Langages Formels

Licence Informatique – ENS Cachan

Examen du 26 mai 2014

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 Mélanges de mots

Soit Σ un alphabet et $u, v \in \Sigma^*$ deux mots. Le mélange de u et v est le langage

$$u \sqcup\!\sqcup v = \{u_1 v_1 \cdots u_n v_n \mid n \geq 0,\ u_i, v_i \in \Sigma \cup \{\varepsilon\},\ u = u_1 \cdots u_n \text{ et } v = v_1 \cdots v_n\}.$$

Par exemple, $ab \sqcup\!\sqcup ca = \{abca, acba, caba, acab, caab\}$.

Cette opération est étendue aux langages en posant $K \sqcup\!\sqcup L = \bigcup_{u \in K, v \in L} u \sqcup\!\sqcup v$ pour $K, L \subseteq \Sigma^*$.

[2] **a)** Soit $\Sigma = \{a, b, c\}$ un alphabet. Construire l'automate minimal du langage $\Sigma^* ab \Sigma^* \sqcup\!\sqcup \Sigma^* b \Sigma^*$.

Dans la suite, l'alphabet Σ est de nouveau arbitraire. On introduit une copie $\overline{\Sigma}$ de l'alphabet Σ et on note $\overline{a} \in \overline{\Sigma}$ la copie de la lettre $a \in \Sigma$. On considère les morphismes Π, Π_1 et Π_2 de $(\Sigma \cup \overline{\Sigma})^*$ dans Σ^* définis pour $a \in \Sigma$ par $\Pi(a) = \Pi(\overline{a}) = a$, $\Pi_1(a) = \Pi_2(\overline{a}) = a$, et $\Pi_1(\overline{a}) = \Pi_2(a) = \varepsilon$.

[3] **b)** Montrer que $K \sqcup\!\sqcup L = \Pi(\Pi_1^{-1}(K) \cap \Pi_2^{-1}(L))$.
En déduire que si K est rationnel (resp. linéaire ou algébrique) et que L est rationnel alors $K \sqcup\!\sqcup L$ est aussi rationnel (resp. linéaire ou algébrique).

[2] **c)** Le langage $\{a^n b^n \mid n \geq 0\} \sqcup\!\sqcup \{c^p d^p \mid p \geq 0\}$ est-il algébrique ? Justifier la réponse.

On suppose dans la suite que Σ est partitionné en $\Sigma = \Sigma_1 \cup \Sigma_2$ avec $\Sigma_1 \cap \Sigma_2 = \emptyset$. On note π_1 et π_2 les projections de Σ^* sur Σ_1^* et Σ_2^* respectivement. On considère deux langages $L_1 \subseteq \Sigma_1^*$ et $L_2 \subseteq \Sigma_2^*$ et on note $L = L_1 \sqcup\!\sqcup L_2$.

[2] **d)** Montrer que $L = \{u \in \Sigma^* \mid \pi_1(u) \in L_1 \wedge \pi_2(u) \in L_2\}$.
Pour $i \in \{1, 2\}$, on suppose que L_i est reconnu par un morphisme $\varphi_i : \Sigma_i^* \to M_i$ où M_i est un monoide fini. Construire un monoide fini M et un morphisme $\varphi : \Sigma^* \to M$ qui reconnaît L.

[2] **e)** Soit $u \in \Sigma^*$. On note $u_1 = \pi_1(u)$ et $u_2 = \pi_2(u)$. Montrer que $u^{-1}L = u_1^{-1}L_1 \sqcup\!\sqcup u_2^{-1}L_2$.
Exprimer le nombre n de résiduels de L en fonction des nombres n_1 et n_2 de résiduels de L_1 et L_2. Justifier votre réponse.
Attention : ne pas oublier que L_1 et L_2 peuvent avoir un résiduel vide.

2 Analyse LL

On considère la grammaire G définie par

$$\begin{aligned}\mathbf{Stmt} &\to \mathtt{id} \mid \mathbf{Var} := \mathbf{Exp}\\ \mathbf{Var} &\to \mathtt{id} \mid \mathbf{Var}[\mathbf{Exp}]\\ \mathbf{Exp} &\to \mathtt{num} \mid \mathbf{Var}\end{aligned}$$

sur l'alphabet terminal $\Sigma = \{\mathtt{id}, \mathtt{num}, :=, [,]\}$ et l'alphabet non terminal $V = \{\mathbf{Stmt}, \mathbf{Var}, \mathbf{Exp}\}$.

[5] **a)** Pour chaque variable $x \in V$, calculer $\mathrm{First}_2(x)$ et $\mathrm{Follow}_2(x)$.
Montrer que G n'est pas fortement LL(2).
Existe-t-il k tel que G soit une grammaire LL(k) ?

3 Fonctions séquentielles

On considère les mots sur l'alphabet binaire $\Sigma = \{0, 1\}$ qui codent des entiers en binaire en commençant par le bit de poids faible : le mot $u = a_0 a_1 \cdots a_n$ code l'entier $\overline{u}^2 = \sum_{i=0}^{n} a_i 2^i$.

[3] **a)** On considère la fonction $g \colon \Sigma^* \to \Sigma^*$ qui associe à un mot $u \in \Sigma^*$ le mot $v \in \Sigma^*$ tel que $|v| = |u|$ et $\overline{v}^2 = \overline{u}^2 \mathbin{\mathtt{div}} 3$.
Montrer que la fonction g n'est pas séquentielle.

[3] **b)** On considère la fonction $h \colon \Sigma^* \to \Sigma^*$ qui coïncide avec g sur les mots $u \in \Sigma^*$ tels que $\overline{u}^2 \mathbin{\mathtt{mod}} 3 = 0$, et qui n'est pas définie sur les mots $u \in \Sigma^*$ tels que $\overline{u}^2 \mathbin{\mathtt{mod}} 3 \neq 0$.
Montrer que la fonction h est séquentielle. Construire son automate séquentiel minimal.

4 Logique et automates d'arbres

On considère la logique du premier ordre $\mathrm{FO}(\Sigma, <, \downarrow_1, \downarrow_2)$ sur les arbres binaires étiquetés par l'alphabet Σ ($x < y$ signifie que le nœud x est un ancêtre du nœud y).

[2] **a)** Donner des formules closes du premier ordre pour les propriétés suivantes :

ψ_1 la concaténation des étiquettes de chaque branche est un mot de $(ab)^+$,

ψ_2 chaque branche issue d'un nœud étiqueté a comporte au moins un nœud étiqueté b.

[4] **b)** On considère la formule φ

$$\forall x \forall z \left[\big(P_a(x) \wedge x < z \wedge \neg \exists y\, z < y \big) \to \exists y \Big(x < y \leq z \wedge P_c(y) \wedge \forall z\, (x < z < y \to P_b(z)) \Big) \right]$$

Donner un automate d'arbres $\mathcal{A}$ déterministe (ascendant) et complet qui reconnaît $\mathcal{L}(\varphi)$.
Il faut bien sûr prouver que $\mathcal{A}$ reconnaît $\mathcal{L}(\varphi)$.