# Langages et Automates
# Automates (et langages ?)

Engel Lefaucheux

Prépas des INP

# Objectif du cours

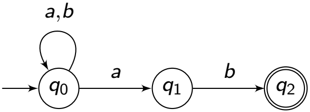
- Découvrir encore d'autres opérations usuelles sur les automates
- Voir ce qu'on peut en faire

# 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 $(q, a, q') \in T$.

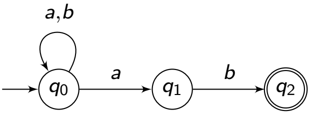


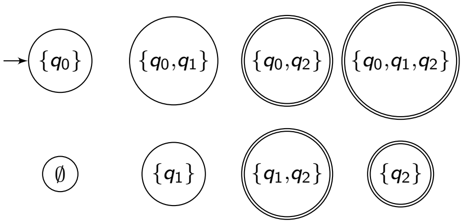
# Déterminisation d'un automate

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 $(q, a, q') \in T$.



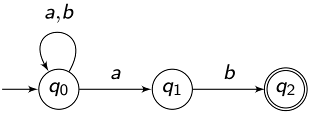



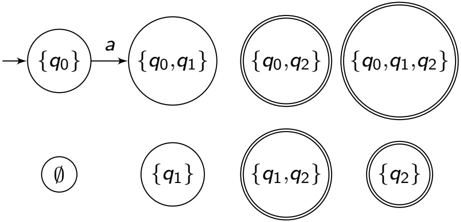
# Déterminisation d'un automate

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 $(q, a, q') \in T$.



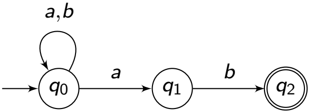



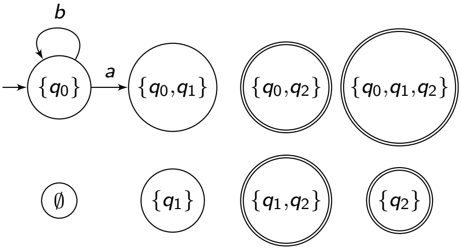
# Déterminisation d'un automate

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 $(q, a, q') \in T$.



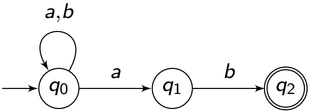



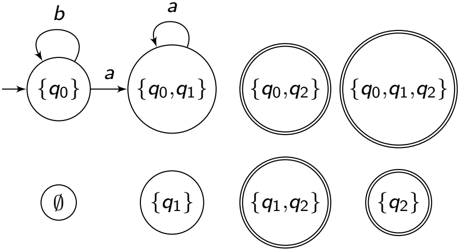
# Déterminisation d'un automate

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 $(q, a, q') \in T$.



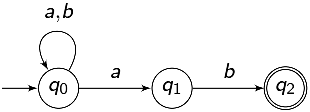



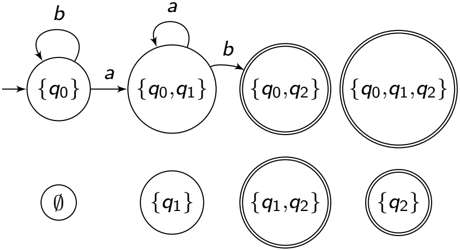
# Déterminisation d'un automate

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 $(q, a, q') \in T$.



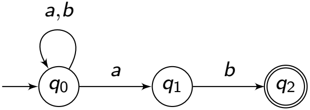



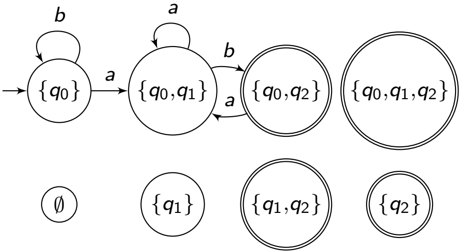
# Déterminisation d'un automate

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 $(q, a, q') \in T$.



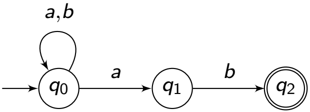



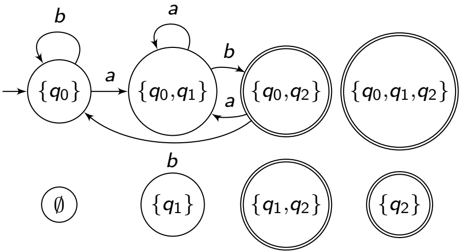
# Déterminisation d'un automate

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

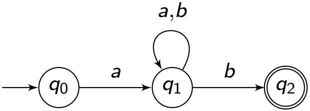
Construire $\mathcal{A}' = (Q', \Sigma, T', I', F')$ tel que

- $Q' = 2^Q$ 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 $(q, a, q') \in T$.

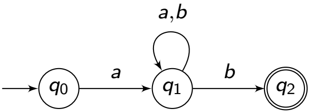


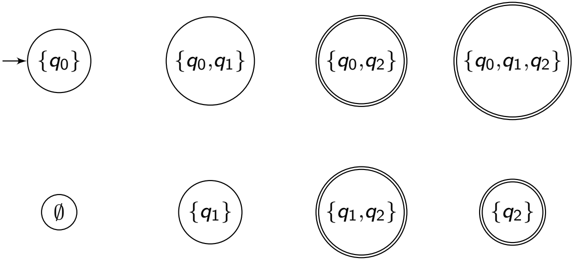



# Second exemple

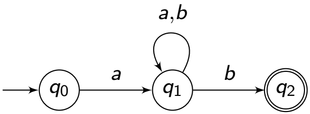


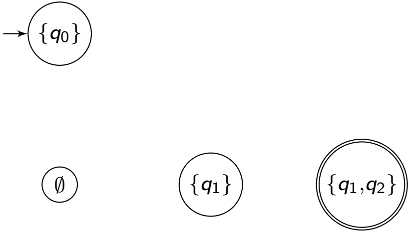
# Second exemple



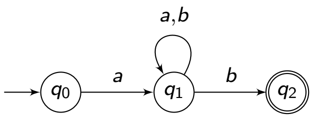



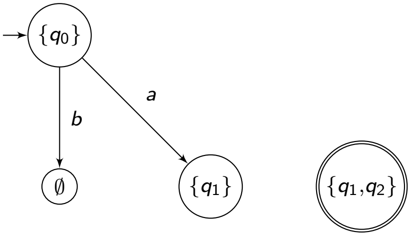
# Second exemple



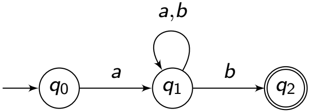



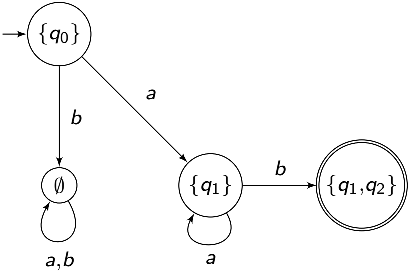
# Second exemple



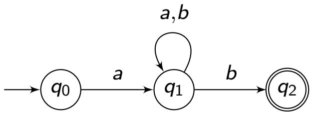



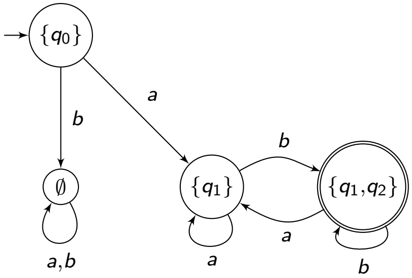
# Second exemple



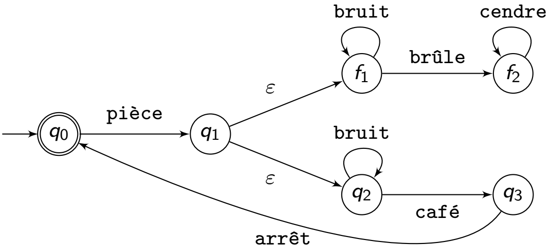
# Second exemple



# Exercice

Déterminisez et émondez l'automate suivant



# 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 $\mathcal{A}_2$ sont déterministes et produisant 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 déduire du résultat de déterminisation ?

# Exercices

Construire un automate déterministe pour les langages suivants

- $L_1 = (a + b + c)^* a (a + b + c)^*$
- $L_2$ : les mots ne contenant pas le facteur *aab*