Chapitre 5 : Automates

Option Informatique – MP

Lycée Chateaubriand

Table des matières

Plan

Introduction

On a défini des langages mais on aimerait avoir un moyen simple de déterminer si un mot appartient à langage donné (au moins dans le cas où le langage est un langage rationnel).

Nous allons voir la notion d'automates finis qui permet de faire cela. Le principe général est le suivant :

Introduction

On a défini des langages mais on aimerait avoir un moyen simple de déterminer si un mot appartient à langage donné (au moins dans le cas où le langage est un langage rationnel).

Nous allons voir la notion d'automates finis qui permet de faire cela. Le principe général est le suivant :

- Un automate possède un ou plusieurs états

Introduction

On a défini des langages mais on aimerait avoir un moyen simple de déterminer si un mot appartient à langage donné (au moins dans le cas où le langage est un langage rationnel).

Nous allons voir la notion d'automates finis qui permet de faire cela. Le principe général est le suivant :

- Un automate possède un ou plusieurs états
- Chaque lettre de l'alphabet engendre une transition qui permet d'aller d'un état donné à un autre état (ou des autres états voir plus loin)

Introduction

On a défini des langages mais on aimerait avoir un moyen simple de déterminer si un mot appartient à langage donné (au moins dans le cas où le langage est un langage rationnel).

Nous allons voir la notion d'automates finis qui permet de faire cela. Le principe général est le suivant :

- Un automate possède un ou plusieurs états
- Chaque lettre de l'alphabet engendre une transition qui permet d'aller d'un état donné à un autre état (ou des autres états voir plus loin)
- Il y a des états initiaux et des états finals.

Automates finis déterministes

Définition 1 (Automate fini déterministe)

Un automate (fini) déterministe ou AFD sur l'alphabet $\mathscr{A}$ est un quadruplet $A = (Q, q_0, F, \delta)$ où

Définition 1 (Automate fini déterministe)

Un automate (fini) déterministe ou AFD sur l'alphabet $\mathscr{A}$ est un quadruplet $A = (Q, q_0, F, \delta)$ où

- *L'ensemble Q est un ensemble fini. C'est l'ensemble des états de l'automate*

Définition 1 (Automate fini déterministe)

Un automate (fini) déterministe ou AFD sur l'alphabet $\mathscr{A}$ est un quadruplet $A = (Q, q_0, F, \delta)$ où

- *L'ensemble Q est un ensemble fini. C'est l'ensemble des états de l'automate*
- *L'élément q_0 est un élément de Q. Il s'appelle l'état initial*

Automates finis déterministes

Définition 1 (Automate fini déterministe)

Un automate (fini) déterministe ou AFD sur l'alphabet $\mathscr{A}$ est un quadruplet $A = (Q, q_0, F, \delta)$ où

- *L'ensemble Q est un ensemble fini. C'est l'ensemble des états de l'automate*
- *L'élément q_0 est un élément de Q. Il s'appelle l'état initial*
- *L'ensemble F est une partie de Q. Ses éléments s'appellent les états finals (ou acceptants)*

Automates finis déterministes

Définition 1 (Automate fini déterministe)

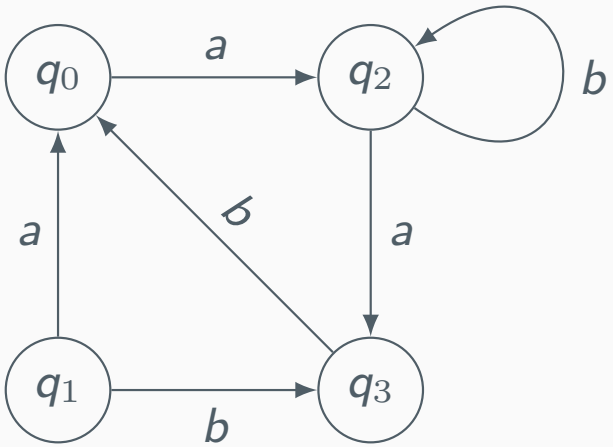
Un automate (fini) déterministe ou AFD sur l'alphabet $\mathscr{A}$ est un quadruplet $A = (Q, q_0, F, \delta)$ où

- *L'ensemble Q est un ensemble fini. C'est l'ensemble des états de l'automate*
- *L'élément q_0 est un élément de Q. Il s'appelle l'état initial*
- *L'ensemble F est une partie de Q. Ses éléments s'appellent les états finals (ou acceptants)*
- *L'application δ est définie sur une partie de $Q \times \mathscr{A}$ et à valeurs dans Q. C'est la fonction de transition.*

On veut définir un automate sur $\mathscr{A} = \{a, b\}$. On se donne $Q = \{q_0, q_1, q_2, q_3\}$ avec q_0 l'état initial, $F = \{q_2\}$ et δ donnée par

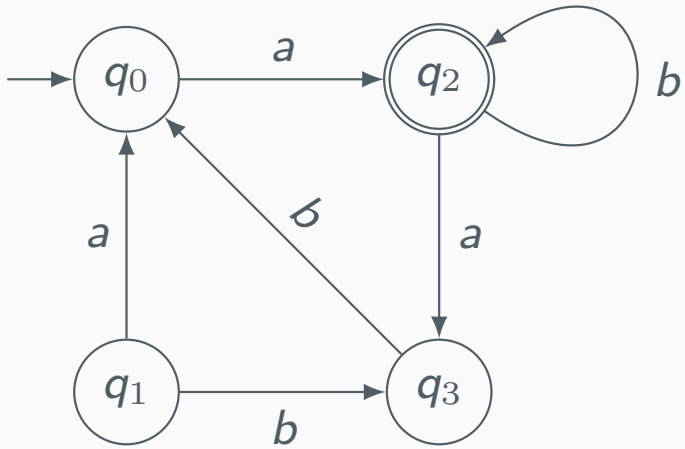
	q_0	q_1	q_2	q_3
a	q_2	q_0	q_3	
b		q_3	q_2	q_0

Afin de mieux visualiser l'automate on représente les états et les transitions ainsi :



On indique l'état initial par une flèche entrante

On désigne les états acceptants par un double cercle. On trouve aussi une flèche sortante vers rien.



Exemple

On peut par exemple penser au monnayeur d'une machine :

Exemple

On peut par exemple penser au monnayeur d'une machine :

- Les états désignent alors le montant inséré.

Exemple

On peut par exemple penser au monnayeur d'une machine :

- Les états désignent alors le montant inséré.
- L'état initial est l'état qui représente la somme nulle.

Exemple

On peut par exemple penser au monnayeur d'une machine :

- Les états désignent alors le montant inséré.
- L'état initial est l'état qui représente la somme nulle.
- Les transitions sont calculées en ajoutant à chaque étape l'argent ajouté à la somme déjà dans la machine.

Exemple

On peut par exemple penser au monnayeur d'une machine :

- Les états désignent alors le montant inséré.
- L'état initial est l'état qui représente la somme nulle.
- Les transitions sont calculées en ajoutant à chaque étape l'argent ajouté à la somme déjà dans la machine.
- Les états finals sont ceux où la somme est supérieure à la somme voulue par la machine pour délivrer le bien.

Terminologie

Soit (Q, q_0, F, δ) un automate fini déterministe :

Terminologie

Soit (Q, q_0, F, δ) un automate fini déterministe :

- Un blocage de l'automate est un couple $(q, x) \in Q \times \mathscr{A}$ tel que $\delta(q, x)$ n'est pas défini.

Terminologie

Soit (Q, q_0, F, δ) un automate fini déterministe :

- Un blocage de l'automate est un couple $(q, x) \in Q \times \mathscr{A}$ tel que $\delta(q, x)$ n'est pas défini.
- Un automate sans blocages est dit complet.

Terminologie

Soit (Q, q_0, F, δ) un automate fini déterministe :

- Un blocage de l'automate est un couple $(q, x) \in Q \times \mathscr{A}$ tel que $\delta(q, x)$ n'est pas défini.
- Un automate sans blocages est dit complet.
- L'automate est dit standard si pour tout $(q, x) \in Q \times \mathscr{A}$, $\delta(q, x) \neq q_0$. C'est-à-dire que l'automate ne « retombe » pas dans l'état initial.

Terminologie

Soit (Q, q_0, F, δ) un automate fini déterministe :

- Un blocage de l'automate est un couple $(q, x) \in Q \times \mathscr{A}$ tel que $\delta(q, x)$ n'est pas défini.
- Un automate sans blocages est dit complet.
- L'automate est dit standard si pour tout $(q, x) \in Q \times \mathscr{A}$, $\delta(q, x) \neq q_0$. C'est-à-dire que l'automate ne « retombe » pas dans l'état initial. Nous verrons plus loin l'intérêt des automates standards.

Calcul dans un automate

Nous allons maintenant « calculer » dans l'automate. Cela consiste à lire un mot de $\mathscr{A}^\star$ en parcourant en conséquence les différents états de l'automates.

Pour cela on va étendre la fonction de transition à des mots de $\mathscr{A}^\star$ en faisant évoluer l'état de l'automate à chaque lettre du mot.

Calcul dans un automate

Définition 2

Soit $A = (Q, q_0, F, \delta)$ un automate fini déterministe sur $\mathscr{A}$ on définit une application de transition notée $\delta^\star : Q \times \mathscr{A}^\star \to Q$ par :

- $\forall q \in Q, \delta^\star(q, \varepsilon) = q$

Calcul dans un automate

Définition 2

Soit $A = (Q, q_0, F, \delta)$ un automate fini déterministe sur $\mathscr{A}$ on définit une application de transition notée $\delta^\star : Q \times \mathscr{A}^\star \to Q$ par :

- $\forall q \in Q, \delta^\star(q, \varepsilon) = q$
- $\forall a \in \mathscr{A}, \forall w \in \mathscr{A}^\star, \delta(q, aw) = \delta^\star(\delta(q, a), w)$

Remarques

1. Il arrive que l'on note aussi simplement δ pour $\delta^\star$.

Remarques

1. Il arrive que l'on note aussi simplement δ pour $\delta^\star$.
2. Si l'automate est complet, la fonction $\delta^\star$ est définie pour tout $q \in Q$ et tout $w \in \mathscr{A}^\star$.

Remarques

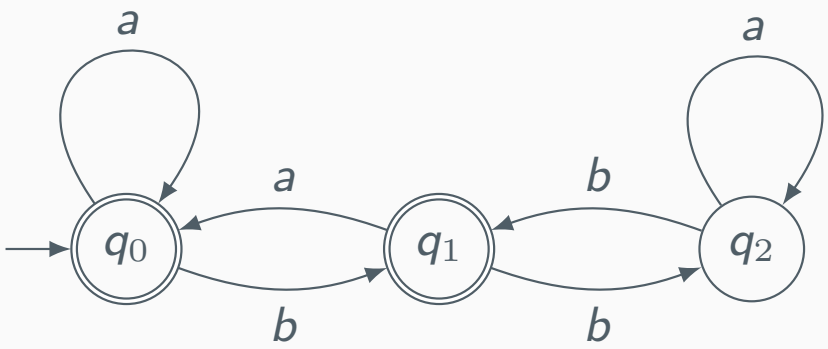
1. Il arrive que l'on note aussi simplement δ pour $\delta^\star$.
2. Si l'automate est complet, la fonction $\delta^\star$ est définie pour tout $q \in Q$ et tout $w \in \mathscr{A}^\star$.
3. Pour $w \in \mathscr{A}^\star$, on appelle donc calcul dans l'automate sur le mot $w = w_0 w_1 \cdots w_{n-1}$ issu de l'état q un « chemin »dans l'automate :

$$q = \alpha_0 \xrightarrow{w_0} \alpha_1 \xrightarrow{w_1} \cdots \xrightarrow{w_{n-1}} \alpha_n$$

où pour tout $i \in [\![0, n-1]\!]$, $\delta(\alpha_i, w_i) = w_{i+1}$. On a alors $\delta^\star(q, w) = \alpha_n$.

Exemple

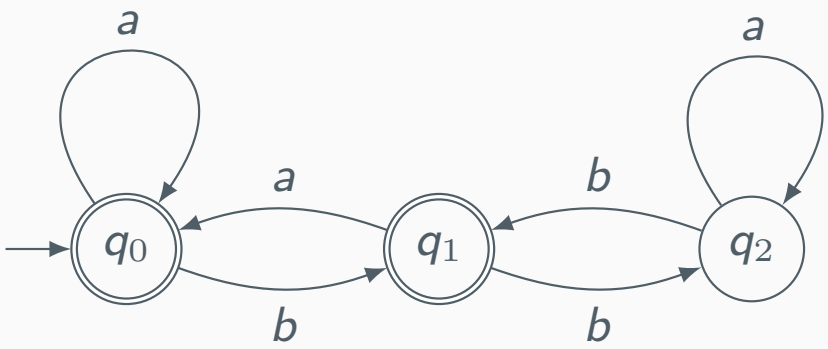
Considérons l'automate complet suivant



- $\delta^\star(q_0, abba) = q_2$
- $\delta^\star(q_1, aabb) =$

Exemple

Considérons l'automate complet suivant



- $\delta^\star(q_0, abba) = q_2$
- $\delta^\star(q_1, aabb) = q_2$

Langage reconnu par un automate

Définition 3 (Langage reconnu par un automate)

Soit $A = (Q, q_0, F, \delta)$ un automate fini déterministe.

Langage reconnu par un automate

Définition 3 (Langage reconnu par un automate)

Soit $A = (Q, q_0, F, \delta)$ un automate fini déterministe.

1. *Un mot w sur $\mathscr{A}$ est dit reconnu par l'automate si $\delta^\star(q_0, w)$ est défini et s'il appartient à F.*

Langage reconnu par un automate

Définition 3 (Langage reconnu par un automate)

Soit $A = (Q, q_0, F, \delta)$ un automate fini déterministe.

1. *Un mot w sur $\mathscr{A}$ est dit reconnu par l'automate si $\delta^\star(q_0, w)$ est défini et s'il appartient à F.*
2. *L'ensemble des mots reconnus par l'automate A se note $L(A)$. C'est le langage reconnu par l'automate.*

Remarques

- Le but d'un automate est de reconnaitre des mots.

Remarques

- Le but d'un automate est de reconnaitre des mots.
- On dit aussi que les mots reconnus sont les mots acceptés.

Remarques

- Le but d'un automate est de reconnaitre des mots.
- On dit aussi que les mots reconnus sont les mots acceptés.
- Un mot $w \in \mathscr{A}^\star$ **n'est pas reconnu** par A si

Remarques

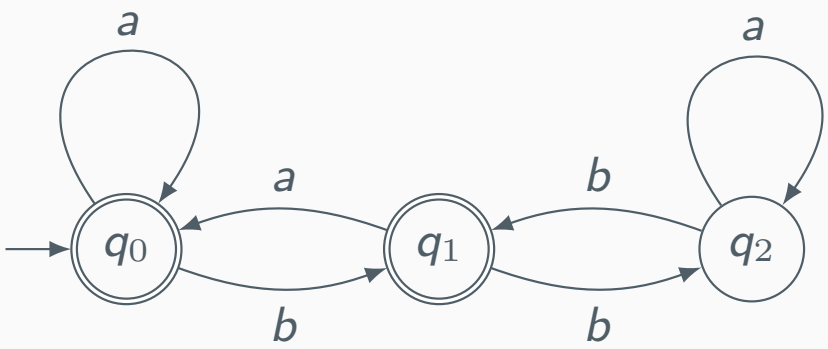
- Le but d'un automate est de reconnaitre des mots.
- On dit aussi que les mots reconnus sont les mots acceptés.
- Un mot $w \in \mathscr{A}^\star$ **n'est pas reconnu** par A si
  - La lecture du mot par l'automate engendre un blocage, c'est-à-dire qu'il existe un préfixe u tel que $w = uav$ et $(\delta^\star(q_0, u), a)$ est un blocage de A.

Remarques

- Le but d'un automate est de reconnaitre des mots.
- On dit aussi que les mots reconnus sont les mots acceptés.
- Un mot $w \in \mathscr{A}^\star$ **n'est pas reconnu** par A si
  - La lecture du mot par l'automate engendre un blocage, c'est-à-dire qu'il existe un préfixe u tel que $w = uav$ et $(\delta^\star(q_0, u), a)$ est un blocage de A.
  - La lecture du mot peut se faire mais $\delta^\star(q_0, w) \notin F$.

Exemple

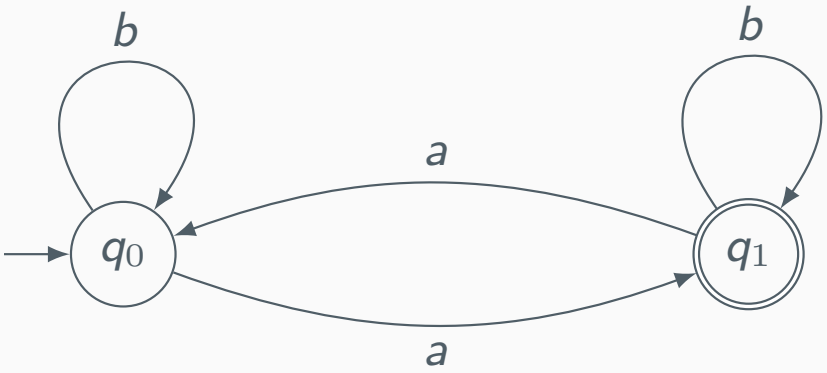
Si on reprend l'automate précédent



- On voit que *a*, *b*, *aab* sont reconnus
- Le mot *abba* n'est pas reconnu.

Exercice 1

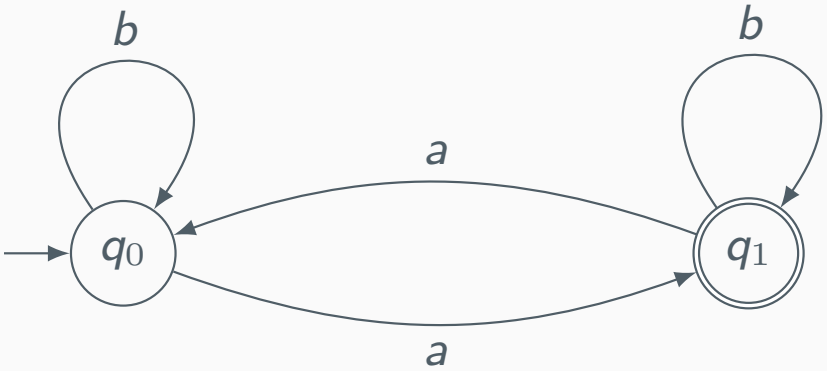
Soit A l'automate



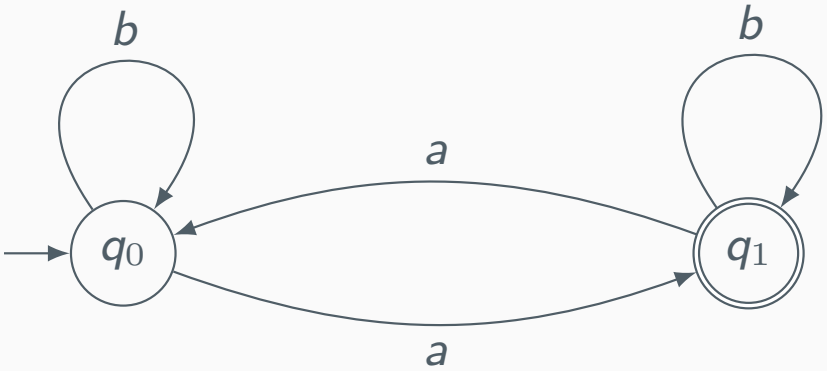
Déterminer des mots reconnus par A et des mots qui ne sont pas reconnus.

Quel est le langage reconnu par l'automate ?

Exercice 1

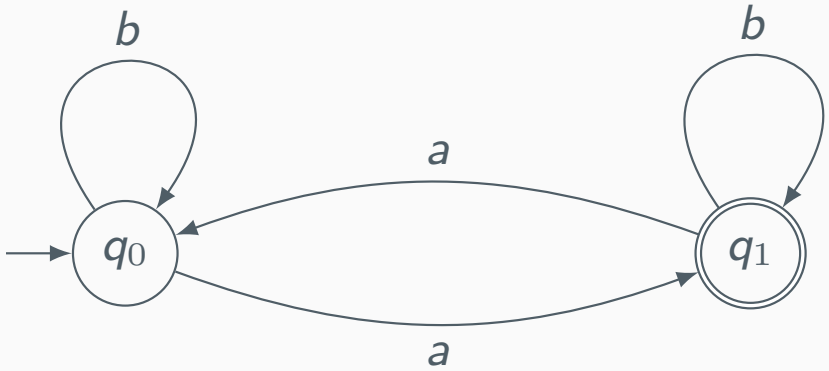


Exercice 1



- Le mot a est reconnu alors que aa ne l'est pas

Exercice 1



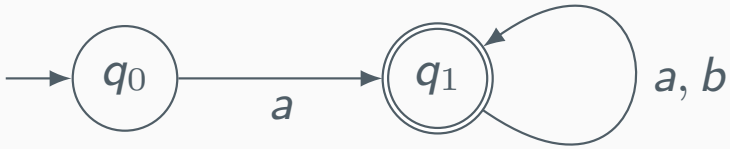
- Le mot a est reconnu alors que aa ne l'est pas
- L'automate reconnait les mots qui comportent un nombre impair de a. De fait, l'état q_0 correspond à « on a lu un nombre pair de a » et q_1 correspond à « on a lu un nombre pair de a ».

Exercice 2

Déterminer un automate dont le langage reconnu est $a\mathscr{A}^\star$.

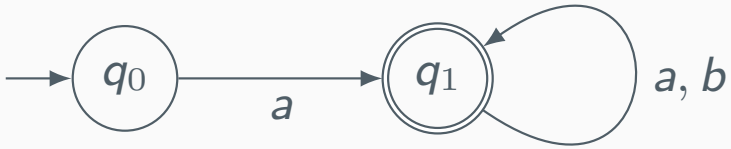
Exercice 2

Déterminer un automate dont le langage reconnu est $a\mathscr{A}^\star$.

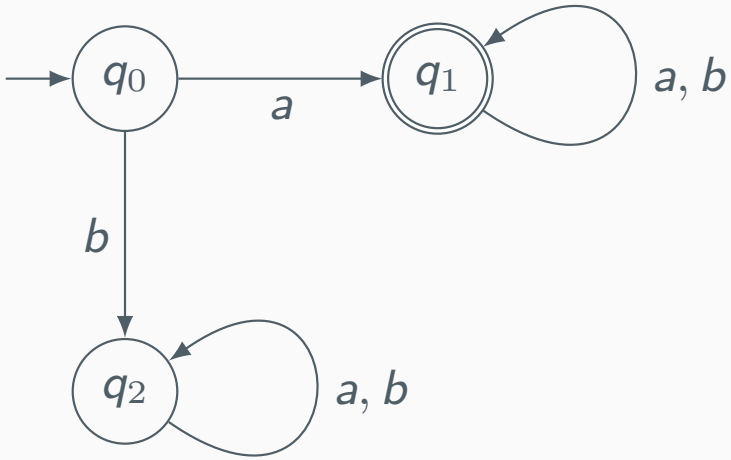


Exercice 2

Déterminer un automate dont le langage reconnu est $a\mathscr{A}^\star$.



ou



Complétion d'un automate

Théorème 1

Tout langage reconnu par un automate fini déterministe est reconnu par un automate fini déterministe complet

Complétion d'un automate

Théorème 1

Tout langage reconnu par un automate fini déterministe est reconnu par un automate fini déterministe complet

Démonstration : Soit L un langage et $A = (Q, q_0, \delta, F)$ un automate qui le reconnait. On peut construire un automate complet A' tel que $L(A') = L(A) = L$. Il suffit d'ajouter à Q un nouvel état p que l'on appelle un état puit. On pose $A' = (Q \cup \{p\}, q_0, \delta', F)$ où δ' est définie par :

Complétion d'un automate

Théorème 1

Tout langage reconnu par un automate fini déterministe est reconnu par un automate fini déterministe complet

Démonstration : Soit L un langage et $A = (Q, q_0, \delta, F)$ un automate qui le reconnait. On peut construire un automate complet A' tel que $L(A') = L(A) = L$. Il suffit d'ajouter à Q un nouvel état p que l'on appelle un état puit. On pose $A' = (Q \cup \{p\}, q_0, \delta', F)$ où δ' est définie par :

- Pour $q \in Q$ et $x \in \mathscr{A}$ tel que (q, x) n'est pas un blocage $\delta'(q, x) = \delta(q, x)$.

Complétion d'un automate

Théorème 1

Tout langage reconnu par un automate fini déterministe est reconnu par un automate fini déterministe complet

Démonstration : Soit L un langage et $A = (Q, q_0, \delta, F)$ un automate qui le reconnait. On peut construire un automate complet A' tel que $L(A') = L(A) = L$. Il suffit d'ajouter à Q un nouvel état p que l'on appelle un état puit. On pose $A' = (Q \cup \{p\}, q_0, \delta', F)$ où δ' est définie par :

- Pour $q \in Q$ et $x \in \mathscr{A}$ tel que (q, x) n'est pas un blocage $\delta'(q, x) = \delta(q, x)$.
- Pour $q \in Q$ et $x \in \mathscr{A}$ tel que (q, x) est un blocage $\delta'(q, x) = p$.

Complétion d'un automate

Théorème 1

Tout langage reconnu par un automate fini déterministe est reconnu par un automate fini déterministe complet

Démonstration : Soit L un langage et $A = (Q, q_0, \delta, F)$ un automate qui le reconnait. On peut construire un automate complet A' tel que $L(A') = L(A) = L$. Il suffit d'ajouter à Q un nouvel état p que l'on appelle un état puit. On pose $A' = (Q \cup \{p\}, q_0, \delta', F)$ où δ' est définie par :

- Pour $q \in Q$ et $x \in \mathscr{A}$ tel que (q, x) n'est pas un blocage $\delta'(q, x) = \delta(q, x)$.
- Pour $q \in Q$ et $x \in \mathscr{A}$ tel que (q, x) est un blocage $\delta'(q, x) = p$.
- Pour $x \in \mathscr{A}$, on pose $\delta'(p, x) = p$.

Complétion d'un automate

Prouvons alors proprement que $L(A) = L(A')$.

- Soit $w = w_0 w_1 \dots w_n \in L(A)$. Il existe un calcul dans A

$$q_0 \xrightarrow{w_0} \cdots \xrightarrow{w_n} q$$

  où $q \in F$. Ce calcul est aussi possible dans A' donc $w \in L(A')$. Finalement $\boxed{L(A) \subset L(A')}$.

- Soit $w \notin L(A)$. Il y a deux cas :
  - Le calcul dans A est impossible à cause d'un blocage. Dans ce cas, $(\delta')^\star(q_0, w) = p \notin F$.
  - Le calcul est possible dans A mais $\delta(q_0, w) \notin F$. Dans ce cas, le calcul est identique dans A' et donc on a encore $(\delta')^\star(q_0, w) \notin F$.

  Dans les deux cas $w \notin L(A')$. Finalement, par contraposée, $\boxed{L(A') \subset L(A)}$.

On a bien construit A' complet qui reconnait L.

Langage reconnaissable

Définition 4

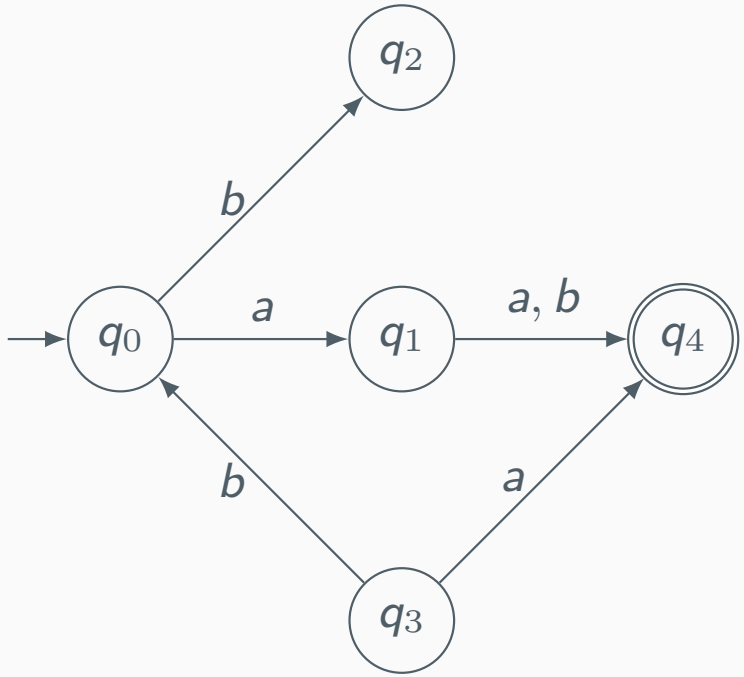
Un langage L est dit reconnaissable s'il existe un automate fini déterministe (que l'on peut supposer complet) A tel que $L = L(A)$.

Etats accessibles et co-accessibles

Quand on considère un automate A qui reconnait un langage L, on ne s'intéresse qu'aux états que l'on va « atteindre » à partir de l'état initial lors d'un calcul.

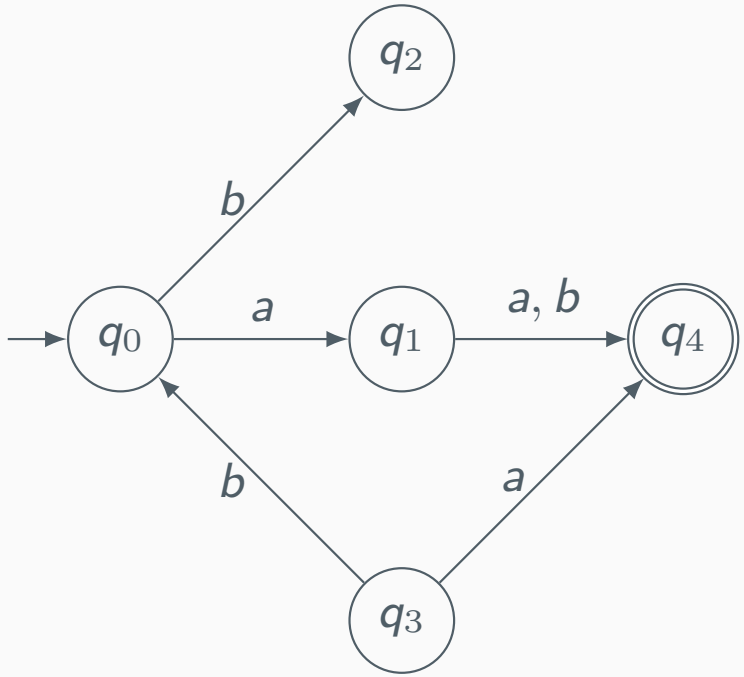
Etats accessibles et co-accessibles

Par exemple dans l'automate



Etats accessibles et co-accessibles

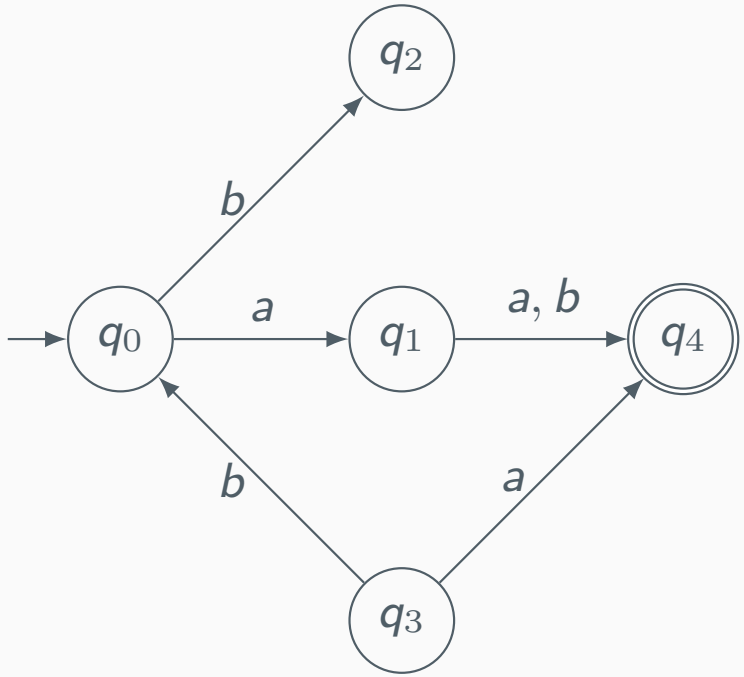
Par exemple dans l'automate



- On ne peut pas atteindre l'état q_3 en partant de q_0

Etats accessibles et co-accessibles

Par exemple dans l'automate



- On ne peut pas atteindre l'état q_3 en partant de q_0
- Si on est à q_2 on ne peut plus rejoindre l'état final q_4.

Etats accessibles et co-accessibles

Définition 5

Soit $A = (Q, q_0, F, \delta)$ un automate fini deterministe.

Etats accessibles et co-accessibles

Définition 5

Soit $A = (Q, q_0, F, \delta)$ un automate fini deterministe.

- *Un état $q \in Q$ est dit **accessible** s'il existe un mot $w \in \mathscr{A}^\star$ tel que $\delta(q_0, w) = q$. Autrement dit, si on peut atteindre le sommet q en partant du sommet initial.*

Etats accessibles et co-accessibles

Définition 5

Soit $A = (Q, q_0, F, \delta)$ un automate fini deterministe.

- *Un état $q \in Q$ est dit **accessible** s'il existe un mot $w \in \mathscr{A}^\star$ tel que $\delta(q_0, w) = q$. Autrement dit, si on peut atteindre le sommet q en partant du sommet initial.*
- *Un état $q \in Q$ est dit **co-accesible** s'il existe un mot $w \in \mathscr{A}^\star$ tel que $\delta(q, w) \in F$. Autrement dit, si on peut atteindre un sommet acceptant en partant du sommet q.*

L'automate est dit émondé si tous ses états sont accessibles et co-accessibles.

Automate émondé

À tout automate $A = (Q, q_0, F, \delta)$ reconnaissant un langage non vide (c'est-à-dire que q_0 est co-accessible) on lui associe l'automate émondé
$A' = (Q', q_0, F \cap Q', \delta')$ où :

Automate émondé

À tout automate $A = (Q, q_0, F, \delta)$ reconnaissant un langage non vide (c'est-à-dire que q_0 est co-accessible) on lui associe l'automate émondé
$A' = (Q', q_0, F \cap Q', \delta')$ où :

- Q' est l'ensemble des états accessibles et co-accesibles de A

Automate émondé

À tout automate $A = (Q, q_0, F, \delta)$ reconnaissant un langage non vide (c'est-à-dire que q_0 est co-accessible) on lui associe l'automate émondé
$A' = (Q', q_0, F \cap Q', \delta')$ où :

- Q' est l'ensemble des états accessibles et co-accesibles de A
- δ' est telle que pour tout $q \in Q'$ et $a \in \mathscr{A}$,

$$\delta'(q,a) = \begin{cases} \delta(q,a) & \text{si } \delta(q,a) \text{ existe et appartient à } Q' \\ \text{est un blocage} & \text{sinon.} \end{cases}$$

Automate émondé

Théorème 2

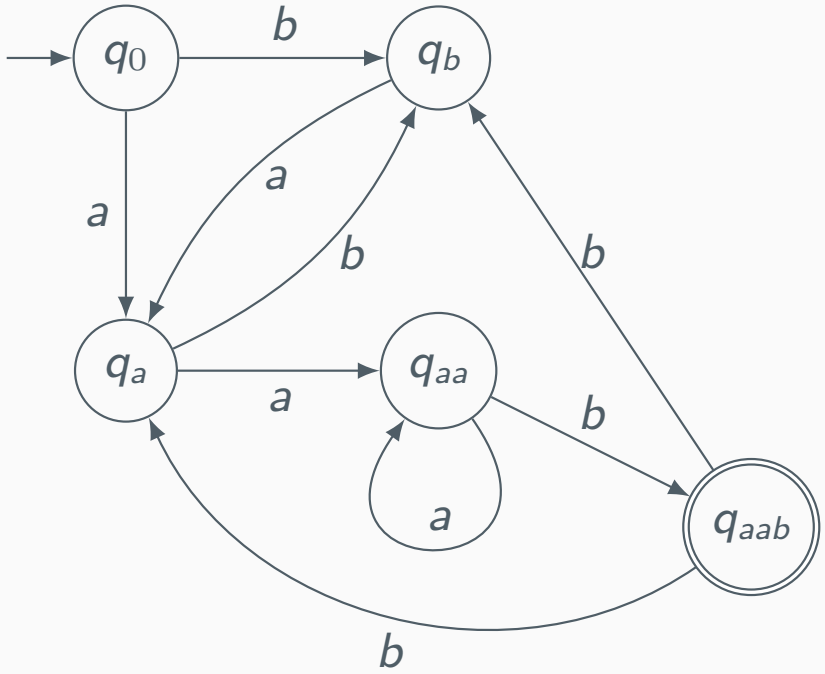
L'automate A et l'automate A' reconnaissent le même langage.

Introduction

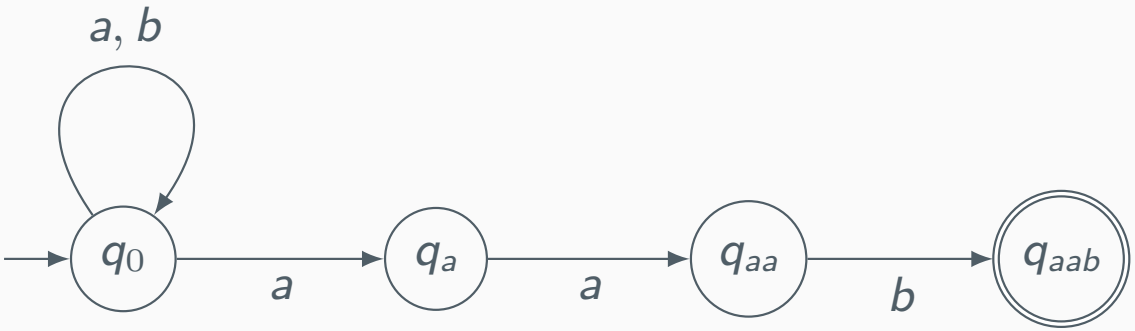
Dans certains cas, il va être plus facile de construire des automates d'un type différent. Commençons par un exemple.

On veut construire un automate reconnaissant le langage $L = \mathscr{A}^\star aab$ des mots qui se terminent par aab.

On peut le faire avec un automate déterministe :


q_0
q_b
q_a
q_{aa}
q_{aab}
b
a
a
b
b
a
b
a
b

On peut aussi le faire avec ce type d'automate



La différence est que là, en étant à l'état q_0 on peut aller « en lisant » la lettre a en q_0 ou en q_a. C'est en cela que cela va s'appeler un automate **non**-déterministe.

Un calcul d'un mot de $\mathscr{A}^\star aab$ « peut » aboutir à l'état acceptant.

Introduction

Dans les automates que nous avons vus, il y avait une **unique** transition à un état fixé et une lettre fixée. Nous allons regarder maintenant des automates non-déterministes où, d'un état donné et avec une lettre donnée on peut passer à plusieurs états.

Définition 6 (Automate fini non déterministe)

Un automate (fini) non déterministe ou AFND sur l'alphabet $\mathscr{A}$ est un quadruplet $A = (Q, I, F, \delta)$ où

- *L'ensemble Q est un ensemble fini. C'est l'ensemble des états de l'automate.*
- *L'ensemble I est une partie de Q. Ce sont les états initiaux*
- *L'ensemble F est une partie de Q. C'est l'ensemble des états finals (ou acceptants)*
- *L'application δ est définie sur $Q \times \mathscr{A}$ et à valeurs dans $\mathscr{P}(Q)$. C'est la fonction de transition.*

Remarques

- Il y a deux différences (qui ne sont qu'une seule) entre un automate déterministe et un automate non déterministe. La première est que dans ce dernier il peux y avoir plusieurs états initiaux. La deuxième est que, partant d'un état donné, une même lettre nous permet d'atteindre zéro, un ou plusieurs états donnés par une partie de Q.

Remarques

- Il y a deux différences (qui ne sont qu'une seule) entre un automate déterministe et un automate non déterministe. La première est que dans ce dernier il peux y avoir plusieurs états initiaux. La deuxième est que, partant d'un état donné, une même lettre nous permet d'atteindre zéro, un ou plusieurs états donnés par une partie de Q.
- Il n'y a plus de blocage au sens précédent, cependant, $\delta(q, x)$ peut être vide.

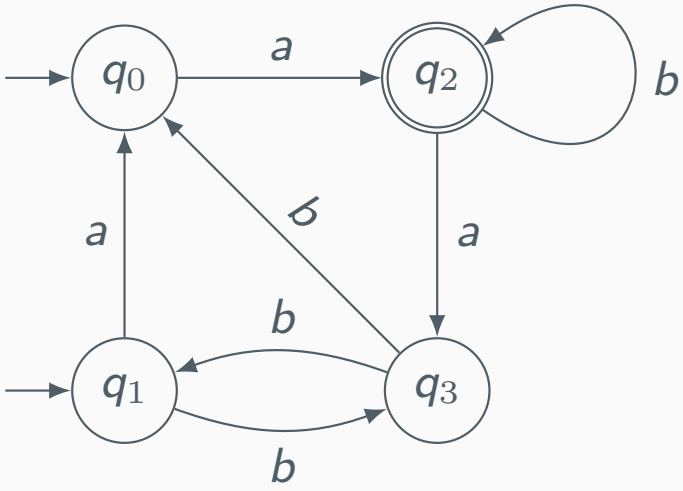
ATTENTION

Attention

La plupart du temps, un automate (sans précision) désigne un automate **non**-déterministe. **CEPENDANT** il est essentiel de bien lire les notations / définitions dans l'introduction d'un problème sur les automates.

Représentation

On les représente comme précédemment



Définition 7

Soit $A = (Q, I, F, \delta)$ un automate (fini) non déterministe.

1. *On étend la fonction δ à l'ensemble des mots de $\mathscr{A}^\star$ comme précédemment. Précisément on construit $\delta^\star : Q \times \mathscr{A}^\star \to \mathscr{P}(Q)$ par*
   - $\forall q \in Q, \delta^*(q, \varepsilon) = \{q\}$

Définition 7

Soit $A = (Q, I, F, \delta)$ un automate (fini) non déterministe.

1. *On étend la fonction δ à l'ensemble des mots de $\mathscr{A}^\star$ comme précédemment. Précisément on construit $\delta^\star : Q \times \mathscr{A}^\star \to \mathscr{P}(Q)$ par*
   - $\forall q \in Q, \delta^*(q, \varepsilon) = \{q\}$
   - $\forall a \in \mathscr{A}, \forall w \in \mathscr{A}^\star, \delta^\star(q, aw) = \bigcup_{q' \in \delta(q,a)} \delta^\star(q', w).$

Définition 7

Soit $A = (Q, I, F, \delta)$ un automate (fini) non déterministe.

1. *On étend la fonction δ à l'ensemble des mots de $\mathscr{A}^\star$ comme précédemment. Précisément on construit $\delta^\star : Q \times \mathscr{A}^\star \to \mathscr{P}(Q)$ par*
   - $\forall q \in Q, \delta^*(q, \varepsilon) = \{q\}$
   - $\forall a \in \mathscr{A}, \forall w \in \mathscr{A}^\star, \delta^\star(q, aw) = \bigcup_{q' \in \delta(q,a)} \delta^\star(q', w)$.
2. *Soit w un mot de $\mathscr{A}^\star$, il est reconnu par l'automate s'il existe $q_0 \in I$ tel que $\delta(q_0, w) \cap F \neq \emptyset$.*

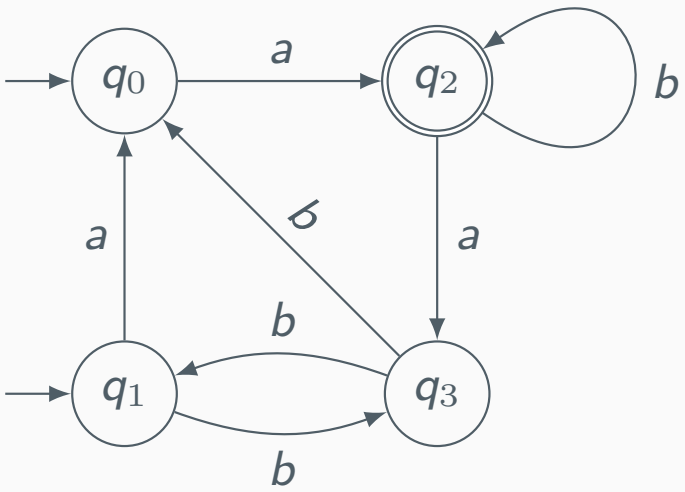
Attention

Attention

Un mot w est donc reconnu par A s'il existe **un** calcul dans A sur w qui aboutit à un état acceptant.

Exemple

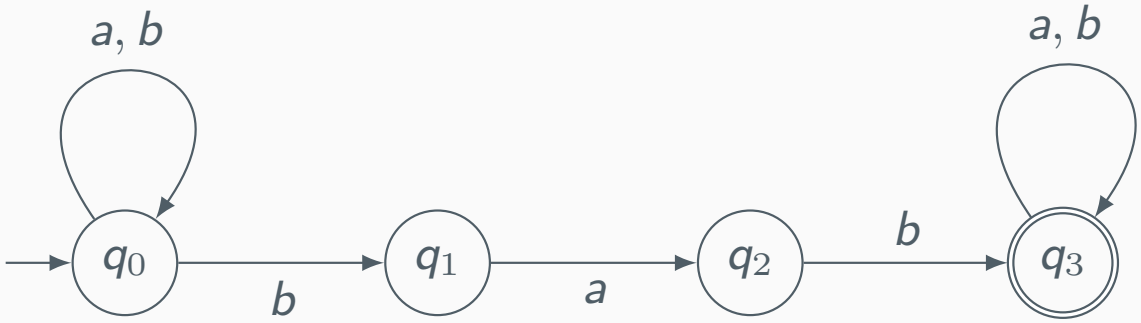
On reprend l'automate précédent



Les mots a, aa, bba sont reconnus. Par contre ba ne l'est pas.

Exercice 1

On considère l'automate suivant :



Déterminer des mots acceptés par l'automate et des mots qui ne sont pas acceptés. Quel est le langage des mots reconnus pas l'automate ?

Exercice 1

- Les mots *bab* et *ababab* sont reconnus

Exercice 1

- Les mots *bab* et *ababab* sont reconnus
- Le mot *abb* n'est pas reconnu

Exercice 1

- Les mots *bab* et *ababab* sont reconnus
- Le mot *abb* n'est pas reconnu
- Les mots reconnus sont les mots qui ont un facteur de la forme *bab*.

$$L(A) = L(\mathscr{A}^\star(bab)\mathscr{A}^\star).$$

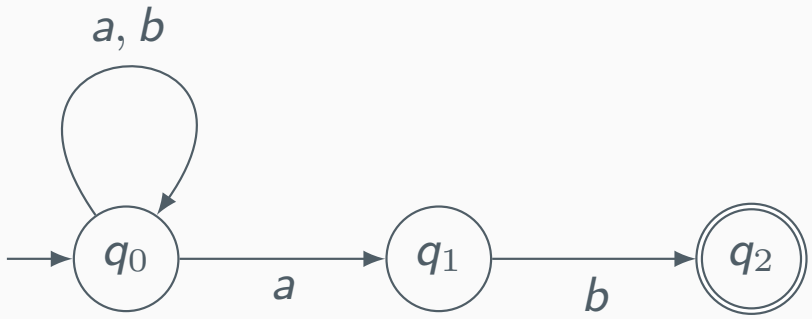
Exercice 2

Déterminer un automate (éventuellement non déterminé) reconnaissant les mots finissant par ab.

Exercice 2

Déterminer un automate (éventuellement non déterminé) reconnaissant les mots finissant par ab.

On peut prendre



Attention

Il est important dans les problèmes de complexité sur les automates de faire la différence entre :

- Le temps de **construire** l'automate (et la place qu'il occupe)
- Le temps pour **calculer** dans l'automate

Théorème 3

Soit A un automate.

1. On suppose que A est déterministe. L'appartenance d'un mot w à $L(A)$ à pour complexité :
   - Complexité temporelle :
   - Complexité en espace :

   Dans le cas général. L'appartenance d'un mot w à $L(A)$ à pour complexité :
   - Complexité temporelle :
   - Complexité en espace :

Théorème 3

Soit A un automate.

1. On suppose que A est déterministe. L'appartenance d'un mot w à $L(A)$ à pour complexité :
   - Complexité temporelle : $O(|Q| + |w|)$
   - Complexité en espace :

   Dans le cas général. L'appartenance d'un mot w à $L(A)$ à pour complexité :
   - Complexité temporelle :
   - Complexité en espace :

Théorème 3

Soit A un automate.

1. On suppose que A est déterministe. L'appartenance d'un mot w à $L(A)$ à pour complexité :
   - Complexité temporelle : $O(|Q| + |w|)$
   - Complexité en espace : $O(|Q|)$

   Dans le cas général. L'appartenance d'un mot w à $L(A)$ à pour complexité :
   - Complexité temporelle :
   - Complexité en espace :

Théorème 3

Soit A un automate.

1. On suppose que A est déterministe. L'appartenance d'un mot w à $L(A)$ à pour complexité :
   - Complexité temporelle : $O(|Q| + |w|)$
   - Complexité en espace : $O(|Q|)$

   Dans le cas général. L'appartenance d'un mot w à $L(A)$ à pour complexité :
   - Complexité temporelle : $O(|Q|.|w|)$
   - Complexité en espace :

Théorème 3

Soit A un automate.

1. On suppose que A est déterministe. L'appartenance d'un mot w à $L(A)$ à pour complexité :
   - Complexité temporelle : $O(|Q| + |w|)$
   - Complexité en espace : $O(|Q|)$

   Dans le cas général. L'appartenance d'un mot w à $L(A)$ à pour complexité :
   - Complexité temporelle : $O(|Q|.|w|)$
   - Complexité en espace : $O(|Q|)$

Démonstration

- Dans le cas déterministe, il faut charger l'automate ($O(|Q|)$) puis lire le mot dans l'automate. La lecture de chaque lettre est en temps constant donc l'algorithme est en $O(|Q| + |w|)$.
- Dans le cas général, calculer $\delta(X, x)$ où $X \subset Q$ et $x \in \mathscr{A}$ est en $O(|X|)$. On en déduit une complexité dans le pire des cas en $O(|Q|.|w|)$.

Déterminisation

Soit L un langage reconnu par un automate A non déterministe. On vient de voir qu'il est préférable d'avoir un automate déterministe reconnaissant ce langage. On va donc construire cet automate déterministe. L'idée principale est de remplacer l'ensemble Q des états par $\mathscr{P}(Q)$.

Automate des parties

Définition 8

Automate des parties Soit $A = (Q, I, F, \delta)$ *un automate non déterministe. On appelle automate des parties l'automate déterministe* $A' = (Q', I, F', \delta')$ *où :*

Automate des parties

Définition 8

Automate des parties Soit $A = (Q, I, F, \delta)$ *un automate non déterministe. On appelle automate des parties l'automate déterministe* $A' = (Q', I, F', \delta')$ *où :*

- $Q' = \mathscr{P}(Q)$.

Automate des parties

Définition 8

Automate des parties Soit $A = (Q, I, F, \delta)$ *un automate non déterministe. On appelle automate des parties l'automate déterministe* $A' = (Q', I, F', \delta')$ *où :*

- $Q' = \mathscr{P}(Q)$.
- $F' = \{X \in \mathscr{P}(Q) \mid X \cap F \neq \emptyset\}$

Automate des parties

Définition 8

Automate des parties Soit $A = (Q, I, F, \delta)$ *un automate non déterministe. On appelle automate des parties l'automate déterministe* $A' = (Q', I, F', \delta')$ *où :*

- $Q' = \mathscr{P}(Q)$.
- $F' = \{X \in \mathscr{P}(Q) \mid X \cap F \neq \emptyset\}$
- *L'application* δ' *définie par* $\forall X \in \mathscr{P}(Q), \forall x \in \mathscr{A},$

Définition 8

Automate des parties Soit $A = (Q, I, F, \delta)$ *un automate non déterministe. On appelle automate des parties l'automate déterministe* $A' = (Q', I, F', \delta')$ *où :*

- $Q' = \mathscr{P}(Q)$.
- $F' = \{X \in \mathscr{P}(Q) \mid X \cap F \neq \emptyset\}$
- *L'application* δ' *définie par* $\forall X \in \mathscr{P}(Q), \forall x \in \mathscr{A}$,

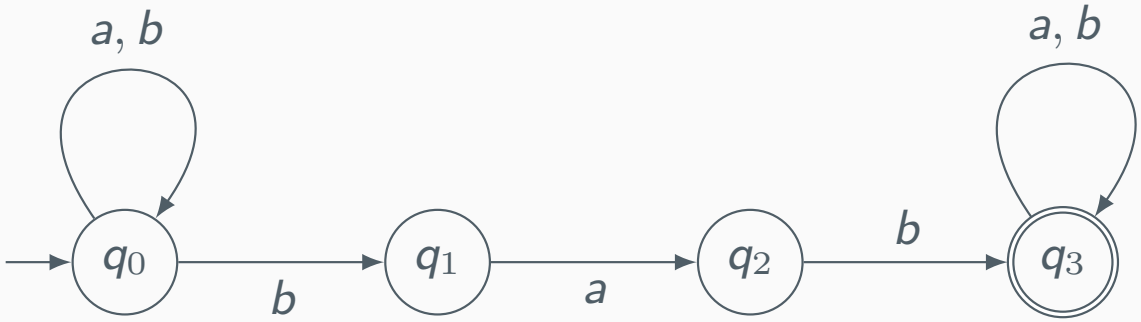
$$\delta'(X, x) = \bigcup_{q \in X} \delta(q, x).$$

Remarque

Notons qu'ici I est un **élément** de $\mathscr{P}(Q)$.

Exemple

Reprenons l'automate



Il a 4 états ce qui fait que l'automate des parties aura $2^4 = 16$ états. Cependant, de nombreux états ne seront pas accessibles

Exemple

En pratique :

Exemple

En pratique :

- On part de l'état initial I (qui est une partie de Q).

Exemple

En pratique :

- On part de l'état initial I (qui est une partie de Q).
- On considère tous les états que l'on obtient en partant de I pour tout lettre de l'alphabet

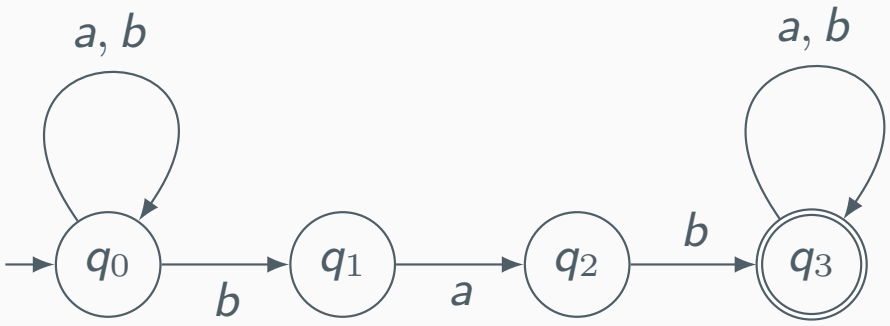
Exemple

En pratique :

- On part de l'état initial I (qui est une partie de Q).
- On considère tous les états que l'on obtient en partant de I pour tout lettre de l'alphabet
- Dès que l'on considère un nouvel état, on rajoute à la liste des états tous les états que l'on peut atteindre à partir de lui.

Exemple

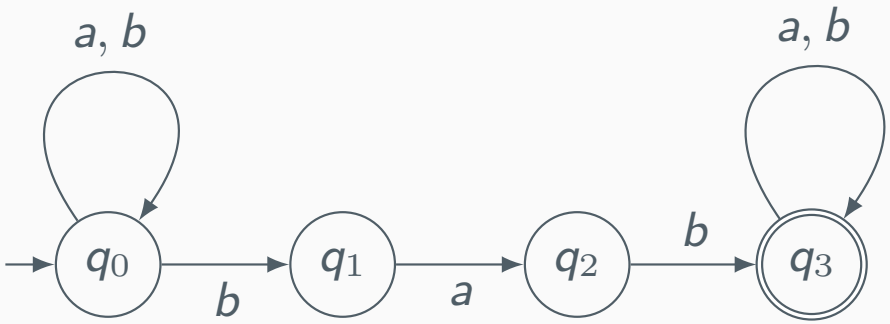
Sur l'automate



état	a	b
$\{q_0\}$	$\{q_0\}$	$\{q_0, q_1\}$
$\{q_0, q_1\}$		

Exemple

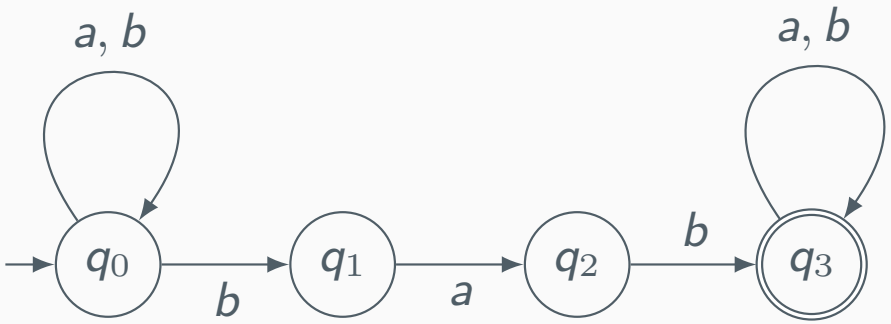
Sur l'automate



état	a	b
$\{q_0\}$	$\{q_0\}$	$\{q_0, q_1\}$
$\{q_0, q_1\}$	$\{q_0, q_2\}$	$\{q_0, q_1\}$
$\{q_0, q_2\}$		

Exemple

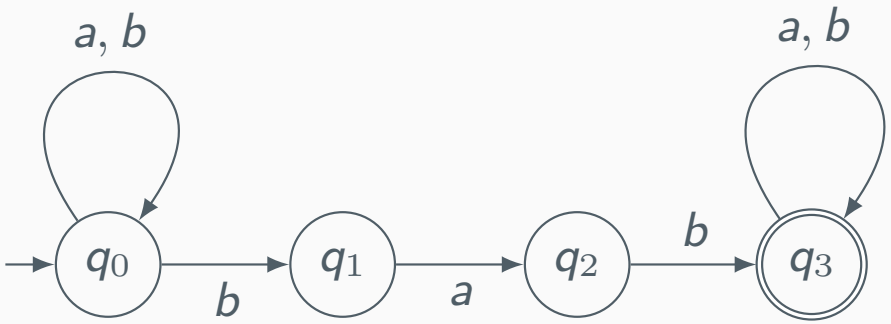
Sur l'automate



état	a	b
$\{q_0\}$	$\{q_0\}$	$\{q_0, q_1\}$
$\{q_0, q_1\}$	$\{q_0, q_2\}$	$\{q_0, q_1\}$
$\{q_0, q_2\}$	$\{q_0\}$	$\{q_0, q_1, q_3\}$
$\{q_0, q_1, q_3\}$		

Exemple

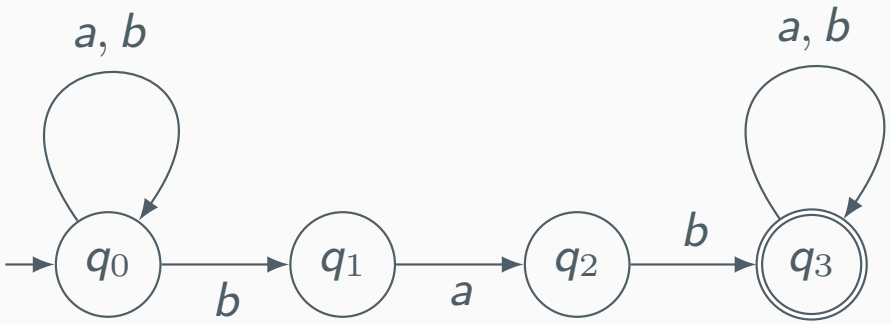
Sur l'automate



état	a	b
$\{q_0\}$	$\{q_0\}$	$\{q_0, q_1\}$
$\{q_0, q_1\}$	$\{q_0, q_2\}$	$\{q_0, q_1\}$
$\{q_0, q_2\}$	$\{q_0\}$	$\{q_0, q_1, q_3\}$
$\{q_0, q_1, q_3\}$	$\{q_0, q_2, q_3\}$	$\{q_0, q_1, q_3\}$
$\{q_0, q_2, q_3\}$		

Exemple

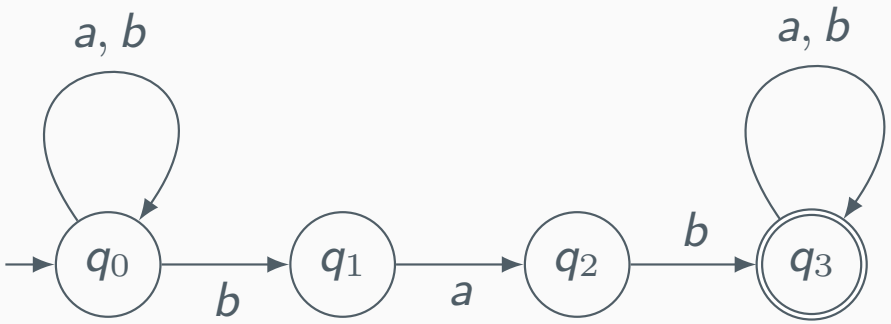
Sur l'automate



état	a	b
$\{q_0\}$	$\{q_0\}$	$\{q_0, q_1\}$
$\{q_0, q_1\}$	$\{q_0, q_2\}$	$\{q_0, q_1\}$
$\{q_0, q_2\}$	$\{q_0\}$	$\{q_0, q_1, q_3\}$
$\{q_0, q_1, q_3\}$	$\{q_0, q_2, q_3\}$	$\{q_0, q_1, q_3\}$
$\{q_0, q_2, q_3\}$	$\{q_0, q_3\}$	$\{q_0, q_1, q_3\}$
$\{q_0, q_3\}$		

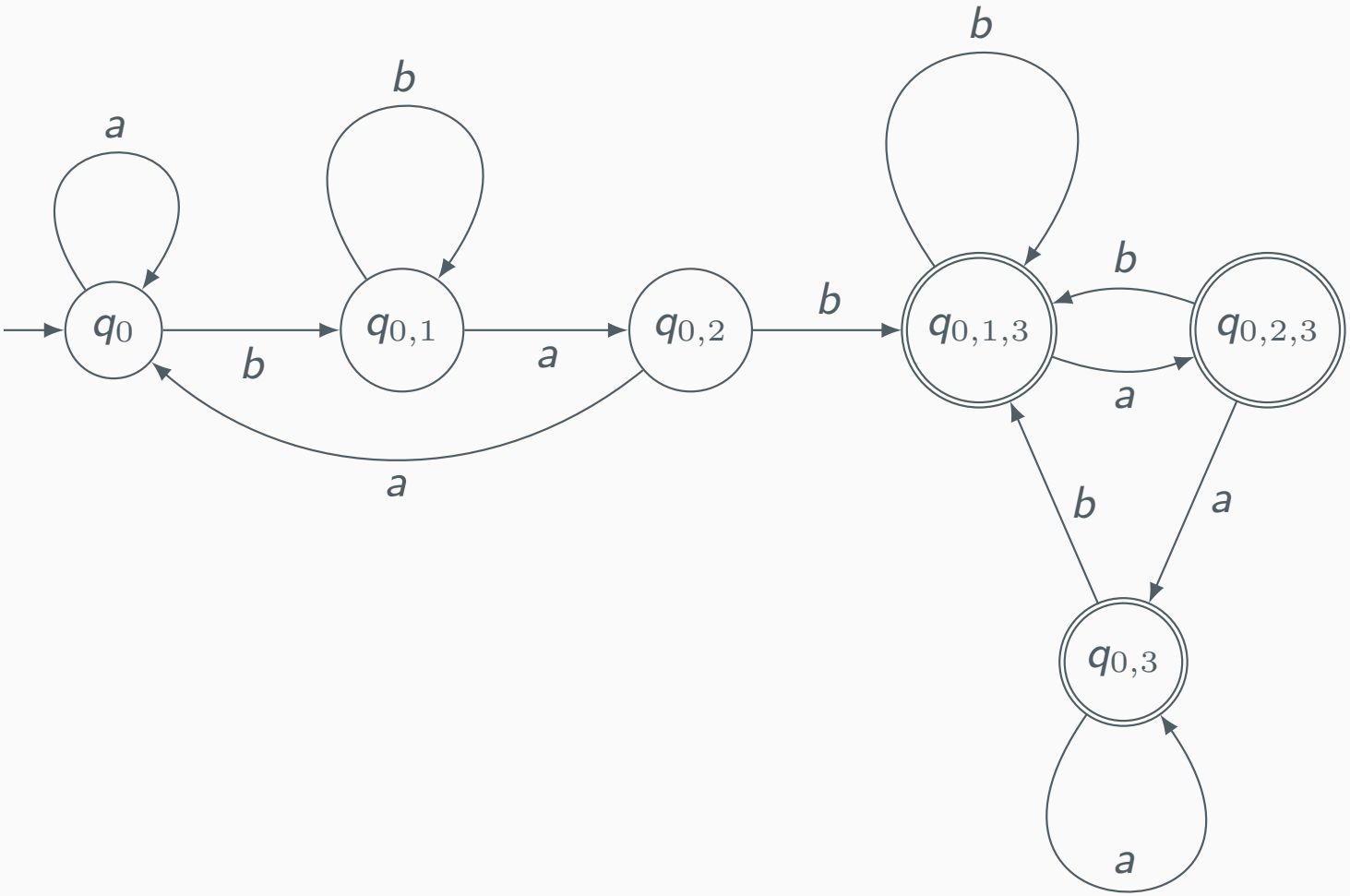
Exemple

Sur l'automate



état	a	b
$\{q_0\}$	$\{q_0\}$	$\{q_0, q_1\}$
$\{q_0, q_1\}$	$\{q_0, q_2\}$	$\{q_0, q_1\}$
$\{q_0, q_2\}$	$\{q_0\}$	$\{q_0, q_1, q_3\}$
$\{q_0, q_1, q_3\}$	$\{q_0, q_2, q_3\}$	$\{q_0, q_1, q_3\}$
$\{q_0, q_2, q_3\}$	$\{q_0, q_3\}$	$\{q_0, q_1, q_3\}$
$\{q_0, q_3\}$	$\{q_0, q_3\}$	$\{q_0, q_1, q_3\}$

Les états acceptants étant $\{q_0, q_1, q_3\}$ et $\{q_0, q_2, q_3\}$. On obtient donc



Déterminisation

Théorème 4

Soit $A = (Q, I, F, \delta)$ un automate non déterministe et A' l'automate des parties : $A' = (\mathscr{P}(Q), I, F', \delta')$.

1. L'automate A' est déterminé
2. Les automates A et A' reconnaissent les mêmes langages

1. Par construction
2. Soit $w \in \mathscr{A}^\star$, on veut vérifier que $w \in L(A) \iff w \in L(A')$.
   On sait que

$$w \in L(A) \iff \exists q_0 \in I, \delta^\star(q_0, w) \cap F \neq \emptyset$$

   et

$$w \in L(A') \iff (\delta')^\star(I, w) \in F'.$$

   Maintenant, $(\delta')^\star(I, w) = \bigcup_{q_0 \in I} \delta^\star(q_0, w)$ donc

$$\begin{aligned} (\delta')^\star(I, w) \in F' &\iff \left(\bigcup_{q_0 \in I} \delta^\star(q_0, w) \right) \cap F \neq \emptyset \\ &\iff \exists q_0 \in I, \delta^\star(q_0, w) \cap F \neq \emptyset \end{aligned}$$

Corollaire

Tout langage L reconnu par un automate non déterministe est reconnu par un automate déterministe.
Cela signifie que l'ensemble des langages reconnaissables est l'ensemble des langages L tels qu'il existe un automate (pas nécessairement déterministe) fini A tel que $L = L(A)$.

Complexité

On peut donc déterminiser un automate afin de pouvoir dire plus aisément / rapidement si un mot appartient (ou non) à un langage donné.

Cependant, il faut prendre en compte la complexité de la déterminisation et surtout le fait qu'il peut y avoir une explosion du nombre d'états.

A priori, le nombre d'états de l'automate des partie croit exponentiellement en fonction du nombre d'états de l'automate de départ.

Nous verrons en exercice un langage reconnu par un automate non déterministe à $N+2$ états mais qui n'est pas reconnu par un automate déterministe avec moins de 2^{N+1} états.

Propriétés de cloture

Introduction

Automate produit et intersection

Réunion

Complémentaire

Différence et différence symétrique

Concaténation

Compléments

Lemme de l'étoile et applications

Réciproque du théorème de Kleene