Théorie des langages

Alexis Nasr

Automates finis

Un automate fini est un 5-uplet $\langle Q, \Sigma, \delta, q_0, F \rangle$

- Q est l'ensemble des états,
- Σ est l'alphabet de l'entrée
- δ est la fonction de transition :

$$\delta \ : \ Q \times \Sigma \rightarrow Q$$

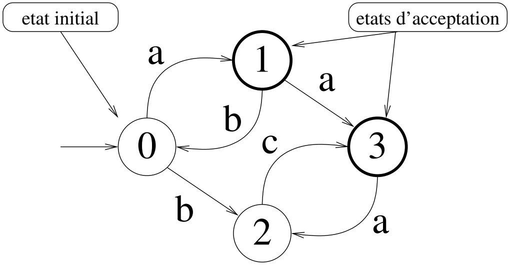
- $q_0 \in Q$ est l'état initial,
- $F \subseteq Q$ est l'ensemble des états d'acceptation.

Exemple

$\langle Q, \Sigma, \delta, q_0, F \rangle$
$Q = \{0, 1, 2, 3\}$
$\Sigma = \{a, b, c\}$
$\delta(0, a) = \{1\}$
$\delta(0, b) = \{2\}$
$\delta(1, a) = \{3\}$
$\delta(1, b) = \{0\}$
$\delta(2, c) = \{3\}$
$\delta(3, a) = \{2\}$
$q_0 = 0$
$F = \{1, 3\}$

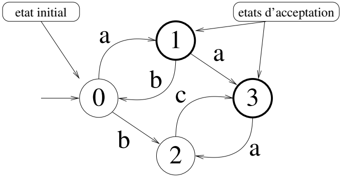
Représentation graphique

$\langle Q, \Sigma, \delta, q_0, F \rangle$
$Q = \{0, 1, 2, 3\}$
$\Sigma = \{a, b, c\}$
$\delta(0, a) = \{1\}$
$\delta(0, b) = \{2\}$
$\delta(1, a) = \{3\}$
$\delta(1, b) = \{0\}$
$\delta(2, c) = \{3\}$
$\delta(3, a) = \{2\}$
$q_0 = 0$
$F = \{1, 3\}$



Représentation matricielle

$\langle Q, \Sigma, \delta, q_0, F \rangle$
$Q = \{0, 1, 2, 3\}$
$\Sigma = \{a, b, c\}$
$\delta(0, a) = \{1\}$
$\delta(0, b) = \{2\}$
$\delta(1, a) = \{3\}$
$\delta(1, b) = \{0\}$
$\delta(2, c) = \{3\}$
$\delta(3, a) = \{2\}$
$q_0 = 0$
$F = \{1, 3\}$



		a	b	c
$\rightarrow$	0	1	2	
$\leftarrow$	1	3	0	
	2			3
$\leftarrow$	3	2		

Configurations et mouvement

$A = \langle Q, \Sigma, \delta, q_0, F \rangle$

- Configuration : $(q, m) \in Q \times \Sigma^*$ où :
  - q représente l'état courant de l'automate
  - m est la partie du mot à reconnaître non encore lue. Le premier symbole de m (le plus à gauche) est celui qui se trouve sous la tête de lecture. Si $m = \varepsilon$ alors tout le mot a été lu.
- Configuration initiale : (q_0, m) où m est le mot à reconnaître
- Configuration d'acceptation : (q, ε) avec $q \in F$
- Mouvement : $(q, aw) \vdash (q', w)$ si $q' = \delta(q, a)$.

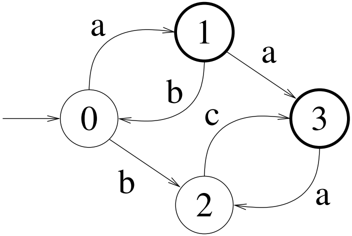
Langages reconnaissables

- Un mot m est reconnu par l'automate s'il existe une suite de mouvements menant de la configuration initiale ((q_0, m)) à une configuration d'acceptation ((q, ε) avec $q \in F$).
- Le langage reconnu par un automate A, noté $L(A)$, est l'ensemble des mots reconnus par ce dernier :

$$L(A) = \{m \in \Sigma^* | (q_0, m) \overset{*}{\vdash} (q, \varepsilon) \text{ avec } q \in F\}$$

- Un langage L sur Σ est reconnaissable s'il existe au moins un automate fini A ayant Σ comme alphabet d'entrée tel que $L = L(A)$.

Reconnaissance

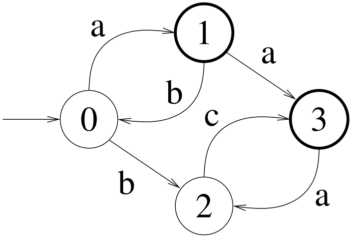


$$
\begin{array}{ll}
 & (0, ababaa) \\
\vdash & (1, babaa) \\
\vdash & (0, abaa) \\
\vdash & (1, baa) \\
\vdash & (0, aa) \\
\vdash & (1, a) \\
\vdash & (3, \varepsilon)
\end{array}
$$

Tout le mot est lu, l'automate est dans un état d'acceptation :

$$ababaa \in L(A)$$

Reconnaissance

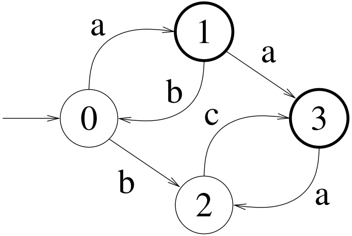


$$\begin{array}{ll} & (0, abab) \\ \vdash & (1, bab) \\ \vdash & (0, ab) \\ \vdash & (1, b) \\ \vdash & (0, \varepsilon) \end{array}$$

Tout le mot est lu mais l'automate n'est pas dans un état d'acceptation :

$$abab \notin L(A)$$

Reconnaissance



$$\begin{array}{ll} & (0, aac) \\ \vdash & (1, ac) \\ \vdash & (3, c) \end{array}$$

L'automate est coincé, il n'y a pas de transition sur c depuis l'état 3

$$aac \notin L(A)$$

Automate complet

- Un automate $A = \langle Q, \Sigma, \delta, q_0, F \rangle$ est complet si A peut transiter depuis chaque état vers un autre état sur tous les symboles de Σ.

$$\forall q \in Q, \forall a \in \Sigma, \exists e \in Q, \delta(q, a) = e$$

- Si un automate n'est pas complet, on peut le compléter en lui ajoutant un nouvel état, qui n'est pas un état d'acceptation, appelé état puits dans lequel aboutiront toutes les transitions qui "manquaient".

Exemple

Etats accessibles

- Un état q de A est dit accessible s'il est possible d'y accéder depuis l'état initial ou, plus précisément, s'il existe un mot $m \in \Sigma^*$ permettant d'effectuer une suite de mouvements menant de l'état initial à q :

$$q \text{ accessible } \Leftrightarrow \exists m \in \Sigma^* \ (q_0, m) \overset{*}{\vdash} (q, \varepsilon)$$

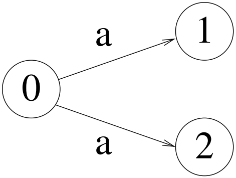
- il est dit co-accessible s'il est possible d'accéder à un état d'acceptation depuis cet état, ou encore, s'il existe un mot $m \in \Sigma^*$ permettant d'effectuer une suite de mouvements menant de q à un état d'acceptation :

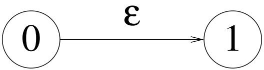
$$q \text{ co-accessible } \Leftrightarrow \exists m \in \Sigma^* \ (q, m) \overset{*}{\vdash} (e, \varepsilon) \text{ avec } e \in F$$

Exemple

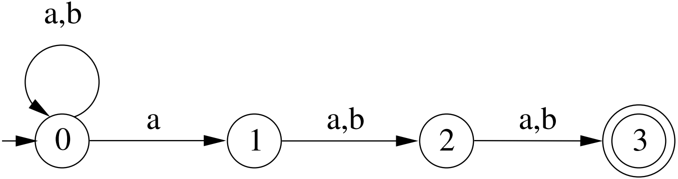
Non déterminisme

- Un automate A est déterministe si pour toute configuration de A, il existe au plus un mouvement possible.
- Un automate est non déterministe s'il existe des configurations pour lesquelles plus d'un mouvement est possible.
- Deux origines :

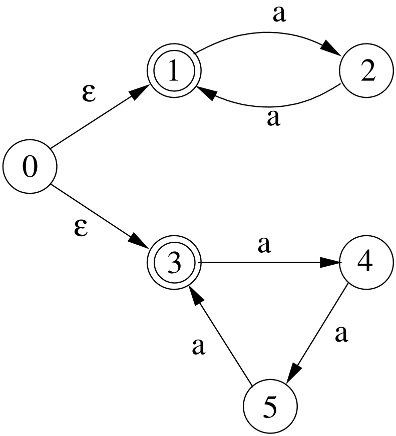




Exemple 1



Exemple 2



Automates finis non déterministes

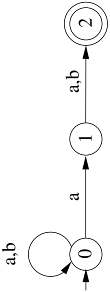
Un automate fini non déterministe est un 5-uplet $\langle Q, \Sigma, \delta, q_0, F \rangle$

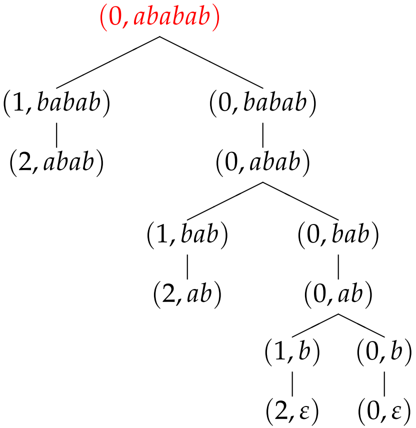
- Q est l'ensemble des états,
- Σ est l'alphabet de l'entrée
- δ est la fonction de transition :

$$\delta \ : \ Q \times \{\Sigma \cup \varepsilon\} \rightarrow \mathcal{P}(Q)$$

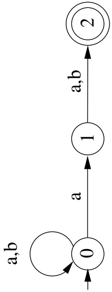
- $q_0 \in Q$ est l'état initial,
- $F \subseteq Q$ est l'ensemble des états d'acceptation.

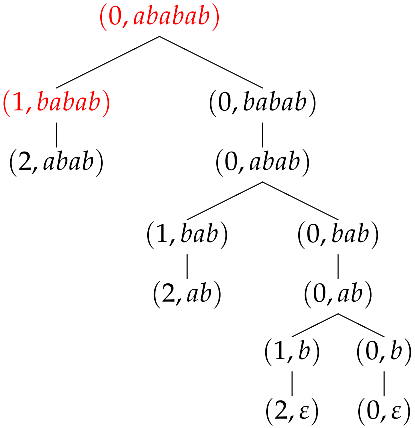
Reconnaissance d'un mot par un automate non déterministe



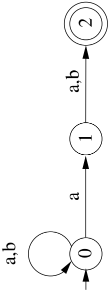


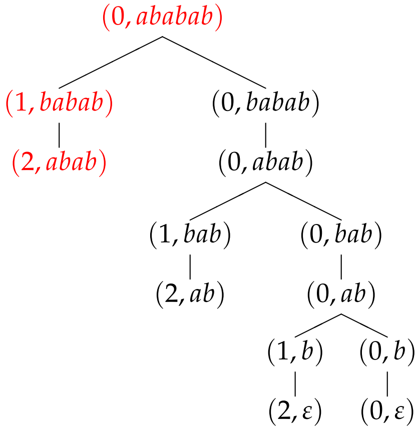
Reconnaissance d'un mot par un automate non déterministe



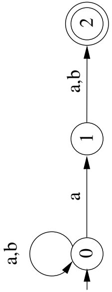


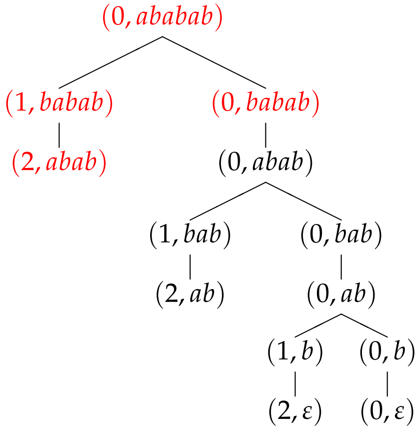
Reconnaissance d'un mot par un automate non déterministe



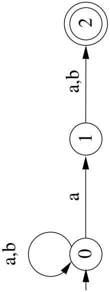


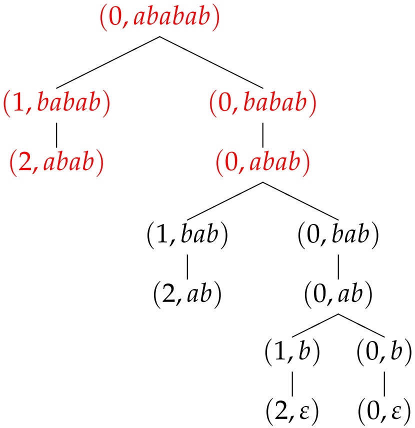
Reconnaissance d'un mot par un automate non déterministe



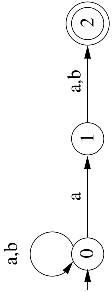


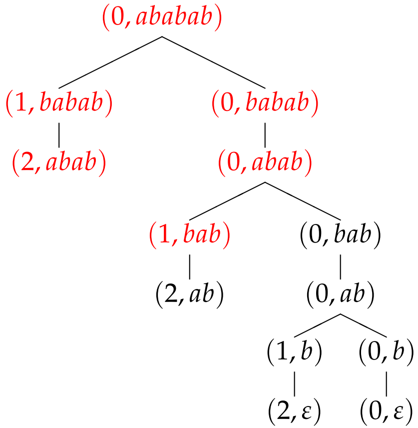
Reconnaissance d’un mot par un automate non déterministe



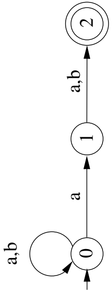


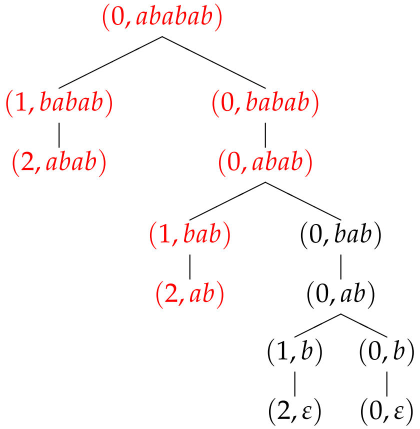
Reconnaissance d'un mot par un automate non déterministe



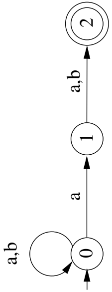


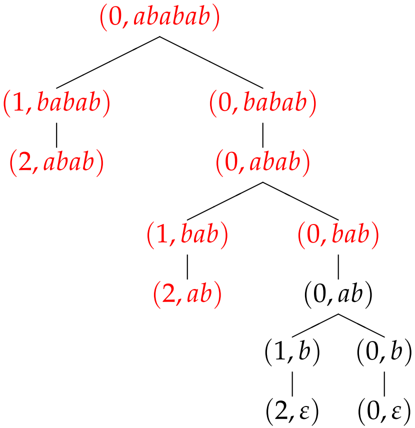
Reconnaissance d'un mot par un automate non déterministe



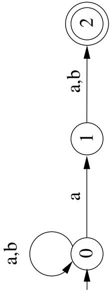


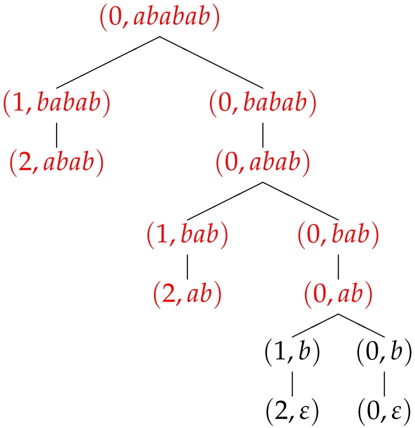
Reconnaissance d’un mot par un automate non déterministe



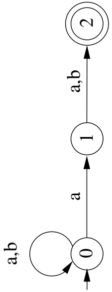


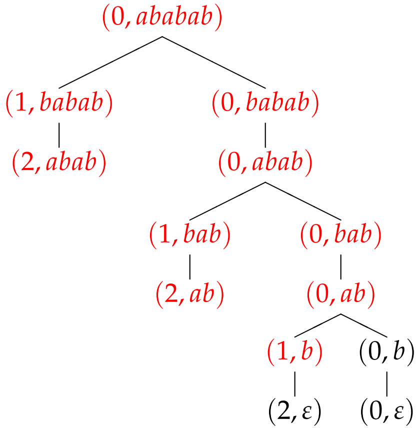
Reconnaissance d'un mot par un automate non déterministe



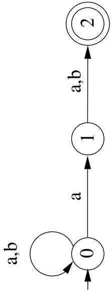


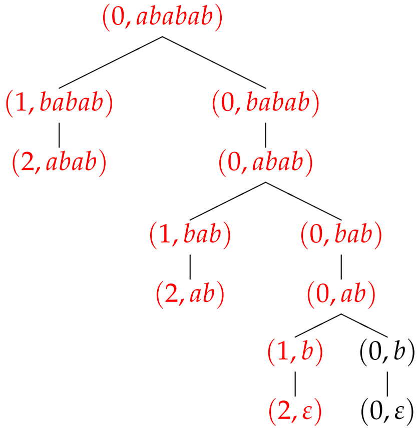
Reconnaissance d'un mot par un automate non déterministe



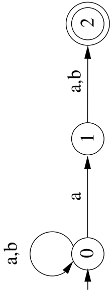


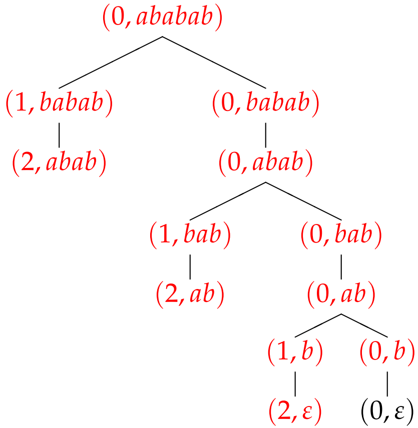
Reconnaissance d'un mot par un automate non déterministe



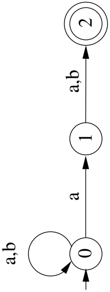


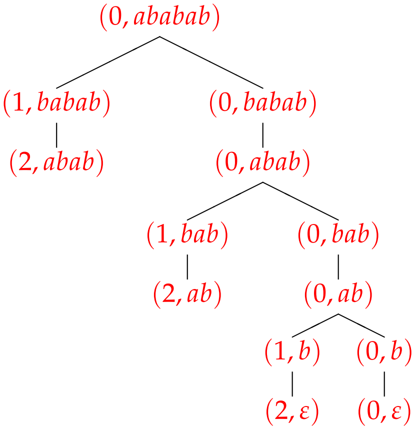
Reconnaissance d'un mot par un automate non déterministe





Reconnaissance d'un mot par un automate non déterministe





Equivalence

- Les automates finis déterministes et non déterministes reconnaissent la même classe de langages.
- Ce résultat est à la fois surprenant et pratique.
  - Il est surprenant car les automates non déterministes semblent plus puissants que les automates déterministes.
  - Il est pratique car il est souvent plus simple de décrire un langage à l'aide d'un automate non déterministe.
- Preuve par construction : méthode permettant de transformer un automate non déterministe en un automate déterministe reconnaissant le même langage.