TD de programmation fonctionnelle et logique

TD 3 : récursivité

Suite de Fibonacci

La suite de Fibonacci est définie comme suit :

$$\text{Fib}(n) = \begin{cases} 1 \text{ si } n = 0 \\ 1 \text{ si } n = 1 \\ \text{Fib}(n-1) + \text{Fib}(n-2) \text{ sinon.} \end{cases}$$

1. Écrire une fonction récursive qui calcule $\text{Fib}(n)$.
2. Écrire une fonction récursive qui calcule, pour $n > 0$, $(\text{Fib}(n), \text{Fib}(n-1))$. L'utiliser pour écrire une fonction calculant $\text{Fib}(n)$.

Calcul de (cos(nx), sin(nx))

Écrire une fonction qui prend en entrée un entier `n` et une paire de valeurs réelles qui sont en fait les valeurs du cosinus et du sinus d'un certain angle `x`, et qui renvoie la paire `(cos(nx), sin(nx))`. Autrement dit, le deuxième argument de la fonction est une paire `(a,b)` telle que `a = cos x` et `b = sin x` . Le schéma de calcul doit bien évidemment être récursif.

On pourra se servir des formules de trigonométrie suivantes :

```
cos(nx) = cos((n-1)x) cos(x) - sin((n-1)x) sin(x)
sin(nx) = sin((n-1)x) cos(x) - cos((n-1)x) sin(x)
```

Fonctions classiques sur les ensembles triés

On considère des ensembles triés représentés par des listes. Écrire des fonctions réalisant les opérations suivantes :

1. Intersection ;
2. Union ;
3. Différence ;
4. Différence symétrique ;
5. Produit cartésien (résultat donné dans l'ordre lexicographique).