Introduction à la Calculabilité

Examen du 6 janvier 2003

Livres fermés. Durée : 4h00.

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

1. Soit A un automate fini non déterministe. L'ensemble des mots acceptés par A est-il dénombrable ? Justifier.

2. Soit la grammaire hors-contexte G suivante définie sur l'alphabet $\Sigma = \{a,b,c\}$:

$$\begin{array}{lcl} S & \to & XcX \\ X & \to & YX \mid Xb \mid \varepsilon \\ Y & \to & YY \mid a. \end{array}$$

   Appelons L le langage généré par G. On demande de :

   (a) construire une expression régulière qui représente L ;

   (b) construire une grammaire régulière qui génère L ;

   (c) construire un automate fini non déterministe qui accepte L ;

   (d) déterminiser l'automate ainsi obtenu.

3. (a) Enoncer et démontrer le théorème du gonflement (seconde version) pour les langages réguliers.

   (b) Soit le langage suivant, défini sur l'alphabet $\Sigma = \{a,b\}$:

$$L = \{w \in \Sigma^* \mid |w|_a = (|w|_b)!\}.$$

   Est-il possible de construire une grammaire hors-contexte qui génère L ? Si oui, donner une telle grammaire. Si non, démontrer cette impossibilité en prenant soin d'énoncer tous les théorèmes utilisés.
   Rappel : $|w|_a$ dénote le nombre de a dans le mot w et $n!$ correspond à la factorielle de n.

4. (a) Soit un automate à pile M. Montrer comment il est possible de construire une machine de Turing qui accepte le même langage que M.
   Suggestion : Utiliser une des extensions des machines de Turing, que l'on prendra soin de définir.

   (b) Est-il toujours possible de convertir une machine de Turing non déterministe en une machine de Turing déterministe en préservant le langage accepté ? Si oui, montrer comment procéder. Si non, justifier.

5. Etablir un tableau comparatif des propriétés de fermeture par les opérations d'union, d'intersection et de complémentation, pour les langages réguliers, les langages hors-contexte, les langages hors-contexte déterministes et les langages récursivement énumérables.

6. (a) Un naturel $y \in \mathbb{N}$ est dit *relativement premier* à un naturel $x \in \mathbb{N}$ si et seulement si leur plus grand commun diviseur est égal à 1. On définit alors la *fonction d'Euler* $\phi(x)$ comme étant le nombre de naturels $y > 0$ strictement inférieurs à x tels que y est relativement premier à x. La fonction d'Euler est-elle primitive récursive? Justifier.

   Remarque : Il n'est pas nécessaire de prouver le caractère primitif récursif de fonctions vues au cours théorique ou aux répétitions.

   (b) Montrer qu'il existe des fonctions calculables qui ne sont pas primitives récursives.

7. (a) Tout langage généré par une grammaire générale (de type 0) peut-il être accepté par une machine de Turing? Justifier.

   (b) Soit le problème prenant pour arguments deux machines de Turing déterministes T_1 et T_2 et consistant à tester si l'exécution de T_1 sur le mot vide prend strictement moins d'étapes que l'exécution de T_2 sur le mot vide. Ce problème est-il décidable? Justifier.

8. (a) Définir les classes de complexité P, NP et co-NP.

   (b) Le complément d'un langage appartenant à la classe P est-il aussi dans la classe P? Prouver ce résultat.

   (c) En utilisant la question (8.b), montrer que si NP $\neq$ co-NP, alors P $\neq$ NP.

   (d) A l'aide d'un diagramme, représenter les relations d'inclusion existant entre les langages réguliers, les langages hors-contexte et les classes P, NP, NPC, co-NP, PSPACE, NPSPACE, EXPTIME, R et RE. Existe-t-il dans ce diagramme une inclusion dont on ne sait pas encore à ce jour si elle est stricte?