Arbres, induction et récursion

1. On appelle ici arbre (binaire) tout terme de la syntaxe FO associée à la signature $B := (cons_0 : 0; cons_2 : 2)$.

   a) Définir par récursion pour chaque tel arbre :

   son nombre de feuilles ; son nombre de noeuds ; sa hauteur ; sa largeur.

   b) Enoncer et démontrer des inégalités reliant ces quantités.

   c) Définir par récursion la transformation s qui reçoit un tel arbre et y échange partout les sous-arbres droit et gauche. Démontrer que s^2 est l'identité.

   d) Définir par récurrence une injection de l'ensemble des entiers dans cet ensemble d'arbres.

2. On appelle ici arbre (ternaire) tout terme de la syntaxe FO associée à la signature $T := (cons'_0 : 0; cons_3 : 3)$ et on note T l'ensemble de ces arbres ternaires.

   a) Définir par récursion pour chaque arbre :

   son nombre de feuilles ; son nombre de noeuds ; sa hauteur ; sa largeur.

   b) Définir beaucoup d'injections de l'ensemble $\hat{B}$ dans l'ensemble $\hat{T}$ respectant la hauteur. Démontrer que l'une de vos injections respecte la hauteur.

   c) Définir par récursion une injection de $\hat{T}$ dans $\hat{B}$. Montrer par induction que cette injection en est bien une.

3. On appelle ici arbre (quelconque) tout terme de la syntaxe FO associée à la signature $A := (cons_i : i)_{i \geq 0}$.

   a) Définir par récursion l'ensemble des chemins et l'ensemble des feuilles d'un élément de $\hat{A}$.

   b) Définir par récursion une injection de $\hat{A}$ dans $\hat{B}$.Justifier.

4. On appelle ici arbre binaire étiqueté dans $\mathbf{N}$ tout terme de la syntaxe FO associée à la signature $E := \{nil_n : 0 | n \in \mathbf{N}\} \amalg \{cons_n : 2 | n \in \mathbf{N}\}$.

   a) Définir par récursion l'arbre ("nu") associé à un arbre étiqueté.

   b) Définir par récursion la somme des étiquettes d'un arbre étiqueté.

   c) Définir par récursion une injection de l'ensemble de ces arbres étiquetés dans celui des arbres binaires (nus).