Université de Strasbourg **UFR de Mathématiques et Informatique**
Master ILC et ISI **novembre 2011**

Complexité et Calculabilité

Contrôle Continu n°1

Durée : 1 heure
Responsable : Prof. Christian RONSE
Tous documents en papier autorisés mais non partagés
Calculettes inutiles
Téléphones et appareils électroniques éteints et rangés dans un sac fermé

Justifiez soigneusement vos réponses

(1) Machines de Turing.
Donner des machines de Turing décidant les langages suivants sur $\Sigma = \{a, b, c, d\}$:

(i) $\{uv \mid u \in \{a, b\}^*,\ v \in \{c, d\}^*,\ |u| = |v|\}$ (ensemble des mots qui sont concaténation de deux mots de même longueur, le premier formé des lettres a, b, le second formé des lettres c, d).

(ii) $\{a^m b^n c^p d^q \mid m, n, p, q \in \mathbb{N}\}$.

En déduire une machine de Turing décidant le langage

(iii) $\{a^m b^n c^p d^q \mid m, n, p, q \in \mathbb{N},\ m + n = p + q\}$.

Ces machines de Turing peuvent être décrites par des assemblages de machines élémentaires vues en cours, et il faudra en expliquer brièvement le fonctionnement.

(2) Grammaire.
Donner une grammaire engendrant le langage $\{a^{2^n} \mid n \in \mathbb{N}\}$ (ensemble des mots formés de la lettre a dont la longueur est une puissance de 2).

(3) Fonction μ-récursive.
Soit $f : \mathbb{N} \to \mathbb{N}$ une fonction μ-récursive bijective. Montrer que la bijection inverse $g : \mathbb{N} \to \mathbb{N}$ définie par $g(m) = n \Leftrightarrow f(n) = m$ pour $m, n \in \mathbb{N}$, est une fonction μ-récursive. Plus concrètement, donner un algorithme pour calculer g au moyen de f, des fonctions de base, des opérations de composition et récursion, et de la minimisation de fonctions minimisables.