

Arborescences

Chapitre 17

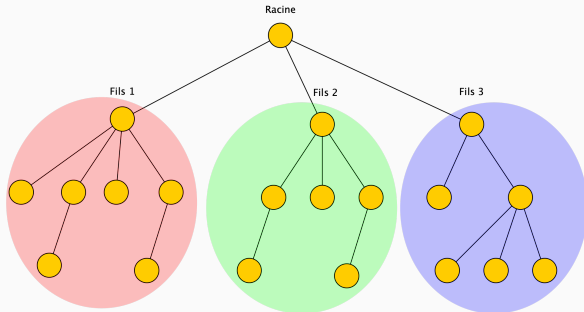
NSI2

14 mars 2022

Définition

On appelle **arborescence** un ensemble *non vide* de nœuds tel que

- un de ces nœuds est appelé **racine** de l'arborescence;
- les autres nœuds sont partagés en n sous-ensembles distincts qui sont eux-mêmes des arborescences;
- la racine est reliée aux n racines de ces arborescences, qu'on appelle ses **fls**.



Non :

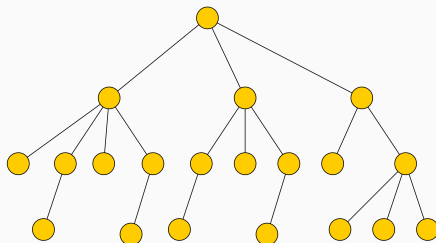
- il n'y a pas d'arborescence vide tandis que l'arbre binaire vide existe;
- dans un arbre binaire, il y a un ordre sur les fils (le gauche et le droit), dans une arborescence, il n'y a pas d'ordre sur les fils.

Si le concept d'arborescence ne généralise pas celui d'arbre, il y a tout de même beaucoup de points communs.

Taille et hauteur

Les définitions sont les mêmes que pour les arbres binaires :

- la **taille** d'une arborescence est le nombre de ses nœuds;
- la **hauteur d'un nœud** est le nombre d'arêtes le reliant à la racine;
- la **hauteur d'une arborescence** est la plus grande des hauteurs de ses nœuds.



Arbre de taille 20 et de hauteur 3.

Une implémentation possible

```
class NodeTS:
    def __init__(self, value):
        self.value = value
        self.children = []
```

Un nœud est une instance de la classe **NodeTS** (TS pour *Tree Structure*, arborescence en Anglais).

Il possède un attribut **value** pour stocker une donnée, et un attribut **children** pour stocker la liste de ses fils, eux-mêmes instances de la classe **NodeTS**.

On retrouve la notion de parcours préfixe :

- traiter d'abord le nœud courant;
- traiter ensuite les fils du nœud (peu importe l'ordre).

On retrouve également la notion de parcours postfixe :

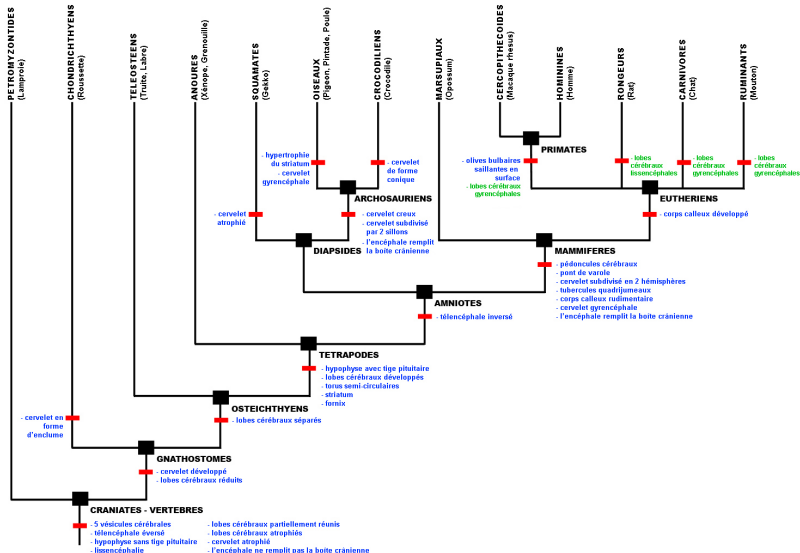
- traiter d'abord les fils du nœud (peu importe l'ordre);
- traiter ensuite le nœud courant.

Il n'y a pas de parcours infixé.

Le système de fichiers d'un lecteur logique WINDOWS est une arborescence :

- la racine est... la racine du lecteur : **C:** par exemple
- chaque nœud est un dossier contenant un ensemble de fichiers, et avec des dossiers fils.

NSI2



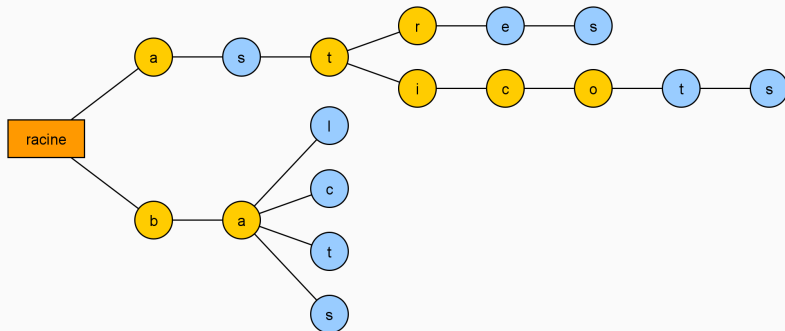

```
<recette difficulté="facile">
  <titre>Crêpes sucrées</titre>
  <temps>1h</temps>
  <note>pour 10 crêpes</note>
  <ingredients>
    <i q="200g">farine</i>
    <i q="40g">sucre</i>
    <i q="2">œufs</i>
    <i q="40cl">lait</i>
  </ingredients>
  <etapes>
    <e>mélanger les ingrédients solides</e>
    <e>ajouter le lait</e>
    <e>laisser reposer</e>
    <e>cuire sur une poêle beurrée</e>
  </etapes>
</recette>
```

C'est un langage à balises pour décrire des données structurées :

- une recette, comme dans l'exemple précédent;
- la valeur des attributs d'un objet...

Un document XML est une arborescence. Toute portion *complète du code*, c'est-à dire :

- une portion commençant par une balise et terminant par la balise fermante correspondante, correspond à un nœud;
- une portion sans balises.



On part d'une racine « vide ». Chaque nœud contient une lettre et un booléen indiquant si cette lettre est la fin d'un mot ou non.

Ici le trie contient les mots as, astre, astres, asticot, asticots, bal, bac, bat et bas.

Elle diffère légèrement de celle d'une arborescence traditionnelle.

Nous verrons cela en exercice.