

# IN 406 – Théorie des Langages

## Cours 3 : Automate fini minimum

Franck Quessette – [Franck.Quessette@uvsq.fr](mailto:Franck.Quessette@uvsq.fr)

Université de Versailles – Saint-Quentin

V4 2020–2021

# Rappel 1

## Notions déjà vues :

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

## Rappel 2

### Algorithmes déjà vus :

- ▶ AFN **avec**  $\varepsilon$ -transitions  $\implies$  AFN **sans**  $\varepsilon$ -transition ;
- ▶ AFN **sans**  $\varepsilon$ -transitions  $\implies$  AFD ;
- ▶ AFD  $\implies$  AFD complet ;

## Rappel 2

### Algorithmes déjà vus :

- ▶ AFN **avec**  $\varepsilon$ -transitions  $\implies$  AFN **sans**  $\varepsilon$ -transition ;
- ▶ AFN **sans**  $\varepsilon$ -transitions  $\implies$  AFD ;
- ▶ AFD  $\implies$  AFD complet ;
- ▶ **AFD minimum complet** .

## Rappel 2

### Algorithmes déjà vus :

- ▶ AFN **avec**  $\varepsilon$ -transitions  $\implies$  AFN **sans**  $\varepsilon$ -transition ;
- ▶ AFN **sans**  $\varepsilon$ -transitions  $\implies$  AFD ;
- ▶ AFD  $\implies$  AFD complet ;
- ▶ **AFD minimum complet** .

### Résultats :

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

## Rappel 2

### Algorithmes déjà vus :

- ▶ AFN **avec**  $\varepsilon$ -transitions  $\implies$  AFN **sans**  $\varepsilon$ -transition ;
- ▶ AFN **sans**  $\varepsilon$ -transitions  $\implies$  AFD ;
- ▶ AFD  $\implies$  AFD complet ;
- ▶ **AFD minimum complet** .

### Résultats :

- ▶ équivalence des représentations : AFN et AFD ;
- ▶ ensemble des langages rationnels = ensemble des langages reconnaissables ;
- ▶ **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  $q, q'$  éléments de  $Q$  et pour tout mot  $w$  sur l'alphabet  $\Sigma$  si :

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

on dit que  $q'$  est **accessible** depuis  $q$  avec  $w$  et on note

$$q \cdot w = q'.$$



## Algorithme

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

**DÉBUT**

**A**  $\leftarrow \{q_0\}$

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

**Fin Tant que**

Renvoyer **A**

**FIN**

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

## Séparabilité

### Définition

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

$$\begin{aligned} &\exists w \in \Sigma^*, q \cdot w \in F \Rightarrow q' \cdot w \notin F \\ \text{OU} \\ &\exists w \in \Sigma^*, q \cdot w \notin F \Rightarrow q' \cdot w \in F \end{aligned}$$

### Définition

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

$$\forall w \in \Sigma^*, q \cdot w \in F \iff q' \cdot w \in F$$

# Séparabilité

## Algorithme

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

### DÉBUT

$\forall q \in F, \forall q' \notin F$  **marquer**  $(q, q')$

**Tant que**  $\exists (q_1, q_2)$  **non marqué**,  $\exists a \in \Sigma$ ,  $\exists (q'_1, q'_2)$  **marqué**  
tels que  $(q_1, a, q'_1) \in T$  et  $(q_2, a, q'_2) \in T$  **Faire**

**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 **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 **minimum** si et seulement si :

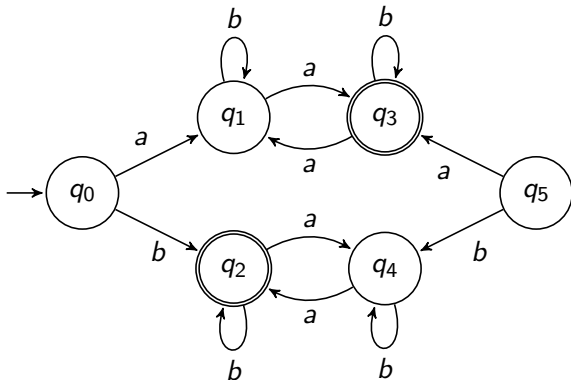
- ▶ tous les états de  $Q$  sont **accessibles** depuis  $q_0$  ;

**ET**

- ▶ tous les états de  $Q$  sont deux à deux **séparables**.

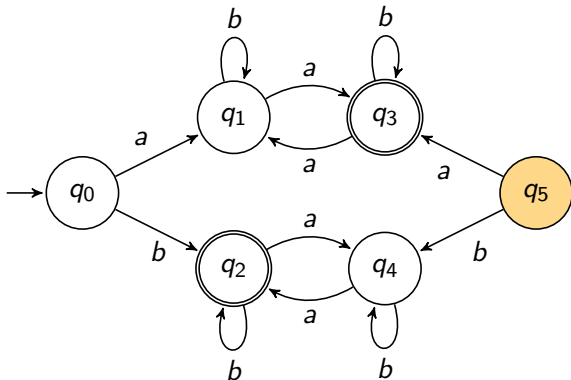
## Exemple minimisation

Étape 1 : AFD complet



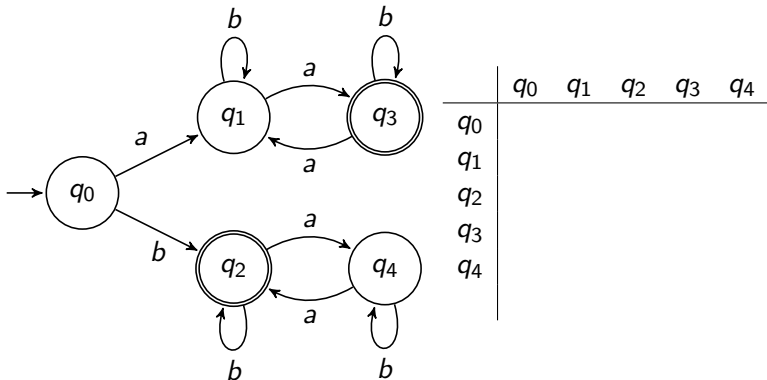
## Exemple minimisation

Étape 2 :  $q_5$  inaccessible



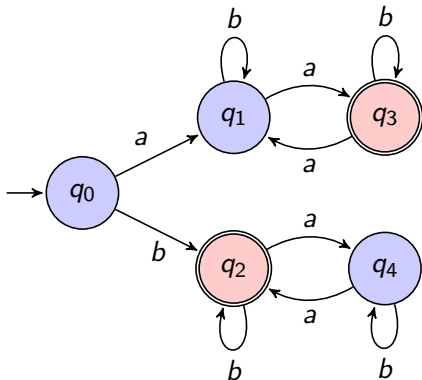
## Exemple minimisation

**Étape 3** : séparabilité



## Exemple minimisation

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

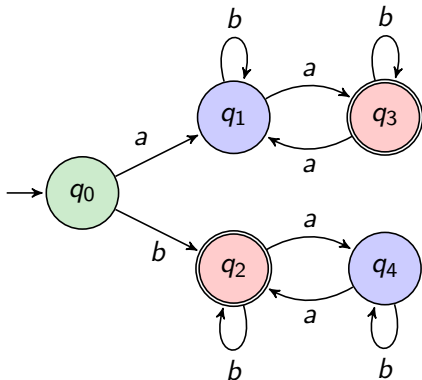


	$q_0$	$q_1$	$q_2$	$q_3$	$q_4$
$q_0$			×	×	
$q_1$			×	×	
$q_2$	×	×			×
$q_3$	×	×			×
$q_4$			×	×	



## Exemple minimisation

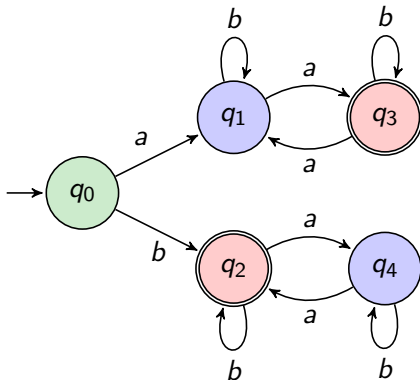
**Étape 5** :  $q_0 \cdot a \notin F$ ,  $q_1 \cdot a \in F$ ,  $q_4 \cdot a \in F$



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

## Exemple minimisation

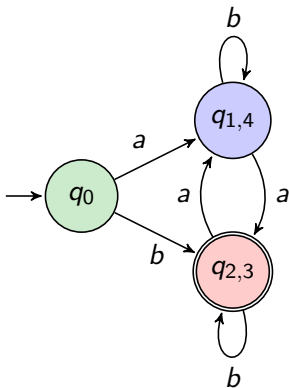
**Étape 6** :  $q_1$  et  $q_4$  non séparables,  $q_2$  et  $q_3$  non séparables



	$q_0$	$q_1$	$q_2$	$q_3$	$q_4$
$q_0$	.	×	×	×	×
$q_1$	×	.	×	×	.
$q_2$	×	×	.	.	×
$q_3$	×	×	.	.	×
$q_4$	×	.	×	×	.

## Exemple minimisation

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



## Exemple minimisation

**Fin** : Automate fini déterministe minimal complet

