Compilateurs & Interprètes

Arbres d'expressions arithmétiques (suite)

Octobre/Novembre 2020

But

Construction d'un arbre de dérivation pour évaluer une formule d'expression arithmétique contenant des variables.

Enoncé

On rajoute à la grammaire de la série précédente quelques règles pour définir une formule contenant des variables :

PROG → LISTVAR FORM

LISTVAR → DECLVAR LISTVAR

DECLVAR → # id = nb

LISTVAR → ε

FORM → E

E → T D

D → + E

D → ε

T → F G

G → * T

G → ε

F → (E)

F → nb

F → id (règle ayant été rajoutée)

L'axiome est la première règle. Elle nous dit qu'un programme est donné par une liste de variables et une formule. Les symboles non-terminaux sont en majuscules et les symboles terminaux sont en minuscules. Le symbole terminal *id* désigne un identifiant de variable qui devra commencer obligatoirement par une lettre de l'alphabet. Le symbole terminal *nb* indique un nombre réel ou un entier et enfin le symbole *ε* désigne le mot vide (pour forcer l'arrêt de la récursivité).

Ecrire un programme qui interprète un *PROG* défini par cette grammaire. Cette fois il faudra construire une table des symboles par rapport aux variables qui auront été spécifiées. Tester l'interpréteur en écrivant plusieurs formules. Si une formule contient une erreur lexicale ou syntaxique, elle devra être détectée. Enfin, toute formule syntaxiquement correcte devra être évaluée.

Le listing de ce travail pratique est à rendre au plus tard le 15 novembre 2020. Il pourra être réalisé par groupe de 2 personnes ; une démonstration sera effectuée au laboratoire à l'enseignant.