LifLF – Théorie des langages formels

Sylvain Brandel

2019 – 2020

sylvain.brandel@univ-lyon1.fr

CM 6

MINIMISATION DES ÉTATS

Minimisation des états
Théorème de Myhill – Nerode

Contexte à droite relativement à un langage

- Définition

  Soit $L \subseteq \Sigma^*$ un langage

  On définit pour un mot $u \in \Sigma^*$ son contexte à droite relativement à L :

$$R_L(u) = \{ z \in \Sigma^* \mid uz \in L \}$$

Minimisation des états

Théorème de Myhill – Nerode

Equivalence des mots suivant un langage

- Définition

  Soit $L \subseteq \Sigma^*$ un langage et $x, y \in \Sigma^*$ deux mots.

  On dit que x et y sont équivalents suivant L, et on note $x \approx_L y$,
  si pour tout mot z de Σ^* :

  $$xz \in L \text{ ssi } yz \in L$$

- Propriété

  $\approx_L$ est une relation d'équivalence

  On note $[w]_L$ la classe d'équivalence du mot w.

Minimisation des états

Théorème de Myhill – Nerode

Equivalence des mots suivant un langage

- Exemple $L = (ab \cup ba)^*$

  Chercher les classes suivant $\approx_L$ dans Σ^*

  - $[\varepsilon] = L$
  - $[a] = La$
  - $[b] = Lb$
  - $[aa] = L\,(aa \cup bb)\,\Sigma^*$
  - $[bb] = [aa]$
  - $[bba] = [aa]$
  - $[aaa] = [aa]$
  - $[ab] = L = [ba] = [\varepsilon]$ car $ab \in L$

  ...

  On montre facilement par récurrence qu'on a toutes les classes

  - On obtient 4 classes d'équivalence :

$$\Sigma^* = L \cup La \cup Lb \cup L\,(aa \cup bb)\,\Sigma^*$$

Minimisation des états
Théorème de Myhill – Nerode

Equivalence des mots suivant un automate

- Définition

  Soit $M = (K, \Sigma, \delta, s, F)$ un automate déterministe (fini), et $x, y \in \Sigma^*$ deux mots.

  On dit que x et y sont équivalents relativement à M, on note $x \sim_M y$,

  ssi il existe un état q de K tel que :

  $$(s, x) \vdash_M^* (q, \varepsilon) \text{ et } (s, y) \vdash_M^* (q, \varepsilon)$$

- M étant déterministe, si on note $q_M(x)$ l'état auquel on parvient dans M en lisant x, on a :

  $$x \sim_M y \text{ ssi } q_M(x) = q_M(y)$$

Minimisation des états

Théorème de Myhill – Nerode

Equivalence des mots suivant un automate

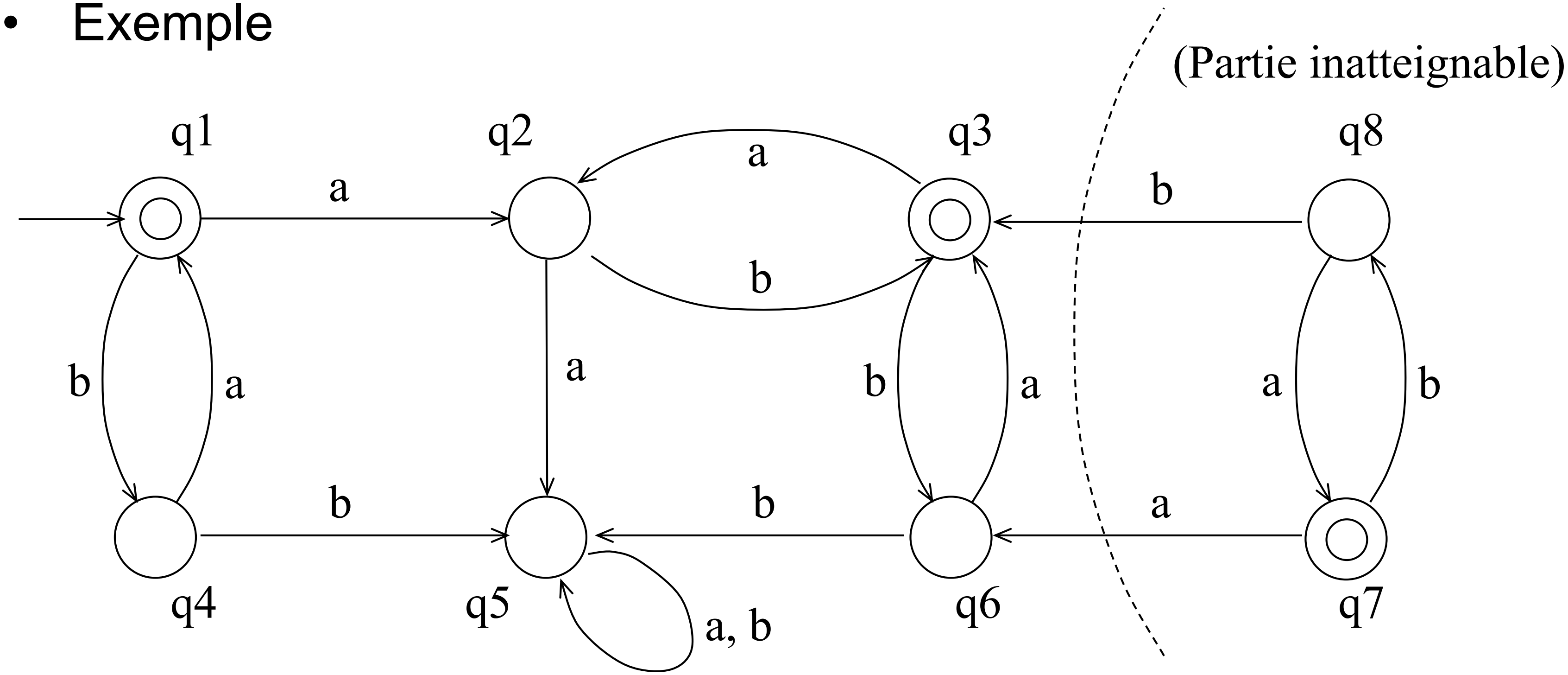
- Propriétés
  - $\sim_M$ est une relation d'équivalence.
  - On note E_q la classe d'équivalence des mots x tels que $q_M(x) = q$

    $E_q = \varnothing$ si l'état est inatteignable
  - Il y a autant de classes d'équivalence que d'états atteignables (accessibles).

Minimisation des états
Théorème de Myhill – Nerode

Equivalence des mots suivant un automate

- Exemple



$[\varepsilon] = E_{q1} \cup E_{q3}$
$[a] = E_{q2}$
$[b] = E_{q4} \cup E_{q6}$
$[aa] = E_{q5}$

Minimisation des états
Théorème de Myhill – Nerode

Equivalence des mots suivant un automate

- $E_q = L(M^q)$ avec $M^q = (K, \Sigma, \delta, s, \{q\})$

- Théorème

Pour tout automate fini déterministe M et deux mots x, y $\in \Sigma^*$, on a :

si $x \sim_M y$ alors $x \approx_{L(M)} y$

(on dit que $\sim_M$ raffine $\approx_{L(M)}$)

Les classes de $\sim_M$ sont plus petites (et incluses) dans les classes de $\approx_{L(M)}$

Minimisation des états
Théorème de Myhill – Nerode

Automate standard

- Théorème de Myhill – Nerode

  Soit $L \subseteq \Sigma^*$ un langage rationnel.
  Il existe un automate déterministe ayant $| \Sigma^* / \approx_L |$ états acceptant L.
  (C'est-à-dire autant d'états que le nombre de classes d'équivalence suivant $\approx_L$.)

- Cet automate a le plus petit nombre d'états possibles
- Il n'y a qu'un seul automate vérifiant cela
- On appelle cet automate l'automate standard de L
  (ou automate minimal de L).

Minimisation des états
Théorème de Myhill – Nerode

Automate standard

- Preuve (constructive) :

  $M = (K, \Sigma, \delta, s, F)$ automate standard d'un langage L, avec
  - $K = \{ [x], x \in \Sigma^* \}$
  - $s = [\varepsilon]$
  - $F = \{ [x], x \in L \}$
  - δ : définie par $\delta([x], a) = [xa]$

Minimisation des états

Théorème de Myhill – Nerode

Automate standard

- K est fini car L est reconnu par un automate déterministe M'

  $$| \Sigma^* / \sim_{M'} | \geq | \Sigma^* / \approx_L |$$

- δ bien définie : si [x] = [y] alors [xa] = [ya]

  (clair car : $x \approx_L y \Rightarrow xa \approx_L ya$)

- L = L(M)

  (i) on montre d'abord que $([x], y) \vdash_M^* ([xy], \varepsilon)$ (par induction sur |y|)

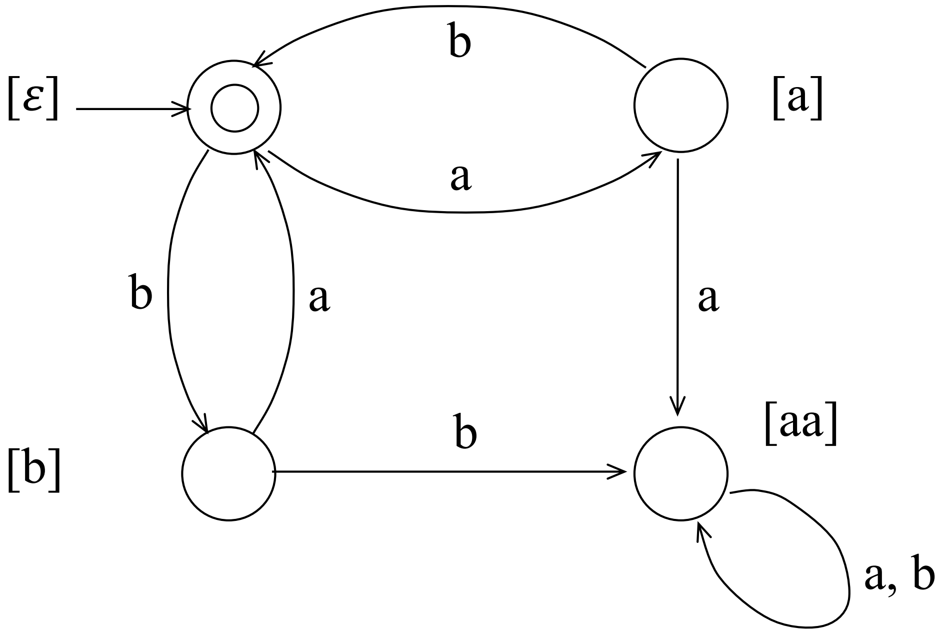
  (ii) $\forall x \in \Sigma^*$, $x \in L(M)$ ssi $([\varepsilon], x) \vdash_M^* ([x], \varepsilon)$ et $[x] \in F$ ie $x \in L$

  (par définition de F)

Minimisation des états
Théorème de Myhill – Nerode

Automate standard

- Exemple



Minimisation des états
Théorème de Myhill – Nerode

Corollaire (Myhill – Nerode)

- Théorème

L est rationnel ssi $\approx_L$ a un nombre fini de classes d'équivalence.

Minimisation des états

Minimisation d'un automate donné

- Définition

  Soit $M = (K, \Sigma, \delta, s, F)$ un automate fini déterministe.

  On note $M(q)$, pour $q \in K$, l'automate $(K, \Sigma, \delta, q, F)$.

  (Avec cette notation, $M = M(s)$.)

  On dit que p et q sont deux états équivalents de M, et on note $p \equiv q$,

  ssi $L(M(p)) = L(M(q))$.

- $p \equiv q$ ssi $\forall\, w \in \Sigma^*$,
  - soit $(p, w) \vdash_M^* (f_1, \varepsilon)$ et $(q, w) \vdash_M^* (f_2, \varepsilon)$ avec $f_1, f_2 \in F$
  - soit $(p, w) \vdash_M^* (g_1, \varepsilon)$ et $(q, w) \vdash_M^* (g_2, \varepsilon)$ avec $g_1, g_2 \notin F$

  classe d'équivalence des mots x tels que $q_M(x) = p$

Minimisation des états

Minimisation d'un automate donné

- Propriété

  $M = (K, \Sigma, \delta, s, F)$ sans états inatteignables ($E_p \neq \varnothing\ \forall\ p \in K$).

  Soient p et $q \in K$. $(\ \exists\ v \in \Sigma^* \mid E_p \subset [v]$ et $E_q \subset [v]\ ) \Leftrightarrow p \equiv q$

- Corollaire

  $\equiv$ sur K induit une relation d'équivalence sur $\Sigma^* / \sim_M$ (notée $\approx$)
  définie par $E_p \approx E_q$ ssi $p \equiv q$

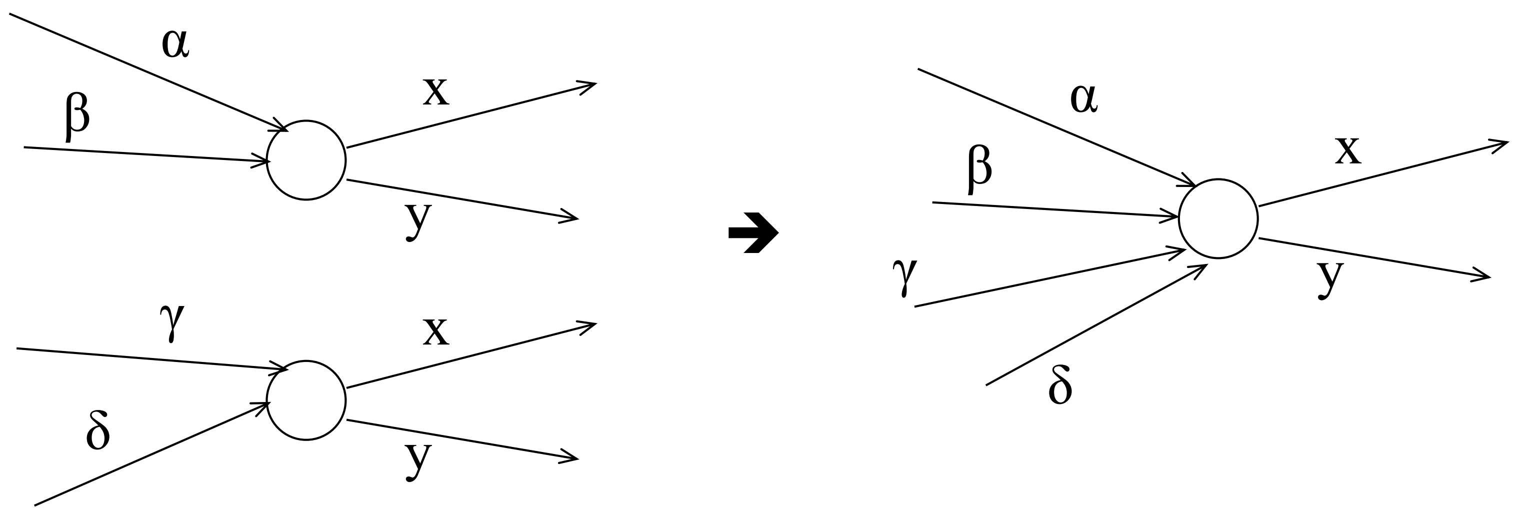
  Chaque classe d'équivalence correspond à une classe de la forme $[v]$.

Minimisation des états

Minimisation d'un automate donné

Deux états équivalents sont ceux qu'on doit fusionner pour obtenir l'automate (minimal) standard.

On garde les flèches entrantes et on oublie les flèches sortantes de l'un des états (qui sont étiquetées par toutes les lettres de Σ)



Minimisation des états

Minimisation d'un automate donné

- Concrètement

On calcule $\equiv$ puis on fait la fusion.

$\equiv$ est calculée comme la limite de $(\equiv_i)_{i \in N}$.

$\equiv_i$ définie par :

$p \equiv_i q$ ssi pour tout mot w de longueur $\leq$ i tel que $(p, w) \vdash_M^* (f_1, \varepsilon)$ et $(q, w) \vdash_M^* (f_2, \varepsilon)$

on a $f_1 \in F \Leftrightarrow f_2 \in F$

$(\text{ou } L(M(p)) \cap (\cup_{k=0}^{i} \Sigma^k) = L(M(q)) \cap (\cup_{k=0}^{i} \Sigma^k))$

$\forall\, i \in N :$ $\quad p \equiv q \Leftrightarrow p \equiv_i q$

et $\quad p \equiv_i q \Leftrightarrow p \equiv_{i-1} q$

Minimisation des états

Minimisation d'un automate donné

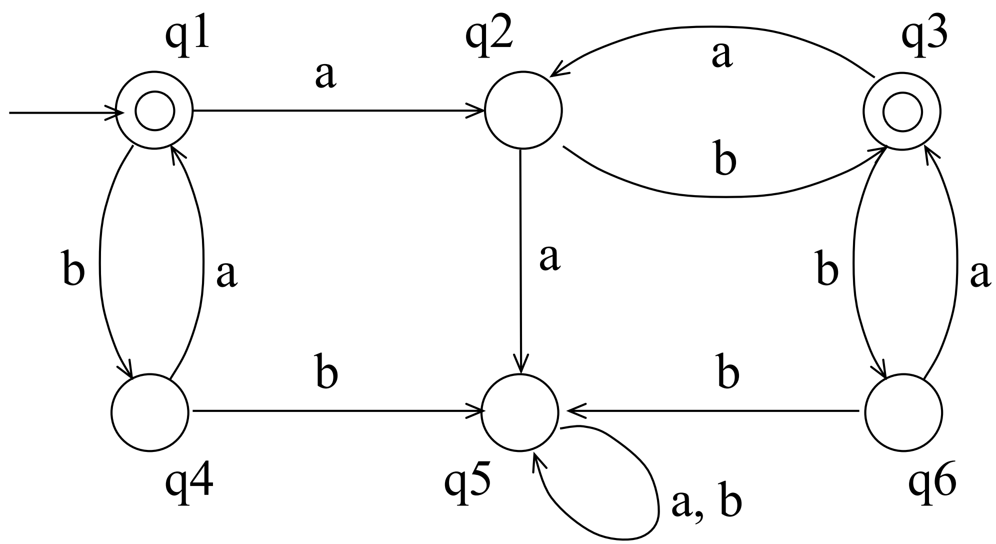
- Propriété

  Pour tout couple $(p, q) \in K^2$ et $n \geq 1$ on a :

  $p \equiv_n q$ ssi $p \equiv_{n-1} q$

  et $\forall\, \sigma \in \Sigma,\ \delta(p, \sigma) \equiv_{n-1} \delta(q, \sigma)$

- Exemple



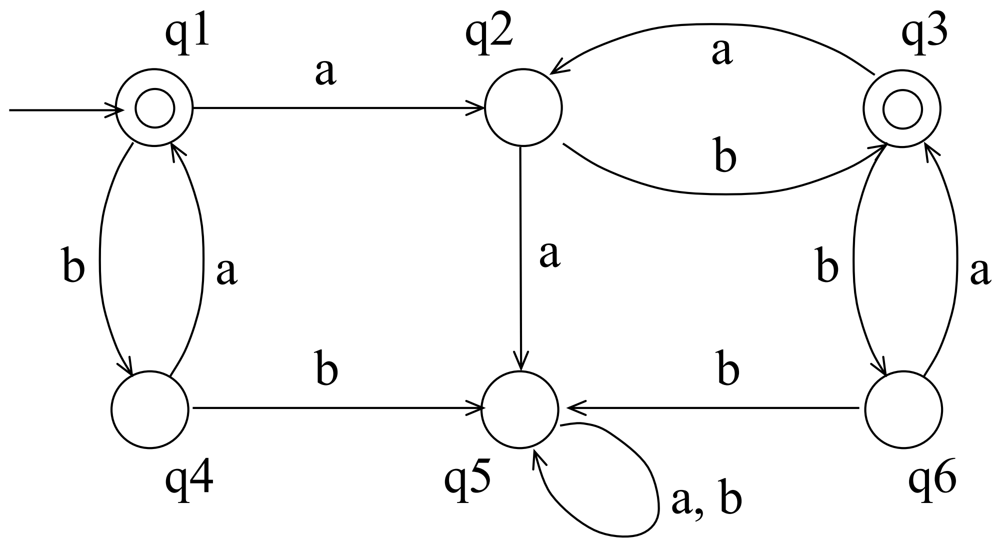
$q_1 \equiv_0 q_3$ car $q_1 \in F$ et $q_3 \in F$

$q_2 \equiv_0 q_4$ car $q_2 \notin F$ et $q_4 \notin F$

Minimisation des états

Minimisation d'un automate donné

- Exemple



$q_1 \equiv_1 q_3$ car $q_1 \equiv_0 q_3$ et $\delta(q_1, a) \equiv_0 \delta(q_3, a)$

$(\delta(q_1, a) = q_2$ et $\delta(q_3, a) = q_2$ et $\{q_2\})$

$\delta(q_1, b) \equiv_0 \delta(q_3, b)$

$(\delta(q_1, b) = q_4$ et $\delta(q_3, b) = q_6$ et $\{q_4, q_6\})$

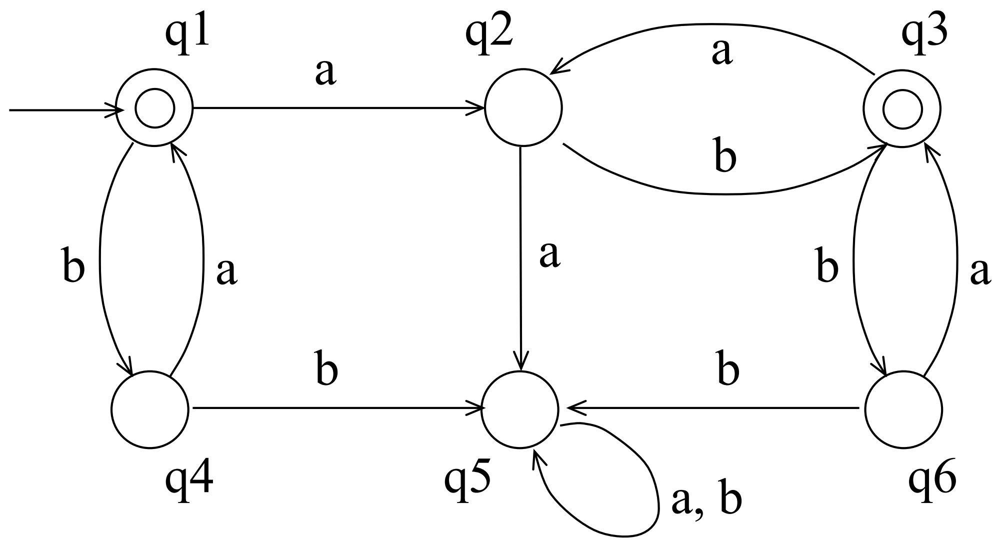
$q_2 \neg\equiv_1 q_4$ car $\delta(q_2, a) \neg\equiv_0 \delta(q_4, a)$

$(\delta(q_2, a) = q_5$ et $\delta(q_4, a) = q_1$ et $\{q_1, q_3\}$ et $\{q_5\})$

Minimisation des états

Minimisation d'un automate donné

- Exemple



$\equiv_0$: $\{q_1, q_3\}, \{q_2, q_4, q_5, q_6\}$

$\equiv_1$: $\{q_1, q_3\}, \{q_2\}, \{q_4, q_6\}, \{q_5\}$

$\equiv_2$: $\{q_1, q_3\}, \{q_2\}, \{q_4, q_6\}, \{q_5\}$

…

(Ça ne change plus)

- Finalement :

$q_1 \equiv q_3$

$q_4 \equiv q_6$