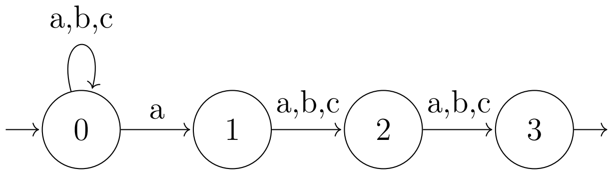
# Corrigé TD2. Automates

## I. Automates et langages

**Exercice 1. Vocabulaire sur les automates**
**1.**




**2.** Cet automate n'est pas déterministe : deux transitions depuis l'état 0 sont étiquetés par la lettre $a$, c'est-à-dire $\delta(0, a) = \{0, 1\}$.
Cet automate n'est pas complet puisque la fonction de transition est partielle. Par exemple, $\delta(3, a)$ n'est pas défini.
**3.** Le mot *baba* est reconnu par le chemin $0 \xrightarrow{b} 0 \xrightarrow{a} 1 \xrightarrow{b} 2 \xrightarrow{a} 3$.
Le mot *cabcb* n'est pas reconnu par l'automate.
**4.** $L(A)$ correspond aux mots ayant un $a$ en troisième position en partant de la fin.

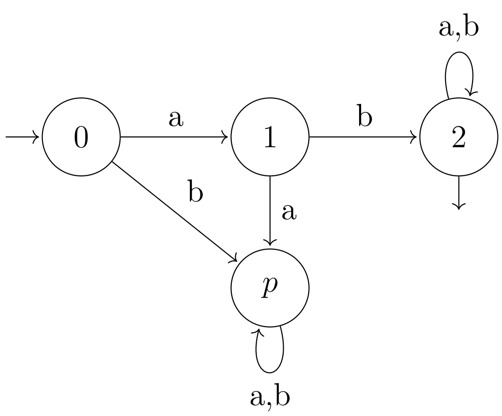
**Exercice 2. Langages reconnus**

1. Cet automate reconnaît les mot terminant par *ba*.
2. Cet automate reconnaît les mots de longueur paire.
3. Cet automate reconnaît les mots qui ne comporte pas le facteur *bbbb*.
4. Cet automate reconnait les mots qui commencent et terminent par la même lettre.

**Exercice 3. Construction d'automates**
*Corrigé en cours.*

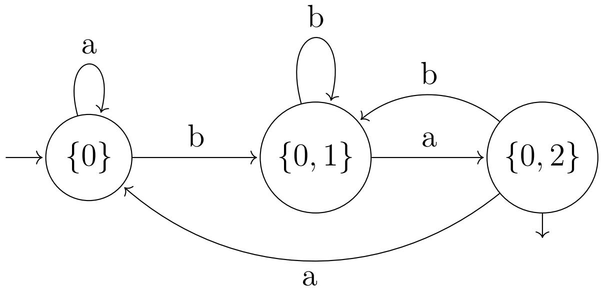
## II. Transformations en automates équivalents

**Exercice 4. Complétion d'automates**
**1.** La fonction de transition est partielle. Par exemple, $\delta(0, b)$ n'est pas défini.
**2.** Cet automate reconnaît les mots commençant par *ab*.
**3.** On ajoute un état puit, comme vu en cours :

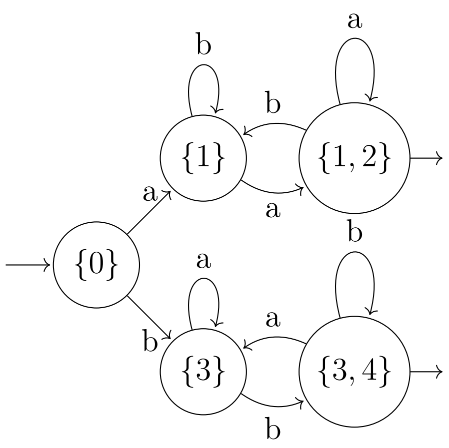


**Exercice 5. Déterminisations**

**1.** Déterminisé de l'automate 1 :

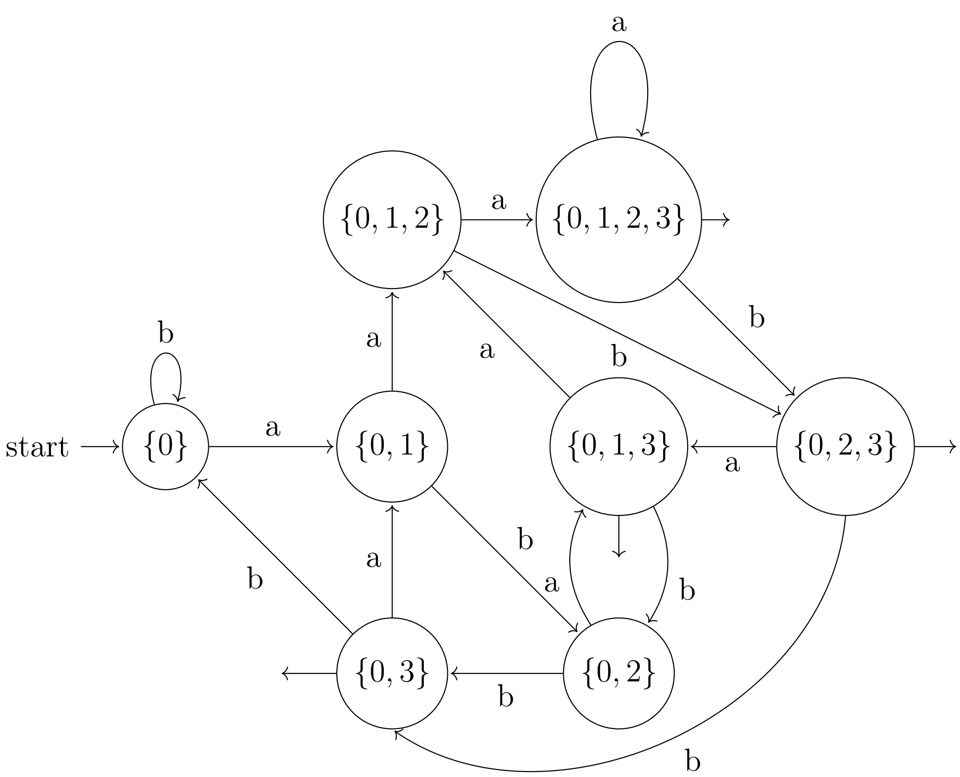



Déterminisé de l'automate 4 :

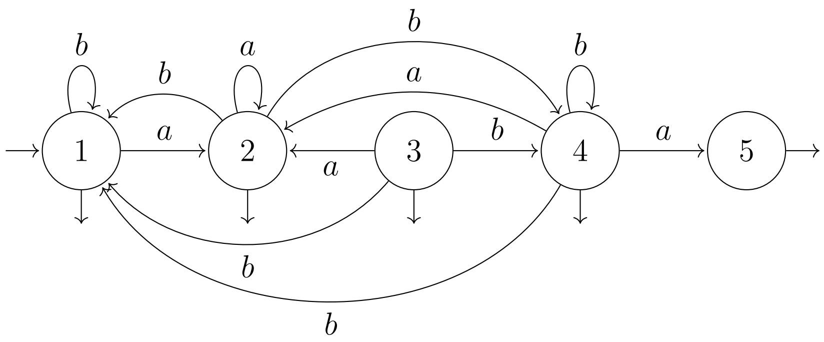



**2.** L'automate reconnaît les mots contenant un $a$ en troisième position à partir de la fin.

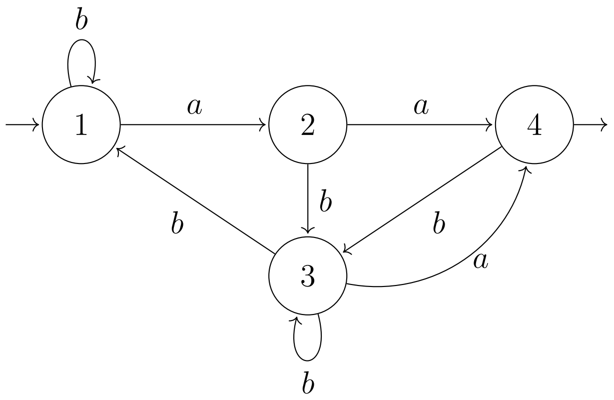
**3.**



**Exercice 6. Suppression des $\varepsilon$-transitions**
**1.**

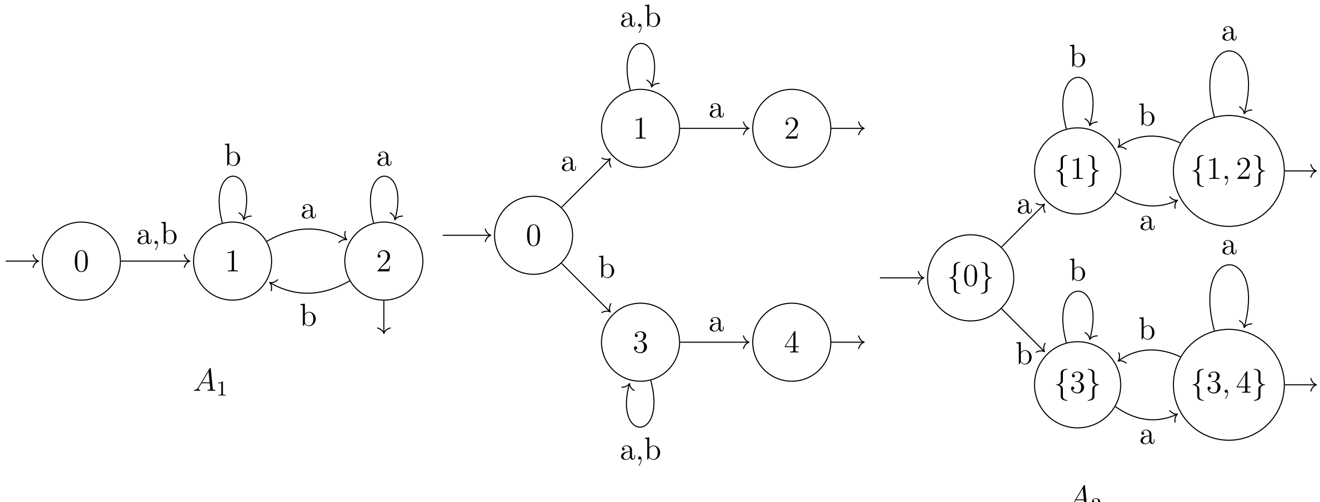



**2.**




**Exercice 7. Déterminisation de taille exponentielle**
**1.** L'automate déterminisé $A_D$ a pour ensemble d'états *les parties de* $Q$, soit $2^n$ états au plus.

**2.** Soit les automates $A_1$, $A_2$ et $A_3$ reconnaissant les mots de taille $\geq 2$ terminant par $a$.




$A_1$ $A_2$ $A_3$

**3.** Soit $A = (\Sigma, Q, I, F, \delta)$ défini par :

- $Q = \{0, 1, ..., n\}$
- $I = \{0\}$
- $F = \{n\}$
- $\delta : (i, c) \to \begin{cases} \{0\} & \text{si } i = 0 \text{ et } c = b \\ \{0,1\} & \text{si } i = 0 \text{ et } c = a \\ \text{non défini} & \text{si } i = n \\ \{i+1\} & \text{sinon} \end{cases}$

Montrons que $L(A) = L_n$. On raisonne par double inclusion.
Soit $w = w_1...w_k \in L_n$. Alors $w_{k-n} = a$.
Le chemin $0 \xrightarrow{w_1} 0 \xrightarrow{w_2} 0 \to ... \xrightarrow{w_{k-n-1}} 0 \xrightarrow{w_{k-n}=a} 1 \xrightarrow{w_{k-n+1}} 2 \to ... \xrightarrow{w_k} n$ est acceptant, donc $w \in L(A)$.

Réciproquement, soit $w \in L(A)$. Alors il existe un chemin acceptant $w$. Ce chemin termine en $F = \{n\}$ et commence en $I = \{0\}$, donc il comporte nécessairement l'arc $0 \xrightarrow{a} 1$, suivi de $n-1$ arcs menant à l'état $n$. Ainsi, la $n$ ième lettre en partant de la fin de $w$ est un $a$, donc $w \in L_n$.

**4.** $A_3$ correspond à l'automate 2. de l'exercice 5.

**5.** Soient deux mots distincts $u$ et $v$ de $\Sigma^n$. Supposons par l'absurde que $\delta^*(q_0, u) = \delta^*(q_0, v)$.

Puisque $u \neq v$, $\exists i \in \{1, ..., n\}$ tq $u_i \neq v_i$.
Supposons sans perte de généralité $v_i = a$.Alors $u_i = b$.
Le mot $v_1...v_i...v_n \underbrace{a...a}_{n-i} \in L_n$ donc est accepté par $D_n$.
Le $u_1...u_i...u_n \underbrace{a...a}_{n-i} \notin L_n$ donc n'est pas accepté par $D_n$.
Or ces deux mots ont le même chemin dans l'automate $D_n$, absurde.

**6.** Ainsi, l'automate $D_n$ contient au moins autant d'états qu'il n'y a de mots dans $\Sigma^n$, c'est-à-dire $2^n$.

**7.** Tout automate déterministe équivalent à $A_n$ comporte au moins $2^n$ états.

## III. Théorème de Kleene

**Exercice 10. Langage préfixe, suffixe et facteur**
Soit $L$ un langage régulier. D'après le théorème de Kleene, il existe un automate $A = (\Sigma, Q, I, F, \delta)$ tq $L(A) = L$. Sans perte de généralité, on suppose $A$ sans $\epsilon$-transition.
Soit $A_p = (\Sigma, Q_p, I_p, F_p, \delta_p)$ défini par :

- $Q_p = Q$
- $I_p = I$
- $F_p = \{q \in Q \mid q \text{ accessible et co-accessible}\}$
- $\delta_p = \delta$

Montrons que $L(A_p) = pref(L)$.

$\subset$ : Soit $w$ un mot reconnu par $L(A_p)$, et $q_0 \xrightarrow{w_1} q_1 \rightarrow ... \xrightarrow{w_n} q_n$ un chemin acceptant $w$ dans $A_p$.
$q_n \in F_p$ donc $q_n$ est un état co-accessible de $Q$. Ainsi, il existe $q_f \in F$ et une suite de transition $q_n \xrightarrow{a_1} q_1' \xrightarrow{a_2} ... \xrightarrow{a_k} q_f$.
Le chemin $q_0 \rightarrow ... \rightarrow q_n \rightarrow q_1' \rightarrow ... \rightarrow q_f$ accepte donc le mot $w.a_1...a_k$ dans $A$, donc $w$ est préfixe d'un mot de $L$.

$\supset$ : Soit $w = w_1...w_n$ un préfixe de $L$. Alors il existe un mot $u = u_1...u_m$ tel que $w.u \in L$.
Soit $q_0 \xrightarrow{w_1} q_1 \rightarrow ... \xrightarrow{w_n} q_n \xrightarrow{u_1} q_{n+1} \rightarrow ... \xrightarrow{u_m} q_{n+m}$ un chemin acceptant $w.u$ dans $A$.
L'état $q_n$ est accessible et co-accessible, il est donc final dans $A_p$. Ainsi, le chemin $(q_i)_{0 \leq i \leq n}$ reconnaît $w$ dans $A_p$.

**Ainsi, $A_p$ reconnaît $pref(L)$, donc le langage $pref(L)$ est régulier.**

Soit $A_s = (\Sigma, Q_s, I_s, F_s, \delta_s)$ défini par :

- $Q_s = Q$
- $I_s = \{q \in Q \mid q \text{ accessible et co-accessible}\}$
- $F_s = F$
- $\delta_p = \delta$

Montrons que $L(A_s) = suff(L)$.

*A faire en s'inspirant de la correction pour $A_p$*

Soit $A_f = (\Sigma, Q_f, I_f, F_f, \delta_f)$ défini par :

- $Q_f = Q$
- $I_f = \{q \in Q \mid q \text{ accessible et co-accessible}\}$
- $F_f = \{q \in Q \mid q \text{ accessible et co-accessible}\}$
- $\delta_f = \delta$

Montrons que $L(A_f) = fact(L)$.

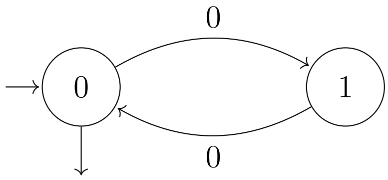
*A faire en s'inspirant de la correction pour $A_p$*

# IV. Automates et calculs

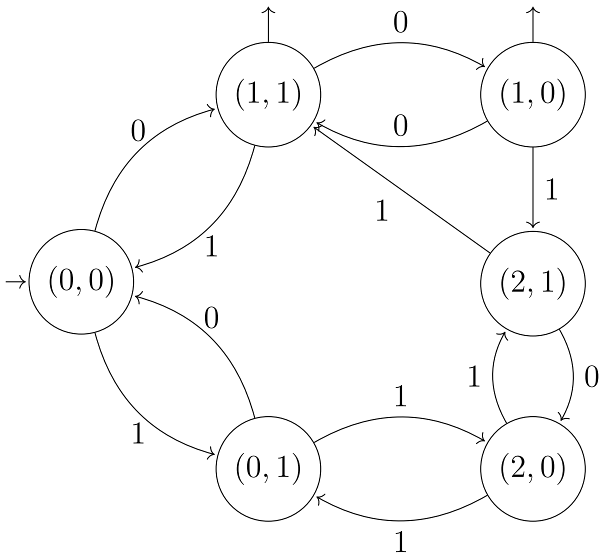
**Exercice 12. Calcul du modulo**

**1.** On montre par récurrence que $2^n \equiv \begin{cases} 1[3] & \text{si } n \text{ est pair} \\ 2[3] & \text{si } n \text{ est impair} \end{cases}$.

L'automate suivant reconnaît les mots sur l'alphabet $\{0\}$ de taille pair, c'est à dire les mots $n$ en unaire tels que $2^n \equiv 1[3]$.




**2.** Soit l'automate suivant :




On montre par récurrence sur $n$ que la lecture d'un mot $b_0 ... b_{n-1}$ aboutit dans l'état $(r, m)$ ssi $(b_{n-1} ... b_0 \equiv r[3]$ et $n \equiv m[2])$.