

# TD/TP PFL

## Création et gestion de nouveaux types

### **Exercice 1 :**

- Définir un type pour les intervalles (un intervalle est un couple de nombres réels ).
- Ecrire deux fonctions prenant en argument un intervalle et retournant respectivement la borne inférieur et celle supérieure.
- Ecrire une fonction permettant de vérifier qu'un intervalle n'est pas vide.

### **Exercice 2 :**

- Définir un type pour les nombres rationnel (un nombre rationnel peut s'écrire sous la forme  $a/b$  où  $a$  et  $b$  sont des entiers).
- Ecrire deux fonctions permettant de calculer la somme et le produit de deux nombres rationnels.

### **Exercice 3 :**

- Définie un type pour les nombres naturels en utilisant l'arithmétique de Peano.
- Ecrire deux fonction permettant de calculer la somme et le produit de deux entiers naturels.
- Définir un type pour les listes d'entiers naturels (il est interdit d'utiliser le type list existant).
- Ecrire une fonction permettant d'additionner les éléments d'une liste d'entiers naturels.

**Exercice 4 :**

- Définir un type avec paramètres  $(a', b')$  *liste* permettant d'avoir des listes avec des valeurs de deux types différents.
- Dans le cas d'une liste du type  $(int, float)$  *liste*, écrire une fonction permettant de calculer la somme de tous les éléments.
- Ecrire une fonction retournant le dernier élément d'une liste du type  $(a', b')$  *liste*.
- Ecrire une fonction inversant une liste du type  $(a', b')$  *liste*.

**Exercice 5 :**

- Définissez le type 'a arbre d'arbres binaires représentant des noeuds dont les étiquettes sont d'un type quelconque (mais toutes du même type).
- Ecrivez une fonction permettant d'additionner les éléments d'un arbre binaire.
- Ecrivez une fonction réalisant la projection verticale d'un arbre binaire sous-forme de liste.