IN 406 – Théorie des Langages
Cours 3 : Automate fini minimum

Franck Quessette – Franck.Quessette@uvsq.fr

Université de Versailles – Saint-Quentin

V3 2019–2020

Rappel 1

Notions déjà vues :

- lettre, alphabet **fini** ;
- mot, langage **fini** ou **infini** ;
- langage rationnel (opérations ensemblistes) ;
- automate **fini** non-déterminsite (AFN), automate **fini** déterministe ;
- langage reconnaissable par automate **fini**.

Rappel 2

Algorithmes déjà vus :

- AFN $\boxed{\textbf{avec}}$ ε-transitions $\Longrightarrow$ AFN $\boxed{\textbf{sans}}$ ε-transition ;
- AFN $\boxed{\textbf{sans}}$ ε-transitions $\Longrightarrow$ AFD ;
- AFD $\Longrightarrow$ AFD complet ;

Rappel 2

Algorithmes déjà vus :

- AFN $\boxed{\textbf{avec}}$ ε-transitions $\Longrightarrow$ AFN $\boxed{\textbf{sans}}$ ε-transition ;
- AFN $\boxed{\textbf{sans}}$ ε-transitions $\Longrightarrow$ AFD ;
- AFD $\Longrightarrow$ AFD complet ;
- $\boxed{\textbf{AFD minimum complet}}$.

Rappel 2

Algorithmes déjà vus :

- AFN $\boxed{\textbf{avec}}$ ε-transitions $\Longrightarrow$ AFN $\boxed{\textbf{sans}}$ ε-transition ;
- AFN $\boxed{\textbf{sans}}$ ε-transitions $\Longrightarrow$ AFD ;
- AFD $\Longrightarrow$ AFD complet ;
- $\boxed{\textbf{AFD minimum complet}}$.

Résultats :

- équivalence des représentations : AFN et AFD ;
- ensemble des langages rationnels = ensemble des langages reconnaissables ;

Rappel 2

Algorithmes déjà vus :

- AFN $\boxed{\textbf{avec}}$ ε-transitions $\Longrightarrow$ AFN $\boxed{\textbf{sans}}$ ε-transition ;
- AFN $\boxed{\textbf{sans}}$ ε-transitions $\Longrightarrow$ AFD ;
- AFD $\Longrightarrow$ AFD complet ;
- $\boxed{\textbf{AFD minimum complet}}$.

Résultats :

- équivalence des représentations : AFN et AFD ;
- ensemble des langages rationnels = ensemble des langages reconnaissables ;
- $\boxed{\textbf{unicité de l'AFD minimum complet}}$.

Automate minimum

Intérêt

Un automate est minimal en nombre d'états.

- Nombre minimum d'états donc stockage minimal.
- L'unicité de l'automate minimal permet de tester l'égalité de langages.

Nouvelles notions

Deux nouvelles notions nécessaires :

- accessibilité ;
- séparabilité.

Définition

Soit $\mathcal{A} = (\Sigma, Q, q_0, F, T)$ un AFD, pour tout $\boxed{\mathbf{q}}$, $\boxed{\mathbf{q'}}$ éléments de Q et pour tout mot $\boxed{\mathbf{w}}$ sur l'alphabet Σ si :

- $\boxed{\mathbf{w}} = a_1 a_2 \ldots a_k$ avec $\forall i \in 1..k, a_i \in \Sigma$;
- $\exists q_1, q_2, \ldots q_k, q_{k+1} \in Q$ tels que :
  - $\boxed{\mathbf{q}} = q_1$;
  - $\forall i \in 1..k, \quad (q_i, a_i, q_{i+1}) \in T$;
  - $\boxed{\mathbf{q'}} = q_{k+1}$;

on dit que $\boxed{\mathbf{q'}}$ est **accessible** depuis $\boxed{\mathbf{q}}$ avec $\boxed{\mathbf{w}}$ et on note

$$\boxed{\boxed{\mathbf{q}} \cdot \boxed{\mathbf{w}} = \boxed{\mathbf{q'}}}.$$

Algorithme

Soit $\mathcal{A} = (\Sigma, Q, q_0, F, T)$, l'ensemble des états accessibles depuis q_0, noté $\boxed{\mathbf{A}}$ se calcule par :

DÉBUT

$\boxed{\mathbf{A}} \leftarrow \{q_0\}$

Tant que il existe $q \in \boxed{\mathbf{A}}$, $q' \notin \boxed{\mathbf{A}}$, $a \in \Sigma$ tel que $(q, a, q') \in T$ **Faire**

  $\boxed{\mathbf{A}} \leftarrow \boxed{\mathbf{A}} \cup \{q'\}$

Fin Tant que

Renvoyer $\boxed{\mathbf{A}}$

FIN

C'est simplement un parcours de graphe à partir du sommet q_0.

Définition

Soit $\mathcal{A} = (\Sigma, Q, q_0, F, T)$ un AFD deux états $\boxed{\mathbf{q}}$ et $\boxed{\mathbf{q'}}$ sont $\boxed{\textbf{séparables}}$ ou $\boxed{\textbf{distingables}}$, si :

$$\exists w \in \Sigma^*,\ \boxed{\mathbf{q}} \cdot w \in F \Rightarrow \boxed{\mathbf{q'}} \cdot w \notin F$$

$$OU$$

$$\exists w \in \Sigma^*,\ \boxed{\mathbf{q}} \cdot w \notin F \Rightarrow \boxed{\mathbf{q'}} \cdot w \in F$$

Définition

Soit $\mathcal{A} = (\Sigma, Q, q_0, F, T)$ un AFD deux états $\boxed{\mathbf{q}}$ et $\boxed{\mathbf{q'}}$ sont $\boxed{\textbf{inséparables}}$ ou $\boxed{\textbf{indistingables}}$, si :

$$\forall w \in \Sigma^*,\ \boxed{\mathbf{q}} \cdot w \in F \iff \boxed{\mathbf{q'}} \cdot w \in F$$

Algorithme

Soit un AFD $\mathcal{A} = (\Sigma, Q, q_0, F, T)$. L'algorithme marque les couples d'états séparables :

DÉBUT

$\forall q \in F, \forall q' \notin F$ $\boxed{\textbf{marquer } (q, q')}$

Tant que $\exists$ $\boxed{(q_1, q_2) \textbf{ non marqué}}$, $\exists a \in \Sigma$, $\exists$ $\boxed{(q_1', q_2') \textbf{ marqué}}$ tels que $(q_1, a, q_1') \in T$ et $(q_2, a, q_2') \in T$ **Faire**

$\boxed{\textbf{marquer } (q_1, q_2)}$

Fin Tant que

FIN

À la fin si un couple d'états est **marqué** les deux états sont **séparables** et si un couple d'état est **non marqué** les deux états sont **inséparables**.

Automate fini déterministe minimum

Définition

Un AFD $\mathcal{A} = (\Sigma, Q, q_0, F, T)$ est $\boxed{\textbf{minimum}}$ si pour tout AFD $\mathcal{A}' = (\Sigma', Q', q_0', F', T')$, tel que $L(\mathcal{A}') = L(\mathcal{A})$ alors :

$$|Q| \leq |Q'|$$

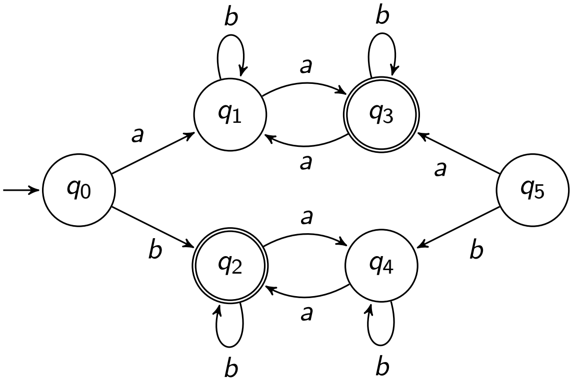
Théorème

Un AFD $\mathcal{A} = (\Sigma, Q, q_0, F, T)$ est $\boxed{\textbf{minimum}}$ si :

- tous les états de Q sont **accessibles** depuis q_0 ;
- tous les états de Q sont deux à deux **séparables**.

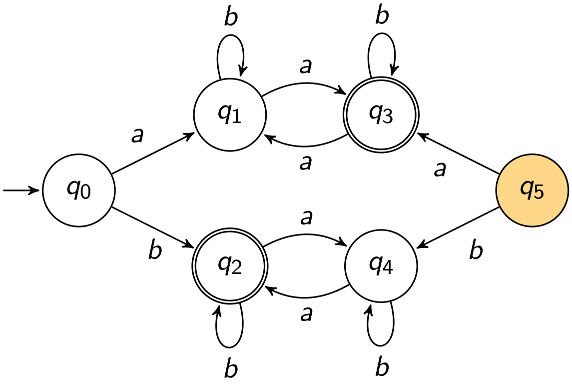
Exemple minimisation

Étape 1 : AFD complet



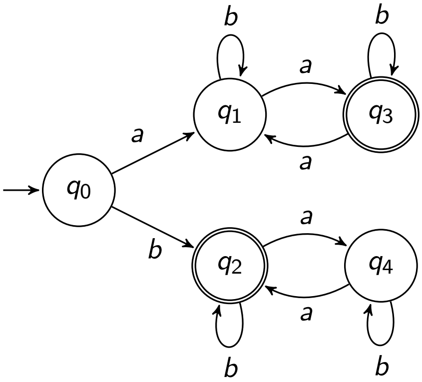
Exemple minimisation

Étape 2 : q_5 inaccessible



Exemple minimisation

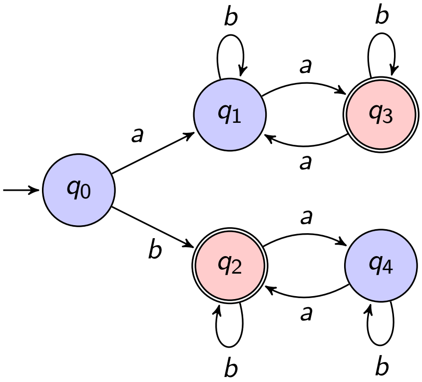
Étape 3 : séparabilité



	q_0	q_1	q_2	q_3	q_4
q_0					
q_1					
q_2					
q_3					
q_4					

Exemple minimisation

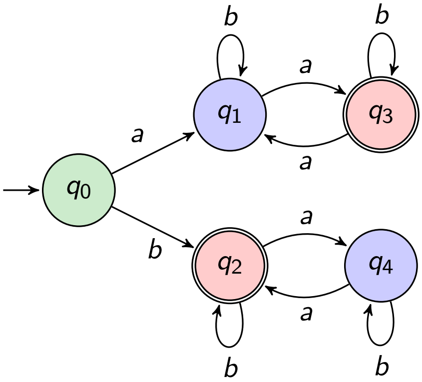
Étape 4 : $q_0, q_1, q_4 \notin F$ et $q_2, q_3 \in F$



	q_0	q_1	q_2	q_3	q_4
q_0			×	×	
q_1			×	×	
q_2	×	×			×
q_3	×	×			×
q_4			×	×	

Exemple minimisation

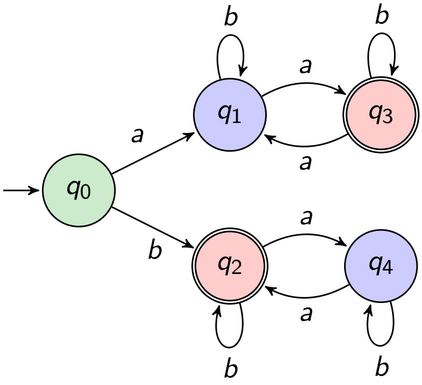
Étape 5 : $q_0 \cdot a \notin F$, $q_1 \cdot a \in F$, $q_4 \cdot a \in F$



	q_0	q_1	q_2	q_3	q_4
q_0		×	×	×	×
q_1	×		×	×	
q_2	×	×			×
q_3	×	×			×
q_4	×		×	×	

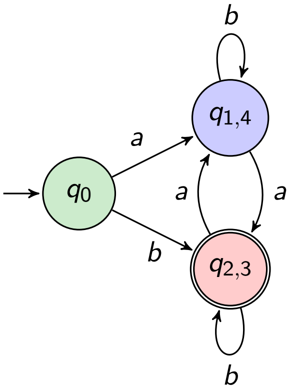
Exemple minimisation

Étape 6 : q_1 et q_4 non séparables, q_2 et q_3 non séparables



	q_0	q_1	q_2	q_3	q_4
q_0	·	×	×	×	×
q_1	×	·	×	×	·
q_2	×	×	·	·	×
q_3	×	×	·	·	×
q_4	×	·	×	×	·

Étape 7 : fusion de q_1 avec q_4 et de de q_2 avec q_3



Exemple minimisation

Fin : Automate fini déterministe minimal complet