Automates standard, émondés et propriétés de fermeture

1 Automate standard

Un automate $M = (\Sigma, Q, I, F, \delta)$ est *standard* s'il possède un unique état initial non réentrant :

$I = \{i\}$
$\forall p \in Q,\ \forall a \in \Sigma,\ |\delta(p, a, i)| = 0$

Exemple 1 :

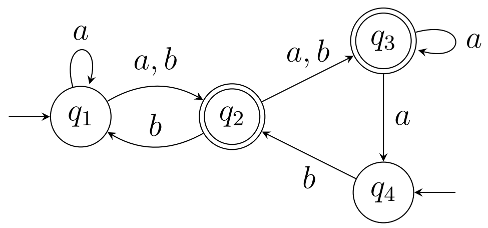


Figure 1 – Automate M_1

l'automate M_1 n'est pas standard car il a deux états initiaux q_1 et q_4 qui, de plus, ont des flèches entrantes.

Exemple 2 :

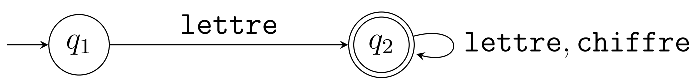


Figure 2 – Automate M_2

M_2 est standard.

Algorithme de standardisation

Théorème : Tout automate admet un automate équivalent standard.

La preuve de ce théorème repose sur un algorithme qui permet de construire effectivement un automate standard équivalent.

Soit $M = (\Sigma, Q, I, F, \delta)$ un automate quelconque, l'automate standardisé de M est $\mathrm{Std}(M) = (\Sigma, Q', I', F', \delta')$ où :

— $Q' = Q \cup \{i'\}$ où $i' \notin Q$

— $I' = i'$

— $F' = \begin{cases} F & \text{si } I \cap F = \emptyset \\ F \cup \{i'\} & \text{si } I \cap F \neq \emptyset \end{cases}$

— $\delta' = \delta \cup \{(i', a, q) \mid \exists i \in I \text{ tq } (i, a, q) \in \delta$

2 Automate émondé

Soit $M = (\Sigma, Q, I, F, \delta)$ un automate. Un état p de M est dit *accessible* s'il existe un chemin allant d'un état initial à p dans M. Un état p de M est dit *co-accessible* s'il existe un chemin allant de p à un état terminal dans M.

Un automate $M = (\Sigma, Q, I, F, \delta)$ est dit *émondé* si tous ses états sont accessibles et co-accessibles.

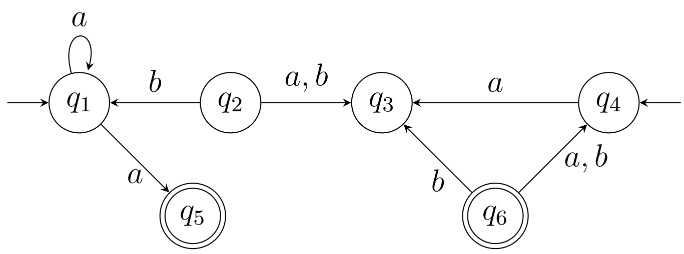


FIGURE 3 – Automate M_3

L'automate M_3 n'est pas émondé. Par exemple, l'état q_6 n'est pas accessible et l'état q_3 n'est pas co-accessible.

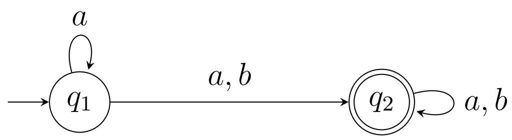


FIGURE 4 – Automate M_4

L'automate M_4 est émondé.

Algorithme d'émondage

Théorème : Tout automate admet un automate équivalent émondé.

La preuve de ce théorème repose sur un algorithme qui permet de construire effectivement un automate émondé équivalent. Cet algorithme construit pas à pas l'ensemble des états accessibles en partant de l'ensemble des états initiaux. On procède de manière symétrique avec les états co-accessibles en partant des états terminaux accessibles et en remontant les chemins ne comportant que des états accessibles. Une fois cet ensemble construit, il suffit de supprimer tous les états non accessibles et les flèches qui s'y rattachent.

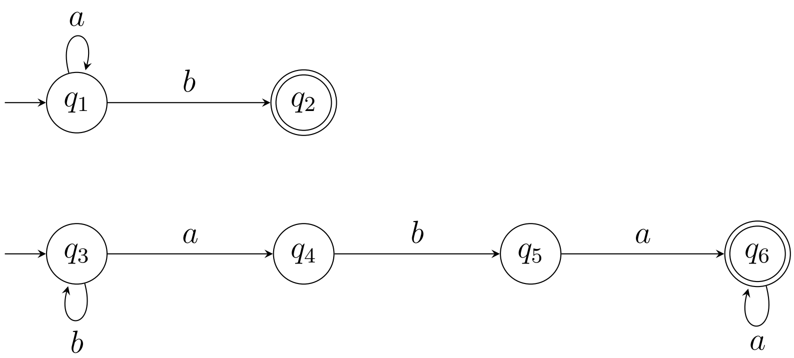
3 Propriétés de fermeture de $\mathrm{Rec}(\Sigma^*)$

Union

Théorème : L'ensemble $\mathrm{Rec}(\Sigma^*)$ est fermé par union finie.

Soient $M = (\Sigma, Q, I, F, \delta)$ et $M' = (\Sigma, Q', I', F', \delta)$ deux automates tels que $Q \cap Q' = \emptyset$ l'automate $M" = (\Sigma, Q \cup Q', I \cup I', F \cup F', \delta \cup \delta')$ reconnait le langage $L(M) \cup L(M')$

Exemple :



Cet automate reconnait le langage $\{a\}^* \cdot \{b\} \cup \{b\}^* \cdot \{aba\} \cdot \{a\}^*$

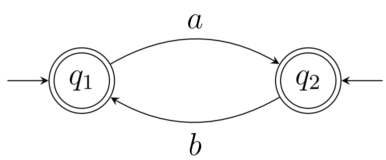
Concaténation

Théorème : L'ensemble $\mathrm{Rec}(\Sigma^*)$ est fermé par concaténation finie.

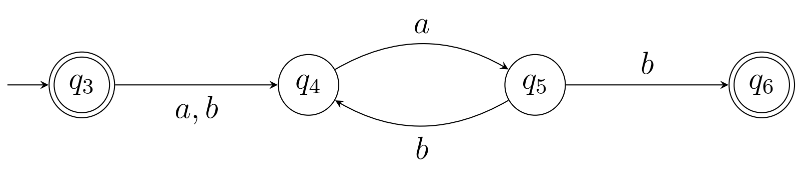
Soient $M = (\Sigma, Q, I, F, \delta)$ et $M' = (\Sigma, Q', \{i'\}, F', \delta)$ deux automates tels que M' est un automate standard. l'automate $M" = (\Sigma, Q \cup Q', I, F'', \delta \cup \delta' \cup \{(p, a, q) \mid p \in F \text{ et } (i', a, q) \in \delta'\})$ avec $F'' = \begin{cases} F \cup F' & \text{si } i' \in F' \\ F' & \text{sinon} \end{cases}$ reconnait le langage $L(M) \cdot L(M')$

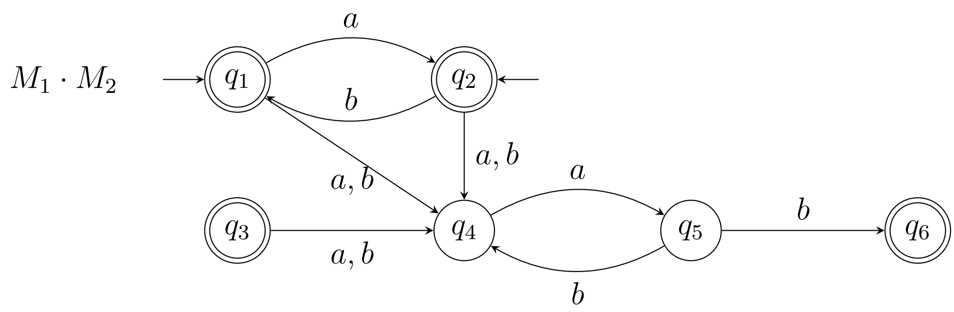
Exemple :

M_1



M_2



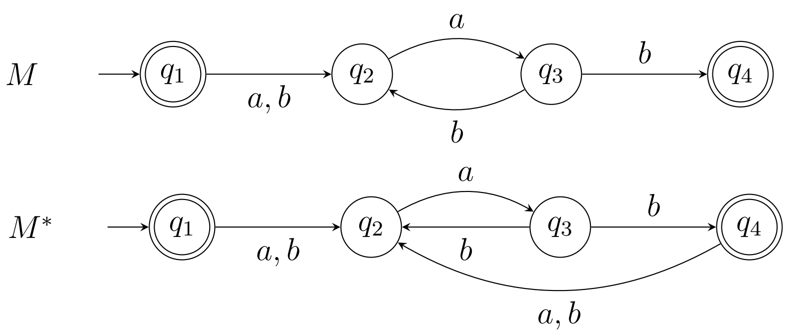


Etoile

Théorème : L'ensemble $\mathrm{Rec}(\Sigma^*)$ est fermé par passage à l'étoile.

Soient $M = (\Sigma, Q, \{i\}, F, \delta)$ un automate standard. L'automate $M' = (\Sigma, Q, \{i\}, F \cup \{i\}, \delta \cup \{(p, a, q) \mid p \in F \text{ et } (i, a, q) \in \delta\})$ reconnaît le langage $L(M)^*$.

Exemple :



Intersection

Théorème : L'ensemble $\mathrm{Rec}(\Sigma^*)$ est fermé par intersection finie.

Soient $M = (\Sigma, Q, \{i\}, F, \delta)$ et $M' = (\Sigma, Q', \{i'\}, F', \delta)$ deux automates standards. L'automate $M'' = (\Sigma, Q \times Q', \{i, i'\}, F \times F', \delta \times \delta')$ reconnait le langage $L(M) \cap L(M')$.