Théorie des Langages 1

Cours 4 : minimisation

L. Rieg

Grenoble INP - Ensimag, 1re année

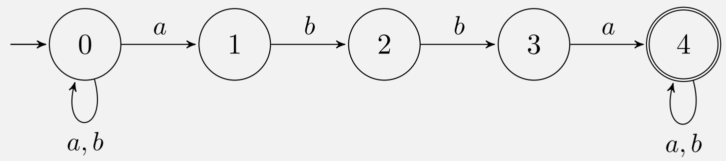
Année 2022-2023

De la déterminisation à la minimisation : exercice

On considère le langage

$$L = \{w \in V^* \mid w \text{ contient le motif } abba\}$$

Voici un automate non-déterministe qui le reconnaît :



Construire un AFD (complet) équivalent.

Solution

Autre illustration de la minimisation

Construire un AFD complet pour $L = (\{0\}^* \{1\}^*)^* \{0, 01\}^* \{0\}^+$.

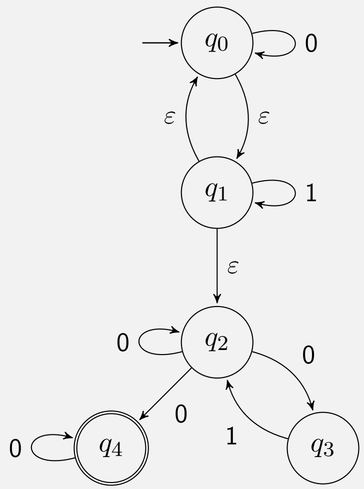
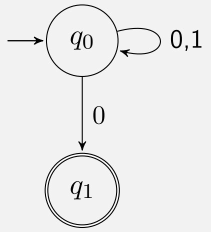


Illustration (suite)

Exercice : On peut montrer que $L = \{0,1\}^* \{0\}$.



Les deux automates déterministes construits sont équivalents.

Définition (Minimalité)

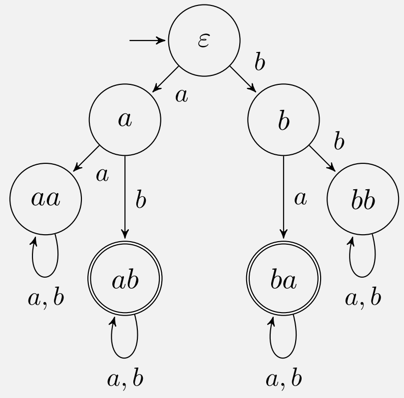
Un AFD complet A est minimal si tout AFD complet équivalent à A a au moins autant d'états que A.
Cet automate minimal est unique au renommage des états près.

Question : comment construire de façon systématique un AFD minimal ?

Principe de construction

On peut facilement « fusionner » certains états.

Exemple : $L = \{ab, ba\} \{a,b\}^*$



Généralisation

Définition (Équivalence de Nerode)

Soit $A = \langle Q, V, \delta, \{q_0\}, F\rangle$ un AFD complet.
Deux états $p, q \in Q$ sont équivalents dans A si et seulement si

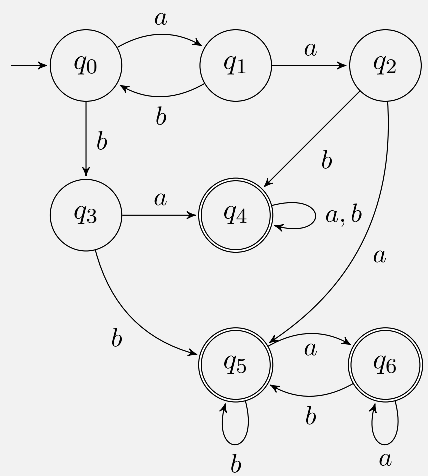
$$\forall w \in V^*,$$

On note alors $p \equiv_A q$, ou simplement $p \equiv q$.

Proposition

Posons $A_p \stackrel{def}{=} \langle Q, V, \delta, \{p\}, F\rangle$ et $A_q \stackrel{def}{=} \langle Q, V, \delta, \{q\}, F\rangle$.
On a alors : $p \equiv_A q$ si et seulement si A_p et A_q sont équivalents.

Exemple



Définition de l'automate minimal

Proposition

1. *$\equiv_A$ est une relation d'équivalence.*
   On note $[p]$ (ou $[p]_A$) la classe d'équivalence de p.
2. *Si $p \equiv_A q$, alors $\forall w \in V^*$, $\delta^*(p,w) \equiv_A \delta^*(q,w)$.*
   Si $[p]_A = [q]_A$ alors $\forall w \in V^*$, $[\delta^*(p,w)]_A = [\delta^*(q,w)]_A$.

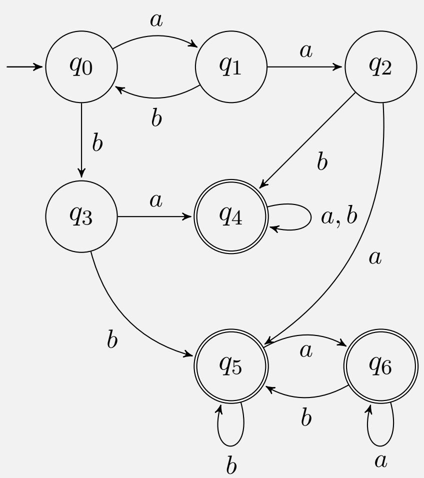
Preuve : exercice.

Définition

Soit $A = \langle Q, V, \delta, \{q_0\}, F \rangle$ un AFD complet et initialement connecté.
On définit $\mu(A) = \langle Q_\mu, V, \delta_\mu, \{[q_0]\}, F_\mu \rangle$, où :

- Q_μ est l'ensemble des classes d'équivalence des états de Q ;
- F_μ est l'ensemble des classes d'équivalence des états de F ;
- $\forall [p] \in Q_\mu$, $\forall a \in V$, $\delta_\mu([p], a) = [\delta(p,a)]$.

Exemple



Construction d'un automate minimal

Soit un AFD complet $A = \langle Q, V, \delta, \{q_0\}, F \rangle$ qu'on souhaite minimiser.

1. Supprimer les états inaccessibles de A
2. Déterminer efficacement la relation $\equiv$
   Par approximations successives (cf. $\mathrm{Acc}_\varepsilon(p)$)
3. Construire l'automate minimal : $\langle Q_\mu, V, \delta_\mu, \{[q_0]\}, F_\mu \rangle$

Définition

Pour $k \geq 0$, on définit la relation $\equiv_k$ sur Q par : $p \equiv_k q$ si et seulement si

p et q sont équivalents pour tous les mots de longueur au plus k.

Formellement :

$$\forall w \in V^*, \text{ si } |w| \leq k, \text{ alors } (\delta^*(p,w) \in F \Leftrightarrow \delta^*(q,w) \in F)$$

Si $p \equiv_k q$, alors les automates A_p et A_q reconnaissent exactement les mêmes mots de longueur au plus k.

Calcul de $\equiv$

Proposition

On a les propriétés suivantes :

$$\equiv_{k+1} \quad \subseteq \quad \equiv_k \qquad\qquad \equiv \quad = \quad \bigcap_{k \geq 0} \equiv_k$$

Proposition (Stabilisation de la suite $\equiv_k$)

Si A est un AFD à n états, alors
il existe $k \leq n$ tel que les relations $\equiv_k$, $\equiv_{k+1}$ et $\equiv$ sont identiques.

Donc, si on sait calculer les relations $\equiv_k$ efficacement,
on saura en déduire la relation $\equiv$.

Calcul de $\equiv$ (suite)

Proposition

On a les propriétés suivantes :

1. $p \equiv_0 q$ *si et seulement si* $p \in F \Leftrightarrow q \in F$
2. $\forall k \geq 0,\ p \equiv_{k+1} q$ *si et seulement si*

$$p \equiv_k q \text{ et } \forall a \in V,\ \delta(p,a) \equiv_k \delta(q,a)$$

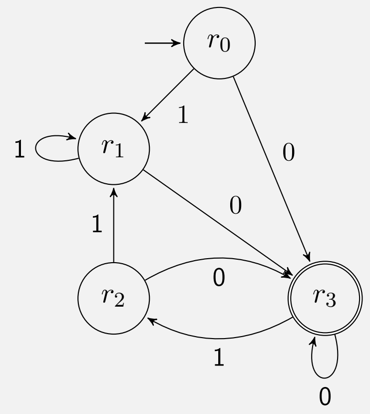
Preuve : exercice

Conséquences

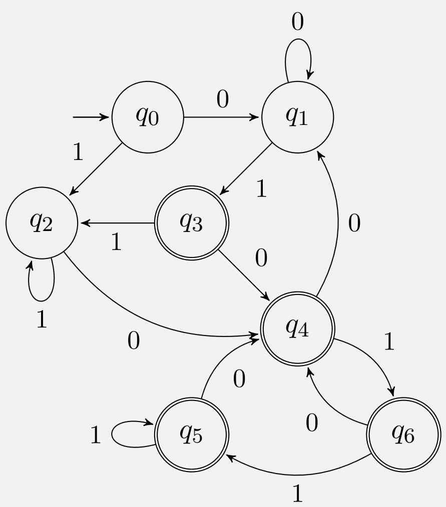
-
-
-

Exemple

Automate de la diapo 4

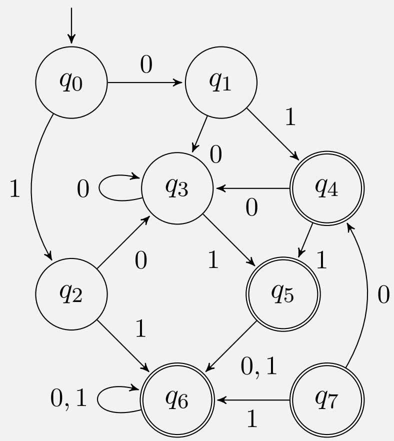


Exemple 2



Exercice

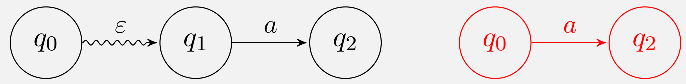
Minimiser l'automate suivant :



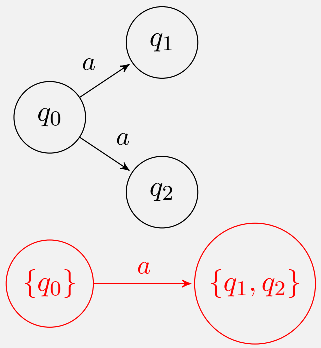
Exercice (suite)

Récapitulatif

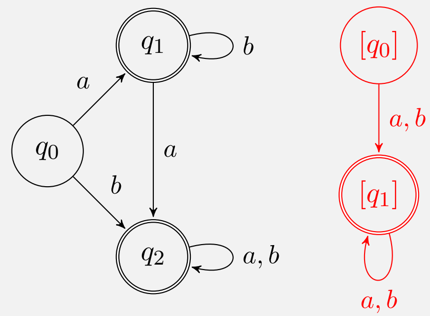
1. Suppression des ε-transitions



2. Déterminisation



3. Minimisation

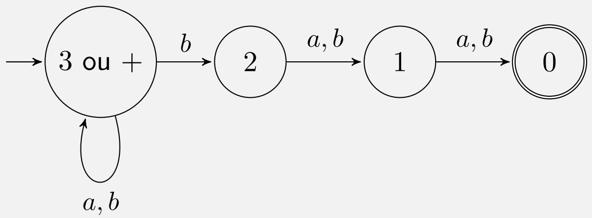


Bonus : exercice de déterminisation

On considère le langage

$$L = \{w \in \{a, b\}^* \mid \text{le 3}^{\text{e}} \text{ symbole en partant de la fin est un } b\}$$

Un automate non-déterministe qui le reconnaît est :



Construire un AFD (complet) équivalent.

Solution