

Rapport sur le Projet de Programmation en Langage C

Analyse et évaluation d'expressions arithmétiques

Introduction

Le présent rapport détaille le travail effectué dans le cadre du projet de programmation en langage C intitulé "Analyse et évaluation d'expressions arithmétiques". Ce projet a été réalisé dans le cadre du cursus universitaire à l'École Supérieure Polytechnique de l'Université Cheikh Anta Diop de Dakar, pendant l'année universitaire 2023-2024, par le groupe constitué de Adolphe Amadou Galland, Cheikh Ahmadou Bamba Sall, Cheikh Ahmed Tidiane Thiandoum, Mouhamed Lamine Faye et Rawane Diouf.

Objectif

L'objectif de ce projet était de concevoir et développer un programme en langage C capable d'analyser la syntaxe des expressions arithmétiques fournies par l'utilisateur et, si la syntaxe est correcte, d'évaluer leur valeur. Le programme devait également être capable de gérer les erreurs de syntaxe de manière appropriée.

Description du Problème

Le problème consiste à écrire un programme qui prend en entrée une expression arithmétique, vérifie sa syntaxe selon une grammaire définie et évalue son résultat si la syntaxe est correcte. Le programme doit pouvoir traiter des opérations d'addition, de soustraction, de multiplication et de division, ainsi que des parenthèses pour définir la priorité des opérations.

Grammaire BNF

La syntaxe des expressions arithmétiques est définie par une grammaire Backus-Naur (BNF), qui comprend des symboles non terminaux (catégories grammaticales), des symboles terminaux (symboles apparaissant tels quels dans les textes analysés), un symbole de départ et des règles de dérivation.

Méthodologie de Développement

Le développement du programme s'est déroulé en deux étapes :

1. **Analyse Syntaxique** : Dans cette phase, un analyseur syntaxique a été développé pour vérifier si une expression donnée est syntaxiquement correcte en fonction de la grammaire BNF fournie.
2. **Évaluation de l'Expression** : Une fois que la syntaxe a été validée, le programme évalue la valeur de l'expression arithmétique en suivant les règles de priorité des opérateurs.

Structure du Programme

Le programme est structuré en différentes fonctions, chacune correspondant à un non-terminal de la grammaire BNF. Les principales fonctions sont :

- `expression()` : pour analyser les expressions arithmétiques.
- `term()` : pour analyser les termes de l'expression.
- `factor()` : pour analyser les facteurs de l'expression.
- `number()` : pour reconnaître les nombres entiers.
- `chiffre()` : pour reconnaître les chiffres.

Fonctionnement du Programme

Le programme lit les expressions fournies par l'utilisateur, vérifie leur syntaxe et évalue leur résultat si la syntaxe est correcte. Il utilise une approche descendante récursive pour l'analyse syntaxique, et évalue les expressions en appliquant les règles de priorité des opérateurs.

Gestion des Erreurs

Dans une première version du programme, la détection d'une erreur de syntaxe entraîne l'affichage d'un message d'erreur et la fin de la session. Une version améliorée pourrait permettre de gérer les erreurs sans interrompre complètement le programme.

Conclusion

Ce projet de programmation en langage C a permis d'approfondir nos connaissances en matière de développement logiciel, en particulier dans la conception et l'implémentation d'un analyseur syntaxique et d'un évaluateur d'expressions arithmétiques. Il a également mis en lumière l'importance de la méthodologie de développement et de la gestion des erreurs dans la création de logiciels fonctionnels et fiables.

Remerciements

Nous tenons à exprimer notre profonde gratitude envers notre enseignant, le Professeur Ibrahima Fall pour sa contribution précieuse à notre formation dans le cadre de ce projet.

Son expertise, son dévouement et son soutien constant ont été des éléments essentiels de notre apprentissage tout au long de ce semestre dans le cours de programmation en langage C.

Annexes

Les annexes contiennent le code source complet du programme ainsi que des exemples d'expressions arithmétiques analysées et évaluées par le programme.