Introduction à la Calculabilité

Examen de janvier 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 les langages L_1, L_2 et L_3 définis sur l'alphabet $\Sigma = \{a, b, c\}$ de la manière suivante où $nb_x(w)$ est le nombre d'occurences du caractère x dans le mot w :
   - $L_1 = \{w \mid nb_a(w) = nb_b(w)\}$,
   - $L_2 = \{w \mid \forall \text{ préfixe } p \text{ de } w,\ |nb_a(p) - nb_b(p)| \leq 1\}$,
   - $L_3 = \{w \mid nb_a(w) + nb_b(w) = nb_c(w)\}$.

   (a) Démontrer que le langage L_1 n'est pas régulier.

   (b) Construire un automate fini déterministe acceptant L_2.

   (c) Le langage $L_1 \cap L_2$ est-il régulier ? Justifier.

   (d) Le langage L_3 est-il hors-contexte ? Justifier.

2. Construire une grammaire hors-contexte générant le langage $a^n\, b^m\, c^p$, avec $p > n + m$.
3. Construire une machine de Turing qui reconnaît le langage $L = \{w \mid nb_0(w) = nb_1(w)\}$ sur l'alphabet $\Sigma = \{0, 1\}$.
   Veillez à décrire le comportement de la machine en français et *sous forme de diagramme, ainsi qu'à limiter le nombre d'états.*
4. (a) Définir les notions suivantes :
   - une fonction *primitive-récursive*,
   - un prédicat *sûr*,
   - une fonction *μ-récursive*.

   (b) Soit la fonction $f\ : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ telle que

   $f(x, y) = z$ avec z étant le plus petit entier supérieur ou égal à $\begin{cases} \sqrt[y]{x} & si\, y > 0 \\ 0 & si\, y = 0 \end{cases}$

   Démontrer que f est primitive-récursive.
   Les résultats vus lors du cours théorique ne doivent pas être redémontrés.
5. Soit le langage $L = \{M \mid M \text{ est une machine de Turing et } \varepsilon \in L(M)\}$.
   Montrez que $L \notin R$ et que $L \in RE$.
   Suggestion : Réduction à partir du problème du langage universel LU.
6. Démontrer que tout langage accepté par une machine de Turing non-déterministe est aussi accepté par une machine de Turing déterministe.

7. Théorème de Cook :
   (a) Enoncer le théorème de Cook. Quel est son rôle dans la théorie de la NP-complétude ?
   (b) Quelles sont les variables propositionnelles utilisées pour démontrer le théorème de Cook. Quel est leur rôle ?
8. Soit le problème IA de l'inéquivalence entre automates finis non-déterministes (AFND). Une instance de ce problème peut s'énoncer comme suit :
   Soient deux AFND $\mathcal{A}_1$ et $\mathcal{A}_2$ ayant le même alphabet d'entrée Σ, est-ce que $\mathcal{A}_1$ et $\mathcal{A}_2$ reconnaissent des langages différents ?.

   (a) Donner de façon schématique un algorithme résolvant ce problème.
   (b) Que pouvez-vous dire au sujet de l'appartenance de ce problème aux classes de complexité/décidabilité RE, R et P ?
   (c) Bien que NP-dur, ce problème n'est pas NP-complet. Qu'est-ce que cela signifie ?
   (d) Que pouvez-vous dire de l'appartenance aux classes de complexité/décidabilité RE, R, P, NP et NPC (classe des langages NP-complets) de la version du problème IA dont les données seraient des automates déterministes.