Théorie des Langages 1

Cours 4 : Automates déterministes

L. Rieg (*thanks* M. Echenim)

Grenoble INP - Ensimag, 1re année

Année 2020-2021

Définition

Définition (Automate déterministe)

Un automate déterministe est un automate fini $\langle Q, V, \delta, I, F \rangle$ sans ε-transition tel que :

- I contient un unique élément
- Pour tout état $p \in Q$ et tout symbole $a \in V$, il existe au plus un état $q \in Q$ tel que $(p, a, q) \in \delta$

Définition

Définition (Automate déterministe)

Un automate déterministe est un automate fini $\langle Q, V, \delta, I, F\rangle$ sans ε-transition tel que :

- I contient un unique élément
- Pour tout état $p \in Q$ et tout symbole $a \in V$, il existe au plus un état $q \in Q$ tel que $(p, a, q) \in \delta$

Conséquences de la définition

- L'automate a un seul état initial
- δ est une fonction partielle : $Q \times V \rightharpoonup Q$:
  Si $(p, a, q) \in \delta$, on pourra noter $\delta(p, a) = q$

Automate déterministe complet

Définition

Un automate déterministe est complet si δ est une fonction totale $\delta : Q \times V \to Q$.

Automate déterministe complet

Définition

Un automate déterministe est complet si δ est une fonction totale $\delta : Q \times V \to Q$.

Remarque

L'algorithme de déterminisation d'un automate qui va être présenté construit un automate déterministe complet.

Automate déterministe complet

Définition

Un automate déterministe est complet si δ est une fonction totale $\delta : Q \times V \to Q$.

Remarque

L'algorithme de déterminisation d'un automate qui va être présenté construit un automate déterministe complet.

Remarque

*Dans la suite de ce cours, **sauf mention explicite**, quand on parlera d'automates déterministes, on supposera toujours qu'ils sont complets.*

Automate déterministe complet

Définition

Un automate déterministe est complet si δ est une fonction totale $\delta : Q \times V \to Q$.

Remarque

L'algorithme de déterminisation d'un automate qui va être présenté construit un automate déterministe complet.

Remarque

Dans la suite de ce cours, ***sauf mention explicite****, quand on parlera d'automates déterministes, on supposera toujours qu'ils sont complets.*

Questions : Pourquoi déterminiser un automate ?
Veut-on toujours le faire ?

Extension de la fonction de transition aux mots

Définition

Soit $A = \langle Q, V, \delta, I, F \rangle$ un AFD (complet). On définit la fonction $\delta^* : Q \times V^* \to Q$ par induction de la façon suivante : pour tout $p \in Q$,

Extension de la fonction de transition aux mots

Définition

Soit $A = \langle Q, V, \delta, I, F \rangle$ un AFD (complet). On définit la fonction $\delta^* : Q \times V^* \to Q$ par induction de la façon suivante : pour tout $p \in Q$,

- $\delta^*(p, \varepsilon) = p$

Extension de la fonction de transition aux mots

Définition

Soit $A = \langle Q, V, \delta, I, F \rangle$ un AFD (complet). On définit la fonction $\delta^* : Q \times V^* \to Q$ par induction de la façon suivante : pour tout $p \in Q$,

- $\delta^*(p, \varepsilon) = p$
- $\delta^*(p, aw) = \delta^*(\delta(p, a), w)$

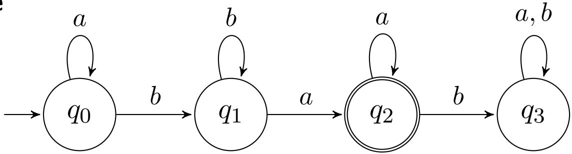
Extension de la fonction de transition aux mots

Définition

Soit $A = \langle Q, V, \delta, I, F \rangle$ un AFD (complet). On définit la fonction $\delta^* : Q \times V^* \to Q$ par induction de la façon suivante : pour tout $p \in Q$,

- $\delta^*(p, \varepsilon) = p$
- $\delta^*(p, aw) = \delta^*(\delta(p, a), w)$

Exemple



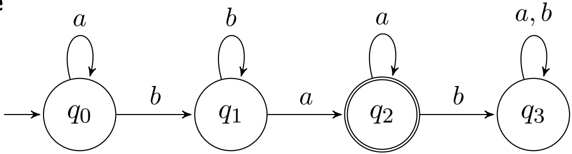
Extension de la fonction de transition aux mots

Définition

Soit $A = \langle Q, V, \delta, I, F \rangle$ un AFD (complet). On définit la fonction δ^* : $Q \times V^* \to Q$ par induction de la façon suivante : pour tout $p \in Q$,

- $\delta^*(p, \varepsilon) = p$
- $\delta^*(p, aw) = \delta^*(\delta(p, a), w)$

Exemple



$\delta^*(q_0, aab) = q_1$ et $\delta^*(q_0, abba) = q_2$

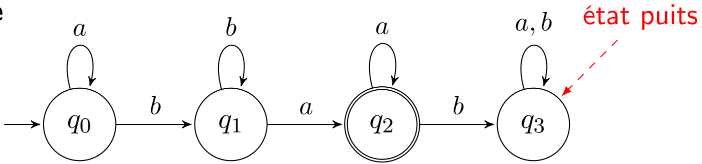
Extension de la fonction de transition aux mots

Définition

Soit $A = \langle Q, V, \delta, I, F\rangle$ un AFD (complet). On définit la fonction $\delta^* : Q \times V^* \to Q$ par induction de la façon suivante : pour tout $p \in Q$,

- $\delta^*(p, \varepsilon) = p$
- $\delta^*(p, aw) = \delta^*(\delta(p, a), w)$

Exemple



$\delta^*(q_0, aab) = q_1$ et $\delta^*(q_0, abba) = q_2$

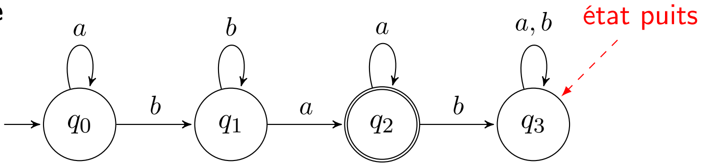
Extension de la fonction de transition aux mots

Définition

Soit $A = \langle Q, V, \delta, I, F \rangle$ un AFD (complet). On définit la fonction $\delta^* : Q \times V^* \to Q$ par induction de la façon suivante : pour tout $p \in Q$,

- $\delta^*(p, \varepsilon) = p$
- $\delta^*(p, aw) = \delta^*(\delta(p, a), w)$

Exemple



$\delta^*(q_0, aab) = q_1$ et $\delta^*(q_0, abba) = q_2$

L'extension de la fonction de transition est parfois notée δ au lieu de δ^*.

Propriétés

Définitions équivalentes

Soit un AFD complet $A = \langle Q, V, \delta, \{q_0\}, F \rangle$

- Langage reconnu par A

Propriétés

Définitions équivalentes

Soit un AFD complet $A = \langle Q, V, \delta, \{q_0\}, F \rangle$

- Langage reconnu par A

$$\mathcal{L}(A) \;=\; \{w \in V^* \mid \delta^*(q_0, w) \in F\}$$

Propriétés

Définitions équivalentes

Soit un AFD complet $A = \langle Q, V, \delta, \{q_0\}, F\rangle$

- Langage reconnu par A

$$\mathcal{L}(A) \;=\; \{w \in V^* \mid \delta^*(q_0, w) \in F\}$$

- L'automate A est initialement connecté si et seulement si

Propriétés

Définitions équivalentes

Soit un AFD complet $A = \langle Q, V, \delta, \{q_0\}, F \rangle$

- Langage reconnu par A

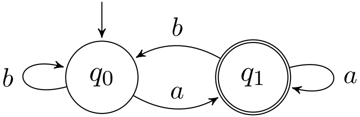
$$\mathcal{L}(A) \;=\; \{w \in V^* \mid \delta^*(q_0, w) \in F\}$$

- L'automate A est initialement connecté si et seulement si

$$\forall p \in Q,\; \exists w \in V^*,\, \delta^*(q_0, w) = p$$

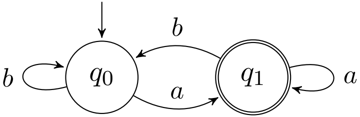
Test d'appartenance

$L = \{a, b\}^* \{a\}^+$



Test d'appartenance

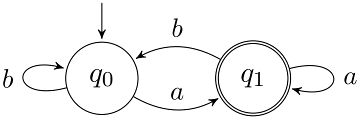
$$L = \{a, b\}^* \{a\}^+$$



$bbaba \in L$

Test d'appartenance

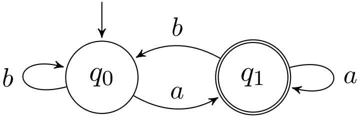
$$L = \{a, b\}^* \{a\}^+$$



$$bbaba \in L \iff \delta^*(q_0, bbaba) \in F$$

Test d'appartenance

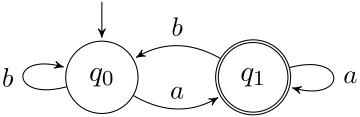
$$L = \{a, b\}^* \{a\}^+$$



$$bbaba \in L \iff \delta^*(q_0, bbaba) \in F \iff \delta^*(q_0, baba) \in F$$

Test d'appartenance

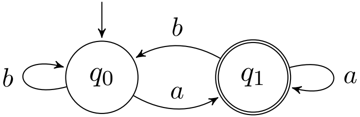
$$L = \{a, b\}^* \{a\}^+$$



$$\begin{aligned} bbaba \in L &\iff \delta^*(q_0, bbaba) \in F \iff \delta^*(q_0, baba) \in F \\ &\iff \delta^*(q_0, aba) \in F \end{aligned}$$

Test d'appartenance

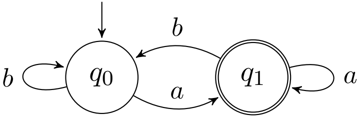
$L = \{a, b\}^* \{a\}^+$



$$\begin{aligned} bbaba \in L &\iff \delta^*(q_0, bbaba) \in F \iff \delta^*(q_0, baba) \in F \\ &\iff \delta^*(q_0, aba) \in F \quad \iff \delta^*(q_1, ba) \in F \end{aligned}$$

Test d'appartenance

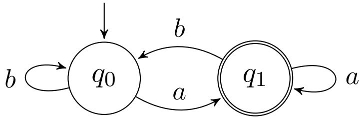
$L = \{a, b\}^* \{a\}^+$



$$
\begin{aligned}
bbaba \in L &\iff \delta^*(q_0, bbaba) \in F \iff \delta^*(q_0, baba) \in F \\
&\iff \delta^*(q_0, aba) \in F \quad \iff \delta^*(q_1, ba) \in F \\
&\iff \delta^*(q_0, a) \in F
\end{aligned}
$$

Test d'appartenance

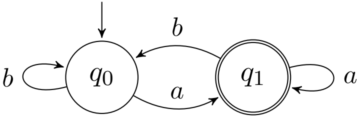
$L = \{a, b\}^* \{a\}^+$



$$\begin{aligned}
bbaba \in L \iff \delta^*(q_0, bbaba) \in F &\iff \delta^*(q_0, baba) \in F \\
\iff \delta^*(q_0, aba) \in F &\iff \delta^*(q_1, ba) \in F \\
\iff \delta^*(q_0, a) \in F &\iff \delta^*(q_1, \varepsilon) \in F
\end{aligned}$$

Test d'appartenance

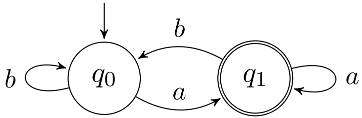
$L = \{a, b\}^* \{a\}^+$



$$\begin{aligned}
bbaba \in L &\iff \delta^*(q_0, bbaba) \in F \iff \delta^*(q_0, baba) \in F \\
&\iff \delta^*(q_0, aba) \in F \quad \iff \delta^*(q_1, ba) \in F \\
&\iff \delta^*(q_0, a) \in F \qquad \iff \delta^*(q_1, \varepsilon) \in F \iff q_1 \in F
\end{aligned}$$

Test d'appartenance

$$L = \{a, b\}^* \{a\}^+$$

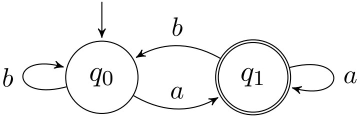


$$\begin{aligned} bbaba \in L &\iff \delta^*(q_0, bbaba) \in F \iff \delta^*(q_0, baba) \in F \\ &\iff \delta^*(q_0, aba) \in F \quad \iff \delta^*(q_1, ba) \in F \\ &\iff \delta^*(q_0, a) \in F \quad\;\; \iff \delta^*(q_1, \varepsilon) \in F \iff q_1 \in F \end{aligned}$$

fonction reconnaître(q : état, w : mot) **renvoie Booléen** =
  tant que $w \neq \varepsilon$ **faire**
    $s \leftarrow$ premier_symbole(w)
    $w \leftarrow$ reste_mot(w)
    $q \leftarrow \delta(q, s)$
  fin tant que
  renvoyer ($q \in F$)

Test d'appartenance

$$L = \{a, b\}^* \{a\}^+$$



$$\begin{aligned} bbaba \in L \iff \delta^*(q_0, bbaba) \in F &\iff \delta^*(q_0, baba) \in F \\ \iff \delta^*(q_0, aba) \in F &\iff \delta^*(q_1, ba) \in F \\ \iff \delta^*(q_0, a) \in F &\iff \delta^*(q_1, \varepsilon) \in F \iff q_1 \in F \end{aligned}$$

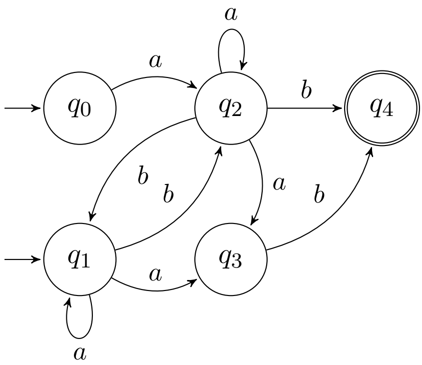
fonction reconnaître(q : état, w : mot) **renvoie Booléen** =
  tant que $w \neq \varepsilon$ **faire**
    $s \leftarrow$ premier_symbole(w)
    $w \leftarrow$ reste_mot(w)
    $q \leftarrow \delta(q, s)$
  fin tant que
  renvoyer ($q \in F$)

$\forall w \in V^*$, $w \in \mathcal{L}(A)$ si et seulement si reconnaître(q_0, w) = **vrai**

Principe de la déterminisation

Idée : suivre tous les chemins en parallèle

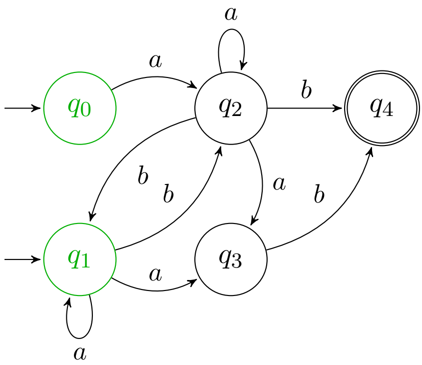
Exemple : L'automate suivant reconnaît-il aab ?



Principe de la déterminisation

Idée : suivre tous les chemins en parallèle

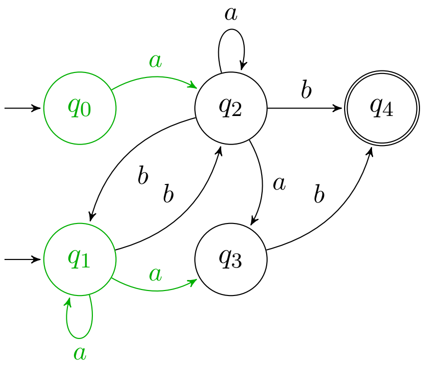
Exemple : L'automate suivant reconnaît-il aab ?



Principe de la déterminisation

Idée : suivre tous les chemins en parallèle

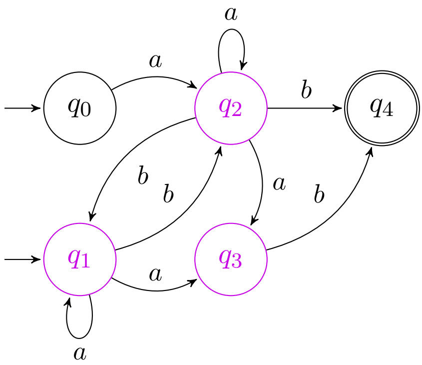
Exemple : L'automate suivant reconnaît-il aab ?



Principe de la déterminisation

Idée : suivre tous les chemins en parallèle

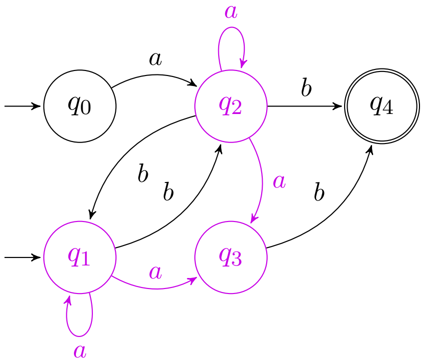
Exemple : L'automate suivant reconnaît-il aab ?



Principe de la déterminisation

Idée : suivre tous les chemins en parallèle

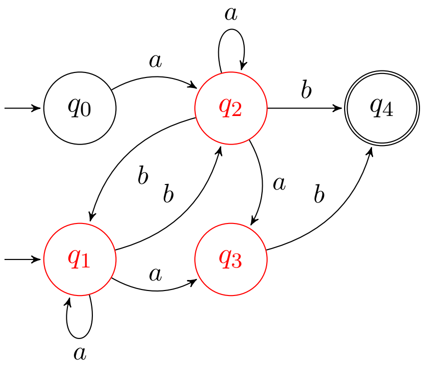
Exemple : L'automate suivant reconnaît-il aab ?



Principe de la déterminisation

Idée : suivre tous les chemins en parallèle

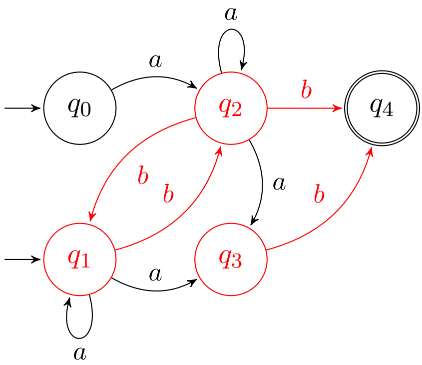
Exemple : L'automate suivant reconnaît-il aab ?



Principe de la déterminisation

Idée : suivre tous les chemins en parallèle

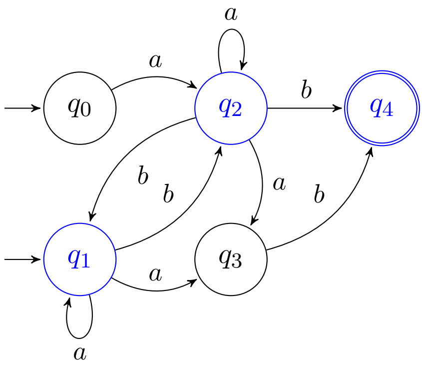
Exemple : L'automate suivant reconnaît-il aab ?



Principe de la déterminisation

Idée : suivre tous les chemins en parallèle

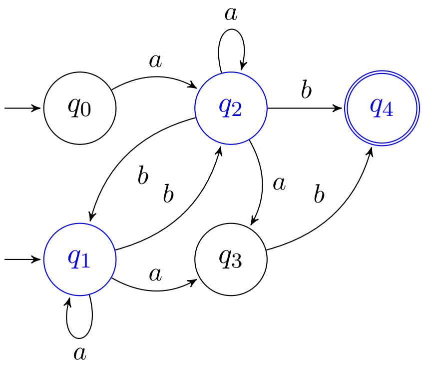
Exemple : L'automate suivant reconnaît-il aab ?



Principe de la déterminisation

Idée : suivre tous les chemins en parallèle

Exemple : L'automate suivant reconnaît-il aab ?

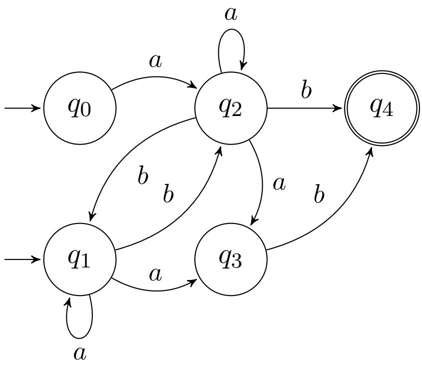


OK : $aab \in \mathcal{L}(A)$

Principe de la déterminisation

Idée : suivre tous les chemins en parallèle

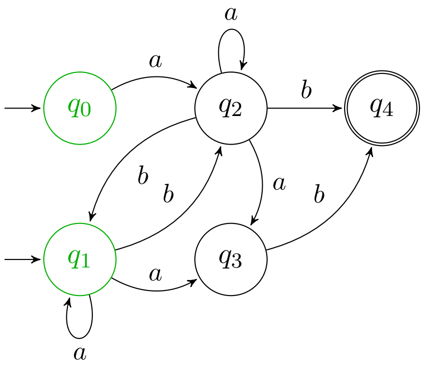
Exemple : L'automate suivant reconnaît-il aab ?



Principe de la déterminisation

Idée : suivre tous les chemins en parallèle

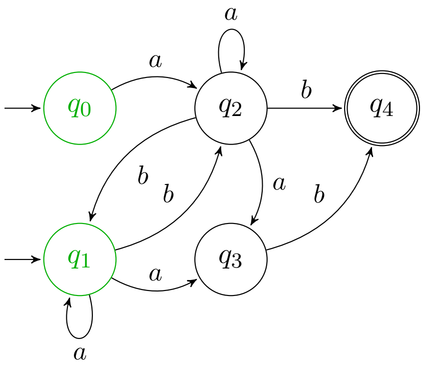
Exemple : L'automate suivant reconnaît-il aab ?

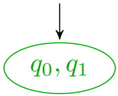


Principe de la déterminisation

Idée : suivre tous les chemins en parallèle

Exemple : L'automate suivant reconnaît-il aab ?

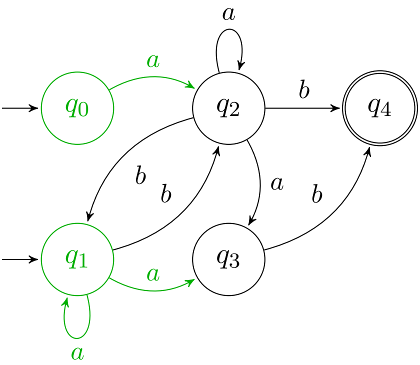


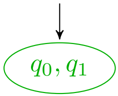


Principe de la déterminisation

Idée : suivre tous les chemins en parallèle

Exemple : L'automate suivant reconnaît-il aab ?

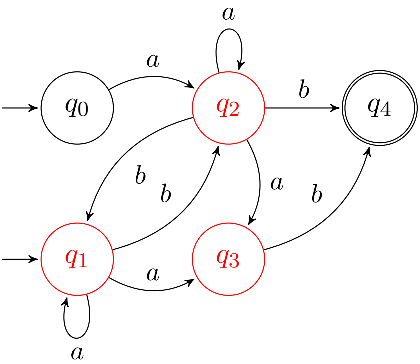


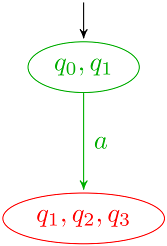


Principe de la déterminisation

Idée : suivre tous les chemins en parallèle

Exemple : L'automate suivant reconnaît-il aab ?

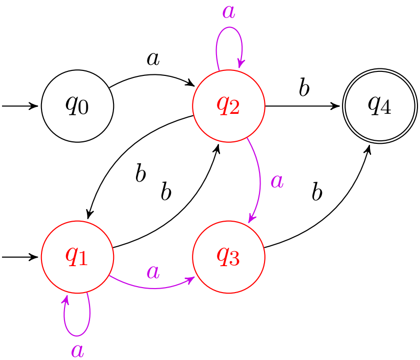


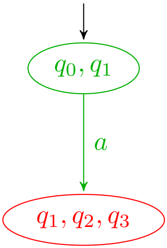


Principe de la déterminisation

Idée : suivre tous les chemins en parallèle

Exemple : L'automate suivant reconnaît-il aab ?

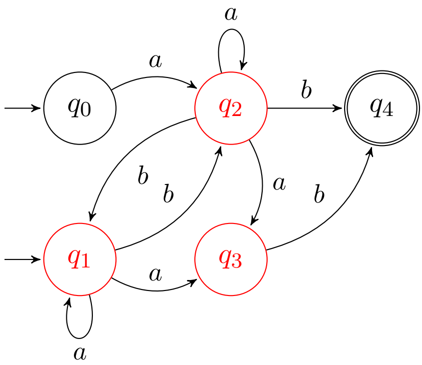


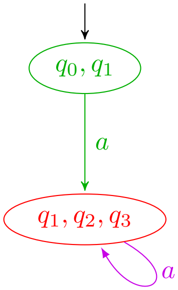


Principe de la déterminisation

Idée : suivre tous les chemins en parallèle

Exemple : L'automate suivant reconnaît-il aab ?

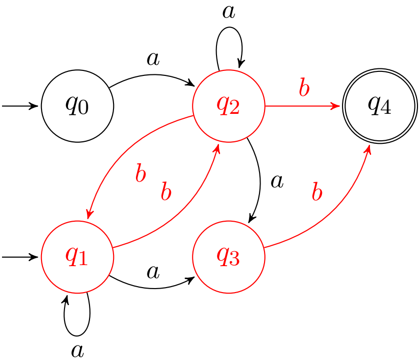


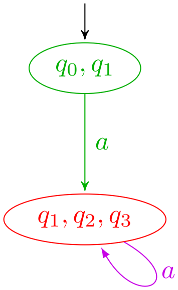


Principe de la déterminisation

Idée : suivre tous les chemins en parallèle

Exemple : L'automate suivant reconnaît-il aab ?

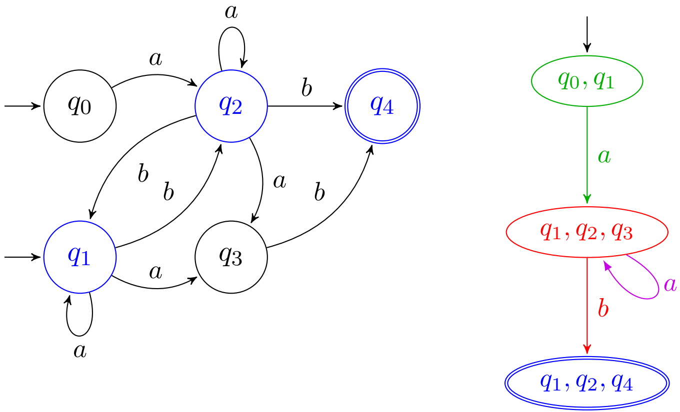




Principe de la déterminisation

Idée : suivre tous les chemins en parallèle

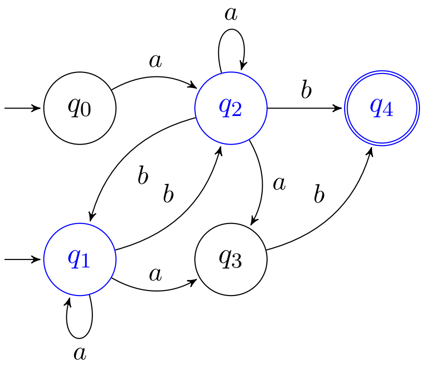
Exemple : L'automate suivant reconnaît-il aab ?

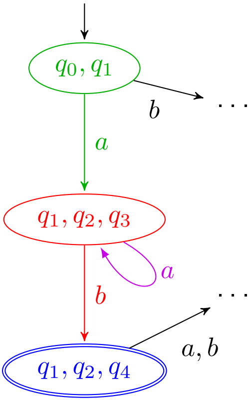


Principe de la déterminisation

Idée : suivre tous les chemins en parallèle

Exemple : L'automate suivant reconnaît-il aab ?





Définition

Définition (Automate des parties)

Etant donné un automate $A = \langle Q, V, \delta_A, I, F_A \rangle$ sans ε-transition, on construit l'automate $B = \langle \mathcal{P}(Q), V, \delta_B, \{I\}, F_B \rangle$, où :

- δ_B est défini par

$$\forall P \subseteq Q, \forall a \in V,\ \delta_B(P, a) = \{q \in Q \mid \exists p \in P : (p, a, q) \in \delta_A\}$$

- $F_B = \{P \subseteq Q \mid P \cap F_A \neq \emptyset\}$

Définition

Définition (Automate des parties)

Etant donné un automate $A = \langle Q, V, \delta_A, I, F_A \rangle$ sans ε-transition, on construit l'automate $B = \langle \mathcal{P}(Q), V, \delta_B, \{I\}, F_B \rangle$, où :

- δ_B est défini par

$$\forall P \subseteq Q, \forall a \in V,\ \delta_B(P, a) \ = \ \{q \in Q \mid \exists p \in P : (p, a, q) \in \delta_A\}$$

- $F_B = \{P \subseteq Q \mid P \cap F_A \neq \emptyset\}$

Remarques

- $P \subseteq Q \iff P \in \mathcal{P}(Q)$ *et* $\emptyset \subseteq Q$ *: un état puits de* B
- *Certains* $P \subseteq Q$ *peuvent ne pas être accessibles depuis* I *donc on construit* B *de proche en proche à partir de* I.

Définition

Définition (Automate des parties)

Etant donné un automate $A = \langle Q, V, \delta_A, I, F_A \rangle$ sans ε-transition, on construit l'automate $B = \langle \mathcal{P}(Q), V, \delta_B, \{I\}, F_B \rangle$, où :

- δ_B est défini par

$$\forall P \subseteq Q, \forall a \in V,\ \delta_B(P, a) = \{q \in Q \mid \exists p \in P : (p, a, q) \in \delta_A\}$$

- $F_B = \{P \subseteq Q \mid P \cap F_A \neq \emptyset\}$

Remarques

- $P \subseteq Q \iff P \in \mathcal{P}(Q)$ *et* $\emptyset \subseteq Q$ *: un état puits de* B
- *Certains* $P \subseteq Q$ *peuvent ne pas être accessibles depuis* I *donc on construit* B *de proche en proche à partir de* I.

Proposition

L'automate B *est un automate fini déterministe complet.*

Propriété

Proposition

Pour tout $w \in V^*$ *et pour tout* $P \subseteq Q$, *on a*
$\delta_B^*(P, w) = \{q \in Q \mid \exists p \in P, \exists$ *un chemin dans* A *de* p *à* q *de trace* $w\}$.

Propriété

Proposition

Pour tout $w \in V^*$ *et pour tout* $P \subseteq Q$, *on a*
$\delta^*_B(P, w) = \{q \in Q \mid \exists p \in P, \exists$ *un chemin dans* A *de* p *à* q *de trace* $w\}$.

Preuve : par induction sur w

Propriété

Proposition

Pour tout $w \in V^*$ *et pour tout* $P \subseteq Q$*, on a*
$\delta^*_B(P, w) = \{q \in Q \mid \exists p \in P, \exists$ *un chemin dans* A *de* p *à* q *de trace* $w\}$.

Preuve : par induction sur w

- $w = \varepsilon$:

Propriété

Proposition

Pour tout $w \in V^*$ et pour tout $P \subseteq Q$, on a
$\delta_B^*(P, w) = \{q \in Q \mid \exists p \in P, \exists$ *un chemin dans* A *de* p *à* q *de trace* $w\}$.

Preuve : par induction sur w

- $w = \varepsilon$:
  - $\delta_B^*(P, \varepsilon) = P$

Propriété

Proposition

Pour tout $w \in V^*$ et pour tout $P \subseteq Q$, on a
$\delta_B^*(P, w) = \{q \in Q \mid \exists p \in P, \exists$ *un chemin dans* A *de* p *à* q *de trace* $w\}$.

Preuve : par induction sur w

- $w = \varepsilon$:
  - $\delta_B^*(P, \varepsilon) = P$
  - $\{q \in Q \mid \exists p \in P, \exists$ un chemin dans A de p à q de trace $\varepsilon\} = P$
    (car A est un automate sans ε-transition)

Propriété

Proposition

Pour tout $w \in V^*$ *et pour tout* $P \subseteq Q$*, on a*
$\delta_B^*(P, w) = \{q \in Q \mid \exists p \in P, \exists$ *un chemin dans* A *de* p *à* q *de trace* $w\}$.

Preuve : par induction sur w

Propriété

Proposition

Pour tout $w \in V^*$ *et pour tout* $P \subseteq Q$, *on a*
$\delta_B^*(P, w) = \{q \in Q \mid \exists p \in P, \exists$ *un chemin dans* A *de* p *à* q *de trace* $w\}$.

Preuve : par induction sur w

- $w = aw'$:

Propriété

Proposition

Pour tout $w \in V^*$ *et pour tout* $P \subseteq Q$*, on a*
$\delta_B^*(P, w) = \{q \in Q \mid \exists p \in P, \exists$ *un chemin dans* A *de* p *à* q *de trace* $w\}$.

Preuve : par induction sur w

- $w = aw'$:
  - Posons $P' \stackrel{\text{def}}{=} \delta_B(P, a) = \{q \in Q \mid \exists p \in P, (p, a, q) \in \delta_A\}$

Propriété

Proposition

Pour tout $w \in V^*$ *et pour tout* $P \subseteq Q$*, on a*
$\delta_B^*(P, w) = \{q \in Q \mid \exists p \in P, \exists$ *un chemin dans* A *de* p *à* q *de trace* $w\}$.

Preuve : par induction sur w

- $w = aw'$:
  - Posons $P' \stackrel{\text{def}}{=} \delta_B(P, a) = \{q \in Q \mid \exists p \in P, (p, a, q) \in \delta_A\}$
    $\delta_B^*(P, aw') = \delta_B^*(\delta_B(P, a), w') = \delta_B^*(P', w')$

Propriété

Proposition

Pour tout $w \in V^*$ *et pour tout* $P \subseteq Q$, *on a*
$\delta_B^*(P, w) = \{q \in Q \mid \exists p \in P, \exists$ *un chemin dans* A *de* p *à* q *de trace* $w\}$.

Preuve : par induction sur w

- $w = aw'$:
  - Posons $P' \overset{\text{def}}{=} \delta_B(P, a) = \{q \in Q \mid \exists p \in P, (p, a, q) \in \delta_A\}$
    $\delta_B^*(P, aw') = \delta_B^*(\delta_B(P, a), w') = \delta_B^*(P', w')$
    $\overset{\text{HI}}{=} \{q' \in Q \mid \exists p' \in P', \exists$ un chemin de p' à q' de trace $w'\}$

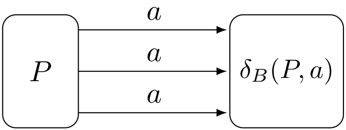
Propriété

Proposition

Pour tout $w \in V^*$ et pour tout $P \subseteq Q$, on a
$\delta_B^*(P, w) = \{q \in Q \mid \exists p \in P, \exists$ *un chemin dans* A *de* p *à* q *de trace* $w\}$.

Preuve : par induction sur w

- $w = aw'$:
  - Posons $P' \stackrel{\text{def}}{=} \delta_B(P, a) = \{q \in Q \mid \exists p \in P, (p, a, q) \in \delta_A\}$
    $\delta_B^*(P, aw') = \delta_B^*(\delta_B(P, a), w') = \delta_B^*(P', w')$
    $\stackrel{\text{HI}}{=} \{q' \in Q \mid \exists p' \in P', \exists$ un chemin de p' à q' de trace $w'\}$



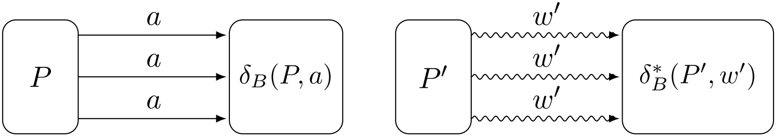
Propriété

Proposition

Pour tout $w \in V^*$ et pour tout $P \subseteq Q$, on a
$\delta_B^*(P, w) = \{q \in Q \mid \exists p \in P, \exists$ *un chemin dans* A *de* p *à* q *de trace* $w\}$.

Preuve : par induction sur w

- $w = aw'$:
  - Posons $P' \stackrel{\text{def}}{=} \delta_B(P, a) = \{q \in Q \mid \exists p \in P, (p, a, q) \in \delta_A\}$
    $\delta_B^*(P, aw') = \delta_B^*(\delta_B(P, a), w') = \delta_B^*(P', w')$
    $\stackrel{\text{HI}}{=} \{q' \in Q \mid \exists p' \in P', \exists$ un chemin de p' à q' de trace $w'\}$



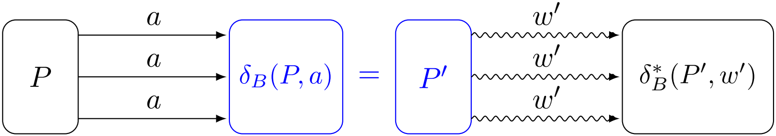
Propriété

Proposition

Pour tout $w \in V^*$ *et pour tout* $P \subseteq Q$, *on a*
$\delta_B^*(P, w) = \{q \in Q \mid \exists p \in P, \exists$ *un chemin dans* A *de* p *à* q *de trace* $w\}$.

Preuve : par induction sur w

- $w = aw'$:
  - Posons $P' \overset{\text{def}}{=} \delta_B(P, a) = \{q \in Q \mid \exists p \in P, (p, a, q) \in \delta_A\}$
    $\delta_B^*(P, aw') = \delta_B^*(\delta_B(P, a), w') = \delta_B^*(P', w')$
    $\overset{\text{HI}}{=} \{q' \in Q \mid \exists p' \in P', \exists$ un chemin de p' à q' de trace $w'\}$



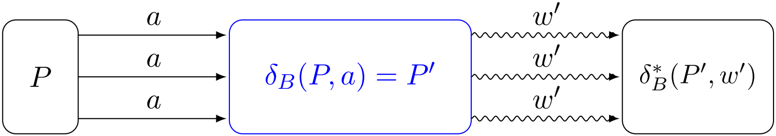
Propriété

Proposition

Pour tout $w \in V^*$ *et pour tout* $P \subseteq Q$*, on a*
$\delta_B^*(P, w) = \{q \in Q \mid \exists p \in P, \exists$ *un chemin dans* A *de* p *à* q *de trace* $w\}$*.*

Preuve : par induction sur w

- $w = aw'$:
  - Posons $P' \stackrel{\text{def}}{=} \delta_B(P, a) = \{q \in Q \mid \exists p \in P, (p, a, q) \in \delta_A\}$
    $\delta_B^*(P, aw') = \delta_B^*(\delta_B(P, a), w') = \delta_B^*(P', w')$
    $\stackrel{\text{HI}}{=} \{q' \in Q \mid \exists p' \in P', \exists$ un chemin de p' à q' de trace $w'\}$



Résultat principal

Théorème

L'automate B est équivalent à A.

Résultat principal

Théorème

L'automate B est équivalent à A.

Preuve :

$$w \in \mathcal{L}(A) \iff \exists p \in I, \exists q \in F_A, \exists \text{ un chemin de } p \text{ à } q \text{ de trace } w$$

Résultat principal

Théorème

L'automate B est équivalent à A.

Preuve :

$$
\begin{aligned}
w \in \mathcal{L}(A) \quad & \iff \quad \exists p \in I,\ \exists q \in F_A,\ \exists \text{ un chemin de } p \text{ à } q \text{ de trace } w \\
& \iff \quad \delta_B^*(I, w) \cap F_A \neq \emptyset
\end{aligned}
$$

Résultat principal

Théorème

L'automate B est équivalent à A.

Preuve :

$$
\begin{aligned}
w \in \mathcal{L}(A) &\iff \exists p \in I,\ \exists q \in F_A,\ \exists \text{ un chemin de } p \text{ à } q \text{ de trace } w \\
&\iff \delta_B^*(I, w) \cap F_A \neq \emptyset \\
&\iff \delta_B^*(I, w) \in F_B
\end{aligned}
$$

Résultat principal

Théorème

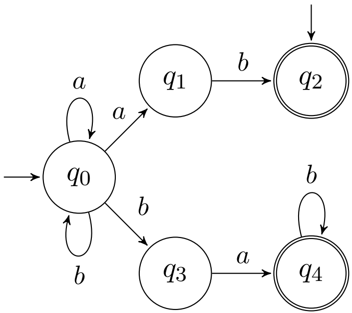
L'automate B est équivalent à A.

Preuve :

$$
\begin{array}{rcl}
w \in \mathcal{L}(A) & \iff & \exists p \in I,\, \exists q \in F_A,\, \exists \text{ un chemin de } p \text{ à } q \text{ de trace } w \\
& \iff & \delta_B^*(I, w) \cap F_A \neq \emptyset \\
& \iff & \delta_B^*(I, w) \in F_B \\
& \iff & w \in \mathcal{L}(B)
\end{array}
$$

Exercice 1

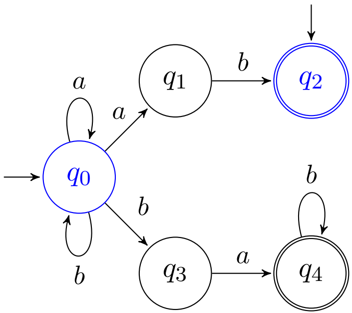
Construire un AFD (complet) équivalent à :



Exercice 1

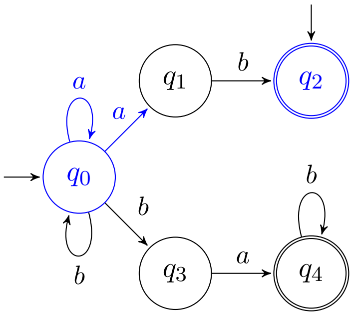
Construire un AFD (complet) équivalent à :

	$\mathcal{P}(Q)$	a	b
$\mathcal{I}$	$\{q_0, q_2\}$		



Exercice 1

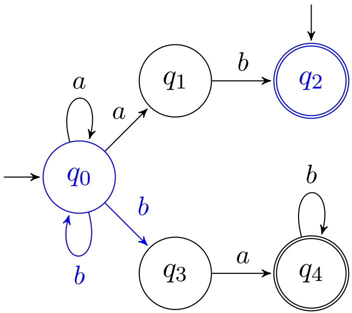
Construire un AFD (complet) équivalent à :



	$\mathcal{P}(Q)$	a	b
$\mathcal{I}$	$\{q_0, q_2\}$	$\{q_0, q_1\}$	
	$\{q_0, q_1\}$		

Exercice 1

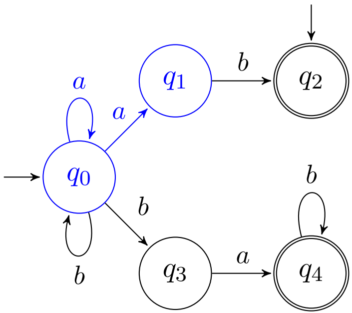
Construire un AFD (complet) équivalent à :



	$\mathcal{P}(Q)$	a	b
$\mathcal{I}$	$\{q_0, q_2\}$	$\{q_0, q_1\}$	$\{q_0, q_3\}$
	$\{q_0, q_1\}$		
	$\{q_0, q_3\}$		

Exercice 1

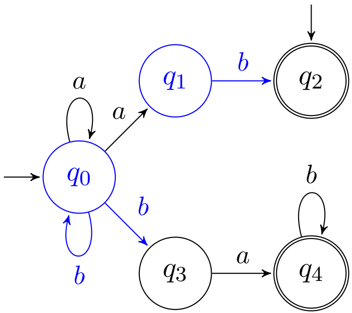
Construire un AFD (complet) équivalent à :



	$\mathcal{P}(Q)$	a	b
$\mathcal{I}$	$\{q_0, q_2\}$	$\{q_0, q_1\}$	$\{q_0, q_3\}$
	$\{q_0, q_1\}$	$\{q_0, q_1\}$	
	$\{q_0, q_3\}$		

Exercice 1

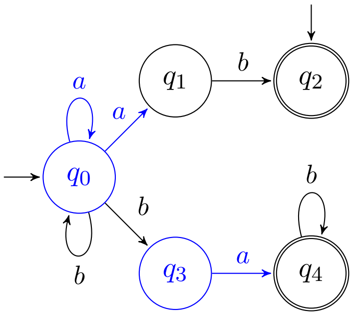
Construire un AFD (complet) équivalent à :



	$\mathcal{P}(Q)$	a	b
$\mathcal{I}$	$\{q_0, q_2\}$	$\{q_0, q_1\}$	$\{q_0, q_3\}$
	$\{q_0, q_1\}$	$\{q_0, q_1\}$	$\{q_0, q_2, q_3\}$
	$\{q_0, q_3\}$		
	$\{q_0, q_2, q_3\}$		

Exercice 1

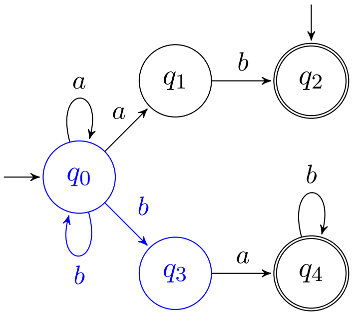
Construire un AFD (complet) équivalent à :



	$\mathcal{P}(Q)$	a	b
$\mathcal{I}$	$\{q_0, q_2\}$	$\{q_0, q_1\}$	$\{q_0, q_3\}$
	$\{q_0, q_1\}$	$\{q_0, q_1\}$	$\{q_0, q_2, q_3\}$
	$\{q_0, q_3\}$	$\{q_0, q_1, q_4\}$	
	$\{q_0, q_2, q_3\}$		
	$\{q_0, q_1, q_4\}$		

Exercice 1

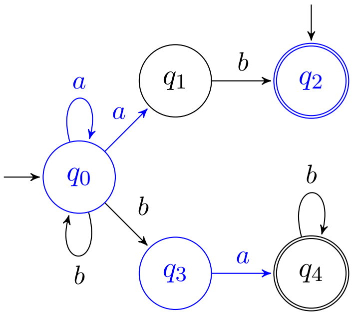
Construire un AFD (complet) équivalent à :



	$\mathcal{P}(Q)$	a	b
$\mathcal{I}$	$\{q_0, q_2\}$	$\{q_0, q_1\}$	$\{q_0, q_3\}$
	$\{q_0, q_1\}$	$\{q_0, q_1\}$	$\{q_0, q_2, q_3\}$
	$\{q_0, q_3\}$	$\{q_0, q_1, q_4\}$	$\{q_0, q_3\}$
	$\{q_0, q_2, q_3\}$		
	$\{q_0, q_1, q_4\}$		

Exercice 1

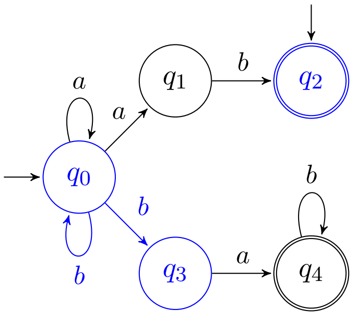
Construire un AFD (complet) équivalent à :



	$\mathcal{P}(Q)$	a	b
$\mathcal{I}$	$\{q_0, q_2\}$	$\{q_0, q_1\}$	$\{q_0, q_3\}$
	$\{q_0, q_1\}$	$\{q_0, q_1\}$	$\{q_0, q_2, q_3\}$
	$\{q_0, q_3\}$	$\{q_0, q_1, q_4\}$	$\{q_0, q_3\}$
	$\{q_0, q_2, q_3\}$	$\{q_0, q_1, q_4\}$	
	$\{q_0, q_1, q_4\}$		

Exercice 1

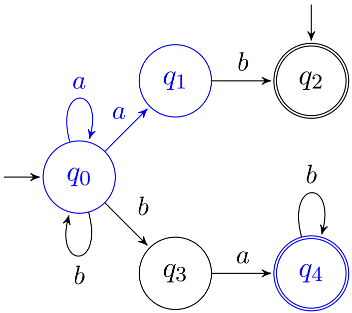
Construire un AFD (complet) équivalent à :



	$\mathcal{P}(Q)$	a	b
$\mathcal{I}$	$\{q_0, q_2\}$	$\{q_0, q_1\}$	$\{q_0, q_3\}$
	$\{q_0, q_1\}$	$\{q_0, q_1\}$	$\{q_0, q_2, q_3\}$
	$\{q_0, q_3\}$	$\{q_0, q_1, q_4\}$	$\{q_0, q_3\}$
	$\{q_0, q_2, q_3\}$	$\{q_0, q_1, q_4\}$	$\{q_0, q_3\}$
	$\{q_0, q_1, q_4\}$		

Exercice 1

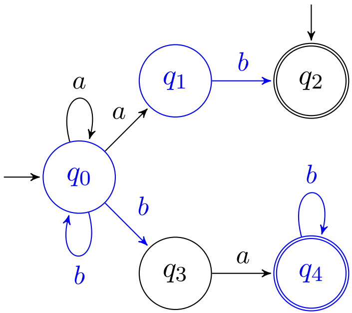
Construire un AFD (complet) équivalent à :



	$\mathcal{P}(Q)$	a	b
$\mathcal{I}$	$\{q_0, q_2\}$	$\{q_0, q_1\}$	$\{q_0, q_3\}$
	$\{q_0, q_1\}$	$\{q_0, q_1\}$	$\{q_0, q_2, q_3\}$
	$\{q_0, q_3\}$	$\{q_0, q_1, q_4\}$	$\{q_0, q_3\}$
	$\{q_0, q_2, q_3\}$	$\{q_0, q_1, q_4\}$	$\{q_0, q_3\}$
	$\{q_0, q_1, q_4\}$	$\{q_0, q_1\}$	

Exercice 1

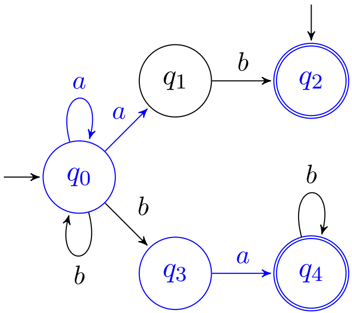
Construire un AFD (complet) équivalent à :



	$\mathcal{P}(Q)$	a	b
$\mathcal{I}$	$\{q_0, q_2\}$	$\{q_0, q_1\}$	$\{q_0, q_3\}$
	$\{q_0, q_1\}$	$\{q_0, q_1\}$	$\{q_0, q_2, q_3\}$
	$\{q_0, q_3\}$	$\{q_0, q_1, q_4\}$	$\{q_0, q_3\}$
	$\{q_0, q_2, q_3\}$	$\{q_0, q_1, q_4\}$	$\{q_0, q_3\}$
	$\{q_0, q_1, q_4\}$	$\{q_0, q_1\}$	$\{q_0, q_2, q_3, q_4\}$
	$\{q_0, q_2, q_3, q_4\}$		

Exercice 1

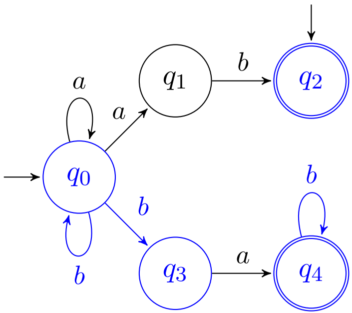
Construire un AFD (complet) équivalent à :



	$\mathcal{P}(Q)$	a	b
$\mathcal{I}$	$\{q_0, q_2\}$	$\{q_0, q_1\}$	$\{q_0, q_3\}$
	$\{q_0, q_1\}$	$\{q_0, q_1\}$	$\{q_0, q_2, q_3\}$
	$\{q_0, q_3\}$	$\{q_0, q_1, q_4\}$	$\{q_0, q_3\}$
	$\{q_0, q_2, q_3\}$	$\{q_0, q_1, q_4\}$	$\{q_0, q_3\}$
	$\{q_0, q_1, q_4\}$	$\{q_0, q_1\}$	$\{q_0, q_2, q_3, q_4\}$
	$\{q_0, q_2, q_3, q_4\}$	$\{q_0, q_1, q_4\}$	

Exercice 1

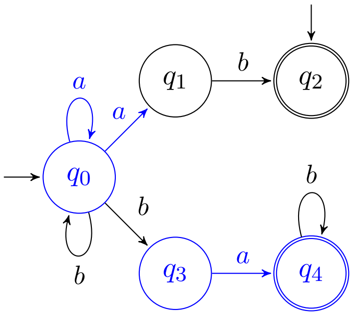
Construire un AFD (complet) équivalent à :



	$\mathcal{P}(Q)$	a	b
$\mathcal{I}$	$\{q_0, q_2\}$	$\{q_0, q_1\}$	$\{q_0, q_3\}$
	$\{q_0, q_1\}$	$\{q_0, q_1\}$	$\{q_0, q_2, q_3\}$
	$\{q_0, q_3\}$	$\{q_0, q_1, q_4\}$	$\{q_0, q_3\}$
	$\{q_0, q_2, q_3\}$	$\{q_0, q_1, q_4\}$	$\{q_0, q_3\}$
	$\{q_0, q_1, q_4\}$	$\{q_0, q_1\}$	$\{q_0, q_2, q_3, q_4\}$
	$\{q_0, q_2, q_3, q_4\}$	$\{q_0, q_1, q_4\}$	$\{q_0, q_3, q_4\}$
	$\{q_0, q_3, q_4\}$		

Exercice 1

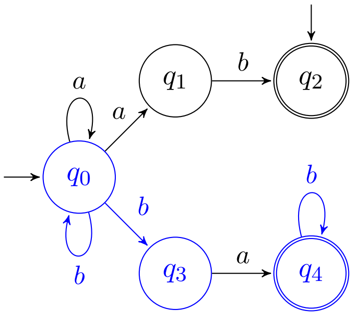
Construire un AFD (complet) équivalent à :



	$\mathcal{P}(Q)$	a	b
$\mathcal{I}$	$\{q_0, q_2\}$	$\{q_0, q_1\}$	$\{q_0, q_3\}$
	$\{q_0, q_1\}$	$\{q_0, q_1\}$	$\{q_0, q_2, q_3\}$
	$\{q_0, q_3\}$	$\{q_0, q_1, q_4\}$	$\{q_0, q_3\}$
	$\{q_0, q_2, q_3\}$	$\{q_0, q_1, q_4\}$	$\{q_0, q_3\}$
	$\{q_0, q_1, q_4\}$	$\{q_0, q_1\}$	$\{q_0, q_2, q_3, q_4\}$
	$\{q_0, q_2, q_3, q_4\}$	$\{q_0, q_1, q_4\}$	$\{q_0, q_3, q_4\}$
	$\{q_0, q_3, q_4\}$	$\{q_0, q_1, q_4\}$	

Exercice 1

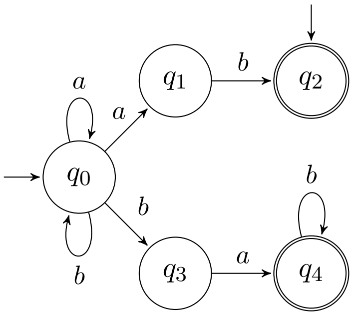
Construire un AFD (complet) équivalent à :



	$\mathcal{P}(Q)$	a	b
$\mathcal{I}$	$\{q_0, q_2\}$	$\{q_0, q_1\}$	$\{q_0, q_3\}$
	$\{q_0, q_1\}$	$\{q_0, q_1\}$	$\{q_0, q_2, q_3\}$
	$\{q_0, q_3\}$	$\{q_0, q_1, q_4\}$	$\{q_0, q_3\}$
	$\{q_0, q_2, q_3\}$	$\{q_0, q_1, q_4\}$	$\{q_0, q_3\}$
	$\{q_0, q_1, q_4\}$	$\{q_0, q_1\}$	$\{q_0, q_2, q_3, q_4\}$
	$\{q_0, q_2, q_3, q_4\}$	$\{q_0, q_1, q_4\}$	$\{q_0, q_3, q_4\}$
	$\{q_0, q_3, q_4\}$	$\{q_0, q_1, q_4\}$	$\{q_0, q_3, q_4\}$

Exercice 1

Construire un AFD (complet) équivalent à :

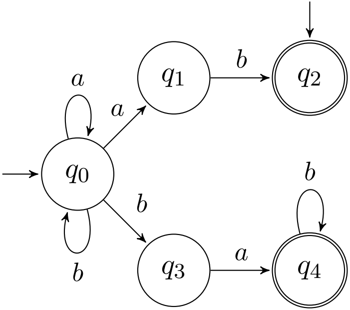


	$\mathcal{P}(Q)$	a	b
$\mathcal{I}$	$\{q_0, q_2\}$	$\{q_0, q_1\}$	$\{q_0, q_3\}$
	$\{q_0, q_1\}$	$\{q_0, q_1\}$	$\{q_0, q_2, q_3\}$
	$\{q_0, q_3\}$	$\{q_0, q_1, q_4\}$	$\{q_0, q_3\}$
	$\{q_0, q_2, q_3\}$	$\{q_0, q_1, q_4\}$	$\{q_0, q_3\}$
	$\{q_0, q_1, q_4\}$	$\{q_0, q_1\}$	$\{q_0, q_2, q_3, q_4\}$
	$\{q_0, q_2, q_3, q_4\}$	$\{q_0, q_1, q_4\}$	$\{q_0, q_3, q_4\}$
	$\{q_0, q_3, q_4\}$	$\{q_0, q_1, q_4\}$	$\{q_0, q_3, q_4\}$

Pas d'autre état accessible

Exercice 1

Construire un AFD (complet) équivalent à :



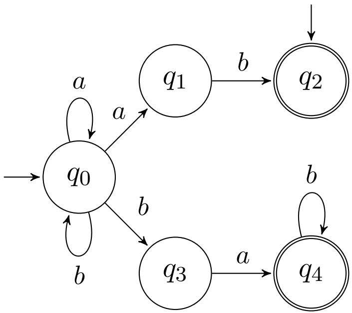
	$\mathcal{P}(Q)$	a	b
$\mathcal{I}$	$\{q_0, q_2\}$	$\{q_0, q_1\}$	$\{q_0, q_3\}$
	$\{q_0, q_1\}$	$\{q_0, q_1\}$	$\{q_0, q_2, q_3\}$
	$\{q_0, q_3\}$	$\{q_0, q_1, q_4\}$	$\{q_0, q_3\}$
	$\{q_0, q_2, q_3\}$	$\{q_0, q_1, q_4\}$	$\{q_0, q_3\}$
	$\{q_0, q_1, q_4\}$	$\{q_0, q_1\}$	$\{q_0, q_2, q_3, q_4\}$
	$\{q_0, q_2, q_3, q_4\}$	$\{q_0, q_1, q_4\}$	$\{q_0, q_3, q_4\}$
	$\{q_0, q_3, q_4\}$	$\{q_0, q_1, q_4\}$	$\{q_0, q_3, q_4\}$

Pas d'autre état accessible

7 états accessibles (sur 32 potentiels)

Exercice 1

Construire un AFD (complet) équivalent à :



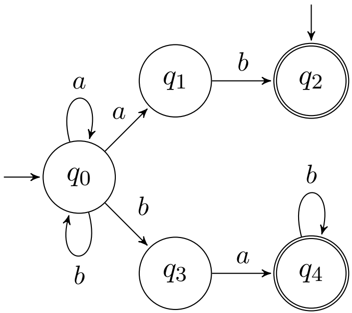
	$\mathcal{P}(Q)$	a	b
$\mathcal{I}$	$\{q_0, q_2\}$	$\{q_0, q_1\}$	$\{q_0, q_3\}$
	$\{q_0, q_1\}$	$\{q_0, q_1\}$	$\{q_0, q_2, q_3\}$
	$\{q_0, q_3\}$	$\{q_0, q_1, q_4\}$	$\{q_0, q_3\}$
	$\{q_0, q_2, q_3\}$	$\{q_0, q_1, q_4\}$	$\{q_0, q_3\}$
	$\{q_0, q_1, q_4\}$	$\{q_0, q_1\}$	$\{q_0, q_2, q_3, q_4\}$
	$\{q_0, q_2, q_3, q_4\}$	$\{q_0, q_1, q_4\}$	$\{q_0, q_3, q_4\}$
	$\{q_0, q_3, q_4\}$	$\{q_0, q_1, q_4\}$	$\{q_0, q_3, q_4\}$

Pas d'autre état accessible

7 états accessibles (sur 32 potentiels)

Exercice 1

Construire un AFD (complet) équivalent à :

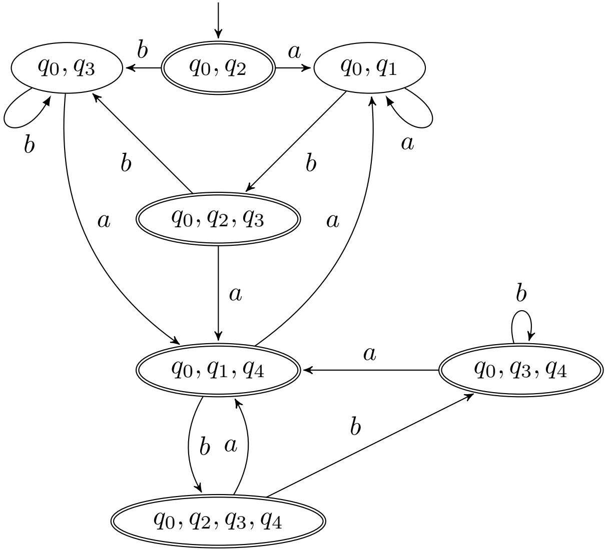


	$\mathcal{P}(Q)$	a	b
$\mathcal{I}$	$\{q_0, q_2\}$	$\{q_0, q_1\}$	$\{q_0, q_3\}$
	$\{q_0, q_1\}$	$\{q_0, q_1\}$	$\{q_0, q_2, q_3\}$
	$\{q_0, q_3\}$	$\{q_0, q_1, q_4\}$	$\{q_0, q_3\}$
	$\{q_0, q_2, q_3\}$	$\{q_0, q_1, q_4\}$	$\{q_0, q_3\}$
	$\{q_0, q_1, q_4\}$	$\{q_0, q_1\}$	$\{q_0, q_2, q_3, q_4\}$
	$\{q_0, q_2, q_3, q_4\}$	$\{q_0, q_1, q_4\}$	$\{q_0, q_3, q_4\}$
	$\{q_0, q_3, q_4\}$	$\{q_0, q_1, q_4\}$	$\{q_0, q_3, q_4\}$

Pas d'autre état accessible

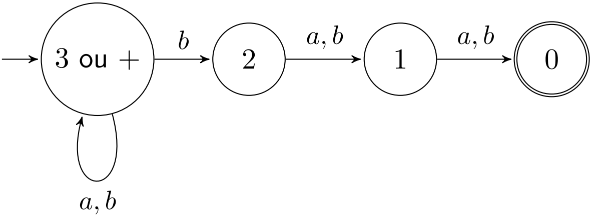
7 états accessibles (sur 32 potentiels)
5 états acceptants accessibles (sur 24 potentiels)

Solution



Exercice 2

On considère le langage
$L = \{w \in V^* \mid$ le 3e symbole en partant de la fin est un $b\}$.
Un automate non-déterministe qui le reconnaît est :



Construire un AFD (complet) équivalent.

Solution