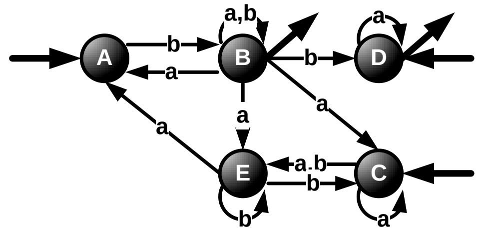
DE Mathématiques pour l'Informatique

Exercice 1. A = {a,b} est l'alphabet.

Soit l'automate fini A_{init} :

	état	a	b
Entrée	A	--	B
Sortie	B	A, B, C, E	B,D
Entrée	C	C, E	E
Entrée	D	D	--
Sortie	E	A	C, E



a) L'automate A_{init} est-il déterministe ? pourquoi ? Si la réponse est « non », donnez toutes les raisons.

Réponse : il n'est pas déterministe car 1) il y a 3 entrées, 2) il y a plusieurs transitions marquées par les mêmes caractères sortant du même état :
BaA, BaB, BaC, BaE

BbB, BbD

CaC,CaE

EbC, EbE

b) Construisez un automate standard A_{st} en standardisant A_{init} .

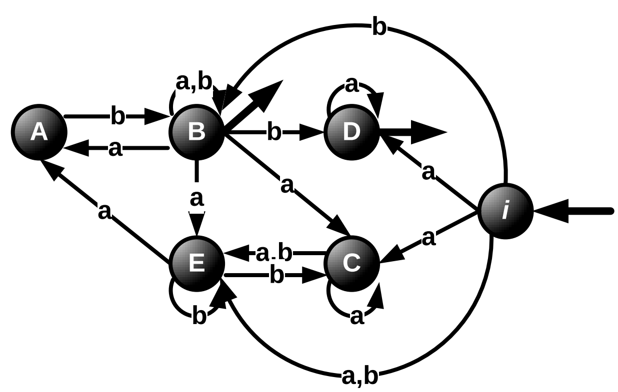
Solution : on ajoute un nouvel état initial *i* , avec les transitions induites par les transitions de A_{init} partant de ses entrées :

AbB		ibB
CaC		iaC
CaE	⟶	iaE
CbE		ibE
DaD		iaD

L'état *i* n'est pas terminal, car aucun état initial de A_{init} ne l'est.

Voici l'automate A_{st} :

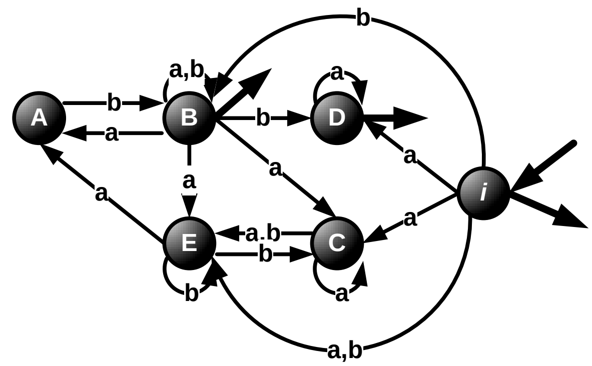
	état	a	b
Entrée	i	C,D,E	B,E
	A	--	B
Sortie	B	A, B, C, E	B,D
	C	C, E	E
	D	D	--
Sortie	E	A	C, E



- Si vous pensez que A_{init} reconnaît le mot vide, expliquez pourquoi et construisez un autre automate standard B_{st} qui ne reconnaît le mot vide mais reconnaît tout le reste du langage reconnu par A_{init}.
- Si vous pensez que A_{init} ne reconnaît pas le mot vide, expliquez pourquoi et construisez un autre automate standard C_{st} qui reconnaît et le mot vide, et tout le langage reconnu par A_{init}.

Solution : A_{init} ne reconnaît pas le mot vide car aucune de ses entrées n'est un état terminal. Voici l'automate C_{st} :

	état	a	b
E/S	i	C,D,E	B,E
	A	--	B
S	B	A, B, C, E	B,D
	C	C, E	E
	D	D	--
S	E	A	C, E



c) Indépendamment des questions (a) et (b), obtenir un automate déterministe complet équivalent à A_{init}.

Solution :

	état	a	b
E	ACD	CDE	BE
S	CDE	ACDE	CE
S	BE	ABCE	BCDE
S	ACDE	ACDE	BCE
S	CE	ACE	CE
S	ABCE	ABCE	BCDE
S	BCDE	ABCDE	BCDE
S	BCE	ABCE	BCDE
S	ACE	ACE	BCE
S	ABCDE	ABCDE	BCDE

L'automate déterministe est trop grand pour qu'il y ait un sens de le dessiner.

Exercice 2

Minimiser l'automate suivant :

	État	a	b
Entrée/sortie	A	B	C
Sortie	B	--	--
Sortie	C	F	D
Sortie	D	E	B
Sortie	E	F	D
Sortie	F	B	C

Solution : Partition initiale : Θ_0 = {T, NT}, NT=P, T={A,B,C,D,E,F}.

	État	a	b	a	b	
NT	A	B	C	NT	NT	
	B	P	P	P	P	fin
	C	F	D	NT	NT	
	D	E	B	NT	NT	
	E	F	D	NT	NT	
	F	B	C	NT	NT	
T	P	P	P	fin		

Θ_1 = {(P),(B), I},où j'ai appelé I={A,C,D,E,F}.

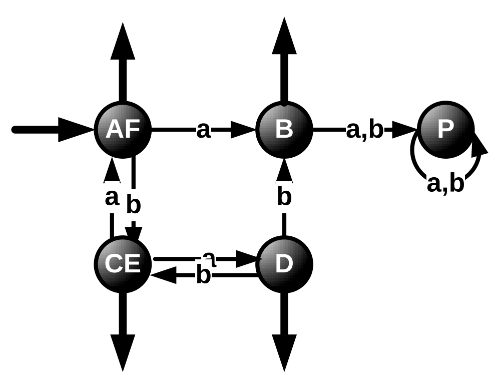
	État	a	b	a (sous Θ_1)	b (sous Θ_1)
I	A	B	C	B	I
	C	F	D	I	I
	D	E	B	I	B
	E	F	D	I	I
	F	B	C	B	I

Θ_2 = {(P),(B), (A , F), (C,E), (D)}.

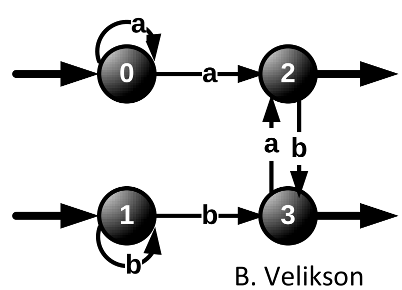
On note que A et F ont les mêmes transitions (vers B en a et vers C en B), et C et E ont les mêmes transitions (vers F en a et vers D en b). Donc ils ne se sépareront pas. Donc Θ_{fin} =Θ_2 ={(P),(B), (A , F), (C,E), (D)}.

La table de transitions de l'automate minimisé :

	État	a	b
E/S	AF	B	CE
S	B	P	P
S	CE	AF	D
S	D	CE	B
	P	P	P



Exercice 3



a) Donner le système d'équations permettant de trouver l'expression rationnelle correspondant au langage reconnu par cet automate.

Solution :

$$\begin{cases} 0 = \varepsilon + 0a \\ 1 = \varepsilon + 1b \\ 2 = 0a + 3a \\ 3 = 1b + 2b \end{cases}$$

$L = 2 + 3$

b) Résoudre ce système d'équations et obtenir le langage reconnu par l'automate.

Solution :

$0 = \varepsilon a^* = a^*$; $1 = \varepsilon b^* = b^*$;
$2 = 0a + 3a = 0a + (1b + 2b)a = a^*a + b^*ba + 2ba = (a^*a + b^*ba)(ba)^*$
$3 = 1b + 2b = 1b + (0a + 3a)b = b^*b + a^*ab + 3ab = (b^*b + a^*ab)(ab)^*$

$L = 2 + 3 = (a^*a + b^*ba)(ba)^* + (b^*b + a^*ab)(ab)^*$

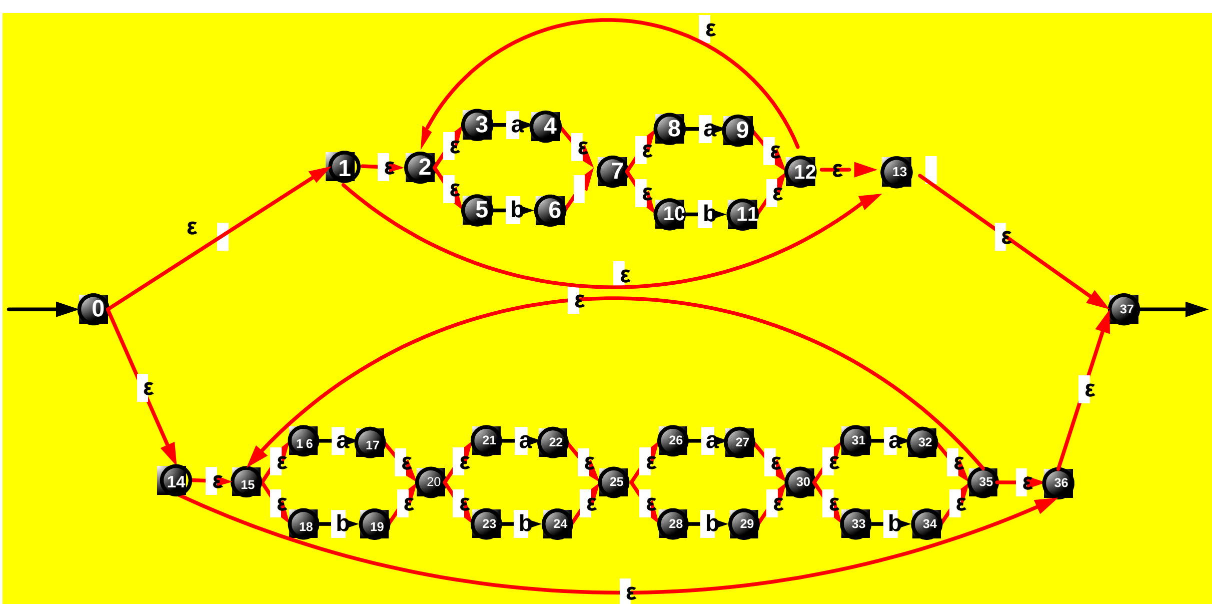
(Il est possible que vous ayez obtenu une expression équivalente, donc correcte, mais d'une forme différente). Entre autres, on peut beaucoup jouer sur l'identité $a(ba)^* = (ab)^*a$:

$a(ba)^* = a(\varepsilon + ba + baba + bababa + \ldots) = a + aba + ababa + abababa + \ldots) = (\varepsilon + ab + abab + ababab + \ldots)a = (ab)^*a$.

Exercice 4

a) Construire un automate asynchrone reconnaissant le langage
$$L = \{((a+b)(a+b))^* + ((a+b)(a+b)(a+b)(a+b))^*\}$$

Solution :



Ensuite**, soit**

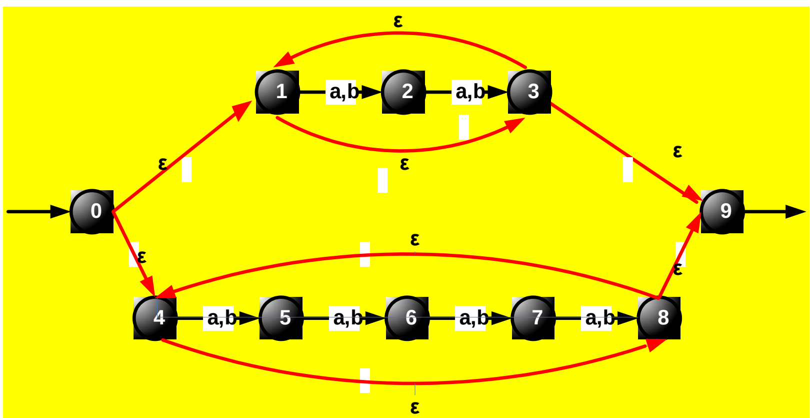
b) déterminiser et
c) minimiser
} l'automate obtenu en (a) (c'est assez compliqué !),

soit

b) simplifier graphiquement et déterminiser et
c) minimiser
} l'automate simplifié graphiquement.

La deuxième proposition est bien plus simple à réaliser.

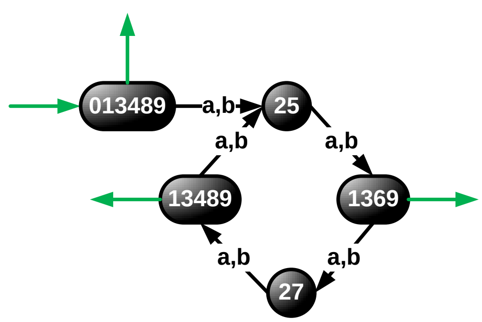
Solution : J'ai choisi de déterminiser un automate simplifié graphiquement :



Attention ! Si vous avez produit un automate simplifié correct mais sans effectuer une déterminisation, ce dessin ne vous donne aucun point.

Déterminisation :

	état	a,b
E/S	013489	25
	25	1369
S	1369	27
	27	13489
S	13489	25



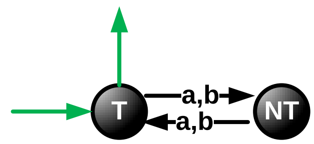
Minimisation :

Θ_0={T,NT} où T={013489,1369,13489} et NT={25,27}.
Aucune séparation ne se produit sous Θ_0 , donc $\Theta_1 = \Theta_0 = \Theta_{fin}$.

		a,b	sous Θ_0 : a,b
T	13489	25	NT
	1369	27	NT
	13489	25	NT
NT	25	1369	T
	27	13489	T

L'automate minimal n'a que 2 états :

	état	a,b
E/S	T	NT
	NT	T

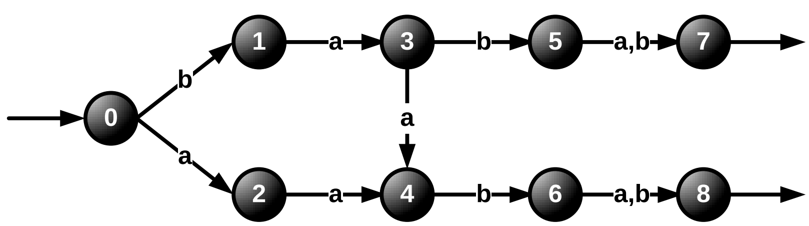


Remarque. Vous avez aussi le droit de produire directement l'automate déterministe complet minimal que vous devriez obtenir en (c), si vous savez le faire *<u>et si vous savez l'expliquer</u>*. Dans ce cas, vous n'êtes pas obligés de faire le (b) et le (c), mais vous devez toujours faire le (a).

> **Réalisation de la remarque** : Nous avons vu que l'automate minimal dont le langage est $(c^n)^*+(c^n)^*$ consiste en un cycle de PCEM(n,m) états si l'un des nombres n'est pas un multiple de l'autre, et de n états si m=kn. Ici, nous avons ce dernier cas, car 4=2*2. La ou les sorties sont dans les états multiples de n ; ici, c'est uniquement l'état 0, que l'on a appelé T lors de la minimisation. Il en sort l'automate minimal ci-dessus.

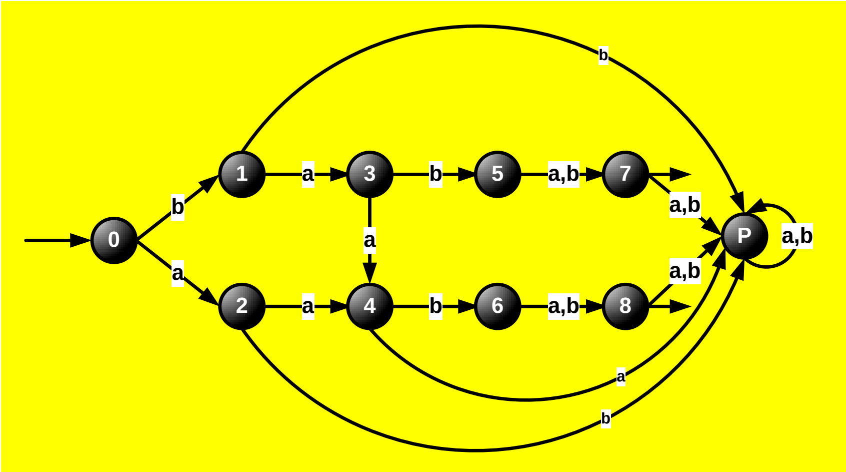
Exercice 5

Construire un automate reconnaissant le complémentaire du langage reconnu par cet automate :

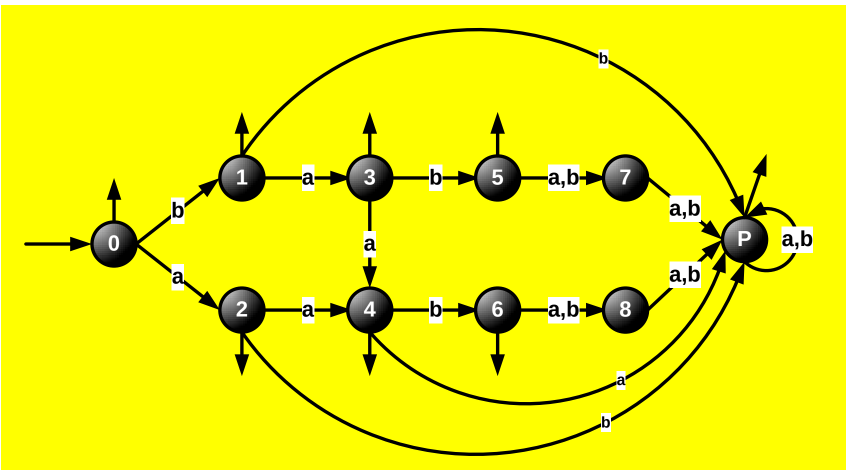


Vous donnerez votre résultat sous forme d'un schéma (un dessin).

Solution : il faut d'abord compléter l'automate :



Ce n'est que maintenant qu'on peut faire l'opération T ↔NT :



Exercice 6

Prouver par récurrence : $\forall n \geq 1, \quad \sum_{i=1}^{n} i(i+1) = \frac{n(n+1)(n+2)}{6}$.

Solution :

<u>En fait</u>**<u>, la formule est fausse.</u>** Déjà pour $n = 1$, $\sum_{i=1}^{n} i(i+1) = 1 \times (1+1) = 2 \neq \frac{1(1+1)(1+2)}{6} = 1.$

Toute personne ayant vu que la formule est fausse, obtient le point pour cet exo ; mais ceux qui ont pu corriger la formule pour qu'elle soit vraie, et qui ont prouvé la formule correcte, obtiennent plus.

On voit que pour corriger la formule pour n=1, il suffit de la modifier ainsi :

$\forall n \geq 1, \quad \sum_{i=1}^{n} i(i+1) = \frac{n(n+1)(n+2)}{3}$. La base est donc assurée.

Hérédité : Soit $P(n) = \left(\sum_{i=1}^{n} i(i+1) = \frac{n(n+1)(n+2)}{3} \right) = TRUE$. Alors

Si $P(n)$=TRUE

$$\sum_{i=1}^{n+1} i(i+1) = \sum_{i=1}^{n} i(i+1) + (n+1)(n+2) = \frac{n(n+1)(n+2)}{3} + (n+1)(n+2) = (n+1)(n+2)\left(\frac{n}{3}+1\right)$$
$$= (n+1)(n+2)\frac{n+3}{3} = \frac{(n+1)(n+2)(n+3)}{3}$$

ce qui est le contenu de la proposition $P(n+1)$.

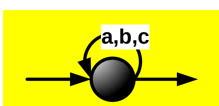
Questions de cours

a) Quel est l'automate minimal équivalent à l'automate suivant :

	état	a	b	c
E/S	0	1	2	3
S	1	2	3	4
S	2	3	0	1
S	3	0	1	2
S	4	4	0	1

Donnez une réponse immédiate, **sans effectuer la procédure de minimisation par des partitions successives,** mais avec une explication !

Réponse : c'est un automate déterministe complet dont tous les états sont terminaux. Donc il reconnait le langage A*, et son automate déterministe complet minimal est



b) Pour un langage donné, un automate déterministe complet qui le reconnaît est-il unique ? **NON**

c) Pour un langage donné, un automate déterministe complet minimal qui le reconnaît est-il unique ? **OUI**

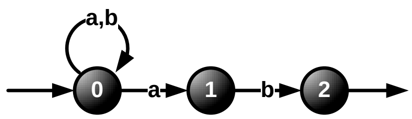
d) Peut-on toujours construire un automate déterministe équivalent à un automate non déterministe ? **OUI**

e) Un automate standard est-il toujours déterministe ? **NON**

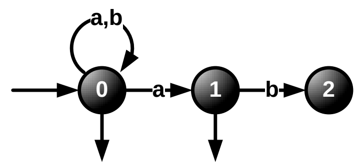
f) Un automate déterministe est-il toujours standard ? **NON**

g) Le langage reconnu par l'automate B, est-il le complément $\overline{L}$ du langage L reconnu par l'automate A ? Expliquer votre réponse (pas de points s'il n'y a pas d'explication correcte). Si la réponse est « non », essayez de trouver un mot qui fait partie de $\overline{L}$ mais qui n'est pas reconnu par l'automate B, ou un mot qui ne fait pas partie de $\overline{L}$ mais qui est reconnu par B.

A



B



Réponse : Non, car l'automate A n'est pas déterministe (il ne faut pas dire qu'il n'est pas complet parce qu'on ne parle pas d'un automate non déterministe complet). Exemple : le mot 'ab' est reconnu et par l'automate A, et par l'automate B .