Théorie des Langages 1

Cours 3 : ε-transitions

L. Rieg (*thanks* M. Echenim)

Grenoble INP - Ensimag, 1re année

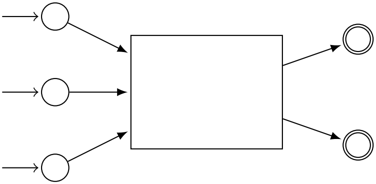
Année 2020-2021

Une propriété

Proposition

Pour tout automate fini A, il existe un automate fini B avec un unique état initial et un unique état final, qui est équivalent à A.

Preuve :

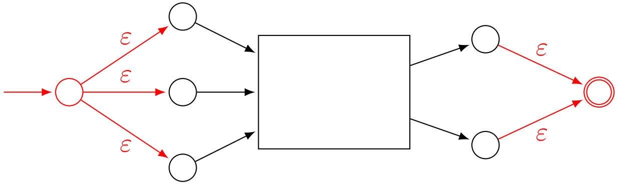


Une propriété

Proposition

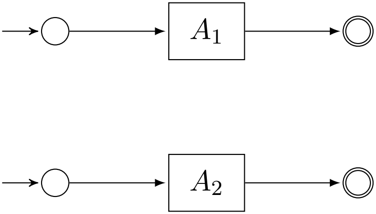
Pour tout automate fini A, il existe un automate fini B avec un unique état initial et un unique état final, qui est équivalent à A.

Preuve :



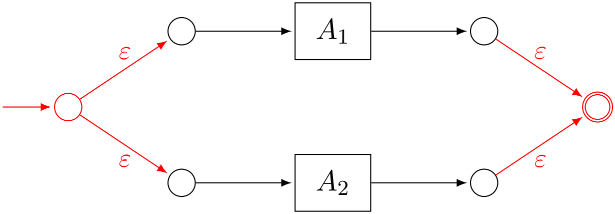
Union

Etant donnés deux automates A_1 et A_2, construire un automate reconnaissant $\mathcal{L}(A_1) \cup \mathcal{L}(A_2)$.



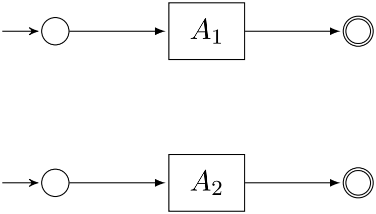
Union

Etant donnés deux automates A_1 et A_2, construire un automate reconnaissant $\mathcal{L}(A_1) \cup \mathcal{L}(A_2)$.



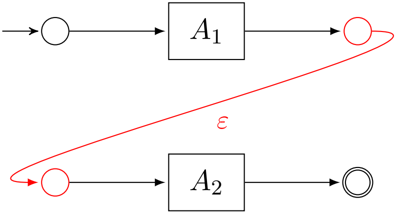
Concaténation

Etant donnés deux automates A_1 et A_2, construire un automate reconnaissant $\mathcal{L}(A_1).\mathcal{L}(A_2)$.



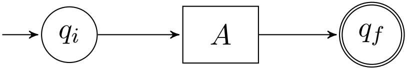
Concaténation

Etant donnés deux automates A_1 et A_2, construire un automate reconnaissant $\mathcal{L}(A_1).\mathcal{L}(A_2)$.



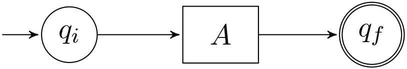
Concaténation itérée

Etant donné un automate A, construire un automate reconnaissant $\mathcal{L}(A)^*$.

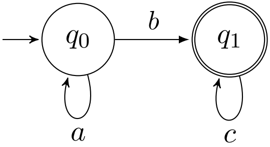


Concaténation itérée

Etant donné un automate A, construire un automate reconnaissant $\mathcal{L}(A)^*$.

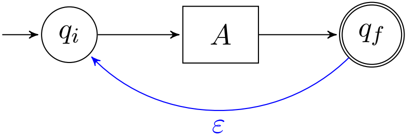


Exemple : $\mathcal{L}(A) = \{a\}^* \{b\} \{c\}^*$

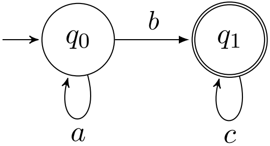


Concaténation itérée

Etant donné un automate A, construire un automate reconnaissant $\mathcal{L}(A)^*$.

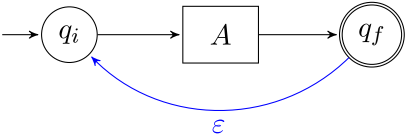


Exemple : $\mathcal{L}(A) = \{a\}^* \{b\} \{c\}^*$

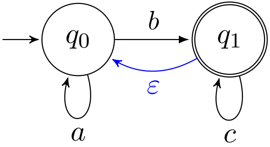


Concaténation itérée

Etant donné un automate A, construire un automate reconnaissant $\mathcal{L}(A)^*$.

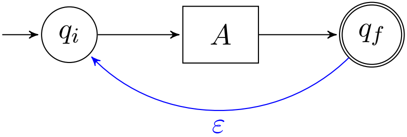


Exemple : $\mathcal{L}(A) = \{a\}^* \{b\} \{c\}^*$

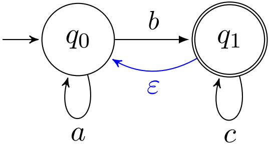


Concaténation itérée

Etant donné un automate A, construire un automate reconnaissant $\mathcal{L}(A)^*$.



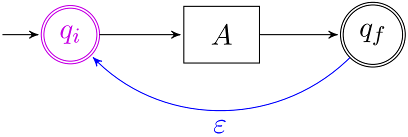
Exemple : $\mathcal{L}(A) = \{a\}^* \{b\} \{c\}^*$



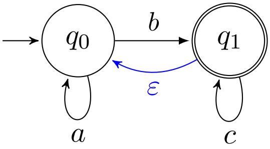
On ne reconnaît pas ε.

Concaténation itérée

Etant donné un automate A, construire un automate reconnaissant $\mathcal{L}(A)^*$.

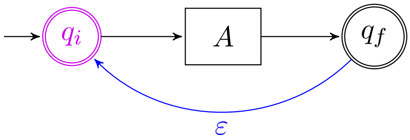


Exemple : $\mathcal{L}(A) = \{a\}^* \{b\} \{c\}^*$

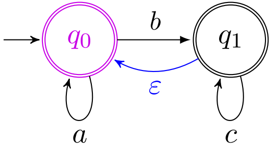


Concaténation itérée

Etant donné un automate A, construire un automate reconnaissant $\mathcal{L}(A)^*$.

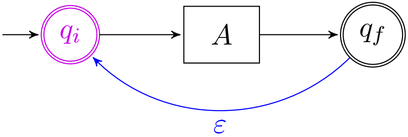


Exemple : $\mathcal{L}(A) = \{a\}^* \{b\} \{c\}^*$

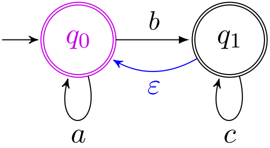


Concaténation itérée

Etant donné un automate A, construire un automate reconnaissant $\mathcal{L}(A)^*$.



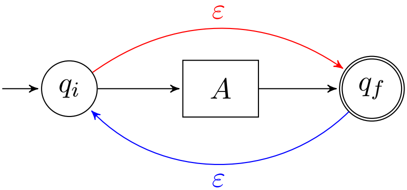
Exemple : $\mathcal{L}(A) = \{a\}^* \{b\} \{c\}^*$



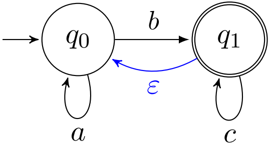
On reconnaît $\{a\}^*$ en trop.

Concaténation itérée

Etant donné un automate A, construire un automate reconnaissant $\mathcal{L}(A)^*$.

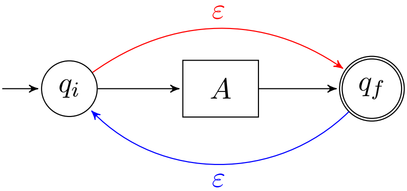


Exemple : $\mathcal{L}(A) = \{a\}^* \{b\} \{c\}^*$

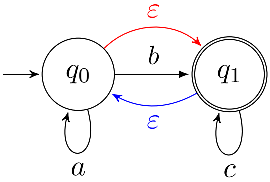


Concaténation itérée

Etant donné un automate A, construire un automate reconnaissant $\mathcal{L}(A)^*$.

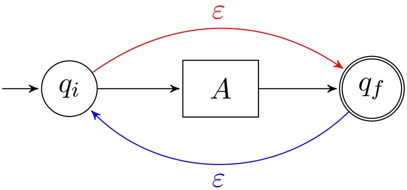


Exemple : $\mathcal{L}(A) = \{a\}^* \{b\} \{c\}^*$

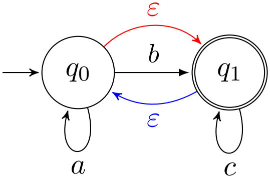


Concaténation itérée

Etant donné un automate A, construire un automate reconnaissant $\mathcal{L}(A)^*$.



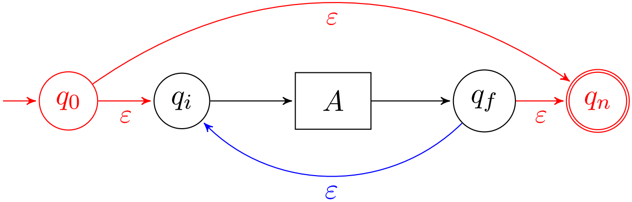
Exemple : $\mathcal{L}(A) = \{a\}^* \{b\} \{c\}^*$



On reconnaît $\{c\}^*$ en trop.

Concaténation itérée

Etant donné un automate A, construire un automate reconnaissant $\mathcal{L}(A)^*$.



Résumé

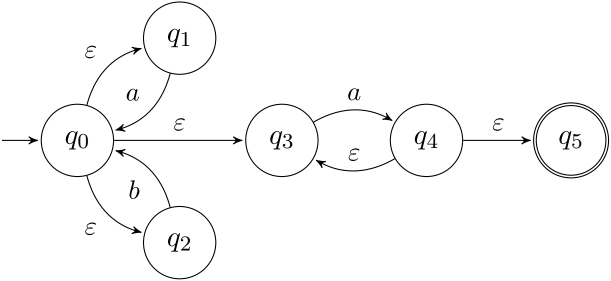
Théorème

La classe des langages réguliers est fermée :

- *par union*
- *par concaténation*
- *par concaténation itérée*

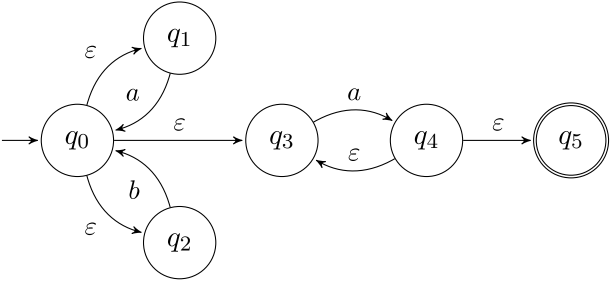
Plusieurs automates pour le même langage

- AFND :

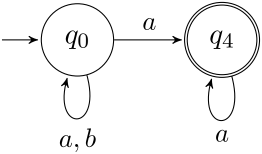


Plusieurs automates pour le même langage

- AFND :

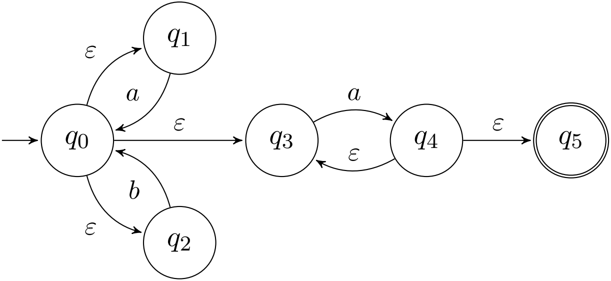


- AFND$-\varepsilon$:

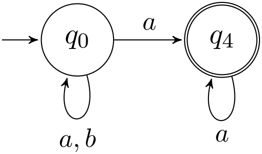


Plusieurs automates pour le même langage

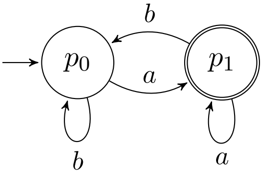
- AFND :



- AFND$-\varepsilon$:



- AFD :



Résumé

Théorème

La classe des langages réguliers est fermée :

- *par union*
- *par concaténation*
- *par concaténation itérée*

Question

Peut-on toujours effectuer les transformations

$$\text{AFND} \iff \text{AFND} - \varepsilon \iff \text{AFD} \quad ?$$

Résumé

Théorème

La classe des langages réguliers est fermée :

- *par union*
- *par concaténation*
- *par concaténation itérée*

Question

Peut-on toujours effectuer les transformations

$$\text{AFND} \iff \text{AFND} - \varepsilon \iff \text{AFD} \quad ?$$

Dans la suite, suppression des ε-transitions.

Chemins

Définition (Chemin)

Soit $A = \langle Q, V, \delta, I, F \rangle$ un automate. L'ensemble des chemins dans A est défini inductivement de la façon suivante :

Base Pour tous $p, q \in Q$ et $a \in V \cup \{\varepsilon\}$,
si $(p, a, q) \in \delta$,
alors (p, a, q) est un chemin dans A de p à q ;

Chemins

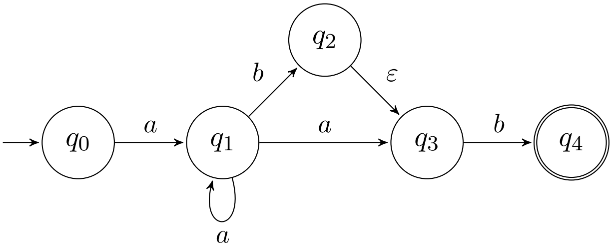
Définition (Chemin)

Soit $A = \langle Q, V, \delta, I, F \rangle$ un automate. L'ensemble des chemins dans A est défini inductivement de la façon suivante :

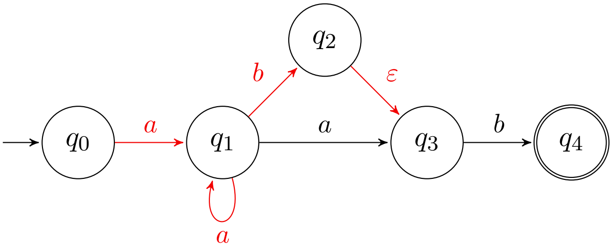
Base Pour tous $p, q \in Q$ et $a \in V \cup \{\varepsilon\}$,
si $(p, a, q) \in \delta$,
alors (p, a, q) est un chemin dans A de p à q ;

Induction Pout tous $p, q, q' \in Q$ et $a \in V \cup \{\varepsilon\}$,
si $(p, a, q) \in \delta$ et χ est un chemin dans A de q à q',
alors $(p, a, q).\chi$ est un chemin dans A de p à q'.

Exemple

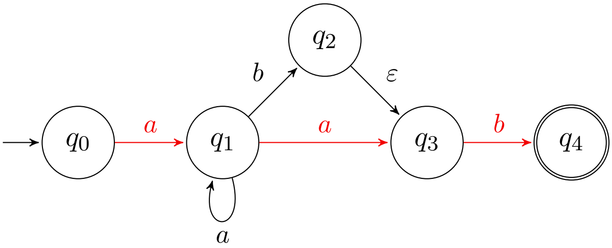


Exemple



$$\chi_1 \quad = \quad (q_0, a, q_1)(q_1, a, q_1)(q_1, b, q_2)(q_2, \varepsilon, q_3)$$

Exemple



$$\begin{aligned}\chi_1 &= (q_0, a, q_1)(q_1, a, q_1)(q_1, b, q_2)(q_2, \varepsilon, q_3)\\ \color{red}{\chi_2} &= (q_0, a, q_1)(q_1, a, q_3)(q_3, b, q_4)\end{aligned}$$

Trace, longueur d'un chemin

Définition

Soit $(q_0, a_1, q_1)(q_1, a_2, q_2) \cdots (q_{n-1}, a_n, q_n)$ un chemin dans A.
Ce chemin est de longueur n et de trace $a_1 a_2 \cdots a_n$.

Trace, longueur d'un chemin

Définition

Soit $(q_0, a_1, q_1)(q_1, a_2, q_2)\cdots(q_{n-1}, a_n, q_n)$ un chemin dans A.
Ce chemin est de longueur n et de trace $a_1 a_2 \cdots a_n$.

Exemple

χ_1 : longueur , trace : ; χ_2 : longueur , trace :

Trace, longueur d'un chemin

Définition

Soit $(q_0, a_1, q_1)(q_1, a_2, q_2) \cdots (q_{n-1}, a_n, q_n)$ un chemin dans A.
Ce chemin est de **longueur** n et de **trace** $a_1 a_2 \cdots a_n$.

Exemple

χ_1 : longueur 4, trace : aab ; χ_2 : longueur 3, trace : aab

Trace, longueur d'un chemin

Définition

Soit $(q_0, a_1, q_1)(q_1, a_2, q_2) \cdots (q_{n-1}, a_n, q_n)$ un chemin dans A.
Ce chemin est de longueur n et de trace $a_1 a_2 \cdots a_n$.

Exemple

χ_1 : longueur 4, trace : aab ; χ_2 : longueur 3, trace : aab

Convention

$\forall q \in Q$, (q, ε, q) est un chemin dans A de q à q, de longueur 0 et de trace ε.

Trace, longueur d'un chemin

Définition

Soit $(q_0, a_1, q_1)(q_1, a_2, q_2) \cdots (q_{n-1}, a_n, q_n)$ un chemin dans A.
Ce chemin est de longueur n et de trace $a_1 a_2 \cdots a_n$.

Exemple

χ_1 : longueur 4, trace : aab ; χ_2 : longueur 3, trace : aab

Convention

$\forall q \in Q$, (q, ε, q) est un chemin dans A de q à q, de longueur 0 et de trace ε.

Proposition

Un mot w est reconnu par A si et seulement si

.

Trace, longueur d'un chemin

Définition

Soit $(q_0, a_1, q_1)(q_1, a_2, q_2) \cdots (q_{n-1}, a_n, q_n)$ un chemin dans A.
Ce chemin est de longueur n et de trace $a_1 a_2 \cdots a_n$.

Exemple

χ_1 : longueur 4, trace : aab ; χ_2 : longueur 3, trace : aab

Convention

$\forall q \in Q$, (q, ε, q) est un chemin dans A de q à q, de longueur 0 et de trace ε.

Proposition

Un mot w est reconnu par A si et seulement si
il existe un chemin dans A d'un $p \in I$ à un $q \in F$, de trace w.

Algorithme de suppression des ε-transitions

Deux étapes :

1. Déterminer les états qu'on peut atteindre en ne se servant que d'ε-transitions
2. Se servir de cette information pour construire l'automate équivalent sans ε-transition

Algorithme de suppression des ε-transitions

Deux étapes :

1. Déterminer les états qu'on peut atteindre en ne se servant que d'ε-transitions
2. Se servir de cette information pour construire l'automate équivalent sans ε-transition

Etape 1

Définition

Etant donné un automate $A = \langle Q, V, \delta, I, F \rangle$, on définit par induction pour tout $p \in Q$ l'ensemble $\mathrm{Acc}_\varepsilon(p)$ des états accessibles depuis p par ε-transitions :

Algorithme de suppression des ε-transitions

Deux étapes :

1. Déterminer les états qu'on peut atteindre en ne se servant que d'ε-transitions
2. Se servir de cette information pour construire l'automate équivalent sans ε-transition

Etape 1

Définition

Etant donné un automate $A = \langle Q, V, \delta, I, F \rangle$, on définit par induction pour tout $p \in Q$ l'ensemble $\mathrm{Acc}_\varepsilon(p)$ des états accessibles depuis p par ε-transitions :

- $p \in \mathrm{Acc}_\varepsilon(p)$

Algorithme de suppression des ε-transitions

Deux étapes :

1. Déterminer les états qu'on peut atteindre en ne se servant que d'ε-transitions
2. Se servir de cette information pour construire l'automate équivalent sans ε-transition

Etape 1

Définition

Etant donné un automate $A = \langle Q, V, \delta, I, F \rangle$, on définit par induction pour tout $p \in Q$ l'ensemble $\mathrm{Acc}_\varepsilon(p)$ des états accessibles depuis p par ε-transitions :

- $p \in \mathrm{Acc}_\varepsilon(p)$
- si $q \in \mathrm{Acc}_\varepsilon(p)$ et $(q, \varepsilon, r) \in \delta$ alors $r \in \mathrm{Acc}_\varepsilon(p)$

Calcul concret de $\mathrm{Acc}_\varepsilon(p)$

Théorème (Admis)

Soit E *un ensemble défini par induction sur l'ensemble d'atomes* B *et l'ensemble de constructeurs* K.
Alors $E = \bigcup_{n\geq 0} E_n$, *où la suite* (E_n) *est définie par :*

Calcul concret de $\mathrm{Acc}_\varepsilon(p)$

Théorème (Admis)

Soit E *un ensemble défini par induction sur l'ensemble d'atomes* B *et l'ensemble de constructeurs* K.

Alors $E = \bigcup_{n \geq 0} E_n$, *où la suite* (E_n) *est définie par :*

$$E_0 \stackrel{\textit{def}}{=} B,$$

Calcul concret de $\mathrm{Acc}_\varepsilon(p)$

Théorème (Admis)

Soit E *un ensemble défini par induction sur l'ensemble d'atomes* B *et l'ensemble de constructeurs* K.
Alors $E = \bigcup_{n\geq 0} E_n$, *où la suite* (E_n) *est définie par :*

$$\begin{aligned} E_0 &\stackrel{def}{=} B, \\ E_{n+1} &\stackrel{def}{=} E_n \cup \{\kappa_i(e_1,\ldots,e_{k_i}) \mid \kappa_i \in K, e_1,\ldots,e_{k_i} \in E_n\} \end{aligned}$$

Calcul concret de $\mathrm{Acc}_\varepsilon(p)$

Théorème (Admis)

Soit E *un ensemble défini par induction sur l'ensemble d'atomes* B *et l'ensemble de constructeurs* K.
Alors $E = \bigcup_{n\geq 0} E_n$, *où la suite* (E_n) *est définie par :*

$$E_0 \stackrel{def}{=} B,$$
$$E_{n+1} \stackrel{def}{=} E_n \cup \{\kappa_i(e_1, \ldots, e_{k_i}) \mid \kappa_i \in K, e_1, \ldots, e_{k_i} \in E_n\}$$

algorithme $\mathrm{Acc}_\varepsilon(p) =$
$n \leftarrow 0,\ A_0 \leftarrow \{p\}$
répéter
$n \leftarrow n + 1$
$A_n \leftarrow A_{n-1} \cup \{r \in Q \mid \exists q \in A_{n-1},\ (q, \varepsilon, r) \in \delta\}$
jusqu'à $A_n = A_{n-1}$
renvoyer A_n

Calcul concret de $\mathrm{Acc}_\varepsilon(p)$

Théorème (Admis)

Soit E *un ensemble défini par induction sur l'ensemble d'atomes* B *et l'ensemble de constructeurs* K.
Alors $E = \bigcup_{n \geq 0} E_n$, *où la suite* (E_n) *est définie par :*

$$\begin{aligned} E_0 &\overset{def}{=} B, \\ E_{n+1} &\overset{def}{=} E_n \cup \{\kappa_i(e_1, \ldots, e_{k_i}) \mid \kappa_i \in K, e_1, \ldots, e_{k_i} \in E_n\} \end{aligned}$$

algorithme $\mathrm{Acc}_\varepsilon(p) =$
  $n \leftarrow 0,\ A_0 \leftarrow \{p\}$
  répéter
    $n \leftarrow n + 1$
    $A_n \leftarrow A_{n-1} \cup \{r \in Q \mid \exists q \in A_{n-1},\ (q, \varepsilon, r) \in \delta\}$
  jusqu'à $A_n = A_{n-1}$
  renvoyer A_n

Question : Est-ce que ça termine toujours ?

Algorithme de suppression des ε-transitions (suite)

Etape 2.

Définition

Etant donné un automate $A = \langle Q, V, \delta, I, F \rangle$, on définit l'automate $B = \langle Q, V, \delta', I, F' \rangle$ de la façon suivante :

Algorithme de suppression des ε-transitions (suite)

Etape 2.

Définition

Etant donné un automate $A = \langle Q, V, \delta, I, F \rangle$, on définit l'automate $B = \langle Q, V, \delta', I, F' \rangle$ de la façon suivante :

- $(p, a, q) \in \delta'$ ssi $a \neq \varepsilon$ et $\exists p' \in \mathrm{Acc}_\varepsilon(p)$ tel que $(p', a, q) \in \delta$

Algorithme de suppression des ε-transitions (suite)

Etape 2.

Définition

Etant donné un automate $A = \langle Q, V, \delta, I, F \rangle$, on définit l'automate $B = \langle Q, V, \delta', I, F' \rangle$ de la façon suivante :

- $(p, a, q) \in \delta'$ ssi $a \neq \varepsilon$ et $\exists p' \in \mathrm{Acc}_\varepsilon(p)$ tel que $(p', a, q) \in \delta$
- $F' = \{p \in Q \mid \mathrm{Acc}_\varepsilon(p) \cap F \neq \emptyset\}$

Algorithme de suppression des ε-transitions (suite)

Etape 2.

Définition

Etant donné un automate $A = \langle Q, V, \delta, I, F \rangle$, on définit l'automate $B = \langle Q, V, \delta', I, F' \rangle$ de la façon suivante :

- $(p, a, q) \in \delta'$ ssi $a \neq \varepsilon$ et $\exists p' \in \mathrm{Acc}_\varepsilon(p)$ tel que $(p', a, q) \in \delta$
- $F' = \{p \in Q \mid \mathrm{Acc}_\varepsilon(p) \cap F \neq \emptyset\}$

Proposition

B est un automate sans ε-transition.

Algorithme de suppression des ε-transitions (suite)

Etape 2.

Définition

Etant donné un automate $A = \langle Q, V, \delta, I, F\rangle$, on définit l'automate $B = \langle Q, V, \delta', I, F'\rangle$ de la façon suivante :

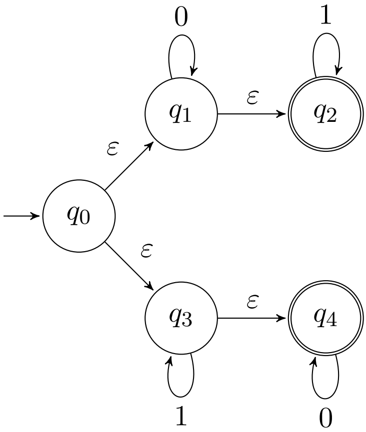
- $(p, a, q) \in \delta'$ ssi $a \neq \varepsilon$ et $\exists p' \in \mathrm{Acc}_\varepsilon(p)$ tel que $(p', a, q) \in \delta$
- $F' = \{p \in Q \mid \mathrm{Acc}_\varepsilon(p) \cap F \neq \emptyset\}$

Proposition

B est un automate sans ε-transition.

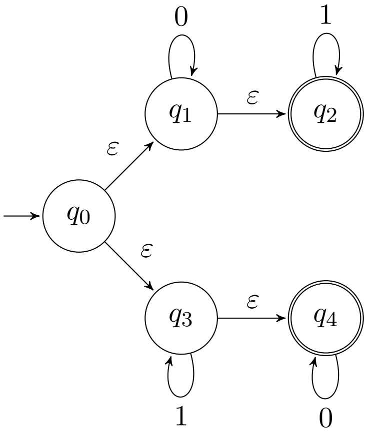
Exercice

Construire B pour l'automate A suivant :



Exercice

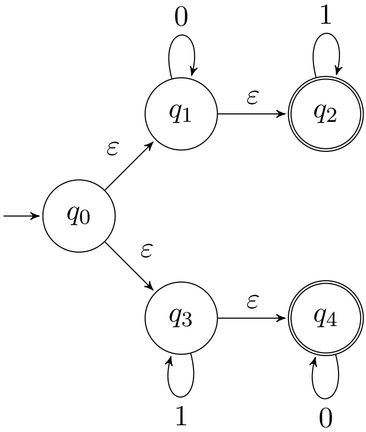
Construire B pour l'automate A suivant :



p	$\mathrm{Acc}_\varepsilon(p)$
q_0	q_0, q_1, q_2, q_3, q_4
q_1	q_1, q_2
q_2	q_2
q_3	q_3, q_4
q_4	q_4

Exercice

Construire B pour l'automate A suivant :

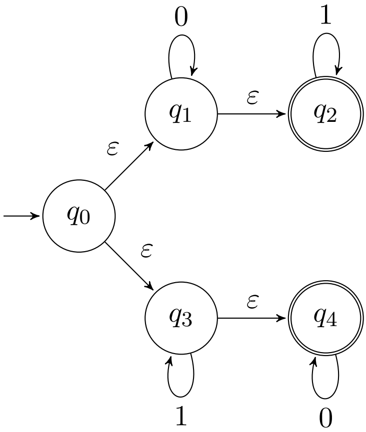


p	$\mathrm{Acc}_\varepsilon(p)$
q_0	q_0, q_1, q_2, q_3, q_4
q_1	q_1, q_2
q_2	q_2
q_3	q_3, q_4
q_4	q_4

δ'	0	1
q_0	q_1, q_4	q_2, q_3
q_1	q_1	q_2
q_2	–	q_2
q_3	q_4	q_3
q_4	q_4	–

Exercice

Construire B pour l'automate A suivant :

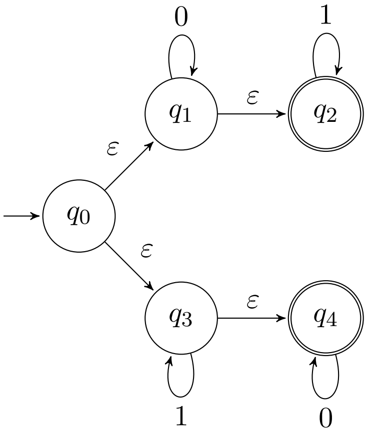


p	$\mathrm{Acc}_\varepsilon(p)$
q_0	q_0, q_1, q_2, q_3, q_4
q_1	q_1, q_2
q_2	q_2
q_3	q_3, q_4
q_4	q_4

δ'	0	1
q_0	q_1, q_4	q_2, q_3
q_1	q_1	q_2
q_2	–	q_2
q_3	q_4	q_3
q_4	q_4	–

Exercice

Construire B pour l'automate A suivant :

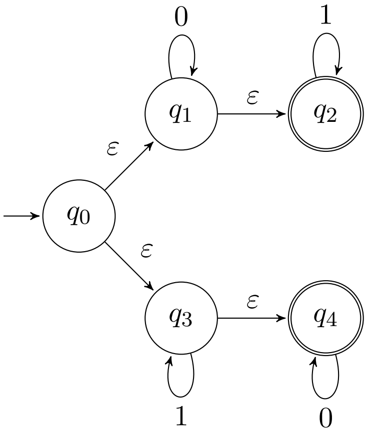


p	$\mathrm{Acc}_\varepsilon(p)$
q_0	q_0, q_1, q_2, q_3, q_4
q_1	q_1, q_2
q_2	q_2
q_3	q_3, q_4
q_4	q_4

δ'	0	1
q_0	q_1, q_4	q_2, q_3
q_1	q_1	q_2
q_2	–	q_2
q_3	q_4	q_3
q_4	q_4	–

Exercice

Construire B pour l'automate A suivant :



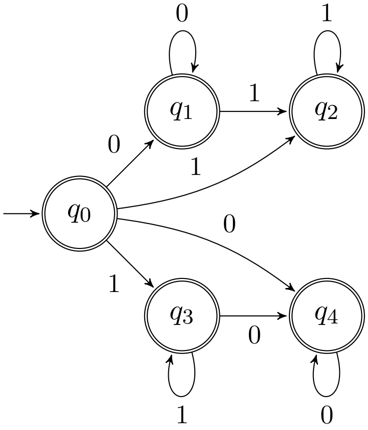
p	$\mathrm{Acc}_\varepsilon(p)$
q_0	q_0, q_1, q_2, q_3, q_4
q_1	q_1, q_2
q_2	q_2
q_3	q_3, q_4
q_4	q_4

δ'	0	1
q_0	q_1, q_4	q_2, q_3
q_1	q_1	q_2
q_2	–	q_2
q_3	q_4	q_3
q_4	q_4	–

$F' = \{q_0, q_1, q_2, q_3, q_4\}$

Exercice (suite)

Solution :



Résultat principal

Théorème

$\forall$ *automate* A*, l'automate* B *défini précédemment est équivalent à* A*.*

Résultat principal

Théorème

$\forall$ *automate* A*, l'automate* B *défini précédemment est équivalent à* A*.*

Lemme intermédiaire

L'automate B vérifie la propriété suivante :

Il existe un chemin de p à q de trace w dans A
si et seulement si
il existe $r \in Q$ tel qu'il existe un chemin de p à r de trace w dans B
et un chemin de r à q de trace ε dans A.

Résultat principal

Théorème

$\forall$ *automate* A*, l'automate* B *défini précédemment est équivalent à* A*.*

Lemme intermédiaire

L'automate B vérifie la propriété suivante :

Il existe un chemin de p à q de trace w dans A
si et seulement si
il existe $r \in Q$ tel qu'il existe un chemin de p à r de trace w dans B
et un chemin de r à q de trace ε dans A.

Graphiquement : $\underset{\boxed{A}}{p \overset{w}{\rightsquigarrow} q} \iff \exists r, \underset{\boxed{B}}{p \overset{w}{\rightsquigarrow} r}$ et $\underset{\boxed{A}}{r \overset{\varepsilon}{\rightsquigarrow} q}$

Résultat principal

Théorème

$\forall$ *automate* A*, l'automate* B *défini précédemment est équivalent à* A*.*

Lemme intermédiaire

L'automate B vérifie la propriété suivante :

Il existe un chemin de p à q de trace w dans A
si et seulement si
il existe $r \in Q$ tel qu'il existe un chemin de p à r de trace w dans B
et un chemin de r à q de trace ε dans A.

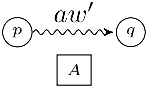
Graphiquement : $\underset{\boxed{A}}{p \overset{w}{\rightsquigarrow} q} \iff \exists r, \underset{\boxed{B}}{p \overset{w}{\rightsquigarrow} r}$ et $\underset{\boxed{A}}{r \overset{\varepsilon}{\rightsquigarrow} q}$

Preuve par induction sur w.

- Base : $w = \varepsilon$. Il suffit de prendre $r \overset{\text{def}}{=} p$.

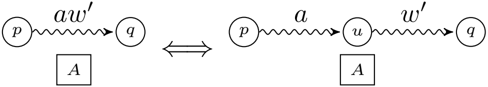
Preuve par induction, suite

- Induction : $w = aw' \quad (a \in V)$



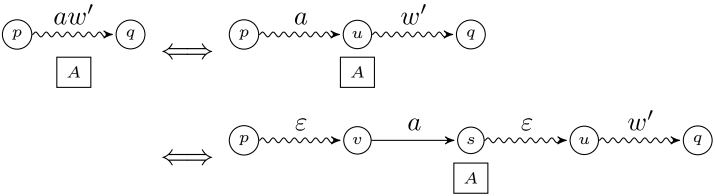
Preuve par induction, suite

- Induction : $w = aw' \quad (a \in V)$



Preuve par induction, suite

- Induction : $w = aw' \quad (a \in V)$



Preuve par induction, suite

- Induction : $w = aw' \quad (a \in V)$

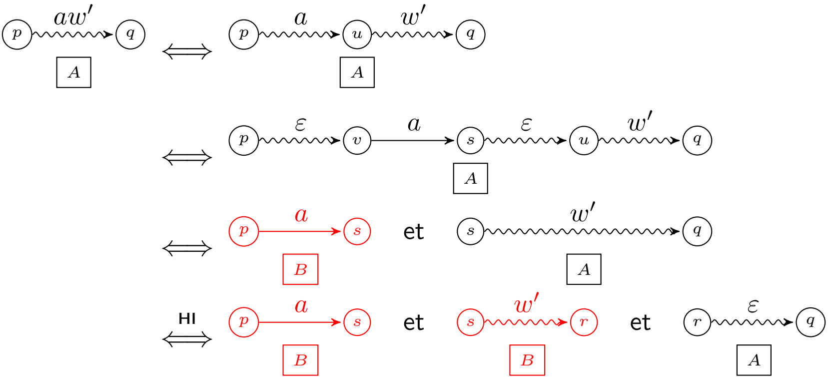
$$\underset{\boxed{A}}{p \overset{aw'}{\rightsquigarrow} q} \iff \underset{\boxed{A}}{p \overset{a}{\rightsquigarrow} u \overset{w'}{\rightsquigarrow} q}$$

$$\iff \underset{\boxed{A}}{p \overset{\varepsilon}{\rightsquigarrow} v \xrightarrow{a} s \overset{\varepsilon}{\rightsquigarrow} u \overset{w'}{\rightsquigarrow} q}$$

$$\iff \underset{\boxed{B}}{p \xrightarrow{a} s} \quad \text{et} \quad \underset{\boxed{A}}{s \overset{w'}{\rightsquigarrow} q}$$

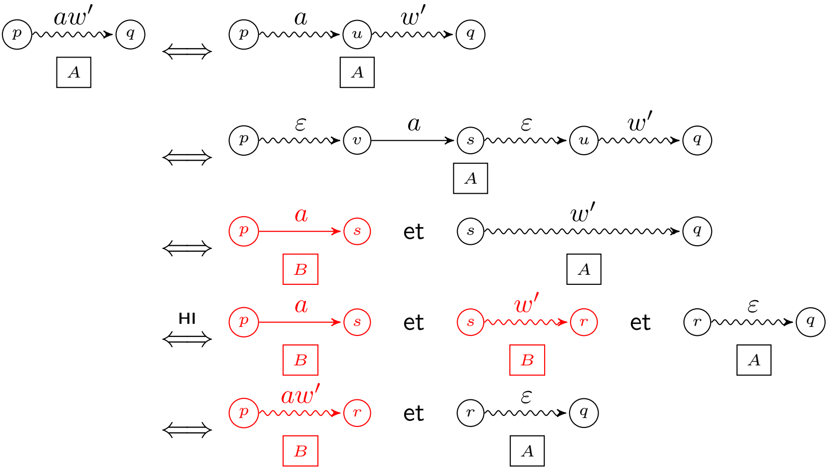
Preuve par induction, suite

- Induction : $w = aw' \quad (a \in V)$



Preuve par induction, suite

- Induction : $w = aw'$ $\quad (a \in V)$



Preuve du théorème

$w \in \mathcal{L}(A) \iff \exists$ un chemin de $q_0 \in I$ à $q_f \in F$ de trace w dans A

Preuve du théorème

$$
\begin{aligned}
w \in \mathcal{L}(A) \quad & \Longleftrightarrow \quad \exists \text{ un chemin de } q_0 \in I \text{ à } q_f \in F \text{ de trace } w \text{ dans } A \\
& \Longleftrightarrow \quad \exists r \in Q, \exists \text{ un chemin de } q_0 \in I \text{ à } r \text{ de trace } w \text{ dans } B \\
& \qquad\quad \text{et } \exists \text{ un chemin de } r \text{ à } q_f \in F \text{ de trace } \varepsilon \text{ dans } A
\end{aligned}
$$

Preuve du théorème

$$
\begin{aligned}
w \in \mathcal{L}(A) \quad &\iff \quad \exists \text{ un chemin de } q_0 \in I \text{ à } q_f \in F \text{ de trace } w \text{ dans } A \\
&\iff \quad \exists r \in Q, \exists \text{ un chemin de } q_0 \in I \text{ à } r \text{ de trace } w \text{ dans } B \\
&\qquad\quad \text{et } \exists \text{ un chemin de } r \text{ à } q_f \in F \text{ de trace } \varepsilon \text{ dans } A
\end{aligned}
$$

Preuve du théorème

$$\begin{array}{rcl} w \in \mathcal{L}(A) & \iff & \exists \text{ un chemin de } q_0 \in I \text{ à } q_f \in F \text{ de trace } w \text{ dans } A \\ & \iff & \exists r \in Q, \exists \text{ un chemin de } q_0 \in I \text{ à } r \text{ de trace } w \text{ dans } B \\ & & \quad \text{et } \exists \text{ un chemin de } r \text{ à } q_f \in F \text{ de trace } \varepsilon \text{ dans } A \\ & \iff & \exists r \in Q, \exists \text{ un chemin de } q_0 \in I \text{ à } r \text{ de trace } w \text{ dans } B \\ & & \quad \text{et } r \in F' \end{array}$$

Preuve du théorème

$$
\begin{array}{rcl}
w \in \mathcal{L}(A) & \iff & \exists \text{ un chemin de } q_0 \in I \text{ à } q_f \in F \text{ de trace } w \text{ dans } A \\
& \iff & \exists r \in Q, \exists \text{ un chemin de } q_0 \in I \text{ à } r \text{ de trace } w \text{ dans } B \\
& & \quad \text{et } \exists \text{ un chemin de } r \text{ à } q_f \in F \text{ de trace } \varepsilon \text{ dans } A \\
& \iff & \exists r \in Q, \exists \text{ un chemin de } q_0 \in I \text{ à } r \text{ de trace } w \text{ dans } B \\
& & \quad \text{et } r \in F'
\end{array}
$$

Preuve du théorème

$$
\begin{aligned}
w \in \mathcal{L}(A) &\iff \exists \text{ un chemin de } q_0 \in I \text{ à } q_f \in F \text{ de trace } w \text{ dans } A \\
&\iff \exists r \in Q, \exists \text{ un chemin de } q_0 \in I \text{ à } r \text{ de trace } w \text{ dans } B \\
&\qquad \text{et } \exists \text{ un chemin de } r \text{ à } q_f \in F \text{ de trace } \varepsilon \text{ dans } A \\
&\iff \exists r \in Q, \exists \text{ un chemin de } q_0 \in I \text{ à } r \text{ de trace } w \text{ dans } B \\
&\qquad \text{et } r \in F' \\
&\iff w \in \mathcal{L}(B)
\end{aligned}
$$

Preuve du théorème

$$
\begin{aligned}
w \in \mathcal{L}(A) \quad & \iff \quad \exists \text{ un chemin de } q_0 \in I \text{ à } q_f \in F \text{ de trace } w \text{ dans } A \\
& \iff \quad \exists r \in Q, \exists \text{ un chemin de } q_0 \in I \text{ à } r \text{ de trace } w \text{ dans } B \\
& \qquad\quad \text{et } \exists \text{ un chemin de } r \text{ à } q_f \in F \text{ de trace } \varepsilon \text{ dans } A \\
& \iff \quad \exists r \in Q, \exists \text{ un chemin de } q_0 \in I \text{ à } r \text{ de trace } w \text{ dans } B \\
& \qquad\quad \text{et } r \in F' \\
& \iff \quad w \in \mathcal{L}(B)
\end{aligned}
$$

Les automates A et B sont bien équivalents.