Introduction à la Calculabilité

Examen du 24 août 2002

Livres fermés. Durée : 3h30.

Répondez à chaque question sur une feuille séparée sur laquelle figurent votre nom et votre section. Soyez bref et concis, mais précis.

1. L'ensemble des valeurs de $x \in \mathbb{R}$ où la fonction $\tan(x)$ est discontinue est-il dénombrable ? Justifier rigoureusement.

2. Soit l'alphabet $\Sigma = \{a,b,c\}$. On définit sur Σ les deux langages suivants : L_1 est le langage décrit par l'expression régulière $(a^*(a \cup b)^*b(b \cup c)^*)^*$, et L_2 est le langage généré par la grammaire régulière suivante :

$$\begin{array}{lcl} S & \to & aaA \mid \varepsilon \\ A & \to & bcA \mid bS. \end{array}$$

   On demande de :

   (a) construire des automates finis non déterministes qui acceptent ces deux langages ;

   (b) déterminiser ces automates ;

   (c) construire un automate qui accepte le complément de L_1 et calculer son intersection avec l'automate déterministe qui accepte L_2. Que peut-on en déduire sur la relation entre L_1 et L_2 ?

3. Supposons que l'on ajoute à la définition des expressions régulières un nouvel opérateur $\circ$ défini comme suit : si α et β sont deux expressions régulières, alors $(\alpha \circ \beta)$ est également une expression régulière, et $L((\alpha \circ \beta)) = \{w \in L(\alpha) \mid (\exists x \in \Sigma^*)(\exists y \in L(\beta))w = xy\}$ (i.e. $L((\alpha \circ \beta))$ est le langage des mots appartenant à $L(\alpha)$ qui ont un suffixe dans $L(\beta)$). Tout langage dénoté par ce nouveau type d'expression régulière est-il régulier ? Justifier.

4. Enoncer et démontrer le théorème du gonflement pour les langages hors-contexte.

5. Soit l'alphabet $\Sigma = \{a,b,\$\}$. Construire une machine de Turing qui, en démarrant de la configuration $(s_0,\varepsilon,\$w\$)$ où s_0 est l'état initial de la machine et où $w \in \{a,b\}^*$, s'arrête dans la configuration $(s_f,\varepsilon,\$w'\$)$ où s_f est un état accepteur de la machine et où w' est défini ainsi :

$$w' = \begin{cases} a^{|w|-i-1}w_0 \ldots w_i & \text{si il existe } i \in \mathbb{N} \text{ pour lequel } w_i = b, \\ & \text{et si pour tout } j \text{ tel que } i < j < |w|, \text{ on a } w_j = a \text{ ;} \\ w & \text{sinon.} \end{cases}$$

Remarque : Pour rappel, w_i désigne la $i°$ lettre du mot w, l'énumération commençant à zéro.

6. (a) Le nombre de Gödel d'un mot $w = w_0 \ldots w_\ell$ défini sur un alphabet Σ comportant k symboles, est le naturel $gd(w)$ égal à :

$$\sum_{i=0}^{l} (k+1)^{\ell-i} gd(w_i)$$

où l'on a préalablement attribué à chaque élément a de Σ, un naturel $gd(a)$ compris entre 1 et k. Soit maintenant une fonction $f(x)$ dont l'argument x est le nombre de Gödel d'un mot w sur l'alphabet Σ et dont la valeur $f(x)$ est définie comme étant la longueur (i.e. le nombre de caractères) du mot w encodé par x. Montrer que cette fonction est $\mu-$récursive.

Remarque : Toutes les fonctions $\mu-$récursives étudiées au cours théorique et aux répétitions sont considérées comme étant connues.

(b) Cette fonction est-elle également primitive récursive? Justifier.

Remarque : Toutes les fonctions primitives récursives étudiées au cours théorique et aux répétitions sont considérées comme étant connues.

7. (a) Définir la classe R et la classe RE. Le langage L_2 de la question (2) appartient-il à la classe R? A la classe RE? Justifier.

(b) Existe-t-il des langages qui ne sont ni dans la classe R, ni dans la classe RE? Si oui, donner un tel langage et démontrer qu'il n'appartient ni à R, ni à RE. Si non, justifier.

(c) Soit M une machine de Turing et s_0 son état initial. Montrer que le problème consistant à tester si on passe au moins deux fois par s_0 dans l'exécution de M sur le mot vide, est indécidable mais récursivement énumérable.

Suggestion : Réduction à partir du problème L_ε consistant à tester si le mot vide appartient au langage accepté par une machine de Turing.

8. (a) Enoncer le théorème de Cook.

(b) Est-il possible de découvrir un algorithme polynomial qui résolve le problème SAT? Expliquer. Votre réponse s'applique-t-elle également au problème du voyageur de commerce? Justifier.

(c) Donner et justifier la structure d'une preuve permettant de démontrer qu'un problème est NP-complet.