

# Langages réguliers – Automates finis

Franck Cassez  
National ICT Australia & CNRS  
Locked bag 6016  
The University of New South Wales  
Sydney 1466  
Australia  
[franck.cassez@cnrs.irccyn.fr](mailto:franck.cassez@cnrs.irccyn.fr)  
[Franck.Cassez@nicta.com.au](mailto:Franck.Cassez@nicta.com.au)  
<http://www.irccyn.fr/~franck>

30 juin 2010

## Avant propos

Si cette introduction vous est utile dans vos études, merci de m’envoyer un mail [ici](#) ou [là](#) pour m’indiquer votre année d’étude et université ; bien entendu vos commentaires et/ou suggestions pour corriger ou améliorer ce petit document sont les bienvenus.

---

## Table des matières

<b>1</b>	<b>Notations</b>	<b>1</b>
<b>2</b>	<b>Automates finis</b>	<b>2</b>
2.1	Langage accepté par un automate fini . . . . .	2
2.2	Langages réguliers . . . . .	2
2.3	Automates déterministes et non déterministes . . . . .	3
2.4	Propriétés des langages réguliers . . . . .	4
<b>3</b>	<b>Expressions régulières</b>	<b>4</b>
3.1	Opérations régulières . . . . .	4
3.2	Expressions régulières . . . . .	4
<b>4</b>	<b>Automates finis et expressions régulières</b>	<b>5</b>
<b>5</b>	<b>Automates minimaux</b>	<b>6</b>

De nombreux livres traitent ce sujet. Pour des références générales on peut consulter [AG92, Sip96, Wol01, Per90]. Pour la construction des compilateurs l’ouvrage de référence est [ASU86].

## 1 Notations

Soit  $A$  un ensemble. On note  $2^A$  l’ensemble des *parties* de  $A$ . Si  $A$  est fini le cardinal de  $A$  est noté  $|A|$ . Un *alphabet*  $\Sigma$  est un ensemble fini de lettres  $\{a_1, a_2, \dots, a_n\}$ . Un *mot non vide* sur

$\Sigma$  est une séquence  $x_1x_2 \cdots x_k$ ,  $k \geq 1$ , telle que pour tout  $1 \leq i \leq k$ ,  $x_i \in \Sigma$ . Le mot vide ne comportant aucune lettre est noté  $\varepsilon$ . La longueur d'un mot  $w = x_1x_2 \cdots x_k$  est  $k$ . La longueur de  $w$  est notée  $|w|$  et  $|\varepsilon| = 0$ . On note  $\Sigma^*$  l'ensemble des mots de longueur  $k \geq 0$  sur  $\Sigma$ . Un langage sur  $\Sigma$  est un sous-ensemble de mots de  $\Sigma^*$ . Soit  $w = x_1x_2 \cdots x_k$  et  $v = y_1y_2 \cdots y_l$  deux mots sur  $\Sigma$ . La concaténation  $w.v$  de  $w$  et  $v$  est définie par le mot  $w.v = x_1x_2 \cdots x_ky_1y_2 \cdots y_l$ . On a  $w.\varepsilon = \varepsilon.w = w$ .

## 2 Automates finis

### 2.1 Langage accepté par un automate fini

**Définition 1 (Automate fini)** Un automate fini est un tuple  $A = (Q, \Sigma, \delta, q_0, F)$  tel que :

- $Q$  est une ensemble fini d'états,
- $\Sigma$  est un alphabet fini, et on note  $\Sigma_\varepsilon = \Sigma \cup \{\varepsilon\}$ ,
- $\delta : Q \times \Sigma_\varepsilon \longrightarrow 2^Q$  est la fonction de transition,
- $q_0 \in Q$ , est l'état initial,
- $F \subseteq Q$  est l'ensemble des états accepteurs (ou terminaux).

Un automate fini est déterministe si  $\delta$  ne comporte pas de transitions  $(q, \varepsilon, q')$  et pour tout  $(q, a) \in Q \times \Sigma$  on a  $|\delta(q, a)| \leq 1$ ; sinon il est non-déterministe. Il est dit complet si pour tout  $(q, a) \in Q \times \Sigma$ ,  $|\delta(q, a)| \geq 1$ .

Un chemin  $\sigma$  de longueur  $k \geq 1$  dans  $A$  est une séquence de transitions  $\sigma = t_1t_2 \cdots t_k$ ,  $k \geq 1$  telle que pour tout  $1 \leq i \leq k$  on a  $t_i = (s_i, a_i, s_{i+1})$  et  $s_{i+1} \in \delta(s_i)$ .  $s_1$  est l'origine de  $\sigma$  et  $s_{k+1}$  le but de  $\sigma$ . On écrit souvent un chemin  $\sigma$  sous la forme

$$\sigma = s_1 \xrightarrow{a_1} s_2 \xrightarrow{a_2} s_3 \cdots s_k \xrightarrow{a_k} s_{k+1}$$

Un chemin de longueur 0 est constitué de 0 transition donc d'un état qui est à la fois origine et but. La trace du chemin  $\sigma$  est le mot  $a_1a_2 \cdots a_k$  et la trace du chemin vide est  $\varepsilon$ . On note

- $trace(\sigma) = a_1a_2 \cdots a_k$  la trace du chemin  $\sigma$ ,
- $s \xrightarrow{w} s'$  s'il existe un chemin de  $s$  à  $s'$  de trace  $w$ , d'origine  $s$  et de but  $s'$ ,
- $s \xrightarrow{*} s'$  s'il existe  $w \in \Sigma^*$  tel que  $s \xrightarrow{w} s'$ .

Un mot  $w \in \Sigma^*$  est accepté par  $A$  à partir de  $q$  si  $q \xrightarrow{w} q'$  et  $q' \in F$ . Un mot est accepté par  $A$  s'il est accepté à partir de  $q_0$ . Le langage  $L(q, A)$  accepté par  $A$  à partir de  $q \in Q$ , est l'ensemble des mots acceptés par  $A$  à partir de  $q$  :

$$L(q, A) = \{w \in \Sigma^* \mid \exists q' \in F \text{ tel que } q \xrightarrow{w} q'\}$$

Le langage accepté par  $A$  est  $L(q_0, A)$  que l'on note  $L(A)$ .

### 2.2 Langages réguliers

**Définition 2 (Langages réguliers)** Un langage  $L \subseteq \Sigma^*$  est régulier s'il est accepté par un automate fini (non déterministe).

**Remarque 1** On peut sans perte de généralité supposer qu'un langage régulier accepté par un automate fini non déterministe a plusieurs états initiaux ou un seul état final. Soit  $A$  un automate fini non déterministe ayant plusieurs états initiaux : il suffit de prendre  $A'$  identique à  $A$  avec un nouvel état initial unique  $s_0$  et une transition d'étiquette  $\varepsilon$  de  $s_0$  vers chaque état initial de  $A$ . Une construction symétrique sur les états finals permet de se restreindre à un état final unique. On utilisera parfois cette propriété dans la suite.

On peut d'ores et déjà conclure qu'il existe des langages non réguliers. En effet, si l'on accepte que l'ensemble des automates finis sur un alphabet  $\Sigma$  est dénombrable (ce qui est assez facile à montrer), il est clair que l'ensemble des langages est plus grand que l'ensemble des langages

réguliers : l'ensemble de tous les langages sur  $\Sigma$  est égal à l'ensemble des sous-ensembles de  $\Sigma^*$  (qui lui est dénombrable). Or l'ensemble des sous-ensembles d'un ensemble (infini) dénombrable n'est pas dénombrable (pour le prouver on utilise la méthode de *diagonalisation* de Cantor).

**Théorème 1** *Il existe des langages non réguliers.*

Un autre théorème peut aussi être obtenu pour le langage *inverse* d'un langage : si  $L$  est un ensemble des mots, l'inverse de  $L$ , noté  $\tilde{L}$ , est constitué de l'ensemble des mots de  $L$  écrits à l'envers *i.e.*  $\tilde{L} = \{a_1 a_2 \cdots a_n, n \geq 0 \mid a_n a_{n-1} \cdots a_2 a_1 \in L\}$ .

**Théorème 2** *La classe des langages réguliers est fermée par passage au langage inverse.*

**Preuve 1** *Si  $L$  est régulier il est accepté par un automate fini  $A$ . On construit donc  $A'$  tel que : les états initiaux<sup>1</sup> (resp. terminaux) de  $A'$  sont les états terminaux (resp. initiaux) de  $A$  ; les transitions de  $A$  du type  $(q, e, q')$  deviennent  $(q', e, q)$  dans  $A'$ .*

### 2.3 Automates déterministes et non déterministes

**Théorème 3** *Soit  $L \subseteq \Sigma^*$ . Si  $L$  est accepté par un automate fini non déterministe alors  $L$  est accepté par un automate fini déterministe.*

**Preuve 2**

*Soit  $A = (Q, \Sigma, \delta, q_0, F)$  un automate fini non déterministe tel que  $L(A) = L$ . Soit  $q \in Q$  et  $a \in \Sigma$ . On définit  $E_a(q)$  comme étant l'ensemble des états  $q' \in Q$  tels qu'il existe un chemin dans  $A$  de  $q$  à  $q'$  constitué d'un nombre fini de transitions  $\varepsilon$  suivi de  $a$  :*

$$E_a(q) = \{q'' \in Q \mid q \xrightarrow{\varepsilon^*} q' \xrightarrow{a} q''\} \quad (\text{E})$$

*On définit ensuite l'automate fini déterministe  $B = (S, \Sigma, \delta', s_0, F')$  par :*

- $S = 2^Q$ ,
- $s_0 = \{q' \in Q \mid q \xrightarrow{\varepsilon^*} q'\}$ ,
- pour  $s \in 2^Q$ ,  $a \in \Sigma$   $\delta'(s, a) = \cup_{x \in s} E_a(x)$
- $F' = \{s \in 2^Q \mid s \cap F \neq \emptyset\}$ .

*$B$  est déterministe et  $L(B) = L(A)$ . La méthode de construction de  $B$  est souvent appelée subset construction.*

On obtient donc le corollaire suivant :

**Corollaire 1** *Soit  $L \subseteq \Sigma^*$ .  $L$  est accepté par un automate fini non déterministe ssi  $L$  est accepté par un automate fini déterministe.*

De ce corollaire on peut conclure que :

**Théorème 4** *La classe des langages réguliers est fermée par passage au complémentaire.*

**Preuve 3** *Soit  $L$  un langage régulier. Alors  $L$  est accepté par un automate fini déterministe  $A$  *i.e.*  $L(A) = L$ . A partir de  $A$  on construit  $A'$  égal à  $A$  mais où les états terminaux et non terminaux sont inversés.  $A'$  accepte le complémentaire de  $A$ . Note : ceci n'est pas toujours vrai si  $A$  est un automate non déterministe (*ex* : l'automate à trois états  $q_0, q_1, q_2$  avec  $q_0 \xrightarrow{a} q_1$  et  $q_0 \xrightarrow{a} q_2$  où  $q_1$  est terminal).*

<sup>1</sup>Cf. remarque 1 pour les états initiaux multiples.

## 2.4 Propriétés des langages réguliers

On suppose l'alphabet  $\Sigma$  fixé.

**Lemme 1 (Pumping Lemma)** Soit  $A$  un automate fini à  $n$  états. Alors pour tout mot  $w \in L(A)$  tel que  $|w| \geq n$ , il existe  $x, u, y \in \Sigma^*$  tels que  $w = x.u.y$  et  $u \neq \varepsilon$ ,  $|x.u| \leq n$ , et pour tout  $i \in \mathbb{N}$ ,  $x.u^i.y \in L(A)$ .

**Preuve 4** Si  $w = a_1 a_2 \cdots a_l$  avec  $l \geq n$  est accepté par  $A$  alors il existe un chemin  $\sigma$  tel que :

$$\sigma = s_0 \xrightarrow{a_1} s_1 \xrightarrow{a_2} s_2 \cdots \xrightarrow{a_l} s_l \xrightarrow{\varepsilon^*} s'$$

avec

- $q \xrightarrow{a} q' \equiv q \xrightarrow{\varepsilon^*} q'' \xrightarrow{a} q'$ ,
- $s_0 = q_0$ ,  $s'$  terminal (on peut donc considérer  $s_l$  comme terminal sans changer le langage accepté).

Comme le nombre d'états de  $A$  est  $n$  il y a forcément parmi les états  $s_i$  deux états identiques i.e.  $s_k = s_j, 0 \leq k < j \leq l + 1$ . On peut donc écrire  $\sigma = s_0 \xrightarrow{x} s_k \xrightarrow{u} s_j \xrightarrow{y} s_l$ . Comme  $k < j$  on a  $|u| \geq 1$  et comme  $j \leq n$ ,  $|x.u| \leq n$ .  $s_l$  peut être considéré comme terminal et puisque  $s_k = s_j$  on a  $x.y \in L(A)$ . De plus on peut ajouter autant de boucles  $s_k \xrightarrow{u} s_j$  dans le chemin  $\sigma$  sans changer l'état initial et terminal.

En utilisant le *pumping lemma*, on peut montrer qu'il existe des langages non réguliers et donc compléter le théorème 1.

**Théorème 5** Le langage  $\{a^n b^n, n \in \mathbb{N}\}$  n'est pas régulier.

**Preuve 5** On suppose que ce langage  $L$  est régulier. Il existe donc un automate fini  $A$  à  $k$  états qui accepte  $L$ . Soit  $w = a^k b^k$  avec  $k \geq n$ . Il est clair que  $w \in L$ . D'après le *pumping lemma*, comme  $w \in L(A)$  et  $|w| \geq k$ ,  $w$  s'écrit  $w = x.u.y$  avec  $|x.u| \leq k$ . Ainsi  $u = a^i, i \geq 1$  et donc  $x.a^i.a^i.y$  serait dans  $L(A)$ . Or  $x.a^i.a^i.y$  n'est pas dans  $L$ .

## 3 Expressions régulières

### 3.1 Opérations régulières

**Définition 3 (Opérations régulières)** Soit  $A \subseteq \Sigma^*$  et  $B \subseteq \Sigma^*$ . Les opérations régulières sur les langages de  $\Sigma^*$  sont définies par :

**Union :**  $A \cup B = \{x \mid x \in A \text{ ou } x \in B\}$  ;

**Concaténation :**  $A \cdot B = \{x.y \mid x \in A \text{ et } y \in B\}$  ;

**Etoile :**  $A^* = \{x_1.x_2.\cdots.x_k \mid k \geq 0 \text{ et } \forall 0 \leq i \leq k, x_i \in A\}$ .

De manière équivalente  $A^* = \cup_{i \in \mathbb{N}} A^i$  avec  $A^0 = \{\varepsilon\}$  et  $A^{i+1} = A \cdot A^i$  pour  $i \geq 0$ .

### 3.2 Expressions régulières

**Définition 4 (Expression régulière)**  $R$  est une expression régulière (sur  $\Sigma$ ) si  $R$  est de la forme :

- $a$  pour  $a \in \Sigma$ , ou
- $\varepsilon$ , ou
- $\emptyset$ , ou
- $R_1|R_2$  avec  $R_1, R_2$  des expressions régulières, ou
- $R_1.R_2$  avec  $R_1, R_2$  des expressions régulières, ou
- $R^*$  avec  $R$  une expression régulière.

**Définition 5 (Langage associé à une expression régulière)** Le langage  $\mathcal{L}(R)$  associé à une expression régulière  $R$  est défini inductivement par :

- $\mathcal{L}(a) = \{a\}$  pour  $a \in \Sigma$ ,
- $\mathcal{L}(\varepsilon) = \{\varepsilon\}$ ,
- $\mathcal{L}(\emptyset) = \emptyset$ ,
- $\mathcal{L}(R_1|R_2) = \mathcal{L}(R_1) \cup \mathcal{L}(R_2)$ ,
- $\mathcal{L}(R_1.R_2) = \mathcal{L}(R_1) \cdot \mathcal{L}(R_2)$ ,
- $\mathcal{L}(R^*) = (\mathcal{L}(R))^*$ .

## 4 Automates finis et expressions régulières

**Théorème 6 (Kleene (1))** Soit  $R$  une expression régulière sur  $\Sigma$ . Alors  $\mathcal{L}(R)$  est un langage régulier.

**Preuve 6** Soit  $R$  une expression régulière. Pour prouver que  $L = \mathcal{L}(R)$  est régulier, on construit de manière inductive un automate fini (non déterministe) qui accepte  $\mathcal{L}(R)$ . On se restreint à construire des automates ayant un seul état initial et un seul état final, distinct de l'état initial (Cf. remarque 1).

- si  $R = \emptyset$ , on construit un automate à deux états,  $i$ , initial, et  $f$  final et aucune transition.
- si  $R = a, a \in \Sigma$  ou  $R = \varepsilon$ , on prend un automate à deux états  $i$ , initial et  $f$ , final; il y a une transition d'étiquete  $R$  de  $i$  à  $f$ .
- si  $R = R_1|R_2$ , on suppose avoir construit  $A(R_1)$  et  $A(R_2)$  qui acceptent respectivement  $\mathcal{L}(R_1)$  et  $\mathcal{L}(R_2)$  et qui ont un seul état final et initial. Soit  $A(R_1|R_2)$  l'automate de la Fig. 1, ayant un seul état initial  $i$  et un seul état final  $f$  tel que : de  $i$  on a une transition  $\varepsilon$  vers chaque état initial de  $A(R_1)$  et  $A(R_2)$ ; de chaque état final de  $A(R_1)$  et  $A(R_2)$  on a une transition  $\varepsilon$  vers  $f$ . Alors  $A(R_1|R_2)$  accepte  $\mathcal{L}(R_1|R_2)$ ;
- si  $R = R_1.R_2$  on construit l'automate  $A(R_1.R_2)$  de la Fig. 2 ( $A(R_1)$  et  $A(R_2)$  sont des automates qui acceptent  $\mathcal{L}(R_1)$  et  $\mathcal{L}(R_2)$ ). Si  $\varepsilon \in \mathcal{L}(R_1)$  on ajoute une transition  $\varepsilon$  de  $i$  à  $i_2$ ; si  $\varepsilon \in \mathcal{L}(R_2)$  on ajoute une transition  $\varepsilon$  de  $i_2$  à  $f$ .
- si  $R = R_1^*$  on construit l'automate  $A(R_1^*)$  à partir de  $A(R_1)$  comme indiqué sur la Fig. 3.

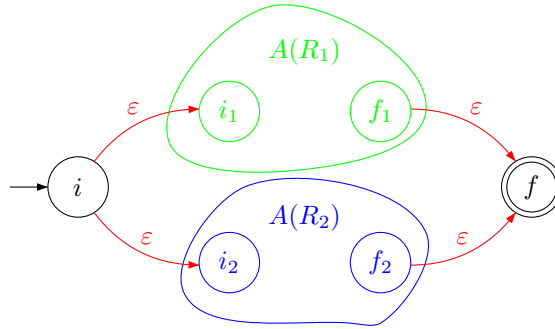


FIG. 1 – Automate  $A(R_1|R_2)$

**Théorème 7 (Kleene (2))** Si  $L$  est un langage régulier sur  $\Sigma$  alors il existe  $R$  expression régulière sur  $\Sigma$  telle que  $L = \mathcal{L}(R)$ .

**Preuve 7** Soit  $A = (Q, \Sigma, \delta, q_0, F)$  un automate fini déterministe<sup>2</sup> acceptant  $L$ . Il est facile de voir que

$$L = \bigcup_{f \in F} L_{q_0, f}$$

<sup>2</sup>Comme on prend un automate déterministe on ne fait plus l'hypothèse de l'unicité de l'état final.

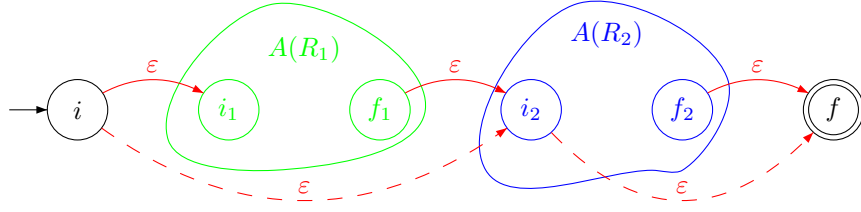


FIG. 2 – Automate  $A(R_1.R_2)$

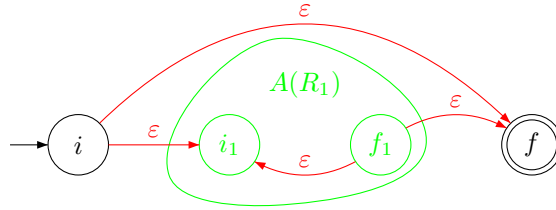


FIG. 3 – Automate  $A(R_1^*)$

où  $L_{q,q'}$  est la langage accepté par  $A(q, q') = (Q, \Sigma, \delta, q, \{q'\})$ . Il suffit donc de montrer que chaque  $L_{q,q'}$  est généré par une expression régulière car les expressions régulières sont closes par union. On suppose que les états de  $Q$  sont  $\{q_0, q_1, q_2, \dots, q_n\}$ . Soit  $L_{q,q'}^k, k \geq 0$  l'ensemble des mots que l'on peut produire en allant de  $q$  à  $q'$  par des chemins ne contenant que des états intermédiaires  $q_i, i < k$ . Formellement :

$$L_{q,q'}^k = \{w_1 w_2 \dots w_j \in \Sigma^*, 0 \leq j \mid q \xrightarrow{w_1} q_{i_1} \xrightarrow{w_2} q_{i_2} \dots q_{i_{j-1}} \xrightarrow{w_j} q' \text{ and } i_j < k\}$$

Pour  $k = 0$  on a  $L_{q,q'}^0 = \{a\}, a \in \Sigma$  ou  $L_{q,q'}^0 = \emptyset$  et donc  $L_{q,q'}^0$  est généré par une expression régulière sur  $\Sigma$ . De plus, pour  $k \geq 0$  on a :

$$L_{q,q'}^{k+1} = L_{q,q'}^k \cup (L_{q,q_{k+1}}^k \cdot (L_{q_{k+1},q_{k+1}}^k)^* \cdot L_{q_{k+1},q'}^k)$$

et donc  $L_{q,q'}^{k+1}$  est généré par une expression régulière si les  $L_{q,q'}^k$  le sont.  $L_{q,q'}^0$  est généré par une expression régulière et par induction on obtient donc que  $L_{q,q'}^k$  est généré par une expression régulière pour tout  $k \geq 0$ . Or  $L_{q,q'} = L_{q,q'}^{n+1}$  si  $q \neq q'$  et  $L_{q,q} = L_{q,q}^{n+1} \cup \{\varepsilon\}$  et ainsi chaque  $L_{q,q'}$  est généré par une expression régulière. Donc  $L$  est généré par une expression régulière.

## 5 Automates minimaux

**Définition 6 (Minimalité)** Un automate  $A = (Q, \Sigma, \delta, q_0, F)$  est minimal si pour tout  $q, q' \in Q$  il existe  $w \in L(q, A) \setminus L(q', A)$ .

Cette définition correspond à l'intuition suivante : il n'y a pas deux états distincts qui acceptent le même langage : il y a donc au moins un mot  $w$  qui est accepté à partir de  $q$  et pas à partir de  $q'$ .

**Théorème 8** Pour tout automate fini déterministe  $A$ , il existe un automate minimal  $B$  tel que  $L(A) = L(B)$ .

**Preuve 8** Soit  $A = (Q, \Sigma, \delta, q_0, F)$ . On définit la relation  $\equiv_i \subseteq Q \times Q$  par :

$$- q \equiv_0 q' \text{ ssi } q \in F \iff q' \in F,$$

- pour  $i \geq 0$ ,  $q \equiv_{i+1} q'$  ssi pour tout  $a \in \Sigma$  tel que  $q \xrightarrow{a} q_1$  on a  $q' \xrightarrow{a} q'_1$  et  $q_1 \equiv_i q'_1$ .

Chaque  $\equiv_i$  est une relation d'équivalence. Elle induit donc une partition de  $Q$  et on note  $Q_{/\equiv_i}$  le quotient de  $Q$  par  $\equiv_i$  (réciproquement toute partition induit une relation d'équivalence).

Une partition  $\mathcal{A} = A_1, A_2, \dots, A_n$  de  $Q$  est plus fine qu'une partition  $\mathcal{B} = B_1, B_2, \dots, B_k$  si 1)  $n \geq k$  et 2) pour tout  $j \in [1..n]$  il existe  $i \in [1..k]$  tel que  $A_j \subseteq B_i$ . On note dans ce cas  $\mathcal{A} \sqsubseteq \mathcal{B}$ .  $\sqsubseteq$  est un ordre partiel sur l'ensemble  $\mathcal{P}(Q)$  des partitions de  $Q$  et  $(\mathcal{P}(Q), \sqsubseteq)$  est un ensemble partiellement ordonné et  $\sqsubseteq$  est un ordre bien fondé. Comme  $Q$  est fini,  $\mathcal{P}(Q)$  est aussi fini.

On peut vérifier que si  $i \geq j$  alors  $Q_{\equiv_i} \sqsubseteq Q_{\equiv_j}$ . Les relations  $\equiv_i, i \geq 0$  forment donc une suite croissante. Le nombre de partitions de  $Q$  étant fini, il s'ensuit que  $\exists n \geq 0$  tel que  $\forall n' \geq n, \equiv_{n'} = \equiv_n$ . On note  $\equiv_*$  la relation  $\equiv_n$ . Pour  $q \in Q$  on note  $[q]$  la classe d'équivalence de  $q$  pour  $\equiv_*$ . On définit l'automate déterministe  $B = (S, \Sigma, \delta', q'_0, F')$  par :

- $S = Q_{/\equiv_*}$ ,
- $q'_0 = [q_0]$ ,
- $F' = [q]$  si il existe  $q \in F$ ,  $F' = \emptyset$  sinon,
- $\delta'([q], a) = [\delta(q, a)]$  pour tout  $a \in \Sigma$ .

On a  $L(B) = L(A)$ . Pour prouver cela on montre que :

$$q \equiv_i q' \iff L(q, A)_{\leq i} = L(q', A)_{\leq i} \quad (1)$$

avec  $L(q, A)_{\leq i} = \{w \mid |w| \leq i \text{ et } w \in L(q, A)\}$ . Pour  $i = 0$  l'équation (1) est vraie. On suppose l'équation (1) vraie pour  $i$  et on montre qu'elle est vraie pour  $i + 1$ . On note  $L(q, A)_{=i}$  les mots de longueur  $i$  de  $L(q, A)$ .

$\implies$  Soit  $w \in L(q, A)_{=i+1}$ . Alors on peut écrire  $w = a.w'$  avec  $|w'| = i$ . Comme  $q \equiv_{i+1} q'$  et que  $q \xrightarrow{a} q_1$  on a  $q' \xrightarrow{a} q'_1$  et  $q_1 \equiv_i q'_1$  (par définition de  $\equiv_{i+1}$ ). Par hypothèse d'induction, comme  $q_1 \equiv_i q'_1$  il s'ensuit que  $L(q_1, A)_{\leq i} = L(q'_1, A)_{\leq i}$ . Ainsi  $w' \in L(q'_1, A)_{\leq i}$  et donc  $w \in L(q, A)_{=i+1}$ , ce qui prouve que  $L(q, A)_{=i+1} \subseteq L(q', A)_{=i+1}$ . Par symétrie on a l'inclusion inverse et donc  $L(q, A)_{=i+1} = L(q', A)_{=i+1}$ .

$\impliedby$  Supposons que  $L(q, A)_{\leq i+1} = L(q', A)_{\leq i+1}$ . On suppose dans cette partie de la preuve que l'automate  $A$  est complet. Soit  $a$  tel que  $q \xrightarrow{a} q_1$ ; par complétude on a  $q' \xrightarrow{a} q'_1$ . De plus  $L(q, A)_{\leq i} = L(q', A)_{\leq i}$  (sinon on aurait  $L(q, A)_{\leq i+1} \neq L(q', A)_{\leq i+1}$ ). Par hypothèse d'induction, on obtient  $q_1 \equiv_i q'_1$  et donc  $q \equiv_{i+1} q'$ .

## Remerciements

Merci à Maud Kerebel (CPGE MP\* Option Informatique, lycée Louis Le Grand) pour ses remarques qui ont permis d'améliorer ce document.

## Références

- [AG92] André Arnold and Irène Guessarian. *Mathématiques pour l'informatique*. 1992.
- [ASU86] Alfred V. Aho, Ravi Sethi, and Jeffrey D. Ullman. *Compilers : Princiles, Techniques, and Tools*. Addison-Wesley, 1986. Existe en français.
- [Per90] Dominique Perrin. *Finite automata*, volume B : Formal Models and Semantics, chapter 1. Elsevier, 1990.
- [Sip96] Michael Sipser. *Introduction to the Theory of Computation*. International Thomson Publishing, 1996.
- [Wol01] Pierre Wolper. *Introduction à la calculabilité*. InterEditions, 2001. Deuxième édition.