
TP2 - Convertir un automate en expression régulière

1 Convertir un automate en expression régulière : rappel

Voir les notes de cours pour la méthode générale. Un automate est un cas particulier de super automate, pour lequel les super transitions sont étiquetées par des expressions régulières qui sont juste des lettres.

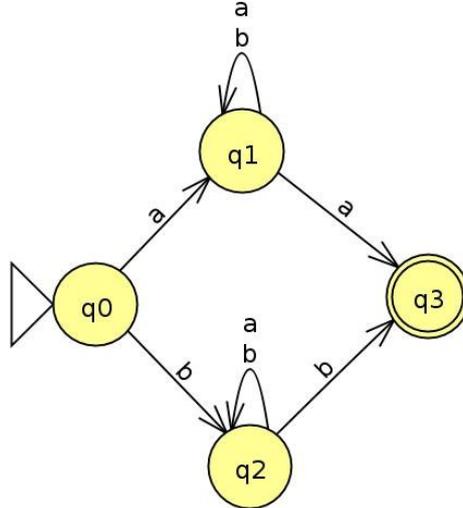
Les 4 modifications permettent de transformer un automate en un super automate à deux états, l'un initial et l'autre acceptant, pour lequel on sait facilement calculer une expression régulière équivalente.

Les cinq étapes du calcul (avec JFLAP, menu "Convert", choisir "Convert FA to RE", et suivre les instructions) :

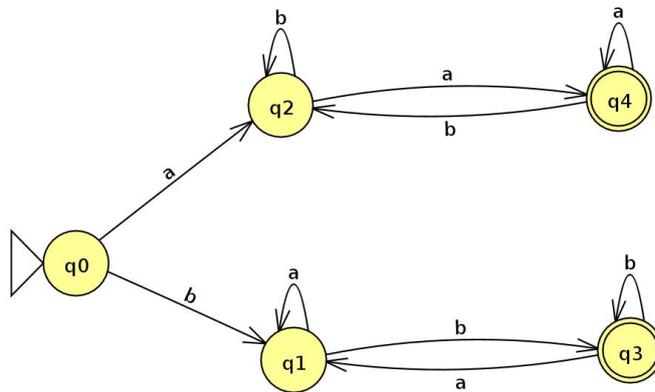
1. "Make single noninitial final state" : au plus un état acceptant distinct de l'état initial, avec transitions λ depuis les anciens états acceptants \rightarrow créer à la main un nouvel état qui sera le nouvel état final (2e bouton) et ajouter les transitions λ (3e bouton)
2. "Reform transitions" : au plus une transition entre 2 états avec $+$ \rightarrow utiliser le 4e bouton pour fusionner des transitions
3. "Reform transitions" : exactement une transition entre 2 états avec \emptyset (seulement nécessaire pour JFLAP) \rightarrow ajouter toutes les transitions à la main (2e bouton)
4. "Remove states" \rightarrow Cœur de la méthode : on supprime l'un après l'autre tous les états sauf l'état initial et l'état acceptant en tenant compte des détours (ajout des expressions régulières décrivant les détours) (5e bouton). Quand on clique sur une ligne du tableau affiché par JFLAP, il affiche en rouge les parties de l'automate concernées. **Faire cette étape état par état, et comprendre ce qui se passe !**
5. lire l'expression régulière sur le super-automate à deux états (affiché par JFLAP en haut de la fenêtre).

2 Exercices

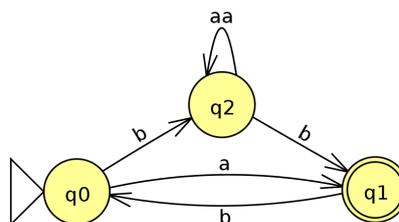
Exercice 1. Convertissez l'automate ci-dessous en expression régulière.



Exercice 2. Convertissez l'automate ci-dessous en expression régulière.



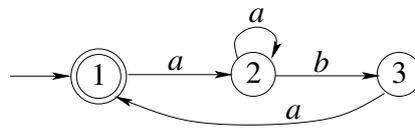
Exercice 3. On considère le « super-automate » ci-dessous. Supprimez l'état q_2 pour obtenir un super-automate équivalent à deux états, puis donnez l'expression régulière correspondant à cet automate.



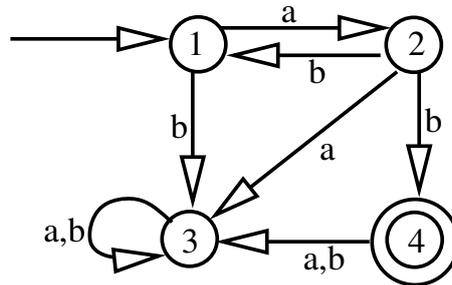
Exercice 4. Pour un langage L donné par plusieurs automates, obtient on une seule expression régulière ?

Exercice 5. Est-ce-que les super-automates (dont les transitions sont des expressions régulières) permettent de reconnaître des langages qu'on ne pourrait pas reconnaître avec aucun automate normal ? Ou au contraire décrivent-ils les mêmes langages que les automates normaux ?

Exercice 6. Déterminer une expression régulière représentant le langage reconnu par l'automate suivant.



Exercice 7. Déterminer une expression régulière représentant le langage reconnu par l'automate suivant.



Exercice 8. Soit le langage L sur l'alphabet $\Sigma = \{a, b\}$ constitué des mots d'au moins deux lettres, dont la dernière lettre apparaît au moins une autre fois dans le mot.

1. Trouver un automate A non déterministe à 4 états reconnaissant L .
2. Convertissez l'automate A en expression régulière.
3. Déterminer l'automate A . Vous appellerez A' l'automate déterministe obtenu.
4. Convertissez l'automate A' en expression régulière.

Exercice 9. Soit l'alphabet $\Sigma = \{a, b\}$.

1. Donner un automate déterministe qui reconnaît les mots ne contenant ni le facteur a^2 ni le facteur b^3 .
2. Donner un automate déterministe qui reconnaît les mots de taille multiple de 3 ne contenant pas b^3 .
3. Donner l'expression rationnelle pour chacun des deux points précédents.