

# Programmation Logique et Fonctionnelle - TP2

## CAML

### 1 Consignes

Lorsqu'il vous est demandé d'implanter une fonction, cela suppose une validation expérimentale de votre code, en appliquant la fonction à des jeux de données pertinents et en vérifiant l'exactitude du résultat obtenu. Si nécessaire, vous pouvez décomposer la fonction demandée en plusieurs fonctions.

Vous devez essayer de déterminer la signature de chacune des fonctions à implanter et vérifier *ensuite* qu'elle correspond à celle inférée par le compilateur. En cas de différence, vous devez identifier et comprendre votre erreur.

Les solutions de certains exercices donnés ici sont certainement disponibles quelque part sur Internet, mais cela ne vous apporterait strictement aucune compétence de saisir et d'exécuter un code que vous ne comprenez pas, ou même que vous croyez comprendre, mais sans avoir su le construire par vous-même. Si vous êtes vraiment bloqués, demandez plutôt des indices à votre enseignant de TP.

### 2 Arbres binaires

#### 2.1

Définissez le type `bintree`, représentant un arbre binaire, sachant qu'un tel arbre est soit une feuille notée  $L$ , soit un nœud, noté  $N$  et ayant trois attributs qui sont un entier (appelé étiquette du nœud), et deux arbres binaires (appelés fils droit et fils gauche).

#### 2.2

On appelle *chemin* d'un arbre  $A$  toute suite de nœuds telle que le premier soit la racine de  $A$ , le dernier soit une feuille, et tout nœud de la suite excepté le premier soit un fils du nœud précédent. On définit le *poids* d'un chemin d'un arbre comme la somme des valeurs des étiquettes des nœuds de ce chemin.

Implantez une fonction qui, appliquée à un arbre binaire  $t$  et un entier  $x$ , retourne une valeur Booléenne égale à `true` si et seulement si il existe dans  $t$  un chemin dont le poids est supérieur à  $x$ .

#### 2.3

Implantez une fonction qui, appliquée à un arbre binaire, retourne la liste des étiquettes des nœuds de cet arbre parcouru en ordre infixe.

Le parcours d'un arbre binaire en ordre infixe consiste, pour chaque nœud  $N$ , à parcourir d'abord le fils droit en ordre infixe, puis à lire l'étiquette de  $N$ , puis à parcourir le fils gauche en ordre infixe.

#### 2.4

Implantez une fonction qui, appliquée à deux arbres binaires  $t$  et  $u$ , retourne un Booléen valant `true` si et seulement si les deux arbres sont identiques (i.e., même structure et mêmes étiquettes).

#### 2.5

On associe à chaque feuille d'un arbre binaire un entier, appelé *rang*. La feuille située la plus à gauche a pour rang 1, et les autres sont numérotées de telle sorte qu'en parcourant l'arbre en ordre infixe, on visite ses feuilles dans l'ordre croissant de leurs rangs.

Implantez une fonction qui, appliquée à deux arbres binaires  $t$  et  $u$  et à un entier  $i$ , retourne l'arbre obtenu en remplaçant dans  $t$  la feuille de rang  $i$  par l'arbre  $u$ . S'il n'existe pas de feuille de rang  $i$  dans  $t$ , la fonction doit retourner  $t$ .

### 3 Arbres généraux

Reprendre les exercices sur les arbres binaires en les adaptant aux arbres généraux. Dans un arbre général, chaque nœud possède un nombre quelconque de fils. Une feuille est simplement un nœud n'ayant aucun fils. On utilisera une liste pour représenter les fils d'un nœud.