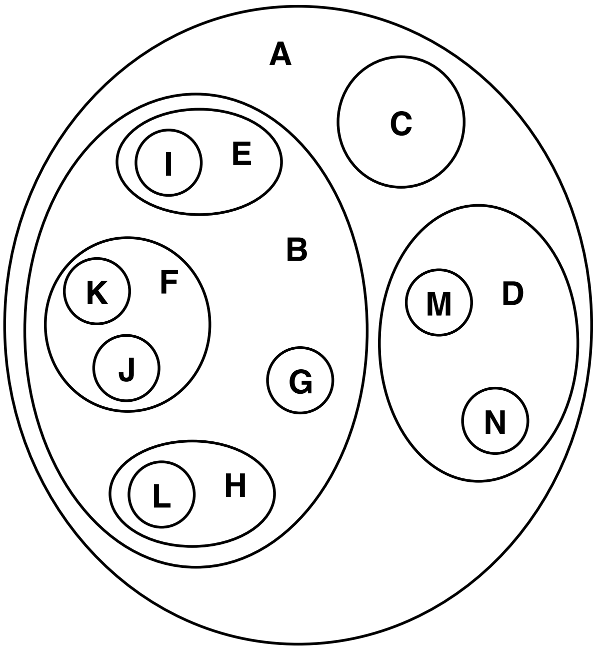
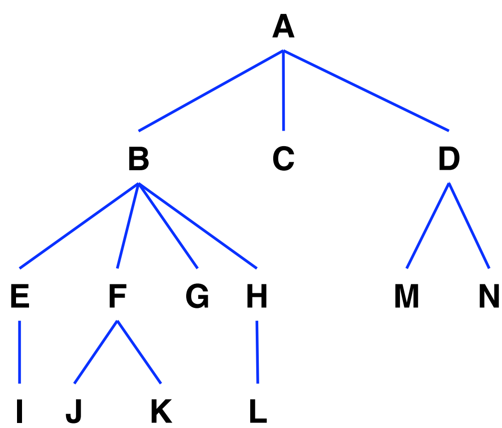
Algorithmique & programmation

Chapitre 5 : Arbres
Définitions

Exemple

- Représentation d'une relation d'inclusion

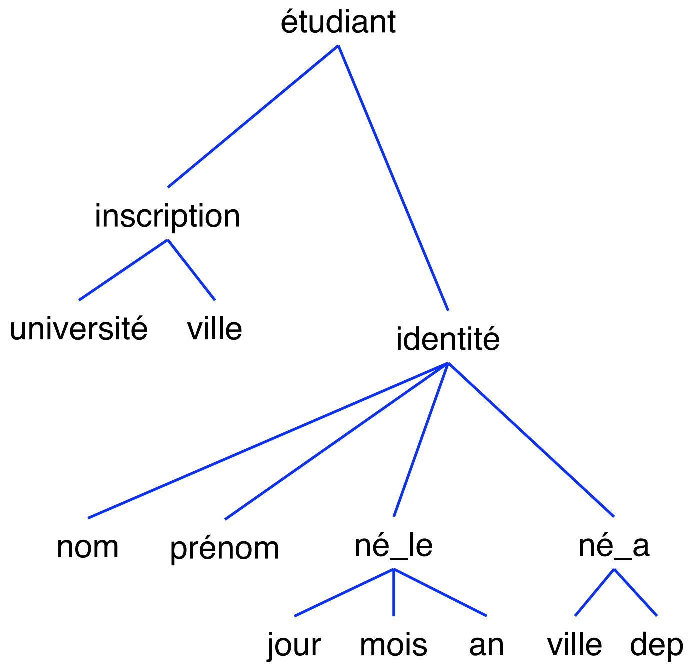




Exemple

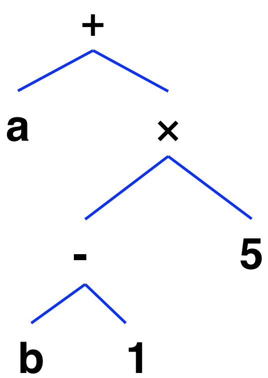
Types structurés

type tdate = structure
jour , mois , an : entier ;
fin ;
type tlieu = structure
ville : chaîne30 ;
departement : chaîne20 ;
fin ;
type tidentite = structure
nom, prenom : chaîne30 ;
né_le : tdate ;
né_à : tlieu ;
fin ;
type tetablissement = structure
universite : chaîne30 ;
ville : chaîne30 ;
fin ;
type étudiant = structure
inscription : tetablissement ;
identite : tidentite ;
fin ;

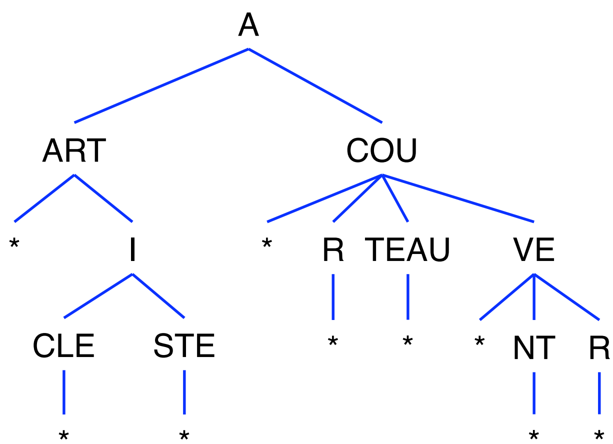


Exemples

- Expression arithmétique
  - a + (b - 1) × 5

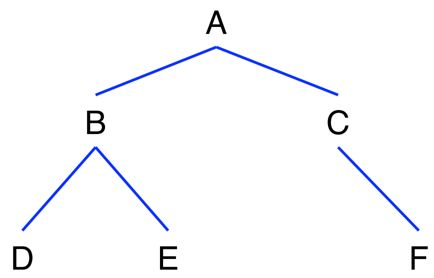


- Dictionnaire arborescent

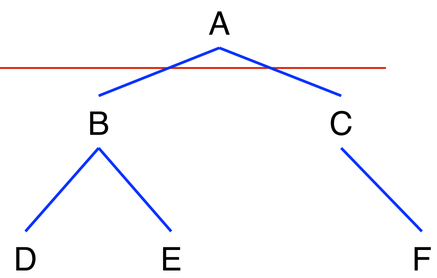


Définitions

- **Un arbre est**
  - soit vide
  - soit constitué d'un **sommet** (ou **nœud**) auquel sont chaînés un ou plusieurs arbres
- **sommet** (ou **nœud**)
  - A, B, C, D, E, F
- **fils**
  - de A : B, C
  - de B : D, E
  - de C : F
  - de D, E, F : aucun

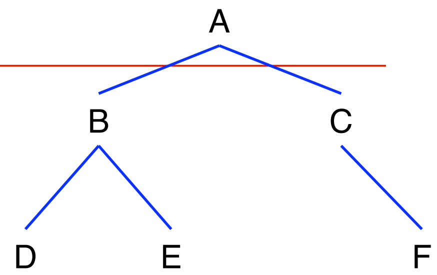


Définitions



- **père**
  - de B et de C : A
  - de G : C
- **feuille** : noeud sans fils (nœud **terminal**)
  - D, E et G
- **racine** de l'arbre : noeud sans père
  - A
- Propriété
  - Tous les noeuds, sauf la racine, n'ont qu'un seul père

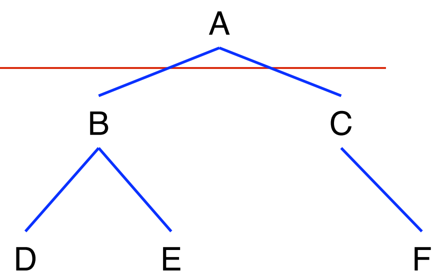
Définitions



- Notion de **sous-arbre**

- **Arbre n-aire**
  - chaque nœud a au plus n fils
  - si n = 2 : arbre binaire
- **Arbres binaire**
  - pour un nœud quelconque, on parle de
    - fils gauche
    - fils droit
    - sous arbre gauche
    - sous arbre droit

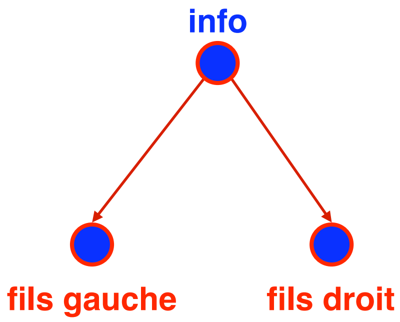
Définitions



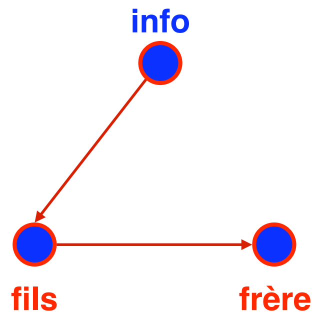
- Définition récursive d'un arbre binaire
  - soit vide,
  - soit composé d'un nœud auquel sont chaînés un sous-arbre gauche et un sous-arbre droit.
- un sous-arbre est un arbre binaire.
- Représentation parenthésée
  - A (B (D ((), ()), E ((), ())), C ((), F ((), ())))
    - où () = arbre vide.
  - A (B (D, E), C (, F))
    - en supprimant les arbres vides

Le type nœud

- Deux représentations possibles




```
type    pointeur = ↑noeud ;
        noeud = structure
                gauche : pointeur ;
                info : t ;
                droite : pointeur ;
            fin ;
```

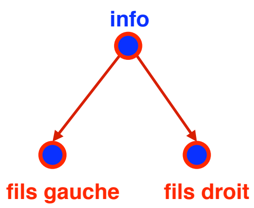



```
type    pointeur = ↑noeud ;
        noeud = structure
                info : t ;
                fils : pointeur ;
                frère : pointeur ;
            fin ;
```

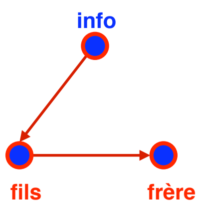

Le type nœud (ada)

- Deux représentations possibles

```
type Tr_Noeud;
type Ta_Noeud is access Tr_Noeud;
```

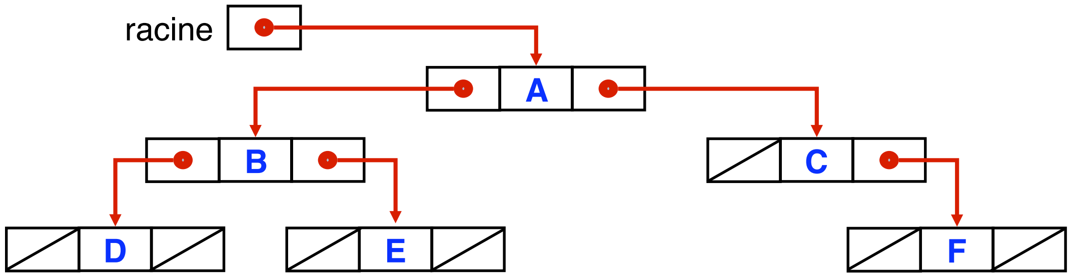



```
type Tr_Noeud is record
     gauche : Ta_Noeud;
     info   : XtypeX;
     droit  : Ta_Noeud;
end record;
```



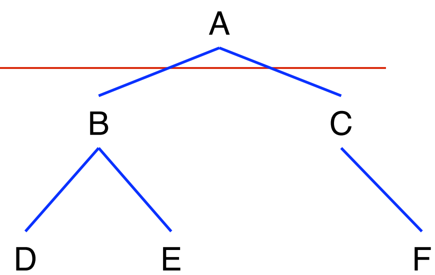

```
type Tr_Noeud is record
     info   : XtypeX;
     fils   : Ta_Noeud;
     frère  : Ta_Noeud;
end record;
```

Représentation d'un arbre



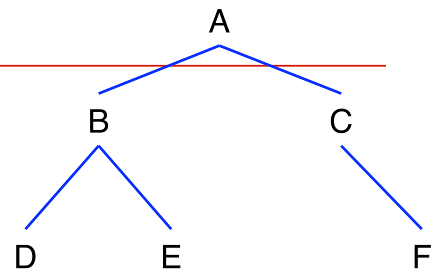
- racine : variable de type pointeur contenant l'adresse du nœud racine de l'arbre
  - *point d'entrée dans l'arbre.*
- Si racine ≠ nil (arbre non vide) alors :
  - racine↑.info ➔ accès à l'information A
  - racine↑.gauche ➔ accès au sous-arbre gauche
  - racine↑.droite ➔ accès au sous-arbre droit

Définition



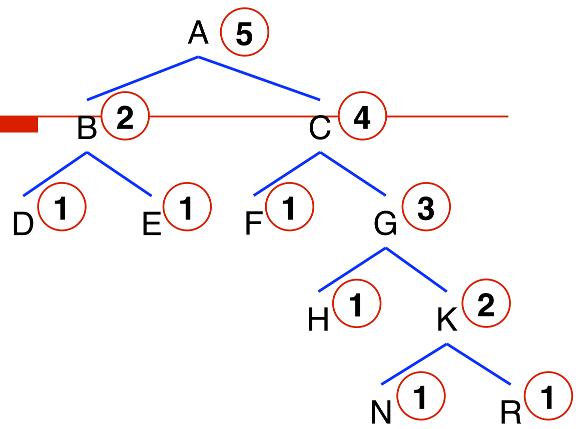
- Mot des feuilles d'un arbre binaire
  - Chaîne formée, de gauche à droite, de la valeur des feuilles de l'arbre.
  - Pour l'arbre ci-dessus, le mot des feuilles est égal à « D E F »

Définition



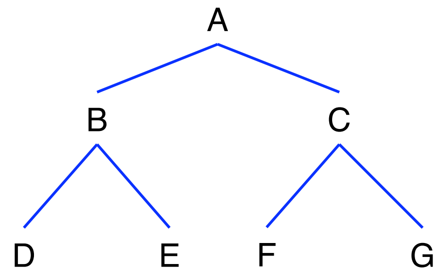
- **Niveau**
  - Deux nœuds sont à un même niveau dans un arbre, s'ils sont issus d'un même nœud après le même nombre de filiations.
    - D, E, F sont situés au même niveau 3 (ils « descendent » de A après deux filiations)
  - niveau de la racine = 1
  - niveau d'un nœud non racine = (niveau de son père) + 1

Définition



- **Hauteur** d'un noeud
  - égale au maximum du niveau des feuilles
- Définition récursive :
  - la hauteur d'un arbre est égale à la hauteur de sa racine
  - la hauteur d'un arbre vide est nulle
  - la hauteur d'un noeud est égale au maximum des hauteurs du sous-arbre gauche et du sous-arbre droit plus un

Définition



- **Arbre binaire complet**
  - Si chaque noeud autre qu'une feuille admet deux descendants
  - et si toutes les feuilles sont au même niveau
- La taille de l'arbre est égale à $2^k - 1$ où k est le niveau des feuilles.