Théorie des Langages

Magistère STIC

Partiel du 28 mars 2007

durée 2 heures

Document autorisé : polycopié du cours.
Les exercices sont indépendants.
Toutes les réponses devront être correctement justifiées.

1 Langages locaux, algorithme de Glushkov

Un langage L sur l'alphabet A est dit *local* s'il existe des parties P et S de A et une partie N de A^2 telles que $L \setminus \{\varepsilon\} = (PA^* \cap A^*S) \setminus A^*NA^*$.

a) Montrer que tout langage local est reconnu par un automate déterministe ayant au plus $|A| + 1$ états.

b) Montrer qu'il existe des langages reconnaissables qui ne sont pas locaux.

c) Montrer que tout langage reconnaissable est l'image par un morphisme strictement alphabétique (c'est-à-dire que l'image d'une lettre est une lettre) d'un langage local.

d) Soient L_1 et L_2 deux langages locaux sur des alphabets A_1 et A_2 disjoints ($A_1 \cap A_2 = \emptyset$). Montrer que $L_1 \cup L_2$ et $L_1 \cdot L_2$ sont des langages locaux.

e) Soit L un langage local sur l'alphabet A. Montrer que L^* est un langage local.

f) Une expression rationnelle est dite *linéaire* si chaque lettre a au plus une occurrence dans l'expression. Montrer qu'une expression linéaire représente un langage local.

g) Montrer que tout langage rationnel est l'image par un morphisme strictement alphabétique d'un langage représenté par une expression rationnelle linéaire.

Soit L un langage sur l'alphabet A, on définit les ensembles

$$\begin{aligned} P(L) &= \{a \in A \mid aA^* \cap L \neq \emptyset\}, \\ S(L) &= \{a \in A \mid A^*a \cap L \neq \emptyset\}, \\ F(L) &= \{u \in A^2 \mid A^*uA^* \cap L \neq \emptyset\}. \end{aligned}$$

h) Montrer que si L est un langage local alors

$$L \setminus \{\varepsilon\} = (P(L)A^* \cap A^*S(L)) \setminus A^*(A^2 \setminus F(L))A^*.$$

i) Donner un algorithme qui, étant donnée une expression rationnelle représentant un langage L, détermine si $\varepsilon \in L$ et calcule les ensembles $P(L)$, $S(L)$ et $F(L)$.

j) En utilisant les questions précédentes, donner un algorithme pour construire un automate (non déterministe) à partir d'une expression rationnelle arbitraire. Exprimer le nombre d'états de l'automate en fonction de l'expression rationnelle.

k) Appliquer cet algorithme à l'expression rationnelle $(a + ba)^*b(ba + b)^*$.

2 Fonctions séquentielles

Un filtre à rebonds est une application $f : \{0,1\}^* \to \{0,1\}^*$ qui, dans un mot d'entrée, remplace un 0 par un 1 s'il est encadré par deux 1, et remplace un 1 par un 0 s'il est encadré par deux 0. On supposera que le mot d'entrée est encadré à gauche et à droite par deux 0 fictifs. Ainsi, le mot d'entrée 0101101 est transformé en 0011110 (le dernier 0 de la sortie s'explique par le fait que le dernier 1 de l'entrée est encadré par un 0 à gauche et par le 0 fictif à droite).

a) Donner (dessiner) un automate séquentiel qui réalise le filtre à rebonds f.

b) Normaliser cet automate et dessiner l'automate obtenu.

c) Calculer par raffinements successifs l'équivalence de Nerode associée à l'automate normalisé. En déduire l'automate minimal du filtre à rebonds et dessiner cet automate.

Remarque : Un filtre à rebonds peut être utilisé pour « lisser » une suite de 0 et de 1 par exemple pour réduire les imperfections d'une image.