

Mathématiques pour l'informatique

Feuille de TD 1

Automates finis

Construction d'automates

Exercice 1

Construire un automate fini dont les états correspondent aux situations de famille possibles d'une personne (célibataire, marié, divorcé, veuf) et dont les flèches correspondent aux changements de situation possible. Etiqueter ces flèches par m (mariage), d (divorce) et v (veuvage). Les états finaux sont ceux où on peut mourir, et selon le code civil français on ne peut pas redevenir célibataire une fois qu'on a été marié.

Exercice 2

- a) Concevoir un automate reconnaissant un seul mot : « coucou ».
- b) Concevoir un automate reconnaissant le mot « coucou » mais aussi n'importe quel nombre non nul de « cou » qui se suivent : « cou », « coucou », « coucoucou »...
Attention : il ne doit pas reconnaître le mot vide !

Exercice 3

Construire des automates finis qui reconnaissent les langages suivants, et caractériser chaque automate comme déterministe ou non déterministe (*précision : dans le cadre de ces exos-ci, un automate non déterministe est aussi bon qu'un automate déterministe !*)

- (a) L'ensemble des mots sur l'alphabet $\{0, 1\}$ dont le dernier symbole est 0. Prendre en considération le fait que le mot « 0 » vérifie lui aussi la condition.
- (b) L'ensemble des mots sur l'alphabet $\{0, 1\}$ qui commencent et finissent par 1 ; construire un automate qui reconnaît ces mots et qui n'a qu'un seul état terminal. Prendre en considération le fait que le mot « 1 » vérifie lui aussi la condition.
- (c) L'ensemble des mots sur l'alphabet $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, +, -, .\}$ qui représentent en C les constantes numériques. Une constante numérique en C peut avoir une des structures suivantes :
 - Elle doit contenir au moins un chiffre ;
 - Elle peut contenir un signe (+ ou -) (un seul) mais uniquement au début ;
 - Elle peut contenir un point (.) (un seul) qui, s'il est présent, doit être précédé **et/ou** suivi par au moins un chiffre.
- (d) L'ensemble des mots sur l'alphabet $\{0, 1\}$ qui comportent au moins une fois le motif '10' et au moins une fois le motif '01' (les deux motifs ont le droit de se croiser, de façon que les séquences '010' et '101' vérifient cette condition).
- (e) $\{a^n b^m \mid n + m \text{ impair}\}$ (Ceci veut dire : une séquence de caractères sur l'alphabet $\{a, b\}$ telle que s'il a des a , ils précèdent à tous les b s'il y en a ; le nombre total de caractères est impair).

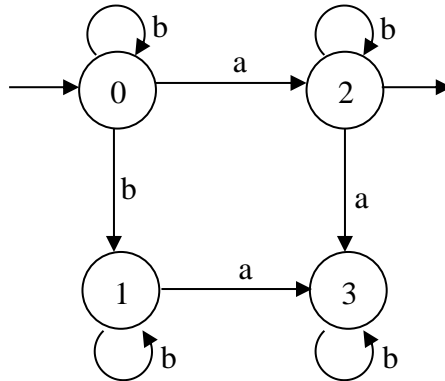
Exercice 4

Construire un automate fini reconnaissant les entiers écrits en base 2 divisibles par 7.

Tuyau : s'inspirer de la solution donnée en cours pour la divisibilité par 3

Exercice 5

Quel est le langage reconnu par l'automate suivant ? Quels états de cet automate sont inutiles, dans le sens que le langage reconnu par l'automate est le même qu'ils soient présents ou non ?



Standardisation

Exercice 6

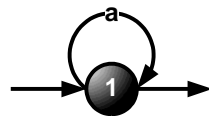
Pour chacun des automates suivantes 6.1 à 6.7,

(a) dire s'il est standard,

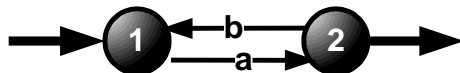
(b) s'il ne l'est pas, le standardiser,

(c) s'il reconnaît le mot vide, construire l'automate qui reconnaît le même langage à l'exception du mot vide :

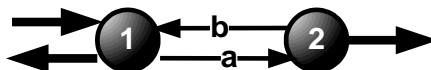
6-1



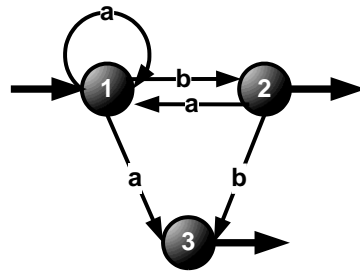
6.2



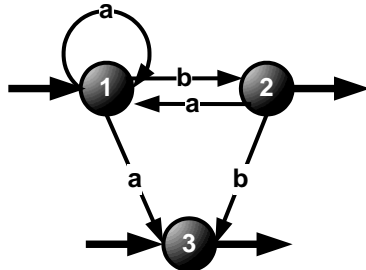
6.3



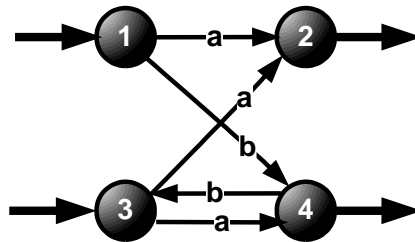
6.4



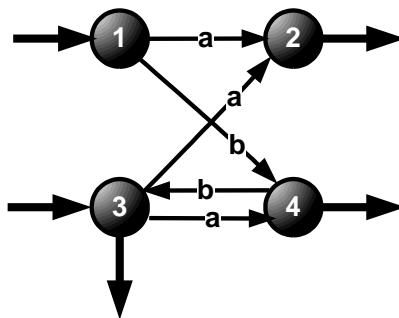
6.5



6.6



6.7



Déterminisation

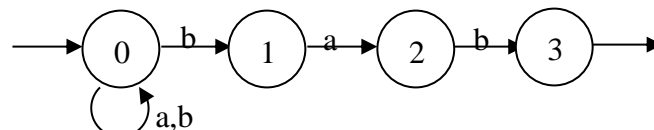
Exercice 7

Voici cinq automates non déterministes notés A, B, C1, C2, C3, D. L'alphabet est $\Sigma = \{a, b\}$.

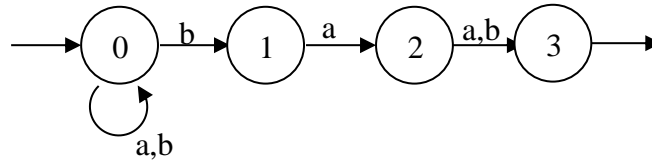
Pour chacun de ces automates :

- Décrire le langage qu'il reconnaît (vous ne savez pas encore comment écrire des expressions rationnelles ; faites donc une description de l'ensemble des mots reconnus en langage ordinaire).
- Trouver l'automate déterministe équivalent et le compléter si besoin est.

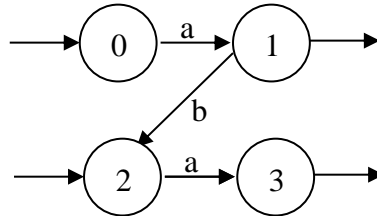
Automate A :



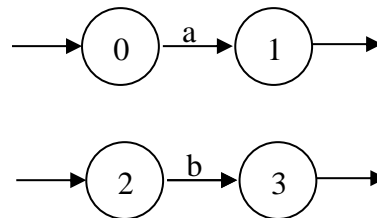
Automate B :



Automate C1:

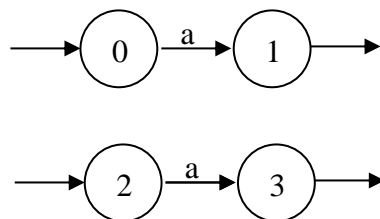


Automate C2:



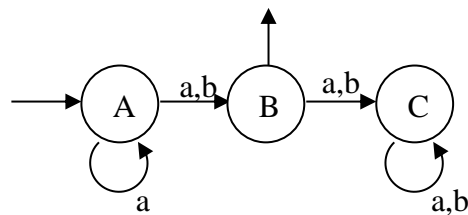
(Cet automate consiste en deux parties non connexes)

Automate C3:



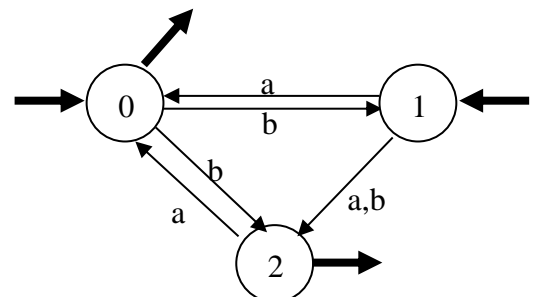
(Cet automate consiste en deux parties non connexes)

Automate D :



Exercice 8

Déterminer et compléter l'automate suivant :
(Etats initiaux: 0 et 1 ; états terminaux : 0 et 2)



Langage complémentaire

Exercice 9

Construire des automates déterministes qui reconnaissent les langages suivants sur l'alphabet $\{0, 1\}$. En déduire des automates déterministes qui reconnaissent les complémentaires de ces langages :

- (a) L'ensemble des mots qui contiennent exactement quatre fois le symbole 0.
- (b) L'ensemble des mots qui contiennent au moins un 1.
- (c) Le mot vide et les mots '0', '1' et '01'.

Langage complémentaire + minimisation

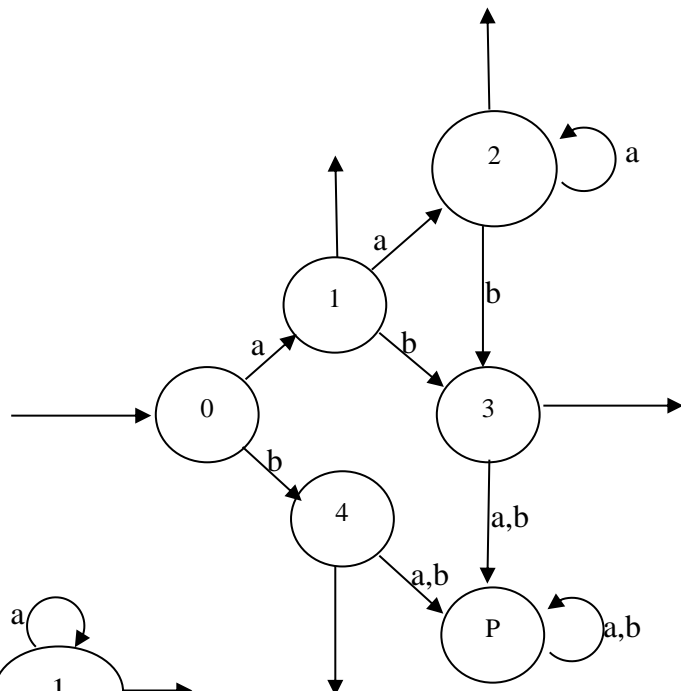
Exercice 10

Construire un automate fini reconnaissant l'ensemble des mots sur l'alphabet $A = \{a, b\}$ qui **ne se terminent pas** par $bbaa$. Le déterminer (si l'automate que vous avez construit n'est pas déterministe) et minimiser.

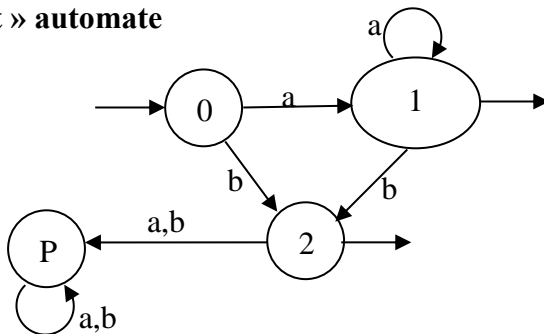
Exercice 11

Minimiser les deux automates déterministes équivalents obtenus en Exo 7D (ici, nous avons renommé les états pour plus de simplicité) :

Le « grand » automate

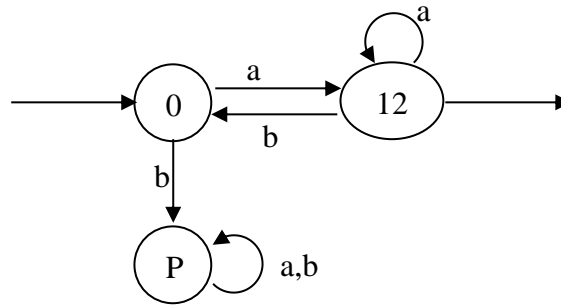


Et le « petit » automate



Exercice 12

Minimiser l'automate suivant :


Exercice 13

Déterminer et minimiser l'automate suivant :

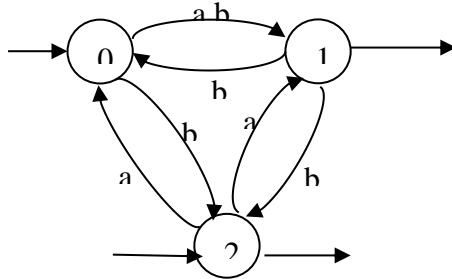


Table de transitions :

	état	a	b
E	0	1	1,2
S	1	--	0,2
E/S	2	0,1	--

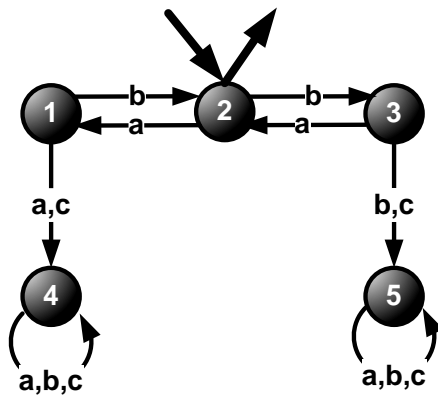
Exercice 14

Minimiser l'automate suivant, puis construire l'automate minimal reconnaissant le langage complémentaire à celui que reconnaît l'automate d'origine :

	état	a	b	c
E	0	4	5	1
	1	5	4	0
S	2	0	--	3
S	3	1	--	6
	4	0	3	5
	5	1	2	4
S	6	0	--	3

Exercice 15

Minimiser l'automate suivant :



Exercice supplémentaire

On définit la famille d'automates suivants :

$$\tilde{A}_n = (Q_n, I, T, E), n \geq 1$$

avec

- $A = \{a, b\}$
- $Q_n = \{0, 1, \dots, n-1\}$
- $I = T = \{0\}$
- Comme flèches étiquetées par **a** l'ensemble de $(q.a.((q+1) \bmod n))$ pour $\forall q : 0 \leq q \leq n-1$
- Comme flèches étiquetées par **b** l'ensemble de $(q.b.0)$ et $(q.b.q)$ pour $\forall q : 1 \leq q \leq n-1$
(attention : la première inégalité commence par 0, et la seconde, par 1)

Dessiner \tilde{A}_3 et \tilde{A}_4 .

Puis montrer que le déterminisé complet de \tilde{A}_n a toujours 2^n états.