5. Equivalences d'automates

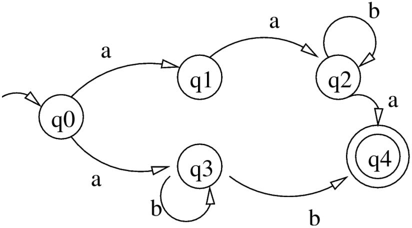
5. Equivalences d'automates

♣ il existe plusieurs types d'Automates d'Etats Finis

♠ ils sont équivalents

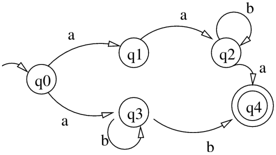
La définition d'un automate d'états finis n'interdit pas les "conflits".

$L = aab^*a + ab^*b$



Comment doit-on interpréter $\delta(q_0, a)$?

Le choix est **non-déterministe**.



5.1. Le problème du déterminisme

Définition

Un automates d'états finis déterministe (AEFD)
(en anglais : Deterministic Finite Automaton (DFA))
est un automate d'états finis tel que, de chaque état $q \in Q$*,*
il part $|\Sigma|$ *transitions, une pour chacune des lettres*
de l'alphabet Σ*.*

Remarque : Pas d'ϵ-transition !

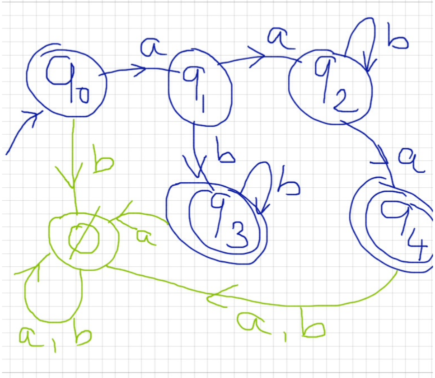
5.1. Le problème du déterminisme

Dans un automate déterministe, on a un '**état poubelle**', vers lequel on envoie toutes les transitions non définies.

$\emptyset$

Souvent, cet état poubelle est implicite.

Exemple d'automate déterministe sur $\Sigma = \{a, b\}$ pour $L = aab^*a + ab^*b$:



5.2. Différentes sortes d'AEF

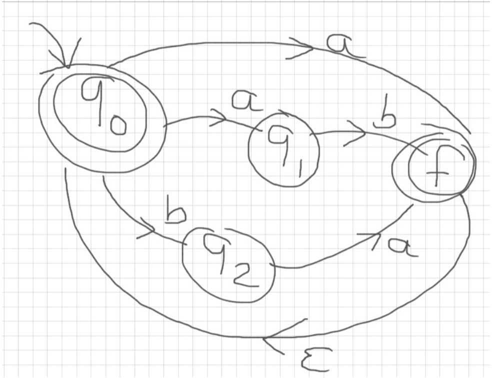
On définit 3 sortes d'automates d'états finis :

1. les automates d'états finis non-déterministes **sans** ϵ-transition (NFA-W, W pour"Without")
2. les automates d'états finis non-déterministes **avec** ϵ-transition (NFA-ϵ)
3. les automates d'états finis **déterministes**

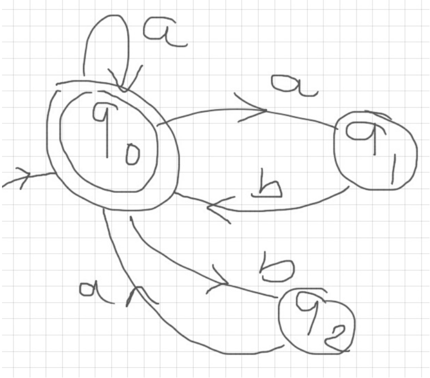
5.2. Différentes sortes d'AEF

Exemple : $L = \{a, ab, ba\}^*$

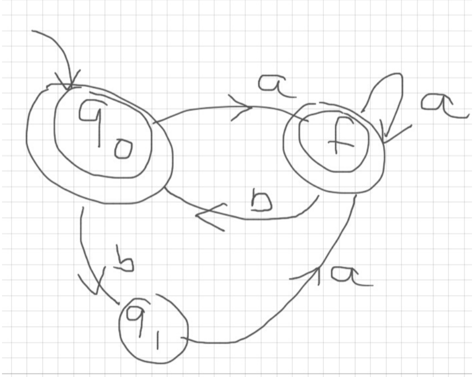
$L = \{a, ab, ba\}$ * non déterministe avec ϵ -transition



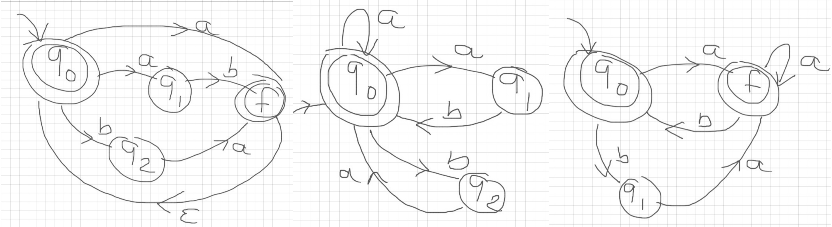
$L = \{a, ab, ba\}$ * non déterministe sans ϵ -transition



$L = \{a, ab, ba\}$ * déterministe



$L = \{a, ab, ba\}$ *



5.2. Différentes sortes d'AEF

Théorème

La classe des langages reconnus par :

- *les automates d'états finis déterministes*
- *les automates d'états finis non-déterministes sans ϵ-transition*
- *les automates d'états finis non-déterministes avec ϵ-transition*

est la même : celle des **langages rationnels**.

Preuve : constructive
(algorithmes de passage d'un type d'AEF à un autre)

5.3. Déterminisation d'un AEF sans ϵ-transition

$\diamondsuit$ On part d'un AEF non déterministe
$A_1 = (\mathscr{Q}, \Sigma, \delta, q_0, F)$ sans ϵ-transition.

$\clubsuit$ On calcule un automate d'états finis déterministe
$A_2 = (\mathscr{Q}', \Sigma, \delta', Q'_0, F')$, avec $Q'_0 = \{q_0\}$

5.3. Déterminisation d'un AEF sans ϵ-transition

Principe : On construit les états et la table de transition de δ' :

1 On construit la table de transition de δ (qui comporte des ensembles d'états)

2 Initialisation de δ'

- on commence par $Q'_0=\{q_0\}$
- on applique chaque caractère x de Σ à Q'_0
- on obtient un ensemble d'états qui est sera état de $\mathscr{Q}'$

5.3. Déterminisation d'un AEF sans ϵ-transition

3 Construction de δ'
   - on choisit un état Q' de $\mathscr{Q}'$ non encore traité
   - on applique chaque caractère x de Σ chaque état de Q' avec δ
   - on obtient un ensemble d'états
      1. si cet ensemble ne correspond pas à un été déjà défini de $\mathscr{Q}'$, on crée un nouvel état de $\mathscr{Q}'$

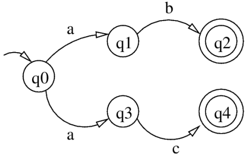
4 les états finaux de A_2 sont ceux qui contiennent au moins un état final de F

Algo DETERMINISATION

Donnee : un automate $A_1 = (\mathscr{Q}, \Sigma, \delta, q_0, F)$
Resultat : un automate déterministe $A_2 = (\mathscr{Q}', \Sigma, \delta', Q'_0, F')$.
Initialisation : $Q'_0 \leftarrow \{q_0\}$;ATRAITER $\leftarrow Q'_0 = \{q_0\}$; $\mathscr{Q}' \leftarrow \{Q'_0\}$;
tant que ATRAITER $\neq \emptyset$ faire
    CHOISIR Q' dans ATRAITER ; ATRAITER $\leftarrow$ ATRAITER - Q' ;
    pour chaque caractère x de Σ faire
        pour chaque état Q de Q' faire
            $\delta'(Q', x) \leftarrow \delta'(Q', x) \cup \delta(Q, x)$;
        si $\delta'(Q', x)$ n'est pas un état de $\mathscr{Q}'$ alors
            $Q'' \leftarrow \delta'(Q', x)$; $\mathscr{Q}' \leftarrow \mathscr{Q}' + \{Q''\}$;
            ATRAITER $\leftarrow$ ATRAITER + $\{Q''\}$;
pour chaque état Q' de $\mathscr{Q}'$ contenant un état de F faire
    AJOUTER Q' à F' ;
Retourner$((\mathscr{Q}', \Sigma, \delta', Q'_0, F'))$.

Exemple :

$L = \{ab, ac\}$

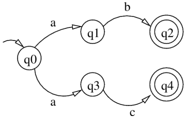


δ	a	b	c
q_0	$\{q_1, q_3\}$	$\{\emptyset\}$	$\{\emptyset\}$
q_1	$\{\emptyset\}$	$\{q_2\}$	$\{\emptyset\}$
q_2	$\{\emptyset\}$	$\{\emptyset\}$	$\{\emptyset\}$
q_3	$\{\emptyset\}$	$\{\emptyset\}$	$\{q_4\}$
q_4	$\{\emptyset\}$	$\{\emptyset\}$	$\{\emptyset\}$

5.3. Déterminisation d'un AEF sans ϵ-transition

Initialisation : $Q'_0 = \{q_0\}$

$\delta'(Q'_0, a) = \{q_1, q_3\}$: on crée un nouvel état $Q'_1 = \{q_1, q_3\}$

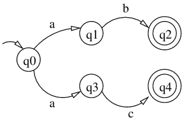


$\delta'(Q'_0, b) = \{\emptyset\}$; $\delta'(Q'_0, c) = \{\emptyset\}$

5.3. Déterminisation d'un AEF sans ϵ-transition

On a un état $Q'_1 = \{q_1, q_3\}$ qui n'est pas traité :

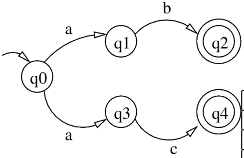
$\delta'(Q'_1, a) = \emptyset$
$\delta'(Q'_0, b) = \{q_2\}$: on crée un nouvel état $Q'_2 = \{q_2\}$
$\delta'(Q'_0, c) = \{q_4\}$: on crée un nouvel état $Q'_3 = \{q_4\}$



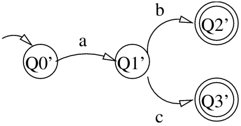
5.3. Déterminisation d'un AEF sans ϵ-transition

On traite l'état $Q'_2 = \{q_2\}$: $\delta'(Q'_2, a) = \delta'(Q'_2, b) = \delta'(Q'_2, c) = \emptyset$
On traite l'état $Q'_3 = \{q_4\}$: $\delta'(Q'_3, a) = \delta'(Q'_3, b) = \delta'(Q'_3, c) = \emptyset$
Etats finaux : Q'_2 parce qu'il contient q_2, et Q'_3 parce qu'il contient q_4.

A la fin, on obtient :



δ	a	b	c
q_0	$\{q_1,q_3\}$	$\{\emptyset\}$	$\{\emptyset\}$
q_1	$\{\emptyset\}$	$\{q_2\}$	$\{\emptyset\}$
q_2	$\{\emptyset\}$	$\{\emptyset\}$	$\{\emptyset\}$
q_3	$\{\emptyset\}$	$\{\emptyset\}$	$\{q_4\}$
q_4	$\{\emptyset\}$	$\{\emptyset\}$	$\{\emptyset\}$



δ'	a	b	c
$Q'_0=\{q_0\}$	$\{q_1,q_3\}=Q'_1$	$\{\emptyset\}$	$\{\emptyset\}$
$Q'_1=\{q_1,q_3\}$	$\{\emptyset\}$	$\{q_2\}=Q'_2$	$\{q_4\}=Q'_3$
$Q'_2=q_2$	$\{\emptyset\}$	$\{\emptyset\}$	$\{\emptyset\}$
$Q'_3=q_4$	$\{\emptyset\}$	$\{\emptyset\}$	$\{\emptyset\}$

Remarque :

Etant donné un AEF non déterministe à k états, l'AEF déterministe correspondant peut avoir 2^k états.

5.4. Déterminisation d'un AEF avec ϵ-transitions

On étend la technique de déterminisation en étendant la fonction de transition $\delta'(Q_i, x)$ à une fonction donnée par les mots $\epsilon^* x \epsilon^*$.

5.4. Déterminisation d'un AEF avec ϵ-transitions

Définition

On appelle ϵ-**fermeture** *d'un état* q *l'ensemble des états* q_i *atteignables à partir de* q *par un chemin étiqueté uniquement par le mot vide* ϵ.

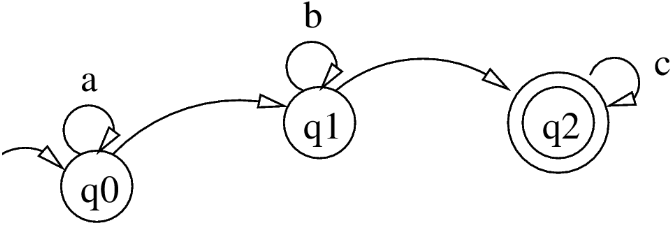
Définition

On appelle ϵ-**fermeture** *d'un ensemble* Q *d'états l'union des* ϵ*-fermetures des états appartenant à* Q.

5.4. Déterminisation d'un AEF avec ϵ-transitions

Exemple :

$a^* b^* c^*$



ϵ-fermeture$(q_0) = \{q_0, q_1, q_2\}$
ϵ-fermeture$(q_1) = \{q_1, q_2\}$
ϵ-fermeture$(q_2) = \{q_2\}$

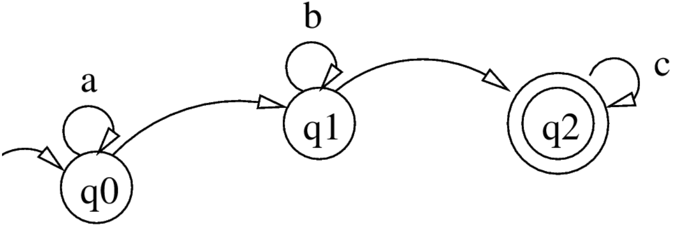
5.4. Déterminisation d'un AEF avec ϵ-transitions

Principe de déterminisation :
Pour un état Q' de l'AEF déterministe en cours de calcul :

1. On part de l'ϵ-fermeture de Q'.
2. On calcule $\delta(Q')$
3. On calcule l'ϵ-fermeture de $\delta(Q')$
4. On obtient un état du nouvel automate

Les états finaux du nouvel automate sont ceux qui contiennent au moins un état final de l'automate de départ.

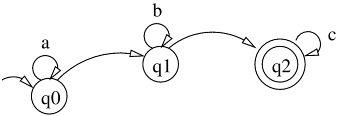
Exemple : $a^*b^*c^*$:



δ	a	b	c	ϵ
q_0	$\{q_0\}$	$\{\emptyset\}$	$\{\emptyset\}$	$\{q_0,q_1,q_2\}$
q_1	$\{\emptyset\}$	$\{q_1\}$	$\{\emptyset\}$	$\{q_1,q_2\}$
q_2	$\{\emptyset\}$	$\{\emptyset\}$	$\{q_2\}$	$\{q_2\}$

Construction de δ', Q'_0=ϵ-fermeture(q_0)=$\{q_0,q_1,q_2\}$

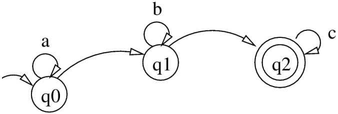
$a^* b^* c^*$



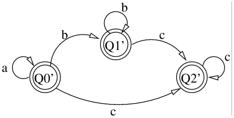
δ'	a	b	c
$Q'_0 = \{q_0, q_1, q_2\}$	$\{q_0\} \rightarrow Q'_0$	$\{q_1\} \rightarrow \{q_1,q_2\}=Q'_1$	$\{q_2\} \rightarrow \{q_2\}=$
$Q'_1 = \{q_1, q_2\}$	$\{\emptyset\}$	$\{Q'_1\}$	$\{Q'_2\}$
$Q'_2 = \{q_2\}$	$\{\emptyset\}$	$\{\emptyset\}$	$\{Q'_2\}$

5.4. Déterminisation d'un AEF avec ϵ-transitions

$a^* b^* c^*$



Tous les états sont finaux, on obtient l'automate :



5.5. Minimisation d'un AEF déterministe

Théorème

(de Nérode - Myhill) :
Pour un langage rationnel donné L, il existe un automate d'états finis déterministe **canonique** *(uniquement défini), et qui comporte un nombre* **minimum** *d'états (parmi tous les automates déterministes), reconnaissant L.*

$\rightsquigarrow$ Il existe un algorithme très efficace de minimisation.

5.5. Minimisation d'un AEF déterministe

Principe de minimisation d'un automate d'états finis déterministe : utilise le principe algorithmique d'**éclatement de partitions**.

5.5. Minimisation d'un AEF déterministe

Rappel : une **partition** d'un ensemble est la définition d'un ensemble de **classes**, tel que l'union de toutes les classes est l'ensemble de départ et l'intersection de deux classes est vide (une partition correspond à une **relation d''équivalence**)

Principe algorithmique d'**éclatement de partitions** (ou d'affinement de partitions)

1. on part d'une (ou plusieurs) (grandes) classes
2. on a un critère qui permet de partitionner un classe en plusieurs classes plus petites
3. on arrête quand chaque classe obtenue est non-partitionnable

5.5. Minimisation d'un AEF déterministe

Pour minimiser un AEF déterministe :

1. on retire les états non atteignables ;
2. on partitionne l'ensemble des états en deux classes :
    1. les états finaux
    2. les états non finaux (y compris l'état poubelle $\emptyset$)

Etape d'éclatement d'une classe C_i :

1. appliquer à C_i une transition par un caractère x de $\in \Sigma$;
2. séparer les éléments de C_i qui n'aboutissent pas à la même classe

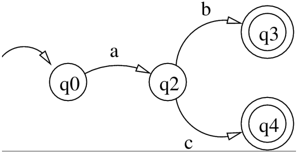
$\rightsquigarrow$On répète jusqu'à ce qu'il n'y ait plus d'éclatement possible.

A la fin, on a pour toute classe C_i obtenue :

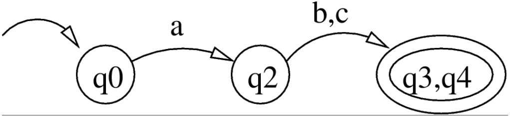
$$\forall x \in \Sigma,\ \forall q, q' \in C_i,\ \delta(q, x) = \delta(q', x)$$

$\rightsquigarrow$On obtient la description d'un nouvel AEF déterministe,
dont l'état initial est l'état contenant q_0
et dont les états finaux sont les états contenant un état final de l'automate de départ.

Exemple : $L = \{ab, ac\}$



On part de $C_1 = \{q_3, q_4\}$ et $C_2 = \{q_0, q_2, \emptyset\}$
$C_2 = \{q_0, q_2, \emptyset\}$ avec b se partitionne en : $\{q_0, \emptyset\}|\{q_2\}$
$\{q_0, \emptyset\}$ avec a se partitionne en : $\{q_0\}|\{\emptyset\}$
$\{q_3, q_4\}$ ne se partitionne ni avec a ni avec b.
On obtient finalement la partition :
$\emptyset|\{q_0\}|\{q_2\}|\{q_3, q_4\}$
On obtient un automate d'états finis déterministe à trois états :



Algo MINIMISATION

Donnee : un automate déterministe $A_1 = (\mathcal{Q}, \Sigma, \delta, q_0, F)$

Resultat : l'automate déterministe minimum
$A_2 = (\mathcal{Q}', \Sigma, \delta', Q_0', F')$.

Initialisation : $\mathscr{C} \leftarrow \{Q - F, F\}$; $b \leftarrow 1$;

SUPPRIMER de A_1 les états non atteignables ;

tant que b=1 faire

  $b \leftarrow 0$;

  pour chaque classe C de $\mathscr{C}$ faire

    pour chaque caractère x de Σ faire

      si par δ on n'aboutit pas dans une même classe de $\mathscr{C}$ alors

REMPLACER C dans $\mathscr{C}$ par les classes obtenues ; $b \leftarrow 1$;

$\delta' \leftarrow$ fonction de passage d'une classe de $\mathscr{C}$ à une autre ;

$F' \leftarrow$ ensemble des classes de $\mathscr{C}$ classes contenant au moins un état de F ;

Retourner$(\mathscr{C}, \Sigma, \delta', q_0, F')$.

5.5. Minimisation d'un AEF déterministe

Théorème

Pour un langage rationnel L donné, il existe un unique automate d'états fini déterministe minimum engendrant L.

Conséquence fondamentale :
Les langages rationnels sont non ambigüs.