**Automates finis [AF3] - Notes de cours 6**
**Résiduels - Automate minimal**
**Minimisation d'un automate fini**
*Jean-Marie Rifflet - Novembre 2005*

# 1. Introduction

## 1.1. Le problème

Étant donné un langage reconnaissable $L$, calculer un AFD $\mathcal{A}_L$ reconnaissant le langage $L$ et possédant le plus petit nombre d'états possible (existence et unicité).

## 1.2. Un problème connexe, la minimisation d'un automate

Étant donné un AFD $\mathcal{A}$, calculer l'automate $\mathcal{A}_L$ minimal reconnaissant le même langage que $\mathcal{A}$.

# 2. Automates et équivalences régulières à droite

$\hookrightarrow$ à un AFD quelconque $\mathcal{A} = < Q, X, \delta, q_0, Q' >$, on associe la relation binaire $\Theta_{\mathcal{A}}$ sur $X^*$ définie par

$$\underline{\forall u, v \in X^*,\ u\ \Theta_{\mathcal{A}}\ v \text{ si et seulement si } \delta^*(q_0, u) = \delta^*(q_0, v).}$$

La relation $\Theta_{\mathcal{A}}$ a les propriétés suivantes :

- c'est une **relation d'équivalence** : elle est réflexive, symétrique et transitive ;
- elle possède autant de classes que l'automate $\mathcal{A}$ a d'états. En particulier, si $\mathcal{A}$ est un AFD, elle possède un nombre fini de classes ;
- elle est **régulière à droite** : si $\underline{u\ \Theta_{\mathcal{A}}\ v \text{ alors } \forall w \in X^*, u.w\ \Theta_{\mathcal{A}}\ v.w}$ ;
- elle **sature** le langage $L(\mathcal{A})$ : le langage $L(\mathcal{A})$ est égal à l'union d'un certain nombre de classes de l'équivalence $\Theta_{\mathcal{A}}$.

  En d'autres termes, si $u\ \Theta_{\mathcal{A}}\ v$ et $u \in L(\mathcal{A})$ alors $v \in L(\mathcal{A})$.

$\hookrightarrow$ inversement, étant donné un langage $L$ sur l'alphabet $X$ et une relation d'équivalence régulière à droite $\Theta$ possédant un nombre fini de classes (c'est-à-dire dont l'ensemble quotient $X^*/\Theta$ est fini) et saturant le langage $L$, on peut lui associer un automate fini $\mathcal{A}_\Theta$ reconnaissant le langage $L$.

En désignant par $\overline{u}$ la classe d'un mot $u$, cet automate est défini par :

$$\mathcal{A}_\Theta = < X^*/\Theta, X, \delta, \overline{\epsilon}, \{\overline{u} \mid u \in L\} >, \text{ avec } \forall u \in X^*, x \in X,\ \delta(\overline{u}, x) = \overline{u.x}$$

# 3. Résiduel d'un langage $L$ par rapport à un mot $u$

## 3.1. Définition

Étant donné un langage $L$, on note $L/u = \{w \mid w \in X^* \text{ et } u.w \in L\}$ et on appelle $L/u$ résiduel de $L$ par (rapport à) $u$.

## 3.2. Propriétés des résiduels

($\alpha$) $\forall u \in X^*, \forall x \in X,\ L/(u.x) = (L/u)/x$

($\beta$) $\forall u \in X^*,\ (L_1 \cup L_2)/u = (L_1/u) \cup (L_2/u)$

($\gamma$) $\forall x \in X,\ (L_1.L_2)/x$

$$= (L_1/x).L_2 \text{ si } \epsilon \notin L_1 \text{ (règle } \gamma 1)$$
$$= (L_1/x).L_2 \cup L_2/x \text{ si } \epsilon \in L_1 \text{ (règle } \gamma 2)$$

($\delta$) $\forall x \in X,\ L^*/x = (L/x).L^*$

## 4. La relation $\Theta_L$ associée à un langage $L$

### 4.1 Définition de $\Theta_L$

À un langage $L$ quelconque, on associe la relation $\Theta_L$ définie par :

$u \ \Theta_L \ v$ si et seulement si $L/u = L/v$.

### 4.2 Propriétés de la relation $\Theta_L$

La relation $\Theta_L$ a les propriétés suivantes :

- $\Theta_L$ est une relation d'équivalence ;
- $\Theta_L$ est régulière à droite ;
- $\Theta_L$ sature $L$ ;
- $\Theta_L$ est l'équivalence régulière à droite **la plus grossière** saturant $L$ : pour toute relation d'équivalence régulière à droite $\Theta$ saturant L, $u\Theta v \Rightarrow u\Theta_L v$.

  Toute classe de $\Theta_L$ s'exprime donc comme union de classes de $\Theta$.

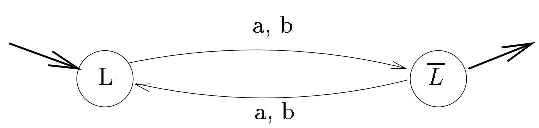
### 4.3 Exemple

Considérons par exemple le langage $L$ sur l'alphabet $\{a, b\}$ contenant exactement tous les mots dans lesquels les nombres d'occurences de $a$ et de $b$ sont de parité différente (ainsi *abaaaba* et *babba* appartiennent à $L$ mais pas *ababba* ou *aabb* par exemple).

La relation $\Theta_L$ a deux classes :

- le langage $L$ lui même : il contient tous les mots (tels que $a$ ou $b$) auxquels on peut ajouter un mot ayant des nombres d'occurrences de $a$ et de $b$ de même parité (c'est-à-dire des mots du complément $\overline{L}$ de $L$) ;
- le complément $\overline{L}$ du langage $L$ : il contient tous les mots (tels que $\epsilon$ ou $ab$) auxquels on peut ajouter un mot ayant des nombres d'occurrences de $a$ et de $b$ de parité différente (c'est-à-dire des mots de $L$).

Cette équivalence sature le langage $L$ ($L$ est exactement l'une des deux classes de $\Theta_L$). L'automate déduit de $\Theta_L$ est le suivant :

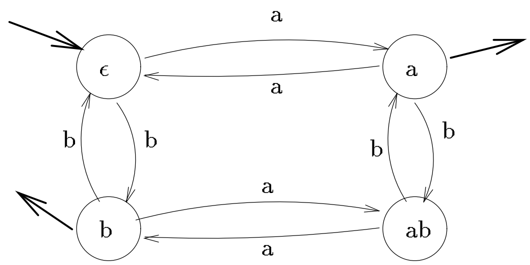



Considèrons la relation $\Theta_1$ dont les classes d'équivalence correspondent aux parités des nombres d'occurrences des lettres $a$ et $b$. Elle a 4 classes :

- la classe $\overline{\epsilon}$ du mot vide $\epsilon$ (contenant tous les mots contenant un nombre pair d'occurrences de $a$ et un nombre pair d'occurrences de $b$) ;
- la classe $\overline{a}$ du mot $a$ (contenant tous les mots contenant un nombre impair d'occurrences de $a$ et un nombre pair d'occurrences de $b$) ;
- la classe $\overline{b}$ du mot $b$ (contenant tous les mots contenant un nombre pair d'occurrences de $a$ et un nombre impair d'occurrences de $b$) ;
- la classe $\overline{ab}$ du mot $b$ (contenant tous les mots contenant un nombre impair d'occurrences de $a$ et un nombre impair d'occurrences de $b$).

Cette équivalence régulière à droite sature $L$. En effet $L = \overline{a} \cup \overline{b}$.

Il lui correspond l'automate à 4 états suivants :




L'équivalence $\Theta_L$ est plus grossière que $\Theta_1$ et il n'en existe évidemment pas de plus grossière saturant $L$.

## 5. Le cas des langages reconnaissables

### 5.1. Automate minimal d'un langage reconnaissable

Si un langage $L$ est reconnaissable, il existe un automate fini $\mathcal{A}$ le reconnaissant. Cet automate possède $n$ états et la relation $\Theta_{\mathcal{A}}$ a un nombre fini de classes (il est égal à $n$).

Cette relation est plus fine que la relation $\Theta_L$ et cette dernière relation a donc un nombre de classes $m$ fini ($m \leq n$).

L'AFD $\mathcal{A}_{\Theta_L}$, déduit de $\Theta_L$ par le procédé décrit précédemment, possède $m$ états et est parmi tous les AFD qui reconnaissent $L$ celui qui possède le moins d'états.

Il est appelé **automate minimal de** $L$.

On peut ainsi caractériser les langages reconnaissables (ou rationnels) comme les langages dont la relation $\Theta_L$ associée possède un nombre fini de classes.

### 5.2. Calcul de l'automate minimal d'un langage reconnaissable

#### 5.2.1. Principe de calcul

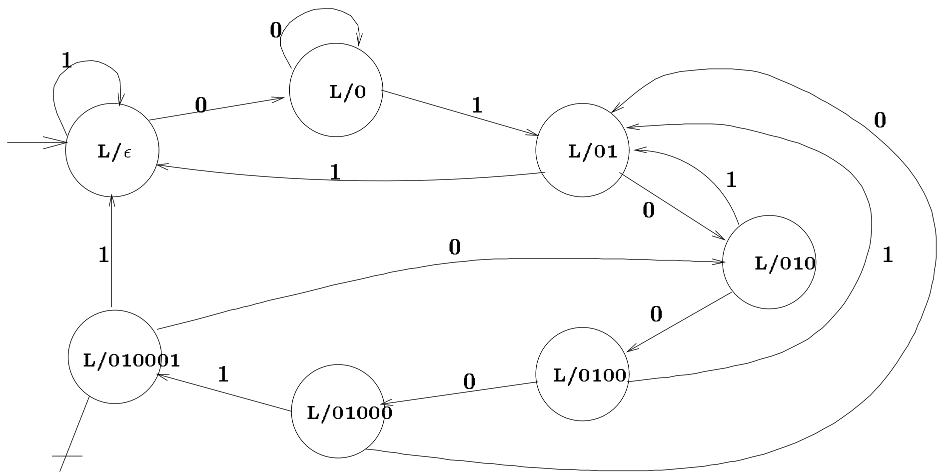
Étant donné un langage $L$, par application des règles de calcul sur les résiduels, on peut calculer un arbre de résiduels $L/w_i$ dont la racine est $L/\epsilon = L$ (le résiduel de $L$ par rapport au mot vide étant évidemment $L$ lui-même). Le résiduel $L/w$ ayant été calculé, on calculera les

Le calcul des résiduels donne (comme d'habitude, on confond langages et expressions afin de faciliter l'écriture):

- $L/\epsilon = L$
- $L/0 = [(0+1)^*/0)]010001 + 010001/0$ (règle $\gamma 2$)
  $= (0+1)^*010001 + 10001$ **(car** $(0+1)^*/0 = (0+1)^*$**)**
  $= L + 10001$ **(nouveau à cause de** $10001$**)**
- $L/1 = [(0+1)^*/1)]010001 + 010001/1$ (régle $\gamma 2$)
  $= (0+1)^*010001 + \emptyset$ **(car** $(0+1)^*/1 = (0+1)^*$**)**
  $= L = L/\epsilon$ **(déjà rencontré)**
- $L/00 = (L/0)/0$ **(règle** $\alpha$**)**
  $= (L + 10001)/0$
  $= L/0 + 10001/0$ **(règle** $\beta$**)**
  $= L/0 + \emptyset$
  $= L/0$ **(déjà rencontré)**
- $L/01 = (L/0)/1$ **(règle** $\alpha$**)**
  $= (L + 10001)/1$
  $= L/1 + 10001/1$ **(règle** $\beta$**)**
  $= L/1 + 0001 = L + 0001$ **(nouveau à cause de** $0001$**)**
- $L/010 = (L/01)/0$ **(règle** $\alpha$**)**
  $= (L/1 + 0001)/0 = (L + 0001)/0$
  $= L/0 + 0001/0$ **(règle** $\beta$**)**
  $= L/0 + 001$ **(nouveau à cause de** $001$**)**
- $L/011 = (L/01)/1$ **(règle** $\alpha$**)**
  $= (L/1 + 0001)/1 = (L + 0001)/1$
  $= L/1 + 0001/1$ **(règle** $\beta$**)**
  $= L/1 + \emptyset = L/1 = L$ **(déjà rencontré)**
- $L/0100 = (L/010)/0$ **(règle** $\alpha$**)**
  $= (L/0 + 001)/0$
  $= (L/0)/0 + 001/0$ **(règle** $\beta$**)**
  $= L/00 + 01 = L/0 + 01$ **(nouveau à cause de** $01$**)**
- $L/0101 = (L/010)/1$ **(règle** $\alpha$**)**
  $= (L/0 + 001)/1$
  $= (L/0)/1 + 001/1$ **(règle** $\beta$**)**
  $= L/01 + \emptyset = L/01$ **(déjà rencontré)**
- $L/01000 = (L/0100)/0$ **(règle** $\alpha$**)**
  $= (L/0 + 01)/0$
  $= (L/0)/0 + 01/0$ **(règle** $\beta$**)**
  $= L/00 + 1 = L/0 + 1$ **(nouveau à cause de** $1$**)**
- $L/01001 = (L/0100)/1$ **(règle** $\alpha$**)**
  $= (L/0 + 01)/1$
  $= (L/0)/1 + 01/1$ **(règle** $\beta$**)**
  $= L/01 + \emptyset = L/01$ **(déjà rencontré)**
- $L/010000 = (L/01000)/0$ **(règle** $\alpha$**)**
  $= (L/0 + 1)/0$
  $= (L/0)/0 + 1/0$ **(règle** $\beta$**)**
  $= L/00 + \emptyset = L/00 = L/0$ **(déjà rencontré)**

- $L/010001 = (L/01000)/1$ **(règle $\alpha$)**
  $= (L/0 + 1)/1$
  $= (L/0)/1 + 1/1$ **(règle $\beta$)**
  $= L/01 + \epsilon$ **(nouveau à cause de $\epsilon$)**
- $L/0100010 = (L/010001)/0$ **(règle $\alpha$)**
  $= (L/01 + \epsilon)/0$
  $= (L/01)/0 + \epsilon/0$ **(règle $\beta$)**
  $= L/010 + \emptyset = L/010$ **(dèjà rencontré)**
- $L/0100011 = (L/010001)/1$ **(règle $\alpha$)**
  $= (L/01 + \epsilon)/1$
  $= (L/01)/1 + \epsilon/1$ **(règle $\beta$)**
  $= L/011 + \emptyset = L/1 = L$ **(dèjà rencontré)**

On en déduit l'automate (minimal reconnaissant le langage $L$) correspondant au graphe suivant dont les états sont les différents résiduels, l'état initial est $L/\epsilon$ et les états terminaux sont ceux correspondant à des résiduels contenant $\epsilon$, c'est-à-dire ici $L/010001$ :




<u>Minimisation d'un automate</u>

$\hookrightarrow$ étant donné un automate fini $\mathcal{A} = < Q, X, \delta, q_0, Q' >$, supposé complet et accessible, on définit **l'équivalence de Nérode** $\beta$ sur ses états par :

$$\underline{\forall q, q' \in Q,\ q \equiv q'\ [\beta] \text{ si et seulement si } \forall w \in X^*,\ q.w \in Q' \Leftrightarrow q'.w \in Q'}$$

Si on note $\overline{q}$ la classe d'équivalence de l'état $q$, on définit l'automate fini quotient

$$\mathcal{A}_\beta =< Q/\beta,\ X,\ \overline{\delta},\ \overline{q_0},\ Q'/\delta > \text{ avec } \forall q \in Q, x \in X,\ \overline{\delta}(\overline{q}, x) = \delta(q, x).$$

Cet automate est obtenu en fusionnant en un seul et même état des états équivalents et est, à un renommage des états près, l'automate minimal du langage $L_{\mathcal{A}}$.

$\hookrightarrow$ **Calcul de l'équivalence $\beta$ - Algorithme de Moore**

On calcule la suite $\beta_0, \beta_1, \ldots, \beta_i, \beta_{i+1}, \ldots$ d'équivalences de la manière suivante :

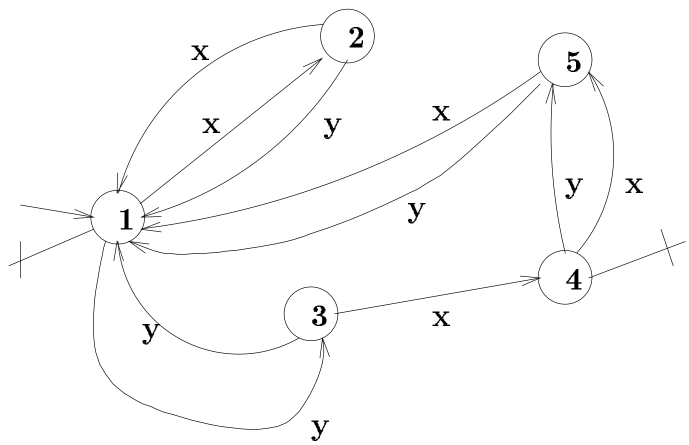
- $q \equiv q'\ [\beta_0]$ si et seulement si $q \in Q' \Leftrightarrow q' \in Q'$
- $\forall i > 0,\ q \equiv q'\ [\beta_{i+1}]$ si et seulement si
  - $q \equiv q'\ [\beta_i]$
  - $\forall x \in X,\ \delta(q, x) \equiv \delta(q', x)\ [\beta_i]$

*Propriété fondamentale*

$$\underline{\exists n, \text{ tel que } \beta_n = \beta_{n+1} = \beta_{n+2} = \ldots} \text{ et donc } \beta_n = \beta$$

$\hookrightarrow$ **Exemples**

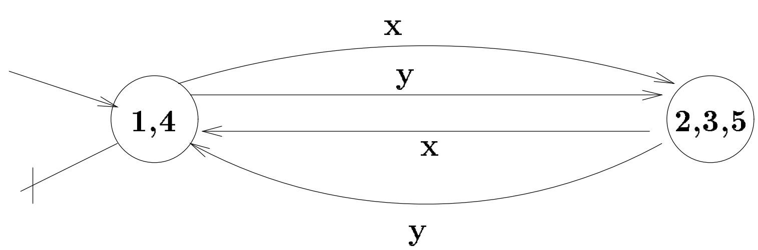
1) Considérons l'automate sur l'alphabet $\{x, y\}$ correspondant au graphe suivant :



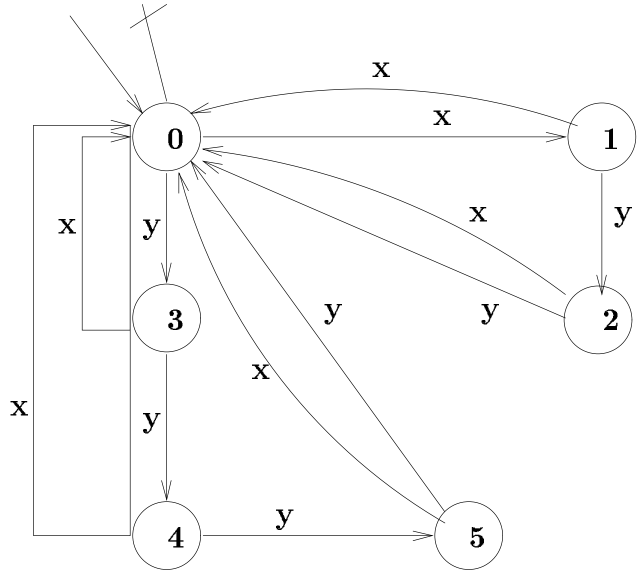

On obtient (les états terminaux sont soulignés) :

| $\epsilon$ | $x$ | $y$ |
|---|---|---|
| $\underline{1}$ | 2 | 3 |
| 2 | $\underline{1}$ | $\underline{1}$ |
| 3 | $\underline{4}$ | $\underline{1}$ |
| $\underline{4}$ | 5 | 5 |
| 5 | $\underline{1}$ | $\underline{1}$ |

La classe $\beta_0$ a deux classes ($\{1, 4\}$ et $\{2, 3, 5\}$) et la classe $\beta_1$ possède exactement les mêmes. On en déduit l'automate minimal en fusionnant en un seul état d'une part 1 et 4 et d'autre part 2, 3 et 5, ce qui donne le graphe suivant :



2) Considérons l'automate sur l'alphabet $\{x, y\}$ correspondant au graphe suivant :




La procédure de minimisation donne (les états terminaux sont soulignés) :

| $\epsilon$ | $x$ | $y$ | $xx$ | $xy$ | $yx$ | $yy$ | $xxx$ | $xxy$ | $xyx$ | $xyy$ | $yxx$ | $yxy$ | $yyx$ | $yyy$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\underline{0}$ | 1 | 3 | $\underline{0}$ | 2 | $\underline{0}$ | 4 | 1 | 3 | $\underline{0}$ | $\underline{0}$ | 1 | 3 | $\underline{0}$ | 5 |
| 1 | $\underline{0}$ | 2 | 1 | 3 | $\underline{0}$ | $\underline{0}$ | $\underline{0}$ | 2 | $\underline{0}$ | 4 | 1 | 3 | 1 | 3 |
| 2 | $\underline{0}$ | $\underline{0}$ | 1 | 3 | 1 | 3 | $\underline{0}$ | 2 | $\underline{0}$ | 4 | $\underline{0}$ | 2 | $\underline{0}$ | 4 |
| 3 | $\underline{0}$ | 4 | 1 | 3 | $\underline{0}$ | 5 | $\underline{0}$ | 2 | $\underline{0}$ | 4 | 1 | 3 | $\underline{0}$ | $\underline{0}$ |
| 4 | $\underline{0}$ | 5 | 1 | 3 | $\underline{0}$ | $\underline{0}$ | $\underline{0}$ | 2 | $\underline{0}$ | 4 | 1 | 3 | 1 | 3 |
| 5 | $\underline{0}$ | $\underline{0}$ | 1 | 3 | 1 | 3 | $\underline{0}$ | 2 | $\underline{0}$ | 4 | $\underline{0}$ | 2 | $\underline{0}$ | 4 |

- L'équivalence $\beta_0$ a deux classes : $\{0\}$ et $\{1, 2, 3, 4, 5\}$
- L'équivalence $\beta_1$ a trois classes : $\{0\}$, $\{1, 3, 4\}$ et $\{2, 5\}$
- L'équivalence $\beta_2$ a quatre classes : $\{0\}$, $\{1, 4\}$, $\{2, 5\}$ et $\{3\}$
- L'équivalence $\beta_3$ a quatre classes comme $\beta_2$.

On a donc atteint l'équivalence $\beta$ et on peut donc confondre d'une part les états 1 et 4 et d'autre part les états 2 et 5. L'automate minimal équivalent à l'automate de départ a 4 états et correspond au graphe suivant (après avoir renommé 1 la classe $\{1, 4\}$ et 2 la classe $\{2, 5\}$) :