

TD 3
 alphabet
 Σ
 ensemble des états
 Q
 fonction de transition $Q \times \Sigma \rightarrow P_2(Q)$
 δ
 états finaux $F \subseteq Q$
 I
 états initiaux $I \subseteq Q$

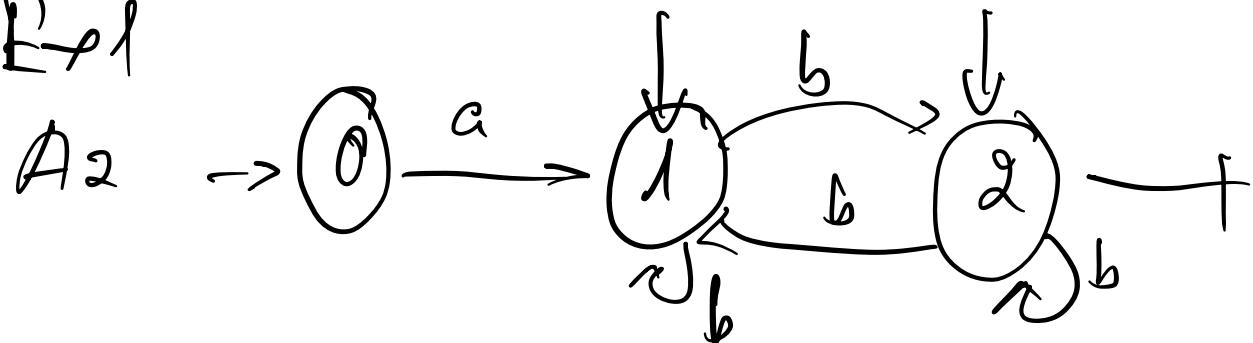
$$D(A) = (\Sigma, P(Q), I, \delta', F')$$

$$\delta' : P(Q) \times \Sigma \rightarrow P(Q)$$

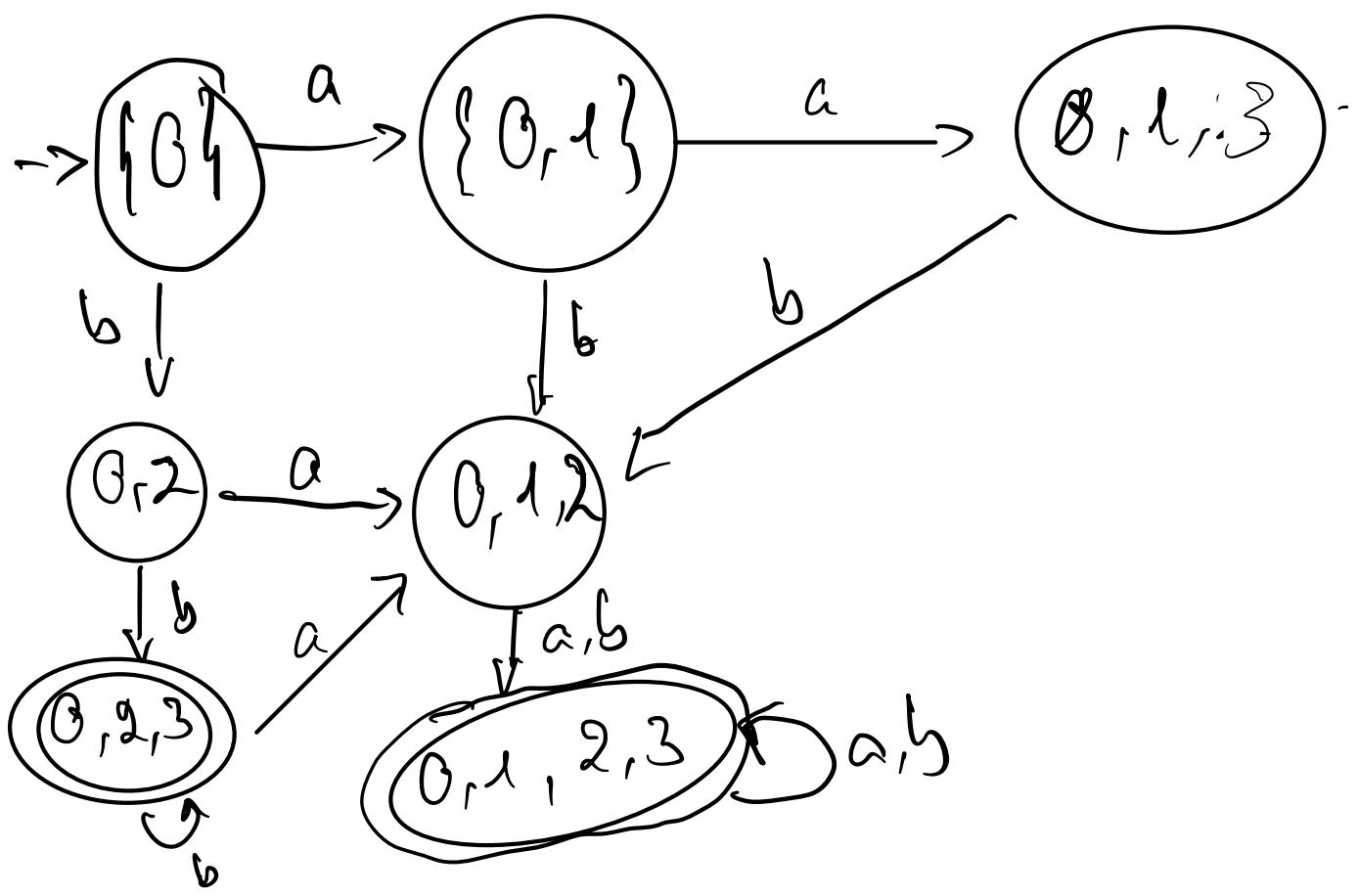
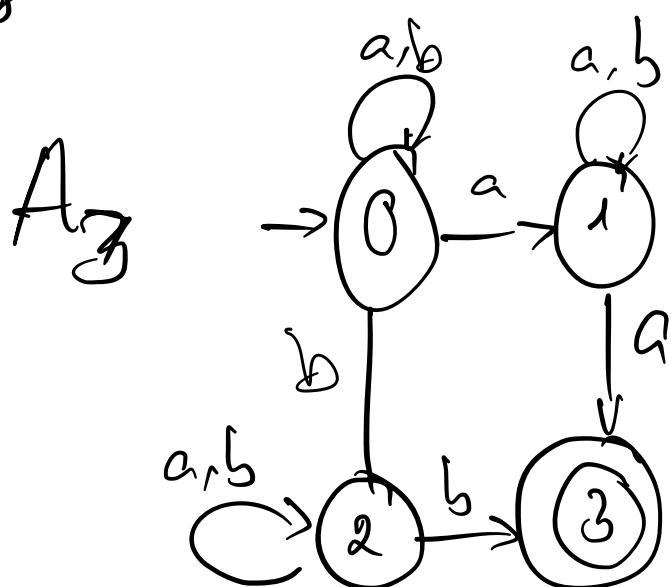
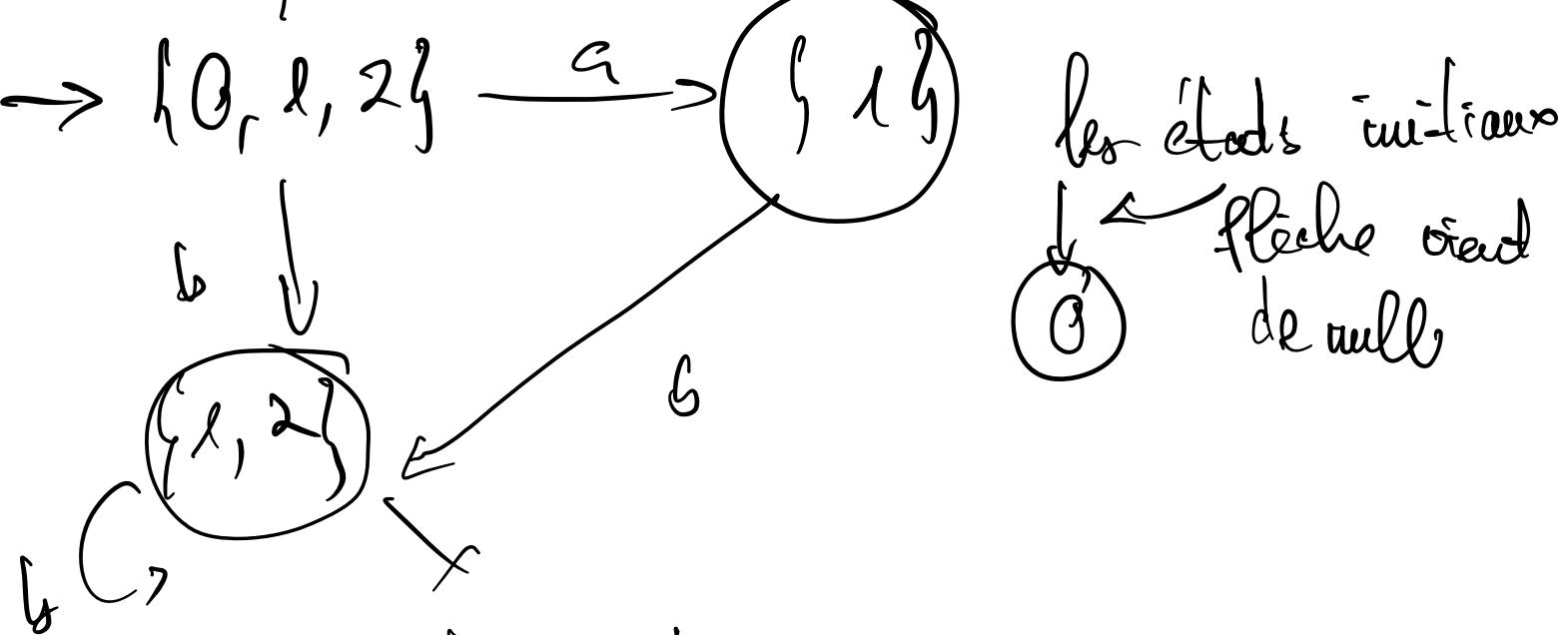
$$(x, v) \rightarrow \{q \in Q \mid \exists z \in X, q \in \delta(z, v)\}$$

$$F' = \{x \mid x \cap F \neq \emptyset\}$$

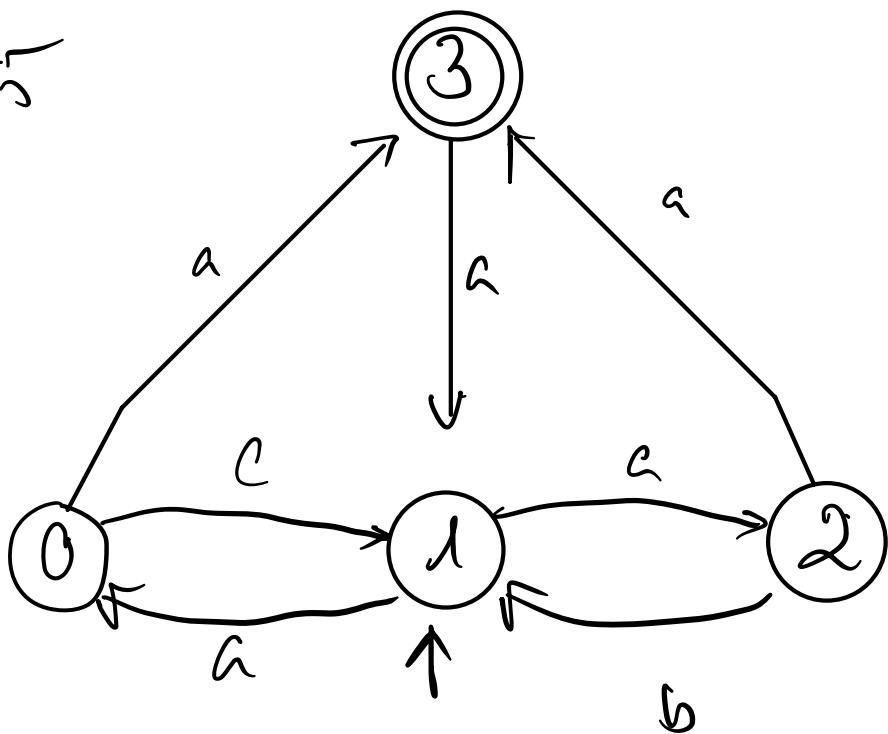
Ex 1



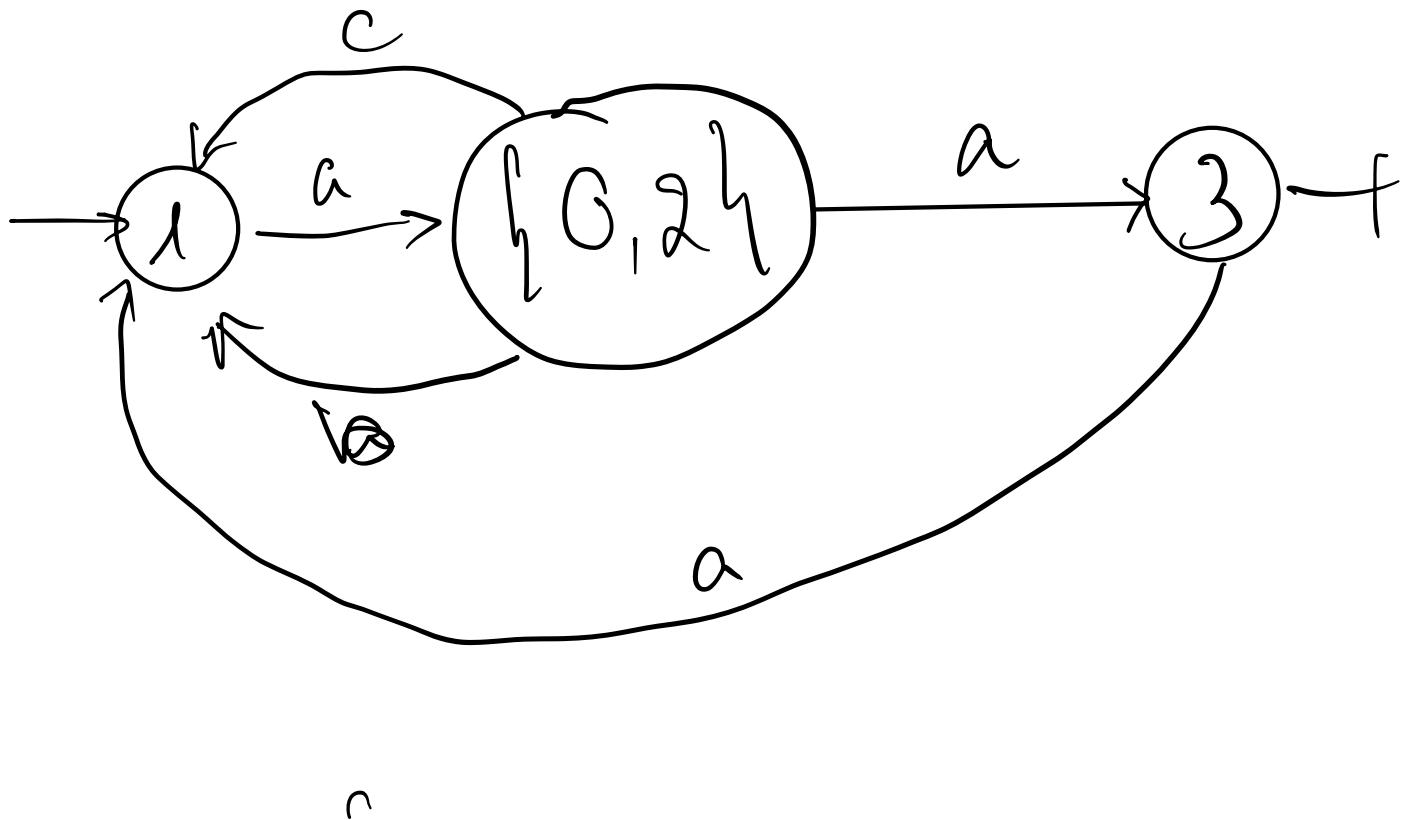
Il est pas déterministe car il a plusieurs choix
 et plusieurs états initiaux (a, b)



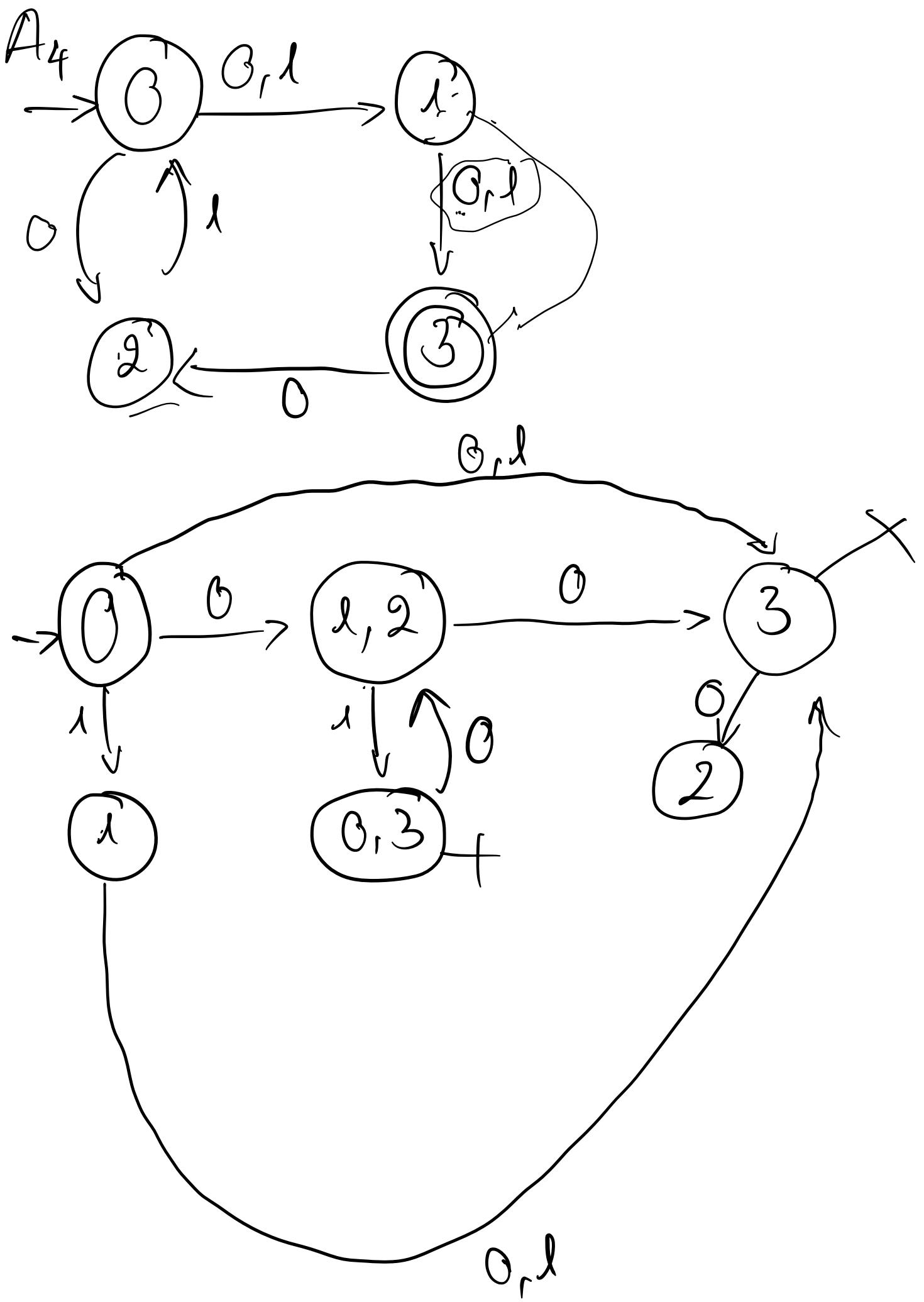
A_{57}



b



c

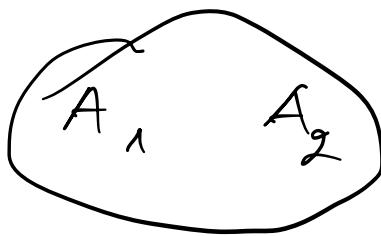


Ex 3x

Σ_1 est reconnu par A_1

Σ_2 est reconnu par A_2

$\Sigma_1 \cup \Sigma_2$



$$\Sigma_1 \cap \Sigma_2 = \overline{\Sigma_1} \cup \overline{\Sigma_2}$$

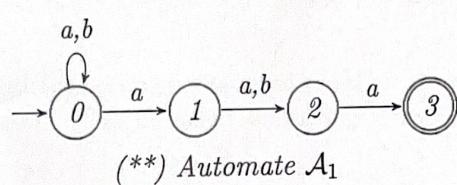
TD n°3

Déterminisation et Automate produit

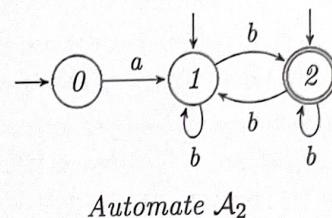
Les exercices ou questions marqués d'une ou plusieurs étoiles sont facultatifs ; le nombre d'étoiles donne une idée de la difficulté ou de la longueur des calculs.

Déterminisation

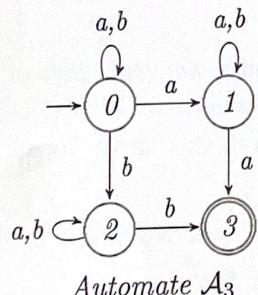
Exercice 1 Déterminiser les automates suivants :



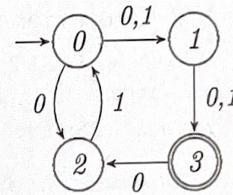
(**) Automate A_1



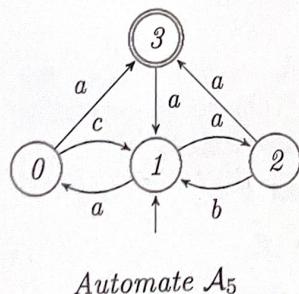
Automate A_2



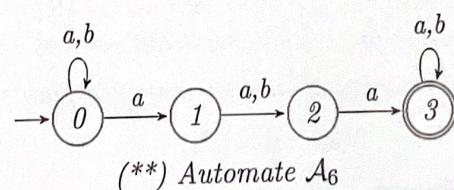
Automate A_3



(*) Automate A_4



Automate A_5



(**) Automate A_6

Exercice 2 (Digicode) On veut écrire directement sans passer par l'algorithme de déterminisation deux automates déterministes et complets qui reconnaissent l'entrée du « mot de passe » d'un digicode. Il n'y a que des chiffres possibles en entrée. Le code est 1165.

1. Construire un automate qui arrive dans un état acceptant pour toute séquence tapée qui finit par le bon code.
2. Construire un automate qui lit un code de taille 4, l'accepte si c'est le bon, refuse sinon, et permet ensuite de retenter sa chance.

Exercice 3 (Complémentaire, Intersection) Pour commencer, on veut déterminer si les langages constitués de tous les mots qui ne contiennent pas un motif donné sont reconnaissables. En particulier, on s'intéressera au langage $\mathcal{L} = \{u \in \{a, b, c\}^* \mid u \text{ ne contient pas le facteur } aba\}$.

1. Donner un automate \mathcal{A}_1 non déterministe pour $\mathcal{L}_1 = \{u \in \{a, b, c\}^* \mid u \text{ contient le facteur } aba\}$.
2. Donner un automate \mathcal{A}_2 déterministe pour \mathcal{L}_1 .
3. Donner un automate \mathcal{A}_3 déterministe pour \mathcal{L} à partir de \mathcal{A}_2 .
4. Peut-on généraliser ce raisonnement et conclure que le complémentaire d'un langage reconnaissable est toujours reconnaissable ?
5. En utilisant ce résultat et le fait que l'union de deux langages reconnaissables est reconnaissable (pourquoi ?), comment peut-on conclure que l'intersection de deux langages reconnaissables est toujours reconnaissable ?

Automate produit

Exercice 4 (Produit d'automates) Pour deux automates déterministes et complets $\mathcal{A} = (\Sigma, Q, q_0, F, \delta)$ et $\mathcal{A}' = (\Sigma, Q', q'_0, F', \delta')$, on définit l'automate déterministe $\mathcal{A}'' = (\Sigma, Q \times Q', (q_0, q'_0), F'', \delta'')$, dit automate produit, avec $\delta''((q, q'), a) = (\delta(q, a), \delta'(q', a))$, et l'ensemble F'' des états acceptants dépend de ce que l'on veut calculer.

1. Dessiner un automate \mathcal{A}_1 déterministe et complet qui reconnaît le langage \mathcal{L}_1 des mots sur $\{a, b\}$ qui commencent par a . (3 états devraient suffire.)
2. Dessiner un automate \mathcal{A}_2 déterministe et complet qui reconnaît le langage \mathcal{L}_2 des mots sur $\{a, b\}$ qui finissent par b . (2 états devraient suffire.)
3. Dessiner le produit des deux automates \mathcal{A}_1 et \mathcal{A}_2 , sans s'occuper des états acceptants. Éliminer le(s) état(s) non accessible(s) éventuel(s). (On dit qu'un état est accessible si on peut atteindre cet état en lisant un mot depuis un état initial.)
4. Comment choisir les états acceptants pour obtenir :

$$(a) \quad \mathcal{L}_1 \cap \mathcal{L}_2, \quad (b) \quad \mathcal{L}_1 \cup \mathcal{L}_2, \quad (c) \quad \overline{\mathcal{L}_1 \cap \mathcal{L}_2}.$$

5. **Bonus :** Pour lesquels de ces calculs était-il possible d'utiliser des automates déterministes non complets pour \mathcal{A}_1 et \mathcal{A}_2 ?