Introduction à la Calculabilité

Examen du 13 janvier 2012

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) L'ensemble des fonctions de $\{0,1\}$ dans $\mathbb{N}$ est-il dénombrable ? Justifiez.

   (b) L'ensemble des fonctions de $\mathbb{N}$ dans $\{0,1\}$ est-il dénombrable ? Justifiez.

2. (a) Construisez un automate fini déterministe acceptant le langage L_1 dénoté par l'expression régulière $(a^*bc)^*$.

   (b) Construisez un automate fini déterministe acceptant le langage L_2 des mots construits sur l'alphabet $\{a,b,c\}$ qui contiennent un nombre impair de c.

   (c) Construisez un automate fini déterministe acceptant $L_1 \cap L_2$.

3. (a) Un mot x est un *facteur* d'un mot w s'il existe des mots y et z tels que $w = yxz$. Soit L un langage *régulier*. Le langage $\mathrm{Fact}(L)$ des facteurs des mots de L est-il nécessairement régulier[1] ? Justifiez.

   (b) En utilisant la théorie des automates finis, décrivez un algorithme permettant de trouver dans une chaîne de caractères w toutes les instances d'une expression régulière α.

4. (a) Enoncez le théorème du gonflement pour les langages hors-contexte.

   (b) Soit L le langage des mots construits sur l'alphabet $\{a,b\}$ qui commencent et se terminent par le même symbole, et qui possèdent le même nombre de symboles a que de symboles b. Construisez un automate à pile acceptant L.

   (c) i. Définissez les notions de *grammaire hors-contexte* et de *grammaire régulière*.

   ii. Est-il toujours possible de transformer une grammaire hors-contexte en une grammaire régulière ? Si oui, démontrez-le. Si non, donnez un contre-exemple et justifiez.

[1] Formellement, si L est construit sur un alphabet Σ, alors le langage $\mathrm{Fact}(L)$ est défini par $\mathrm{Fact}(L) = \{x \in \Sigma^* \mid (\exists y, z \in \Sigma^*)(yxz \in L)\}$.

5. (a) Définissez pour les machines de Turing les notions de *configuration*, *dérivation de configurations*, *exécution*, *langage accepté*, *langage décidé*.
   (b) Un langage décidé par une machine de Turing est-il toujours accepté par une machine de Turing ? Et inversement ?

6. Soient $n, m \in \mathbb{N}$ avec $m \neq 0$. On définit divceil(n, m) par la valeur de n/m arrondie à l'unité supérieure, i.e.,

$$\text{divceil}(n, m) = \left\lceil \frac{n}{m} \right\rceil .$$

   Par exemple, divceil$(6, 3) = 2$ et divceil$(8, 3) = 3$. La fonction divceil(n, m) est-elle primitive récursive et/ou μ-récursive ? Justifiez.
   Remarque : Il n'est pas nécessaire de démontrer le caractère primitif récursif des fonctions étudiées dans le cours théorique.

7. (a) Soient M une machine de Turing et x, u, y trois mots. On considère le problème consistant à déterminer si chacun des mots du langage xu^*y est accepté par M. Démontrez que ce problème est indécidable.
   Suggestion : utilisez LU.
   (b) Démontrez que si un langage est généré par une grammaire, alors il est accepté par une machine de Turing.

8. (a) Définissez la notion de temps d'exécution d'une machine de Turing non déterministe.
   (b) Définissez la notion de complexité en temps d'une machine de Turing non déterministe.
   (c) Définissez la classe de complexité NP.
   (d) Dans la démonstration du théorème de Cook, on montre l'existence d'une certaine transformation polynomiale.
      i. Quels sont les arguments de cette transformation et quel est son résultat ? En fonction de quel(s) argument(s) doit-elle être polynomiale ?
      ii. Quelle est l'utilité de ce théorème dans la théorie de la complexité ?