Langages Formels

Licence Informatique – ENS Cachan

Partiel du 24 mars 2016

durée 2 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 Relations reconnaissables

Un entier est codé en binaire en commençant par le bit de poids faible. Soit $B = \{0, 1\}$ l'alphabet binaire. On note $\overline{w}^2$ l'entier codé par le mot $w \in B^*$. Par exemple, $\overline{1011}^2 = 13$ (en base 10), $\overline{01100}^2 = 6$ et $\overline{\varepsilon}^2 = 0$. On code aussi un k-uple d'entiers $(n_1, \ldots, n_k)$ par un mot W sur l'alphabet B^k : l'entier n_i est codé par le mot w_i obtenu en projetant W sur la i-ème composante. On note $\overline{W}^2 = (n_1, \ldots, n_k)$. Par exemple, le triplet $(5, 2, 16)$ est codé par le mot $W = (1, 0, 0)(0, 1, 0)(1, 0, 0)(0, 0, 0)(0, 0, 1)$ sur l'alphabet B^3, puisque $\overline{10100}^2 = 5$, $\overline{01000}^2 = 2$ et $\overline{00001}^2 = 16$.

[1] **a)** Montrer que le langage $L_1 = \{W \in (B^2)^* \mid \overline{W}^2 = (n_1, n_2) \text{ avec } n_1 \leq n_2\}$ est reconnaissable en donnant une expression rationnelle pour L_1.

[1] **b)** Montrer que le langage $L_2 = \{W \in (B^2)^* \mid \overline{W}^2 = (n_1, n_2) \text{ avec } 3n_1 = n_2\}$ est reconnaissable en donnant un automate déterministe ayant au plus 3 états pour $L2$.

[1] **c)** Montrer que le langage $L_3 = \{W \in (B^2)^* \mid \overline{W}^2 = (n_1, n_2) \text{ avec } 3n_1 \leq n_2\}$ est reconnaissable en utilisant des propriétés de clôture des langages reconnaissables. On *ne* donnera *pas* explicitement un automate ou une expression rationnelle pour L_3.

2 Grammaires

On considère la grammaire $G = (\Sigma, V, P, S)$ avec $\Sigma = \{a, b\}$, $V = \{S\}$ et les trois règles suivantes : $S \to Sa + SS + b$.

[1] **a)** Construire une grammaire $G' = (\Sigma, V', P', S)$ équivalente à G, i.e., telle que $\mathcal{L}_G(S) = \mathcal{L}_{G'}(S)$, et telle que $P' \subseteq V' \times \Sigma(\Sigma \cup V')^*$.

3 Automate à pile

On considère une extension des automates à pile dans laquelle les transitions sont conditionnées par des contraintes rationnelles sur le contenu de pile.

Un automate à pile généralisé est un tuple $\mathcal{A} = (Q, \Sigma, Z, \mathtt{Push}, \mathtt{Pop}, q_0, F)$ avec Q un ensemble fini d'états, Σ l'alphabet d'entrée, Z l'alphabet de pile, $q_0 \in Q$ l'état initial, $F \subseteq Q$ l'ensemble des états acceptants et les ensembles *finis* de transitions

$$\begin{aligned} \mathtt{Push} &\subseteq Q \times \mathrm{Rat}(Z^*) \times (\Sigma \cup \{\varepsilon\}) \times Z \times Q \\ \mathtt{Pop} &\subseteq Q \times Z \times \mathrm{Rat}(Z^*) \times (\Sigma \cup \{\varepsilon\}) \times Q \end{aligned}$$

Comme pour les automates à pile classiques, la sémantique est définie par un système de transitions infini $\mathcal{T}$ dont les configurations sont de la forme $qh \in QZ^*$ avec $q \in Q$ et $h \in Z^*$. La configuration initiale est q_0. L'ensemble des configurations acceptantes est FZ^*. Les transitions de $\mathcal{T}$ sont définies par :

— $qh \xrightarrow{a} q'zh$ s'il existe $(q, K, a, z, q') \in \mathtt{Push}$ avec $h \in K$,

— $qzh \xrightarrow{a} q'h$ s'il existe $(q, z, K, a, q') \in \mathtt{Pop}$ avec $zh \in K$.

L'objectif est de montrer que les automates à pile généralisés ont le même pouvoir d'expression que les automates à pile classiques.

[1] **a)** Etant donnés des langages rationnels $K_1, \ldots, K_n$ sur l'alphabet Z, montrer que l'on peut construire un automate fini déterministe et complet $\mathcal{B} = (P, Z, \delta, p_0)$ et des ensembles d'états $P_1, \ldots, P_n \subseteq P$ tels que pour tout $1 \leq i \leq n$, le langage K_i est reconnu par $\mathcal{B}$ en utilisant $P_i \subseteq P$ comme ensemble d'états acceptants :

$$K_i = \{h \in Z^* \mid \delta(p_0, h) \in P_i\}\,.$$

[1] **b)** Etant donné un automate à pile généralisé $\mathcal{A}$, construire un automate à pile classique $\mathcal{A}'$ équivalent, i.e., tel que $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}')$.
Il faut bien sûr prouver que $\mathcal{A}$ et $\mathcal{A}'$ acceptent le même langage.

[1] **c)** Expliquer comment généraliser à des transitions qui peuvent empiler ou dépiler un nombre arbitraire de symboles :

$$\begin{aligned} \mathtt{Push} &\subseteq Q \times \mathrm{Rat}(Z^*) \times (\Sigma \cup \{\varepsilon\}) \times Z^* \times Q \\ \mathtt{Pop} &\subseteq Q \times Z^* \times \mathrm{Rat}(Z^*) \times (\Sigma \cup \{\varepsilon\}) \times Q \end{aligned}$$

Les ensembles de transitions `Push` et `Pop` sont bien sûr toujours finis.
Pour cette question, on ne demande pas une construction détaillée.