 \\ L_3 &= aL_1 + aL_3 \end{cases}$

   Ensuite il suffit de remplacer L_1 par $(a + b)L_2$ dans l'équation 3 (on se retrouve avec un système de 2 équations à 2 variables, puis on recalcule L_2 en fonction de L_3) et on résout le tout comme pour la question précédente. $L_2 = a^*L_3$ puis $L_3 = ((a(a + b)a^*) + a)^*$ et donc $L_2 = a^*(((a(a + b)a^*) + a)^*)$ et $L_1(a + b) = a^*(((a(a + b)a^*) + a)^*)$.

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. $A_3 = \{a^{2^n} | n \geq 0\}$.

Eléments de solution de l'exercice 10

La preuve utilise le lemme de la pompe.

Rappel du lemme de la pompe : Soit L un langage régulier. Alors il existe $n \geq 1$ tel que pour tout $z \in L$, si $|z| \geq n$ alors z est de la forme uvw, où :
— $|v| \geq 1$;
— $|uv| \leq n$;
— $\forall i \geq 0$, $uv^iw \in L$.

1. Idée de la preuve. Le lemme de la pompe implique qu'on peut *pomper* (itérer un nombre de fois quelconque) à l'intérieur d'un mot de A_1 tout en restant dans A_1. Or si on prend une sous-chaine v d'un mot de A_1, on l'un des cas suivants :
   — z est constitué exclusivement de 0 (resp. de 1, resp. de 2). Le pompage va augmenter le nombre de 0 (resp. de 1, resp. de 2) sans modifier le nombre des autres symboles ce qui est contradictoire avec la définition de A_1
   — z est constitué d'au moins de symboles différents (par exemple 0 et 1). Le pompage va entrainer d'avoir des alternances de 0 et de 1 (des 0 suivis de 1 puis de nouveau des 0) ce qui est contradictoire avec la définition de A_1.

   Conclusion : A_1 n'est pas régulier. Il s'agit de la même preuve vue en cours pour prouver que $A = a^n b^n | n \in \mathbb{N}$ n'est pas régulier.
2. Même technique de preuve.
3. Supposons que A_3 est un langage régulier. Il existe donc $n \geq 1$ tel que pour tout $z \in L$, si $|z| \geq n$ alors z est de la forme uvw. Soit $z \in A_3$ tel que $|z| \geq n$. Donc $z = uvw = a^{2^m}$. donc $z = a^\alpha a^\beta a^\gamma$ et pour tout $i \geq 0$, $a^\alpha a^{i*\beta} a^\gamma \in A_3$. Il existe donc α, β, γ tels que $\forall i \geq 0 \exists k, \alpha + i\beta + \gamma = 2^k$. Donc pour $i = 0$, $\alpha + \gamma = 2^{k_0}$ et pour $i = 1$, $\alpha + \beta + \gamma = 2^{k_1}$.

   Si $\alpha + \gamma = 1$ alors on a $\forall i, 1 + i\beta \in 2^{\mathbb{N}^*}$ ce qui est absurde car pour $i = 2$ $1 + 2\beta$ est impair.

   Si $\alpha + \gamma \neq 1$ alors le seul cas pour que $\alpha + \gamma + x$ soit une puissance de 2 est que $x = \alpha + \gamma$ et donc $\beta = \alpha + \gamma$ et donc $\alpha + 2\beta + \gamma = 3\beta$ ce qui est absurde car 3β n'est pas une puissance de 2.

   Conclusion : A_3 n'est pas un langage régulier.

Exercice 11

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

Eléments de solution de l'exercice 11

1. Il est facile de voir (et de démontrer) que $B_n = ((a^n))^*$ et donc B_n est régulier.
2. pour un entier n on construit un automate d'état fini sur le vocabulaire $V = \{0, 1\}$ de n états, où chaque état i reconnait les entiers codés en binaire qui sont congrus à i modulo n. Le nombre d'états est fini et on a donc un AFD.

Exercice 12

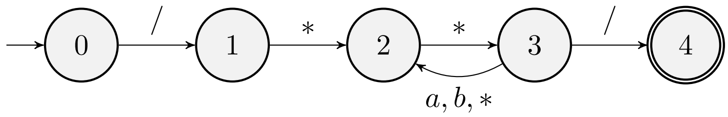
Soit $\Sigma = \{0, 1, +, =\}$ et $ADD = \{x = y+z | x, y, z$ sont des entiers binaires et x est la somme de y et de $z\}$. Montrer que ADD n'est pas régulier.

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.

Eléments de solution de l'exercice 13

1. 

2. L'expression régulière est immédiate à partir de l'automate.