Université Bordeaux I
UFR de Mathématiques et Informatique

Session de Février 1998

Théorie des Langages

durée 2 heures; documents interdits

Nota: *L'épreuve comprend deux exercices notés sur 5 points chacuns, et un problème, noté sur 10 points, constitué de 2 questions de 5 points chacunes. La troisième question, facultative, a pour seul but de montrer la différence entre une méthode générale et une méthode spécifique, et ne sera pas notée. Par contre dans les questions obligatoires, la plus grande attention devra être portée à la qualité de la rédaction qui devra être concise et précise.*

1 Exercices

Exercice 1 (5 points) On considère les fonctionnelles H et K qui associent à f et g, fonctions binaires, les fonctions:

- $H(f,g) : (x,y) \mapsto f(g(x,y), f(x,y))$,
- $K(f,g) : x \mapsto (f(g(x,x), f(x,x))$.

1. Définir ces fonctionnelles par des λ-termes.
2. Calculer les types les plus généraux de ces deux termes et vérifier la compatibilité des types trouvés avec les définitions de H et de K.

Exercice 2 (5 points) Soit l'équation $S = SS \cup \{ab\} \cup \{\epsilon\}$ dont on cherche les solutions incluses dans $\{a,b\}^\star$.
1. Quelle est sa plus petite solution?
2. Quelle est sa plus grande solution?
3. Les langages suivants sont-ils solution de l'équation précédente?

1. $(ab \cup aab)^\star$
2. $(\{a^p b \;/\; p = 1 \text{ } ou \text{ } p \text{ } premier\})^\star$

2 Problème (10 points)

Pour tout langage L sur l'alphabet X on définit

$$FG(L) = \{f \in X^\star \;/\; \exists h \in X^\star \text{ et } fh \in L\}.$$

Le but du problème est d'étudier les langages $FG(L)$ pour certains langages algébriques L.
1. Construire les grammaires qui engendrent $FG(L_i)$ où L_i est le langage suivant (5 points):

1. $L_1 = \{a^n b^m \ / \ n, m \geq 0\}$

2. $L_2 = \{a^n b^n \ / \ n \geq 0\}$

3. $L_3 = \{u \in \{a, b\}^\star \ / \ |u|_a = |u|_b\}$

4. L_4, engendré par $G =< \{a, b\}, \{S\}, \{S \rightarrow aSS, S \rightarrow b\}, S >$

2. Etude particulière du langage des parenthèses. (5 points)
On considère le langage des parenthèses D sur l'alphabet $X = \{a, b\}$, caractérisé par la condition, $\forall f \in D, |f|_a = |f|_b$ et $\forall g$ facteur gauche de f, $|g|_a \geq |g|_b$.
On rappelle que ce langage est engendré par la grammaire G_D de règles de production $S \rightarrow aSbS$, $S \rightarrow \epsilon$. On pose $F = FG(D)$.

1. Donner les mots de F, classés par ordre de longueur croissante, jusqu'à la longueur 4 incluse.

2. Montrer que tout mot $f \in F \backslash D$ s'écrit de façon unique en $f = gah$ avec $g \in D$ et $h \in F$.

3. En déduire une grammaire G_F de F, non ambiguë, sachant que la grammaire G_D est non ambiguë.

3. Cas général **(facultatif)**.
On considère un langage algébrique L engendré par une grammaire $G =< X, V, \mathcal{P}, S >$ sous forme normale de Chomsky. On lui associe la grammaire $\bar{G} =< X, W, \mathcal{Q}, \Sigma >$ obtenue par la transformation suivante, notée $\mathcal{T}$:

- $\bar{V}$ est une copie de V: $\bar{V} = \{\bar{A} \ / \ A \in V\}$.
- $W = V \cup \bar{V} \cup \{\Sigma\}$ où Σ est un nouveau symbole.
- A toute règle $A \rightarrow BC \in \mathcal{P}$ on associe le triplet de règles $\{A \rightarrow B\bar{C}, A \rightarrow B, A \rightarrow \bar{B}\}$.
- Soit $\bar{\mathcal{P}}$ le nouvel ensemble de règles obtenu, et $\mathcal{Q} = \mathcal{P} \cup \bar{\mathcal{P}} \cup \{\Sigma \rightarrow S, \Sigma \rightarrow \bar{S}\}$

1. Soit $G =< \{a, b, c\}, \{S\}, \{S \rightarrow aSa, \ S \rightarrow bSb, \ S \rightarrow c\}, \ S >$. Mettre G sous forme normale de Chomsky et lui appliquer la transformation $\mathcal{T}$.

2. Monter que si G est une grammaire sous forme normale de Chomsky, et si $L = L(G)$, alors $L(\mathcal{T}(G))$ engendre $FG(L)$.

3. Généraliser la transformation $\mathcal{T}$ à toute grammaire algébrique et l'appliquer à la grammaire du langage des parenthèses vue ci-dessus.

4. Montrer par un exemple que la grammaire ainsi obtenue est ambiguë.