Introduction à la Calculabilité

Examen du 29 août 2006

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. (a) Expliquez par un argument de cardinalité pourquoi il existe nécessairement des problèmes qui ne sont pas solubles par une procédure effective.

   (b) Soit l'alphabet $\Sigma = \{0, 1\}$. Donnez une expression régulière qui dénote le langage de tous les mots sur Σ qui comportent exactement 4 occurrences du symbole 0 ou qui commencent et se terminent par le symbole 1.

2. (a) Soit un alphabet $\Sigma = \{a, b, c, d\}$ et L le langage des mots sur Σ qui comportent exactement deux occurrences de b (pas nécessairement consécutives) ou qui se terminent par ab. Donnez

      i. un automate non déterministe qui accepte L ;

      ii. un automate déterministe qui accepte L ;

      iii. une grammaire régulière qui génère le complément de L.

   (b) Soit L un langage régulier construit sur l'alphabet Σ. Décrivez brièvement les procédures effectives permettant de décider si $L = \emptyset$ et si $L = \Sigma^*$.

3. Démontrez que le langage $L = \{a^m b^n c^{\max(m,n)} | m, n \geq 0\}$ défini sur l'alphabet $\Sigma = \{a, b, c\}$ n'est pas régulier en utilisant la deuxième version du théorème du gonflement.

4. (a) Soit l'alphabet $\Sigma = \{a, b, c\}$. Donnez un automate à pile qui accepte le langage des mots où le nombre d'occurrences de la lettre a est égal au nombre d'occurrences de la lettre b et où toute occurrence de la lettre a est suivie par deux occurrences de la lettre c.

   (b) Indiquez quelle condition un automate à pile doit satisfaire pour qu'il soit déterministe. Donnez un exemple de langage hors-contexte qui ne peut pas être accepté par un automate à pile déterministe.

5. (a) Définissez pour les machines de Turing les notions de *configuration*, *dérivation entre configurations*, *exécution*, *langage accepté*, *langage décidé*.

   (b) Un langage accepté par un automate à pile peut-il toujours être décidé par une machine de Turing ? Justifiez brièvement votre réponse.

   (c) Construisez une machine de Turing dont l'alphabet d'entrée est $\{0, 1\}$ et qui permute le premier et le dernier symbole de son mot d'entrée. Explicitez en quelques mots le rôle de chaque état de la machine construite.

6. (a) Définissez les notions de *fonction primitive récursive*, *prédicat primitif récursif* et *fonction $\mu-$récursive*.

(b) Montrez que la fonction $maxdiv(n)$ qui calcule le plus grand diviseur de n, autre que n lui-même, est primitive récursive.

7. (a) Soit un langage L et son complément $\bar{L}$. Donnez toutes les positions possibles des langages L et $\bar{L}$ par rapport aux classes de décidabilité **R** et **RE**. Justifiez.

   (b) Montrez par la technique de la réduction que le langage universel $LU = \{ < M, w > | M \text{ accepte } w\}$ est indécidable. Ce langage est-il partiellement décidable ?

8. (a) Définissez les problèmes du voyageur de commerce et du circuit hamiltonien et donnez une transformation polynomial du problème du circuit hamiltonien vers le problème du voyageur de commerce.

   (b) Soit la classe de complexité NP et un langage L appartenant à NP. Montrez qu'il existe une machine de Turing déterministe M et un polynôme $p(n)$ tel que M décide L et est de complexité en temps bornée par $2^{p(n)}$.

   (c) Il n'a pas encore été démontré que P $\neq$ NP. Est-il concevable que l'on arrive à montrer que certains problèmes NP-complets seraient solubles par un algorithme polynomial, alors que d'autres ne le seraient pas ?