Exercice 1 : Shuffle On fixe l'alphabet fini $\Sigma = \{a, b\}$. Étant donné deux mots $w_1 = a_1 a_2 \dots a_n$ et $w_2 = b_1 b_2 \dots b_n$ (avec pour tout i $a_i, b_i \in \Sigma$), on définit le shuffle de w_1 et w_2 par $w_1 \bowtie w_2 = a_1 b_1 a_2 b_2 \dots a_n b_n$.

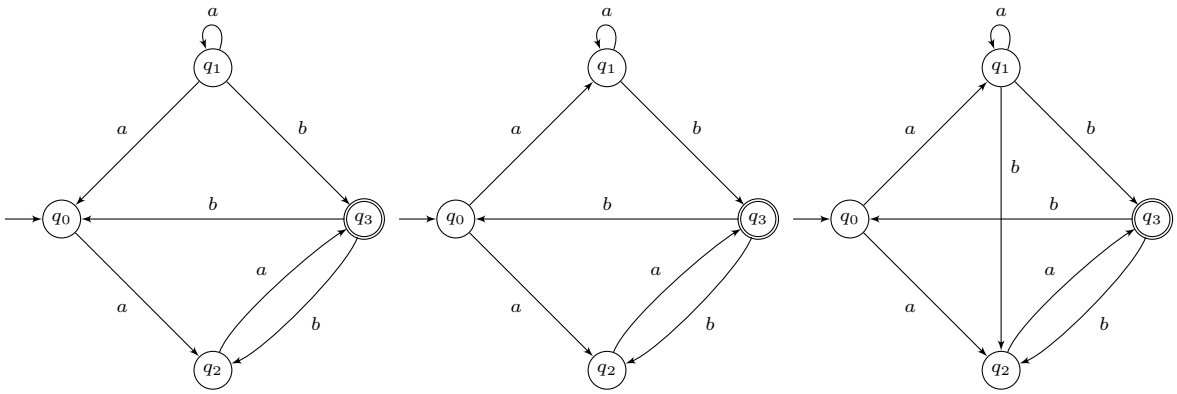
— Est-ce que $\bowtie$ est un opérateur commutatif ?

— Calculez $abbab \bowtie abaab$.

— On étend $\bowtie$ aux langages : $L_1 \bowtie L_2 = \{w_1 \bowtie w_2 | w_1 \in L_1, w_2 \in L_2, |w_1| = |w_2|\}$.

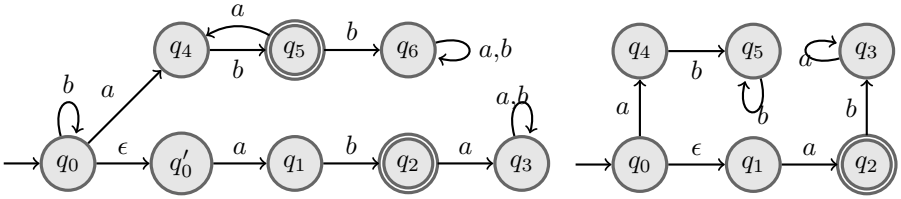
Décrivez une méthode pour construire l'automate reconnaissant $L_1 \bowtie L_2$ à partir des automates reconnaissant L_1 et L_2.

Exercice 2 : Automates vers expression régulière Construisez les expressions régulières des automates suivants



Exercice 3 : Opacité d'un système (simplifié)

Dans cet exercice, nous nous intéressons à une propriété importante en sécurité des systèmes informatiques : l'opacité. Le contexte est le suivant. Nous supposons qu'un attaquant observe un système dont le comportement est modélisé par un automate. Les états finaux de l'automate représentent le "secret" : lors d'une exécution du système, l'attaquant ne doit pas être en mesure de savoir avec certitude que le système est dans un état secret. Si lors d'une exécution du système, l'attaquant est en mesure de déterminer que le système est dans un état secret, alors on dit que cette exécution révèle le secret. Un système est dit opaque s'il n'existe pas d'exécution qui révèle le secret.



1. Nous considérons le système représenté par l'automate de gauche ci-dessus. Lorsque l'attaquant observe a, b, ab, quels sont les états courants possibles du système ?
2. Dire si ce système est opaque.
3. Même questions avec le système modélisé par l'automate de droite ci-dessus.
4. Est-il possible à partir de l'automate modélisant le système, de construire un automate qui indique la connaissance de l'attaquant en fonction de son observation ? Construisez-le.