Automates finis

Automate fini

- Un AF est un quintuplet

$$A = (\Sigma, Q, \delta, q_0, F)$$

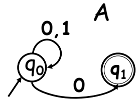
1. Σ : un alphabet fini (symboles du ruban)
2. Q : un ensemble fini d'états
3. δ : fonction de transition (règle de changement d'état)

$$\delta \subseteq Q \times \Sigma \cup \{\varepsilon\} \times Q$$

4. $q_0 \in Q$: état initial
5. $F \subseteq Q$: états de reconnaissance (ou finals)

Exemple d'automate fini

A



A = $(\Sigma, Q, \delta, q_0, F)$

$\Sigma = \{0,1\}$
$Q = \{q_0, q_1\}$
$\delta = \{(q_0, 0, q_0), (q_0, 0, q_1), (q_0, 1, q_0)\}$
q_0 est l'état initial
F = $\{q_1\}$
A accepte $m \in \Sigma^*$ s'il existe un chemin de q_0 à un état de F étiqueté par les lettres de m.
A reconnaît le langage L(A) décrit par l'expression rationnelle : $(0+1)^*0$

Automates finis déterministes

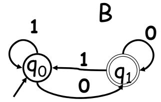
- Un automate fini est **déterministe** (D.F.A.) si et seulement si δ est une fonction de transition telle que :

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

- D'un état donné, il part **au plus** une transition étiquetée par une lettre donnée.

Exemple d'automate fini déterministe

B



B = $(\Sigma, Q, \delta, q_0, F)$

$\Sigma = \{0,1\}$
$Q = \{q_0, q_1\}$
$\delta = \{(q_0, 1, q_0), (q_0, 0, q_1), (q_1, 1, q_0), (q_1, 0, q_1)\}$
q_0 est l'état initial
F = $\{q_1\}$
B accepte $m \in \Sigma^*$ s'il existe un chemin de q_0 à un état de F étiqueté par les lettres de m.
B reconnaît le langage L(B) décrit par l'expression rationnelle : $(0+1)^*0$

Autres représentations des automates

- déterministes

	0	1
$\rightarrow q_0$	q_1	q_0
$\leftarrow q_1$	q_1	q_0

- non-déterministes

	0	1
$\rightarrow q_0$	$\{q_0, q_1\}$	q_0
$\leftarrow q_1$	-	-

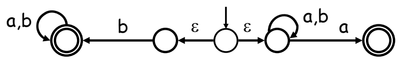
- δ : $(q_0,0) \rightarrow q_1$
  $(q_0,1) \rightarrow q_0$
  $(q_1,0) \rightarrow q_1$
  $(q_0,1) \rightarrow q_0$

Automate fini non déterministe

- Du même état, on peut avoir plusieurs transitions étiquetées par la même lettre
  - ce n'est plus une fonction mais
    - une relation $\delta \subseteq Q \times \Sigma \cup \{\varepsilon\} \times Q$
    - une fonction avec un autre co-domaine

      $\delta : Q \times \Sigma \cup \{\varepsilon\} \rightarrow P(Q)$
- On peut avoir des transitions vides appelées

  ε-transitions
- On peut avoir plusieurs états initiaux



7

différences

déterministe	non-déterministe
▪ fonction de transition	▪ relation de transition
▪ un seul état initial	▪ plusieurs états initiaux
▪ pas d'ε-transitions	▪ ε-transitions possibles
▪ unicité de la lecture	▪ pluralité de lecture
$\delta : Q \times \Sigma \rightarrow Q$	$\delta \subseteq Q \times \Sigma \cup \{\varepsilon\} \times Q$ $\delta : Q \times \Sigma \cup \{\varepsilon\} \rightarrow P(Q)$

8

Lecture/calcul de l'automate

- **Réussie** ou ratée
- lecture = suite des états pris par l'AFD
  - commence par l'état initial
  - termine
    - par un état de reconnaissance $\Rightarrow$ **réussie**
    - par un autre état $\Rightarrow$ **ratée**
  - ne termine pas (pas de transition applicable) $\Rightarrow$ **ratée**

9

Formalisation: configuration

- Représente
  - l'état courant
  - la partie du mot qui reste à lire

    (état courant, suffixe à lire)$\in Q \times \Sigma^*$
- application d'une transition modifie la configuration: dérivation

  $(q,a.w) \rightarrow^a (q',w)$ si $(q,a,q') \subseteq \delta$

  $(q,a.w) \rightarrow^\varepsilon (q',a.w)$ si $(q,\varepsilon,q') \subseteq \delta$

10

lecture = suite des configurations

- On étend la notion de configuration aux mots:

  $(q,w) \rightarrow^* (q',\varepsilon)$
- *w* reconnu si *A* termine la lecture dans un état final
- *w* pas reconnu si
  - *1) w* a été entièrement lu et $q' \notin F$
  - *2) w* n'a pas été entièrement lu et plus de dérivation possible

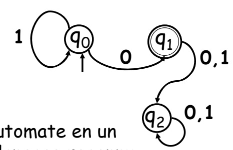
11

Automates finis complets

Un automate fini (déterministe) est ***complet*** ssi δ est une <u>fonction totale</u> sur $Q \times \Sigma$.

De chaque état, part <u>exactement</u> une flèche étiquetée par chacune des lettres de Σ.

état	0	1
q_0	q_1	q_0
q_1	q_2	q_2
q_2	q_2	q_2



On peut toujours transformer un automate en un automate complet sans modifier le langage reconnu

12

Exemple

1	0	0	1	0

↑

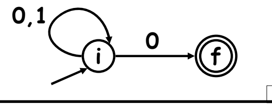
un mot à reconnaître

(i,10010) une configuration initiale

sa table de transition

	0	1
→i	{i,f}	i
←f	-	-

un AFND



Equivalence AFD et AFND

Théorème : Tout langage reconnu par un automate fini l'est par un automate fini déterministe.

Preuve : si l'automate de départ est déterministe, évident;

Si l'automate de départ est non-déterministe :

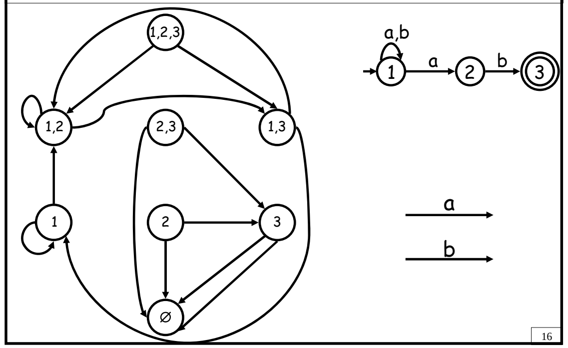
- construire un AFD qui intègre tous les choix du non déterministe (*algorithme de déterminisation*)
- prouver que l 'AFD reconnaît le même langage

Principe

- Simulation fonctionnement AFND : mémoriser dans quel ensemble d'états on est et, en lisant une lettre, voir dans quel ensemble d'états on arrive.
- *C'est exactement ce que va faire un AFD qui simule un AFND*

⇒ état AFD=ens états AFND

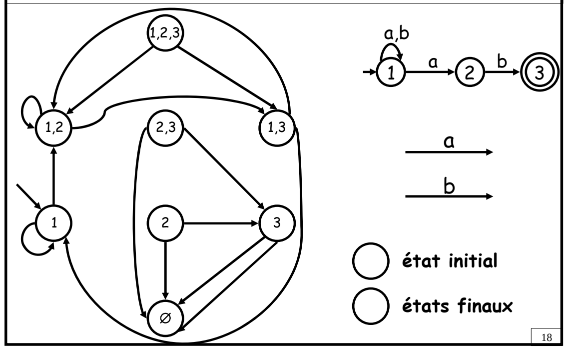
Illustration



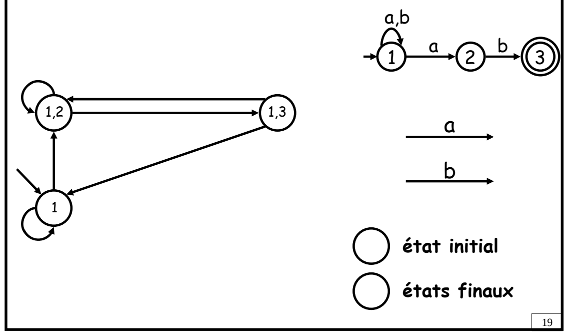
Tout ne sert pas...

- états utiles : accessibles
  - *q* accessible s'il existe une suite de dérivations de l'initial vers *q*
- états particuliers : initial et finaux
  - initial : le même que celui (ceux) de l'AFND
  - finals : toutes les parties contenant un état de reconnaissance de l'AFND

Exemple



Etats utiles



Structure de la preuve

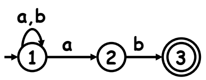
- AFND $N=(\Sigma, Q, q_0, \delta, F) \rightarrow$ AFD $A=(\Sigma, Q' \subseteq \mathcal{P}(Q), q_0', \delta', F')$
- $q_0'=E(q_0)$
- $\delta'(R,a) = \{q \in Q \mid q \in E(\delta(r,a))$ pour $r \in R\}$
  $\delta'(R,a)= \cup_{r \in R} E(\delta(r,a))$
- $F'=\{R \mid R \in Q'$ et $\{f\} \in R, \{f\} \in F\}$
- où $E(R):=\{q \mid q$ accessible de R en suivant 0 ou plusieurs ε-transitions$\}$ (ε-clôture)

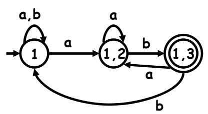
Cas sans ε-transition

- $q_0'=\{q_0\}$
- Pour $R \in Q'$ et $a \in \Sigma$,
- $\delta'(R,a) = \{q \in Q \mid q \in \delta(r,a)$ pour $r \in R\}$
  $\delta'(R,a)= \cup_{r \in R} \delta(r,a)$
- $F'= \{R \in Q' \mid \{f\} \in R, f \in R\}$

L(N) = (a+b)*ab

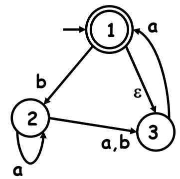
	a	b
→1	{1,2}	1
2	-	3
←3	-	-
→{1}	{1,2}	{1}
{1,2}	{1,2}	{1,3}
←{1,3}	{1,2}	{1}





Exemple avec ε-transition

	a	b	ε
↔1	-	2	3
2	{2,3}	3	-
3	1	-	-



Exemple avec ε-transition

	a	b	ε
↔1	-	2	3
2	{2,3}	3	-
3	1	-	-
↔{1,3}	{1,3}	{2}	
{2}	{2,3}	{3}	
{2,3}	{1,2,3}	{2}	
{3}	{1,3}	-	
←{1,2,3}	{1,2,3}	{2,3}