

Expression arithmétique

Décembre 2016

Exercice 1. Lecture et écriture d'expression arithmétique

On veut lire une expression arithmétique donnée dans un fichier (ou éventuellement sur la ligne de commande). L'objectif est de la représenter par un arbre. Les opérations sont $+$ et $*$ et les valeurs manipulées sont des flottants.

- a. Proposer un type d'arbre, appelé `arithExpr`, qui permet de stocker une expression arithmétique.
- b. Soit s une chaîne de caractères. On suppose que s commence par une parenthèse ouvrante. Donner un programme qui trouve la position de la parenthèse fermant la première parenthèse.
- c. On suppose pour simplifier que les expressions lues ont la forme suivante :
 - f où f est un float
 - $(expr_1)op(expr_2)$ ou op est une opération $+$ ou $*$ et $expr_1$ et $expr_2$ sont des sous-expressionsPar exemple $((0.1)+(2.23))*(5.0)$ est une expression de valeur 11.65. Écrire une fonction qui prend en entrée une chaîne de caractères représentant une expression arithmétique, et renvoie l'`arithExpr` correspondant.
- d. Donner un algorithme qui affiche le contenu d'un arbre de type `arithExpr`. Comment s'en servir pour écrire dans un fichier la chaîne de caractères correspondant ?

Exercice 2. Évaluation

Écrire une fonction qui calcule la valeur d'une expression arithmétique. L'expression sera donnée dans un fichier dont le nom est passé sur la ligne de commande. L'exécution de `./evalarith test` devra afficher la valeur de l'expression arithmétique écrite dans le fichier `test`.

Exercice 3. Simplification et libération

- a. Écrire une fonction de simplification qui remplace dans un arbre de type `arithExpr` $0 + expr$ par $expr$ et $0 * expr$ par 0 .
- b. Écrire une fonction qui libère la mémoire utilisée par un arbre.

Exercice 4. Génération aléatoire

Programmer une fonction qui crée un arbre aléatoire complet de hauteur n dont les feuilles ont des valeurs flottantes tirées aléatoirement entre $-m$ et m .

Exercice 5. Pour aller plus loin

- a. Ajouter les opérations de division et soustraction.
- b. Ajouter l'opérateur `if then else`, les booléens et les comparaisons.
- c. Gérer les erreurs arithmétiques comme les divisions par zéro et les dépassements de taille.