Quadtree

Tout le monde

February 10, 2019

Contents

1 Implémentation : comparaison entre structure récursive et représentation en liste

1.1 Structure récursive

Une représentation qui semble la plus naturelle est d'implémenter une structure avec auto-référence. En effet en s'inspirant de la définition inductive on peut voir un arbre binaire comme une structure contenant essentiellement 3 champs : une clef et deux arbres binaires, ses sous-arbres gauche et droit. En terme d'implémentation cela dépend du langage utilisé.
En C cela passera par la création d'un type spécifique arbre avec en variable des pointeurs vers des variables de ce même type, en plus d'une variable clef de type entier par exemple.
En suivant le paradigme de la programmation objet, on créé une classe arbre ayant deux attributs de type arbre en plus de l'attribut clef (et d'autres attributs si nécessaires). La souplesse de Python sur les types rend la création de la classe particulièrement aisée mais il faudra être vigilant sur l'utilisation. On pourra déclarer la classe comme ceci :

```
class ArbreBinaire():
    def __init__(self,clef,gauche = None ,droite = None):
        """ Anything x ArbreBinaire x ArbreBinaire -> ArbreBinaire
        Constructeur d'arbre"""
        self.clef = clef
        self.gauche = gauche
        self.droite = droite
```

Exemple d'instanciation :

```
Feuille1 = ArbreBinaire(1)
Feuille2 = ArbreBinaire(2)
Arbre = ArbreBinaire(3,Feuille1,Feuille2)
```

L'arbre créé est :