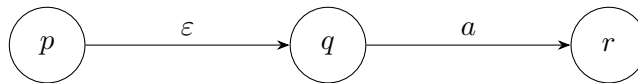


Élimination des ε -transitions dans un automate fini

Tout automate fini non déterministe avec ε -transitions peut être transformé en un automate fini non déterministe sans ε -transitions qui reconnaît exactement le même langage.

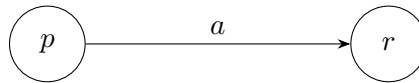
Supposons que l'on ait la situation suivante :



À partir de p , on peut :

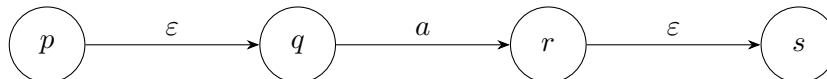
1. aller gratuitement en q ;
2. lire a ;
3. arriver en r .

On doit donc ajouter la possibilité :



Mais il faut faire attention : des ε -transitions peuvent également être empruntées **après** avoir lu la lettre.

Par exemple :



La lecture de a depuis p peut donc conduire à r , mais également à s .

En pratique, il suffit de choisir l'ordre dans lequel on élimine les ε -transitions : ici, on éliminera celle la plus à droite en premier afin que le phénomène ait disparu au moment où on élimine celle la plus à gauche.

La formule générale, elle, utilise une ε -fermeture **avant et après** la lecture du symbole.

Soit

$$A = (Q, I, F, \delta, \varphi)$$

un ε -AFN. On note $K(A)$ la fermeture par epsilon transition d'une partie A .

On veut construire un automate

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

sans ε -transition.

Pour $q \in Q$ et $a \in \Sigma$, on définit

$$\delta'(q, a) = K \left(\bigcup_{p \in K(q)} \delta(p, a) \right).$$

Cette formule se lit en trois étapes.

$$q \xrightarrow{\varepsilon^*} p \xrightarrow{a} r \xrightarrow{\varepsilon^*} s$$

Autrement dit :

1. on calcule tous les états accessibles depuis q par ε -transitions ;
2. depuis chacun de ces états, on effectue une transition étiquetée par a ;
3. on reprend ensuite toutes les ε -transitions possibles.

Le symbole ε^* signifie ici : « zéro, une ou plusieurs ε -transitions ».

La suppression des transitions ne concerne pas seulement la fonction de transition.

Elle peut aussi modifier l'ensemble des états finaux.

Considérons :



L'automate accepte le mot vide ε , puisque l'on peut atteindre l'état final q_1 sans lire de symbole.

Si l'on supprime simplement la transition ε , cette propriété serait perdue.

Il faut donc rendre q_0 final dans le nouvel automate.

Plus généralement,

$$q \in F' \iff K(q) \cap F \neq \emptyset.$$

Ainsi, un état devient final dès qu'il peut atteindre un ancien état final en utilisant uniquement des ε -transitions.

Algorithme général :

Pour éliminer toutes les ε -transitions d'un automate, on peut suivre la méthode suivante.

1. Calculer $K(q)$ pour chaque état q .
2. Pour chaque état q et chaque lettre $a \in \Sigma$, calculer

$$\delta'(q, a) = K \left(\bigcup_{p \in K(q)} \delta(p, a) \right).$$

3. Calculer les nouveaux états finaux :

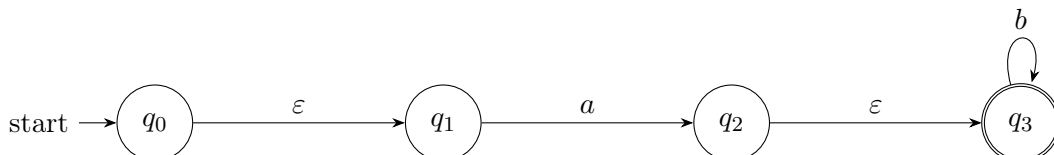
$$F' = \{q \in Q \mid K(q) \cap F \neq \emptyset\}.$$

4. Supprimer toutes les transitions étiquetées par ε .
5. Supprimer éventuellement les états devenus inutiles ou inaccessibles. Cette dernière étape est facultative.

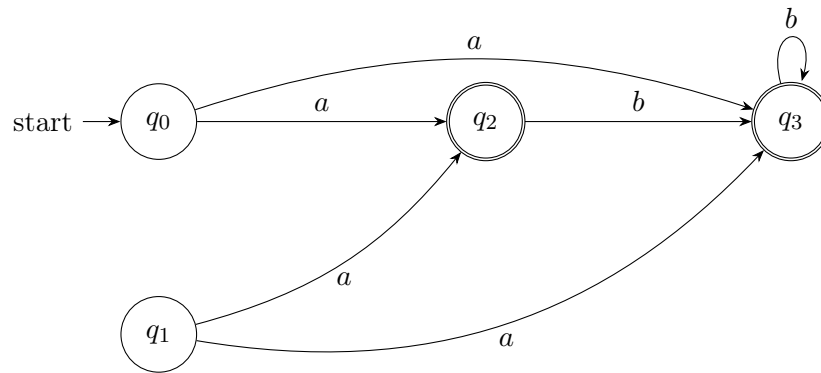
Exemple :

Considérons maintenant l'automate suivant sur l'alphabet

$$\Sigma = \{a, b\}.$$



Après suppression des ε -transitions, on obtient :



Si on veut éliminer les transitions à la main, alors il suffit de le faire dans un ordre bien choisi pour éviter de calculer trop d'épsilon-cloture. Ici, on peut reprendre l'exemple en éliminant successivement les transitions de la droite vers la gauche et on obtient alors :

