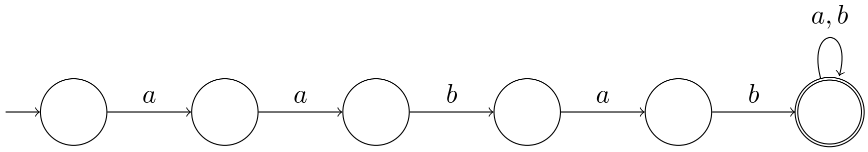
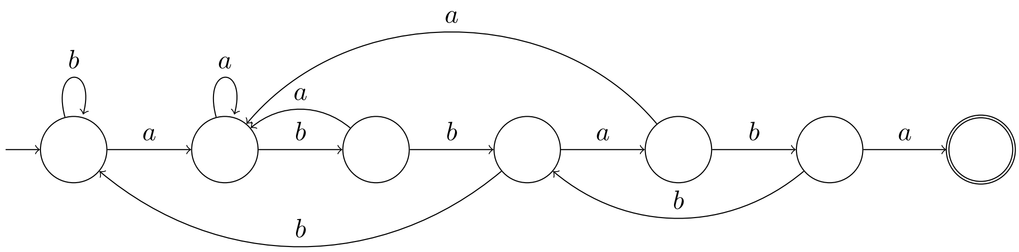
Feuille d'exercice 1

Exercice 1 – Donnez des automates reconnaissants :

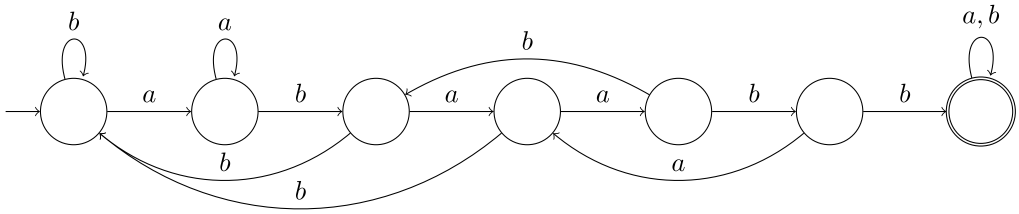
- Les mots commençant par *aabab*.



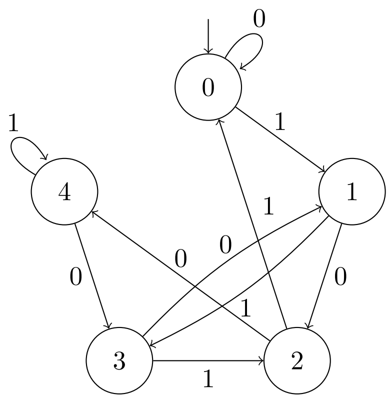
- Les mots finissant par *abbaba*.



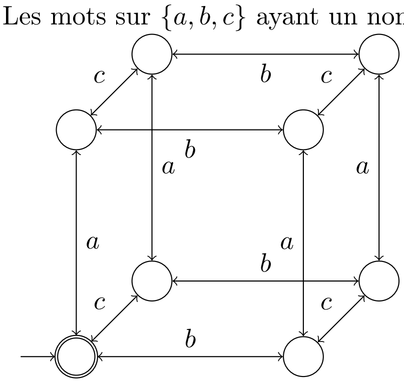
- Les mots contenant *abaabb*.



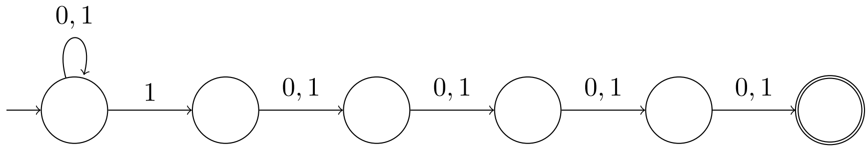
- Les représentations en binaire des multiples de 5.



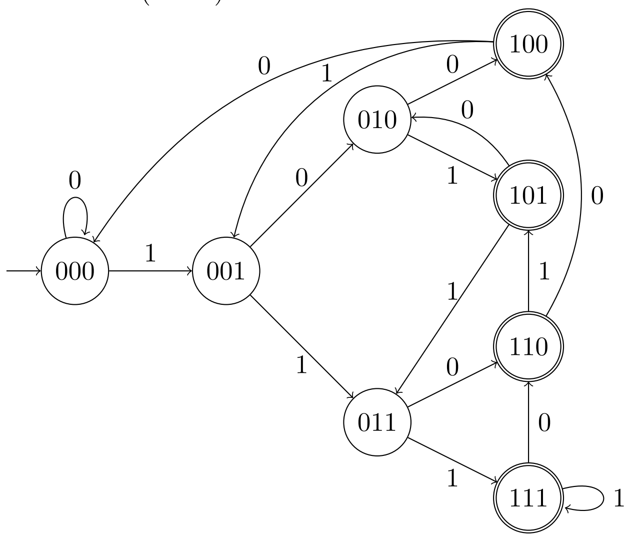
- Les mots sur $\{a, b, c\}$ ayant un nombre pair de a, pair de b et pair de c.



- les mots sur $\{0,1\}$ donc la k^{ieme} lettre avant la fin est 1 (k est une constante)
  — Non déterministe ($k = 5$) :



  On a $k+1$ états
  — déterministe ($k = 3$) :



  On a 2^k états

Exercice 2 – Soit deux automates déterministes $\mathcal{A}$ et $\mathcal{B}$

Pour savoir si un mot est dans $L(\mathcal{A}) \cap L(\mathcal{B})$ on parcours $\mathcal{A}$ et $\mathcal{B}$ en parallèle.
Pendnat le parcours on a donc une paire (q_a, q_b) en mémoire.
On peut alors contruire l'automate produit

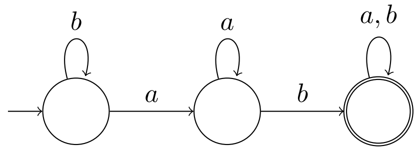
$$\text{Automate Produit} = \begin{cases} Q & = Q_a \times Q_b \\ Q_0 & = ({q_0}_a, {q_0}_b) \\ F & = F_a \times F_b \\ T & = \left\{ (pq) \xrightarrow{a} (\bar{p}\bar{q}) | p \xrightarrow{a} \bar{p} \wedge q \xrightarrow{a} \bar{q} \right\} \end{cases}$$

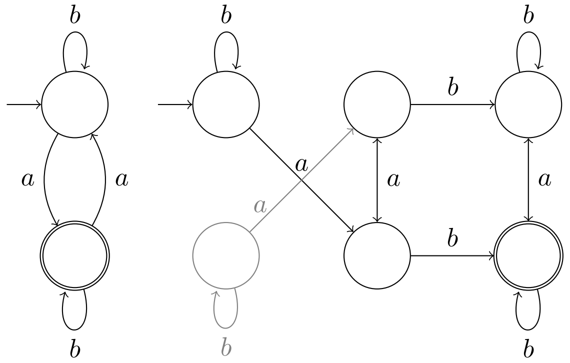
Cette automate à donc $|Q_a| \times |Q_b|$ états.
On peut montrer $m \in L(\mathcal{A}) \cap L(\mathcal{B}) \Leftrightarrow m \in L(\mathcal{A} \times \mathcal{B})$

$\Rightarrow$: On suppose que $m \in L(\mathcal{A}) \cap L(\mathcal{B})$
  On a donc ${q_0}_a \xrightarrow{m} {q_F}_a \wedge {q_0}_b \xrightarrow{m} {q_F}_b$
  Par contruction de $\mathcal{A} \times \mathcal{B}$ on a bien un chemin ${q_0}_a {q_0}_b \xrightarrow{m} {q_F}_a {q_F}_b$

$\Leftarrow$: On suppose que $m \in L(\mathcal{A} \times \mathcal{B})$
  On a donc ${q_0}_a {q_0}_b \xrightarrow{m} {q_F}_a {q_F}_b$
  On peut donc bien contuire une chemin dans $\mathcal{A}$ et dans $\mathcal{B}$
  — chemin dans $\mathcal{A}$ on prend la première coordonée
  — chemin dans $\mathcal{B}$ on prend la seconde coordonée

On veut construire un automate qui reconnait les mots sur $\{a, b\}$ ayant un nombre pair de a et contenant ab





On peut voir que l'état en bas à gauche est inaccessible.

On remarque dans la preuve qu'on n'est jamais obligé de se restraindre à des automates deterministe complet. Ainsi pour les automates non déterministes il sufit juste d'utiliser le même algo.

Pour l'union on peut utiliser cette construction :

$$\text{Automate Produit} = \begin{cases} Q & = Q_a \times Q_b \\ Q_0 & = (q_{0_a}, q_{0_b}) \\ F & = \{(q_a, q_b) | q_a \in F_a \vee q_b \in F_b\} \\ T & = \left\{(pq) \xrightarrow{a} (\bar{p}\bar{q}) | p \xrightarrow{a} \bar{p} \wedge q \xrightarrow{a} \bar{q}\right\} \end{cases}$$

On peut montrer $m \in L(\mathcal{A}) \cup L(\mathcal{B}) \Leftrightarrow m \in L(\mathcal{A} \times_\cup \mathcal{B})$

$\Rightarrow$: On suppose que $m \in L(\mathcal{A}) \cup L(\mathcal{B})$

- Si on a un chemin acceptant m dans A :
  On a la première coordonée du chemin dans dans l'automate. Ensuite pour trouver la deuxième coordonée on prend un chemin dans B qui lit m. Mais pour qu'un telle chemain existe quelque soit l'automate il faut qu'il soit complet.
- Si on a un chemin acceptant m dans B :
  On fait la même chose que pour A et on se retrouve avec la même hypothèse pour A (il doit être complet)

$\Leftarrow$: On suppose que $m \in L(\mathcal{A} \times_\cup \mathcal{B})$

On a donc $q_{0_a} q_{0_b} \xrightarrow{m} q_{n_a} q_{n_b}$

On peut donc bien contuire une chemin dans $\mathcal{A}$ ou dans $\mathcal{B}$ selon si : $q_{n_a} \in F_A$ ou si $q_{n_a} \in F_B$ en prennant la première coordonée si $q_{n_a} \in F_A$ (chemin acceptant m dans $\mathcal{A}$).

Même chose pour l'autre cas en prennant la seconde coordonnée.

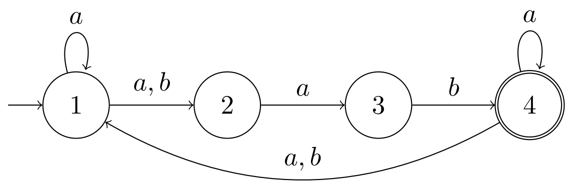
Exercice 3 – Algo : On parcours l'automate et on mémorise l'ensembles des états attégnables. On a focément une mémoire finie car il y a un nombre finie d'états

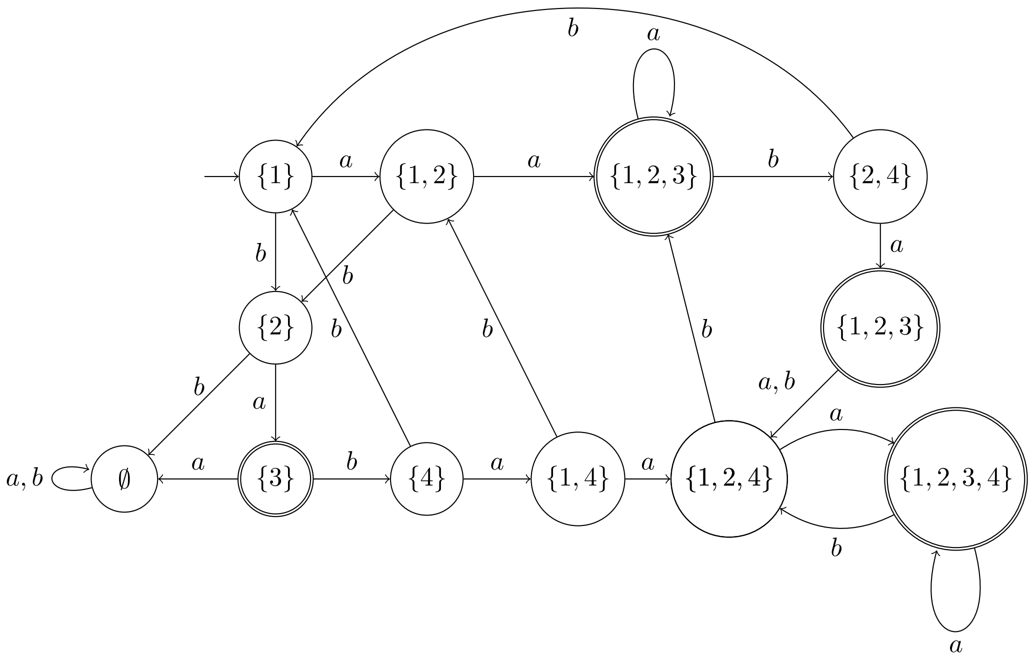
$$\begin{cases} \bar{Q} & = \mathcal{P}(Q) \\ \bar{Q}_0 & = \{Q_0\} \\ \bar{F} & = \left\{q \in \bar{Q} | \exists \bar{q} \in q, \bar{q} \in Q_f\right\} = \{P | P \cap F \neq \emptyset\} \\ \bar{T} & = P \xrightarrow{a} \bigcup_{p \in P} \left\{q | p \xrightarrow{a} q\right\} \end{cases}$$

Il y a donc $2^{|Q|}$ état après la determinisation (On peut enlever ceux inaccessible).

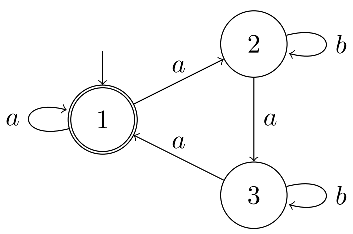
Exercice 4 – Determinisez :

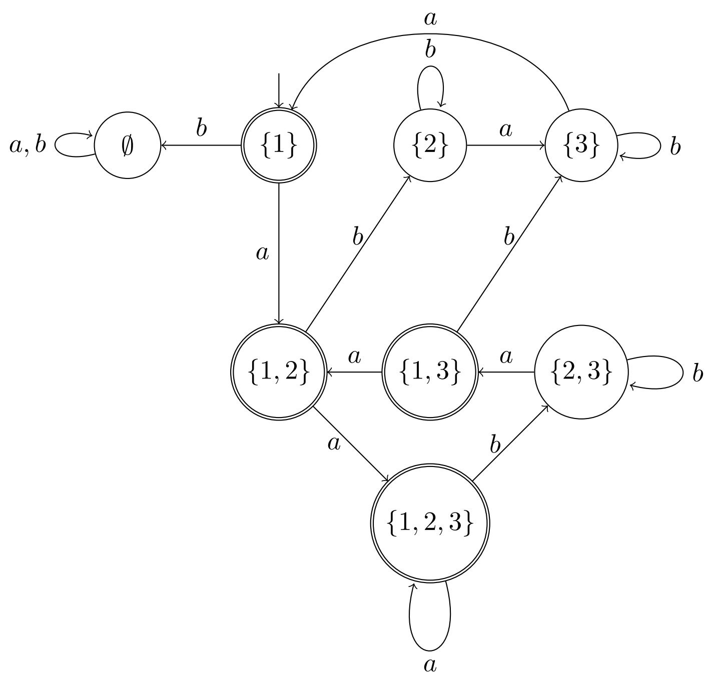
1. Automate :





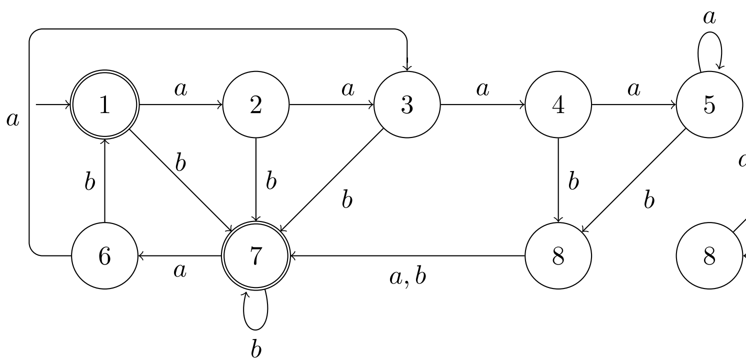
2. automate :

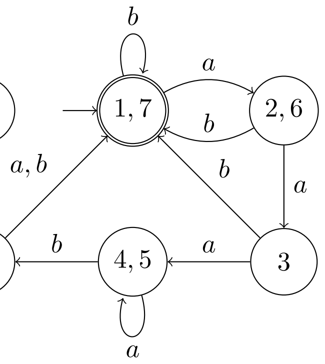




Exercice 5 – Minimisez :

1. Automate à minimiser :



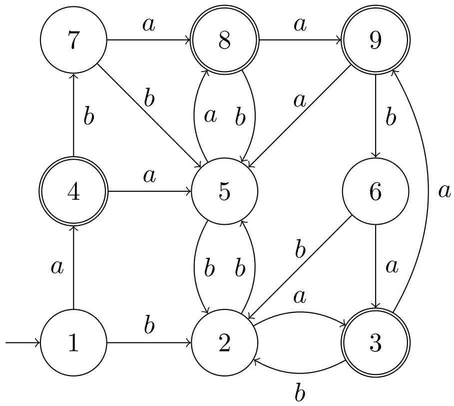


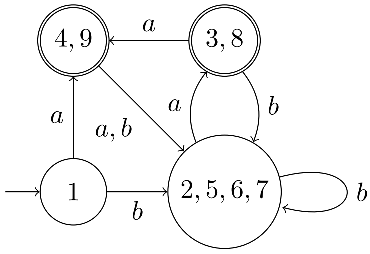
On applique l'algo de minimisation :

	1	2	3	4	5	6	7	8
a	2	3	4	5	5	3	6	7
b	7	7	7	8	8	1	7	7

i	classes
0	17/234568
1	17/236/45/8
2	17/236/45/8
3	17/26/3/45/8
4	17/26/3/45/8

2. Automate à minimiser :





On applique l'algo de minimisation :

	1	2	3	4	5	6	7	8	9
a	4	3	9	5	8	3	8	9	5
b	2	5	2	7	2	2	5	5	6

i	classes
0	3478/12567
1	38/49/12567
2	38/49/1/2567

Exercice 6 – • $L = \{a^n b^n | n \in \mathbb{N}\}$

On cherche le $Future_L(u)$ de chaque mot $u \in \{a,b\}^*$

u	$Future_L(u)$
ε	$a^n b^n$
a	$a^n b^{n+1}$
a^2	$a^n n^{n+2}$
...	...
$\forall p \geq 0, a^p$	$a^d b^{p+d}$
...	...
$a^n b$	b^{n-1}
...	...
$\forall p \geq 0, d > 0, a^{p+d} b^d$	a^p
...	...
$a^n b^n$	ε
reste	$\emptyset$

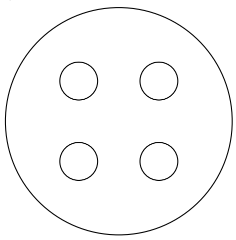
On a $a^p b$ et $a^q b$ avec $p < q$. En parant de q_0 c'est de mots ne peuvent pas conduire au même état. car ils n'ont pas le même future.
L n'est pas reconnaissable car on a un nombre infinie de classes.

- On construit l'automate déterministe complet et propre qui se souvient des k dernière lettres lue (il a 2^k états).
Soit p et q deux états différents. Ils se souvient alors de u et v ($u \neq v$).
Comme $u \neq v$ on à $u = u_1 a u_2$ et $v = v_1 b v_2$ avec $|u_1| = |v_1| = x$ et $|u_2| = |v_2| = y$.
On à $T(p, a^x)$ non final et $T(q, a^x)$ final donc p et q non pas le même future donc p et q ne peuvent pas être fusioné. Donc l'automate à 2^k états est minimal.

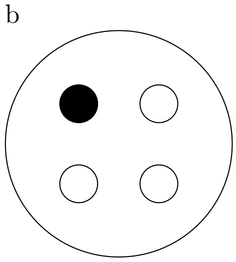
Exercice 7 – Le Barman aveugle

1. Il y a 2^4 configurations. La pluspart sont équivalentes Au final on a que 4 configurations possibles.

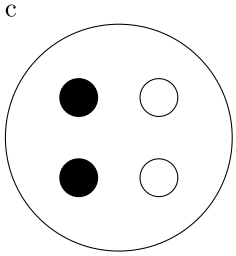
a



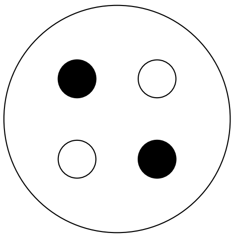
b



c

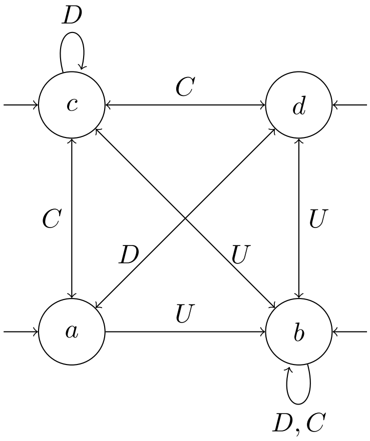


d

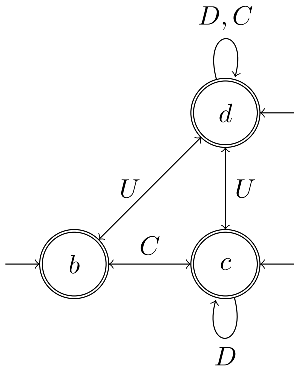


Le barman peut en retrouner un (U), peut retourner une colonne (C) et une diagonale (D). Il n'a donc que 3 mouvements.

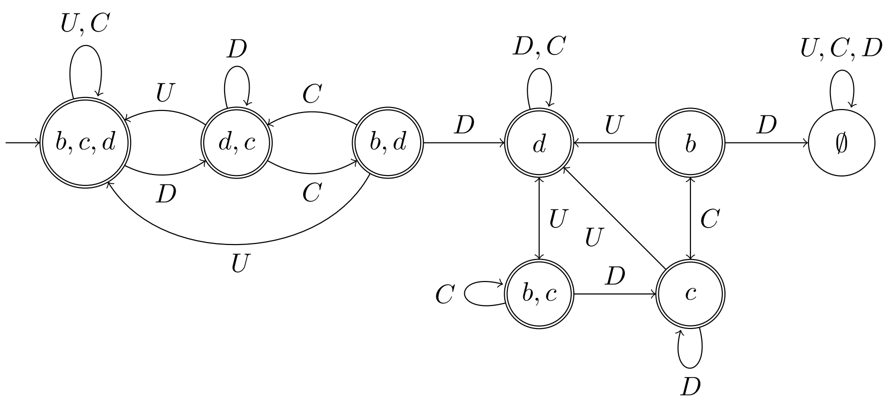
2. On peut donc construire un automate qui représente les différents états du jeu :



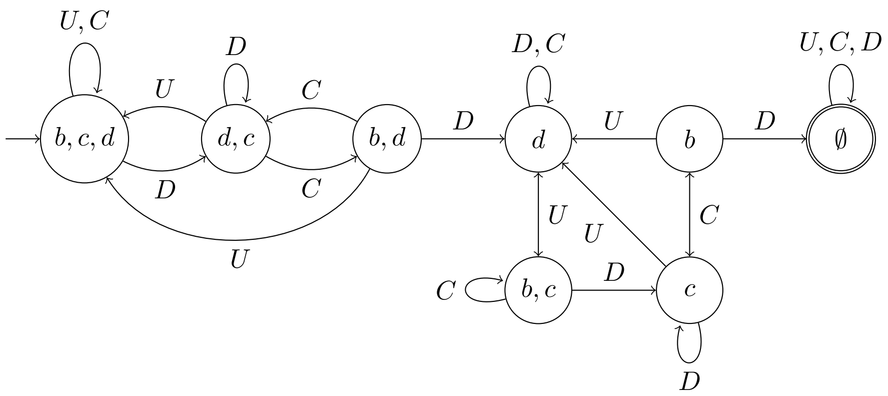
3. Séquence pour laquel le client gagne :



On déterminise et on complète l'automate pour l'exercice suivant



4. Pour avoir la contraposer d'un langage on inverse juse les états terminaux et les états normaux. En faisait ça sur l'automate présédent on obtient la liste des coups pour que le barman gagne



On peut donc conclure que ce n'est pas une bonne idée de jouer avec le barman.