Université de Strasbourg **UFR de Mathématiques et Informatique**

Master 1 ILC et ISI **23 novembre 2015**

Complexité et Calculabilité

Contrôle Continu n°2

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) Grammaire.

Soit Σ un alphabet. Donner une grammaire calculant la fonction $\Sigma^* \to \Sigma^*$ qui dans un mot w garde la première occurrence de chaque caractère et supprime toutes les occurrences suivantes, par exemple

$$abracadabra \mapsto abrcd$$
$$repeter \mapsto rept$$
$$babebibobu \mapsto baeiou$$

NB. La solution peut s'inspirer de l'algorithme suivant : le curseur étant initialement à gauche du mot, tant que le curseur n'est pas arrivé à droite du mot, faire : déplacer le curseur d'une place à droite, lire le caractère puis (à l'aide d'un second curseur) supprimer toutes les répétitions du caractère à droite du curseur.

(2) Récursion et minimisation.

Soit $P : \mathbb{N} \to \{0, 1\}$ un prédicat récursif primitif. Supposons qu'il existe une infinité de $n \in \mathbb{N}$ tels que $P(n) = 1$. Définissons la fonction $f_P : \mathbb{N} \to \mathbb{N}$ en posant, pour tout $n \in \mathbb{N}$, $f_P(n)$ égale au n-ième plus petit $m \in \mathbb{N}$ tel que $P(m) = 1$ (on numérote à partir du 0-ième pour le plus petit). Par exemple, si $P(n) = 1$ pour n premier, on a $f_P(0) = 2$, $f_P(1) = 3$, $f_P(2) = 5$, $f_P(3) = 7$, etc. Montrer comment calculer f_P au moyen de la récursion primitive et de la minimisation de fonction minimisable. Que peut-on en déduire concernant f_P ?