Introduction à la Calculabilité

Examen de Deuxième Session, année 1999-2000

Livres fermés.
Durée : 3 heures 30 minutes.

Sur chacune de vos feuilles d'examen, indiquez votre nom, prénom et section.
Des réponses brèves mais précises sont souhaitées.

1. Soit le langage décrit par l'expression suivante :

$$\Big((a(ab)^* \cup ab)a\Big)^*$$

   (a) Ce langage est-il régulier ? Justifiez.
   (b) Donnez un automate non-déterministe acceptant ce langage.
   (c) Donnez un automate déterministe acceptant ce langage.

2. Donnez une procédure effective qui détermine si un langage régulier L_R est *universel*, c'est-à-dire, si $L_R = \Sigma^*$.

3. Soit le langage L décrit par l'expression suivante :

$$a^n b^m a^m b^n, \text{ avec } n,\ m > 0$$

   (a) Montrez à l'aide du théorème du gonflement que ce langage n'est pas régulier.
   (b) Ce langage est-il hors contexte ? Justifiez.
   (c) Ce langage est-il hors contexte déterministe ? Justifiez.

4. Démontrez le théorème du gonflement pour les langages hors-contexte.

5. Construisez une machine de Turing décidant le langage L décrit à la question **??**.
   Veillez à décrire le comportement de la machine en français <u>*et*</u> *sous forme de diagramme, ainsi qu'à limiter le nombre d'états.*

6. Montrez que la fonction *modulo*(x,y) qui renvoie x modulo y (*modulo*$(10, 7) = 3$) est primitive récursive.
   Vous pouvez vous servir des fonctions primitives récursives vues au cours théorique.
7. Montrez, (sans démontrer tous les détails), que les notions de fonction μ-récursive et de fonction calculable par machine de Turing sont équivalentes.
8. Soit les langages L_1 et L_2, respectivement acceptés par les machines de Turing M_1 et M_2. Montrez que le problème consistant à déterminer si $L_1 \subseteq L_2$ est indécidable.
9. (a) Quand peut-on dire que deux langages sont *polynomialement équivalents* ?
   (b) Définissez les classes P, NP et NPC.
   (c) Montrez que tout langage NP-complet est décidable.
   (d) Montrez que le problème de l'*ensemble indépendent*, décrit ci-après, est un problème NP-complet. Les données sont une constante J et un graphe défini par l'ensemble de ses sommets V et l'ensemble de ses arcs E. Le problème consiste à déterminer s'il y a un *ensemble indépendent* de taille $\geq J$ dans le graphe $G = (V, E)$, c'est-à-dire un sous-ensemble $V' \subseteq V$ tel que $|V'| \geq J$ et que pour tout $u, v \in V'$, l'arc u,v n'est pas dans E.
   Suggestion : Une transformation polynomiale du problème de la *couverture de sommets* (*vertex cover*) vers le problème de l'*ensemble indépendant* permet de démontrer que celui-ci est NP-dur. Pour rappel, le problème de la couverture de sommets a pour données un graphe et une constante J. Il consiste à déterminer s'il existe un sous-ensemble des sommets du graphe de taille au plus J qui *couvre* tous les arcs du graphe, à savoir tel que chaque arc a au moins une de ses extrémités dans l'ensemble.