Langages Formels

Licence Informatique – ENS Cachan

Partiel du 18 mars 2010

durée 3 heures

Document autorisé : polycopié du cours.
Toutes les réponses devront être correctement justifiées.
Le chiffre en regard d'une question est une indication sur sa difficulté ou sa longueur.

1 Automates

Remarque : Toutes les questions du type "Donner un automate qui ..." doivent être justifiées, *i.e.*, il faut donner l'automate mais aussi prouver qu'il vérifie bien la propriété requise.

On considère l'alphabet $\Sigma = \{a,b,c\}$ et le langage $L = \Sigma^* ab \Sigma^* ab \Sigma^*$ des mots contenant au moins 2 occurrences du facteur ab.

[1] **a)** Donner un automate non déterministe avec 5 états qui reconnaît le langage L.
Montrer que l'on ne peut pas reconnaître le langage L avec un automate, déterministe ou non, ayant au plus 4 états.

[1] **b)** Donner un automate déterministe pour le langage L.

[1] **c)** Donner l'automate minimal du langage L.

On considère maintenant l'alphabet $\Sigma = \{a,d,c,d\}$ et l'ensemble T des arbres étiquetés par Σ dont les nœuds internes sont binaires et étiquetés par d et la frontière est dans le langage L défini ci-dessus.

[3] **d)** Donner un automate d'arbres déterministe ascendant qui reconnaît le langage T.

[2] **e)** Montrer que le langage T ne peut pas être reconnu par un automate d'arbres déterministe descendant.

2 Langages de Dyck

Soit $\Sigma_n = \{a_1, \ldots, a_n\} \cup \{\bar{a}_1, \ldots, \bar{a}_n\}$ l'alphabet formé de n paires de parenthèses.
Soit $G_n = (\Sigma_n, \{S\}, P_n, S)$ la grammaire dont les règles sont

$$S \to a_1 S \bar{a}_1 S + \cdots + a_n S \bar{a}_n S + \varepsilon \ .$$

Le langage $D_n^* = \mathcal{L}_{G_n}(S)$ est appelé langage de Dyck sur n paires de parenthèses

[2] **a)** Montrer que $D_n^* = \mathcal{L}_{G'_n}(S)$ où la grammaire $G'_n = (\Sigma_n, \{S\}, P'_n, S)$ a pour règles

$$S \to SS + a_1 S \bar{a}_1 + \cdots + a_n S \bar{a}_n + \varepsilon \ .$$

[2] **b)** Montrer que $D_1^* = \{w \in \Sigma_1^* \mid |w|_{a_1} = |w|_{\bar{a}_1}$ et $|v|_{a_1} \geq |v|_{\bar{a}_1}$ pour tous préfixes $v \leq w\}$.

On considère le système de réécriture (type 0) $G''_n = (\Sigma_n, P''_n)$ dont les règles sont

$$P''_n = \{a_i \bar{a}_i \to \varepsilon \mid 1 \leq i \leq n\} \ .$$

[3] **c)** Montrer que $D_n^* = \{w \in \Sigma_n^* \mid w \xrightarrow{*} \varepsilon$ dans $G''_n\}$.

[1] **d)** Soit $\mathcal{A} = (Q, \Sigma_n, T, I, F)$ un automate fini. Étant donné un couple $(p,q) \in Q^2$, montrer que l'on peut décider s'il existe un mot de Dyck $w \in D_n^*$ qui est l'étiquette d'un chemin de p à q dans $\mathcal{A}$: $p \xrightarrow{w} q$.

Soit Γ un alphabet disjoint de Σ_n, $\Sigma = \Sigma_n \cup \Gamma$ et $L \subseteq \Gamma^*$ un langage.
On définit la clôture $\mathrm{clot}(L) = \{v \in \Gamma^* \mid \exists w \in L$ tel que $w \xrightarrow{*} v$ dans $G''_n\}$.

[3] **e)** Montrer que si L est reconnaissable, alors $\mathrm{clot}(L)$ aussi.

On définit la réduction $\mathrm{red}(L) = \{v \in \mathrm{clot}(L) \mid v \not\to$ dans $G''_n\}$.

[1] **f)** Montrer que si L est reconnaissable, alors $\mathrm{red}(L)$ aussi.

[4] **g)** Soit $\mathcal{A} = (Q, \Sigma_n, T, I, F)$ un automate fini. On définit pour $k \geq 0$ les ensembles

$$\begin{aligned} R &= \{(p,q) \in Q^2 \mid \exists w \in D_n^* \text{ tel que } p \xrightarrow{w} q \text{ dans } \mathcal{A}\} \\ R_0 &= \{(p,p) \mid p \in Q\} \\ R_{k+1} &= \{(p,q) \mid \exists r \in Q \text{ tel que } (p,r) \in R_k \wedge (r,q) \in R_k\} \cup \\ &\qquad \{(p,q) \mid \exists (p',q') \in R_k, \exists i \in \{1, \ldots, n\} \text{ tels que } p \xrightarrow{a_i} p' \wedge q' \xrightarrow{\bar{a}_i} q \text{ dans } \mathcal{A}\} \end{aligned}$$

Montrer que la suite $(R_k)_{k \geq 0}$ est croissante et que $R = \bigcup_{k \geq 0} R_k$.
En déduire un algorithme pour calculer R et donner sa complexité en fonction de la taille de $\mathcal{A}$.