1 Problématique

En 1920 le mathématicien Jan Łukasiewicz présente la *notation polonaise* qui permet d'exprimer des expressions mathématiques sans utiliser de parenthèse, mais traitant néanmoins toute formule sans ambiguïté.
L'expression arithmétique

$$2 \times (3 + 4)$$

devient en notation polonaise

$$\times\ 2 + 3\ 4$$

Dans les années 50, Charles L. Hamblin s'intéresse à variante *inversée* de cette notation. Elle est en effet particulièrement bien adaptée à la manière dont les processeurs traitent leurs opérandes. En notation polonaise inversée, l'expression précédente s'écrit

$$2\ 3\ 4\ +\ \times$$

En 1972 Hewlett-Packard sort une calculatrice financière en notation polonaise inversée.

évite erreurs avec oubli de parenthèses ; après un temps d'adaptation, gain de temps (moins de touches à utiliser)

Quelle représentation en mémoire permet de réaliser un calcul en notation polonaise inversée ?

2 Arbre binaire

2.1 Définition

Un *arbre binaire* est un cas particulier des structures arborescentes.

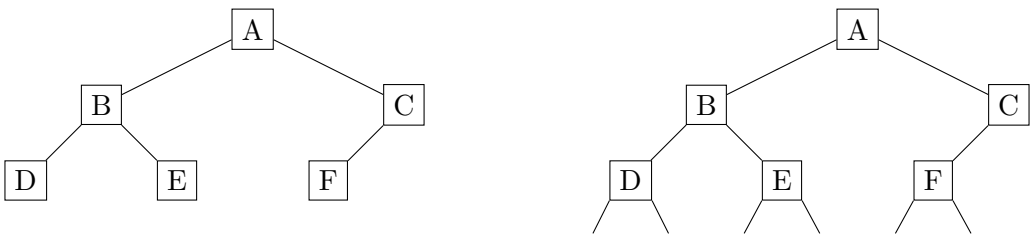
À retenir

Un **arbre binaire** est une structure arborescente où chaque nœud possède **au plus** deux fils.
L'ordre des nœuds-fils est pris en compte : on parle alors de fils *gauche* et fils *droit*.

Le vocabulaire défini précédemment s'applique donc pour ce cas de figure.
sous-arbre gauche et droit
nœud interne = qui a au moins 1 enfant (pas les feuilles donc)

La représentation d'un nœud n'est pas généralisée dans la littérature (figure 1).

FIGURE 1 – Représentations d'un nœud



Christophe Viroulaud

FIGURE 2 – Représentations d'un arbre binaire

Un arbre binaire est :

— **équilibré** si pour chaque nœud interne, les *sous-arbres gauche et droite* ont une hauteur qui diffère au plus de 1,

— **complet** si tous les niveaux sont remplis sauf éventuellement le dernier ; les feuilles sont alors *tassées à gauche*,

— **parfait** si tous les niveaux sont remplis.

2.2 Hauteur

La **taille** représente le nombre de nœuds qui composent l'arbre. La **hauteur (ou profondeur)** est la longueur du plus grand chemin entre la racine et une feuille.

À retenir

Dans un arbre binaire, la taille N et la hauteur h sont liées par les inégalités :

$$h+1 \leqslant N \leqslant 2^{h+1}-1$$

hauteur arbre vide = -1

Remarque

Si la définition de la hauteur est définie comme le nombre maximum de nœuds entre la racine et une feuille, cette propriété s'écrit :

$$h \leqslant N \leqslant 2^{h}-1$$

3 Représentation d'une expression mathématique

Une expression mathématique applique une *opération* sur deux *opérandes*. Un arbre binaire permet donc de représenter n'importe quelle opération.

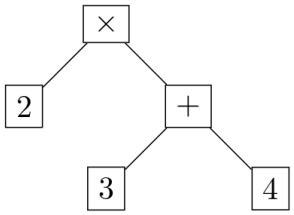


FIGURE 3 – Expression mathématique

4 Représentation d'un arbre binaire en Python

4.1 Structure

Nous modifions légèrement le nœud utilisé dans les structures arborescentes. Un arbre vide est représenté par *None*.

Christophe Viroulaud

```python
class Noeud:

    def __init__(self, v, g, d):
        self.valeur = v
        self.gauche = g
        self.droite = d
```

Code 1 – Nœud d'un arbre binaire

Activité 1 :

1. Construire la variable *arbre* qui représente l'expression mathématique (figure 3).
2. Écrire la fonction *récursive* **taille(a : Noeud) → int** qui renvoie le nombre de nœuds de l'arbre.
3. Écrire la fonction *récursive* **hauteur(a : Noeud) → int** qui renvoie la hauteur de l'arbre.

4.2 Parcours en profondeur d'un arbre binaire

La notion n'est pas nouvelle, cependant le positionnement étant fixé dans un arbre binaire, on commencera par parcourir le sous-arbre gauche avant celui de droite. Ensuite des variations existent selon le moment où on *affiche* la valeur du nœud traversé :

— **Parcours préfixe :**

```
parcours préfixe(arbre)
       affiche(valeur)
       parcours préfixe(sous-arbre gauche)
       parcours préfixe(sous-arbre droit)
```

— **Parcours infixe :**

```
parcours infixe(arbre)
       parcours infixe(sous-arbre gauche)
       affiche(valeur)
       parcours infixe(sous-arbre droit)
```

— **Parcours préfixe :**

```
parcours postfixe(arbre)
       parcours postfixe(sous-arbre gauche)
       parcours postfixe(sous-arbre droit)
       affiche(valeur)
```

Activité 2 :

1. Écrire les trois fonctions *récursives* de parcours qui affichent (*print*) directement la valeur du nœud traversé.
2. Adapter ces fonctions pour renvoyer un tableau ordonné des nœuds traversés.
3. Quel parcours implémente la notation polonaise inverse ?

Christophe Viroulaud