Langages Formels

Licence Informatique – ENS Cachan

Partiel du 31 mars 2011

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 Arithmétique de Presburger

On s'intéresse ici à la théorie logique du premier ordre des entiers munis de l'addition (mais pas de la multiplication). Plus précisément, on fixe un ensemble infini $\mathcal{X}$ de variables. On définit l'arithmétique de PRESBURGER, comme le plus petit ensemble $\mathcal{P}$ de formules logiques tel que
- si $x, y, z \in \mathcal{X}$ alors $x = 0$ et $x + y = z$ sont des formules de $\mathcal{P}$,
- si $x \in \mathcal{X}$ et $\varphi, \psi \in \mathcal{P}$, alors $\varphi \wedge \psi$, $\varphi \vee \psi$, $\neg\varphi$, $\forall x\ \varphi$ et $\exists x\ \varphi$ sont des formules de $\mathcal{P}$.

Dans la suite, $\mathcal{V} \subseteq \mathcal{X}$ dénotera un ensemble fini de variables. Une $\mathcal{V}$-assignation est un tuple $\sigma \in \mathbb{N}^{\mathcal{V}}$. On note $\mathrm{Free}(\varphi)$ l'ensemble des variables libres d'une formule φ. Si $\mathrm{Free}(\varphi) \subseteq \mathcal{V}$ et $\sigma \in \mathbb{N}^{\mathcal{V}}$, on note $\sigma \models \varphi$ si la formule φ est vraie pour l'assignation σ.

[2] **a)** Donner des formules de Presburger équivalentes à $x = y$, à $x < y$, à "x est pair", et à $x = 1$.

On choisit de coder les entiers en binaire, avec bit de poids fort à gauche. On définit une fonction de décodage $\nu : \{0,1\}^* \longrightarrow \mathbb{N}$ par

$$\nu(\varepsilon) = 0 \qquad \nu(w0) = 2\nu(w) \qquad \nu(w1) = 1 + 2\nu(w)$$

Remarquons que cette fonction est surjective, totale, mais pas injective.

Une assignation $\sigma \in \mathbb{N}^{\mathcal{V}}$ sera codée par un mot sur l'alphabet $\Sigma_{\mathcal{V}} = \{0,1\}^{\mathcal{V}}$. Soit $w \in (\Sigma_{\mathcal{V}})^*$ un mot et $x \in \mathcal{V}$ une variable, on note w_x la projection du mot w sur la composante x de l'alphabet $\Sigma_{\mathcal{V}}$. On note $\overline{\nu}(w) = (\nu(w_x))_{x \in \mathcal{V}} \in \mathbb{N}^{\mathcal{V}}$ l'assignation codée par le mot w. Par exemple, $\overline{\nu}\begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 18 \\ 7 \\ 6 \end{pmatrix}$.

Remarquons que si $\mathcal{V} = \emptyset$, alors $\Sigma_{\mathcal{V}} = \emptyset$ et donc $(\Sigma_{\mathcal{V}})^* = \{\varepsilon\}$ est réduit au mot vide.

Finalement, pour $\varphi \in \mathcal{P}$ et $\mathcal{V}$ contenant $\mathrm{Free}(\varphi)$, on note $[\![\varphi]\!]_{\mathcal{V}} = \{w \in (\Sigma_{\mathcal{V}})^* \mid \overline{\nu}(w) \models \varphi\}$.

[2] **b)** Donner des expressions rationnelles pour les langages $[\![x = 0]\!]_{\{x\}}$ et $[\![x + y = z]\!]_{\{x,y,z\}}$.

[4] **c)** Montrer que pour toute formule $\varphi \in \mathcal{P}$ et pour tout $\mathcal{V}$ contenant $\mathrm{Free}(\varphi)$, on peut effectivement construire un automate fini reconnaissant le langage $[\![\varphi]\!]_{\mathcal{V}}$.
Indication : on pourra utiliser des projections $(\Sigma_{\mathcal{W}})^* \to (\Sigma_{\mathcal{U}})^*$ pour $\mathcal{U} \subseteq \mathcal{W}$.

[2] **d)** En déduire que l'arithmétique de Presburger est décidable, i.e., qu'on peut décider si une formule close est valide. Quelle est la complexité de cette procédure de décision ?

2 Expressions préfixes

Dans les expressions arithmétiques en notation préfixe (ou polonaise inverse), l'opérateur s'écrit avant les opérandes et on n'utilise pas de parenthèses. Par exemple, l'expression infixe parenthésée $((a+b)\times(b+c))$ s'écrit $\times + a\ b + b\ c$ en notation préfixe.

[3] Donner une grammaire algébrique G qui engendre les expressions arithmétiques en notation préfixe sur l'alphabet $\Sigma = \{+, \times, a, b, c\}$, i.e., avec les opérateurs binaires $+$ et $\times$ et les constantes a, b et c.
Donner l'arbre de dérivation de $\times + a\ b + b\ c$.
Construire un automate d'arbres déterministe ascendant qui reconnaît les arbres de dérivation de la grammaire G.

3 Rationnels, Linéaires et Algébriques

Soit Σ un alphabet, $\# \in \Sigma$ une lettre de Σ et $A = \Sigma \setminus \{\#\}$. Pour $K \subseteq A^*$, on définit le langage $L_K = \{u\#v \in K\#A^* \mid |u| = |v|\}$.

[4] **a)** Montrer que K est rationnel si et seulement si L_K est linéaire.
Indication : considérer (ou construire) une grammaire linéaire droite pour le langage K.

[3] **b)** Montrer que si L_K est algébrique alors L_K est linéaire.

4 Mélanges de mots

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

$$u \sqcup\!\sqcup\, v = \{u_1v_1\cdots u_nv_n \mid n \geq 0,\ u_i, v_i \in \Sigma^*,\ u = u_1\cdots u_n \text{ et } v = v_1\cdots v_n\}.$$

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 $\{a^nb^n \mid n \geq 0\} \sqcup\!\sqcup\, \{c^pd^p \mid p \geq 0\}$ est-il algébrique ? Justifier la réponse.