Introduction à la Calculabilité
Examen de Seconde Session 1998

Livres fermés. Durée : 3 heures 30 minutes.

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

1. (a) Démontrer qu'un langage est régulier si et seulement si il est généré par une grammaire régulière.

   (b) Donner un automate fini non déterministe et un automate fini déterministe acceptant le langage défini par l'expression régulière suivante. Donner également une grammaire régulière générant ce langage.

   $$(b(a^* \cup c))^+$$

2. (a) Démontrer que le complément d'un langage régulier est un langage régulier. Donner aussi une procédure effective pour déterminer si un langage régulier L est *universel*, c'est-à-dire si $L = \Sigma^*$.

   (b) Énoncer les théorèmes du gonflement (deux pour les langages réguliers et un pour les langages hors-contexte).

   (c) Le langage $L = \{w \in \Sigma^* | N_a(w) > N_b(w)\}$, où $N_a(w)$ est le nombre de symboles a dans le mot w et $N_b(w)$ le nombre de symboles b, est-il régulier ? Justifier soit en présentant une grammaire du type requis, soit à l'aide des théorèmes du gonflement.

   (d) Le langage $L = \{w \in \Sigma^* | N_a(w) > N_b(w)\}$ est-il hors-contexte ? Justifier comme au point (c).

3. (a) Donner un automate à pile acceptant le langage des mots de la forme $(a^n b^* c^n) \cup (a \cup b)^* ab(a \cup b)^*$

   (b) Tout langage régulier est-il déterministe hors-contexte ? Justifier.

4. (a) Démontrer que tout langage accepté par une machine de Turing non déterministe est aussi accepté par une machine de Turing déterministe.

   (b) Donner une machine de Turing décidant le langage $L = \{a^{2i} b^i | i \geq 0\}$

5. (a) Existe-t-il des fonctions calculables qui ne sont pas primitives récursives ? Démontrer votre réponse.

   (b) Montrer que la fonction $arrondi(x)$ qui renvoie le plus grand multiple de 10 inférieur ou égal à son argument est primitive récursive.

6. (a) Étant donné une énumération $w_0, w_1, w_2, \ldots$ des mots et $M_0, M_1, M_2, \ldots$ des machines de Turing, démontrer que le langage $\overline{L_0} = \{w | w = w_i \wedge M_i \; accepte \; w_i\}$ est dans la classe RE.

7. (a) Qu'est-ce qui distingue les transformations polynomiales utilisées pour démontrer la NP-complétude des réductions utilisées pour démontrer l'indécidabilité ?

   (b) Montrer que tout problème NP-complet de taille n peut être résolu en un temps $\mathcal{O}(2^{p(n)})$ où $p(n)$ désigne un polynôme.

   (c) Montrer que si $L_1 \propto L_2$, alors

   - si $L_2 \in$ P alors $L_1 \in$ P,
   - si $L_1 \notin$ P alors $L_2 \notin$ P.

   (d) Énoncer le théorème de Cook. Le problème SAT est-il décidable ? Justifier.

   (e) Montrer que le problème de l'*ensemble dominant* défini ci-après est un problème NP-complet. On dispose d'un graphe dirigé $G = (V, E)$ et d'un entier $J \leq |V|$. Le problème de l'ensemble dominant consiste à déterminer s'il existe un *sous-ensemble dominant* $V' \subseteq V$ de taille J ou moins ($|V'| \leq J$); c'est-à-dire un ensemble $V' \subseteq V$ tel que pour tout sommet $u \in V - V'$, il y ait un sommet $v \in V'$ tel que $\{u, v\} \in E$.

   Suggestion: Une transformation polynomiale du problème de la *couverture de sommets* (*vertex cover*) vers le problème de l'ensemble dominant 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.

   L'idée de la transformation polynomiale est d'éliminer les sommets isolés, d'ajouter au graphe d'origine un sommet par arc et de relier chaque nouveau sommet aux extrémités de l'arc correspondant.