Introduction à la Calculabilité

Examen de Septembre 2001

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 L, un langage défini sur l'alphabet $\{a, b\}$. L se compose de tous les mots contenant au maximum un b **ou** au minimum 3 a.
   (a) Donnez une expression régulière représentant le langage L.
   (b) Donnez un automate fini non-déterministe acceptant le langage L.
   (c) Donnez un automate fini déterministe acceptant le langage $\Sigma^* \setminus L$.
2. Soit le langage $L = (abc)^n d^n$.
   (a) Énoncez les théorèmes du gonflement pour les langages réguliers.
   (b) Démontrez à l'aide d'un théorème du gonflement que le langage L n'est pas régulier.
   (c) Construisez un automate à pile acceptant L.
3. Soit la fonction $C(x, y)$ qui donne le coefficient binômial de x et y (choix de y objets parmi x objets différents).
   Pour rappel, $C(x, y) = \frac{x!}{y!\,(x-y)!}$. Ainsi, $C(4, 2) = \frac{4!}{2!\,(4-2)!} = 6$
   La fonction $C(x, y)$ est-elle *primitive récursive*? *μ-récursive*? Justifiez.
   Remarque : Il n'est pas nécessaire de redémontrer le caractère primitif récursif ou μ-récursif des fonctions vues au cours théorique.
4. Construisez une machine de Turing calculant la fonction $f(x, y, z) = 2 * (x + y + z)$, x, y et z étant représentés en notation unaire : la configuration initiale est :

$$(s_0,\ \epsilon,\ 0\underbrace{1..1}_{x}0\underbrace{1..1}_{y}0\underbrace{1..1}_{z}0\#\#\#\ldots)$$

   et la configuration finale doit être :

$$(s_f,\ \epsilon,\ 0\ \underbrace{1..1}_{2*(x+y+z)}\ 0\#\#\#\ldots).$$

   Veillez à décrire le comportement de la machine en français et sous forme de diagramme, ainsi qu'à limiter le nombre d'états.

5. *La non-calculabilité*
   (a) Définissez les classes R et RE.
   (b) Démontrez que R est strictement inclu dans RE.
   Remarque : Il n'est pas nécessaire de démontrer les résultats auxiliaires que vous utilisez.
   (c) Montrez que le problème consistant à déterminer si une machine de Turing donnée admet une exécution infinie est indécidable.
6. Un problème NP-complet est-il décidable ? Justifiez.
7. Montrez que le problème de la couverture d'ensemble décrit ci-dessous est NP-Complet.

   Données :
   - Un ensemble fini X d'éléments.
   - Une famille F de sous-ensembles s_i de X ($X = \bigcup_{s_i \in F} s_i$).
   - Un entier k.

   Problème :
   Existe-t-il une couverture d'ensemble $C \subseteq F$ couvrant X et dont la taille est inférieure à k ($\mid C \mid \leq k$ et $X = \bigcup_{s_i \in C} s_i$) ?

   Indication : Montrez qu'il existe une transformation polynomiale du problème de la couverture de sommets (Vertex Cover) vers le problème de la couverture d'ensemble.