

Introduction à la Calculabilité

Examen du 12 janvier 2004

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) Soit l'alphabet $\Sigma = \{a, b, c\}$ et L le langage des mots w tel que $|w|_a + |w|_c$ ($|w|_\alpha$ donne le nombre d'occurrences du symbole α dans w) n'est pas divisible par trois.
 - i. Construisez un automate fini qui accepte L .
 - ii. Construisez un automate fini qui accepte le complément de L .
 - iii. Donnez une grammaire qui génère le complément de L .
- (b) Soit R une expression régulière, donnez schématiquement la procédure qui permet de transformer R en un automate fini qui accepte le langage dénoté par R .
2. Soit les langage $L_1 = \{0^{n_1}1^{n_2}0^{n_3} \mid n_3 = n_1 + n_2 \wedge n_1, n_2, n_3 \geq 0\}$ et $L_2 = \{0^n 1^n \mid n \geq 0\}$ définis sur l'alphabet $\Sigma = \{0, 1\}$.
 - (a) Démontrez que le langage L_1 n'est pas régulier en utilisant le théorème du gonflement pour les langages réguliers.
 - (b) Le langage $L' = L_1 \cup L_2$ est-t-il hors-contexte? Si oui, donnez un automate à pile acceptant ce langage.
3. Soit une grammaire hors-contexte G , présentez schématiquement un algorithme qui détermine si $L(G) = \emptyset$
4. (a) Définissez pour les machines de Turing les notions de *configuration*, *dérivation entre configurations*, *exécution*, *langage accepté*, *langage décidé*.
 - (b) 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. Expliquez en quelques mots le rôle de chaque état de la machine construite.
5. (a) Démontrez que la fonction *SommeCarres* qui associe à un entier n la somme des carrés des entiers de 0 à n (ex: $SommeCarres(3) = 1 + 2^2 + 3^2$) est primitive récursive.
 - (b) Démontrez qu'il existe des fonctions *calculables* qui ne sont pas primitives récursives.

6. (a) Est-il possible de déterminer algorithmiquement si un programme écrit dans un langage de programmation usuel s'arrête pour des valeurs fixées de ses données? Justifiez.
- (b) Montrez que pour tout langage récursivement énumérable, il existe une procédure effective qui génère successivement tous les mots de ce langage.
7. (a) Définissez la notion de transformation polynomiale.
- (b) Définissez les problèmes du voyageur de commerce et du circuit hamiltonien et donnez une transformation polynomiale du problème du circuit hamiltonien vers le problème du voyageur de commerce.
8. (a) Donnez un algorithme déterministe qui résout le problème SAT. Quelle est sa complexité?
- (b) Le langage L_{SAT} des instances positives de SAT est-il dans la classe \mathbf{R} ? Justifiez.
- (c) **Bonus:** Le langage L_{SAT} est-il régulier? Justifiez.