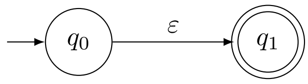
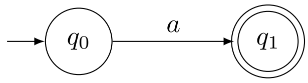
Exercice 4

1. On propose :
   - pour $\emptyset$:

   - pour ε :



   - pour $a \in \Sigma$:



2. - $L(A) \cup L(B)$: on crée une copie de A et de B et on rajoute un état initial q_0 et un état final q_f. On rajoute des ε-transition depuis q_0 vers les états initiaux de A et B et des ε-transitions des états finaux de A et B vers q_f.
   - $L(A)L(B)$: on crée une copie de A et de B et on rajoute une ε-transition entre l'état final de A et l'état initial de B. On considère que l'état initial de l'automate est celui de A et l'état final est celui de B.
   - $L(A)^*$: on crée une copie de A et on rajoute un état initial q_0 et un état final q_f. On rajoute des ε-transition depuis q_0 vers l'état initial de A et vers q_f, et des ε-transitions de l'état final de A vers son état initial et q_f.
3. Voir fichier `.ml`
4. Voir fichier `.ml`
5. Le problème est l'existence d'un cycle étiqueté par ε : le backtracking tournera en boucle sur les mêmes transitions.
6. On remarque que le problème se produit lorsqu'on calcule l'étoile d'une expression régulière dont l'interprétation contient le mot vide, car il existe alors un calcul de l'état initial à l'état final de l'automate de Thompson étiqueté par ε. On remarque que toute expression régulière e est équivalente à une expression de la forme :
   - e si $V(e) = \emptyset$;
   - $e'|\varepsilon$ avec $\mathcal{L}(e') = \mathcal{L}(e) \setminus \{\varepsilon\}$ sinon.

   On remarque notamment que si e et f sont des expressions régulières dont l'interprétation ne contient pas ε, alors :
   - $e(f|\varepsilon) \simeq ef|e$, $(e|\varepsilon)f \simeq ef|f$, $(e|\varepsilon)(f|\varepsilon) \simeq (ef|e|f)|\varepsilon$;
   - $e|(f|\varepsilon) \simeq (e|\varepsilon)|f \simeq (e|\varepsilon)|(f|\varepsilon) \simeq (e|f)|\varepsilon$;
   - $(e|\varepsilon)^* \simeq ee^*|\varepsilon$.

   On peut appliquer ce processus récursivement pour obtenir une expression régulière dont on est sûr qu'elle ne contient pas de cycle étiqueté par ε.

   Voir fichier `.ml` pour l'implémentation.