TD11 - Piles

1. Analyse syntaxique

Par exemple, problème de la reconnaissance des expressions bien parenthèses. Ecrire un algorithme qui :

- accepte les mots comme (), ()(()) ou ((()())()) ;
- rejette les mots comme)(, ()(() ou ((()()))).

2. Calcul d'une expression postfixée

La notation postfixée (polonaise) consiste à placer les opérandes devant l'opérateur. La notation infixée (parenthèsée) consiste à entourer les opérateurs par leurs opérandes. Les parenthèses sont nécessaires uniquement en notation infixée.
Les notations préfixée et postfixée sont d'un emploi plus facile puisqu'on sait immédiatement combien d'opérandes il faut rechercher.
Exemple:
(3 + 5) * 2 s'écrira en notation postfixée (notation polonaise) : 3 5 + 2 *
alors que 3 + (5 * 2) s'écrira: 3 5 2 * +
Pour 6 5 2 3 + 8 * + 3 + * = 6*(5 + ((2+3) * 8) + 3) = 6*48 = 288

Ecrire une fonction qui permet de calculer la valeur d'une expression postfixée.

3. Transformation d'une expression infixée en postfixée

Pour éviter le parenthésage, il est possible de transformer une expression infixée en une expression postfixée en faisant "glisser" les opérateurs arithmétiques à la suite des expressions auxquelles ils s'appliquent.
L'algorithme passe les opérandes à la forme postfixe, mais sauvegarde les opérateurs dans la pile jusqu'à ce que tous les opérandes soient tous traduits.
Exemple: a+b*c+(d*e+f)*g = abc*+de*f+g*+

Précédence des opérateurs (pour cet algorithme) :
3 : opérateurs unaires
2 : / *
1 : + -
0 : (