TD2

Grammaires et fonctions numériques

1 Grammaires

a-On considère le langage $L_2 = \{a^{n^2} | n \in \mathbb{N}\}$. Définissez une grammaire G telle que $L(G) = L_2$.

b- On rappelle qu'une grammaire G calcule une fonction f si et seulement si pour tout mot $w \in \Sigma_0^*$, on a $SwS \Rightarrow^* v$ si et seulement si $f(w) = v$. Définissez un grammaire G_i calculant la fonction f_i dans chacun des cas suivants :

- $f_1(I^n) = I^{n+1}$
- $f_2(n) = n + 1$ (les entiers seront représentés en binaires)
- $f_3(w) = ww$
- $f_4(n, m) = n + m$

2 Fonctions récursives au sens de Church : μ-récursivité

a- Soit $f : \mathbb{N} \to \mathbb{N}$ une fonction primitive récursive, on note F la fonction définie à partir de f par : $F : \mathbb{N} \to \mathbb{N}$ et $F(n) = f(f(...f(n)...))$ où il y a n compositions de la fonction f. Montrez que la fonction F est aussi une fonction primitive récursive.

b- Montrez que la fonction factorielle est une fonction primitive récursive.

c- Faites de même pour la fonction *booléenne* qui teste la primalité d'un entier.

d- Soit f une fonction μ-récursive qui est, de plus, une bijection de $\mathbb{N}$ sur $\mathbb{N}$. Montrez que la fonction inverse f^{-1} est également μ-récursive.