

Carnet de Travaux Libres

Théorie des Langages

Licence 3 Informatique

Julien BERNARD

Table des matières

1	Alphabets, mots, langages	3
	Exercice 1 : Propriétés sur les langages	3
	Exercice 2 : Langages sur $\{a, b\}$	4
2	Grammaires	5
	Exercice 3 : Grammaire des expressions arithmétiques	5
	Exercice 4 : Grammaire pour téléphone	6
	Exercice 5 : Grammaire ambiguë	7
	Exercice 6 : Grammaire équivalente	8
3	Automates finis	9
	Exercice 7 : Automates de langages simples	9
	Exercice 8 : Automates sur l'alphabet $\Sigma = \{a, b\}$	10
	Exercice 9 : Détermination d'un automate	11
	Exercice 10 : Détermination d'un automate	12
	Exercice 11 : Détermination d'un automate	13
	Exercice 12 : Opérations ensemblistes sur les automates	14
	Exercice 13 : Digicode	15
4	Expressions régulières	16
	Exercice 14 : Expressions régulières sur l'alphabet $\Sigma = \{0, 1\}$	16
	Exercice 15 : Automate standard	17
	Exercice 16 : Intersection de langages	18
	Exercice 17 : Expression régulière à partir d'un automate	19
	Exercice 18 : Expression régulière à partir d'un automate	20
	Exercice 19 : Expression régulière à partir d'un automate	21
5	Langages réguliers et non-réguliers	22
	Exercice 20 : Langage miroir	22
	Exercice 21 : Langage des mots contenant autant de a que de b	23
	Exercice 22 : Langages réguliers ou non-réguliers	24

6	Minimisation d'automates	25
	Exercice 23 : Minimisation d'un automate	25
	Exercice 24 : Minimisation d'automate	26
	Exercice 25 : Automates, détermination, minimisation	27
	Exercice 26 : Expression régulière, détermination et minimisation . .	28
7	Langages algébriques	29
	Exercice 27 : Nettoyage de grammaire	29
	Exercice 28 : Grammaire réduite	30
	Exercice 29 : Grammaire propre	31
	Exercice 30 : Grammaire propre	32
	Exercice 31 : Élimination des récursivités à gauche	33
	Exercice 32 : Élimination des récursivités à gauche	34
8	Automates à piles	35
	Exercice 33 : Automate à pile	35
	Exercice 34 : Automate à pile et expressions bien parenthésées	36
	Exercice 35 : Automate à pile	37

1 Alphabets, mots, langages

Exercice 1 : Propriétés sur les langages

Soit L, L_1, L_2, L_3 des langages sur un alphabet A .

Question 1.1 Montrer que $(L_1.L_2).L_3 = L_1.(L_2.L_3)$

Question 1.2 Montrer que $L_1.(L_2 \cup L_3) = (L_1 \cup L_2).(L_1 \cup L_3)$

Question 1.3 Montrer que $L_1.(L_2 \cap L_3) \subseteq (L_1.L_2) \cap (L_1.L_3)$. A-t-on toujours l'égalité ?

Question 1.4 Montrer que $\varepsilon \in L^*$, $L \subseteq L^*$, $(L^*)^* = L^*$

Question 1.5 Montrer que $L_1 \subseteq L_2 \implies L_1^* \subseteq L_2^*$

Question 1.6 Comparer $(L_1 \cap L_2)^*$ et $L_1^* \cap L_2^*$

Exercice 2 : Langages sur $\{a, b\}$

Soient L_1, L_2, L_3 les langages sur l'alphabet $\Sigma = \{a, b\}$:

$$L_1 = \{ a^n b (a + b)^n, n \in \mathbb{N} \}$$

$$L_2 = \{ (a + b)^n b a^n, n \in \mathbb{N} \}$$

$$L_3 = \{ (a + b)^n b (a + b)^n, n \in \mathbb{N} \}$$

Question 2.1 Montrer que L_1, L_2 et L_3 ne sont pas égaux.

Question 2.2 Soit $L_4 = \{ (a + b)^m b a^n, m \in \mathbb{N}, n \in \mathbb{N} \}$. Montrer que $L_2 \neq L_4$.

Question 2.3 Donner les grammaires qui engendrent L_2 et L_4 .

2 Grammaires

Exercice 3 : Grammaire des expressions arithmétiques

On donne la grammaire $G = (\{E, T, F\}, \{+, *, (,), a, b\}, E, R)$ avec R :

- $E \rightarrow E + T \mid T$
- $T \rightarrow T * F \mid F$
- $F \rightarrow (E) \mid a \mid b$

Question 3.1 Pour chacun des mots suivants dire si ils sont dans $\mathcal{L}(G)$ ou non :

1. $a * (b + a)$
2. $a + (b * c)$
3. $(b * (a * a))$
4. $b * a + *b$
5. $a * (b * (b * a + b))$

Exercice 4 : Grammaire pour téléphone

Considérons un téléphone portable dont le clavier est verrouillé. Pour débloquer le clavier, il faut taper une suite qui se termine par 159.

Question 4.1 Proposer une grammaire G (régulière ou algébrique) pour engendrer le langage L dont tous les mots permettent de déverrouiller le clavier.

Exercice 5 : Grammaire ambiguë

Soit la grammaire $G = (\{S\}, \{\text{if, then, else, } a, b\}, S, R)$ avec R :

• $S \rightarrow \text{if } b \text{ then } S \text{ else } S \mid \text{if } b \text{ then } S \mid a$

Question 5.1 Que peut-on dire à propos de la grammaire G ?

Question 5.2 Montrer que G est ambiguë.

Question 5.3 Proposer des solutions qui contournent/repoussent/résolvent ce problème d'ambiguïté. Désigner celles parmi ces solutions qui vous semblent la meilleure et la pire.

Exercice 6 : Grammaire équivalente

Question 6.1 Calculer une grammaire sans production vide (sauf éventuellement $X' \rightarrow \varepsilon$) équivalente à la grammaire $G = (\{X, A, B, C, D, E\}, \{a, b\}, X, R)$ avec R :

- $X \rightarrow aX \mid XX \mid AB \mid aA$
- $A \rightarrow \varepsilon \mid BB \mid CDE$
- $B \rightarrow DE \mid AC \mid bA$
- $C \rightarrow CaC \mid \varepsilon$
- $D \rightarrow aDb \mid BE$
- $E \rightarrow XaX$

3 Automates finis

Exercice 7 : Automates de langages simples

Question 7.1 Donner un automate qui reconnaît le langage vide $L = \emptyset$.

Question 7.2 Donner un automate qui reconnaît le langage unité $L = \{\varepsilon\}$.

Question 7.3 Donner un automate qui reconnaît le langage composé d'un seul mot d'une seule lettre $L = \{a\}$.

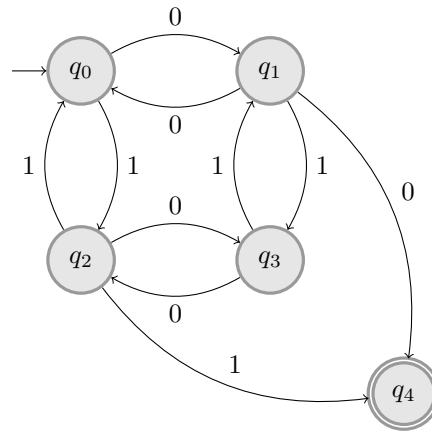
Exercice 8 : Automates sur l'alphabet $\Sigma = \{a, b\}$

Question 8.1 Proposer un automate \mathcal{A} d'états finis sur l'alphabet $\Sigma = \{a, b\}$ pour reconnaître l'ensemble de tous les mots qui contiennent :

1. au moins une lettre a ;
2. deux lettres b consécutives ;
3. toujours un a immédiatement après un b ;
4. un a après un b .

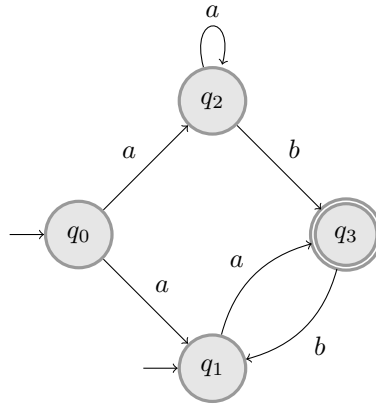
Exercice 9 : Déterminisation d'un automate

Question 9.1 Déterminiser l'automate suivant :



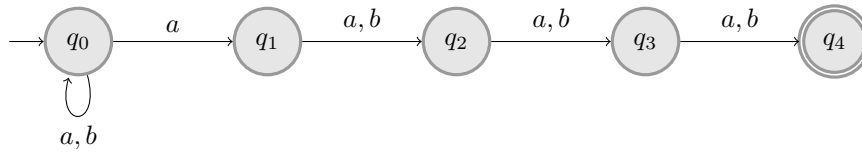
Exercice 10 : Détermination d'un automate

Question 10.1 Déterminiser l'automate suivant :



Exercice 11 : Détermination d'un automate

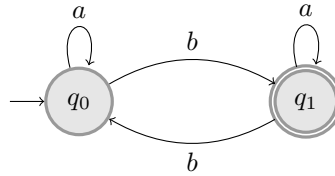
Question 11.1 Déterminer l'automate suivant :



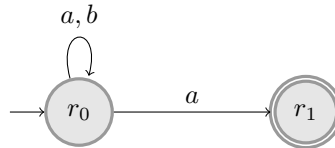
Question 11.2 Qu'observe-t-on ? Généraliser cette observation.

Exercice 12 : Opérations ensemblistes sur les automates

Soit \mathcal{A}_Q l'automate qui reconnaît le langage L_Q des mots qui ont un nombre impair de b :



Soit \mathcal{A}_R l'automate qui reconnaît le langage L_R des mots qui finissent par un a :



On veut construire des automates qui reconnaissent une combinaison ensembliste de L_Q et L_R . Pour cela, on s'autorise à utiliser des ε -transitions et on cherche une méthode générale qui puisse s'appliquer à n'importe quelle paire d'automates.

Question 12.1 Construire l'automate qui reconnaît le langage $L_Q \cup L_R$

Question 12.2 Construire l'automate qui reconnaît le langage $L_Q \cdot L_R$

Question 12.3 Construire l'automate qui reconnaît le langage $L_Q \cap L_R$

Exercice 13 : Digicode

Considérons un digicode d'entrée dans un bâtiment : trois touches a , b et c sont possibles ; la porte s'ouvre dès que l'on a tapé la suite aba . Une fois la porte fermée, il faut de nouveau taper le code correct pour rentrer.

Question 13.1 Proposer un automate d'états fini qui reconnaît les mots du langage L d'ouverture de la porte.

Question 13.2 Construire un automate déterministe équivalent à l'automate proposé précédemment.

4 Expressions régulières

Exercice 14 : Expressions régulières sur l'alphabet $\Sigma = \{0, 1\}$

Question 14.1 Proposer des expressions régulières pour des langages dont tous les mots sont des éléments de $\{0, 1\}^*$ satisfaisant les conditions suivantes :

1. ils terminent soit par 011, soit par 101 ou par 110 ;
2. ils ont chacune des chaînes 011 et 101 ;
3. $\{01^n, n \geq 2\}$;
4. $\{0^m 1^n, n \geq 2, m \geq 2\}$.

Exercice 15 : Automate standard

Soit $(a^*b^*)^* + aa + bb$ une expression régulière.

Question 15.1 Donner un automate standard \mathcal{A}_s reconnaissant le langage décrit par cette expression.

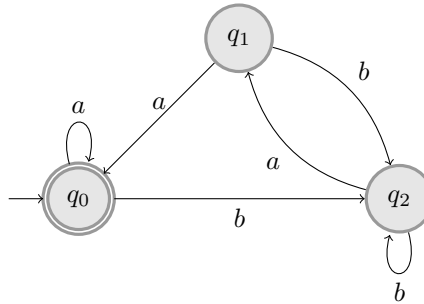
Exercice 16 : Intersection de langages

Soit $e_1 = a(a+b)^* + ab + ba$ et $e_2 = (a+b)^* + ca + b(c+a)$ deux expressions régulières.

Question 16.1 Donner un automate A_{12} reconnaissant l'intersection des langages décrits par e_1 et e_2 .

Exercice 17 : Expression régulière à partir d'un automate

On considère l'automate \mathcal{A} suivant :



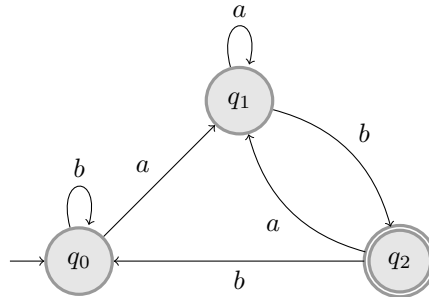
Question 17.1 Déterminer une expression régulière représentant le langage accepté par l'automate \mathcal{A} en utilisant un système d'équation.

Question 17.2 Déterminer une expression régulière représentant le langage accepté par l'automate \mathcal{A} en utilisant la méthode de Brzozowski et McCluskey.

Question 17.3 Montrer que les deux expressions régulières obtenues sont égales.

Exercice 18 : Expression régulière à partir d'un automate

On considère l'automate \mathcal{A} suivant :



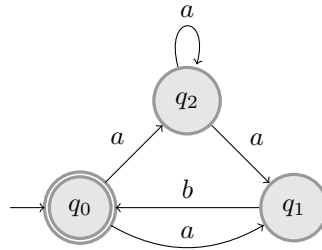
Question 18.1 Déterminer une expression régulière représentant le langage accepté par l'automate \mathcal{A} en utilisant un système d'équation.

Question 18.2 Déterminer une expression régulière représentant le langage accepté par l'automate \mathcal{A} en utilisant la méthode de Brzozowski et McCluskey.

Question 18.3 Montrer que les deux expressions régulières obtenues sont égales.

Exercice 19 : Expression régulière à partir d'un automate

Soit \mathcal{A} l'automate suivant :



Question 19.1 En utilisant le lemme d'Arden, donner une expression régulière qui représente le langage accepté par l'automate \mathcal{A} . Détailler les calculs.

Question 19.2 En utilisant la méthode de Brzozowski et McCluskey, donner une expression régulière qui représente le langage accepté par l'automate \mathcal{A} . Détailler les étapes de calcul.

Question 19.3 Montrer que ces expressions régulières sont égales à $(a^*ab)^*$.

Question 19.4 Déterminer l'automate \mathcal{A}

Question 19.5 Proposer une grammaire qui engendre le langage accepté par l'automate \mathcal{A}

5 Langages réguliers et non-réguliers

Exercice 20 : Langage miroir

Soit $w = w_1w_2 \dots w_{n-1}w_n$ un mot sur un alphabet Σ , on définit w^R le mot miroir de w par $w^R = w_nw_{n-1} \dots w_2w_1$. Soit L un langage, on définit L^R le langage miroir par $L^R = \{w^R, w \in L\}$.

Question 20.1 Soit $w = abbabaa$, donner w^R .

Question 20.2 Montrer que si L est régulier, alors L^R est également régulier.

Exercice 21 : Langage des mots contenant autant de a que de b

Soit $A = \{a, b\}$ un alphabet et $L = \{w \in A^*, |w|_a = |w|_b\}$.

Question 21.1 Montrer que L n'est pas régulier.

Exercice 22 : Langages réguliers ou non-réguliers

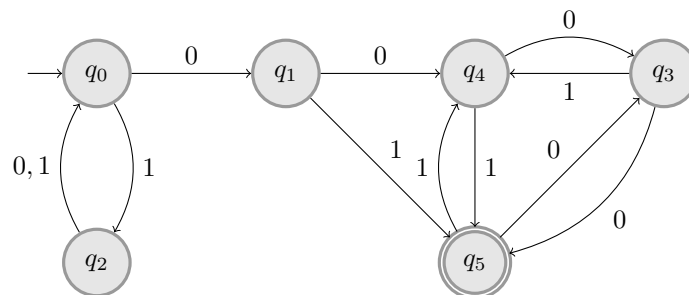
Question 22.1 Parmi les langages suivants, montrer lesquels sont réguliers et lesquels ne sont pas réguliers ?

1. $L_1 = \{a^{2^n}, n \geq 1\}$
2. $L_2 = \{a^{2^n}, n \geq 1\}$
3. $L_3 = \{ww, w \in \{a, b\}^*\}$
4. $L_4 = \{w \in \{a, b\}^*, w \text{ n'a pas trois } b \text{ consécutifs}\}$

6 Minimisation d'automates

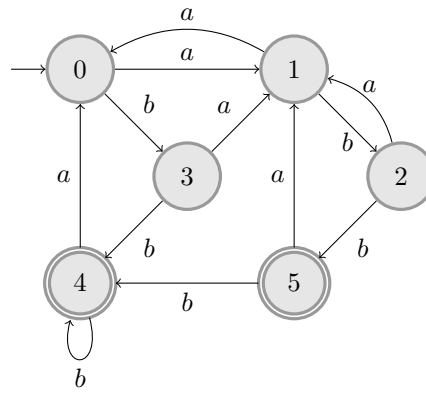
Exercice 23 : Minimisation d'un automate

Question 23.1 Minimiser l'automate suivant :



Exercice 24 : Minimisation d'automate

Question 24.1 Minimiser l'automate \mathcal{A} suivant :



Exercice 25 : Automates, déterminisation, minimisation

Question 25.1 Proposer un automate (déterministe ou non-déterministe) pour chacun des langages suivants :

1. le langage des représentations binaires des nombres pairs ;
2. $(a + b)^*ba^*$;
3. $(a + b + c)^*a(a + b + c)^*b + c(a + b)^*$;
4. $(a + b)^*a^*(bc)^* + (b(ca)^* + ac)^*b^*$;
5. le langage des représentations décimales des multiples de 3 ;
6. le langage d'écriture des nombres réels en décimal.

Question 25.2 Pour les automates non-déterministes, construire les automates déterministes équivalents.

Question 25.3 Minimiser les automates déterministes construits.

Exercice 26 : Expression régulière, déterminisation et minimisation

On considère le langage régulier L représenté par l'expression régulière :

$$(ab + bc)^*(bb + ca)^*$$

Question 26.1 Proposer un automate pour L

Question 26.2 Si l'automate n'est pas déterministe, construire un automate déterministe équivalent

Question 26.3 Quel est le nombre d'états de l'automate avant minimisation ? Minimiser l'automate déterministe construit. Quel est le nombre d'états de l'automate minimisé ?

7 Langages algébriques

Exercice 27 : Nettoyage de grammaire

On considère la grammaire $G = (\{S, A, B, C, D, E, F\}, \{a, b\}, S, R)$ avec R :

- $S \rightarrow A \mid B$
- $A \rightarrow aE \mid \varepsilon$
- $B \rightarrow aF$
- $D \rightarrow b$
- $E \rightarrow aEE \mid aFC$
- $F \rightarrow b \mid aEF \mid aFD$

Question 27.1 Réduire la grammaire

Question 27.2 Supprimer les règles unitaires

Question 27.3 Quel est le langage engendré par cette grammaire ?

Exercice 28 : Grammaire réduite

Question 28.1 Réduire la grammaire $G = (\{S, A, B, C, D, E\}, \{a, b\}, S, R)$ avec R :

- $S \rightarrow CD \mid AA$
- $A \rightarrow CC$
- $B \rightarrow bA \mid EE$
- $C \rightarrow a \mid AD$
- $D \rightarrow Ab \mid BE$
- $E \rightarrow aE$

Exercice 29 : Grammaire propre

On considère la grammaire $G = (\{S, A, B, C, D, E\}, \{a, b\}, S, R)$ avec R :

- $S \rightarrow aS \mid SS \mid AB \mid aA$
- $A \rightarrow \varepsilon \mid BB \mid CDE$
- $B \rightarrow DE \mid AC \mid bA$
- $C \rightarrow CaC \mid \varepsilon$
- $D \rightarrow aDb \mid BE$
- $E \rightarrow SaS$

Question 29.1 Calculer une grammaire propre équivalente à G au mot vide près.

Exercice 30 : Grammaire propre

Question 30.1 Rendre propre la grammaire $G_1 = (\{S, X, Y\}, \{a, b\}, S, R)$ avec R :

- $S \rightarrow aXbXa$
- $X \rightarrow aY \mid \varepsilon$
- $Y \rightarrow aa \mid b$

Question 30.2 Rendre propre la grammaire $G_2 = (\{S, X, Y\}, \{a, b, c\}, S, R)$ avec R :

- $S \rightarrow aSbX \mid X$
- $X \rightarrow XYc \mid Y$
- $Y \rightarrow ab \mid bc \mid ac$

Exercice 31 : Élimination des récursivités à gauche

On considère la grammaire $G = (\{E, T, F, Q\}, \{;, +, *, -, (,), a\}, E, R)$ avec R :

- $E \rightarrow E + T \mid T$
- $T \rightarrow T * F \mid F$
- $F \rightarrow Q \mid -Q$
- $Q \rightarrow (E) \mid a \mid E; E$

Question 31.1 Éliminer toutes les récursivités à gauche (immédiates et non-immédiates).

Exercice 32 : Élimination des récursivités à gauche

On considère la grammaire $G = (\{E, T, Q\}, \{;, +, *, -, (,), a\}, E, R)$ avec R :

- $E \rightarrow E + T \mid E - T \mid T$
- $T \rightarrow T * F \mid Q$
- $Q \rightarrow (E) \mid a \mid E; E$

Question 32.1 Éliminer les récursivités à gauche de la grammaire G

8 Automates à piles

Exercice 33 : Automate à pile

Soit le langage $L = \{a^n b^p c^r, n = p \text{ ou } p = r\}$.

Question 33.1 Montrer simplement que ce langage est algébrique

Question 33.2 Donner un automate à pile qui reconnaisse L

Question 33.3 Est-ce que le mot $aabbccc$ est accepté par l'automate ? Si oui, faire fonctionner l'automate sur le mot $aabbccc$

Question 33.4 Est-ce que l'automate proposé est déterministe ?

Exercice 34 : Automate à pile et expressions bien parenthésées

Soit le langage L sur l'alphabet $\Sigma = \{ (,), a, b, + \}$ le langage des expressions bien parenthésées.

Question 34.1 Donner un automate à pile qui reconnaisse L

Question 34.2 Faire fonctionner l'automate sur le mot $((a + b) + (b + a))$

Question 34.3 Est-ce que l'automate proposé est déterministe ?

Exercice 35 : Automate à pile

On considère le langage L des mots sur l'alphabet $\{a, b\}$ qui contiennent autant de a que de b :

$$L = \{w, |w|_a = |w|_b\}$$

Par exemple, les mots suivants appartiennent à L : ε , ab , $baaabb$, $abbabaab$.

Question 35.1 Montrer que ce langage n'est pas régulier.

Question 35.2 Proposer un automate à pile sans ε -transition qui reconnaît le langage L . Expliquer votre démarche. Préciser :

- si l'automate à pile reconnaît les mots par pile vide ou par état final
- si l'automate à pile est déterministe