LYCÉE LECONTE DE LISLE

Théorème de Myhill-Nerode et applications

Vincent Picard

Résiduels

Soit Σ un alphabet, L un langage sur Σ et u un mot sur Σ, on appelle **quotient à gauche** de L par u le langage :

$$u^{-1}L = \{v \in \Sigma^* \,/\, uv \in L\}$$

- C'est une notation, l'inverse d'un mot n'existe pas
- On appelle aussi les quotients d'un langage les **résiduels** du langage
- Exemple :
  - $L_1 = \{babbbb, baba, ababa, b, bba, babb, ba\}$
  - alors $(ba)^{-1}L_1 = \{bbbb, ba, bb, \varepsilon\}$

Exemples de résiduels : exercice

- $a^{-1}(a^*) =$
- $a^{-1}(a^*b^*) =$
- $a^{-1}(b^*a^*) =$
- $(ba)^{-1}\{\text{mots qui commencent par baba}\} =$
- $a^{-1}\{u \ / \ |u|_a \text{ est pair}\} =$
- $a^{-1}(ab)^* =$
- $b^{-1}(ab)^* =$

Exemples de résiduels : solutions

- $a^{-1}(a^*) = a^*$
- $a^{-1}(a^*b^*) = a^*b^*$
- $a^{-1}(b^*a^*) = a^*$
- $(ba)^{-1}\{\text{mots qui commencent par baba}\} = \{\text{mots qui commencent par ba}\}$
- $a^{-1}\{u \ / \ |u|_a \text{ est pair}\} = \{u \ / \ |u|_a \text{ est impair}\}$
- $a^{-1}(ab)^* = b(ab)^*$
- $b^{-1}(ab)^* = \varnothing$

Une propriété utile pour calculer les résiduels

Soit L un langage sur l'alphabet Σ, u et v deux mots sur Σ alors

$$(uv)^{-1}L = v^{-1}u^{-1}L$$

Démonstration

Pour tout mot w, on a les équivalences :

$$\begin{aligned} w \in v^{-1}u^{-1}L &\Leftrightarrow vw \in u^{-1}L \\ &\Leftrightarrow uvw \in L \\ &\Leftrightarrow w \in (uv)^{-1}L \end{aligned}$$

- Exemple : $(ab)^{-1}(ab)^* = b^{-1}a^{-1}(ab)^* = b^{-1}b(ab)^* = (ab)^*$

Exercice

Soit $\Sigma = \{a, b\}$. En utilisant la propriété précédente, répondre aux problèmes suivants

- Soit $L_1 = (ab)^*$, combien L_1 a-t-il de résiduels ?
- Soit $L_2 = \{a^n b^n, n \in \mathbb{N}\}$, combien L_2 a-t-il de résiduels ?

Exercice (solutions)

Soit $\Sigma = \{a, b\}$. En utilisant la propriété précédente, répondre aux problèmes suivants

- Soit $L_1 = (ab)^*$, combien L_1 a-t-il de résiduels ?
  - $\varepsilon^{-1}L_1 = L_1$
  - $b^{-1}L_1 = \varnothing$
  - $a^{-1}L_1 = b(ab)^*$
  - $a^{-1}\varnothing = b^{-1}\varnothing = \varnothing$
  - $a^{-1}b(ab)^* = \varnothing$
  - $b^{-1}b(ab)^* = (ab)^* = L_1$
  - **Conclusion :** il y a un nombre fini de résiduels $\varnothing$, $b(ab)^*$ et $(ab)^* = L_1$.
- Soit $L_2 = \{a^n b^n, n \in \mathbb{N}\}$, combien L_2 a-t-il de résiduels ?
  - $\varepsilon^{-1}L_2 = L_2$
  - $b^{-1}L_2 = \varnothing$
  - $a^{-1}L_2 = \{a^n b^{n+1}, n \in \mathbb{N}\} = H_1$
  - $a^{-1}H_1 = \{a^n b^{n+2}, n \in \mathbb{N}\} = H_2$
  - $a^{-1}H_2 = \{a^n b^{n+3}, n \in \mathbb{N}\} = H_3$
  - ...
  - $(a^k)^{-1} = \{a^n b^{n+k}, n \in \mathbb{N}\}$
  - **Conclusion :** il y a un nombre infini de résiduels

Équivalence de Myhill-Nerode

Soit L un langage sur l'aphabet Σ, u et v sont dit équivalents au sens de Myhill-Nerode lorsque $u^{-1}L = v^{-1}L$. On notera $u \sim_L v$.

$\sim_L$ est une relation d'équivalence sur Σ^*.

- Deux mots u et v sont équivalents lorsqu'ils conduisent au même résiduel par quotient à gauche.
- $u \sim_L v$ signifie qu'après avoir lu u on va reconnaitre les mêmes mots que si on avait commencé par lire v à la place de v.
- Exemples
  - Pour $L = a^*$, $\quad a^{k_1} \sim_L a^{k_2}$ pour tous $k_1, k_2 \in \mathbb{N}$
  - Pour $L = (ab)^*$, $\quad a^5b^3 \sim_L a^7b^2$ car après avoir lu a^5b^3 ou a^7b^2, il faut finir par un mot de b^* pour être dans L.

Théorème de Myhill-Nerode (1957)

Le langage L est **régulier** (ou **reconnaissable** par automate fini)

$\Leftrightarrow$

L possède un nombre **fini** de résiduels

$\Leftrightarrow$

le nombre de classes d'équivalences de $\sim_L$ est **fini**

- La seconde équivalence est évidente :
  - Il y a bijection entre les résiduels de L et les classes d'équivalence de $\sim_L$.
  - $\varphi : u^{-1}L \mapsto \mathrm{Cl}(u)$
- Pour la première équivalence, on va donner une preuve constructive.

Un langage régulier possède un nombre fini de résiduels

- Soit L un langage reconnu par un afd **complet** $A = (Q, q_0, F, \delta)$. Pour tout état $q \in Q$ on note L_q le langage des mots reconnus **à partir de l'état q** :

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

- Soit u un mot sur Σ alors

$$\begin{aligned} u^{-1}L &= \{v \ / \ uv \in L\} \\ &= \{v \ / \ \delta^*(q_0, uv) \in F\} \\ &= \{v \ / \ \delta^*(\delta^*(q_0, u), v) \in F\} \\ &= L_{\delta^*(q_0, u)} \end{aligned}$$

- Ainsi, un résiduel de L est nécessairement l'un des langages L_q avec $q \in Q$, et comme l'ensemble des états de l'automate est fini, il ne peut y avoir qu'un nombre fini de résiduels.

Si un langage possède un nombre fini de résiduels alors il est régulier

- Notons R l'ensemble fini des résiduels d'un langage L. On construit l'afd suivant : $A = (R, q_0, F, \delta)$ avec
  - Les états sont les résiduels R
  - L'état initial est le résiduel $\varepsilon^{-1}L = L$
  - Les états finaux sont les résiduels qui contiennent le mot ε (Si on ajoute plus rien au mot déjà lu alors il est dans le langage L).
  - Si ρ est un résiduel et a une lettre alors $\delta(\rho, a) = a^{-1}\rho$.
- Remarquons deux choses importantes :
  - L'automate est complet
  - δ est bien défini : car le résiduel d'un résiduel est toujours un résiduel

L'automate des résiduels reconnaît bien L

Montrons que cet automate sert à calculer les quotients de L par tout mot u : $\delta^*(q_0, u) = u^{-1}L$. On le prouve par récurrence sur la longueur du mot u :

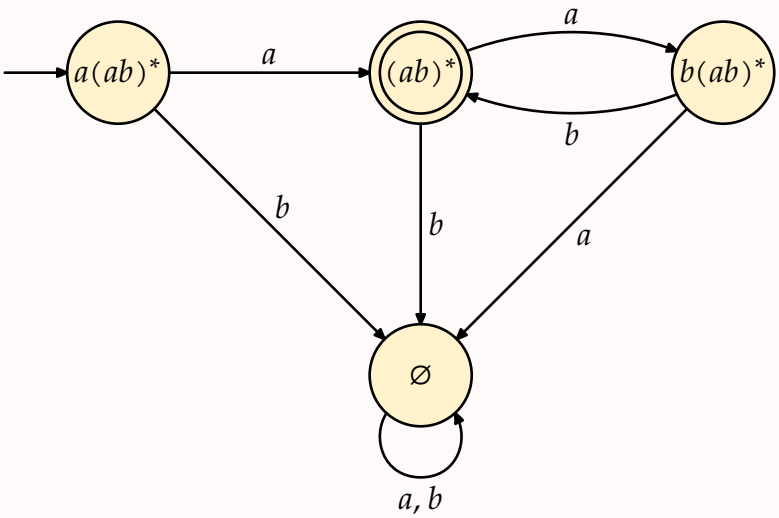
- **Initialisation** : $\delta^*(q_0, \varepsilon) = q_0 = L = \varepsilon^{-1}L$
- **Hérédité** : on suppose la propriété vraie pour tout mot de longueur n et soit $u = va$ un mot de longueur $n + 1$, alors

$$\begin{aligned}
\delta^*(q_0, va) &= \delta(\delta^*(q_0, v), a) \\
&= \delta(v^{-1}L, a) \text{ par hypothèse de récurrence} \\
&= a^{-1}v^{-1}L \\
&= (va)^{-1}L
\end{aligned}$$

- Et on en déduit : $u \in L \Leftrightarrow \varepsilon \in u^{-1}L \Leftrightarrow \varepsilon \in \delta^*(q_0, u) \Leftrightarrow \delta^*(q_0, u) \in F$.

Exemple

- Construisons l'automate des résiduels du langage $a(ab)^*$:



- car $a^{-1}(ab)^* = b(ab)^*, \quad b^{-1}b(ab)^* = (ab)^*, \quad etc$
- le résiduel $(ab)^*$ est final car $\varepsilon \in (ab)^*$

Application : montrer qu'un langage n'est pas régulier

Sans le lemme de l'étoile...

- $L = \{a^n b^n, n \in \mathbb{N}\}$ n'est pas régulier.
- $L = \{a^p b^q \ / \ p < q\}$ n'est pas régulier.

Application : Automate minimal

Si L est un langage à $n \in \mathbb{N}^*$ résiduels alors il existe un plus petit automate fini déterministe complet à n états qui reconnaît L. Cet automate est appelé **automate minimal**.

Démonstration

- Le sens réciproque de la preuve du théorème de Myhill-Nerode prouve l'existence d'un tel automate.
- Réciproquement, si $A = (Q, q_0, F, \delta)$ est un afd complet qui reconnaît L alors, et qu'on considère deux mots u et v qui ne sont pas équivalents au sens de Myhill-Nerode : $u^{-1}L \neq v^{-1}L$ alors nécessairement $\delta^*(q_0, u) \neq \delta^*(q_0, v)$. Ceci implique que Q possède au moins autant d'états que de classes d'équivalences de $\sim_L$ c'est-à-dire n.
- On peut aussi montrer que si un tel automate possède exactement n états, alors les langages L_q correspondant aux mots acceptés depuis q sont les résiduels de L et que l'automate est isomorphe à l'automate des résiduels (unicité).

Minimisation d'automates par fusion d'états

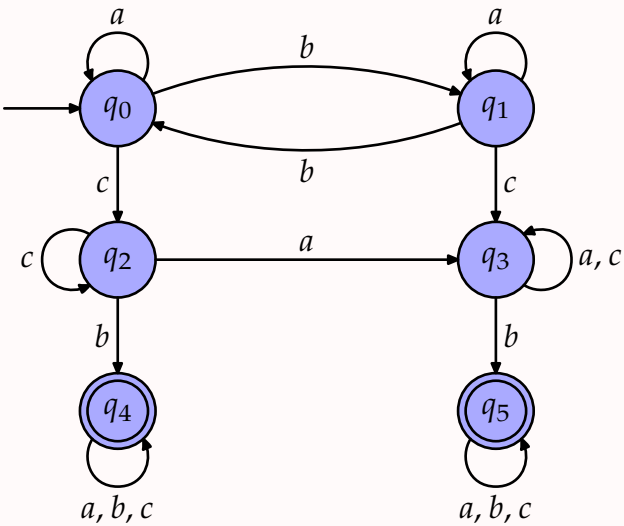
Soit $A = (Q, q_0, F, \delta)$ un automate fini déterministe complet.

Soit q et q' deux états de l'automate, on dit que q et q' sont **équivalents au sens de Nerode** lorsque $L_q = L_{q'}$.

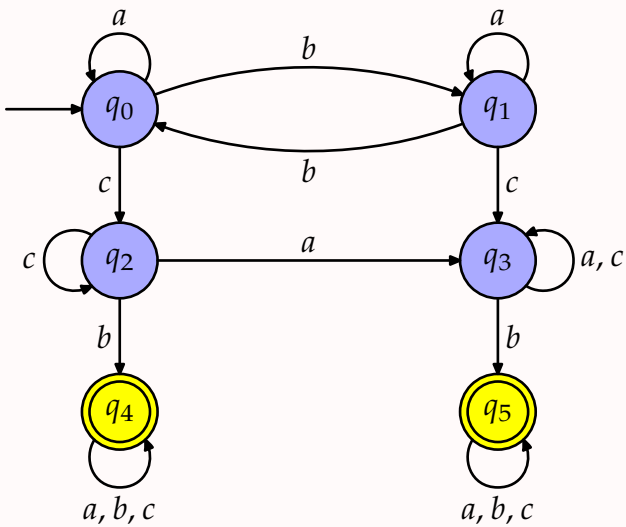
- Ainsi, ces états sont indistingables, quand on tombe sur q on va reconnaître par la suite exactement les mêmes mots que si on état tombé sur q'.
- Autrement dit, ces deux états correspondent à un même résiduel du langage reconnu.
- Pour minimiser un automate, il suffit de **fusionner** les états équivalents en 1 seul, jusqu'à aboutir sur l'automate minimal.
- Il existe des algorithmes permettant de détecter efficacement quels états sont équivalents ou pas dans un automate (algorithme par raffinement de Moore) et donc de minimiser un automate donné.

Exemple

On veut minimiser l'automate suivant :

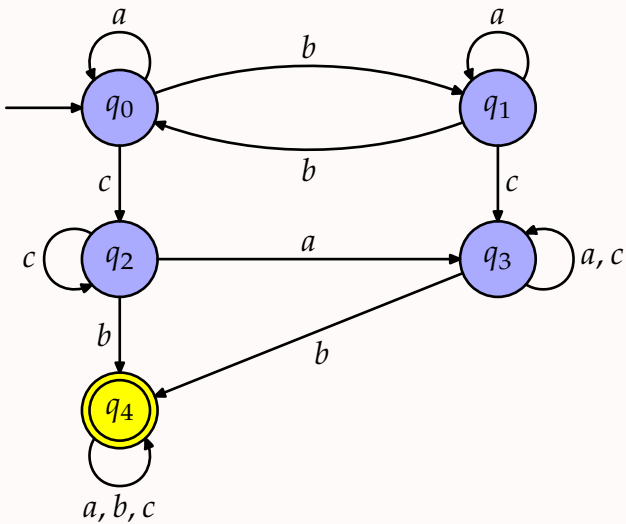


Exemple



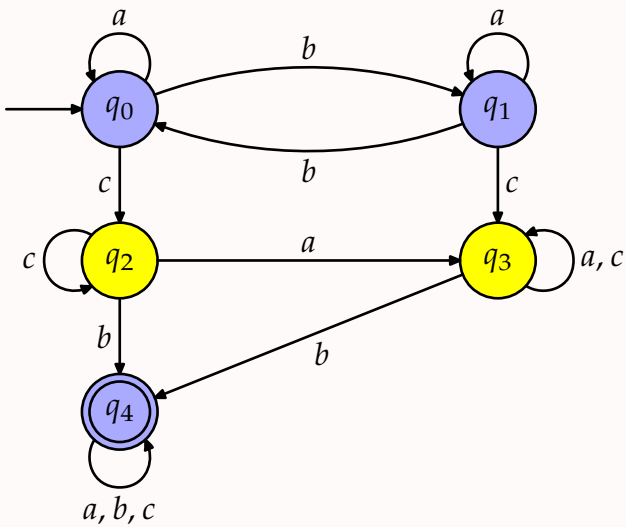
- On remarque que $L_{q_4} = L_{q_5} = \{a, b, c\}^*$.

Exemple



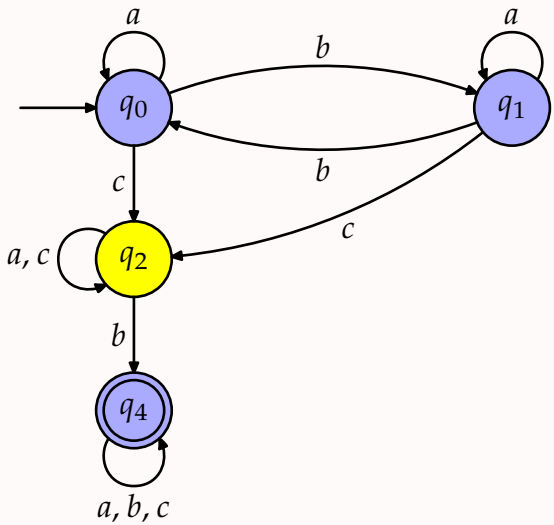
- On fusionne q_4 et q_5 :
  - On supprime q_5 et ses transitions sortantes
  - Toutes les transitions qui menaient à q_5 mènent désormais à q_4

Exemple



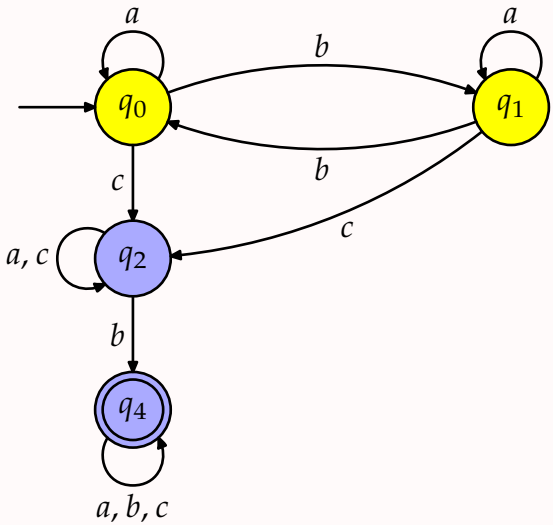
- On remarque que $L_{q_2} = L_{q_3} = \{a, c\}^* \, b L_{q_4}$.

Exemple



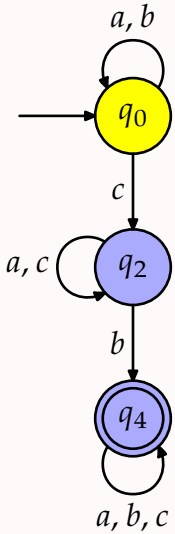
- On fusionne q_2 et q_3 :
  - On supprime q_3 et ses transitions sortantes
  - Toutes les transitions qui menaient à q_3 mènent désormais à q_2

Exemple



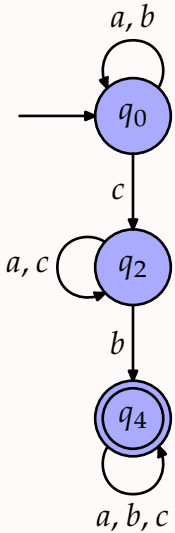
- On remarque que $L_{q_0} = L_{q_1} = \{a, b\}^* c L_{q_2}$.

Exemple



- On fusionne q_0 et q_1 :
  - On supprime q_1 et ses transitions sortantes
  - Toutes les transitions qui menaient à q_1 mènent désormais à q_0

Exemple



- La minimisation est terminée car les états sont maintenant 2 à 2 non équivalents (L_{q_0}, L_{q_2}, L_{q_4} sont distincts 2 à 2)
- L'automate obtenu est l'automate minimal qui reconnaît le langage initial dénoté par : $(a|b)*c(a|c)*b(a|b|c)*$