TELECOM Nancy 1A — Mathématiques Appliquées pour l'Informatique

Théorie des langages : minimisation des automates

Introduction

L'objectif de ce TD est de montrer qu'à tout langage régulier on peut associer un automate déterministe unique (à un isomorphisme près) avec un nombre minimal d'états, qu'on appelle *automate minimal* du langage et qui reconnaît ce langage.

Dans un premier temps nous présentons les idées clés de la minimisation, puis nous donnons un algorithme qui permet de construire l'automate minimal.

1 Première étape : enlever les états inaccessibles

Soit un automate déterministe $\mathcal{A} = (A,\ Q,\ q_0,\ \delta,\ T)$. Si un état p est inaccessible, il n'existe pas de mot α appartenant à A^* tel que $\delta^*(q_0, \alpha) = p$. On peut donc retirer cet état de l'automate sans que le langage reconnu ne change (rappel : le langage reconnu par $\mathcal{A}$ est l'ensemble de mots β tel que $\delta^*(q_0, \beta) \in T$).

Question 1

On considère l'automate $\mathcal{A}_1 = (A,\ Q,\ 0,\ \delta,\ T)$ où $A = \{a,\ b\}$, $Q = \{0,\ 1,\ 2,\ 3,\ 4,\ 5,\ 6,\ 7\}$, $T = \{0,\ 3,\ 4,\ 6,\ 7\}$ et δ est donnée par la table suivante :

δ	0	1	2	3	4	5	6	7
a	2	2	3	2	5	6	5	5
b	1	1	2	6	4	5	7	3

Éliminer les états inaccessibles de $\mathcal{A}_1$.

Indication : mettre en évidence un algorithme calculant les états accessibles en partant de l'état initial de l'automate.

Dans la suite, on va supposer que $\mathcal{A}$ n'a pas d'états inaccessibles.

2 Deuxième étape : identification des états équivalents

Pour réduire le nombre d'états d'un automate, on cherche à reconnaître quels états de l'automate sont superflus. Pour cela, on cherche des états qui sont équivalents à d'autres états, c'est-à-dire qui peuvent être fusionnés sans changer le langage accepté par l'automate.

Pour tout état $q \in Q$, on considère L_q, le langage associé à l'état q, ou plus intuitivement l'ensemble des mots qui $\ll$ envoient q dans un état final $\gg$:

$$L_q = \{\alpha,\ \alpha \in A^* \ et\ \delta^*(q,\ \alpha) \in T\}$$

On dit que deux états p et q sont équivalents, noté $p \sim q$, ssi $L_p = L_q$. Cette relation s'appelle *Équivalence de Nerode*. Donc, si $p \sim q$, pour tout mot α on a $\delta^*(p,\ \alpha) \in T$ ssi $\delta^*(q,\ \alpha) \in T$.

Autrement dit, si $p \sim q$, tout mot α est accepté à partir de p ssi il est accepté à partir de q. Donc, intuitivement, p et q sont équivalents, et on peut les fusionner sans changer le langage accepté par l'automate.

Question 2
Donner L_{q_0}, L_{q_1}, L_{q_2} et L_{q_3}. En déduire les états équivalents de l'automate $\mathcal{A}_2$ donné en figure 1.

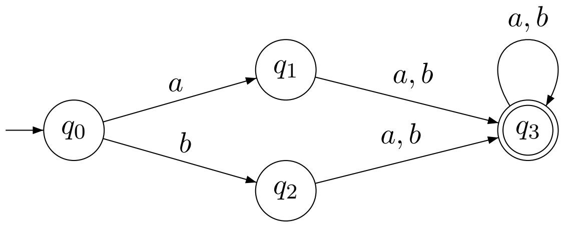

FIGURE 1 – Automate $\mathcal{A}_2$

La relation d'équivalence $\sim$ partitionne l'ensemble des états Q dans des classes d'équivalence, c'est-à-dire des sous ensembles de Q tels que tous les états dans chaque classe sont équivalents par rapport à $\sim$.

Ainsi, l'automate minimal est caractérisé par son alphabet A, son ensemble d'états c'est-à-dire l'ensemble des classes d'équivalence par rapport à $\sim$, son état initial, la classe d'équivalence de q_0 et l'ensemble des états terminaux qui est l'ensemble des classes d'équivalence qui contiennent des éléments de T. Les transitions découlent directement de l'automate $\mathcal{A}$ en prenant des représentants dans les classes d'équivalence.

Définition 2.1 (Automate minimal) *Étant donné un automate déterministe* $\mathcal{A} = (A,\ Q,\ q_0,\ \delta,\ T)$, *on définit un nouvel automate déterministe* $\mathcal{A}/\!\sim\, = (A,\ Q/\!\sim,\ [q_0]_\sim,\ \delta_\sim,\ T/\!\sim)$ *tel que*
- $Q/\!\sim$ *est l'ensemble quotient de* Q *par la relation d'équivalence* $\sim$*, c'est-à-dire l'ensemble des classes d'équivalence par la relation d'équivalence* $\sim$
- $[q_0]_\sim$ *est la classe d'équivalence de* q_0
- $\delta_\sim$ *est la fonction de transition définie par* $\delta_\sim([p]_\sim,\ a) = [q]_\sim \ \Leftrightarrow\ \delta(p,\ a) = q$
- $T/\!\sim$ *est l'ensemble quotient de* T *par* $\sim$

Question 3
Construire l'automate minimal pour $\mathcal{A}_2$.

3 Algorithme de minimisation

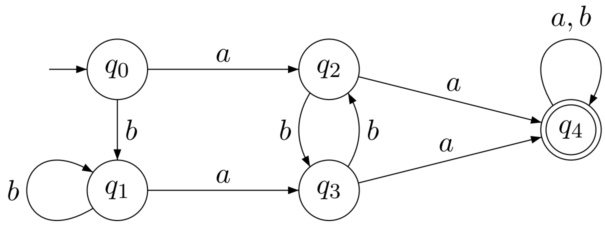

FIGURE 2 – Automate $\mathcal{A}_3$

L'identification des états équivalents peut devenir plus compliqué pour des automates de plus grande taille. L'algorithme suivant permet de les identifier de façon systématique, en utilisant un tableau qui contient des cases pour toutes les paires d'états. Par exemple, la figure 3 contient des cases pour toutes les paires d'états de l'automate $\mathcal{A}_3$ donné en figure 2. Dans ce tableau une case remplie signifiera que les deux états concernés ne sont pas équivalents.

L'idée initiale de l'algorithme est la recherche de contre-exemples : si deux états p et q ne sont pas équivalents, alors il existe un mot α tel que $\delta^*(p, \alpha) \in T$ et $\delta^*(q, \alpha) \notin T$, ou inversement $\delta^*(p, \alpha) \notin T$ et $\delta^*(q, \alpha) \in T$. Premièrement si on prend $\alpha = \epsilon$, cela implique que tout état terminal ne peut être équivalent à un état non terminal. Ceci sera utilisé dans l'initialisation de l'algorithme.

	q_0	q_1	q_2	q_3
q_1				
q_2				
q_3				
q_4				

FIGURE 3 – Table de minimisation de $\mathcal{A}_3$

De plus, cette observation permet de construire les contre-exemples itérativement. Soient p et q des états. S'il existe une lettre $a \in A$ telle que les transitions sur a à partir de p et q amènent dans des états p' et q' qui ne sont pas équivalents, alors p et q ne le sont pas non plus :

$$\exists a \in A : \delta(p, a) \not\sim \delta(q, a) \Rightarrow p \not\sim q$$

Cela est utilisé dans l'itération de l'algorithme.

L'algorithme procède de la manière suivante :

— Initialisation : toutes les cases (p, q) telles que $p \in T, q \notin T$ ou $p \notin T, q \in T$ sont marquées avec un cercle : $\bigcirc$

— Itération : on parcourt toutes les cases (p, q) actuellement vides. Si il existe un caractère a tel que la case $(\delta(p, a), \delta(q, a))$ ou $(\delta(q, a), \delta(p, a))$ est cochée, on coche la case (p, q) avec une croix : $\times$

— Condition d'arrêt : Si au cours de la dernière itération aucune case n'est marquée ou si toutes les cases sont marquées, on s'arrête.

Comme Q est fini, il n'y a qu'un nombre fini de cases et l'algorithme va terminer. A l'arrêt de l'algorithme, les cases vides correspondent à des paires d'états équivalents.

Exemple :
Considérons l'automate $\mathcal{A}_3$ décrit en figure 2. q_4 est le seul état terminal, après l'initialisation on obtient donc le tableau suivant :

	q_0	q_1	q_2	q_3
q_1				
q_2				
q_3				
q_4	$\bigcirc$	$\bigcirc$	$\bigcirc$	$\bigcirc$

Dans la première itération, on remplit les quatre cases suivantes

	q_0	q_1	q_2	q_3
q_1				
q_2	$\times$	$\times$		
q_3	$\times$	$\times$		
q_4	$\bigcirc$	$\bigcirc$	$\bigcirc$	$\bigcirc$

car :

— Pour la case (q_1, q_0) on considère les deux lettres a et b :
  — Pour a on a $\delta(q_1, a) = q_3$, $\delta(q_0, a) = q_2$ mais la case (q_2, q_3) est vide.
  — Pour b on a $\delta(q_1, b) = q_1 = \delta(q_0, b)$.
  Donc ni pour a ni pour b on tombe dans une case remplie, et la case (q_1, q_0) reste vide.

— Pour la case (q_2, q_0) on a $\delta(q_2, a) = q_4$, $\delta(q_0, a) = q_2$ et la case (q_4, q_2) est cochée ; on coche donc la case (q_2, q_0).

— Pour la case (q_2, q_1) on a $\delta(q_2, a) = q_4$, $\delta(q_1, a) = q_3$ et la case (q_4, q_3) est cochée ; on coche donc la case (q_2, q_1).

— etc.

Dans la deuxième itération aucune case supplémentaire n'est remplie, on obtient donc que $q_1 \sim q_0$ et $q_3 \sim q_2$.

4 Exercices

Question 4
Donner la table des transitions de l'automate minimal pour $\mathcal{A}_3$ ainsi que ses états initiaux et terminaux, et dessiner son diagramme sagittal.

Question 5
On considère l'automate de la Question 1.

Déterminer l'automate minimal de $\mathcal{A}_1$ en appliquant l'algorithme décrit ci-dessus (Enlever d'abord les états inaccessibles identifiés dans la question 1 !). Donner la table des transitions de l'automate minimal et dessiner son diagramme sagittal.

Question 6
On considère l'automate $\mathcal{A}_4 = (A,\ Q,\ 0,\ \delta,\ T)$ tel que $A = \{a,\ b\}$, $Q = \{0,\ 1,\ 2,\ 3,\ 4,\ 5,\ 6,\ 7,\ 8,\ 9,\ 10,\ 11,\ 12,\ 13,\ 14\}$, $T = \{1,\ 2,\ 6,\ 7,\ 9,\ 10,\ 12\}$ et δ est donnée en figure 4.

δ	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
a	3	2	8	4	13	0	5	11	0	6	7	3	2	4	3
b	6	13	6	7	2	10	7	2	1	4	13	12	0	7	10

Figure 4 – Table de transition de $\mathcal{A}_4$

1. Éliminer les états inaccessibles de $\mathcal{A}_4$.
2. Déterminer l'automate minimal de $\mathcal{A}_4$. Donner la table des transitions de l'automate minimal et dessiner son diagramme sagittal.

Question 7
On considère l'automate $\mathcal{A}_5 = (A,\ Q,\ 0,\ \delta,\ T)$ où $A = \{a,\ b,\ c\}$,

$Q = \{0,\ 1,\ 2,\ 3,\ 4,\ 5,\ 6,\ 7,\ 8,\ 9\}$, $T = \{5,\ 6,\ 7,\ 8,\ 9\}$ et δ est donnée par la table suivante :

δ	0	1	2	3	4	5	6	7	8	9
a	1	1	4	4	1	6	6	9	6	6
b	2	3	2	2	3	7	8	7	7	8
c	0	0	0	5	5	5	5	5	5	5

Minimiser $\mathcal{A}_5$ et donner tous les éléments de l'automate minimal obtenu et sa fonction de transition sous forme d'une table.