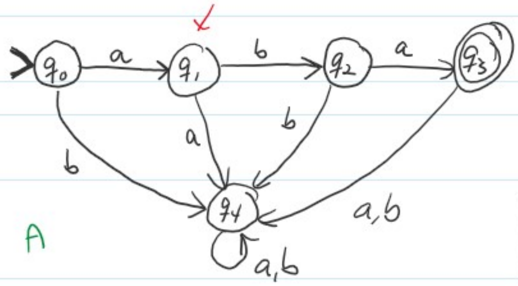
Automates à états finis

2 avril 2024 10:30

Un automate à états finis est une structure discrète qui sert à détecter des mots (chaîne de caractères) avec une syntaxe qu'on appelle <u>régulière</u>.

$\delta: q_1, b \to q_2$

ex:



○ = état

>○ : état initial

→ : transition

◎ : état acceptant

δ

(q_0, a, q_1)
(q_0, b, q_4)
(q_1, a, q_4)
⋮

$L(A) = \{aba\}$

ex: aba $\quad q_0 \to q_1 \to q_2 \to q_3$ fin dans un état acceptant

abab $\quad q_0 \to q_1 \to q_2 \to q_3 \to q_4$ fin dans un état non-acceptant

bab $\quad q_0 \to q_4 \to q_4 \to q_4$ non-acceptant

- Sert à faire de la recherche dans des textes
  *mat*115*
- Sert à valider des formats
  ex: courriel: $[a-z]^* @ [a-z]^* . [a-z]^*$

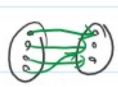
Automate fini déterministe (AFD)

Une AFD est un quintuplet $\langle Q, \Sigma, \delta, q_0, F \rangle$ où

1) Q est un ensemble fini d'états $\quad$ ex: $Q = \{q_0, q_1, q_2, q_3, q_4\}$
2) Σ est l'alphabet supporté $\quad \Sigma = \{a, b\}$
3) δ : transitions (plus tard)

4) $q_0 \in Q$ est l'état initial
5) $F \subseteq Q$ est l'ensemble des états acceptants

Revenons sur δ:



δ est une fonction de $Q \times \Sigma$ vers Q

Revenons sur δ :

δ est une fonction de $Q \times \Sigma$ vers Q

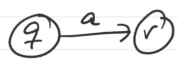
$q \, , \, a \rightarrow r$

état courant, symbole $\rightarrow$ état suivant

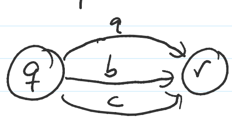
Les élts de δ ont la forme $((q,a),r)$

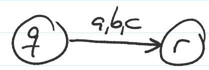
On écrit souvent (q,a,r)

ou



- On peut représenter plusieurs transitions d'un état q à r par une seule flèche

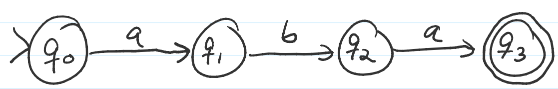




- Un état puits est un état duquel on ne peut pas sortir.
  ex: q_4

- Si une transition n'est pas spécifiée, alors on suppose qu'elle va dans un puits non-représenté.

ex:



équivalent à l'exemple plus haut

- Un mot est une séquence de symboles d'un alphabet Σ.
  Représentations possibles: $[\sigma_1, \sigma_2, \ldots, \sigma_n]$ $\forall i, \sigma_i \in \Sigma$
  $(\sigma_1, \sigma_2, \ldots, \sigma_n)$
  $\sigma_1 \sigma_2 \sigma_3 \ldots \sigma_n$

  ex: allo $[a, l, l, o]$
  (a, l, l, o)
  allo

- On dénote par Σ^* l'ensemble infini de tous les mots possibles sur Σ.

- Soit $A = \langle Q, \Sigma, \delta, q_0, F \rangle$ un AFD.

  Un mot $w \in \Sigma^*$ est accepté par A si, en suivant les transitions des caractères de w sur A, on termine dans un état de F.

  └ déf formelle: voir notes de cours

- On définit

  $L(A) = \{ w \mid w \in \Sigma^* \text{ et } A \text{ accepte } w \}$

• On définit

$$L(A) = \{ w \mid w \in \Sigma^* \text{ et } A \text{ accepte } w \}$$

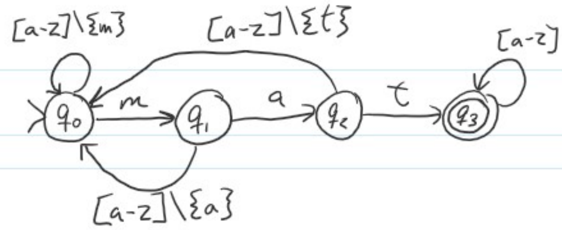
qu'on appelle le langage de A.

ex: mots qui contiennent "mat" comme sous-chaîne $\Sigma = [a-z]$

ex: allomatxy

Notation: $[a-z] = \{a, b, c, d, \ldots, x, y, z\}$

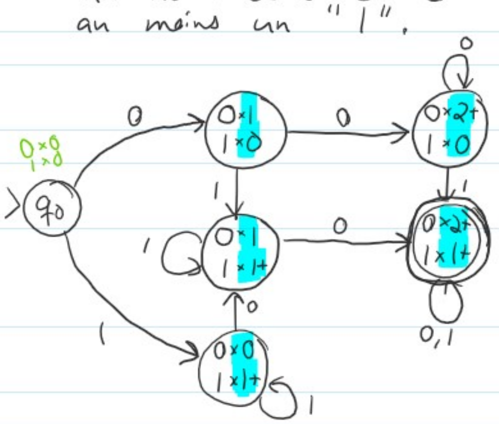
$[0-9] = \{0, 1, 2, \ldots, 9\}$



xmxmatx

maxmat

ex: mots sur $\Sigma = \{0, 1\}$ qui contiennent au moins deux "0" et au moins un "1".



0×2+
1×0
veut dire
j'ai vu 2 ou plus "0" et
0 fois "1"

ex: 0100

ex: 111011

Automates fini non-déterministe (AFND)

Un AFND est un quintuplet $A = \langle Q, \Sigma, \delta, q_0, F \rangle$

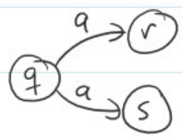
où $\underset{\text{états}}{Q}, \underset{\text{alpha}}{\Sigma}, \underset{\text{init}}{q_0}, \underset{\text{accept.}}{F}$ = comme avec les AFD

mais

$$\delta \subseteq Q \times \Sigma \times Q$$

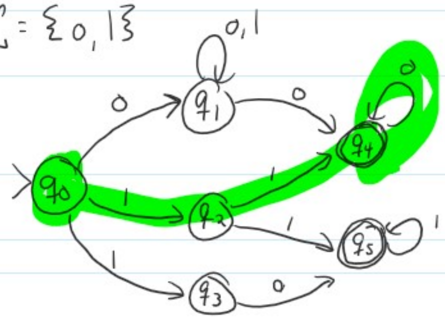
Ceci rend possible, par ex, $\begin{cases} (q, a, r) \\ (q, a, s) \end{cases}$ $r \neq s$

$(q, a, s) \quad r \neq s$



- Un mot $w \in \Sigma^*$ est accepté par A s'il existe un choix de séquence de transitions possible qui termine dans un état de F.

ex: $\Sigma = \{0, 1\}$



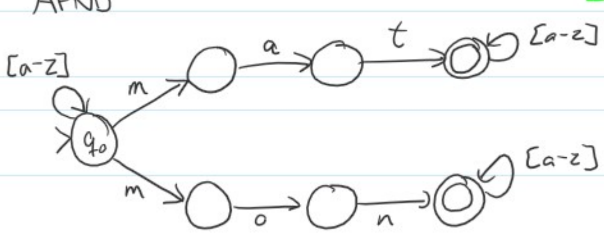
ex: 1100
Le chemin
$q_0 \rightarrow q_2 \rightarrow q_4 \rightarrow q_4$
est possible
$\Rightarrow$ 1100 accepté

ex: 100101 $\quad q_0 \rightarrow q_3 \rightarrow q_5 \rightarrow$ puits $\quad$ ce chemin échoue

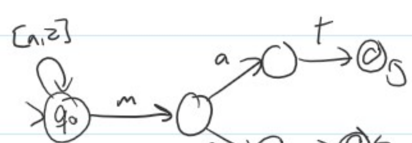
$q_0 \rightarrow q_2 \rightarrow$ puits $\quad$ échoue $\quad$ pas d'autre chemin

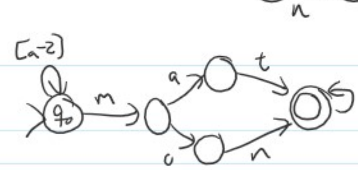
$\Rightarrow$ rejeté

- Exprimer $M \subseteq \Sigma^*$ contenant "mat" ou "mon"
AFND



ex: xymoxmatx $\qquad$ xmmonx





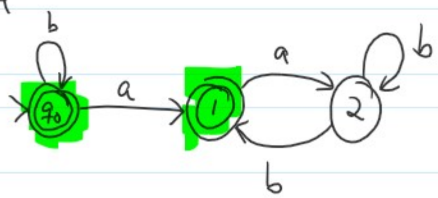
Déterminisation d'un AFND

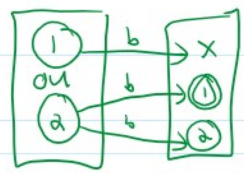
Déterminisation d'un AFND

AFND $A \longrightarrow$ AFD A' équivalent

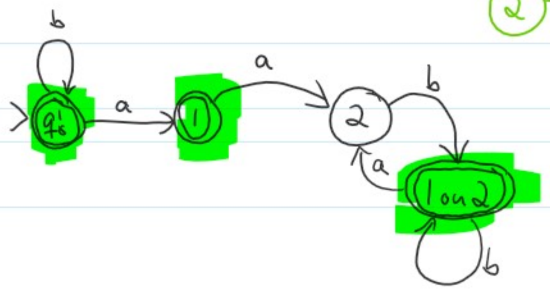
$L(A) = L(A')$ // $L(A)$ = langage de A

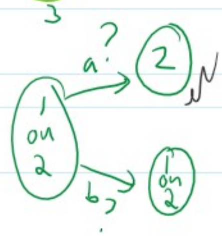
AFND A

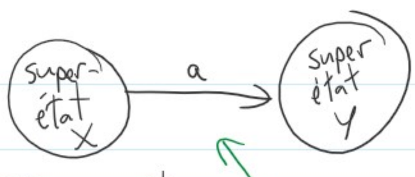




AFD A'







AFND pourrait choisir d'aller dans n'importe quel état de X

de X, l'AFND pourrait choisir d'aller n'importe où dans Y sur a

Pour déterminiser AFND $A = \langle Q, \Sigma, \delta, q_0, F \rangle$
en AFD
$A' = \langle Q', \Sigma, \delta', q_0', F' \rangle$

1) États de A' : $Q' = \mathbb{P}(Q) = \{ R \mid R \subseteq Q \}$
2) Départ de A' : $q_0' = \{ q_0 \}$
3) États finaux de A' : $F' = \{ R \mid R \subseteq Q \land R \cap F \neq \emptyset \}$
4) Transitions de A' : δ'

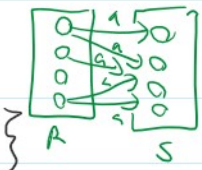
... contient tous

4) Transitions de A' : δ'

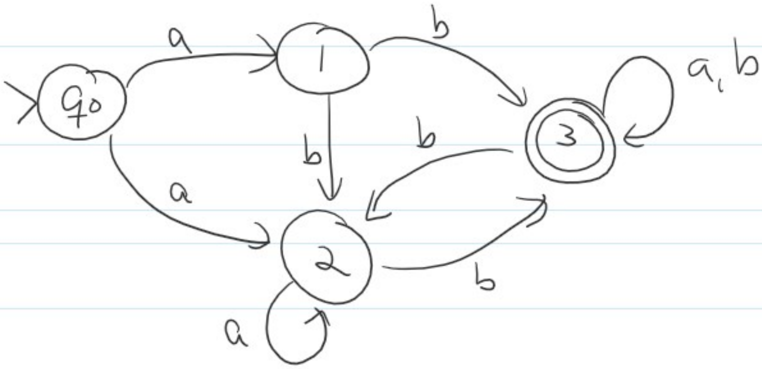
S contient tous les états accessibles à partir de R sur a

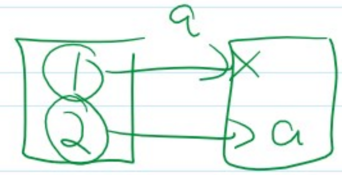
On ajoute à δ' (R, a, S) si

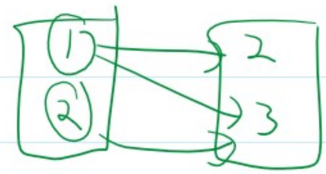
$$S = \{q \mid \exists r \cdot r \in R \wedge (r,a,q) \in \delta\}$$

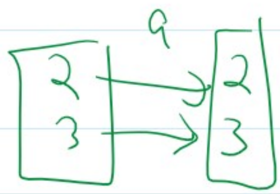


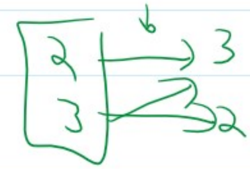
ex:



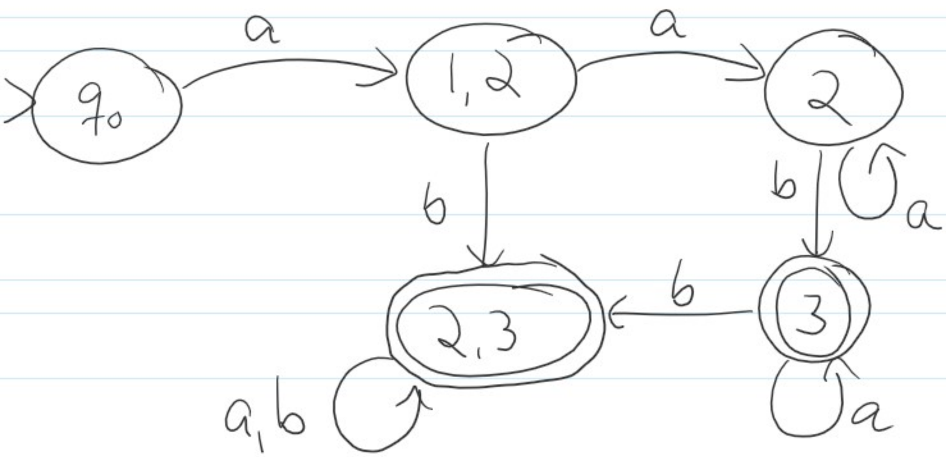








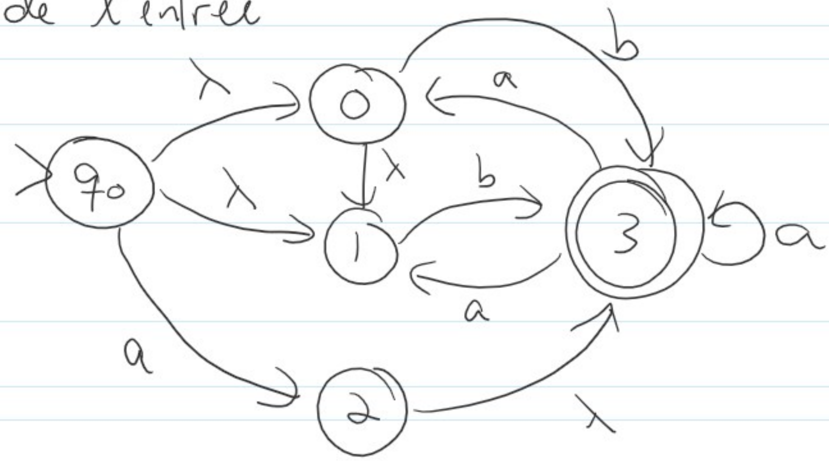
AFD:



λ - transitions   λ = lambda

λ = transition vide (sur AFND)

On peut aller de p à q sans consommer de caractère du mot de l'entrée



Mot

a

accepté via

$q_0 \xrightarrow{a} 2 \xrightarrow{\lambda} 3$

bbb × non

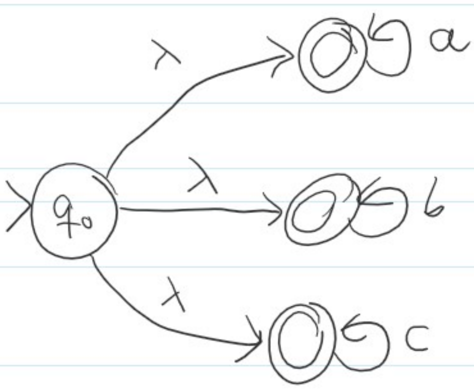
bba × non

aaa ✓ oui

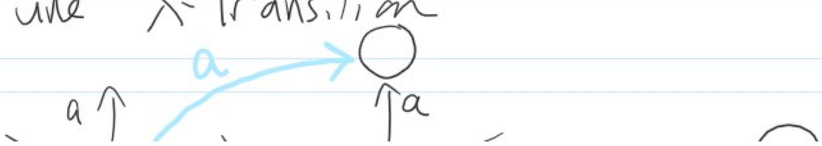
baaabaa :

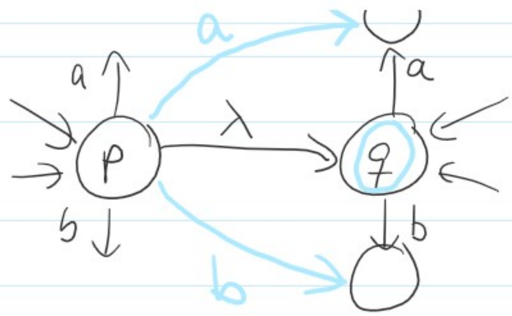
Exemple : mots qui ne contiennent que des a
ou que des b
ou que des c

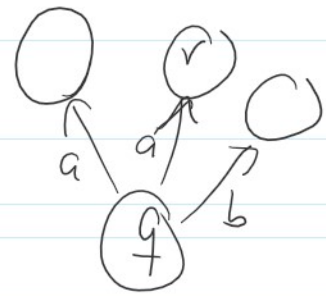
aaa aaaa
bbbb
cc



Retirer une λ-transition

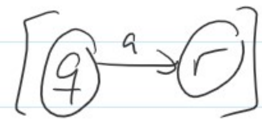






Retirer (p, λ, q) :

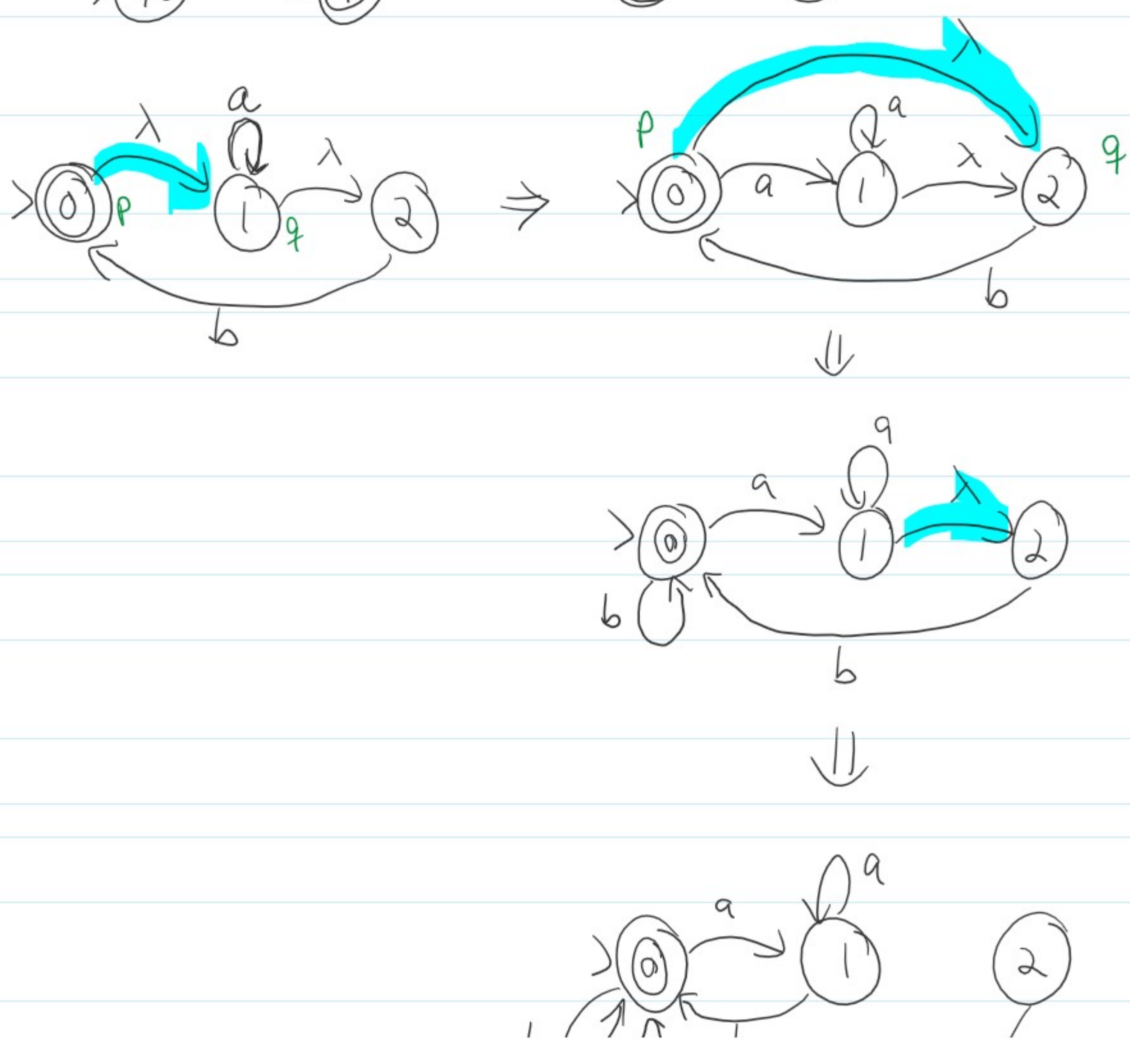
Pour chaque $(q, a, r) \in \delta$



ajouter (p, a, r) à δ

si $q \in F$, alors ajouter p à F

ex:




0
1
2
b
b
b