Université de Strasbourg **UFR de Mathématiques et Informatique**
Master 1 ILC et ISI **décembre 2013**

Complexité et Calculabilité

Contrôle Continu n°3

Durée : 40 minutes
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) Réduction polynomiale.
(A) Soient L_1 et L_2 deux langages sur l'alphabet Σ, et soient $\overline{L_1} = \Sigma^* \setminus L_1$ et $\overline{L_2} = \Sigma^* \setminus L_2$ leurs complémentaires.

(i) S'il existe une réduction polynomiale de L_1 vers L_2, qu'y a-t-il entre $\overline{L_1}$ et $\overline{L_2}$?

(ii) S'il existe une réduction polynomiale de L_1 vers L_2, et L_2 est de complexité NP, que peut-on dire de la complexité de L_1 ?

(B) Soit L_0 un langage NP-complet sur Σ. Supposons qu'on puisse démontrer que son complémentaire $\overline{L_0}$ est de complexité NP.

(iii) Etant donné un langage L de complexité NP, prouver qu'alors son complémentaire $\overline{L}$ l'est aussi.

(iv) Etant donné un langage L NP-complet, prouver qu'alors son complémentaire $\overline{L}$ l'est aussi.

(2) Homomorphisme non effaçant.
Soit Σ un alphabet (fini), et soit $f : \Sigma \to \Sigma^+$ une application associant à tout caractère $c \in \Sigma$ un mot *non-vide* $f(c)$. On en déduit alors l'*homomorphisme* $g : \Sigma^* \to \Sigma^*$ appliquant f à chaque caractère d'un mot, c.-à-d.

$$g(\varepsilon) = \varepsilon \qquad \text{et} \qquad g(c_1 \cdots c_n) = f(c_1) \cdots f(c_n) \quad (n > 0) \ .$$

Comme $f(c) \neq \varepsilon$ pour tout $c \in \Sigma$, cet homomorphisme est *non effaçant*.

Donner un algorithme non-déterministe polynomial qui décide l'ensemble des $g(w)$, $w \in \Sigma^*$. Plus précisément, à partir d'un mot $m \in \Sigma^*$, il peut produire, s'il existe, un mot $w \in \Sigma^*$ tel que $g(w) = m$ (mais il peut aussi échouer), et il échouera toujours s'il n'existe pas.