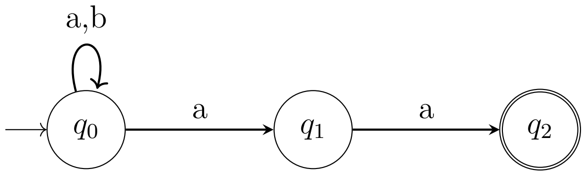
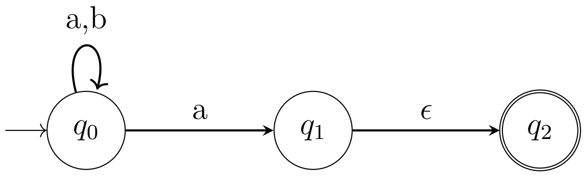
Définition 1. Un **Automate fini** est **non déterministe** si :

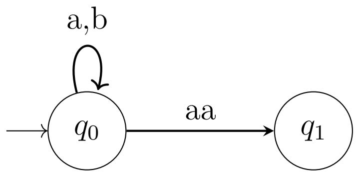
1. Il existe plusieurs transitions correspondant à une même lettre à partir d'un état :



2. Il existe des transitions sur le mot vide :



3. Il existe des transitions sur des mots de longueur supérieure à 1:



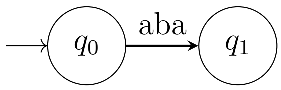
Remarques 1. Dans le cas où aucune de ces conditions n'est vérifiée, on dit que l'automate fini est **déterministe**.

Définition 2. Deux automates sont **équivalents** s'ils acceptent le même langage.

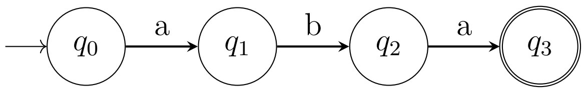
Théorème 1. *Pour tout automate fini non déterministe, on peut construire un automate fini déterministe équivalent. Il est dit* **déterministe complet** *si, de plus, une transition est toujours possible pour chaque lettre.*

Méthode pour rendre un automate déterministe

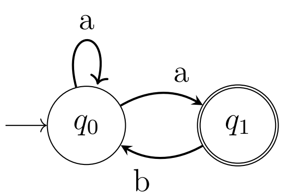
1. On élimine les transitions sur plusieurs lettres :



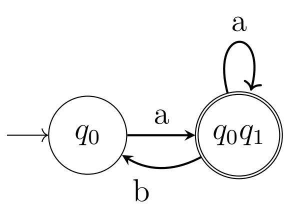
devient



2. On élimine les transitions multiples pour une même lettre :

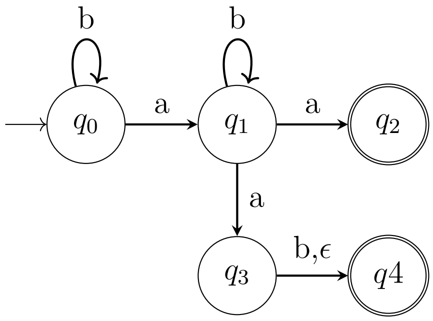


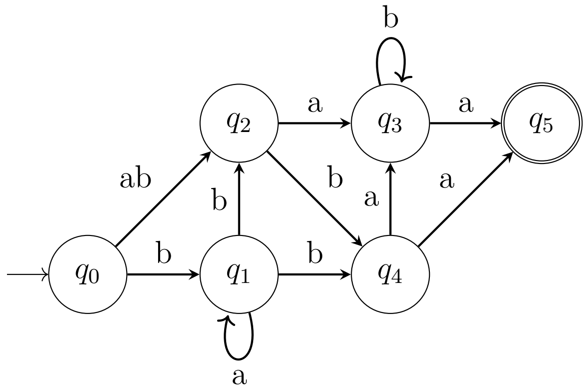
devient

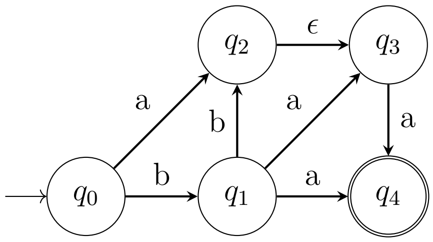


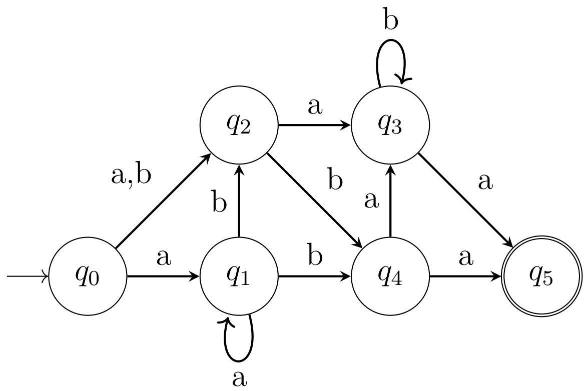
3. Pour finir on élimine les transitions multiples sur le mot vide.

Exercice 1. Rendre déterministes les automates suivants :









Exercice 2. Donner un automate déterministe pour chacun des langages réguliers suivants :

1. $L_1\ :\ (a|b)b^+|c^*$.
2. $L_2\ :\ (ab)^*c|c^*ac^*$.
3. $L_3\ :\ ((ab)^*|c)^*$.
4. $L_4\ :\ (a|b)^*(c|\epsilon)^2$.

Exercice 3. Soit $\Sigma = \{a,b\}$ et L_1 l'ensemble des mots de Σ^* contenant aba.

1. Donner une expression régulière et un automate fini **non déterministe à deux états** reconnaissant le langage L_1.
2. Rendre déterministe cet automate.