Chapitre 4
Automates

Table des matières

1 Automates finis

1.1 Définition d'un automate fini.

Définition 1 :

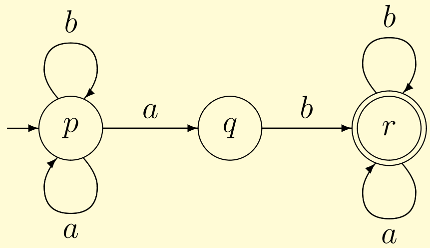
Un automate fini $\mathcal{A}$ est un quintuplet $\mathcal{A} = (Q, A, E, I, T)$ avec :

1. A alphabet fini appelé alphabet d'entrée.
2. Q un ensemble fini appelé ensemble d'états.
3. Deux sous-ensemble I et T de Q ; I est l'ensemble des états initiaux et T est l'ensemble des états finaux (ou terminaux).
4. Un sous-ensemble E de $Q \times A \times Q$, appelé ensemble des transitions.

Exemple 1 :

- Un alphabet $A = \{a, b\}$.
- Un ensemble $Q = \{p, q, r\}$
- États initiaux : $I = \{p\}$.
- États finaux : $T = \{r\}$
- L'ensemble des transitions : $E = \{(p, a, p), (p, b, p), (p, a, q), (q, b, r), (r, a, r), (r, b, r)\}$.

On a donc : $\mathcal{A} = (Q, A, E, I, T)$ Que l'on représentera par un graphe étiqueté :



1.2 Langages.

Définition 2 :

Un **calcul** c dans $\mathcal{A}$ est une suite de transitions telles que l'origine de chacune coïncide avec l'extrémité de la précédente, ce que l'on peut noter :

$$c = p_0 \xrightarrow{a_1} p_1 \xrightarrow{a_2} p_2 ... \xrightarrow{a_n} p_n$$

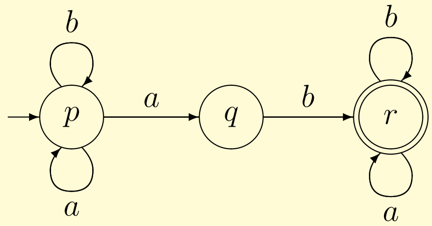
L'état p_0 est l'origine du calcul.
L'état p_n est l'extrémité.
La longueur est n. (le nombre de transitions)
L'étiquette de c est le produit des étiquettes des transitions : $a_1a_2...a_n$.
On peut noter :

$$c = p_0 \xrightarrow{a_1a_2...a_n} p_n$$

Un calcul est **réussi** si son origine est un état initial et son extrémité un état final.
Un mot de A^* est dit **reconnu** (ou **accepté**) par $\mathcal{A}$ si c'est l'étiquette d'un calcul réussi de $\mathcal{A}$.

Exemple 2 :

Déterminer des exemples de mot reconnu, et non reconnu par l'automate suivant :



Définition 3 :

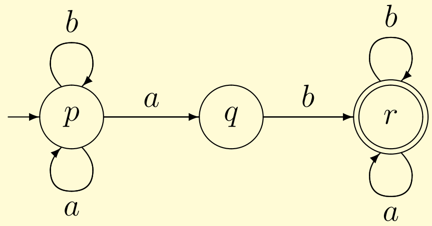
- Le langage reconnu par $\mathcal{A}$ est l'ensemble des mots reconnus par $\mathcal{A}$, on le note $L(\mathcal{A})$:

$$L(\mathcal{A}) = \left\{ f \in A^* | \exists p \in I, \exists q \in T, p \xrightarrow{f} p \right\}$$

- Deux automates sont **équivalents** s'ils reconnaissent le même langage.

Exemple 3 :

Déterminer de façon ensembliste le langage reconnu par l'automate suivant :



Définition 4 :

- On appelle **reconnaissable** un langage reconnu par un automate fini : Une partie L de A^* est reconnaissable s'il existe un automate fini $\mathcal{A}$ sur l'alphabet A tel que $L = L((\mathcal{A})$.
- La famille des langages reconnaissable de A^* est notée : $\text{Rec}A^*$.

<u>Exercice 1 :</u> Déterminer un automate fini reconnaissant :

1. le langage vide.
2. le singleton ϵ.
3. le singleton f, pour $f \in A^*$.

2 Propriétés

2.1 Image miroir d'un langage reconnaissable :

Définition 5 :

On appelle automate transposé de $\mathcal{A}$ l'automate, noté $\mathcal{A}^t$, tel que :
Si $\mathcal{A} = (Q, A, E, I, T)$, $\mathcal{A}^t = (Q, A, E^t, T, I)$,
Avec $E^t = \{(p, a, q) \in Q \times A \times Q | (q, a, p) \in E\}$
On a $L(\mathcal{A}^t) = [L(\mathcal{A})]^t = \{$ miroir de f,pour f élément de $L(\mathcal{A})\}$

Propriété 1 :

L'image miroir d'un langage reconnaissable est un langage reconnaissable. RecA^* est fermée par image miroir.

Preuve :

2.2 Union de langages reconnaissables :

Définition 6 :

Soient deux automates : $\mathcal{A}' = (Q', A, E', I', T')$ et $\mathcal{A}'' = (Q'', A, E'', I'', T'')$ avec Q' et Q'' disjoints.
On appelle union de $\mathcal{A}'$ et $\mathcal{A}''$ l'automate défini par :

$$\mathcal{A} = \mathcal{A}' \cup \mathcal{A}'' = (Q' \cup Q'', A, E' \cup E'', I' \cup I'', T' \cup T'')$$

On a alors $L(\mathcal{A}) = L(\mathcal{A}') \cup L(\mathcal{A}'')$

Propriété 2 :

L'union de deux langages reconnaissables est un langage reconnaissable. RecA^* est fermée par union finie.
Conséquence : toute partie finie de A^* est reconnaissable.

Preuve :

2.3 Intersection de langages reconnaissables :

Définition 7 :

Soient deux automates : $\mathcal{A}' = (Q', A, E', I', T')$ et $\mathcal{A}'' = (Q'', A, E'', I'', T'')$.
On appelle produit cartésien de $\mathcal{A}'$ et $\mathcal{A}''$ l'automate défini par :

$$\mathcal{A} = \mathcal{A}' \times \mathcal{A}'' = (Q' \times Q'', A, E, I' \times I'', T' \times T'')$$

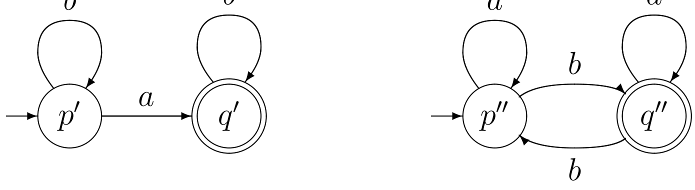
avec $E = \{((p', p''), a, (q', q'')) \,|(p', a, q') \in E' \text{ et } (p'', a, q'') \in E''\}$
On a alors $L(\mathcal{A}) = L(\mathcal{A}') \cap L(\mathcal{A}'')$

Propriété 3 :

L'intersection de deux langages reconnaissables est un langage reconnaissable. RecA^* est fermée par intersection.

Preuve :

Exercice 2 : On considère les automates $\mathcal{A}'$ et $\mathcal{A}''$ suivants :



1. Décrire les langages reconnus par les deux automates.

2. Construire l'automate obtenu par le produit cartésien.

3 Automate émondé.

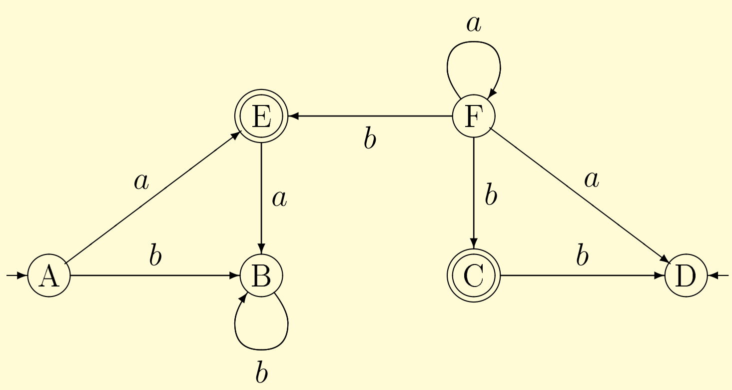
Définition 8 :

Soit automate fini $\mathcal{A} = (Q, A, E, I, T)$.

- Un état q de $\mathcal{A}$ est **accessible à partir de** l'état p s'il existe un calcul dans $\mathcal{A}$ dont l'origine est p et dont l'extrémité est q.
- On dira que l'état q de $\mathcal{A}$ est **accessible** s'il est accessible à partir d'un état initial.
- Un état p de $\mathcal{A}$ est **co-accessible à** l'état q s'il existe un calcul dans $\mathcal{A}$ dont l'origine est p et dont l'extrémité est q.
- On dira que l'état p de $\mathcal{A}$ est **co-accessible** s'il est co-accessible à un état final.
- Un état p est **utile** s'il est à la fois accessible et co-accessible.
- Un automate est **accessible** si tous ses états le sont.
- Un automate est **co-accessible** si tous ses états le sont.
- Un automate est **émondé** si tous ses états utiles.

Exemple 4 :

Dans l'automate suivant, déterminer les états **accessibles** , les états **co-accessible** et les états **utiles** :



Propriété 4 :

Un état p est utile si, et seulement si, il existe un calcul réussi qui passe par p.

Preuve :

Définition 9 :

Soit automate fini $\mathcal{A} = (Q, A, E, I, T)$.
L'émondé de $\mathcal{A}$ est l'automate $\mathcal{A}_{em} = (Q_{em}, A, E_{em}, I_{em}, T_{em})$ tel que Q_{em} est l'ensemble des états utiles de $\mathcal{A}$, $I_{em} = I \cap Q_{em}$, $T_{em} = T \cap Q_{em}$ et $E_{em} = E \cap (Q_{em} \times A \times Q_{em})$.

Propriété 5 :

Tout automate fini est équivalent à un automate fini émondé.

Preuve :

Propriété 6 :

L'ensemble des préfixes (resp. des suffixes, des facteurs) d'un langage reconnaissable est reconnaissable.

Preuve :

4 Automate complet.

Définition 10 :

Soit automate fini $\mathcal{A} = (Q, A, E, I, T)$.
On dira que $\mathcal{A}$ est complet si, pour tout p dans Q et tout a de A, il existe q dans Q tel que (p, a, q) est une transition, soit $(p, a, q) \in E$.

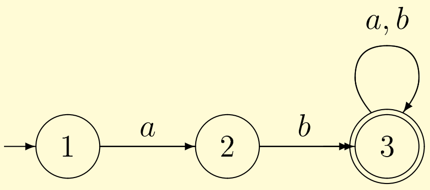
Remarque : Il n'y donc pas de blocage dans une automate complet.

Définition 11 :

Soit automate fini $\mathcal{A} = (Q, A, E, I, T)$, si $\mathcal{A}$ n'est pas complet, le **complété** de $\mathcal{A}$ est l'automate fini $\mathcal{A}_{comp}(Q', A, E', I, T)$ avec $Q' = Q \cup \{z\}$ où $z \notin Q$, et $E' = E \cup \{(z, a, z) | a \in A\} \cup \{(p, a, z) |$ il n'existe pas de $q \in Q,$ tq $(p, a, q) \in E\}$. z est appelé état puits.
Si $\mathcal{A}$ est complet, on pose $\mathcal{A}_{comp} = \mathcal{A}$.

Exemple 5 :

Déterminer le complété de l'automate suivant :



Propriété 7 :

Tout automate est équivalent à un automate accessible complet.

Preuve :

5 Critère de reconnaissabilité :

Propriété 8 :

Lemme de l'étoile :
Si un langage L de A^* est reconnaissable, alors il existe un entier N tel que, pour tout f de L de longueur supérieur à N, il existe une factorisation de f : $f = u.v.w$ avec v non vide, telle que, pour tout n de $\mathbb{N}$, le mot $u.v^n.w$ est dans L. (soit $u.v^n.w \in L, \forall n \in \mathbb{N}$)

Preuve :

Propriété 9 :

Lemme de l'étoile :
Si un langage L de A^* est reconnaissable, alors il existe un entier N tel que, pour tout f de L tel que $f = uvw$ avec v de longueur supérieur à N, il existe une factorisation de f : $f = u.v_1v_2v_3.w$ avec $v_2 \neq \epsilon$ telle que : pour tout n de $\mathbb{N}$, le mot $u.v_1.v_2^n.v_3.w$ est dans L.
(soit $u.v_1.v_2^n.v_3.w \in L$, $\forall n \in \mathbb{N}$)

Exercice 3 : Montrer que le langage $L = \{a^nb^n | n \in \mathbb{N}\}$ n'est pas reconnaissable.

6 Automates déterministes.

6.1 Table de transition :

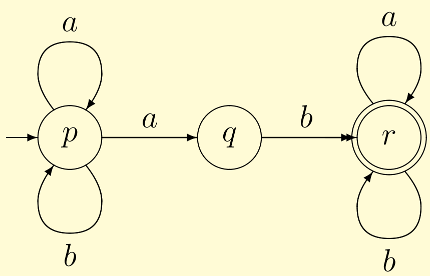
Définition 12 :

Soit automate fini $\mathcal{A} = (Q, A, E, I, T)$.
On appelle δ la fonction de transition :

$$\begin{array}{rcccl} \delta & : & Q \times A & \to & \mathscr{P}(Q) \\ & & (p, a) & \mapsto & \{q \mid (p, a, q) \in E\} \end{array}$$

On appelle table de transition le tableau à double entrée représentant les valeurs de la fonction de transition.

Exemple 6 :



La table de transition est :

	${}^{I}p$	q	r^{T}
a			
b			

6.2 Définition d'un automate déterministe :

Définition 13 :

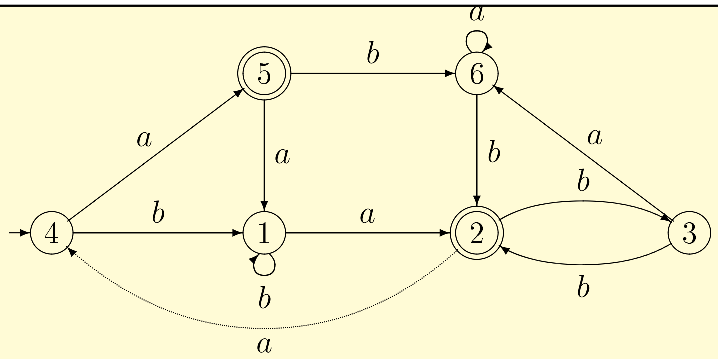
Soit automate fini $\mathcal{A} = (Q, A, E, I, T)$.
On dira que $\mathcal{A}$ est déterministe (AFD) si :

- La fonction de transition est une application de $Q \times A$ dans Q, i.e pour tout p de Q, et tout a de A, $\delta(p, a)$, s'il est défini, est un singleton.
- Il y a un unique état initial, i.e I est un singleton.

On note alors $\mathcal{A} = (Q, A, \delta, i, T)$.
Autrement dit : $\mathcal{A}$ admet un unique état initial, et pour tous les états q, q' et q'', et pour toute lettre a :

$$((q, a, q') \in E) \wedge ((q, a, q'') \in E) \Rightarrow q' = q''$$

Exemple 7 :



Propriété 10 :

Soit $\mathcal{A}$ un AFD, et soit u un mot.

- Pour tout état q de $\mathcal{A}$, il existe au plus un chemin étiqueté u partant de q.
- Si $u \in L(\mathcal{A})$, alors u est l'étiquette d'un seul chemin réussi.

Preuve :

Définition 14 :

Soit $\mathcal{A} = (Q, A, \delta, i, T)$ un AFD,
on définie une fonction de transition étendue $\delta^* : Q \times A^* \to Q$ qui associe à tout couple $(q, u) \in Q \times A^*$ l'état q' si il existe tel que $q \xrightarrow{u} q'$.
Donc $\delta^*(p, f)$ est définie de façon inductive par :

$$\begin{array}{lrcl} \forall p \in Q & \delta^*(p, \epsilon) & = & p \\ \forall p \in Q, \forall f \in A^*, \forall a \in A & \delta^*(p, fa) & = & \left\{ \begin{array}{l} \delta(\delta^*(p, f), a) \text{ si } \delta^*(p, f) \text{ et } \delta(\delta^*(p, f), a) \text{ existent.} \\ \text{non définie sinon.} \end{array} \right. \end{array}$$

On a alors, $\forall p \in Q$, $\forall f, g \in A^*$, $\delta^*(p, fg) = \delta^*\left(\delta^*(p, f), g\right)$.
Ainsi : $L(\mathcal{A}) = \{f \in A^* | \delta^*(i, f) \in T\}$

6.3 Déterminisation :

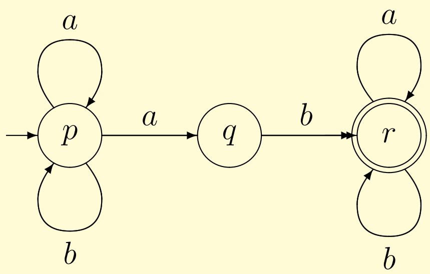
Propriété 11 :

Tout automate $\mathcal{A}$ est équivalent à un automate déterministe complet $\mathcal{B}$.
Si $\mathcal{A}$ est un automate avec n états, on peut construire $\mathcal{B}$ avec au plus 2^n états.

Preuve :

Exemple 8 :

Déterminiser l'automate suivant " par la méthode des sous-ensembles". On ne s'intéresse qu'aux éléments de $\mathscr{P}(Q)$ accessibles.



7 Automates normalisés :

Définition 15 :

Soit automate fini $\mathcal{A} = (Q, A, E, I, T)$. On dira que $\mathcal{A}$ est normalisé si :

- $I = \{i\}$ est un singleton qui n'est l'extrémité d'aucune transition de $\mathcal{A}$.
- $T = \{t\}$ est un singleton qui n'est l'origine d'aucune transition de $\mathcal{A}$.

Remarque : L'automate qui reconnaît le seul mot vide n'est pas normalisé.

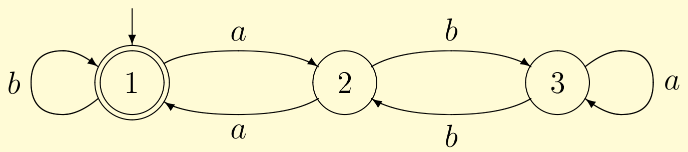
Propriété 12 :

Soit automate fini $\mathcal{A} = (Q, A, E, I, T)$. $L(\mathcal{A})\backslash\{\epsilon\}$ est reconnu par un automate fini normalisé.

Preuve :

Exemple 9 :

Déterminer l'automate normalisé de l'automate suivant :



8 Fermeture par concaténation :

Propriété 13 :

Le produit de deux langages reconnaissables est reconnaissable.

Preuve :

Exercice 4 : Chercher l'automate qui reconnaît a^*b^* en utilisant la méthode décrite précédemment, puis le déterminiser.

9 Fermeture par itération :

Propriété 14 :

L'étoile d'un langage reconnaissable est reconnaissable.

Preuve :

10 Le théorème de Kleene :

Théorème 1 :
Un langage de A^* est rationnel si, et seulement si, il est reconnu par un automate fini sur A. On a donc :

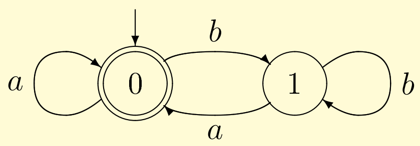
$$\mathrm{Rat}(A^*) = \mathrm{Rec}(A^*)$$

Remarque : On admettra ce théorème.

Méthode de recherche d'une expression rationnelle qui dénote le langage reconnu par un automate fini :
On considère un automate fini $\mathcal{A} = (Q, A, E, I, T)$.
On note $L_p = \left\{ f | \exists t \in T, p \xrightarrow{f} t \right\}$.
On a donc $L(\mathcal{A}) = \bigcup\limits_{p \in I} L_p$.
Un mot f de A^* appartient à L_p si :

- $f = \epsilon$ et $p \in T$
  ou
- $f = ag$, où $(p, a, q) \in E$ et $g \in L_q$

Exemple 10 :
On considère l'automate suivant :



On a donc :
$$\begin{cases} L_0 & = & aL_0 + bL_1 + \epsilon \\ L_1 & = & aL_0 + bL_1 \end{cases}$$
Il s'agit maintenant de résoudre ce système linéaire grâce au lemme d'Arden : Soient K et L deux partie de A^* , où K ne contient pas le mot vide.
Alors K^*L est l'unique solution de l'équation d'inconnue X :

$$X = KX + L$$

La résolution du système donne donc $L_1 = b^*aL_0$. On remplace dans la première ligne : $L_0 = b^*aL_0 + \epsilon$. Soit $L = (b^*a)^*$, ce qui nous donne l'expression rationnelle cherchée.

11 L'algorithme de Berry-Sethi et automate de Glushkov

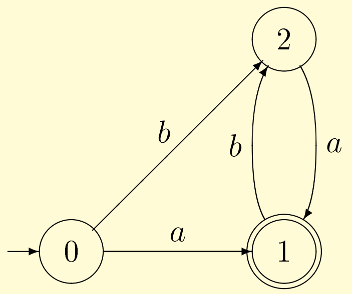
Définition 16 :

Un automate déterministe $\mathcal{A} = (Q, A, \delta, i, T)$ est dit **local** lorsque pour toute lettre a il existe un état $q \in Q$ tel que pour tout $p \in Q$, $\delta(p, a)$ n'est pas définie ou $\delta(p, a) = q$.
Il est dit **standard** lorsqu'il n'existe pas de lettre a et d'état q tel que $\delta(p, a) = i$.

Remarque : Un automate est local lorsque pour chaque lettre a toutes les transitions étiquetées par a arrivent dans un même état, et il est standard lorsqu'il n'existe pas de transition aboutissant à l'état initial.

Exemple 11 :

Un automate local standard :



Propriété 15 :

Si L est un langage local, le langage $L \setminus \{\epsilon\}$ est reconnaissable par un automate local standard.

Preuve :

Exercice 5 : Construire par la méthode précédente un automate reconnaissant le langage défini par : $L = (a + b)^*c$.

Construction de l'automate de Glushkov

On considère une expression rationnelle e ne comportant ni le symbole $\varnothing$, ni ϵ, et composée de n lettres, non nécessairement distinctes.
L'idée est de procéder a un marquage de ces lettres dans l'ordre de leur apparition. On note alors c_k la k-ième lettre, et on obtient ainsi une expression linéaire dont les lettres sont les c_k.
On procède alors à la création de l'automate locale standard.
Il suffit ensuite de revenir aux lettres initiales connaissant la fonction de marquage.
L'automate obtenue s'appelle l'automate de **Glushkov**.

Exemple 12 :

On considère l'expression rationnelle suivante :

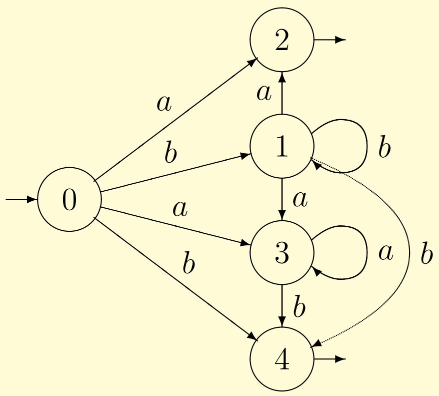
$$e = b^*(a + (a^*b))$$

On a

$$e' = ((c_1)^*.(c_2 + ((c_3)^*.c_4)))$$

On cherche alors les ensembles P, S, et F :

- $P(e') = \{c_1, c_2, c_3, c_4\}$
- $S(e') = \{c_2, c_4\}$
- $F(e') = \{c_1c_1, c_1c_2, c_1, c_3, c_1c_4, c_3c_3, c_3c_4\}$



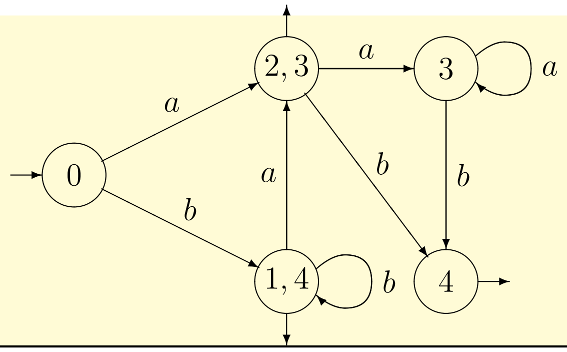
Le tableau des transitions est

	0	1	2	3	4
a	$2,3$	$2,3$		3	
b	$1,4$	$1,4$		4	

L'algorithme de déterminisation par méthode des sous-ensembles donne la table suivante

	0	$2,3$	$1,4$	3	4
a	$2,3$	3	$2,3$	3	
b	$1,4$	4	$1,4$	4	

et l'automate est donc



12 Automate généralisé. Algorithme de Brzozowski et McCluskey

Définition 17 :

Un automate généralisé est défini comme un automate fini non déterministe traditionnel, avec les particularités suivantes :

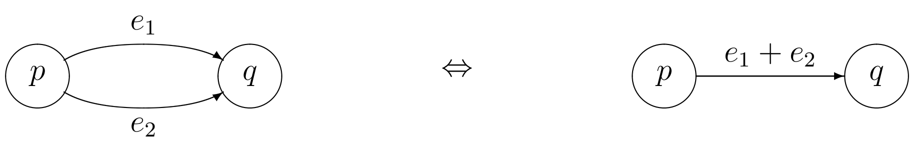
- il possède un seul état initial i et un seul état final f.
- les transitions sont étiquetées par des expressions rationnelles.
- aucune transition n'entre dans i et aucune transition ne sort de l'état final j.

Algorithme de Brzozowski et McCluskey :
Le but de cet algorithme est, pour un automate déterministe, de construire une expression rationnelle associée.
On travaille ainsi sur l'automate généralisé.
Considérons un automate déterministe $\mathcal{A} = (Q, A, \delta, q_0, T)$:

1. On ajoute deux nouveaux états i et f. On relie i à l'état initial q_0 par une transition étiquetée par ϵ. On relie les états terminaux à f par une transition ϵ. Le but est d'obtenir un automate standard qui n'a qu'un état final.

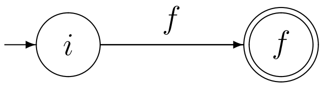
2. On simplifie l'automate en appliquant les deux méthodes ci-dessus autant que possible.

   - S'il existe deux états p, q et deux transition $p \xrightarrow{e_1} q$ et $p \xrightarrow{e_2} q$ on les remplace par $p \xrightarrow{e_1+e_2} q$.(p peut-être égal à q) :



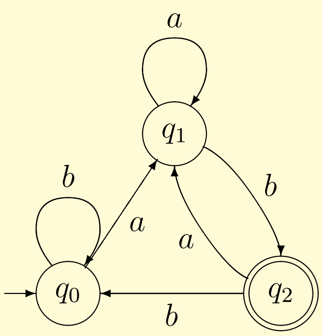
   - On supprime un état q (différent de i et f), et pour tous les états (p, r) distincts de q (mais éventuellement égaux) tels qu'il existe des transitions $p \xrightarrow{e_1} q$ et $q \xrightarrow{e} q$, et $p \xrightarrow{e_2} r$ on ajoute une transition $p \xrightarrow{e_1 e^* e_2} r$.

En réitérant autant de fois que possible, on aboutit à un automate :

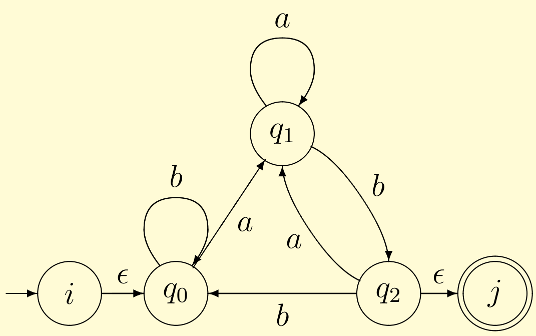


Exemple 13 :

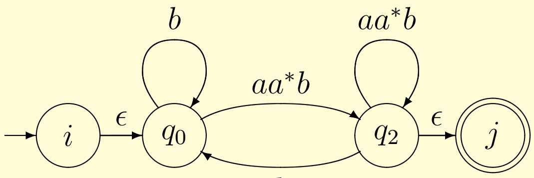
On considère l'automate suivant :



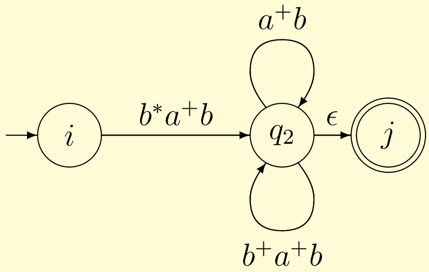
1. On ajoute les états i et j :



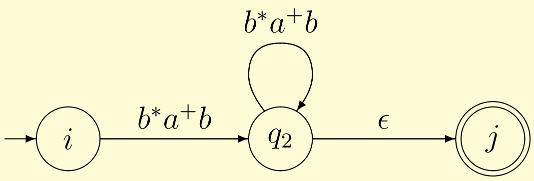
2. On supprime l'état q_1 :



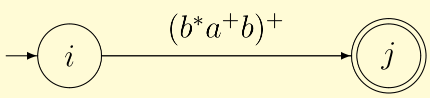
3. On supprime l'état q_0 :



4. On simplifie, en remarquant que
$a^+b + b^+a^+b = b^*a^+b$



5. On obtient alors :



On en déduit alors l'expression rationnelle : $e = (b^*a^+b)^+$.