

TP # 3 Programmation Logique L3-UPN 2009/2010

(Les arbres binaires)

Un arbre binaire est composé d'un noeud racine et de deux noeuds fils qui définissent récursivement des sous-arbres binaires.

En Prolog, nous allons représenter un arbre binaire par un terme « $bt(X,FG,FD)$ », où X est le noeud racine, FG est le sous-arbre binaire associé au fils gauche de X , et FD est le sous-arbre binaire associé au fils droit de X . Le noeud vide est défini par le terme « nil ». Donc, une noeud feuille X est définie par le terme « $bt(X,nil,nil)$ ».

Quel arbre binaire est défini par le terme :

$T = bt(a, bt(b,nil,nil), bt(c, bt(d, bt(f, nil, nil), bt(g, nil, nil)), nil)) ?$

Exercice 1. Écrire un prédicat *istree/1* qui est vérifié si et seulement si son argument est un terme représentant un arbre binaire. Exemple :

$?- istree(bt(a, bt(b,nil,nil), bt(c, bt(d, bt(f, nil, nil), bt(g, nil, nil)), nil))) .$

Ye s

Exercice 2. Écrire un prédicat *nnodes/2* qui compte le nombre de noeuds d'un arbre binaire donné. Exemple :

$?- nnodes(bt(a, nil, bt(b, nil, nil)), N).$

$N = 2$

Exercice 3. Définir un prédicat qui vérifie l'appartenance d'un élément (noeud) à un arbre binaire.

Exercice 4. Définir un prédicat qui construit la liste des feuilles d'un arbre binaire donné.

Exercice 5. Écrire un prédicat *profondeur(Tree, Liste)* qui construit la liste en profondeur d'abord à partir de la racine de l'arbre binaire $Tree$.