

Chapitre 14 – Les arbres n-aires

Structure arborescente.

Un arbre est un ensemble d'éléments appelés nœuds (dont un se singularise comme la racine) liés par une relation hiérarchique. Un nœud, comme tout élément d'une liste, peut-être de n'importe quel type.

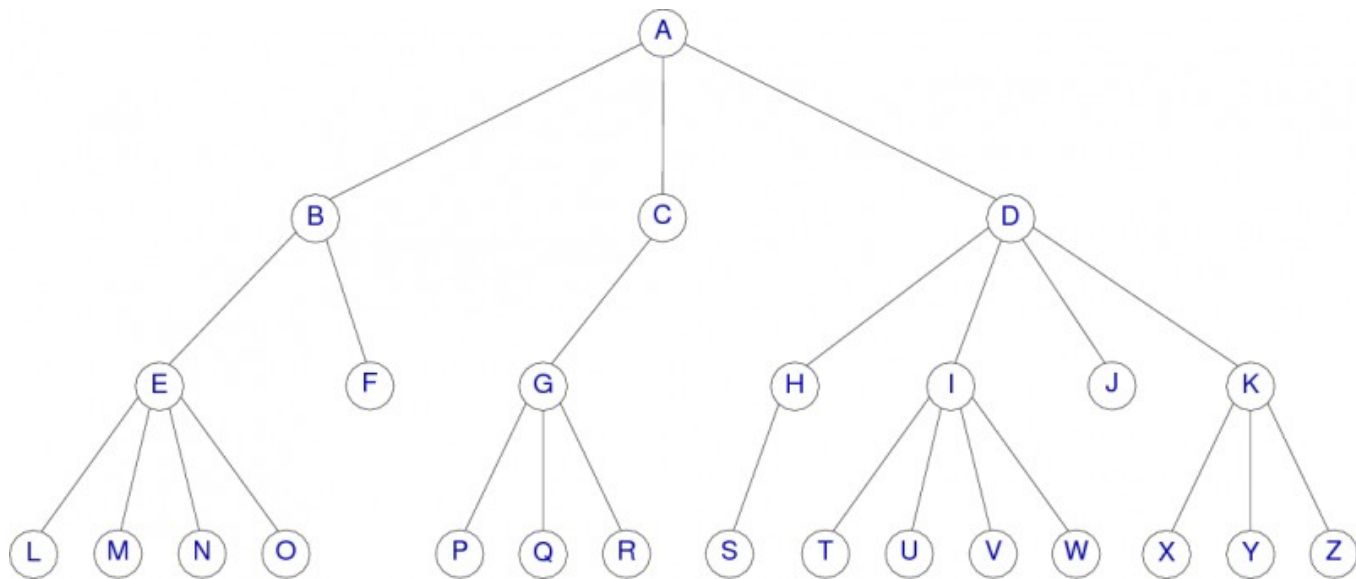
Définition : arbre

- Un nœud unique est un arbre. Il est la racine de l'arbre.
- Si n est un noeud et $A_1, \dots, A_k$ sont des arbres de racine respective $n_1, \dots, n_k$, on peut construire un nouvel arbre en associant comme parent unique aux noeuds $n_1, \dots, n_k$ le noeud n . Dans cet arbre, n est la racine et $A_1, \dots, A_k$ sont les sous-arbres de cette racine. Les noeuds $n_1, \dots, n_k$ sont appelés les fils de n .

Même vocabulaire que pour les arbres binaires.

L'arbre vide (noté Λ) est un arbre sans noeud.

Exemple



TAD ArbreNaire

Sorte : ArbreNaire

Utilise : Noeud, Forêt (liste d'ArbreNaire), Booléen, Élément

Opérations :

Créer-arbre-vidé() $\rightarrow$ ArbreNaire

Créer(Noeud, Forêt) $\rightarrow$ ArbreNaire

Racine(ArbreNaire) $\rightarrow$ Noeud

Père(Noeud, ArbreNaire) $\rightarrow$ Noeud

Fils(ArbreNaire) $\rightarrow$ Forêt

Est-vidé(ArbreNaire) $\rightarrow$ Booléen

Affecter-val(Noeud, Élément) $\rightarrow$ Noeud

Val(Noeud) $\rightarrow$ Élément

Parcours en profondeur

Ordre : A, B, E, L, M, N, O, F, C, G, P, Q, R, D, H, S, I, T, U, V, W, J, K, X, Y, Z

```
Procédure profGauche (E A : ArbreNaire)
Var lf : Liste d'ArbreNaire
    pfils : place
Début
    Si est-vide(A)
        Alors terminaison
        Sinon lf ← fils(A)
            pfils ← tête(lf)
            TantQue pfils ≠ fini(lf) Faire
                profgauche(contenu(pfils))
                pfils ← succ(pfils)
            FinTantQue
        FinSi
Fin
```

Parcours en largeur

Ordre : A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z

Procédure parcoursLargeur(E A : ArbreNaire)

Var lf : Liste d'ArbreNaire

 f : file d'ArbreNaire

 sa : ArbreNaire

 pfils : place

Début

 f ← Créer-file-vide()

 Ajouter(f,A)

TantQue ¬est-vide(f) Faire

 sa ← premier(f)

 Retirer(f)

 lf ← fils(A)

 pfils ← tête(lf)

TantQue pfils ≠ fini(lf) Faire

 ajouter(f,contenu(pfils))

 pfils ← succ(pfils)

FinTantQue

FinTantQue

Fin
