

Compilateurs & Interprètes

Formules avec Variables

Octobre-Novembre 2024

But

Technique de descente récursive pour évaluer une formule d'expression arithmétique contenant des variables.

Enoncé

On définit une grammaire avec deux opérateurs arithmétiques (+, *), ayant pour axiome la 1^{ère} règle :

SCRIPT → LISTINSTR

LISTINSTR → INSTR LISTINSTR

LISTINSTR → ϵ

INSTR → id = PD_AFF ;

PD_AFF → E

PD_AFF → inv E (commentaire : *inv* indique l'inverse par la division)

INSTR → afficher E ;

INSTR → aff_ral ; (commentaire : affiche un retour à la ligne)

E → T D

D → + E

D → ϵ

T → F G

G → * T

G → ϵ

F → (E)

F → nb

F → id

L'axiome est la première règle. Elle nous dit qu'un *script* est donné par une liste d'instructions. 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 *script* défini par cette grammaire. On vous conseille d'utiliser la technique de descente récursive vue au cours. Il faudra construire une table des symboles par rapport aux variables qui auront été spécifiées. Tester l'interpréteur en écrivant plusieurs (petits) *scripts*. Si une formule contient une erreur lexicale ou syntaxique, elle devra être détectée. Enfin, tout *script* syntaxiquement correct devra être évalué.

Ce travail pratique est à rendre au plus tard le 18 novembre 2024 (à 12h) sur Moodle (juste le programme avec le(s) nom(s) d'une ou deux personnes).