THÉORIE DES LANGAGES, AUTOMATES FINIS ET GRAMMAIRES

1 Chapitre 1 : Introduction aux langages

1.1 Définitions et rappels de cours

✓ Définition

* Un alphabet est un ensemble fini de symboles ou lettres ou caractères.
* Un mot est une suite finie de caractères.
* La longueur d'un mot est le nombre de caractères qui le constituent. La longueur du mot m est notée $|m|$.
* Le mot vide est noté ε, il est de longueur 0.

Notation

* L'ensemble des mots que l'on peut écrire sur un alphabet Σ est noté Σ^*.
* Si $m \in \Sigma^*$ et $m \neq \varepsilon$, alors $m = (m_i)_{1\leq i\leq k}$ où $m_i \in \Sigma$ et $k \in \mathbb{N}^*$, avec $|m| = k$.

✓ Définition (Concaténation)

Soient $a = (a_i)_{1\leq i\leq k}$ et $b = (b_i)_{1\leq i\leq h}$ deux mots de Σ^*. La concaténation de a et b est le mot $m = a\cdot b = ab$ avec $m = (m_i)_{1\leq i\leq k+h}$, où $m_i = a_i$ pour $1 \leq i \leq k$ et $m_i = b_i$ pour $k+1 \leq i \leq k+h$.
On a $|m| = |a| + |b|$, et $a\cdot\varepsilon = \varepsilon\cdot a = a$.
Pour tout mot a et pour tout entier $n \geq 0$, on définit a^n par $a^0 = \varepsilon$ et $a^{n+1} = a^n \cdot a = a\cdot a^n$. On a $|a^n| = n|a|$.

✓ Définition

* Un langage sur l'alphabet Σ est un sous-ensemble de Σ^*.
* Un langage qui possède un nombre fini de mots est appelé langage fini.
* Opérations sur les langages : union, intersection, complémentation, concaténation.
* La concaténation de deux langages est déduite de la définition de la concaténation de mots. Soient L et L' deux langages sur Σ. Alors $L\cdot L'$ est le langage sur Σ défini par

$$L\cdot L' = \{m\cdot m'/m \in L \text{ et } m' \in L'\}.$$

* Pour tout langage L et pour tout entier $n \geq 0$, on définit L^n par $L^0 = \{\varepsilon\}$ et $L^{n+1} = L^n\cdot L = L\cdot L^n$.
* On définit enfin l'étoile de Kleene par $L^* = \bigcup_{n\in\mathbb{N}} L^n$ et $L^+ = \bigcup_{n\in\mathbb{N}^*} L^n$.

Notation

* Si $L = \{a\}$, on notera $L\cdot L' = \{a\}\cdot L' = aL'$, $L^n = \{a\}^n = a^n$, $L^* = \{a^n/n \in \mathbb{N}\} = \{a\}^* = a^*$.
* Soient L, L' deux langages sur Σ. On note $L \setminus L'$ le langage des mots de L qui ne sont pas dans L'.

✓ **Définition**

La classe des langages réguliers est la classe la plus petite qui contienne les langages finis et qui soit stable par les opérations de réunion, concaténation, étoile de Kleene.
Autrement dit, les seuls langages réguliers sont les langages qu'il est possible de construire à l'aide des deux règles suivantes :

1. tout langage fini est régulier ;
2. si L et L' sont réguliers, alors $L \cup L'$, $L \cdot L'$ et L^* sont réguliers.

Proposition

Soit Σ un alphabet fini. Une expression régulière est une expression qui peut être obtenue par application d'une ou plusieurs des règles suivantes :

1. $\emptyset$ est une expression régulière, *(désigne le langage vide)*
2. ε est une expression régulière, *(désigne le langage $\{\varepsilon\}$)*
3. si x est un caractère de Σ, alors x est une expression régulière, *(désigne le langage $\{x\}$)*
4. si r et s sont des expressions régulières, *(désignant les langages $L(r)$ et $L(s)$)*
   - ⋆ alors $r|s$ est une expression régulière, *(désigne $L(r) \cup L(s)$)*
   - ⋆ alors $r \cdot s$ est une expression régulière, *(désigne $L(r) \cdot L(s)$)*
   - ⋆ alors r^* est une expression régulière, *(désigne $[L(r)]^*$)*

Les langages réguliers sont les langages qui peuvent être décrits par une expression régulière.

1.2 Exercices

Exercice 1.1

Soient les langages sur $\Sigma = \{a, b, c\}$

$$L_1 = \{a, ab, ba\}, \qquad L_2 = \{b, ab\}\ , \qquad L_3 = \{cab, ba\}, \qquad L_4 = \{aa, ba, \varepsilon\}.$$

Décrire chacun des langages suivants :

- ⋆ $L_1 \cup L_2$;
- ⋆ $L_1 \cap L_2$;
- ⋆ $L_1 \setminus L_2$;
- ⋆ $L_2 \setminus L_1$;
- ⋆ L_1^2 ;
- ⋆ L_2^2 ;
- ⋆ L_2^* ;
- ⋆ $L_1 \cdot L_2$;
- ⋆ $L_2 \cdot L_1$;
- ⋆ $L_3 \cup L_4$;
- ⋆ $L_3 \cap L_4$;
- ⋆ $L_3 \setminus L_4$;
- ⋆ $L_4 \setminus L_3$;
- ⋆ L_3^2 ;
- ⋆ L_4^2 ;
- ⋆ L_4^* ;
- ⋆ $L_3 \cdot L_4$;
- ⋆ $L_4 \cdot L_3$.

Exercice 1.2

Soit $\Sigma = \{a, b\}$.

1. Soit le langage $L = \{a\}$. Quel est le nombre de mots de L^6 ? de L^k où k est un entier positif non nul ? de $\bigcup_{1 \le k \le n} L^k$ où n et k sont des entiers positifs non nuls ? de $\bigcup_{0 \le k \le n} L^k$?
2. Mêmes questions si $L = \{\varepsilon, a\}$.
3. Mêmes questions si $L = \{a, aa\}$. L^6 a^6 $a^5(aa)$
4. Mêmes questions si $L = \{a, b\}$.

Exercice 1.3

Décrire le langage des entiers naturels sur l'alphabet $\Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$.
Attention : on n'autorise pas les écritures d'entiers non nuls commençant par 0.

Exercice 1.4

Soit l'alphabet $\Sigma = \{1, 2, 3\}$ et les langages $L_1 = \{x \in \Sigma^*/x \text{ pair}\}$, $L_2 = \{x \in \Sigma^*/x \leq 20\}$ et $L_3 = \{1, 3\}$.
Décrire les langages L_1, $\Sigma^* \setminus L_1$, L_1^*, L_2, $L_1 \cap L_2$, $L_1 \cup L_2$, $\Sigma^* L_3$.

Exercice 1.5

Soit $\Sigma = \{a, b\}$ et les langages

$$L_1 = \{a^n b^p/n, p \in \mathbb{N}\}, \quad L_2 = \{a^n b^n/n \in \mathbb{N}\}, \quad L_3 = \{a^n/n \in \mathbb{N}\} = a^*, \quad L_4 = \{b^n/n \in \mathbb{N}\} = b^* .$$

1. Donner des mots de chacun de ces langages.
2. Déterminer $L_1 \cap L_2$.
3. Déterminer $L_1 \setminus L_3$, puis $L_2 \setminus L_4$.
4. Pourquoi peut-on écrire : $L_1 = L_3 \cdot L_4$, c'est-à-dire $L_1 = a^* b^*$? Peut-on aussi écrire $L_2 = a^* b^*$?

Exercice 1.6

Soit $\Sigma = \{a, b, c\}$. L_1 désigne le langage des mots sur Σ qui contiennent au moins un a et L_2 le langage des mots sur Σ qui se terminent par b.

1. Donner des exemples de mots appartenant à L_1, à L_2 et de mots n'appartenant pas à L_1, n'appartenant pas à L_2.
2. Caractériser les mots appartenant à $L_3 = L_1 \cup L_2$ et donner des exemples de mots appartenant à L_3 et de mots n'appartenant pas à L_3.
3. Reprendre la question 2 en remplaçant L_3 par $L_4 = L_1 \cap L_2$; $L_5 = \bar{L_1}$; $L_6 = \bar{L_2}$; $L_7 = \bar{L_3}$; $L_8 = \bar{L_4}$.
4. Donner des expressions régulières de chacun des langages L_i, $1 \leq i \leq 8$.

Exercice 1.7

Soit $\Sigma = \{a, b\}$. Montrer les résultats suivants :

1. $\Sigma^* \Sigma^* = \Sigma^*$
2. $(a^*)^* = a^*$
3. $(a^*|b^*)^* = \Sigma^*$
4. $(a^* b^*)^* = \Sigma^*$
5. $\Sigma^* \neq a^* b^*$
6. $(ab)^* a = a(ba)^*$
7. $\Sigma^* = \Sigma^* ab \Sigma^* | b^* a^*$
8. $b^* | ab^* = (\varepsilon | a) b^*$

Exercice 1.8
Soit l'alphabet $\Sigma = \{a\}$. On note $L_1 = a^*$, $L_2 = (a^*)^3$, $L_3 = (a^2)^*$, $L_4 = a(a^2)^*$, $L_5 = (a^3)^*$.

1. Montrer que le mot $m_1 = a$ appartient à L_1, à L_2 et à L_4 mais pas à L_3 ni à L_5.
2. Soient les mots $m_2 = \varepsilon$, $m_3 = a^2$, $m_4 = a^7$, $m_5 = a^{12}$, $m_6 = a^{14}$, $m_7 = a^{15}$. Dire pour chacun de ces mots à quel(s) langage(s) L_i $(1 \leq i \leq 5)$ il appartient. Justifier les réponses.
3. Montrer que
   (a) $L_1 = L_2$
   (b) $L_2 \neq L_5$
   (c) $L_3 \cap L_4 = \emptyset$ et $L_3 \cup L_4 = L_1$
4. Quels sont les mots du langage $L_3 \cap L_5$? du langage $L_4 \cap L_5$?

Exercice 1.9
Soient les langages sur $\Sigma = \{a, b\}$

$$L_1 = (ab)^*, \quad L_2 = (ab)^*a, \quad L_3 = b(ab)^*, \quad L_4 = (ba)^*b, \quad L_5 = a^*b^*, \quad L_6 = (aa)^*\ b^*.$$

Pour chacun des mots suivants m_i $(1 \leq i \leq 8)$, dire auxquels de ces six langages L_j $(1 \leq j \leq 6)$, définis ci-dessus il appartient et à quels langages il n'appartient pas

$$m_1 = ab^2, \quad m_2 = a^2b, \quad m_3 = abab, \quad m_4 = bab, \quad m_5 = ab, \quad m_6 = b, \quad m_7 = aba, \quad m_8 = \varepsilon.$$

Exercice 1.10
Soient les langages sur $\Sigma = \{a, b\}$ définis par $L_1 = a^*b(a|b)^*b$ et $L_2 = b^*ba^*b$.

1. Pour chacun des mots suivants m_i $(1 \leq i \leq 6)$, dire s'il appartient à L_1 et s'il appartient à L_2

$$m_1 = ab^2, \quad m_2 = b^5, \quad m_3 = a^2b, \quad m_4 = b^4a^7b^2, \quad m_5 = bababa^8b, \quad m_6 = ba^6b.$$

2. Ecrire l'expression générale d'un mot de L_2 et en déduire que si un mot appartient à L_2, alors il appartient à L_1. On en conclura donc que L_2 est inclus dans L_1.
3. Pourquoi peut-on dire que $L_1 \cup L_2 = L_1$?

Exercice 1.11
Soit l'alphabet $\Sigma = \{a, b, c\}$. Pour chacun des langages suivants, proposer une expression régulière le décrivant :

1. langage des mots sur Σ commençant par a ; $a\Sigma^*$
2. langage des mots sur Σ commençant et se terminant par a ; $a\Sigma^*a$
3. langage des mots sur Σ commençant et se terminant par la même lettre ; $\Sigma\Sigma^*\Sigma$
4. langage des mots sur Σ commençant et se terminant par des lettres différentes ;
5. langage des mots sur Σ contenant au moins un a ;
6. langage des mots sur Σ contenant exactement un a ;
7. langage des mots sur Σ ne contenant aucun a ;
8. langage des mots sur Σ contenant au moins une fois le facteur aa.
9. langage des mots sur Σ contenant un nombre pair de a ;
10. langage des mots sur Σ contenant un nombre impair de a ;

a

2 Chapitre 2 : Automates finis

2.1 Définitions et lecture de mots

✓ **Définition**

Un automate fini (AF) est un quintuplet $\mathcal{A} = (\Sigma, E, D, A, \Theta)$ où

- Σ est un alphabet
- E est un ensemble fini d'éléments appelés états
- D est un sous-ensemble de E dont les éléments sont appelés états de départ (ou états initiaux ou entrées)
- A est un sous-ensemble de E dont les éléments sont appelés états acceptants (ou états accepteurs ou états terminaux)
- Θ est un sous ensemble de $E \times \Sigma \times E$ dont les éléments sont appelés transitions.

✓ **Définition**

Une transition est un triplet (e, x, f) où e et f sont des états de E et où x est un caractère de Σ. Θ sera représenté par une table ou par un graphe.

✓ **Définition**

Un automate fini déterministe est un automate fini $\mathcal{A} = (\Sigma, E, D, A, \Theta)$ tel que :

- D possède au plus un élément, c'est-à-dire qu'il y a au plus un état de départ.
- Pour tout état e de E et pour tout caractère x de Σ, il existe au plus un état f de E tel que le triplet (e, x, f) soit une transition de Θ.

✓ **Définition**

Un automate fini complet est un automate fini $\mathcal{A} = (\Sigma, E, D, A, \Theta)$ tel que :

- $D \neq \emptyset$.
- Pour tout état e de E et pour tout caractère x de Σ, il existe au moins un état f de E tel que le triplet (e, x, f) soit une transition de Σ.

✓ **Définition**

Un automate fini déterministe complet (AFdc) est un automate fini $\mathcal{A} = (\Sigma, E, D, A, \Theta)$ tel que :

- D possède exactement un élément.
- Pour tout état e de E et pour tout caractère x de Σ, il existe un et un seul état f de E tel que le triplet (e, x, f) soit une transition de Θ.

✓ **Définition**

Soit $\mathcal{A} = (\Sigma, E, D, A, \Theta)$ un automate fini. Un chemin de longueur n dans l'automate fini $\mathcal{A}$ est une suite de n transitions de Θ

$$(e_1, x_1, e_2),\ (e_2, x_2, e_3),\ (e_3, x_3, e_4), \cdots, (e_n, x_n, e_{n+1}).$$

L'état e_1 est l'état de départ de ce chemin et e_{n+1} est l'état d'arrivée de ce chemin. Le mot $m = x_1 x_2 x_3 \cdots x_n$ de Σ^* est la trace de ce chemin.

✓ **Définition**

- Un mot de Σ^* est reconnu par l'automate fini $\mathcal{A}$ s'il est la trace d'au moins un chemin dans l'automate $\mathcal{A}$, où e_1 est un état de départ de D et e_{n+1} est un état acceptant de A.
- Le langage reconnu par l'automate fini $\mathcal{A}$ est l'ensemble des mots reconnus par l'automate $\mathcal{A}$. Ce langage est noté $L(\mathcal{A})$.

Notation

- Soit F un sous-ensemble d'états de E et x un caractère de Σ. On notera $F \cdot x = \{f_1, f_2, \cdots, f_p\}$ $(p \geq 0)$ le sous-ensemble de E constitué des états f_j tels qu'il existe au moins une transition (e, x, f_j) de Θ, où e est un état de F. On a

$$F \cdot x = \bigcup_{e \in F} \{f_j / (e, x, f_j) \in \Theta\}.$$

- Attention, on peut avoir $F \cdot x = \emptyset$.
- On a $\emptyset \cdot x = \emptyset$.
- On a $F \cdot \varepsilon = F$.

✓ Définition (Lecture d'un mot $m = x_1 x_2 x_3 \cdot x_n$ de Σ^* par l'automate fini $\mathcal{A}$)

La lecture d'un mot est la suite des sous-ensembles d'états dans lequel se trouve l'automate fini au cours de la lecture de ce mot. Le dernier sous-ensemble, $D \cdot m$, est décisif. On obtient ainsi tous les états qui appartiennent aux chemins dont les états de départ sont dans D et dont m est la trace. On écrira

$$\begin{aligned} D \cdot m = D \cdot x_1 x_2 x_3 \cdots x_n &= \{e_{11}, e_{12}, \cdots, e_{1p}\} \cdot x_2 x_3 \cdots x_n, \text{ où } D \cdot x_1 = \{e_{11}, e_{12}, \cdots, e_{1p}\},\ p \geq 0, \\ &= \{e_{21}, e_{22}, \cdots, e_{2q}\} \cdot x_3 \cdots x_n, \text{ où } \{e_{11}, e_{12}, \cdots, e_{1p}\} \cdot x_2 = \{e_{21}, e_{22}, \cdots, e_{2q}\},\ q \geq 0, \\ &= \{e_{n-1,1}, e_{n-1,2}, \cdots, e_{n-1,h}\} \cdot x_n,\ h \geq 0, \\ &= \{e_{n1}, e_{n2}, \cdots, e_{nh}\} \cdot \varepsilon = \{e_{n1}, e_{n2}, \cdots, e_{nh}\},\ k \geq 0. \end{aligned}$$

$D \cdot m$ est l'ensemble des états d'arrivée de tous les chemins dans l'automate fini $\mathcal{A}$ dont l'état de départ est dans D et la trace est le mot m. Le mot m de Σ^* est reconnu par l'automate fini $\mathcal{A}$ si et seulement si $D \cdot m = \{e_{n1}, e_{n2}, \cdots, e_{nh}\} \cap A \neq \emptyset$.

☞ Remarques

- Deux automates sont équivalents si et seulement si ils reconnaissent le même langage.
- Si l'automate fini est déterministe complet, quel que soit le mot de Σ^*, tous les sous-ensembles d'états rencontrés dans la lecture de ce mot sont des singletons, c'est-à-dire des sous-ensembles possédant un unique état.
- Si l'automate fini n'est pas complet, on peut trouver l'ensemble vide au cours de la lecture d'un mot.
- Si l'automate fini n'est pas déterministe, certains sous-ensembles d'états peuvent posséder plusieurs éléments.

✓ Définition

Un langage L est reconnaissable par un automate fini si et seulement s'il existe un automate fini $\mathcal{A}$ tel que $L(\mathcal{A}) = L$.

Théorème

- Tout langage régulier est reconnaissable par un automate fini.
- Tout langage reconnu par un automate fini est régulier.

☞ Exemple

Exemple de langage non reconnaissable par un AF : $L = \{a^k b^k / k \in \mathbb{N}\}$.

Notation

Soit L un langage sur l'alphabet Σ. $\bar{L}$ désigne le langage complémentaire de L dans Σ^*

$$\bar{L} = \Sigma^* \setminus L = \{m/m \in \Sigma^* \text{ et } m \notin L\}.$$

$\bar{L}$ est donc le langage des mots de Σ^* qui n'appartiennent pas à L.

Proposition

Soit $\mathcal{A} = (\Sigma, E, D, A, \Theta)$ un automate fini déterministe complet (AFdc) qui reconnaît le langage L : $L(\mathcal{A}) = L$. On définit $\mathcal{A}' = (\Sigma, E, D, \bar{A}, \Theta)$, où $\bar{A} = E \setminus A = \{e/e \in E \text{ et } e \notin A\}$ est le complémentaire de A dans E.

Alors tout m de Σ^* est reconnu par $\mathcal{A}'$ si et seulement si m n'est pas reconnu par $\mathcal{A}$, c'est-à-dire $L(\mathcal{A}') = \bar{L}$.

✓ **Définition**

Un automate fini $\mathcal{A} = (\Sigma, E, D, A, \Theta)$ avec des ε-transitions est un automate fini non déterministe qui possède des transitions étiquetées par le mot vide : (e_i, ε, e_j), où e_i et e_j sont des états de E.

Notation

Soit $e_j \in E$ et $x \in \Sigma$, alors

$$\theta(e_j, x) = \{\{e_j\} \cdot x, \{e_j \cdot x\}\varepsilon^n \text{ où } n \in \mathbb{N}\} \subset E$$

est l'ensemble des états atteignables à partir de e_j en utilisant la transition x et les ε-transitions, car quel que soit $x \in \Sigma$, $x = x\varepsilon^n$.

Proposition

Tout AF avec des ε-transitions est équivalent à un AFdc sans ε-transitions.

2.2 Exercices

Exercice 2.1

Soit l'alphabet $\Sigma = \{a, b\}$.

1. Dessiner l'automate fini $\mathcal{A} = (\Sigma, E, D, A, \Theta)$ où $E = \{1, 2, 3\}$, $D = \{1\}$, $A = \{3\}$, et Θ est défini par sa table de transition :

Θ	a	b
- 1	1	2
2	3	2
3 +	3	3

2. Cet automate fini est-il déterministe ? complet ?
3. Comment cet automate fini lit-il les mots suivants :

$$m_1 = ba,\ m_2 = ab,\ m_3 = \varepsilon,\ m_4 = a,\ m_5 = b,\ m_6 = baba,\ m_7 = abab,\ m_8 = aabbaa.$$

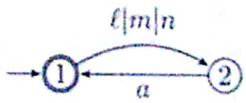
4. Quel langage $L(\mathcal{A})$ reconnaît-il ?

Exercice 2.2

Soit l'alphabet $\Sigma = \{a, \ell, m, n\}$.
On considère les six automates finis $\mathcal{A}_i = (\Sigma, E, D_i, A_i, \Theta)$, $1 \leq i \leq 6$ définis ainsi :
$E = \{1, 2, 3\}$, $D_i = \{1\}$ pour $i \neq 5$ et $D_5 = \{2\}$, $A_1 = \{1\}$, $A_2 = \{2\}$, $A_3 = \emptyset$, $A_4 = \{1, 2, 3\}$,
$A_5 = \{1\}$, $A_6 = \{3\}$, et

Θ	a	ℓ	m	n
1	3	2	2	2
2	1	3	3	3
3	3	3	3	3

1. On considère l'automate $\mathcal{A}_1$.
   (a) Cet automate est-il déterministe ? complet ?
   (b) Comment sont lus les mots ℓama ? $maman$? $\ell aam\ell\ell$? ε ?
   (c) Quel est le langage reconnu par $\mathcal{A}_1$?
   (d) Soit l'automate $\mathcal{A}'_1$ défini par son graphe :



   Quel est le langage reconnu par $\mathcal{A}'_1$? Cet automate est-il déterministe ? complet ?
2. Pour chacun des automates $\mathcal{A}_i$, $2 \leq i \leq 6$, donner une expression régulière du langage $L(\mathcal{A}_i)$ qu'il reconnaît.

Exercice 2.3

Soit l'alphabet $\Sigma = \{a, b, c\}$. On considère les quatre automates finis $\mathcal{A}_i = (\Sigma, E, D, A_i, \Theta_i)$, $1 \leq i \leq 4$, définis par $E = \{1, 2, 3\}$, $D = \{1\}$, $A_1 = \{1\}$, $A_2 = \{2\}$, $A_3 = A_4 = \{3\}$, pour tout $i \leq 3$

Θ_i	a	b	c
- 1	1	2	
2		1	3
3			

et

Θ_4	a	b	c
- 1	1	2	
2		1	3
3 +			1

1. Pour chacun de ces automates, dire s'il est déterministe ? complet ?
2. Comment l'automate fini $\mathcal{A}_i$, $1 \leq i \leq 3$, lit-il :

$$m_1 = ab,\ m_2 = abc,\ m_3 = abb,\ m_4 = abba,\ m_5 = bcbc,\ m_6 = bab,\ m_7 = abbbc,\ m_8 = bccbc.$$

3. Quel langage $L(\mathcal{A}_i)$ reconnaît-il ?

Exercice 2.4

Construire des automates finis reconnaissant les langages réguliers de l'exercice 1.11. Ces automates finis ne seront pas nécessairement déterministes ou complets.

Exercice 2.5

Soit les langages sur $\Sigma = \{a, b\}$

$$L_1 = (ab)^*,\quad L_2 = (ab)^*a,\quad L_3 = b(ab)^*,\quad L_4 = (ba)^*b,\quad L_5 = a^*b^*,\quad L_6 = (aa)^*\, b^*.$$

1. Pour chacun des six langages L_j, $1 \leq j \leq 6$, dessiner le graphe d'un automate fini reconnaissant ce langage.
2. Comment chacun de ces automates finis lit-il les mots :

$$m_1 = ab^2,\ m_2 = a^2b,\ m_3 = abab,\ m_4 = bab,\ m_5 = ab\ ,\ m_6 = b\ ,\ m_7 = aba\ ,\ m_8 = \varepsilon.$$

Comparer avec les résultats trouvés dans l'exercice 1.9.

Exercice 2.3

Soit l'alphabet $\Sigma = \{a, b, c\}$. On considère les quatre automates finis $\mathcal{A}_i = (\Sigma, E, D, A_i, \Theta_i)$, $1 \leq i \leq 4$, définis par $E = \{1, 2, 3\}$, $D = \{1\}$, $A_1 = \{1\}$, $A_2 = \{2\}$, $A_3 = A_4 = \{3\}$, pour tout $i \leq 3$

Θ_i	a	b	c
- 1	1	2	
2		1	3
3			

et

Θ_4	a	b	c
- 1	1	2	
2		1	3
3 +			1

1. Pour chacun de ces automates, dire s'il est déterministe ? complet ?
2. Comment l'automate fini $\mathcal{A}_i$, $1 \leq i \leq 3$, lit-il :

$$m_1 = ab,\ m_2 = abc,\ m_3 = abb,\ m_4 = abba,\ m_5 = bcbc,\ m_6 = bab,\ m_7 = abbbc,\ m_8 = bccbc.$$

3. Quel langage $L(\mathcal{A}_i)$ reconnaît-il ?

Exercice 2.4

Construire des automates finis reconnaissant les langages réguliers de l'exercice 1.11. Ces automates finis ne seront pas nécessairement déterministes ou complets.

Exercice 2.5

Soit les langages sur $\Sigma = \{a, b\}$

$$L_1 = (ab)^*, \quad L_2 = (ab)^*a, \quad L_3 = b(ab)^*, \quad L_4 = (ba)^*b, \quad L_5 = a^*b^*, \quad L_6 = (aa)^*\ b^*.$$

1. Pour chacun des six langages L_j, $1 \leq j \leq 6$, dessiner le graphe d'un automate fini reconnaissant ce langage.
2. Comment chacun de ces automates finis lit-il les mots :

$$m_1 = ab^2,\ m_2 = a^2b,\ m_3 = abab,\ m_4 = bab,\ m_5 = ab\ ,\ m_6 = b\ ,\ m_7 = aba\ ,\ m_8 = \varepsilon.$$

Comparer avec les résultats trouvés dans l'exercice 1.9.

Exercice 2.6

Soit l'alphabet $\Sigma = \{a, b, c\}$.

1. Soit L_1 le langage des mots sur Σ qui contiennent le mot bac et se terminent par a.
   (a) Ecrire une expression régulière de L_1.
   (b) Déduire de cette expression régulière un automate fini $\mathcal{A}_1$ reconnaissant ce langage : $L(\mathcal{A}_1) = L_1$.
   (c) Indiquer comment cet automate fini $\mathcal{A}_1$ lit chacun des mots suivants :

   $$m_1 = abaca, \quad m_2 = abacb, \quad m_3 = abacbaca, \quad m_4 = bcaca\ , \quad m_5 = bcab\ , \quad m_6 = \varepsilon.$$

2. Soit L_2 le langage des mots sur Σ qui contiennent le mot bac ou se terminent par a.
   (a) Ecrire une expression régulière de L_2.
   (b) Déduire de cette expression régulière un automate fini $\mathcal{A}_2$ reconnaissant ce langage : $L(\mathcal{A}_2) = L_2$.
   (c) Indiquer comment cet automate fini $\mathcal{A}_2$ lit chacun des mots suivants :

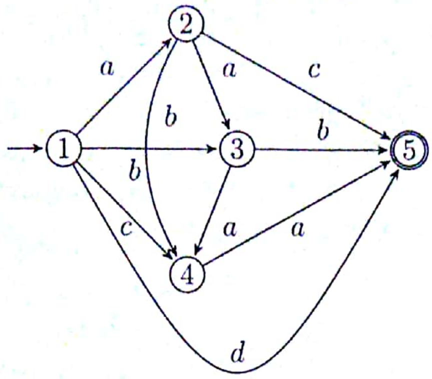
   $$m_1 = abaca, \quad m_2 = abacb, \quad m_3 = abacbaca, \quad m_4 = bcaa\ , \quad m_5 = bcab\ , \quad m_6 = \varepsilon.$$

3. Soit L_3 le langage des mots sur Σ qui commencent par ab ou se terminent par ca.
   (a) Ecrire une expression régulière de L_3.
   (b) Déduire de cette expression régulière un automate fini $\mathcal{A}_3$ reconnaissant ce langage : $L(\mathcal{A}_3) = L_3$. Cet automate a 2 états de départ.
   (c) Indiquer comment cet automate fini $\mathcal{A}_3$ lit chacun des mots suivants :

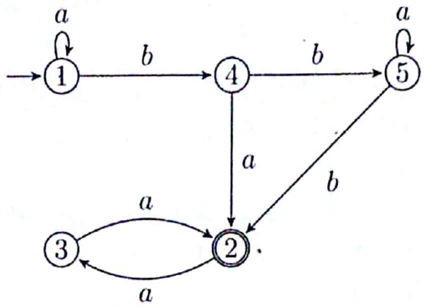
   $$m_1 = abaca, \quad m_2 = abacb, \quad m_3 = abacbaca, \quad m_4 = bcaa\ , \quad m_5 = bcab\ , \quad m_6 = \varepsilon.$$

Exercice 2.7

1. Déterminer le nombre de mots de longueur 2 reconnu par l'automate suivant, puis le nombre de mots de longueur 3, 4 et 5.



2. Pour l'automate suivant, déterminer le nombre de mots de longueur 5 et le nombre de mots de longueur 10.



On donne les puissances successives suivantes de la matrice $A = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix}$:

$$A^2 = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 \end{pmatrix}, A^3 = \begin{pmatrix} 1 & 2 & 1 & 1 & 2 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 2 & 1 & 0 & 1 \\ 0 & 2 & 1 & 0 & 1 \end{pmatrix}, A^4 = \begin{pmatrix} 1 & 4 & 2 & 1 & 3 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 2 & 2 & 0 & 1 \\ 0 & 2 & 2 & 0 & 1 \end{pmatrix}, A^5 = \begin{pmatrix} 1 & 6 & 4 & 1 & 4 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 3 & 2 & 0 & 1 \\ 0 & 3 & 2 & 0 & 1 \end{pmatrix}, A^6 = \begin{pmatrix} 1 & 9 & 6 & 1 & 5 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 3 & 3 & 0 & 1 \\ 0 & 3 & 3 & 0 & 1 \end{pmatrix},$$

$$A^7 = \begin{pmatrix} 1 & 12 & 9 & 1 & 6 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 4 & 3 & 0 & 1 \\ 0 & 4 & 3 & 0 & 1 \end{pmatrix}, A^8 = \begin{pmatrix} 1 & 16 & 12 & 1 & 7 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 4 & 4 & 0 & 1 \\ 0 & 4 & 4 & 0 & 1 \end{pmatrix}, A^9 = \begin{pmatrix} 1 & 20 & 16 & 1 & 8 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 5 & 4 & 0 & 1 \\ 0 & 5 & 4 & 0 & 1 \end{pmatrix}, A^{10} = \begin{pmatrix} 1 & 25 & 20 & 1 & 9 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 5 & 5 & 0 & 1 \\ 0 & 5 & 5 & 0 & 1 \end{pmatrix}$$

3. Même question que précédemment si l'état 5 est également acceptant.

3 Algorithmes de simplification d'automates finis

3.1 Rappels de cours

3.1.1 Automates finis déterministes complets

Proposition (Théorème de transformation d'un AF en AFdc)

Soit $\mathcal{A} = (\Sigma, E, D, A, \Theta)$ un automate fini. Soit $\hat{\mathcal{A}} = (\Sigma, \hat{E}, \hat{D}, \hat{A}, \hat{\Theta})$, où

* $\hat{E} = \{F/F \subset E \text{ et } \exists m \in \Sigma^*, D \cdot m = F\}$: les états de $\hat{\mathcal{A}}$ sont des sous-ensembles d'états de $\mathcal{A}$, c'est-à-dire que les éléments de $\hat{E}$ sont des sous-ensembles de E.
* $\hat{D} = \{D\}$
* $\hat{A} = \{F/F \in \hat{E} \text{ et } F \cap A \neq \emptyset\}$
* $\hat{\Theta} = \{(F, x, F \cdot x)/F \in \hat{E} \text{ et } x \in \Sigma\}$

Alors $\hat{\mathcal{A}}$ est un automate fini déterministe complet qui reconnaît le même langage que $\mathcal{A}$. On dit que $\hat{\mathcal{A}}$ est équivalent à $\mathcal{A}$ car $L(\mathcal{A}) = L(\hat{\mathcal{A}})$.

Algorithme (Algorithme de "subset construction" pour construire l'AFdc $\hat{\mathcal{A}}$)

Construction des états de $\hat{\mathcal{A}}$ et de la table de transition $\hat{\Theta}$:

1. Initialisation : l'état de départ de $\hat{\mathcal{A}}$ est le sous-ensemble D des états de départs de $\mathcal{A}$.
2. Ecrire, pour tout x de Σ, les états $D \cdot x$, sous-ensembles d'états de $\mathcal{A}$. Ces états sont impérativement numérotés par ordre d'apparition.
3. Pour chaque nouvel état e', écrire, pour tout x de Σ, les états $e' \cdot x$, sous-ensembles d'états de $\mathcal{A}$, états que l'on numérote impérativement par ordre d'apparition.
4. Lorsque l'on ne crée plus de nouvel état, la construction est terminée.
5. Les états acceptants de $\hat{\mathcal{A}}$ sont les états de $\hat{E}$ qui contiennent au moins un état acceptant de $\mathcal{A}$.

Remarques

* Si pour un certain e' et un certain x, on trouve $e' \cdot x = \emptyset$, alors $\emptyset$ est un nouvel état de $\hat{E}$, qui aura le numéro correspondant à son ordre de création. Pour tout x de Σ, on aura $\emptyset \cdot x = \emptyset$.
* Si $\mathcal{A}$ possède n états, c'est-à-dire si E a n éléments, alors $\hat{\mathcal{A}}$ peut avoir jusqu'à 2^n états.
* $\hat{D} = \{D\}$, l'automate $\hat{\mathcal{A}}$ a donc bien un unique état de départ, $\hat{D}$. De plus, quel que soit l'état e de $\hat{E}$ et quel que soit le caractère x de Σ, alors $\{e\} \cdot x = \{f\}$ est un singleton de $\hat{E}$. Autrement dit, (e, x, f) est l'unique transition de $\hat{\Theta}$ commençant par (e, x). L'AF $\hat{\mathcal{A}}$ est donc bien déterministe complet.

3.1.2 Automates finis déterministes complets minimaux

Définition

Soit $\mathcal{A} = (\Sigma, E, D, A, \Theta)$ un AFdc. Parmi tous les AFdc équivalents à $\mathcal{A}$, $\mathcal{A}' = (\Sigma, E', D', A', \Theta')$ est un AFdc minimal si et seulement si tout AFdc équivalent à $\mathcal{A}$ a un nombre d'états égal ou supérieur au nombre d'états de E'.

Proposition

On peut démontrer l'unicité de l'automate minimal équivalent à $\mathcal{A}$.

Algorithme (Algorithme de Nérode pour construire l'AFdc minimal $\mathcal{A}'$)
Construction de l'AFdc minimal $\mathcal{A}' = (\Sigma, E', D', A', \Theta')$ tel que $L(\mathcal{A}') = L(\mathcal{A})$.

1. $\mathcal{A} = (\Sigma, E, D, A, \Theta)$ est supposé être un AFdc, sinon on le rend AFdc.
2. Soit $\bar{A}$ le sous-ensemble des états non-acceptants de E, et soit A le sous-ensemble des états acceptants de E.

   On constitue la liste L_0 des paires d'états non-acceptants et des paires d'états acceptants.
3. On construit la table de transition suivante : pour toute paire d'états $\{e, e'\}$ de L_0 et pour tout x de Σ, on écrit $\{e, e'\} \cdot x$.
4. **Principe de l'algorithme :**
   (a) Pour chaque paire d'états d'une liste L, si pour tout x de Σ, $\{e, e'\} \cdot x$ est un singleton ou bien un élément de L, alors $\{e, e'\}$ reste dans la liste L.
   (b) Sinon, c'est-à-dire s'il existe au moins un x de Σ tel que $\{e, e'\} \cdot x$ n'est ni un singleton ni un élément de la liste L, alors $\{e, e'\}$ est une paire d'états non regroupables, elle est retirée de la liste L.

   On pose $L = L_0$.
   Tant que L contient des paires d'états non regroupables,
   Faire $L = L \setminus \{$paires d'états non regroupables$\}$
   Renvoyer L
5. Pour toute paire d'états $\{e, e'\}$ de la liste L obtenue en sortie de l'algorithme ci-dessus, on supprime l'état e' de l'ensemble des états E. On dit que les états e et e' de E sont regroupables.
6. L'AFdc minimal $\mathcal{A}' = (\Sigma, E', D', A', \Theta')$ est obtenu en renumérotant les états et en déduisant la nouvelle table de transition Θ'.

Remarques

* Si pour une paire d'états $\{e, e'\}$, et pour tout x de Σ, $\{e, e'\} \cdot x$ est un singleton, alors cette paire d'états sera toujours gardée dans les listes L successives, les états e et e' sont donc regroupables.
* Si la liste finale L vérifie $L = \emptyset$, alors l'automate $\mathcal{A} = (\Sigma, E, D, A, \Theta)$ est minimal.

3.1.3 Automates finis à ε-transitions

Algorithme (Construction d'un AFdc $\check{\mathcal{A}}$ sans ε-transitions équivalent à un AF $\mathcal{A}$)
On cherche $\check{\mathcal{A}} = (\Sigma, \check{E}, \check{D}, \check{A}, \check{\Theta})$ sans ε-transitions équivalent à l'automate fini $\mathcal{A} = (\Sigma, E, D, A, \Theta)$ avec des ε-transitions. Les états de $\check{\mathcal{A}}$ sont des sous-ensembles d'états de $\mathcal{A}$, c'est-à-dire que les états de $\check{E}$ sont des sous-ensembles d'états de E. $\check{D}$ est un singleton de $\check{E}$, c'est-à-dire que $\check{\mathcal{A}}$ a un seul état de départ F_1 construit ainsi :

$$F_1 = \{e_i \in D, \{e_i\} \cdot \varepsilon^n \text{ où } n \in \mathbb{N}\}.$$

Si $F = \{e_{i_k} \in E / i_1 \leq i_k \leq i_h\} \in \check{E}$ est un état de $\check{\mathcal{A}}$ composé de h ($h \in \mathbb{N}^*$) états de E, alors

$$F \cdot x = F' = \bigcup_{i_1 \leq i_k \leq i_h} \theta(e_{i_k} \cdot x) \in \check{E}.$$

Les états acceptants de $\check{\mathcal{A}}$ sont ceux qui contiennent au moins un état acceptant de $\mathcal{A}$.

3.2 Exercices

Exercice 3.1

Soit l'alphabet $\Sigma = \{a, b, c\}$. Soit L le langage sur Σ des mots qui commencent par a et se terminent par a.

1. Ecrire une expression régulière de L.
2. En déduire un automate fini $\mathcal{A}$ qui reconnaisse L. Cet automate devrait être ni déterministe ni complet.
3. Utiliser l'algorithme de "subset construction" pour trouver un automate fini déterministe complet équivalent à $\mathcal{A}$.
4. Soit l'automate fini $\mathcal{A}_0$ dont la table de transition est :

	a	b	c
- 1	2		
2	2,3	2	2
3 +			
- 4	5		
5 +			

Vérifier que $L = L(\mathcal{A}) = L(\mathcal{A}_0)$. Utiliser l'algorithme de "subset construction" pour trouver un automate fini déterministe complet équivalent à $\mathcal{A}_0$.

Exercice 3.2

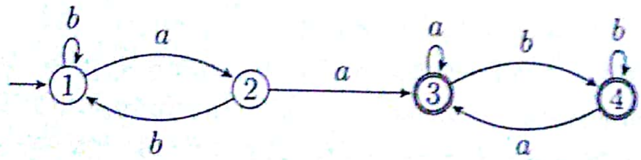
Soit l'alphabet $\Sigma = \{a, b\}$. Soit L le langage sur Σ des mots qui contiennent au moins une fois le facteur aa.

Partie 1 :

1. Ecrire une expression régulière de L.
2. En déduire un automate fini $\mathcal{A}$ qui reconnaisse L. Cet automate devrait être ni déterministe ni complet.
3. Utiliser l'algorithme de "subset construction" pour trouver un automate fini déterministe complet équivalent à $\mathcal{A}$.

Partie 2 :

Construire grâce à l'algorithme de Nérode l'automate fini déterministe complet minimal équivalent à l'AFdc ci-dessous.



Exercice 3.3

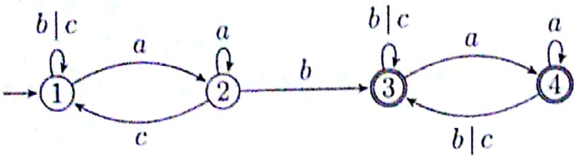
Soit l'alphabet $\Sigma = \{a, b, c\}$. Soit L le langage des mots sur Σ qui contiennent au moins une fois le facteur ab.

Partie 1 :

1. Ecrire une expression régulière de L.
2. En déduire un automate fini $\mathcal{A}$ qui reconnaisse L. Cet automate devrait être ni déterministe ni complet.
3. Utiliser l'algorithme de "subset construction" pour trouver un automate fini déterministe complet équivalent à $\mathcal{A}$.

Partie 2 :

1. Construire grâce à l'algorithme de Nérode l'automate fini déterministe complet minimal $\mathcal{A}_0$ équivalent à $\mathcal{A}$ déterministe complet donné ci-dessous :



2. Déduire de $\mathcal{A}_0$ un automate fini déterministe complet qui reconnaisse le langage des mots sur Σ qui ne contiennent pas le facteur ab.
3. Utiliser le résultat des questions précédentes pour proposer une expression régulière du langage des mots sur Σ qui ne contiennent pas le facteur ab.

Exercice 3.4

Soit $\Sigma = \{a, b, c\}$. Construire un AFdc minimal reconnaissant $L = ab\Sigma^* \mid \Sigma^* ca\Sigma^*$.

Exercice 3.5

Partie 1 : Construire les automates finis déterministes complets reconnaissants les langages de l'exercice 2.6.

Partie 2 : Rendre minimaux les automates finis déterministes complets de la partie 1.

Exercice 3.6

Soit l'alphabet $\Sigma = \{a, b\}$. Soit l'automate $\mathcal{A} = (\Sigma, E, D, A, \Theta)$ où $E = \{1, 2\}$, $D = \{1\}$, $A = \{2\}$ et Θ est donné par sa table de transition :

	Θ	a	b
-	1	1,2	1
	2 +	2	2

1. Cet automate est-il déterministe ? complet ?
2. Construire l'automate fini déterministe complet équivalent. Vérifier qu'il est minimal.
3. En déduire une expression régulière du langage L reconnu par l'automate $\mathcal{A}$ ainsi qu'une expression régulière du langage complémentaire de L dans Σ^*.
4. Attention : si un automate fini $\mathcal{A}$ n'est pas déterministe, on ne peut pas en déduire simplement un automate fini reconnaissant le langage complémentaire de $L(\mathcal{A})$. Le vérifier sur cet exemple.

Exercice 3.7

Soit l'alphabet $\Sigma = \{a, b\}$. Soit l'automate fini $\mathcal{A} = (\Sigma, E, D, A, \Theta)$ où $E = \{1, 2, 3, 4\}$, $D = \{1\}$, $A = \{4\}$ et Θ est donné par sa table de transition :

Θ		a	b
- 1		2	3
2		4	3
3		2	4
4	+	4	4

On cherche à montrer que le langage $L(\mathcal{A})$ reconnu par cet automate est le langage L des mots sur Σ qui contiennent au moins une fois le facteur aa ou le facteur bb.

Partie 1 :

1. Dessiner l'automate fini $\mathcal{A}$.
2. Écrire une expression régulière du langage L.
3. En déduire du résultat de la question 2 un automate fini $\mathcal{A}_0$ reconnaissant L.
4. Construire un automate fini déterministe complet équivalent à $\mathcal{A}_0$.

Partie 2 : On admet que l'AFdc $\mathcal{A}_0$ donné par la table de transition suivante reconnait L :

		a	b
- 1		2	3
2		4	3
3		2	5
4	+	4	5
5	+	4	5

1. Donner l'automate fini déterministe complet minimal équivalent à $\mathcal{A}_0$.
2. Pourquoi peut-on conclure que $L(\mathcal{A}) = L$?
3. Écrire une expression régulière du langage des mots sur Σ qui ne contiennent ni le facteur aa ni le facteur bb.

Exercice 3.7
Soit l'alphabet $\Sigma = \{a, b\}$. Soit l'automate fini $\mathcal{A} = (\Sigma, E, D, A, \Theta)$ où $E = \{1, 2, 3, 4\}$, $D = \{1\}$, $A = \{4\}$ et Θ est donné par sa table de transition :

Θ	a	b
- 1	2	3
2	4	3
3	2	4
4 +	4	4

On cherche à montrer que le langage $L(\mathcal{A})$ reconnu par cet automate est le langage L des mots sur Σ qui contiennent au moins une fois le facteur aa ou le facteur bb.

Partie 1 :

1. Dessiner l'automate fini $\mathcal{A}$.
2. Ecrire une expression régulière du langage L.
3. En déduire du résultat de la question 2 un automate fini $\mathcal{A}_0$ reconnaissant L.
4. Construire un automate fini déterministe complet équivalent à $\mathcal{A}_0$.

Partie 2 : On admet que l'AFdc $\mathcal{A}_0$ donné par la table de transition suivante reconnait L :

	a	b
- 1	2	3
2	4	3
3	2	5
4 +	4	5
5 +	4	5

1. Donner l'automate fini déterministe complet minimal équivalent à $\mathcal{A}_0$.
2. Pourquoi peut-on conclure que $L(\mathcal{A}) = L$?
3. Ecrire une expression régulière du langage des mots sur Σ qui ne contiennent ni le facteur aa ni le facteur bb.

Exercice 3.8

Soit l'alphabet $\Sigma = \{a, b\}$. Soit l'automate fini $\mathcal{A} = (\Sigma, E, D, A, \Theta)$ où $E = \{1, 2, 3, 4\}$, $D = \{1\}$,

$A = \{1\}$ et Θ est donné par sa table de transition :

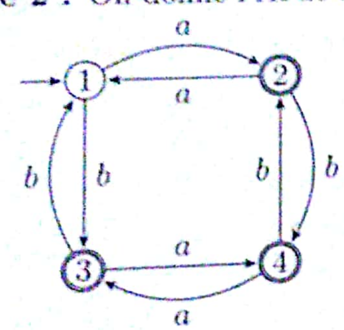
Θ	a	b
- 1 +	4	2
2	3	1
3	2	4
4	1	3

On cherche à montrer que le langage $L(\mathcal{A})$ reconnu par cet automate est le langage L des mots sur Σ qui contiennent un nombre pair de a et un nombre pair de b.

Partie 1 :

1. Dessiner l'automate fini $\mathcal{A}$.
2. Définir le langage $\bar{L}$ complémentaire de L dans Σ^* et en donner une expression régulière.
3. En déduire un automate fini reconnaissant $\bar{L}$, puis un automate fini déterministe complet équivalent.

Partie 2 : On donne l'AFdc suivant reconnaissant $\bar{L}$:



1. Déterminer l'automate fini déterministe complet minimal équivalent à l'AF ci-dessus.
2. Donner un automate fini déterministe complet reconnaissant L.
3. Montrer que $L(\mathcal{A}) = L$.

Exercice 3.9

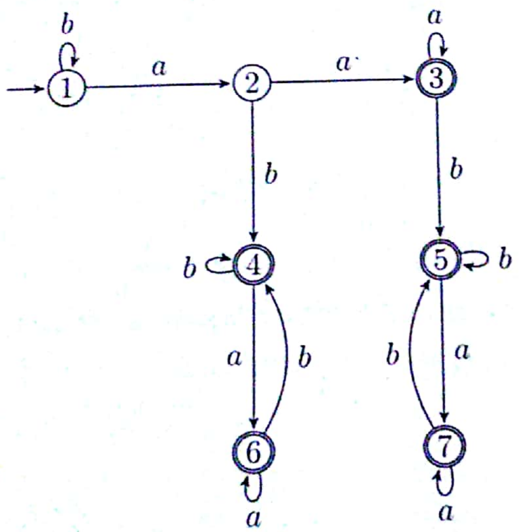
Soit l'alphabet $\Sigma = \{a\ , b\}$. On définit l'AF $\mathcal{A}$ par sa table de transition :

	Θ		a	b
-	1		1,2	1
	2			3
	3	+	3	3
-	4		1,5	4
	5		6	
	6	+	6	6

Partie 1 :

1. Dessiner l'AF $\mathcal{A}$ et en déduire une expression régulière du langage L reconnu par $\mathcal{A}$. Comment peut-on caractériser les mots de ce langage ?
2. L'AF $\mathcal{A}$ est-il déterministe ? complet ? Justifier les réponses.
3. En utilisant l'algorithme de "subset construction", construire un AFdc équivalent à $\mathcal{A}$. *Indication : cet AFdc possède 7 états.*

Partie 2 : On donne l'AFdc suivant qui reconnait L :



1. En utilisant l'algorithme de Nérode, trouver l'AFdc minimal reconnaissant L. *(Indication : cet AFdc possède 3 états.)*
2. Utiliser l'AFdc trouvé à la question 1 pour proposer une expression régulière pour le langage L (différente de l'expression régulière trouvée en partie 1).
3. Soit L' le langage des mots sur Σ qui ne contiennent ni aa ni ab. Que peut-on dire des langages L et L' ?
4. Déduire des questions précédentes un AFdc $\mathcal{A}'$ reconnaissant L' et une expression régulière du langage L'.

Exercice 3.10

1. Dans cette question, on travaille avec l'alphabet $\Sigma = \{a, b\}$.
   - (a) Soit L le langage des mots sur Σ qui se terminent par aa. Ecrire une expression régulière de L.
   - (b) En déduire un AF $\mathcal{A}$ reconnaissant $L' = L \cup \{a\}$ (cet AF aura 2 états de départ).
   - (c) Construire, en utilisant les algorithmes vus en cours, un AFdc équivalent à $\mathcal{A}$ (cet AFdc possède 5 états), puis l'AFdc minimal équivalent à $\mathcal{A}$ (cet AFdc minimal possède 3 états).
2. Dans cette question, on travaille avec l'alphabet $\Sigma' = \{0\ , 1\}$.

   Rappel : tout entier x strictement positif se décompose de manière unique comme somme de puissances de 2, c'est-à-dire selon la forme :

   $$x = x_k 2^k + x_{k-1} 2^{k-1} + \cdots + x_2 2^2 + x_1 2^1 + x_0 2^0, \text{ où } \forall i,\ 0 \leq i < k,\ x_i \in \{0, 1\} \text{ et } x_k = 1.$$

   L'écriture de cet entier x en binaire est donc le mot $m(x)$ sur Σ' : $m(x) = x_k x_{k-1} \cdots x_2 x_1 x_0$. Ainsi $233 = 1 \times 2^7 + 1 \times 2^6 + 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$, et l'écriture en binaire de 233 est donc $m(233) = 11101001$.
   - (a) Ecrire en binaire 18, 24, 15, 36, 0. Parmi ces entiers, y a-t-il des multiples de 4 ? Lesquels ?
   - (b) Comment caractériser les entiers divisibles par 4 à partir de leur écriture en binaire ?
   - (c) Donner une expression régulière du langage sur Σ' constitué des entiers positifs écrits en binaire divisibles par 4. Ce langage sera noté $L(4)$.
   - (d) Proposer un AFdc reconnaissant ce langage $L(4)$. *Indication : utiliser le résultat de la question 1(c).*

Exercice 3.11

Soit l'alphabet $\Sigma = \{a, b\}$. Montrer que le langage $L = \{a^k b^k / k \in \mathbb{N}\}$ sur Σ n'est pas un langage régulier.

Indication : faire un raisonnement par l'absurde. Supposer qu'il existe un automate fini reconnaissant L. Montrer que cet automate fini reconnaît aussi des mots de la forme $a^k b^n$ où n et k sont des entiers différents.

Exercice 3.12

Soit $\mathcal{A}$ l'automate fini $\mathcal{A} = (\Sigma, E, D, A, \Theta)$ avec des ε-transitions tel que $L(\mathcal{A}) = L = b^*a^*b$, dont la table de transition est donnée par :

	ε	a	b
- 1	2		1
2		2	3
3 +			

Construire un AFdc $\breve{\mathcal{A}}$ sans ε-transitions équivalent à $\mathcal{A}$.

Exercice 3.13

Soit $\mathcal{A}$ l'automate fini $\mathcal{A} = (\Sigma, E, D, A, \Theta)$ avec des ε-transitions tel que $L(\mathcal{A}) = L = a^*b^*|b^*a^*$,

dont la table de transition est donnée par :

		ε	a	b
-	1	2,4		
	2 +	3	2	
	3 +			3
	4 +	5		4
	5 +		5	

1. Dessiner l'automate $\mathcal{A}$.
2. Construire un AFdc $\breve{\mathcal{A}}$ sans ε-transitions équivalent à $\mathcal{A}$. Dessiner l'automate obtenu.
3. Déterminer une expression régulière du langage $\bar{L}$.
4. $\breve{\mathcal{A}}$ est-il minimal ?

Exercice 3.14

Soit l'alphabet $\Sigma = \{a, b\}$. Soit $\mathcal{A}$ l'automate fini dont la table de transition est donnée ci-dessous :

		ε	a	b
-	1	2	1	
	2	3		2
	3 +		3	

1. Trouver $\mathcal{A}'$, l'automate fini déterministe complet minimal équivalent à $\mathcal{A}$. *(On devrait trouver que $\mathcal{A}'$ possède 4 états.)*
2. Ecrire une expression régulière du langage $L = L(\mathcal{A}) = L(\mathcal{A}')$.

Exercice 3.15

Soit l'alphabet $\Sigma = \{a, b\}$.

1. Soit les langages sur Σ

$$L_1 = b^*ab^*, \qquad L_2 = a^*, \qquad L_3 = L_1 \cdot L_2, \qquad L_4 = bab^*a, \qquad L_5 = L_3 \cup L_4.$$

   (a) Dessiner le graphe d'un automate fini reconnaissant L_1, puis le graphe d'un automate fini reconnaissant L_2, puis le graphe d'un automate fini reconnaissant L_4.

   (b) En utilisant des ε-transitions, dessiner le graphe d'un automate fini reconnaissant L_3, puis le graphe d'un automate fini reconnaissant L_5.

2. Soit $\mathcal{A}$ l'automate fini dont la table de transition est donnée ci-dessous :

	ε	a	b
- 0	1,4		
1		2	1
2	3		2
3 +		3	
4			5
5		6	
6		7	6
7 +			

   (a) Expliquer pourquoi cet automate $\mathcal{A}$ n'est ni déterministe ni complet.

   (b) En utilisant l'algorithme vu en cours, construire un automate fini déterministe complet sans ε-transitions $\mathcal{A}_1$ équivalent à $\mathcal{A}$, c'est-à-dire tel que $L(\mathcal{A}_1) = L(\mathcal{A})$. *(On détaillera avec soin les différentes étapes de l'algorithme. On devrait trouver que $\mathcal{A}_1$ possède 8 états.)*

   (c) En utilisant l'algorithme de Nérode, construire l'automate fini déterministe complet (AFdc) minimal $\mathcal{A}_0$ équivalent à $\mathcal{A}_1$ et à $\mathcal{A}$, c'est-à-dire tel que $L(\mathcal{A}_0) = L(\mathcal{A}_1) = L(\mathcal{A})$, ou encore tel que les automates $\mathcal{A}_0$, $\mathcal{A}_1$ et $\mathcal{A}$ reconnaissent le même langage. *(On détaillera avec soin les différentes étapes de l'algorithme. On devrait trouver que $\mathcal{A}_0$ possède 4 états.)*

3. (a) Dessiner le graphe de $\mathcal{A}_0$ et en déduire une expression régulière L_0 du langage $L(\mathcal{A}_0)$ reconnu par $\mathcal{A}_0$.

   (b) Dessiner le graphe de $\mathcal{A}$ et en déduire une expression régulière L du langage $L(\mathcal{A})$ reconnu par $\mathcal{A}$.

   (c) Par construction (cf. 2(c)), $\mathcal{A}_0$ et $\mathcal{A}$ sont équivalents et donc $L_0 = L(\mathcal{A}_0) = L(\mathcal{A}) = L$. Expliquer pourquoi les expressions régulières trouvées en 3(a) et 3(b) sont bien égales. Quel lien y a-t-il avec la question 1 ?

Exercice 3.16

Soit l'alphabet $\Sigma = \{a, b\}$. L désigne le langage sur Σ : $L = a^*b^*aa^*|ab^*a$.

1. Trouver un automate fini avec des ε-transitions reconnaissant le langage L.
2. En utilisant les algorithmes vus en cours, trouver l'automate fini déterministe complet minimal reconnaissant le langage L *(Indication : cet AFdc minimal a 5 états)*.
3. Vérifier ainsi que $L = a^*b^*aa^*$.

4 Grammaires algébriques

4.1 Rappels de cours

✓ **Définition**

Une grammaire algébrique est un quadruplet (Σ, NT, S, P) où :

- Σ est un alphabet dont les caractères sont appelées symboles terminaux
- NT est un alphabet disjoint de Σ dont les lettres sont appelées symboles non terminaux ou variables
- S est un élément de NT appelé symbole initial ou symbole de départ ou axiome
- P est un ensemble fini de règles de réécriture ou règles de production : $X \to \delta$, avec $X \in NT$, $\delta \in (\Sigma \cup NT)^*$ et $(X, \delta) \in P$.

✓ **Définition (Dérivation dans une grammaire algébrique)**

Soient une grammaire algébrique $G = (\Sigma, NT, S, P)$, α et β deux mots de $(\Sigma \cup NT)^*$.

- On dit que β dérive de α en une étape s'il existe des mots α', α'' et δ de $(\Sigma \cup NT)^*$ et un symbole non terminal X tels que :

$$\alpha = \alpha' X \alpha'', \quad \beta = \alpha' \delta \alpha'', \text{ et } (X, \delta) \in P.$$

  On écrit : $\alpha \to \beta$.

- On peut définir par induction la dérivation en n étapes, $n \in \mathbb{N}$ ($n = 0$ correspond au cas où $\alpha = \beta$) : β dérive de α en $n+1$ étapes s'il existe w tel que w dérive de α en n étapes et β dérive de w en une étape :

$$\text{si } \alpha \xrightarrow{n} w \text{ et } w \to \beta, \text{ alors } \alpha \xrightarrow{n+1} \beta.$$

- On dit que β dérive de α s'il existe $n \in \mathbb{N}$ tel que β dérive de α en n étapes, ce qui se note $\alpha \xrightarrow{*} \beta$. La longueur de la dérivation $\alpha \xrightarrow{*} \beta$ est égale au nombre d'étapes.

✓ **Définition**

Le langage engendré par la grammaire algébrique G est l'ensemble des mots de Σ^* qui dérivent de l'axiome S (ces mots ne contiennent aucun symbole non terminal) :

$$L(G) = \{\alpha / \alpha \in \Sigma^* \text{ et } \alpha \text{ dérive de } S\} = \{\alpha / \alpha \in \Sigma^* \text{ et } S \xrightarrow{*} \alpha\}.$$

✓ **Définition**

Deux grammaires G et G' sont équivalentes si et seulement si elles engendrent le même langage : $L(G) = L(G')$.

✓ **Définition (Grammaire linéaire)**

Une grammaire algébrique $G = (\Sigma, NT, S, P)$ est dite linéaire si toutes les règles de production sont du type

$$A \to \alpha \text{ où } A \in NT, \alpha \in (\Sigma \cup NT)^* \text{ et } \alpha \text{ contient au plus un symbole non terminal.}$$

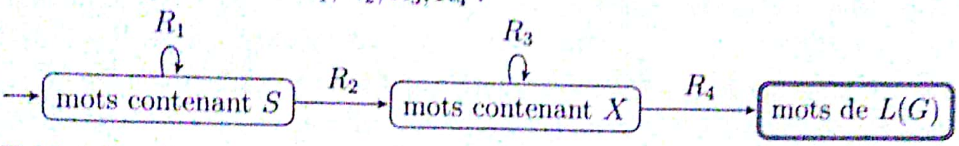
Proposition

Toute grammaire linéaire peut être représentée par une chaîne de production, c'est-à-dire par un graphe orienté valué du type de ceux associés aux automates.

* Les états du graphe sont les ensembles "mots contenant X" pour chaque variable X ainsi que "mots de $L(G)$".
* Ce dernier état est un état acceptant.
* L'état de départ est "mots contenant le symbole initial".
* Chaque règle R_i du type $\{X \to \text{mot contenant } Y\}$ correspond à un arc depuis l'état "mots contenant X" vers l'état "mots contenant Y", étiqueté "R_i".

Exemple

La grammaire $G = (\Sigma = \{a, b\}, NT = \{S, X\}, S, \{S \to aSa, S \to X, X \to bXb, X \to \varepsilon\})$ est linéaire. On peut la représenter par la chaîne de production suivante dans laquelle on a numéroté les règles de production R_1, R_2, R_3, R_4 :



Définition (Classification simplifiée de Chomsky)

Soit une grammaire $G = (\Sigma, NT, S, P)$.

* Les grammaires de type 0 sont les grammaires des langages naturels.
* La grammaire G est de type 1 ou contextuelle ssi toutes ses règles de production sont du type

$$m \to \alpha \text{ où } m \in (\Sigma \cup NT)^* \setminus \Sigma^* \text{ et } \alpha \in (\Sigma \cup NT)^*.$$

* La grammaire G est de type 2 ou non contextuelle ou algébrique ssi toutes les règles de production sont du type :

$$A \to \alpha \text{ où } A \in NT \text{ et } \alpha \in (\Sigma \cup NT)^*.$$

  Le membre de gauche est réduit à un unique symbole non terminal.
* Les grammaires linéaires constituent une classe intermédiaire entre le type 2 et le type 3.
* La grammaire G est de type 3 ou régulière ssi G est algébrique et linéaire à droite [a] : toutes les règles de production sont
  * soit du type $A \to \alpha B$,
  * soit du type $A \to b$,

  avec $A, B \in NT$, $\alpha \in \Sigma^* \setminus \{\varepsilon\}$ et $b \in \Sigma^*$. Le membre de droite comporte au plus un symbole non terminal qui doit alors être à la fin du membre de droite. [b]

---

a. ou linéaire à gauche (mais pas les deux simultanément) : dans ce cas, les règles sont du type : $A \to B\alpha$ et $A \to b$.

b. ou dans le cas d'une grammaire linéaire à gauche, le membre de droite comporte au plus un symbole non terminal qui doit alors être au début du membre de droite.

Définition

Un langage est de type i s'il est possible de trouver une grammaire de type i qui l'engendre $(1 \leq i \leq 3)$.

Remarque

Tout langage de type i est aussi de type j avec $j \leq i$.

Théorème

Il y a identité entre

* les langages réguliers, c'est-à-dire définis par une expression régulière
* les langages reconnaissables, c'est-à-dire reconnus par au moins un automate fini
* les langages de type 3, c'est-à-dire produits par au moins une grammaire de type 3.

Remarque

Une grammaire régulière engendre un langage régulier. Mais il se peut qu'une grammaire algébrique non régulière engendre un langage régulier. Dans ce cas, on pourra définir une grammaire régulière engendrant le même langage.

Théorème d'Arden

Un langage L satisfaisant $L = xL \mid y$, avec x et y deux expressions régulières quelconques telles que $\varepsilon \notin L(x)$, est donné par l'expression régulière $L = x^* y$.

Remarque

Si $\varepsilon \in L(x)$, alors $L = x^* y$ n'est pas le seul ensemble solution. En fait, $L = \Sigma^*$.

Corollaire

Pour $X \in NT$, et $e, f \in (NT \cup \Sigma)^*$, avec $\varepsilon \notin e$, si X vérifie la règle de production $X \to eX \mid f$, alors toutes les dérivations possibles avec cette règle de production donnent $X \xrightarrow{*} e^* f$.

4.2 Exercices

Exercice 4.1

1. Soit la grammaire $G_1 = (\Sigma = \{a, b, c\}, NT = \{S\}, S, \{S \to aSb \mid c\})$.
   (a) Ecrire les dérivations de longueur inférieure ou égale à 4 selon la grammaire G_1.
   (b) Ecrire les mots de longueur inférieure ou égale à 5 dérivés par la grammaire G_1.
   (c) Démontrer que $L(G_1) = \{a^n cb^n / n \in \mathbb{N}\}$.
2. Soit la grammaire $G_2 = (\Sigma = \{a, b, c, d, e\}, NT = \{S, X\}, S, \{S \to aSb \mid X \mid c,\ X \to eXd \mid \varepsilon\})$.
   (a) Quel est le langage engendré par G_2 ?
   (b) Montrer que $L(G_1) \subset L(G_2)$. Pouvait-on s'y attendre ?

Exercice 4.2

1. Soit la grammaire $G_1 = (\Sigma = \{a, b, c\}, NT = \{S, X\}, S, \{S \to XabX,\ X \to aX \mid bX \mid cX \mid \varepsilon\})$.
   (a) Construire un arbre de dérivation du mot $(ab)^2(cb)^2$; du mot $c^2a^2b^2c^2$.
   (b) Le mot a^4cb est-il dérivable dans G_1 ? Pourquoi ?
   (c) De quel type est la grammaire G_1 ?
   (d) Déterminer le langage $L(G_1)$. Ce langage est-il reconnaissable par un automate fini ?
   (e) Déterminer une grammaire régulière G_1' telle que $L(G_1') = L(G_1)$.
2. Soit la grammaire $G_2 = (\Sigma = \{a, b, c\}, NT = \{S, X, Y\}, S, \{S \to XabY,\ X \to bX \mid cX \mid \varepsilon,\ Y \to Yc \mid Ya \mid \varepsilon\})$.
   (a) Construire un arbre de dérivation du mot $cbabac$; du mot $bccabcca$.
   (b) Le mot $c(ab)^2c$ est-il dérivable dans G_2 ? Même question pour le mot ca^2b^2c.
   (c) De quel type est la grammaire G_2 ?
   (d) Déterminer le langage $L(G_2)$. Ce langage est-il reconnaissable par un automate fini ?
   (e) Déterminer une grammaire régulière G_2' telle que $L(G_2') = L(G_2)$.
3. Soit la grammaire $G_3 = (\Sigma = \{a, b\}, NT = \{S\}, S, \{S \to SS \mid aSa \mid bSb \mid \varepsilon\})$.
   (a) Construire un arbre de dérivation du mot $aabaab$. Cet arbre de dérivation est-il unique ? Mêmes questions pour le mot $bbaabb$.
   (b) Le mot $abab$ est-il dérivable dans G_3 ? Même question pour le mot bab.
   (c) De quel type est la grammaire G_3 ?

Exercice 4.3

1. Soit la grammaire $G_1 = (\Sigma = \{a,b,c\}, NT = \{S,X\}, S, \{S \to bSc \mid X,\ X \to Xa \mid a\})$.
   (a) Cette grammaire est-elle régulière ? linéaire ? algébrique ? Justifiez vos réponses.
   (b) Proposez 3 mots produits par la grammaire G_1 en donnant pour chacun d'eux une dérivation qui le produit.
   (c) Dessiner une chaine de production de la grammaire G_1 ;
   (d) Déduire de la question précédente le langage $L(G_1)$.
   (e) Est-il possible de proposer une expression régulière pour décrire $L(G_1)$? Si oui, donner une telle expression et donner une grammaire régulière qui produit ce langage ; sinon justifier votre réponse.
2. Mêmes questions pour la grammaire

$$G_2 = (\Sigma = \{a,b,c\}, NT = \{S,X,Y\}, S, \{S \to aXb,\ X \to aX \mid Yc,\ Y \to \varepsilon\}).$$

Exercice 4.4

Soit la grammaire $G = (\Sigma = \{a,b,c\}, NT = \{S,X,Y\}, S, \{S \to bS \mid XaY,\ X \to bX \mid \varepsilon,\ Y \to Yc \mid \varepsilon\})$.

1. Cette grammaire est-elle régulière ? linéaire ? algébrique ? Justifiez vos réponses.
2. Proposez 3 mots produits par la grammaire G en donnant pour chacun d'eux une dérivation qui le produit.
3. Définir le langage $L(G)$.
4. Est-il possible de proposer une expression régulière pour décrire $L(G)$? Si oui, donner une telle expression ; sinon justifier votre réponse.

Exercice 4.5

Soit le langage L des mots sur $\Sigma = \{a,b,c\}$ qui contiennent un nombre pair de a.

1. Proposer un automate fini qui reconnait L.
2. En déduire une grammaire qui engendre L.
3. Même question avec le langage L' des mots sur $\Sigma = \{a,b,c\}$ qui contiennent un nombre impair de a.

Exercice 4.6

Soit la grammaire $G = (\Sigma = \{a, b, c\}, NT = \{S, X, Y, Z\}, S, \Pi)$ où Π est l'ensemble des règles de production :

$$S \to Xc \,|\, ababY, \quad X \to Sba \,|\, cZ, \quad Y \to abcYabcbababa, \quad Z \to bZbb \,|\, a.$$

1. Cette grammaire est-elle régulière ? linéaire ? algébrique ? Justifiez vos réponses.
2. Proposez 3 mots produits par la grammaire G en donnant pour chacun d'eux une dérivation qui le produit.
3. Dessiner une chaîne de production de la grammaire G. Justifier que certaines règles et certaines variables sont inutiles.
4. Déduire de la question précédente le langage $L(G)$.
5. Est-il possible de proposer une expression régulière pour décrire $L(G)$?
6. On modifie la grammaire précédente en remplaçant la règle $Z \to a$ par la règle $Z \to b$.
   (a) Définir le langage $L(G')$.
   (b) Est-il possible de décrire $L(G')$ par une expression régulière ? Si oui, donner une telle expression ainsi qu'un automate fini qui reconnaît $L(G')$ et une grammaire régulière qui engendre $L(G')$. Sinon, justifier votre réponse.