

1 Reconnaissance d'un langage rationnel par un automate fini

1.1 Automates réduits

Un automate fini sur l'alphabet X (ensemble fini) peut être vu comme un triplet (I, F, δ) , où I est l'ensemble des états initiaux, F l'ensemble des états finaux et δ l'ensemble des transitions, les transitions étant des triplets (q, x, q') où q et q' sont des états de l'automate, en nombre fini, et x un élément de X .

Les automates envisagés ici sous le nom d'automates réduits se distinguent par les propriétés suivantes :

- L'ensemble Q des états est une partie finie de \mathbb{N} contenant 0
- $F = \{0\}$
- $\forall q \in Q \setminus \{0\} \text{ card } \{x \in X : \exists q' \in Q (q, x, q') \in \delta\} = 1$

Un automate réduit (I, F, δ) sur l'alphabet X sera représenté dans la suite par un quadruplet $A = (E, I, f, g)$ où

- E est une partie de \mathbb{N}^* (l'ensemble des états autres que 0) et I une partie de l'ensemble $Q = E \cup \{0\}$ des états
- f est une application de E dans l'ensemble $\mathcal{P}(Q)$ des parties de $Q = E \cup \{0\}$
- g est une application de E dans X

$g(e)$ est l'unique élément x de X pour lequel il existe un élément q de Q tel que (e, x, q) appartienne à δ . $f(e)$ est l'ensemble des éléments q de Q tels que $(e, g(e), q)$ appartienne à δ .

Le mot vide ϵ est reconnu par (E, F, f, g) si et seulement si 0 appartient à I .

Le mot $x_1 x_2 \dots x_n$ de longueur $n \geq 1$ est reconnu si et seulement si

$$\exists (e_1, e_2, \dots, e_n) \in E^n \quad (e_1 \in I \text{ et } \forall i \in \mathbb{N}_{n-1} e_{i+1} \in f(e_i) \text{ et } \forall i \in \mathbb{N}_n g(e_i) = x_i \text{ et } 0 \in f(e_n))$$

On note $L(A)$ le langage reconnu par A .

1.2 Somme de deux automates réduits

Définition : Soient $A_i = (E_i, F_i, f_i, g_i)$ ($i = 1, 2$) deux automates réduits tels que $E_1 \cap E_2 = \emptyset$.

On pose $A_1 + A_2 = (E, F, f, g)$ avec $E = E_1 \cup E_2$, $F = F_1 \cup F_2$, $f|_{E_i} = f_i$, $g|_{E_i} = g_i$

Théorème : L'automate $A_1 + A_2$ reconnaît le langage $L(A_1) \cup L(A_2)$

1.3 Concaténation de deux automates réduits

Définition : Soient $A_i = (E_i, F_i, f_i, g_i)$ ($i = 1, 2$) deux automates réduits tels que $E_1 \cap E_2 = \emptyset$.

On pose $A_1 \cdot A_2 = (E, F, f, g)$ avec

- $E = E_1 \cup E_2$
- $\begin{cases} I = I_1 \text{ si } 0 \notin I_1 \\ I = (I_1 \setminus \{0\}) \cup I_2 \text{ sinon} \end{cases}$
- $\begin{cases} f(e) = f_1(e) \text{ pour } e \in E_1 \text{ tel que } 0 \notin f_1(e) \\ f(e) = (f_1(e) \setminus \{0\}) \cup I_2 \text{ pour } e \in E_1 \text{ tel que } 0 \in f_1(e) \\ f(e) = f_2(e) \text{ pour } e \in E_2 \end{cases}$
- $g|_{E_i} = g_i$

Théorème : L'automate $A_1 \cdot A_2$ reconnaît le langage $L(A_1) \cdot L(A_2)$.

1.4 Etoile d'un automate réduit

Définition : Soit $A = (E, F, f, g)$ un automate réduit. On pose $A^* = (E^*, F^*, f^*, g^*)$ avec

- $E^* = E$
- $I^* = I \cup \{0\}$
- $\begin{cases} f^*(e) = I \cup f(e) \text{ si } 0 \in f(e) \\ f^*(e) = f(e) \text{ sinon} \end{cases}$
- $g^* = g$

Théorème : L'automate A^* reconnaît le langage $L(A)^*$.

1.5 Conclusion

Comme le langage vide est reconnu par l'automate réduit $(\emptyset, \emptyset, \emptyset, \emptyset)$, le langage singleton $\{\epsilon\}$ par l'automate réduit $(\emptyset, \{0\}, \emptyset, \emptyset)$ et le langage singleton $\{x\}$, où x appartient à X , par l'automate réduit $(\{1\}, \{1\}, f, g)$ avec $f(1) = \{0\}$ et $g(1) = x$, tout langage rationnel est reconnu par un automate réduit facile à construire à partir de l'arbre d'une expression rationnelle du langage.

2 Automates finis déterministes

2.1 Déterminisation d'un automate fini

Un Automate Fini Déterministe Complet (AFDC) sur l'alphabet X est un quintuplet $A = (X, Q, d, F, \delta)$:

- X est l'alphabet (ensemble fini)
- Q est l'ensemble des états (ensemble fini)
- d est l'état de départ ou état initial
- F est l'ensemble des états finals (ou finaux, ou terminaux)
- δ est la fonction de transition : δ est une *application* de $Q \times X$ dans Q vérifiant $\delta(q, xy) = \delta(\delta(q, x), y)$ pour tout (q, x, y) appartenant à $Q \times X \times X$.

Le langage $L(A)$ reconnu par A est défini par $w \in L(A) \iff \delta^*(d, w) \in F$ où δ^* est l'extension de δ à $Q \times X^*$ telle que $\forall (q, w, w') \in Q \times X^* \times X^* \quad \delta^*(q, ww') = \delta^*(\delta^*(q, w), w')$.

A étant un automate fini quelconque sur l'alphabet X , représenté par le triplet (D, F, δ) , Q l'ensemble de ses états, notons, pour toute partie P de Q et tout $x \in X$, $\Delta(P, x)$ la partie de Q définie par

$$(q' \in \Delta(P, x)) \iff (\exists q \in P \quad (q, x, q') \in \delta)$$

On définit ainsi une application Δ de $\mathcal{P}(Q) \times X$ dans $\mathcal{P}(Q)$, où $\mathcal{P}(Q)$ est l'ensemble des parties de Q .

Soit $d' = \{d\}$

Soit Q' le plus petit ensemble de parties de Q tel que

- $d' \in Q'$
- $\forall P \in Q' \forall x \in X \quad \Delta(P, x) \in Q'$

Soit F' l'ensemble des éléments de Q' qui rencontrent F . Soit δ' l'application de $Q' \times X$ dans Q' induite par Δ .

Définition : L'AFDC (X, Q', d', F', δ') est appelé déterminisé de A . On le notera $\text{Dét}(A)$.

Théorème : Un automate fini quelconque A et son déterminisé $\text{Dét}(A)$ sont équivalents : $L(\text{Dét}(A)) = L(A)$.

Le déterminisé est un automate accessible. Il existe sur un tel automate un *ordre naturel* sur l'ensemble des états si l'alphabet X est ordonné : soit, pour tout état q d'un automate accessible $A = (X, Q, d, F, \delta)$, $f(q) = \min\{w \in X^* : \delta^*(d, w) = q\}$, le minimum correspondant à l'ordre hiérarchique de X^* . f est une injection de Q dans X^* puisque $\forall q \in Q \quad \delta^*(d, f(q)) = q$. On peut donc convenir que q précède q' lorsque $f(q)$ précède $f(q')$.

2.2 AFDC minimaux

Définition 1 : Deux AFDC (X, Q, d, F, δ) et (X, Q', d', F', δ') sur le même alphabet X sont dits isomorphes lorsqu'il existe une bijection ϕ de Q sur Q' permettant de les identifier, c'est-à-dire telle que

$$\forall (q, x) \in Q \times X \quad \delta(\phi(q), x) = \phi(\delta(q, x))$$

Deux AFDC isomorphes sont en particulier équivalents.

Définition 2 : Un AFDC est dit minimal lorsque, parmi les AFDC qui lui sont équivalents, il a le nombre minimum d'états.

Théorème 1 : Parmi les AFDC équivalents à un automate fini A , l'automate des résiduels de $L(A)$ est minimal et isomorphe à tous ceux qui sont minimaux.

On peut donc parler de l'AFDC minimal (ou automate minimal) équivalent à A .

Calcul de l'automate minimal équivalent à $A = (X, Q, d, F, \delta)$:

R étant une relation d'équivalence quelconque sur Q , notons $\phi(R)$ la relation d'équivalence sur Q définie par

$$(q \phi(R) q') \iff (q R q' \wedge \forall x \in X \quad \delta(q, x) R \delta(q', x))$$

La suite de relations d'équivalence de plus en plus fines $(\phi^n(R))_{(n \in \mathbb{N})}$ est stationnaire car l'ensemble Q est fini. La relation limite R_∞ est telle que $q R_\infty q' \implies \forall x \in X \quad \delta(q, x) R_\infty \delta(q', x)$ ce qui permet de définir une application δ_{R_∞} de $Q/R_\infty \times X$ dans Q/R_∞ par $\delta_{R_\infty}(s(q), x) = s(\delta(q, x))$ où s est la surjection canonique de Q sur Q/R_∞ .

Théorème 2 : l'automate minimal équivalent à A est l'AFDC $(X, Q/R_\infty, s(d), s(F), \delta_{R_\infty})$ où R est la relation d'équivalence associée à la partition $\{Q \setminus F, F\}$ de Q .

2.3 Test de l'équivalence de deux expressions rationnelles

On construit, pour chaque expression rationnelle, un automate simple associé qu'on détermine et minimise en utilisant l'ordre de l'alphabet et l'ordre naturel associé des états. Ceux-ci sont alors identifiés par leurs numéros à partir de 0 (0 est l'état de départ). Les deux expressions rationnelles sont équivalentes (i.e. représentent le même langage) si et seulement si elles ont de cette façon le même automate minimal associé.

3 Rationalité du langage reconnu par un automate fini

Algorithme de McNaughton et Yamada :

Soit un automate fini $A = (D, F, \delta)$ sur l'alphabet X . Soit Q l'ensemble de ses états. Notons, pour tout $(i, j) \in Q \times Q$ et toute partie P de Q , $L(i, j, P)$ le langage formé des étiquettes des chemins de A d'origine i et d'extrémité j dont tous les états intermédiaires appartiennent à P .

On a

- $L(A) = \bigcup \{L(i, j, Q) : (i, j) \in D \times F\},$
- $L(i, i, P) = (L(i, i, P \setminus \{i\}))^* \text{ si } i \in P,$
- $L(i, j, P) = (L(i, i, P \setminus \{i\}))^* L(i, j, P \setminus \{i, j\}) L(j, j, P \setminus \{i, j\})^* \text{ si } \{i, j\} \cap P \neq \emptyset \text{ et } i \neq j,$
- $L(i, j, \emptyset) = \{x \in X : (i, x, j) \in \delta\}$
- $L(i, j, P) = L(i, j, P \setminus \{k\}) \cup L(i, k, P \setminus \{k\}) L(k, k, P \setminus \{k\})^* L(k, j, P \setminus \{k\}) \text{ si } k \in P \text{ et } \{i, j\} \cap P = \emptyset$

ce qui permet un calcul récursif d'une expression rationnelle du langage reconnu par A .

D'une autre façon, si les n états de A sont numérotés de 0 à $n - 1$ et identifiés à leurs numéros, posons $L_0(i, j) = L(i, j, \emptyset)$ et, pour $1 \leq k \leq n$, $L_k(i, j) = L(i, j, \{0, 1, \dots, k - 1\})$.

On a alors, pour $0 \leq k \leq n - 1$ et $(i, j) \in \{0, 1, \dots, n - 1\}^2$, les relations suivantes :

- $L_{k+1}(i, j) = L_k(i, j) \cup L_k(i, k)(L_k(k, k))^* L_k(k, j) \text{ pour } i \neq k \text{ et } j \neq k$
- $L_{k+1}(i, k) = L_k(i, k)(L_k(k, k))^* \text{ pour } i \neq k$
- $L_{k+1}(k, j) = (L_k(k, k))^* L_k(k, j) \text{ pour } k \neq j$
- $L_{k+1}(k, k) = (L_k(k, k))^*$

ce qui permet, à partir de L_0 , de calculer les matrices L_1, L_2, \dots, L_n .

En effet :
$$\begin{cases} L_0(i, j) = \{x \in X : (i, x, j) \in \delta\} & \text{si } i \neq j \\ L_0(i, i) = \{\epsilon\} \cup \{x \in X : (i, x, i) \in \delta\} \end{cases}$$

D'où une expression rationnelle de $L(A)$ puisque $L(A) = \bigcup_{(i, j) \in D \times F} L_n(i, j).$

Automate minimal

Notations :

Tous les automates considérés dans ce problème sont des Automates Finis Déterministes Complets (en abrégé AFDC).

Un AFDC se présente comme un quintuplet $\mathcal{A} = (X, Q, i, F, \delta)$, où

- X est un alphabet (ensemble fini).
- Q est l'ensemble des états de l'automate (ensemble fini).
- i est l'état initial de l'automate ($i \in Q$).
- F est l'ensemble des états terminaux de l'automate ($F \subseteq Q$)
- δ est la fonction de transition de l'automate : δ est une application de $Q \times X$ dans Q .

On note δ^* la complétée de δ : δ^* est une application de $Q \times X^*$ dans Q , où X^* est l'ensemble des mots de longueur finie sur l'alphabet X .

Pour simplifier l'écriture, et ceci quel que soit l'AFDC \mathcal{A} considéré, la fonction de transition complétée sera notée

$$(q, x) \rightarrow q \cdot x \quad (Q \times X^* \rightarrow Q)$$

On rappelle que (propriétés de la fonction de transition complétée) :

- $\forall (q, x, y) \in Q \times X^* \times X^* \quad (q \cdot x) \cdot y = q \cdot (xy)$
- $\forall q \in Q \quad q \cdot \varepsilon = q$, où ε est le mot vide.

et que le langage reconnu par l'AFDC $\mathcal{A} = (X, Q, i, F, \delta)$ est la partie L de X^* définie par

$$x \in L \iff i \cdot x \in F$$

L étant un langage sur l'alphabet X et x un élément de X^* , on appelle *résiduel* de L en x le langage $L \cdot x$ (noté aussi $x^{-1}L$) défini par

$$w \in L \cdot x \iff wx \in L$$

On note $\text{Res } L$ l'ensemble des résiduels du langage L : $\text{Res } L = \{L \cdot x : x \in X^*\}$.

Dans tout le problème L est un langage rationnel sur l'alphabet X et $\mathcal{A} = (X, Q, i, F, \delta)$ un AFDC reconnaissant L .

On définit, pour tout $q \in Q$, le langage $\varphi(q)$ ($\varphi(q) \subseteq X^*$) par

$$w \in \varphi(q) \iff q \cdot w \in F$$

En particulier $L = \varphi(i)$.

I - Résiduels d'un langage, automate minimal

1. Montrer brièvement que $\forall (x, y) \in X^* \times X^* \quad L \cdot (xy) = (L \cdot x) \cdot y$ et que $L \cdot \varepsilon = L$

En déduire que $\forall R \in \text{Res } L \quad \forall x \in X^* \quad R \cdot x \in \text{Res } L$

2. Comparer, pour tout élément x de X^* , $\varphi(i \cdot x)$ à $L \cdot x$
3. Montrer que $\text{Res } L \subseteq \varphi(Q)$
4. Comparer le nombre d'états de \mathcal{A} au nombre de résiduels de L

On définit alors l'automate des résiduels du langage L , noté \mathcal{A}_L par

$$\mathcal{A}_L = (X, \text{Res } L, L, F_L, \delta_L)$$

où F_L est la partie de $\text{Res } L$ définie par

$$R \in F_L \iff \varepsilon \in R$$

et où δ_L est l'application $(R, x) \rightarrow R \cdot x$ de $\text{Res } L \times X$ dans $\text{Res } L$. La complétée δ_L^* est, d'après 1., telle que

$$\forall (R, x) \in \text{Res } L \times X^* \quad \delta_L^*(R, x) = R \cdot x$$

(comme souhaité...)

5. Montrer que \mathcal{A}_L reconnaît L .
6. Montrer que $\forall (q, x) \in Q \times X^* \quad \varphi(q) \cdot x = \varphi(q \cdot x)$
7. Comparer alors entre eux les AFDC minimaux (i.e. ayant le nombre minimum d'états) reconnaissant L .

II - Calcul de l'automate minimal : méthode des partitions

Dans cette partie l'AFDC $\mathcal{A} = (X, Q, i, F, \delta)$ est supposé *accessible* (i.e. tel que $\forall q \in Q \exists x \in X^* q = i \cdot x$). On note E l'ensemble des relations d'équivalence sur Q et, pour toute $\mathcal{R} \in E$, Q/\mathcal{R} l'ensemble des classes d'équivalence pour la relation \mathcal{R} . On définit l'application u de E dans E par

$$q u(\mathcal{R}) q' \iff (q \mathcal{R} q' \text{ et } \forall a \in X \quad q \cdot a \mathcal{R} q' \cdot a)$$

Pour \mathcal{R}_1 et \mathcal{R}_2 éléments de E , on écrit $\mathcal{R}_2 >> \mathcal{R}_1$ lorsque

$$\forall (q, q') \in Q^2 \quad q \mathcal{R}_2 q' \implies q \mathcal{R}_1 q'$$

On observera que $\mathcal{R}_2 >> \mathcal{R}_1 \implies u(\mathcal{R}_2) >> u(\mathcal{R}_1)$ et que $\forall \mathcal{R} \in E \quad u(\mathcal{R}) >> \mathcal{R}$ (on ne demande pas de le justifier)

1. Montrer que, pour toute $\mathcal{R}_0 \in E$, la suite de relations d'équivalence $(\mathcal{R}_n)_{n \geq 0}$ définie par $\mathcal{R}_n = u^n(\mathcal{R}_0)$ (autrement dit par $\forall n \in \mathbb{N} \quad \mathcal{R}_{n+1} = u(\mathcal{R}_n)$) est *stationnaire* (considérer les partitions \mathcal{P}_n de Q associées aux \mathcal{R}_n).

On suppose dans toute la suite que \mathcal{R}_0 est la relation d'équivalence associée à la partition $\{F, Q \setminus F\}$ de Q (i.e. qu'on a $q \mathcal{R}_0 q' \iff ((q \in F \text{ et } q' \in F) \text{ ou } (q \notin F \text{ et } q' \notin F))$) et on note \mathcal{R}_∞ la relation d'équivalence définie par $\mathcal{R}_\infty = \mathcal{R}_k$ où $k \in \mathbb{N}$ est tel que pour tout $n \geq k \quad \mathcal{R}_n = \mathcal{R}_k$. Pour tout $q \in Q$ on note $c(q)$ la classe d'équivalence de q pour la relation \mathcal{R}_∞ (i.e. c désigne la surjection canonique de Q sur $Q/\mathcal{R}_\infty = c(Q)$).

On définit la «relation d'équivalence de Nérde» \mathcal{N} ($\mathcal{N} \in E$) par

$$q \mathcal{N} q' \iff \varphi(q) = \varphi(q')$$

2. Comparer $\text{card}(Q/\mathcal{N})$ et $\text{card}(\text{Res } L)$ (observer que \mathcal{A} est accessible).
3. Montrer que $\mathcal{N} >> \mathcal{R}_0$
4. Montrer que $u(\mathcal{N}) = \mathcal{N}$ (i.e. que $q \mathcal{N} q' \implies \forall a \in X \quad q \cdot a \mathcal{N} q' \cdot a$)
5. Montrer que $\mathcal{N} >> \mathcal{R}_\infty$.
6. Montrer que $q \mathcal{R}_\infty q' \implies \forall a \in X \quad q \cdot a \mathcal{R}_\infty q' \cdot a \implies \forall x \in X^* \quad q \cdot x \mathcal{R}_\infty q' \cdot x$

On pose, pour tout $(q, x) \in Q \times X^*$, $c(q) \cdot x = c(q \cdot x)$ (définition légitime d'après la question précédente), ce qui permet de considérer l'AFDC

$$c(\mathcal{A}) = (X, c(Q), c(i), c(F), \delta_c)$$

tel que $\forall (q, x) \in Q \times X^* \quad \delta_c^*(c(q), x) = c(q \cdot x)$

7. Montrer que $c(\mathcal{A})$ reconnaît L .
8. Montrer que, parmi les AFDC reconnaissant L , $c(\mathcal{A})$ est minimal.

III - Exemple

Appliquer ce qui précède à l'AFDC $\mathcal{A} = (\{a, b\}, \{1, 2, 3, 4, 5, 6, 7\}, 1, \{3, 4, 5\}, \delta)$ défini par la table de transitions suivante :

		a	b
\rightarrow	1	2	4
	2	3	7
\leftarrow	3	7	2
\leftarrow	4	5	7
\leftarrow	5	7	6
	6	5	7
	7	7	7

Déterminer les partitions associées aux relations d'équivalence \mathcal{R}_0 , \mathcal{R}_1 et \mathcal{R}_2 . En déduire l'automate $c(\mathcal{A})$: on donnera sa table de transitions avec l'état initial et les états terminaux comme ci-dessus.

Réponses

I - Résiduels d'un langage, automate minimal

1. $w \in L \cdot (xy) \Leftrightarrow xyw \in L \Leftrightarrow yw \in L \cdot x \Leftrightarrow w \in (L \cdot x) \cdot y$
 $w \in L \cdot \varepsilon \Leftrightarrow \varepsilon w \in L \Leftrightarrow w \in L$
 Soit $R \in \text{Res } L$: écrivons $R = L \cdot y$ avec $y \in X^*$. Alors $R \cdot x = (L \cdot y) \cdot x = L \cdot (yx)$. Donc $R \cdot x \in \text{Res } L$
2. $w \in \varphi(i \cdot x) \Leftrightarrow (i \cdot x) \cdot w \in F \Leftrightarrow i \cdot (xw) \in F \Leftrightarrow xw \in L \Leftrightarrow w \in L \cdot x$. Donc $\varphi(i \cdot x) = L \cdot x$
3. $\forall x \in X^* \quad L \cdot x = \varphi(i \cdot x) \in \varphi(Q)$. Donc $\text{Res } L \subseteq \varphi(Q)$
4. $\text{card Res } L \leq \text{card } \varphi(Q) \leq \text{card } Q$
5. Soit L' le langage reconnu par \mathcal{A}_L . $w \in L' \Leftrightarrow L \cdot w \in F_L \Leftrightarrow \varepsilon \in L \cdot w \Leftrightarrow w\varepsilon \in L \Leftrightarrow w \in L$. Donc $L' = L$
6. $w \in \varphi(q) \cdot x \Leftrightarrow xw \in \varphi(q) \Leftrightarrow q \cdot (xw) \in F \Leftrightarrow (q \cdot x) \cdot w \in F \Leftrightarrow w \in \varphi(q \cdot x)$
7. D'après 4. et 5., \mathcal{A}_L est un AFDC minimal reconnaissant L . Soit maintenant $\mathcal{A} = (X, Q, i, F, \delta)$ un AFDC minimal quelconque reconnaissant L . Alors $\text{card } Q = \text{card Res } L$ et l'application $\varphi : Q \rightarrow \mathcal{P}(X^*)$ est une bijection de Q sur $\text{Res } L$, puisqu'a priori $\text{card Res } L \leq \text{card } \varphi(Q) \leq \text{card } Q$. Et, comme $\forall (q, x) \in Q \times X^* \quad \varphi(q) \cdot x = \varphi(q \cdot x)$, \mathcal{A} s'identifie par φ à \mathcal{A}_L (Noter aussi que $\varphi(i) = L$ et que $\varphi(F) = F_L$). A la numérotation près des états, il n'existe donc qu'un AFDC minimal reconnaissant L .

II - Calcul de l'automate minimal : méthode des partitions

1. Si \mathcal{P}_n est la partition de Q associée à \mathcal{R}_n , la suite d'entiers $(\text{card } \mathcal{P}_n)_{n \in \mathbb{N}}$ est croissante car, pour tout $n \in \mathbb{N}$, \mathcal{P}_{n+1} est plus fine que \mathcal{P}_n , du fait que $\mathcal{R}_{n+1} \gg \mathcal{R}_n$. Or, pour tout $n \in \mathbb{N}$, $\text{card } \mathcal{P}_n \leq \text{card } Q$. Donc il existe k tel que $\forall n \geq k \quad \text{card } \mathcal{P}_n = \text{card } \mathcal{P}_k$, i.e. $\mathcal{P}_n = \mathcal{P}_k$, i.e. $\mathcal{R}_n = \mathcal{R}_k$
2. L'application $\varphi : Q \rightarrow \mathcal{P}(X^*)$ induit une bijection de Q/\mathcal{N} sur $\varphi(Q)$, car $q \mathcal{N} q' \Leftrightarrow \varphi(q) = \varphi(q')$. Donc $\text{card } (Q/\mathcal{N}) = \text{card } \varphi(Q) = \text{card Res } L$: en effet, $\varphi(Q) = \text{Res } L$ car l'automate est accessible (cf. I 2. et 3.).
3. $q \mathcal{N} q' \Leftrightarrow \varphi(q) = \varphi(q') \Leftrightarrow \forall w \in X^* (q \cdot w \in F \Leftrightarrow q' \cdot w \in F)$.
 Donc $q \mathcal{N} q' \Rightarrow (q \in F \Leftrightarrow q' \in F) \Leftrightarrow q \mathcal{R}_0 q'$ (en prenant $w = \varepsilon$)
4. $q \mathcal{N} q' \Leftrightarrow \varphi(q) = \varphi(q') \Rightarrow \forall a \in X \quad \varphi(q \cdot a) = \varphi(q) \cdot a = \varphi(q') \cdot a = \varphi(q' \cdot a) \Leftrightarrow \forall a \in X \quad q \cdot a \mathcal{N} q' \cdot a$ (en utilisant I.6.)
5. Soit $k \in \mathbb{N}$ tel que $\mathcal{R}_\infty = u^k(\mathcal{R}_0)$. $\mathcal{N} \gg \mathcal{R}_0$ (d'après 3.). Donc $\mathcal{N} = u^k(\mathcal{N}) \gg u^k(\mathcal{R}_0) = \mathcal{R}_\infty$ (d'après 4. et l'observation faite en préambule de II).
6. Soit $k \in \mathbb{N}$ tel que $\mathcal{R}_\infty = u^k(\mathcal{R}_0)$. $\mathcal{R}_\infty = u^k(\mathcal{R}_0) = u^{k+1}(\mathcal{R}_0) = u(\mathcal{R}_\infty)$.
 Donc $q \mathcal{R}_\infty q' \Rightarrow \forall a \in X \quad q \cdot a \mathcal{R}_\infty q' \cdot a \Rightarrow \forall x \in X^* \quad q \cdot x \mathcal{R}_\infty q' \cdot x$ (la dernière implication par récurrence sur la longueur de x).
7. Soit L' le langage reconnu par $c(\mathcal{A})$.
 $w \in L' \Leftrightarrow c(i \cdot w) = c(i) \cdot w \in c(F) \Leftrightarrow \exists q \in F \quad c(i \cdot w) = c(q) \Leftrightarrow \exists q \in F \quad i \cdot w \mathcal{R}_\infty q$
 $\Rightarrow \exists q \in F \quad i \cdot w \mathcal{R}_0 q \Rightarrow i \cdot w \in F \Rightarrow w \in L$ (ceci car $\mathcal{R}_\infty \gg \mathcal{R}_0$ et d'après la définition de \mathcal{R}_0).
 Réciproquement, $w \in L \Leftrightarrow i \cdot w \in F \Rightarrow c(i) \cdot w = c(i \cdot w) \in c(F) \Leftrightarrow w \in L'$
 Donc $L = L'$
8. $c(\mathcal{A})$ reconnaît L (question précédente).
 $\text{card } c(Q) \leq \text{card } (Q/\mathcal{N})$ d'après 5. car $c(Q) = Q/\mathcal{R}_\infty$.
 Or $\text{card } (Q/\mathcal{N}) = \text{card Res } L$ d'après 2.
 Donc $c(\mathcal{A})$ est, d'après I, un automate minimal reconnaissant L (et $\text{card } c(Q) = \text{card Res } L$)

III - Exemple

$\begin{cases} \mathcal{P}_0 = \{\{3, 4, 5\}, \{1, 2, 6, 7\}\} \\ \mathcal{P}_1 = \{\{3, 5\}, \{4\}, \{1\}, \{2, 6\}, \{7\}\} \\ \mathcal{P}_2 = \mathcal{P}_1 \end{cases}$	D'où l'automate minimal :	\rightarrow	$\{1\}$	$\{2,6\}$	$\{4\}$
			$\{2,6\}$	$\{3,5\}$	$\{7\}$
		\leftarrow	$\{3,5\}$	$\{7\}$	$\{2,6\}$
		\leftarrow	$\{4\}$	$\{3,5\}$	$\{7\}$
			$\{7\}$	$\{7\}$	$\{7\}$