Langages et Automates
Automates (seconde partie)

Engel Lefaucheux

Prépas des INP

Objectif du cours

- Découvrir encore d'autres opérations usuelles sur les automates
- Voir ce qu'on peut en faire, en déduire
- Découvrir un algorithme pour construire le langage d'un automate

Opérations sur les automates

Quels langages peut-on créer à partir d'automates existants ?

Soit L_1, L_2 deux langages reconnaissables par des automates déterministes. Les langages suivants sont-ils reconnaissables ?

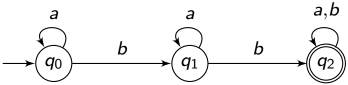
- $L_1 \cup L_2$? $\checkmark$
- $\overline{L_1}$? $\checkmark$
- $L_1 \cdot L_2$? $\checkmark$
- $(L_1)^*$? $\checkmark$
- $mirroir(L_1)$? $\checkmark$
- $L_1 \cap L_2$?

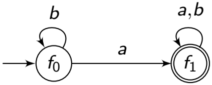
$L_1 \cap L_2$

$\mathcal{A}_1 = (Q_1, \Sigma, T_1, I_1, F_1)$ and $\mathcal{A}_2 = (Q_2, \Sigma, T_2, I_2, F_2)$
On construit $\mathcal{A}_3 = (Q_3, \Sigma, T_3, I_3, F_3)$

- $Q_3 = Q_1 \times Q_2$, $I_3 = I_1 \times I_2$ et $F_3 = F_1 \times F_2$
- $((q_1, q_2), a, (q'_1, q'_2)) \in T_3$ si et seulement si $(q_1, a, q'_1) \in T_1$ et $(q_2, a, q'_2) \in T_2$

$\mathcal{A}_1$ et $\mathcal{A}_2$ doivent être sans ε-transitions.



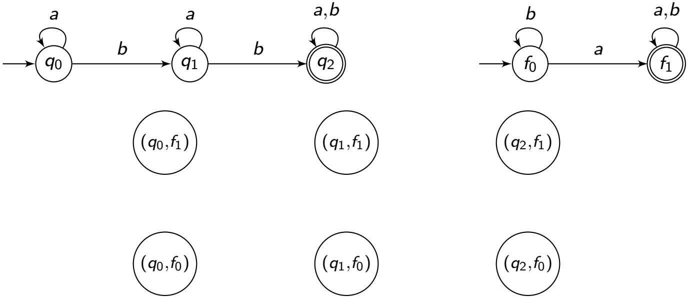


$L_1 \cap L_2$

$\mathcal{A}_1 = (Q_1, \Sigma, T_1, I_1, F_1)$ and $\mathcal{A}_2 = (Q_2, \Sigma, T_2, I_2, F_2)$
On construit $\mathcal{A}_3 = (Q_3, \Sigma, T_3, I_3, F_3)$

- $Q_3 = Q_1 \times Q_2$, $I_3 = I_1 \times I_2$ et $F_3 = F_1 \times F_2$
- $((q_1, q_2), a, (q_1', q_2')) \in T_3$ si et seulement si $(q_1, a, q_1') \in T_1$ et $(q_2, a, q_2') \in T_2$

$\mathcal{A}_1$ et $\mathcal{A}_2$ doivent être sans ε-transitions.

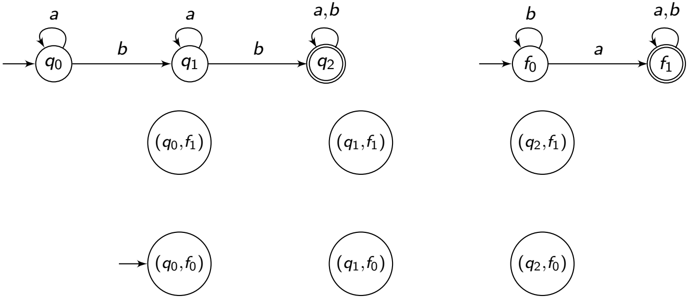


$L_1 \cap L_2$

$\mathcal{A}_1 = (Q_1, \Sigma, T_1, I_1, F_1)$ and $\mathcal{A}_2 = (Q_2, \Sigma, T_2, I_2, F_2)$
On construit $\mathcal{A}_3 = (Q_3, \Sigma, T_3, I_3, F_3)$

- $Q_3 = Q_1 \times Q_2$, $I_3 = I_1 \times I_2$ et $F_3 = F_1 \times F_2$
- $((q_1, q_2), a, (q_1', q_2')) \in T_3$ si et seulement si $(q_1, a, q_1') \in T_1$ et $(q_2, a, q_2') \in T_2$

$\mathcal{A}_1$ et $\mathcal{A}_2$ doivent être sans ε-transitions.

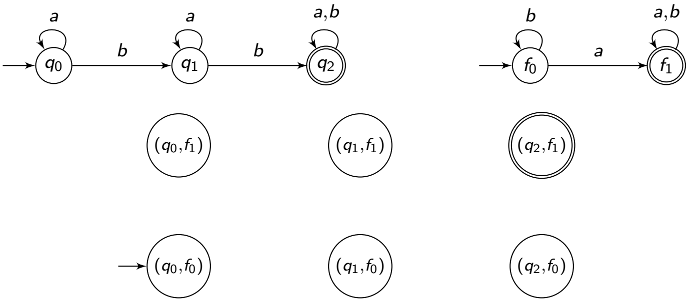


$L_1 \cap L_2$

$\mathcal{A}_1 = (Q_1, \Sigma, T_1, I_1, F_1)$ and $\mathcal{A}_2 = (Q_2, \Sigma, T_2, I_2, F_2)$
On construit $\mathcal{A}_3 = (Q_3, \Sigma, T_3, I_3, F_3)$

- $Q_3 = Q_1 \times Q_2$, $I_3 = I_1 \times I_2$ et $F_3 = F_1 \times F_2$
- $((q_1, q_2), a, (q_1', q_2')) \in T_3$ si et seulement si $(q_1, a, q_1') \in T_1$ et $(q_2, a, q_2') \in T_2$

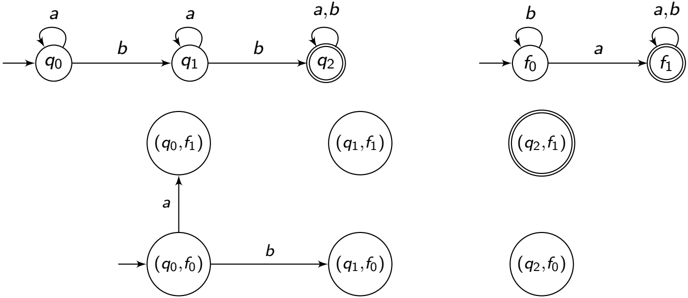
$\mathcal{A}_1$ et $\mathcal{A}_2$ doivent être sans ε-transitions.



$\mathcal{A}_1 = (Q_1, \Sigma, T_1, I_1, F_1)$ and $\mathcal{A}_2 = (Q_2, \Sigma, T_2, I_2, F_2)$
On construit $\mathcal{A}_3 = (Q_3, \Sigma, T_3, I_3, F_3)$

- $Q_3 = Q_1 \times Q_2$, $I_3 = I_1 \times I_2$ et $F_3 = F_1 \times F_2$
- $((q_1, q_2), a, (q_1', q_2')) \in T_3$ si et seulement si $(q_1, a, q_1') \in T_1$ et $(q_2, a, q_2') \in T_2$

$\mathcal{A}_1$ et $\mathcal{A}_2$ doivent être sans ε-transitions.

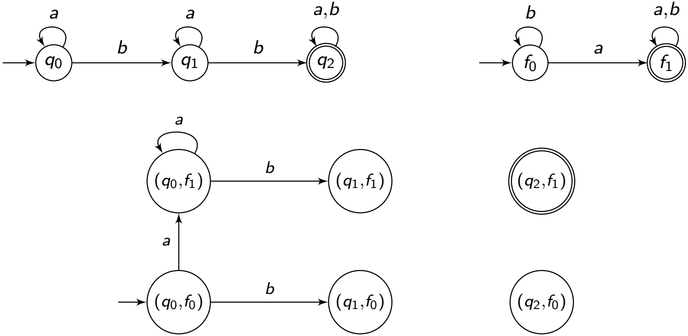


$L_1 \cap L_2$

$\mathcal{A}_1 = (Q_1, \Sigma, T_1, I_1, F_1)$ and $\mathcal{A}_2 = (Q_2, \Sigma, T_2, I_2, F_2)$
On construit $\mathcal{A}_3 = (Q_3, \Sigma, T_3, I_3, F_3)$

- $Q_3 = Q_1 \times Q_2$, $I_3 = I_1 \times I_2$ et $F_3 = F_1 \times F_2$
- $((q_1, q_2), a, (q_1', q_2')) \in T_3$ si et seulement si $(q_1, a, q_1') \in T_1$ et $(q_2, a, q_2') \in T_2$

$\mathcal{A}_1$ et $\mathcal{A}_2$ doivent être sans ε-transitions.

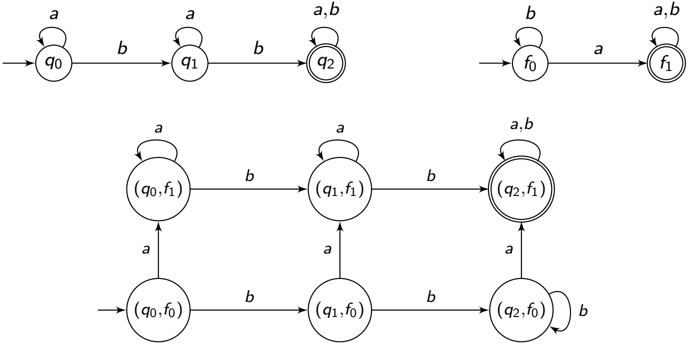


$L_1 \cap L_2$

$\mathcal{A}_1 = (Q_1, \Sigma, T_1, I_1, F_1)$ and $\mathcal{A}_2 = (Q_2, \Sigma, T_2, I_2, F_2)$
On construit $\mathcal{A}_3 = (Q_3, \Sigma, T_3, I_3, F_3)$

- $Q_3 = Q_1 \times Q_2$, $I_3 = I_1 \times I_2$ et $F_3 = F_1 \times F_2$
- $((q_1, q_2), a, (q_1', q_2')) \in T_3$ si et seulement si $(q_1, a, q_1') \in T_1$ et $(q_2, a, q_2') \in T_2$

$\mathcal{A}_1$ et $\mathcal{A}_2$ doivent être sans ε-transitions.



Exercices

Construire les automates pour les langages suivants

- $L_1 = \{w \mid |w| \text{ est pair}\}$
- L_2 : les mots ne contenant pas le facteur *aab*
- $L_2 \cap L_1$

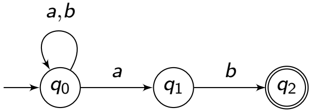
Ces automates sont-ils émondés ? Complets ? Sans ε-transition ?

Déterminisation d'un automate

Soit $\mathcal{A} = (Q, \Sigma, T, I, F)$ un automate

Construire $\mathcal{A}' = (Q', \Sigma, T', I', F')$ tel que

- $Q' = 2^Q$ l'ensemble contenant tous les ensembles d'états de Q
- $I' = \{I\}$
- $F' = \{C \in Q' \mid C \cap F \neq \emptyset\}$
- $(C, A, C') \in T'$ iff for all $q \in C'$, there exists $q \in C$ such that $(q, a, q') \in T$.

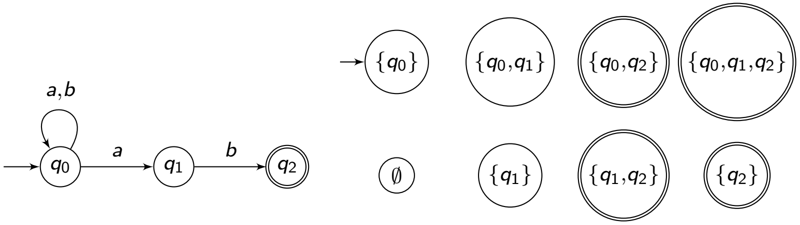


Déterminisation d'un automate

Soit $\mathcal{A} = (Q, \Sigma, T, I, F)$ un automate

Construire $\mathcal{A}' = (Q', \Sigma, T', I', F')$ tel que

- $Q' = 2^Q$ l'ensemble contenant tous les ensembles d'états de Q
- $I' = \{I\}$
- $F' = \{C \in Q' \mid C \cap F \neq \emptyset\}$
- $(C, A, C') \in T'$ iff for all $q \in C'$, there exists $q \in C$ such that $(q, a, q') \in T$.

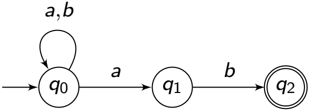


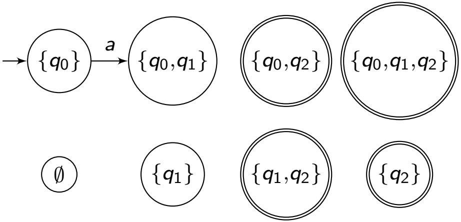
Déterminisation d'un automate

Soit $\mathcal{A} = (Q, \Sigma, T, I, F)$ un automate

Construire $\mathcal{A}' = (Q', \Sigma, T', I', F')$ tel que

- $Q' = 2^Q$ l'ensemble contenant tous les ensembles d'états de Q
- $I' = \{I\}$
- $F' = \{C \in Q' \mid C \cap F \neq \emptyset\}$
- $(C, A, C') \in T'$ iff for all $q \in C'$, there exists $q \in C$ such that $(q, a, q') \in T$.



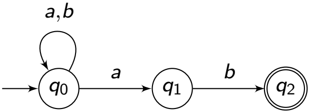


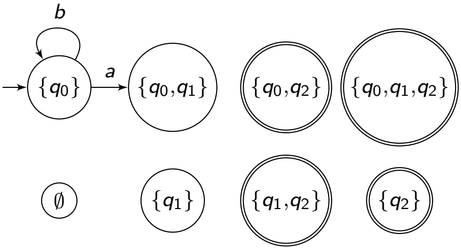
Déterminisation d'un automate

Soit $\mathcal{A} = (Q, \Sigma, T, I, F)$ un automate

Construire $\mathcal{A}' = (Q', \Sigma, T', I', F')$ tel que

- $Q' = 2^Q$ l'ensemble contenant tous les ensembles d'états de Q
- $I' = \{I\}$
- $F' = \{C \in Q' \mid C \cap F \neq \emptyset\}$
- $(C, A, C') \in T'$ iff for all $q \in C'$, there exists $q \in C$ such that $(q, a, q') \in T$.



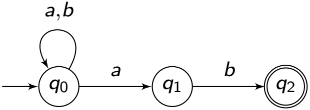


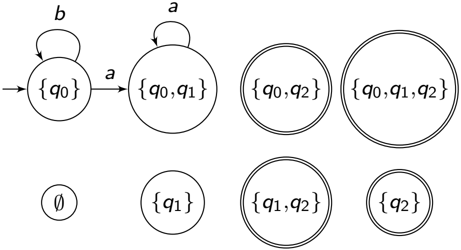
Déterminisation d'un automate

Soit $\mathcal{A} = (Q, \Sigma, T, I, F)$ un automate

Construire $\mathcal{A}' = (Q', \Sigma, T', I', F')$ tel que

- $Q' = 2^Q$ l'ensemble contenant tous les ensembles d'états de Q
- $I' = \{I\}$
- $F' = \{C \in Q' \mid C \cap F \neq \emptyset\}$
- $(C, A, C') \in T'$ iff for all $q \in C'$, there exists $q \in C$ such that $(q, a, q') \in T$.



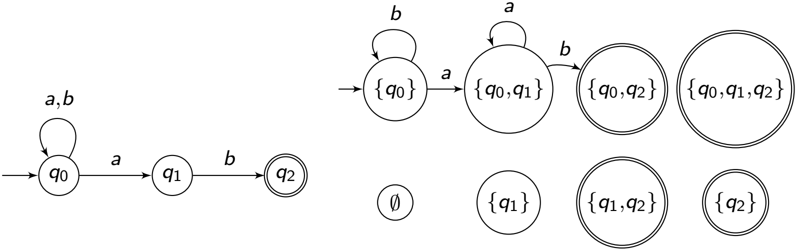


Déterminisation d'un automate

Soit $\mathcal{A} = (Q, \Sigma, T, I, F)$ un automate

Construire $\mathcal{A}' = (Q', \Sigma, T', I', F')$ tel que

- $Q' = 2^Q$ l'ensemble contenant tous les ensembles d'états de Q
- $I' = \{I\}$
- $F' = \{C \in Q' \mid C \cap F \neq \emptyset\}$
- $(C, A, C') \in T'$ iff for all $q \in C'$, there exists $q \in C$ such that $(q, a, q') \in T$.

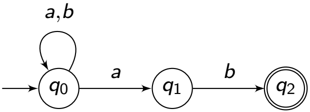


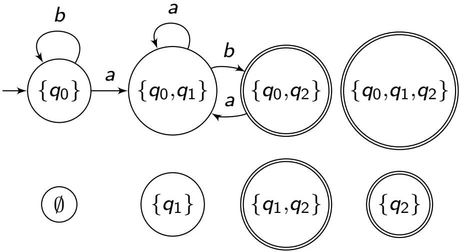
Déterminisation d'un automate

Soit $\mathcal{A} = (Q, \Sigma, T, I, F)$ un automate

Construire $\mathcal{A}' = (Q', \Sigma, T', I', F')$ tel que

- $Q' = 2^Q$ l'ensemble contenant tous les ensembles d'états de Q
- $I' = \{I\}$
- $F' = \{C \in Q' \mid C \cap F \neq \emptyset\}$
- $(C, A, C') \in T'$ iff for all $q \in C'$, there exists $q \in C$ such that $(q, a, q') \in T$.



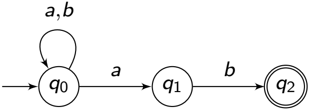


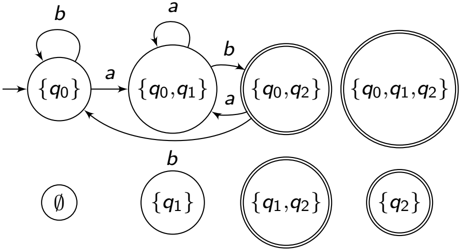
Déterminisation d'un automate

Soit $\mathcal{A} = (Q, \Sigma, T, I, F)$ un automate

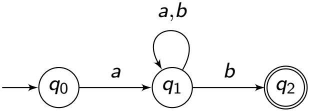
Construire $\mathcal{A}' = (Q', \Sigma, T', I', F')$ tel que

- $Q' = 2^Q$ l'ensemble contenant tous les ensembles d'états de Q
- $I' = \{I\}$
- $F' = \{C \in Q' \mid C \cap F \neq \emptyset\}$
- $(C, A, C') \in T'$ iff for all $q \in C'$, there exists $q \in C$ such that $(q, a, q') \in T$.

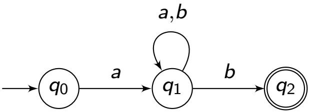


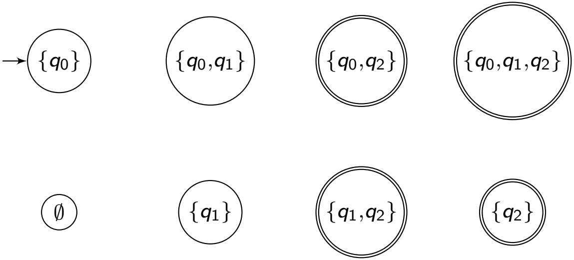


Second exemple

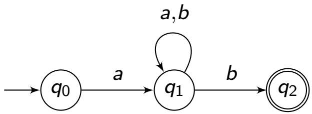


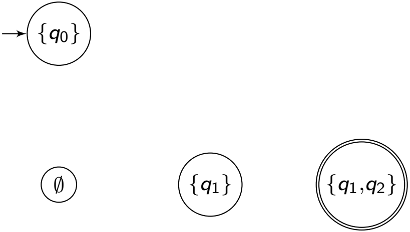
Second exemple



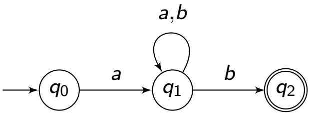


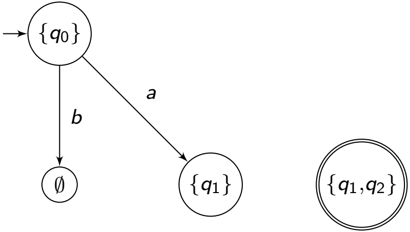
Second exemple



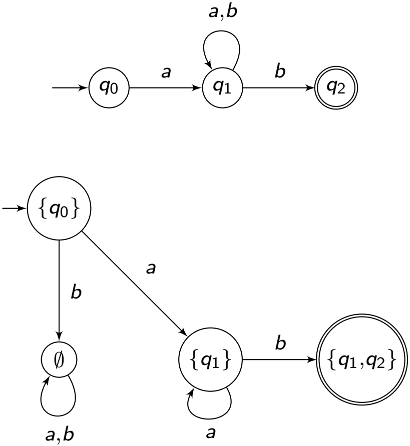


Second exemple

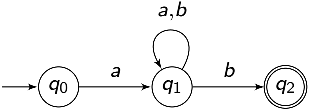


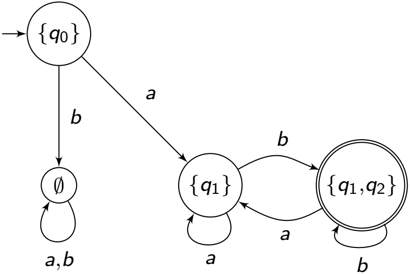


Second exemple



Second exemple





Preuve : les deux automates ont le même langage

On montre par récurrence sur $n = |w|$ que pour tout mot $w \in \Sigma^*$, il existe dans $\mathcal{A}$ un chemin étiqueté par w menant à un état q si et seulement s'il existe dans $\mathcal{A}'$ un chemin étiqueté par w menant à un état P contenant q.

- si $n = 0$, $w = \varepsilon$ et $q \in I$
- supposons que l'hypothèse est correcte pour $n \in \mathbb{N}$. Soit $w = a_1 \dots a_{n+1}$

Considérons un chemin $q_0 \xrightarrow{a_1} q_1 \xrightarrow{a_2} \dots \xrightarrow{a_{n+1}} q_{n+1}$
Par hypothèse, il existe C tel que $q_n \in C$ et C est l'unique état de $\mathcal{A}'$ atteint en lisant $a_1 \dots a_n$
Par définition de T', il existe C' tel que $q_{n+1} \in C'$ et $(C, a_{n+1}, C') \in T'$.
Réciproquement ...

Sachant que $F' = \{C \in Q' \mid C \cap F \neq \emptyset\}$, on atteint dans $\mathcal{A}'$ avec le mot w un ensemble de F' ssi il existe un chemin pour w dans $\mathcal{A}$ terminant en F

Une conséquence de toute ces opérations

On a vu que, si $\mathcal{A}_1$ et $\mathcal{A}_2$ produisent les langages L_1 et L_2, on peut construire l'automate construisant

- $L_1 \cdot L_2$
- L_1^*
- $L_1 \cup L_2$

Que peut-on en déduire en terme de langage ?

Exercices

Construire un automate déterministe pour les langages suivants

- $L_1 = (a + b + c)^* a (a + b + c)^*$
- L_2 : les mots de longueur paire ou multiple de 3

Quel langage pour les automates ?

Par rapport aux grammaires ? Aux expressions régulières ?

Quel langage pour les automates ?

Par rapport aux grammaires ? Aux expressions régulières ?

Theorem (Théorème de Kleene)

Les langages reconaissables sont exactement les langages réguliers.

Quel langage pour les automates ?

Par rapport aux grammaires ? Aux expressions régulières ?

Theorem (Théorème de Kleene)

Les langages reconaissables sont exactement les langages réguliers.

Les automates sont un formalisme bien plus simple à manipuler.

Quel langage pour les automates ?

Par rapport aux grammaires ? Aux expressions régulières ?

Theorem (Théorème de Kleene)

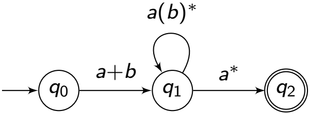
Les langages reconaissables sont exactement les langages réguliers.

Les automates sont un formalisme bien plus simple à manipuler.

Voyons comment passer d'un automate à une expression régulière.

Les automates généralisés

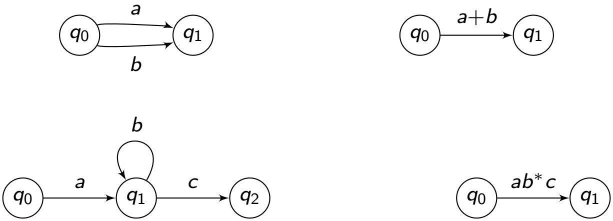
Principe : autoriser les expressions régulières sur les transitions



Produit le langage $(a+b)(a(b)^*)^*a^*$

Simplification d'automates par les expressions régulières

Principe : retirer les états et les transitions un par un

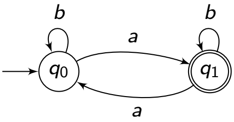


Algorithme :

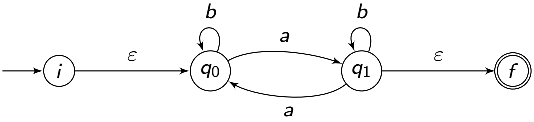
- Ajouter un état initial i et un état final f remplaçant les états initiaux et finaux précédents et relié avec des ε transitions.
- Appliquer les transformations ci-dessus pour retirer les états un par un, sauf i et f et la transition entre i et f

Exemple

Considérons l'automate

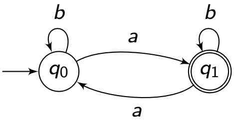


On ajoute d'abord le nouvel état initial et le nouvel état final

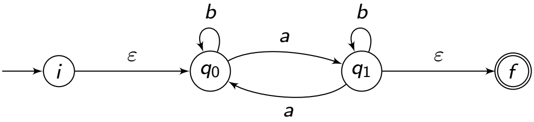


Exemple

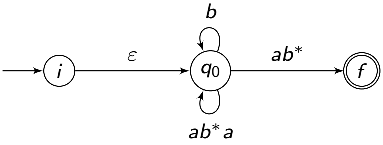
Considérons l'automate



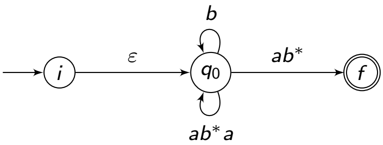
On ajoute d'abord le nouvel état initial et le nouvel état final



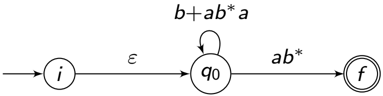
Retirons q_1



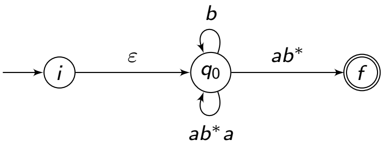
Exemple (suite)



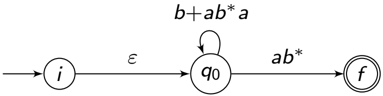
Fusion des boucles sur q_0



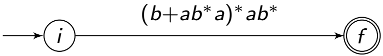
Exemple (suite)



Fusion des boucles sur q_0



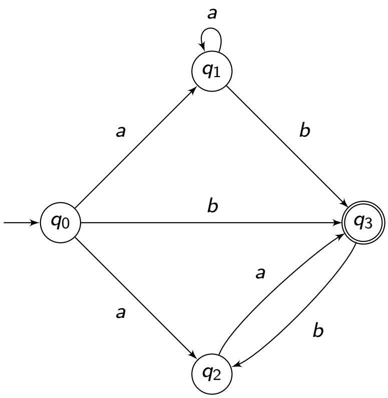
Retirons q_0



Le langage est donc $(b + ab^* a)^* ab^*$

Exercice

Quel langage pour l'automate



Résumé

- On peut construire un automate à partir d'une expression régulière
  - Union d'automate
  - Concaténation
  - Étoile
  - Mirroir
  - Complément
- On peut compléter, déterminiser, émonder, retirer les ε d'un automate
- On peut construire l'expression régulière à partir d'un automate