Chapitre 4

Automates

Version avec preuves.

Table des matières

1 Automates finis

1.1 Définition d'un automate fini.

Définition 1 :

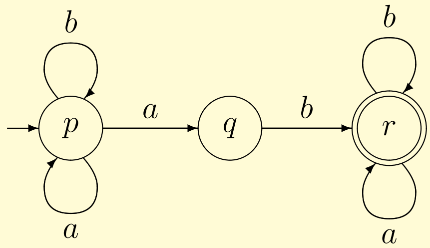
Un automate fini $\mathcal{A}$ est un quintuplet $\mathcal{A} = (Q, A, E, I, T)$ avec :

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 :

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

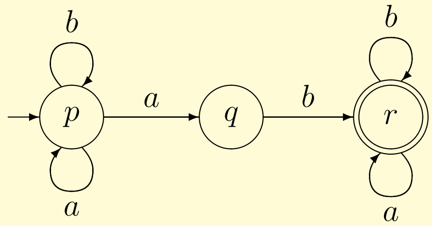
L'état p_0 est l'origine du calcul.
L'état p_n est l'extrémité.
La longueur est n. (le nombre de transitions)
L'étiquette de c est le produit des étiquettes des transitions : $a_1 a_2 ... a_n$.
On peut noter :

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

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 :

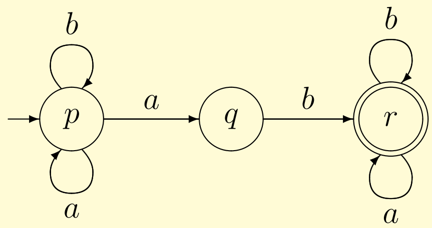
- Le langage reconnu par $\mathcal{A}$ est l'ensemble des mots reconnus par $\mathcal{A}$, on le note $L(\mathcal{A})$:

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

- 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 :

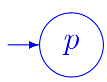
- On appelle **reconnaissable** un langage reconnu par un automate fini : Une partie L de A^* est reconnaissable s'il existe un automate fini $\mathcal{A}$ sur l'alphabet A tel que $L = L((\mathcal{A})$.
- La famille des langages reconnaissable de A^* est notée : $\text{Rec}A^*$.

Exercice 1 : Déterminer un automate fini reconnaissant :

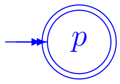
1. le langage vide.
2. le singleton ϵ.
3. le singleton f, pour $f \in A^*$.

Preuve

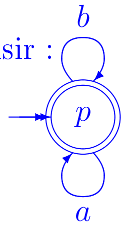
- Pour $L = \emptyset$, on peut choisir :



- Pour $L = \{\epsilon\}$, on peut choisir :



- Pour $L = \{a, b\}^*$, on peut choisir :



- Pour $L = \{f\}$, avec $f \in A^*$ on peut choisir $\mathcal{A}$ est un quintuplet $\mathcal{A} = (Q, A, E, I, T)$ avec :

1. A l'alphabet.
2. Q l'ensemble des préfixes de f.
3. $I = \{\epsilon\}$ et $T = \{f\}$.
4. $E = \{(g, a, ga) | a \in A, g$ prefixe de f, ga prefixe de $f\}$.

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 :

Preuve

Dans l'automate $\mathcal{A}^t = (Q, A, E^t, T, I)$, les transitions sont inversées, avec T les états initiaux, et I les états finals
En notant $m = a_1 ... a_n$ un mot, éventuellement nul, de $L(\mathcal{A})$, on a $m^t = a_n ... a_1$ dans $L(\mathcal{A})^t$

$$\begin{aligned} m^t \in L(\mathcal{A}^t) &\Leftrightarrow \exists t \in T, \exists i \in I, \text{ tel que } t \xrightarrow[\text{avec } E^t]{a_n ... a_1} i \\ &\Leftrightarrow \exists t \in T, \exists i \in I, \text{ tel que } i \xrightarrow[\text{avec } E]{a_1 ... a_n} t \\ &\Leftrightarrow m \in L(\mathcal{A}) \\ &\Leftrightarrow m^t \in L(\mathcal{A})^t \end{aligned}$$

On a donc bien $L(\mathcal{A}^t) = [L(\mathcal{A})]^t$

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 :

Preuve

On peut supposer que les états sont disjoints, quitte à les renommer...
Toutes les transitions de E' et E'' sont donc disjointes.

$$\begin{aligned} m \in L(\mathcal{A}' \cup \mathcal{A}'') &\Leftrightarrow \exists i \in I' \cup I'', \exists t \in T' \cup T'', \text{ tel que } i \xrightarrow[\text{avec } E' \cup E'']{a_1 \ldots a_n} t \\ &\Leftrightarrow \left(\exists i \in I', \exists t \in T', \text{ tel que } i \xrightarrow[\text{avec } E']{a_1 \ldots a_n} t\right) \vee \left(\exists i \in I'', \exists t \in T'', \text{ tel que } i \xrightarrow[\text{avec } E'']{a_1 \ldots a_n} t\right) \\ &\Leftrightarrow (m \in L(\mathcal{A}')) \vee (m \in L(\mathcal{A}'')) \\ &\Leftrightarrow m \in L(\mathcal{A}') \cup L(\mathcal{A}'') \end{aligned}$$

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 :

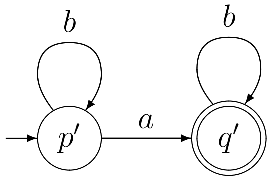
L'intersection de deux langages reconnaissables est un langage reconnaissable. RecA^* est fermée par intersection.

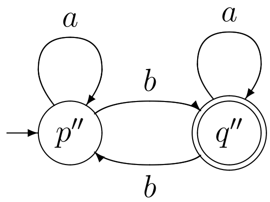
Preuve :

Preuve

$$\begin{aligned} m \in L(\mathcal{A}) &\Leftrightarrow \exists (i', i'') \in I' \times I'', \exists (t', t'') \in T' \times T'', \text{ tel que } (i', i'') \xrightarrow[\text{avec } E]{a_1 \ldots a_n} (t', t'') \\ &\Leftrightarrow \exists (i', i'') \in I' \times I'', \exists (t', t'') \in T' \times T'', \exists (p'_k, p''_k) \in Q' \times Q'' \\ &\qquad \text{tel que } (i', i'') \xrightarrow{a_1} (p'_1, p''_1) \xrightarrow{a_2} \ldots \xrightarrow{a_n} (t', t'') \\ &\Leftrightarrow \exists (i', i'') \in I' \times I'', \exists (t', t'') \in T' \times T'', \exists (p'_k, p''_k) \in Q' \times Q'' \\ &\qquad \text{tel que } \left(i' \xrightarrow{a_1} p'_1 \xrightarrow{a_2} \ldots \xrightarrow{a_n} t'\right) \wedge \left(i'' \xrightarrow{a_1} p''_1 \xrightarrow{a_2} \ldots \xrightarrow{a_n} t''\right) \\ &\Leftrightarrow (m \in L(\mathcal{A}')) \wedge (m \in L(\mathcal{A}'')) \\ &\Leftrightarrow m \in L(\mathcal{A}') \cap L(\mathcal{A}'') \end{aligned}$$

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.

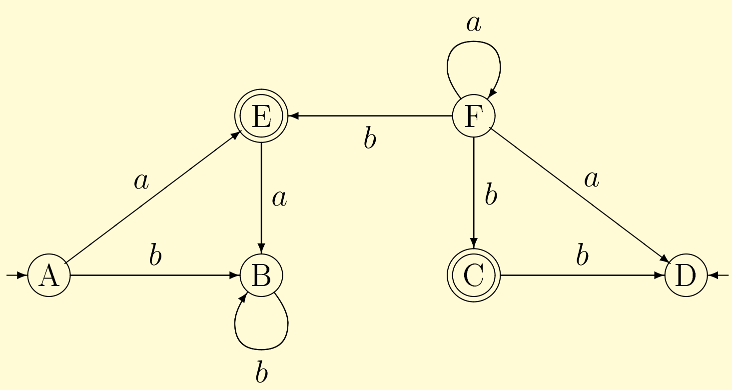
Définition 8 :

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 :

Preuve

Si p est utile, il est accessible donc il existe un chemin d'étiquette u allant d'un état initial i à p. De même, il est co-accessible donc il existe un chemin d'étiquette v allant de p à un état final f. Donc le chemin $i \xrightarrow{u} p \xrightarrow{v} f$ est un chemin réussi.
Réciproquement, s'il existe un tel chemin, p est forcement utile.

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 :

Preuve

Si $u \in L(\mathcal{A})$, donc u est l'étiquette d'un chemin réussi, dont les états sont donc tous utiles, donc ils appartiennent à Q_{em}. Donc $u \in L(\mathcal{A}_{em})$.
Réciproquement, $u \in L(\mathcal{A}_{em})$, comme $Q_{em} \subset Q$, $I_{em} \subset I$, $T_{em \subset T}$ et $E_{em} \subset E$, donc tout chemin de $\mathcal{A}_{em}$ est un chemin de $\mathcal{A}$. Ainsi, $u \in L(\mathcal{A})$.

Propriété 6 :

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

Preuve :

Preuve

il suffit de partir d'un automate émondé, et de remplacer T par Q...

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 $(p, a, q) \in E$.

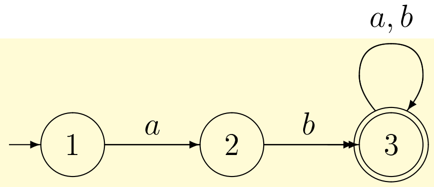
Remarque : Il n'y donc pas de blocage dans une automate complet.

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 :

Preuve

Si $u \in L(\mathcal{A})$, alors forcément, $u \in L(\mathcal{A}_{comp})$ car $Q \in Q'$ et $E \in E'$ et les états initiaux et finals sont les mêmes.
Réciproquement, si $u \in L(\mathcal{A}_{comp})$, alors le chemin dont u est étiquette ne contient par z, car tout chemin issu de z aboutit à z, donc $u \in L(\mathcal{A})$.

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 :

Preuve

On considère un langage reconnaissable L.

- **Premier cas :** si L est fini, alors on choisit $N > \max_{f \in L} |f|$. Alors il n'existe pas de mot f appartenant à L tel que $|f| \geq N$, donc l'assertion à vérifier est triviale.
- **Deuxième cas :** si L est infini. Il existe un automate fini $\mathcal{A} = (Q, A, E, I, T)$ qui reconnaît L. Le nombre d'états de $\mathcal{A}$ étant fini, on note N ce nombre augmenté de 1.($N = |Q| + 1$)
Soit maintenant f un mot de L tel que $f \geq N$. f est reconnu par $\mathcal{A}$, donc il existe un chemin $q_0 \xrightarrow{f} q_n$ réussi dans $\mathcal{A}$, étiqueté par f. Autrement dit, q_0 est un état initial de $\mathcal{A}$, et en partant de cet état on peut lire l'une après l'autre toutes les lettres de f en terminant la lecture par un état q_n acceptant de $\mathcal{A}$. Ceci impose $n = |f| \geq N$, ce qui assure par la définition de N qu'un certain état $q = q_i$ intervient au moins deux fois dans le chemin parcouru pour lire f.
On peut choisir i minimal. q est ainsi décrit comme la première occurrence du premier état q répété au cours du chemin $q_0 \xrightarrow{f} q_n$. On note alors v le mot qui est lu entre q_i et q_j, la deuxième occurrence de q dans le chemin en question (avec donc $n \geq j > i$). On note u le mot lu entre q_0 et q_i , et w le mot lu entre q_j et q_n . Ainsi, $f = u.v.w$.
On remarque que u et w peuvent être vides, mais v ne l'est pas. Donc, pour tout n de $\mathbb{N}$, le mot $u.v^n.w$ est un calcul réussi, donc dans L.

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 $L = \{a^n b^n | n \in \mathbb{N}\}$ n'est pas reconnaissable.

Preuve

On prend $f = a^N b^N$...

6 Automates déterministes.

6.1 Table de transition :

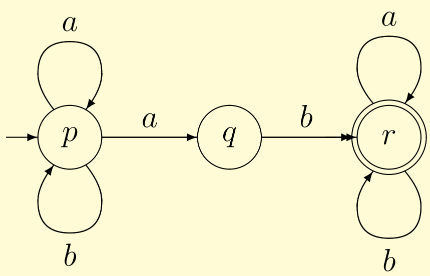
Définition 12 :

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

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

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 :

	${}^Ip$	q	r^T
a			
b			

6.2 Définition d'un automate déterministe :

Définition 13 :

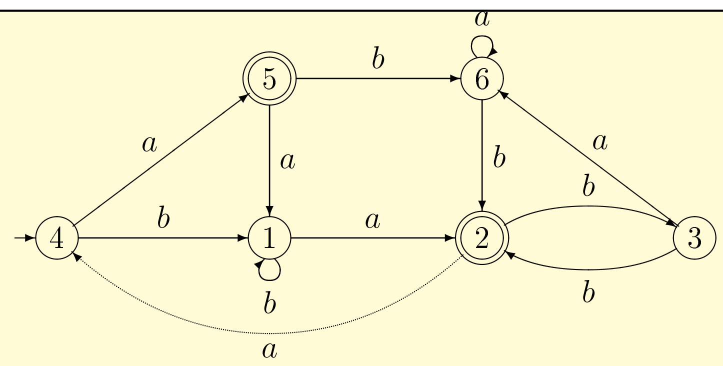
Soit automate fini $\mathcal{A} = (Q, A, E, I, T)$.
On dira que $\mathcal{A}$ est déterministe (AFD) si :

- 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 :

Preuve

Récurrence sur la longueur de u :

- si $|u| = 1$, c'est la définition d'un AFD.
- On suppose que la propriété soit vraie pour tous mots de longueur n, soit u un mot de longueur $n + 1$, on pose $u = u'a$. Il y a donc au plus un chemin étiqueté u' partant de q :
  - Si il n'y en a pas, il n'y a donc pas de chemin pour u.
  - Sinon, il y en a un : il existe q' tel que $q \xrightarrow{u'} q'$ et il y a au maximum une transition tel que $q' \xrightarrow{a} q''$

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 :

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

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 :

Preuve

Soit $\mathcal{A} = (Q, A, \delta, I, T)$ un automate fini, on pose $\mathcal{A}_D = (\mathscr{P}(Q), A, \delta_D, \{I\}, T_D)$, avec :

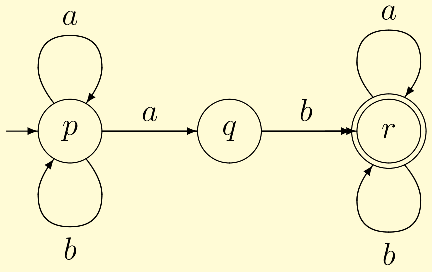
$$\delta_D(S, a) = \bigcup_{s \in S} \delta(s, a) \text{ et } T_D = \{S \subset Q | S \cap T \neq \emptyset\}$$

Cet automate est déterministe.
Montrons l'équivalence, $\forall f \in A^*$:

$$\begin{array}{lll} f \in L(\mathcal{A}) & \Leftrightarrow & \exists i \in I \text{ tel que } \delta^*(i, f) \in T \\ & \Leftrightarrow & \delta_D^*(I, f) \cap T \neq \emptyset \\ & \Leftrightarrow & \delta_D^*(I, f) \in T_D \\ & \Leftrightarrow & f \in L(\mathcal{A}_D) \end{array}$$

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.

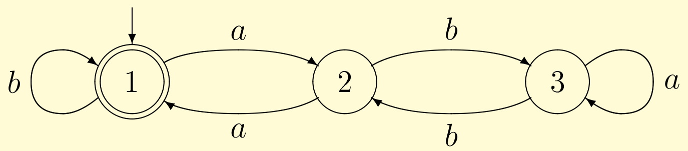
Preuve :

Preuve

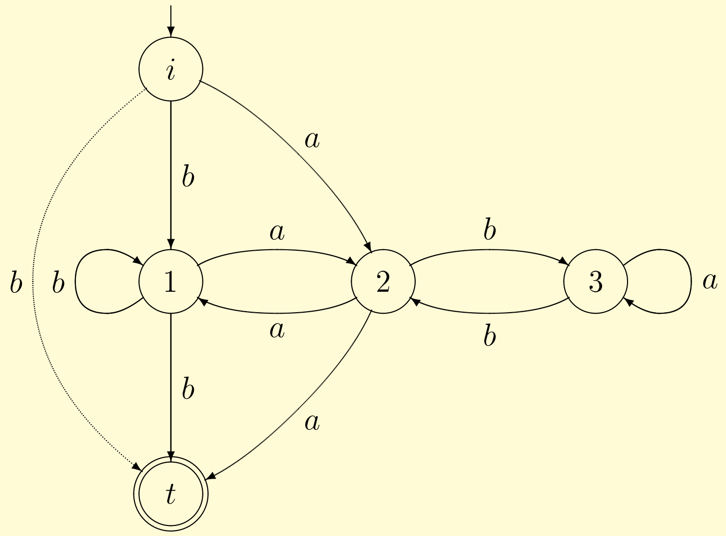
Soit $\mathcal{A} = (Q, A, E, I, T)$, il suffit de rajouter deux états i et t n'appartenant pas à Q :
On pose $\mathcal{A}_{nor} = (Q \cup \{i, t\}, A, E \cup E', \{i\}, \{t\})$, avec :
$E' = \{(i, a, q) | \exists p \in I \text{ tel que } (p, a, q) \in E\}$
$\cup \{(p, a, t) | \exists q \in T \text{ tel que } (p, a, q) \in E\} \cup$
$\{(i, a, t) | \exists p \in I, \exists q \in T \text{ tel que } (p, a, q) \in E\}$

Exemple 9 :

Déterminer l'automate normalisé de l'automate suivant :



Preuve



8 Fermeture par concaténation :

Propriété 13 :

Le produit de deux langages reconnaissables est reconnaissable.

Preuve :

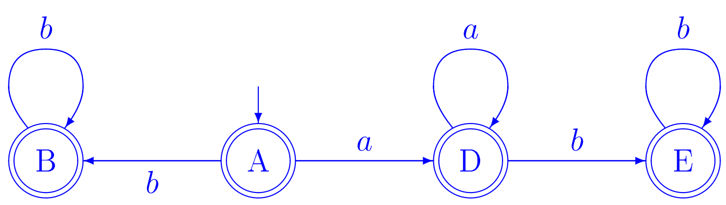
Preuve

On considère deux langages L et K. Si L et K sont privés du mot vide, il suffit alors de considérer les automates normalisés $\mathcal{A}_L$ et $\mathcal{B}_K$ et de les mettre "bout à bout".
Si l'un (ou les deux) contient le mots vide, on travaille alors sur l'union : $L = L \setminus \epsilon + \epsilon$.

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

Preuve

On trouve après déterminisation :



9 Fermeture par itération :

Propriété 14 :

L'étoile d'un langage reconnaissable est reconnaissable.

Preuve :

Preuve

On a la relation : $L^* = (L \setminus \epsilon)^*$, il suffit donc de s'intéresser aux langages ne contenant pas le mot vide. On sait trouver un automate normalisé, il suffit alors d'identifier l'état initial à l'état final.

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 :

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

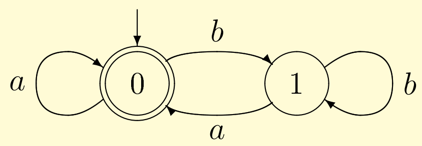
Remarque : On admettra ce théorème.

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

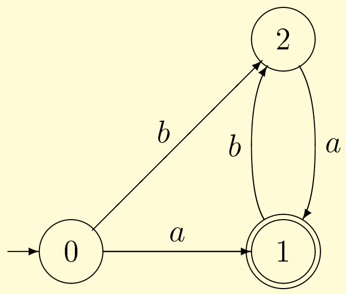
Définition 16 :

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 :

Preuve

Soit L un langage local. On note $P = P(L)$, $S = S(L)$ et $F = F(L)$ respectivement l'ensemble des premières lettre, des dernières lettres et des facteurs de deux lettres.
L'idée est de construire un automate dont les états sont les lettres de l'alphabet et un état initiale noté ϵ :
On pose alors : $\mathcal{A} = (A \cup \{\epsilon\}, A, \delta, \epsilon, S)$ où δ est définie par :

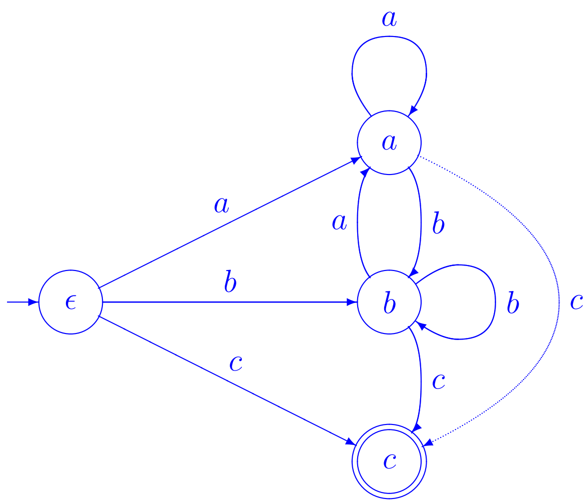
$$\begin{cases} \forall a \in P, & \delta(\epsilon, a) & = & a \\ \forall ab \in F, & \delta(a, b) & = & b \end{cases}$$

Par construction, le mot $u = a_1 a_2 ... a_n$ est reconnu par $\mathcal{A}$ si et seulement si $a_1 \in P$, $a_i a_{i+1} \in F$ et $a_n \in S$.

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

Preuve

On a $P = \{a, b, c\}$, $S = \{c\}$, et $F = \{aa, ab, ac, ba, bb, bc\}$, ce qui donne :



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 :

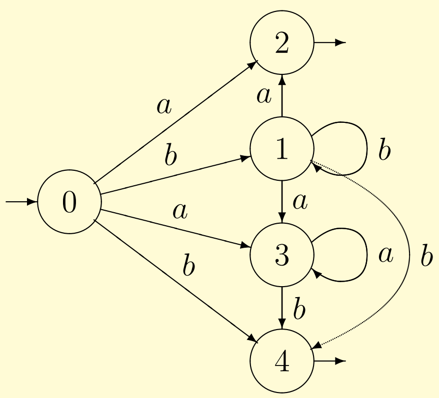
$$e = b^*(a + (a^*b))$$

On a

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

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

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



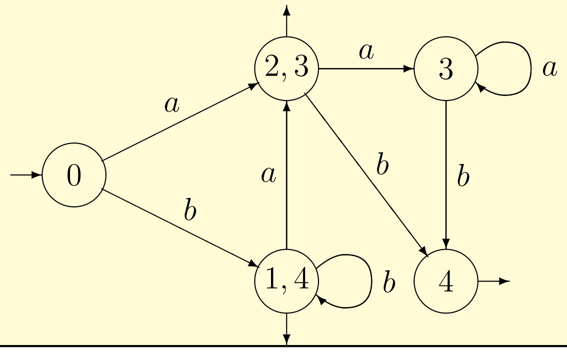
Le tableau des transitions est

	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.

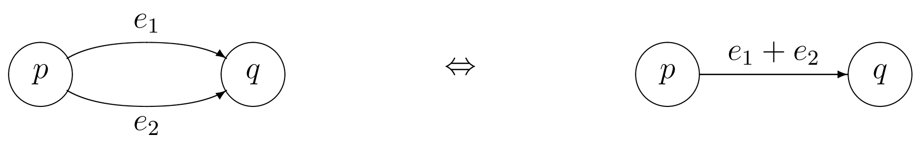
Algorithme de Brzozowski et McCluskey :

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 $\mathcal{A} = (Q, A, \delta, q_0, T)$:

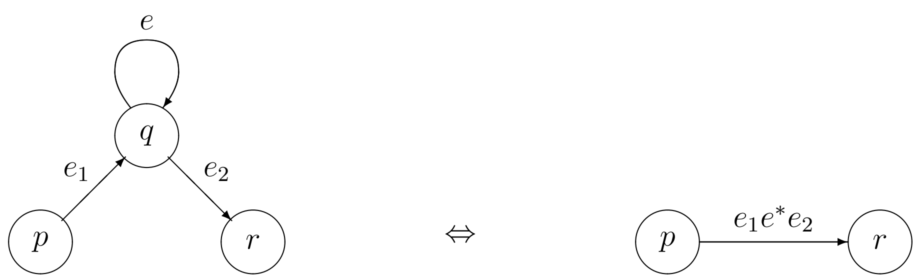
1. On ajoute deux nouveaux états i et f. On relie i à l'état initial q_0 par une transition étiquetée par ϵ. On relie les états terminaux à f par une transition ϵ. Le but est d'obtenir un automate standard qui n'a qu'un état final.

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

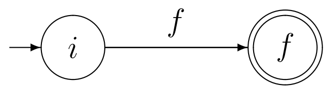
   - S'il existe deux états p, q et deux transition $p \xrightarrow{e_1} q$ et $p \xrightarrow{e_2} q$ on les remplace par $p \xrightarrow{e_1+e_2} q$.(p peut-être égal à q) :



- 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 $p \xrightarrow{e_1 e^* e_2} r$.

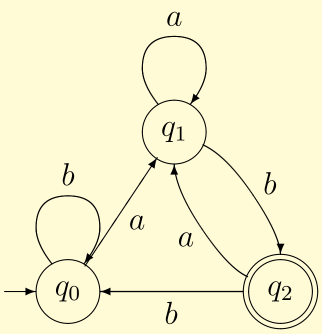


En réitérant autant de fois que possible, on aboutit à un automate :

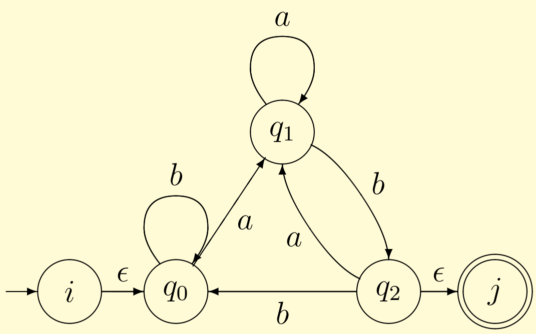


Exemple 13 :

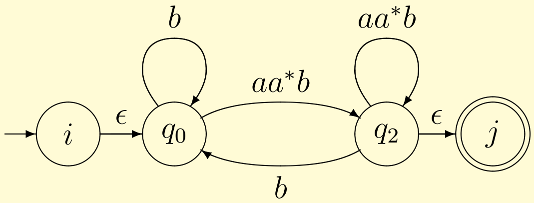
On considère l'automate suivant :



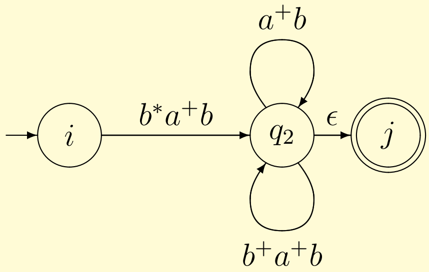
1. On ajoute les états i et j :



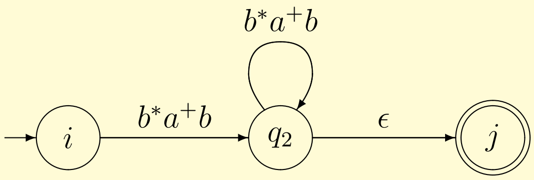
2. On supprime l'état q_1 :



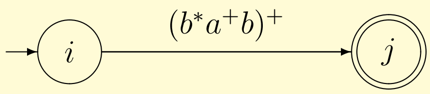
3. On supprime l'état q_0 :



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