

TP MELOG n° III: application des structures arborescentes aux traitements des expressions arithmétiques et fonctionnelles

L'objectif de ce TP est d'appliquer les principes de la structuration des données pour mettre en œuvre un programme de traitement des expressions arithmétiques et fonctionnelles.

Les expressions arithmétiques se représentent simplement par des arbres comme (figure 1) l'expression $(5+(2+3))*((10*10)+(9*9))$ qui s'écrit $(*(+ 5(* 2 3))(+(* 10 10)(* 9 9)))$ en forme préfixée. Les feuilles contiennent des valeurs numériques et les nœuds (binaires) représentent le symbole d'opérateurs.

Les opérations prises en compte sont la somme (+), la différence (-), le produit (*), le quotient (/) et la puissance (^).

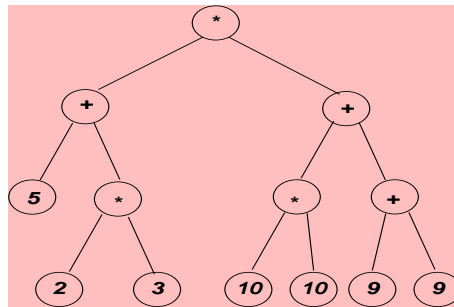


Figure 1 : Représentation d'une expression arithmétique par un arbre

Des tests pourront être effectués grâce à l'utilisation de classes mises à disposition¹ pour la transformation d'expressions de leur forme textuelle à leur format d'arbre selon la classe **Arbre** elle aussi fournie.

Ces tests seront effectués par exemple de la façon suivante :

```
>>> java Sc.java
>>> java Sc "(x+3)*(5-(2*x))"
```

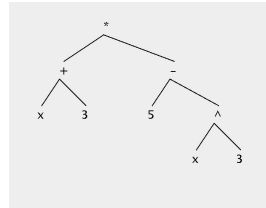
1 Expressions arithmétiques

Question :

- ▶ À partir d'un arbre représentant une expression arithmétique, concevoir l'algorithme et écrire le programme qui permet d'évaluer cette expression pour une valeur donnée de x .
On devrait obtenir :

1. Fichiers fournis : Arbre.java, GUHelper.class, GUHelper\$1.class, JCanvas.class, ManipulationArbre.class, Sc.java

```
>>> java Sc "(x+3)*(5-(2*x))" 5
>>> -40
```



- Après traitements, l'arbre d'une expression peut comporter des sous arbres sur lesquelles des simplifications peuvent être effectuées. Soient, par exemples, si SA est un sous-arbre : $SA+0 \rightarrow SA$, $1*SA \rightarrow SA$, $0*SA \rightarrow 0$, Mettre au point de telles modifications d'arbres grâce aux simplifications données ci-dessus.

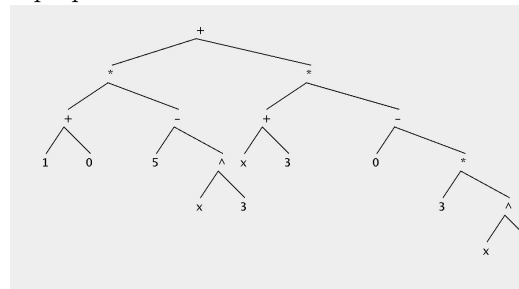
2 Expressions polynomiales et rationnelles

Comme il a été présenté précédemment, on peut donc construire des expressions polynomiales ou expressions rationnelles que l'on peut dériver pour obtenir une nouvelle expression, qui sera également représentée par un arbre.

Questions :

Concevoir les algorithmes et écrire les programmes qui permettent :

- d'obtenir l'arbre qui représente la dérivée d'une expression polynomiale et d'une expression rationnelle,
- d'évaluer l'expression pour une valeur de la variable.
- de simplifier l'arbre de l'expression ainsi obtenue par dérivation



3 Question bonus

Maintenant, les nœuds peuvent aussi contenir en plus le symbole d'une fonction unaire (e.g. fonction trigonométrique) de la variable considérée. On connaît les dérivées de ces fonctions trigonométriques².

Question :

- Concevoir l'algorithme et écrire le programme qui permet d'obtenir l'arbre représentant la dérivée d'une fonction représentée par un arbre donné et ainsi constitué.

2. On pourra se limiter à des fonctions du type $\sin(x)$ (sans tenir compte des fonctions telles que $\sin(a \cdot x^n)$).