Institut Galilée - Université Paris 13 — Théorie des langages
DEUG MIAS 2ème année — 24 mars 2000

Devoir à la maison à rendre le 7 avril

Sauf quand cela vous est explicitement demandé, ne faites pas de preuves, donnez simplement la réponse.
Par ailleurs, on adopte les conventions suivantes :
le mot vide est noté ϵ ; l'alphabet de référence est noté Σ (sigma) ; les non-terminaux sont écrits en majuscule ; les terminaux sont écrits en minuscule.

Merci d'indiquer, en face de chaque question, combien de temps vous y avez consacré. Cela ne modifiera en rien votre note, c'est simplement pour me donner une idée du temps qu'un élève passe en moyenne à tel ou tel type de question avant de réussir (ou pas !) à y répondre.

En cas de problème dans l'énoncé, vous pouvez me contacter à Cyril.Banderier@inria.fr.

1. Monoïdes

1.1. Rappelez la définition d'un monoïde.

1.2. "Moralement", qu'est-ce qu'un monoïde libre ?

1.3. Donnez un exemple de monoïde libre et d'un monoïde non libre.

1.4. L'étoile d'un langage $\mathcal{L}$ est définie par $L^* = \cup_{n\geq 0} U^n$. Donnez tous les mots de longueur < 10 de $\{ab, b\}^*$.

2. Grammaires

Pour les questions suivantes, on travaille avec l'alphabet $\Sigma = \{a, b\}$.

2.1. Pour chacun des langages L_i suivants, donnez une grammaire G_i qui le reconnaisse.
L_1={les mots ne contenant pas deux a consécutifs} ;
L_2={les mots contenant autant de a que de b} ;
L_3 ={les mots ne contenant que des a (au moins un) puis que des b (au moins un)} ;
L_4={les mots ne contenant que des a puis que des b, avec "nombre de b" $>$ "nombre de a" > 0} ;
L_5={les mots dont la parité du nombre de a égale celle du nombre de b}
(par exemple $ababaa \in L_5, bbba \in L_5$ mais $bba \notin L_5$ et $ababa \notin L_5$).

2.2. Deux des langages sont réguliers, lesquels ? Donnez une grammaire *régulière* pour chacun de ces deux langages.

2.3. Prouvez à l'aide du lemme de l'étoile qu'au moins l'un des langages ci-dessus n'est pas régulier.

3. Expressions régulières

Une *expression régulière* est une façon de coder sous forme compacte un langage régulier, en n'utilisant que * (pour l'étoile), $+$ (pour l'union), $-$ (pour la différence), le parenthésage et les lettres de l'alphabet.
$(a + b)$ désigne ainsi l'union de $\{a\}$ et $\{b\}$, c'est-à-dire $\{a, b\}$
$(a + b)^*$ désigne ainsi l'étoile de $\{a, b\}$, c'est-à-dire $\{a, b\}^*$, tous les mots du langage.

3.1. Construisez un automate qui reconnaisse l'expression régulière $(a^*b^*)^*b + (ba)^*$.

3.2. Quel est le langage reconnu par l'automate suivant ? (Donnez la réponse sous la forme d'une expression régulière.)

3.3. Donnez (sous forme d'une expression régulière) le langage engendré par la grammaire suivante :
$A \to aBb|aC|\epsilon \qquad B \to Ab|a \qquad C \to bC|A$ (avec A comme axiome).
Donnez un arbre de dérivation de $aabbb$. La grammaire est-elle ambigüe ?

4. Automates

4.1. Donner un automate qui engendre le même langage que l'automate de la question 3.2, mais dont tous les états soient accessibles.

4.2. Déterminisez ce nouvel automate.

4.3. Minimisez ce dernier automate (c'est-à-dire donnez un automate [pas forcément déterministe] qui reconnaisse le même langage mais qui a le moins d'états possibles).

5. Analyse et théorie des langages

Soit $\mathcal{L}$ un langage et notons ℓ_n le nombres de mots de $\mathcal{L}$ longueur n. Considérons maintenant la série $L(z)$ (dite "série génératrice du langage $\mathcal{L}$")

$$L(z) = \sum_{n \geq 0} \ell_n z^n.$$

Un théorème du à Noam Chomsky et Marcel-Paul Schützenberger (1963) dit que

- si $\mathcal{L}$ est un langage régulier alors $L(z)$ est une fraction rationnelle,
- si $\mathcal{L}$ est engendré par une grammaire alors $L(z)$ est algébrique.

Rappelons qu'une fraction rationnelle est un quotient de deux polynômes et qu'une fonction f est dite algébrique si elle peut être annulée par un polynôme (il existe une relation du genre $f(z)^{27} - 456 f(z)^{13} + 4/6 f(z) - 1 = 0$).

En complément du théorème, on a même une méthode pour trouver "explicitement" $L(z)$: si la grammaire est non-ambigüe, la transformation suivante

$$\begin{array}{lll} S \to aSbScS|dB|ac|\epsilon & \Rightarrow & S(z) = z^3 S(z)^3 + zB(z) + z^2 + 1 \\ B \to aBBc|\epsilon & \Rightarrow & B(z) = z^2 B(z)^2 + 1 \end{array}$$

donne un système d'équations qui, si on le résout, fournit une formule pour $S(z) = L(z)$.

5.1. Une séquence de n tirs à pile ou face donne 2^n mots de longueur n possibles (codons "pile face pile" par "010", etc.). Donner une expression *simple* de la série génératrice $L(z)$ du langage $\mathcal{L}$ "suites des tirs à pile ou face".

Observation : puisque le langage peut être engendré par la grammaire réguliere $S \to 0S|1S|\epsilon$, votre résultat doit être cohérent avec le théorème et vous devez obtenir une fraction rationnelle.

5.2. Sur un quadrillage avec un repère, on part du point $(0,0)$, puis on fait soit des pas $(+1,+1)$ montants en diagonale, ou $(-1,-1)$ descendants en diagonale. On s'intéresse aux chemins partant de (0,0) et atteignant au final l'axe $y = 0$ sans jamais passer en-dessous. En codant par a les pas montants et par b les pas descendants, les premiers chemins valides sont donc : $\epsilon, ab, abab, aabb, ababab, aabbab, abaabb, aababb$... Les mots possibles sont appelés "mots de Dyck" ou encore "chemins de Dyck".

Trouvez une grammaire (non-ambigüe) qui engendre les mots de Dyck, puis, en utilisant la méthode de Schützenberger, donnez une expression simple de la série génératrice $L(z)$ du langage $\mathcal{L}$ des mots de Dyck. Quel est le rayon de convergence de cette série ? Qu'en concluez-vous ? (penser au lien entre coefficients et rayon de convergence).

5.3. Prouvez à l'aide du théorème de Chomsky-Schützenberger qu'un langage de la question 2.1 n'est pas régulier.

6. Automates et chaînes de Markov

Une chaîne de Markov est un automate dont les transitions sont "probabilisées".

6.1. Dessinez la chaîne de Markov qui correspond au jeu suivant : on tire à pile ou face et on s'arrête (état final !) quand on a obtenu "PPP".

6.2. Idem avec "PFP". Qu'en concluez-vous ?

7. Compilation

Pour chacune des phases suivantes, dire en une courte phrase en quoi elle consiste : analyse lexicale, analyse syntaxique, analyse sémantique, optimisation, génération de code, exécution.