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), qu'on appelle *automate minimal* du langage et qui reconnaît ce langage.

Nous présentons dans un premier temps l'algorithme de minimisation qui construit l'automate minimal, puis donnons une justification de cet algorithme en considérant la notion de quotient des langages par les mots.

1 Algorithme de minimisation

Soit un automate déterministe $\mathcal{A} = (A,\ Q,\ q_0,\ \delta,\ T)$ dont tous les états sont accessibles, l'algorithme consiste à calculer une suite de partitions emboîtées de Q de la façon suivante :

— Initialisation : la partition initiale $\mathcal{P}_0 = \{Q \setminus T,\ T\}$ comporte les deux ensembles $Q \setminus T$ et T.

— Itération : on définit $\mathcal{P}_{i+1}$ à partir de $\mathcal{P}_i$, par la relation suivante :

$$p \sim_{i+1} q \quad \Leftrightarrow \quad p \sim_i q \ \ et \ \ (\forall a \in A)\ \delta(p,a) \sim_i \delta(q,a)$$

jusqu'à $\mathcal{P}_i = \mathcal{P}_{i+1}$.

Comme Q est fini, il n'y a qu'un nombre fini de partitions et comme les partitions sont emboîtées la condition d'arrêt de l'algorithme est vérifiée.

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

Question 1

On considère l'automate $\mathcal{A} = (A,\ Q,\ 0,\ \delta,\ T)$ 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 figure 1.

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 1 – Table de transition de $\mathcal{A}$

1. Eliminer les états inaccessibles de $\mathcal{A}$. Indication : mettre en évidence un algorithme calculant les états accessibles en partant de l'état initial de l'automate.
2. Déterminer l'automate minimal de $\mathcal{A}$ en appliquant l'algorithme décrit ci-dessus. Donner la table des transitions de l'automate minimal et dessiner son diagramme sagittal.

2 L'automate des quotients d'un langage

Dans toute la suite nous considérons des automates déterministes n'ayant pas d'états inaccessibles, A désigne un alphabet, $\mathcal{P}(A^*)$ est l'ensemble des langages définis sur A, α est un mot de A^* et X un langage c'est-à-dire un élément de $\mathcal{P}(A^*)$

Définition 2.1 (Quotient gauche d'un langage par un mot) *On pose* $\alpha^{-1}X = \{\beta,\ \beta \in A^* \ et\ \alpha\beta \in X\}$, *$\alpha^{-1}X$ est l'ensemble des mots qui complétent à droite α en des mots de X. $\alpha^{-1}X$ est le* **quotient gauche** *de X par α. $\alpha^{-1}X$ est aussi parfois appelé* **l'ensemble des contextes à droite** *du mot α.*

Question 2
Montrer que $(\forall \alpha \in A^*)$ $(\forall X \subseteq A^*)$ $(\varepsilon \in \alpha^{-1}X \Leftrightarrow \alpha \in X)$
Question 3
Montrer que $(\forall \alpha \in A^*)$ $(\forall \beta \in A^*)$ $(\forall X \subseteq A^*)$ $(\alpha\beta)^{-1}X = \beta^{-1}(\alpha^{-1}X)$

Définition 2.2 (Ensemble des quotients) *Soit L un langage de A^*, $Q_L = \{\alpha^{-1}L,\ \alpha \in A^*\}$ s'appelle* **l'ensemble des quotients** *de L.*

Q_L est un ensemble de langages (i.e. un ensemble d'ensembles de mots).

Question 4
Montrer que pour tout langage L, $L \in Q_L$.
Question 5
Application : soit l'alphabet $A = \{a,\ b\}$ et $L = a^*ba^*$, déterminer Q_L l'ensemble des quotients de L.

Définition 2.3 (Automate des quotients) *Soit L un langage de A^* tel que Q_L soit un ensemble fini, l'automate $\mathcal{A}_L = (A,\ Q_L,\ L,\ \delta_L,\ T_L)$ tel que*
— $\delta_L(\alpha^{-1}L,\ a) = (\alpha a)^{-1}L$
— $T_L = \{\alpha^{-1}L,\ \varepsilon \in \alpha^{-1}L\} \ = \ \{\alpha^{-1}L,\ \alpha \in L\}$
est appelé **automate des quotients** *de L. Cet automate $\mathcal{A}_L$ est aussi appelé* **l'automate syntaxique** *de L.*

Question 6
Montrer que dans la définition , la fonction δ est bien définie c'est-à-dire ne dépend pas des mots α.
Question 7
Montrer $L(\mathcal{A}_L) = L$, c'est-à-dire que le langage reconnu par $\mathcal{A}_L$ est L. Indication : utiliser la remarque 2.1 suivante.

Remarque 2.1 *Etant donné un automate déterministe $\mathcal{A} = (A,\ Q,\ q_0,\ \delta,\ T)$, on rappelle que $\delta : Q \times A \to Q$, peut se prolonger en une application $\delta^* : Q \times A^* \to Q$, que l'on note aussi δ dans la suite de l'exposé.*
De plus on a les propriétés suivantes :
— $(\forall q \in Q)$ $(\forall \alpha \in A^*)$ $(\forall \beta \in A^*)$ $\delta(q, \alpha\beta) \ = \ \delta(\delta(q, \alpha), \beta)$
— $\alpha \in L(\mathcal{A}) \ \Leftrightarrow \ \delta(q_0, \alpha) \in T$

Question 8
Application : déterminer l'automate $\mathcal{A}_L$ correspondant au langage $L = a^*ba^*$.

Lemme 2.1 *Soient $\mathcal{A} = (A,\ Q,\ q_0,\ \delta,\ T)$ un automate déterministe complet*[1]
(et accessible) et $L = L(\mathcal{A})$ le langage qu'il reconnaît. Pour tout $q \in Q$, on note L_q l'ensemble des mots qui "envoient q dans un état final" : $L_q = \{\alpha,\ \alpha \in A^*$ et $\delta(q,\ \alpha) \in T\}$.
On a les propriétés suivantes :
1. $(\forall \alpha \in A^*)$ $(\forall p \in Q)$ $(\forall q \in Q)$ $\delta(p, \alpha) = q \ \Rightarrow \ L_q = \alpha^{-1}L_p$.
2. $(\forall \alpha \in A^*)$ $(\forall q \in Q)$ $\delta(q_0, \alpha) = q \ \Rightarrow \ L_q = \alpha^{-1}L$

1. Un automate (A, Q, I, E, T) est complet si pour tout p dans Q et tout a dans A, il existe un q dans Q tel que (p, a, q) est une transition de E.
Il est simple de vérifier qu'un automate est complet. Il est facile de transformer un automate non complet en un automate complet, il suffit (i) d'ajouter un nouvel état q_N et les transitions (p, a, q_N) pour lesquelles il n'existe pas de transition de la forme (p, a, q) (ii) d'ajouter les transitions (q_N, a, q_N) pour tout a de A.

Question 9
Démontrer le lemme 2.1.

L'étape suivante consiste à montrer que l'automate des quotients d'un langage est l'automate minimal, le théorème suivant établit ce résultat.

Théorème 2.1 *Soient L un langage et $\mathcal{A} = (A,\ Q,\ q_0,\ \delta,\ T)$ un automate déterministe reconnaissant le langage L. Soit $\mathcal{A}_L = (A,\ Q_L,\ L,\ \delta_L,\ T_L)$ l'automate des quotients de L. Il existe une application $\varphi : Q \to Q_L$ telle que :*

1. $\varphi(q_0) = L$
2. $(\forall q \in Q)\ (\forall a \in A)\ \varphi(\delta(q,\ a)) = \delta_L(\varphi(q), a)$
3. $(\forall q \in T)\ \varphi(q) \in T_L$

et φ est surjective si $\mathcal{A}$ est complet.

L'automate $\mathcal{A}_L$ des quotients de L est appelé **l'automate minimal** *de L.*

Question 10
Soit $q \in Q$ et soit $\alpha \in A^*$ tel que $\delta(q_0, \alpha) = q$ (un tel α existe car par hypothèse tous les états de l'ensemble Q sont accessibles de l'état initial q_0), on pose alors $\varphi(q) = \alpha^{-1}L$.

Démontrer le théorème 2.1 en répondant aux questions suivantes.

1. Montrer que la définition de φ ne dépend pas du mot α, c'est-à-dire que si $\delta(q_0, \alpha) = \delta(q_0, \beta)$ alors $\alpha^{-1}L = \beta^{-1}L$.
2. Montrer les trois propriétés de φ décrites dans le théorème 2.1.
3. Montrer que φ est surjective.

Corollaire 2.1 *Un langage L est régulier si et seulement si l'ensemble Q_L des quotients est fini.*

Démonstration. Si L est régulier alors il existe un automate fini $\mathcal{A}$ (Q étant l'ensemble de ses états) reconnaissant L, d'après la proposition comme φ est une surjection de Q vers Q_L, Q_L est fini.

Réciproquement si Q_L est fini, $\mathcal{A}_L$ est un automate fini qui reconnaît L donc L est régulier.

Ce corollaire est un moyen de caractériser les langages réguliers.

Question 11
Soit le langage $L = \{a^i b^i,\ i \in \mathbb{N}\}$, montrer que Q_L est infini et en déduire que L n'est pas un langage régulier.

3 Calcul de l'automate minimal

Nous allons maintenant donner un moyen effectif de calculer l'automate minimal d'un langage régulier à partir d'un automate quelconque reconnaissant ce langage, c'est ce qu'on appelle la minimisation.

Définition 3.1 (Equivalence de Nerode) *Soit $\mathcal{A} = (A,\ Q,\ ,q_0,\ \delta,\ T)$ un automate déterministe. On appelle équivalence de Nerode la relation $\sim$ définie sur Q par*

$$p \sim q \Leftrightarrow L_p = L_q$$

Question 12
Montrer que $\sim$ est une relation d'équivalence.

Question 13
Montrer que pour tout état p et q on a

1. $p \sim q \Rightarrow (\forall \alpha \in A^*)\ \delta(p, \alpha) \sim \delta(q, \alpha)$
2. $p \sim q \Rightarrow (p \in T \Leftrightarrow q \in T)$

Définition 3.2 *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 14
Montrer que la définition de $\delta_\sim$ est valide c'est-à-dire ne dépend pas du choix des représentants p et q des classes d'équivalence.
Question 15
Montrer que l'application φ du théorème 2.1 est une bijection de $Q/\!\sim$ vers Q_L.

Il reste maintenant à trouver comment calculer l'automate minimal $\mathcal{A}/\!\sim\, = (A,\ Q/\!\sim,\ [q_0]_\sim,\ \delta_\sim,\ T/\!\sim)$ à partir de l'automate initial $\mathcal{A} = (A,\ Q,\ q_0,\ \delta,\ T)$. Cela revient à calculer les classes de la relation déquivalence de Nerode $\sim$ définie par

$$p \sim q \ \ ssi \ \ L_p = L_q \ \ ssi \ \ (\forall \alpha \in A^*)\ \delta(p,\alpha) \in T \Leftrightarrow \delta(q,\alpha) \in T$$

cette définition n'est pas exploitable directement pour écrire un algorithme à cause du quantificateur universel portant sur α qui parcourt l'ensemble infini A^*.

On introduit la relation $\sim_i$ suivante (pour i élément de $\mathbb{N}$) :

$$p \sim_i q \overset{def}{\Leftrightarrow} (\forall \alpha \in A^*)(|\alpha| \leq i \Rightarrow (\delta(p,\alpha) \in T \Leftrightarrow (\delta(q,\alpha) \in T))$$

Question 16

1. Montrer que $p \sim q \ \Leftrightarrow\ (\forall i \in \mathbb{N})\ p \sim_i q$
2. Montrer que $(\forall i \in \mathbb{N})\ (p \sim_{i+1} q \Rightarrow p \sim_i q)$
3. Exprimer $p \sim_0 q$.
4. Montrer que $p \sim_{i+1} q \ \Leftrightarrow\ p \sim_i a \ et \ (\forall a \in A)\ \delta(p,a) \sim_i \delta(q,a)$