Une précision

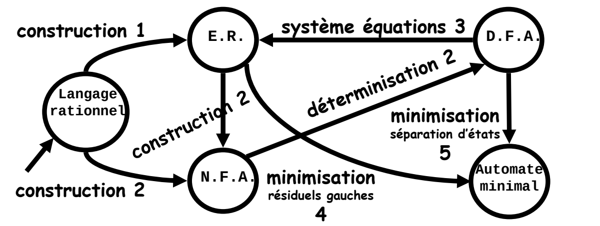
- Constat : on mélange équations de droite et de gauche (parfois nous enseignants … aussi)

Pour l'éviter :

- Dans le terme : Z=ZA+B
  - récursivité à **gauche**
  - linéarité à **droite**
- Dans le terme : Y=AY+B
  - récursivité à **droite**
  - linéarité à **gauche**

Automate minimal

Un méta-automate...



Théorème : Tout langage rationnel est reconnu par un unique automate déterministe minimal*.

* la minimalité porte sur <u>le nombre d'états</u> de l'automate

Automate réduit & minimal

- un automate déterministe A est réduit si pour tout couple d'états distincts p et q de A, p et q ne sont pas équivalents
- un automate (déterministe) réduit A est minimal s'il n'existe pas d'automate reconnaissant le même langage avec moins d'états.

Langage associé à un état

- soit un AFD $A = (\Sigma, Q, \delta, q_0, F)$, on appelle langage associé à q de Q et on note $L_q(A)$ le langage :

$$L_q(A) = \{w \in \Sigma^*, \delta^*(q,w) \in F\}$$

- $L_q(A)$ est le langage reconnu par un automate dont l'état initial serait q et qui aurait F comme ensemble d'états finals.

$$L(A) = L_{q_0}(A)$$

Problème 1

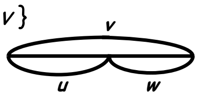
- Donnée : une expression rationnelle E
- Problème : construire un AFD minimal qui reconnaisse le langage décrit par E
- Idée : *les quotients gauches ...*

Quotients gauches

➢obtenir les suffixes d'un mot

- Pour deux **mots** $u,v \in \Sigma^*$,

$$u^{-1}v = \{w \in \Sigma^* \mid u.w = v\}$$



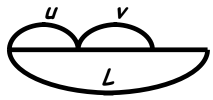
- Pour deux **langages** $X,Y \subseteq \Sigma^*$,

$$X^{-1}Y = \cup_{x\in X} \cup_{y \in Y} x^{-1}y$$

- On va utiliser le quotient d'un langage L par un mot u :

$$u^{-1}L = \{w \in \Sigma^* \mid u.w \in L\}$$

Propriétés



- $\varepsilon^{-1}L = L, \forall L \subseteq \Sigma^*$
- $a^{-1}\varnothing = a^{-1}\varepsilon = \varnothing$
- $a^{-1}a = \varepsilon$
- $a^{-1}b = \varnothing$ pour $a \neq b$

- $(u.v)^{-1} L = v^{-1}(u^{-1}L)$, $\forall L \subseteq \Sigma^*$
- $a^{-1}(X+Y) = a^{-1}X + a^{-1}Y$
- $a^{-1}X^* = (a^{-1}X). X^*$
- $a^{-1}(X.Y) = (a^{-1}X) .Y+(X\cap\{\varepsilon\})\, a^{-1}Y$

Fondement de la minimisation

Théorème : si L est un langage rationnel, alors l'ensemble de ses quotients gauches

$$Q(L) = \{u^{-1}L \mid u \in \Sigma^*\} \text{ est fini.}$$

Proposition : soit $A = (\Sigma, Q, \delta, q_0, F)$ un AFD complet et dont tous les états sont accessibles, on a :

$$Q(L) = \{ L_q(A), q \in Q \}$$

Fondement de la minimisation

Le cardinal de l'ensemble des résiduels est borné par celui du nombre d'états.

- Pour $L \subseteq \Sigma^*$, on définit $A(L) = \big(Q(L), \Sigma, \delta, \{L\}, F(L)\big)$, l'**automate minimal** de L où
  - $F(L) = \{ u^{-1}L \mid \{\varepsilon\} \in u^{-1}L\}$
  - $\delta(Y,a) = a^{-1}Y$ pour $Y \in Q(L)$, $a \in \Sigma$

$Q(L) = \{L_q(A) \mid q \in Q\}$

- $Q(L) \subseteq \{L_q(A) \mid q \in Q\}$

  Soit $u \in \Sigma^*$ et $q = \delta(i,u)$.

  - q existe toujours (A complet et tous ses états sont accessibles).

$$u.w \in L \Leftrightarrow w \in u^{-1}L \Leftrightarrow w \in Q(L) \Leftrightarrow \delta(i,u.w) \in F$$
$$\Leftrightarrow \delta(q,w) \in F \Leftrightarrow w \in L_q(A)$$

Donc $u^{-1}L \subseteq L_q(A)$

$Q(L) = \{L_q(A) \mid q \in Q\}$

- $\{L_q(A) \mid q \in Q\} \subseteq Q(L)$

  Soit $q \in Q$ et $u \in \Sigma^*$ tels que $\delta(i,u) = q$.

  $u \in \Sigma^*$ existe toujours (A complet et tous ses états accessibles).

$$L_q(A) \subseteq u^{-1}L$$

Tout ce qui est reconnu en partant de q correspond aux suffixes de mots de L.

Exemple : $Q(\Sigma^*ab\Sigma^*)$

$a^{-1}(X+Y)= a^{-1}X+ a^{-1}Y$
$a^{-1}X^*=(a^{-1}X).\ X^*$
$a^{-1}(X.Y)=(a^{-1}X)\ .Y+(X\cap\varepsilon)\ a^{-1}Y$

- $a^{-1}L = a^{-1}(\Sigma^*ab\ \Sigma^*)$

$= (a^{-1}\Sigma^*)\ ab\ \Sigma^* + a^{-1}(ab\ \Sigma^*)$
$= (a^{-1}\Sigma)\ \Sigma^*\ ab\ \Sigma^* + (a^{-1}a)b\ \Sigma^* + \emptyset$

$a^{-1}L = \Sigma^*\ ab\ \Sigma^* + b\ \Sigma^* = L + b\ \Sigma^*$ (nouveau)

- $b^{-1}L = b^{-1}(\Sigma^*ab\ \Sigma^*)$

$= (b^{-1}\Sigma^*)\ ab\ \Sigma^* + b^{-1}(ab\ \Sigma^*)$
$= (b^{-1}\Sigma)\ \Sigma^*\ ab\ \Sigma^* + (b^{-1}a)b\ \Sigma^* + \emptyset$

$b^{-1}L = \Sigma^*\ ab\ \Sigma^* = L$ (pas nouveau)

13

Exemple : $Q(\Sigma^*ab\Sigma^*)$

$a^{-1}(X+Y)= a^{-1}X+ a^{-1}Y$
$a^{-1}X^*=(a^{-1}X).\ X^*$
$a^{-1}(X.Y)=(a^{-1}X)\ .Y+(X\cap\varepsilon)\ a^{-1}Y$

- $a^{-1}(a^{-1}L) = a^{-1}(L + b\ \Sigma^*) =$

$= a^{-1}L + a^{-1}(b\ \Sigma^*) = L + b\ \Sigma^* + a^{-1}b\ \Sigma^* + \emptyset = a^{-1}L$

$a^{-1}(a^{-1}L) = a^{-1}L$ (pas nouveau)

- $b^{-1}(a^{-1}L) = b^{-1}(L + b\ \Sigma^*) =$

$= b^{-1}L + b^{-1}(b\ \Sigma^*) = L + \Sigma^* = \Sigma^*$

$b^{-1}(a^{-1}L) = \Sigma^*$ (nouveau)

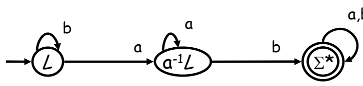
- $a^{-1}(b^{-1}(a^{-1}L)) = a^{-1}(\Sigma^*) = \Sigma^*$
- $b^{-1}(b^{-1}(a^{-1}L)) = b^{-1}(\Sigma^*) = \Sigma^*$

14

$Q(\Sigma^*ab\Sigma^*)$

- $a^{-1}L = L + b\ \Sigma^*$
- $b^{-1}L = \Sigma^*\ ab\ \Sigma^* = L$
- $a^{-1}(a^{-1}L) = a^{-1}L$
- $b^{-1}(a^{-1}L) = \Sigma^*$
- $a^{-1}(b^{-1}(a^{-1}L)) = \Sigma^*$
- $b^{-1}(b^{-1}(a^{-1}L)) = \Sigma^*$

δ	a	b
$\rightarrow L$	$a^{-1}L$	L
$a^{-1}L$	$a^{-1}L$	Σ^*
$\leftarrow \Sigma^*$	Σ^*	Σ^*



15

Problème 2

- **Donnée** : *A* un AFD complet dont chaque état est accessible depuis l'état initial
- **Problème** : construire un AFD minimal qui reconnaisse le même langage que *A*.
- **Idée** : fusionner les états équivalents. En pratique, l'algorithme est fondé sur un principe de *séparation des états* ...

16

Équivalence d'états

Étant donné *A* un AFD, p et q $\in Q$ sont équivalents ($p\approx q$) si

- $L_p(A) = L_q(A)$

c.à.d.

- $\forall w\in\Sigma^*$

$\delta(p,w) \in F$ et $\delta(q,w) \in F$

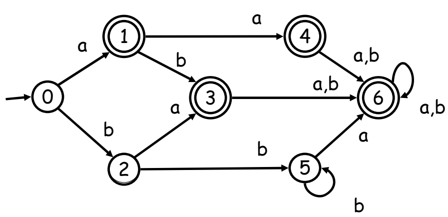
ou

$\delta(p,w) \notin F$ et $\delta(q,w) \notin F$

17

Exemple

$\forall w\in\Sigma^* \begin{cases} \delta(p,w) \in F \text{ et } \delta(q,w) \in F \\ \text{ou} \\ \delta(p,w) \notin F \text{ et } \delta(q,w) \notin F \end{cases}$



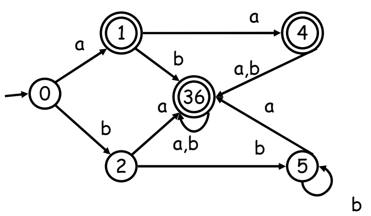
$0\approx1$? Non car $0\notin F$ et $1\in F$

$3\approx6$? Oui car $3,6\in F$, $\delta(3,a)=\delta(6,a)$ et $\delta(3,b)=\delta(6,b)$

18

Exemple

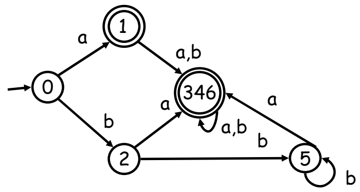
$$\forall w \in \Sigma^* \begin{cases} \delta(p,w) \in F \text{ et } \delta(q,w) \in F \\ \text{ou} \\ \delta(p,w) \notin F \text{ et } \delta(q,w) \notin F \end{cases}$$



4≈6? Oui car $4,6 \in F$, $\delta(4,a)=\delta(6,a)$ et $\delta(4,b)=\delta(6,b)$

Exemple

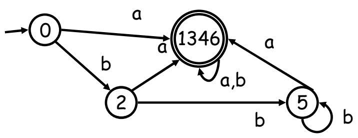
$$\forall w \in \Sigma^* \begin{cases} \delta(p,w) \in F \text{ et } \delta(q,w) \in F \\ \text{ou} \\ \delta(p,w) \notin F \text{ et } \delta(q,w) \notin F \end{cases}$$



1≈6? Oui car $1,6 \in F$, $\delta(1,a) = \delta(6,a)$ et $\delta(1,b) = \delta(6,b)$

Exemple

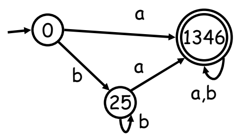
$$\forall w \in \Sigma^* \begin{cases} \delta(p,w) \in F \text{ et } \delta(q,w) \in F \\ \text{ou} \\ \delta(p,w) \notin F \text{ et } \delta(q,w) \notin F \end{cases}$$



2≈5? Oui car $2,5 \notin F$, $\delta(2,a)=\delta(5,a)$ et $\delta(2,b)=\delta(5,b)$

Exemple

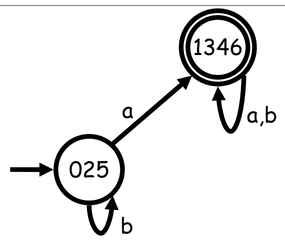
$$\forall w \in \Sigma^* \begin{cases} \delta(p,w) \in F \text{ et } \delta(q,w) \in F \\ \text{ou} \\ \delta(p,w) \notin F \text{ et } \delta(q,w) \notin F \end{cases}$$



0≈2? Oui car $0,2 \notin F$, $\delta(0,a)=\delta(2,a)$ et $\delta(0,b)=\delta(2,b)$

Exemple

$$\forall w \in \Sigma^* \begin{cases} \delta(p,w) \in F \text{ et } \delta(q,w) \in F \\ \text{ou} \\ \delta(p,w) \notin F \text{ et } \delta(q,w) \notin F \end{cases}$$



0≈1? Non car $0 \notin F$ et $1 \in F$

Classe d’équivalence et automate associé

La relation ≈ est une relation d’équivalence (elle est réflexive, symétrique, transitive).

Si q est un état, on note [q] l’ensemble des états qui lui sont équivalents et on définit l’automate des classes d’équivalence :

Classe d'équivalence et automate associé

Étant donné un AFD $A = (\Sigma, Q, \delta, q_0, F)$, l'automate minimal associé à A est :

$$\mu A = (\Sigma, Q', \delta', [q_0], F')$$

- $Q' = \{[q], q \in Q\}$
- $F' = \{[f], f \in F\}$
- $\delta' = \{ ([p],\sigma,[q])$ tels que $\exists\, p' \in [p], \exists\, q' \in [q]$ $(p',\sigma,q') \in \delta \}$

25

Justification

- 3 étapes:
  - L'automate μA des classes d'équivalence de A est bien défini, réduit et $L(\mu A) = L(A)$.
  - Pour tout AFD B tel que $L(B) = L(A)$, $\#\text{états}(B) \geq \#\text{états}(\mu A)$
  - Tout automate minimal B tel que $L(B) = L(A)$, est **isomorphe** à A

Il existe une bijection ϕ entre les états de A et ceux de B qui préserve
- les états spéciaux (initial et d'acceptation)
- les transitions : $\forall p,q \in Q_A,\ \delta_A(p,a)=q \Leftrightarrow \delta_B(\phi(p),a)=\phi(q)$

26

μA est bien défini, réduit et $L(\mu A)=L(A)$

- Soient p et q deux états de A, $p \approx q$:
  - p et q sont tous deux soit dans F soit dans $Q \backslash F$. Les états terminaux de μA sont bien définis.
  - Si $L_p(A)=L_q(A)$ alors $\forall a \in \Sigma$, $L_{\delta(p,a)}(A) = L_{\delta(q,a)}(A)$. Les transitions de μA sont bien définies.
  - Si $w=w_1 \ldots w_n \in \Sigma^*$, et $q_k = \delta(i,w_1 \ldots w_k)$ alors $[q_k] = \delta'([i],w_1 \ldots w_k)$. $\Rightarrow L(A)=L(\mu A)$
  - μA est un automate réduit par construction

27

Pour tout AFD B tel que $L(B)=L(A)$, $\#\text{états}(B) \geq \#\text{états}(\mu A)$

- On suppose tous les états de B accessibles et B complet.
  Soit $B = \langle \Sigma, Q_B, i_B, F_B, \delta_B \rangle$ tel que $L(B)=L(A)$.
- Soit g: $Q_B \rightarrow Q'$ l'application définie par
  $\forall q \in Q_B, \exists u \in \Sigma^* : \delta_B(i_B,u)=q$.
  $g(q) := \delta_{\mu A}([i],u)$
- Comme μA est réduit et $L(\mu A)= L(A)=L(B)$, cette application est bien définie et surjective
  (donc $\# Q_B \geq \#Q'$)

28

Tout automate minimal B tel que $L(B)=L(A)$, est isomorphe à μA

- En ce cas, comme g surjective et $\#\text{états}(\mu A)=\#\text{états}(B)$

g définit une bijection

μA et B sont isomorphes

Reste à construire l'automate réduit

29

Sur les quotients gauches

- L'automate Q(L) des quotients gauches défini comme $\{L_q(A):q \in Q\} = Q(L)$ est-il bien minimal?
- Supposons, par l'absurde qu'il ne le soit pas. Alors il existe au moins p et q, deux états de l'automate tels que $L_p(A)=L_q(A)$, par définition de Q(L).
  - Si tel est le cas, par définition de l'équivalence, $p \approx q$.
  - Il s'ensuit que p et q peuvent être fusionnés, contredisant la minimalité de l'automate des quotients gauches.

30