DESENVOLVIMENTO DE UM COMPILADOR DIDÁTICO PARA EXPRESSÕES MATEMÁTICAS EM PYTHON

IMPLEMENTAÇÃO DE ANÁLISE LÉXICA, SINTÁTICA E INTERPRETAÇÃO

UNIC – Universidade de Cuiabá
Curso: Ciências da Computação
Disciplina: Linguagens Formais e Autômatos
Professor: Felipe Douglas

Matheus Henrique Godoy Leite

https://github.com/matheushgodoy/Matheus-Godoy-LFA

Palavras-chave: compiladores; expressões matemáticas; análise léxica; gramática; Python.

Resumo

Este trabalho apresenta o desenvolvimento de um compilador didático em Python para expressões matemáticas, com foco na aplicação dos conceitos de linguagens formais e autômatos. O projeto tem como objetivo demonstrar de forma prática o funcionamento das etapas fundamentais da compilação: análise léxica, análise sintática e interpretação. A implementação utiliza expressões regulares para o reconhecimento de tokens e uma gramática livre de contexto para validação estrutural das expressões. Os resultados esperados incluem a correta identificação e avaliação de expressões matemáticas, além da detecção de erros sintáticos e léxicos de maneira clara e didática. O compilador desenvolvido serve como uma ferramenta de apoio ao ensino, facilitando a compreensão dos processos internos de tradução e execução de linguagens de programação.

1. Introdução

O estudo de compiladores é um dos pilares fundamentais da Ciência da Computação, pois permite compreender como linguagens de programação são traduzidas para instruções compreensíveis por máquinas. A construção de um compilador envolve diversas etapas, como análise léxica, sintática e semântica, que se apoiam diretamente em conceitos de linguagens formais e autômatos.

O presente trabalho tem como objetivo o desenvolvimento de um compilador didático em Python capaz de reconhecer e interpretar expressões matemáticas, como 2 + 3 * 4 ou (5 - 1) / 2. A escolha desse tema se justifica por sua simplicidade conceitual e relevância pedagógica, pois permite explorar de forma prática o funcionamento interno de um compilador, desde a leitura dos tokens até a interpretação do resultado final.

A relevância deste projeto está na integração entre teoria e prática. Ao construir um compilador funcional, o aluno consolida o entendimento de conceitos abstratos como gramáticas livres de contexto, autômatos finitos e análise sintática descendente, aplicando-os em um projeto real e acessível. Além disso, o uso da linguagem Python torna o desenvolvimento mais claro e didático, permitindo o foco no aprendizado dos princípios fundamentais da compilação.

1.2 Definição do Problema de Pesquisa

O problema que se propõe resolver neste trabalho é: como desenvolver um compilador didático que reconheça, valide e interprete corretamente expressões matemáticas, aplicando conceitos de linguagens formais e autômatos.

Em geral, compiladores comerciais são altamente complexos e possuem etapas avançadas de otimização e geração de código binário. Entretanto, em um contexto educacional, o foco está em compreender o processo de análise léxica e sintática, de forma que seja possível visualizar e entender como uma expressão matemática é decomposta e avaliada.

Assim, o compilador desenvolvido neste trabalho não tem como foco gerar código de máquina, mas sim interpretar diretamente as expressões em memória, retornando o resultado ao usuário e ilustrando as fases internas de análise e execução.

2. Objetivos

Desenvolver um compilador didático em Python capaz de realizar a análise léxica, sintática e a interpretação de expressões matemáticas, aplicando conceitos de linguagens formais e autômatos.

Objetivos Específicos:
- Implementar um analisador léxico capaz de identificar números, operadores e parênteses.
- Criar um analisador sintático que valide a estrutura gramatical das expressões conforme as regras da aritmética.
- Implementar uma Árvore Sintática Abstrata (AST) que represente a estrutura hierárquica da expressão.
- Desenvolver um interpretador que percorra a árvore e calcule o resultado final da expressão.
- Validar o funcionamento do compilador com testes de expressões variadas.
- Documentar o processo de desenvolvimento e as tecnologias utilizadas.

3. Metodologia

O projeto caracteriza-se como uma pesquisa experimental aplicada, com foco na implementação prática de um compilador educacional. O desenvolvimento foi realizado em Python, pela sua legibilidade, simplicidade sintática e suporte a bibliotecas voltadas ao processamento de texto.

Foram utilizados os seguintes conceitos e ferramentas:
- Análise léxica: uso de expressões regulares (módulo re) para identificar números, operadores e delimitadores.
- Análise sintática: implementação de uma gramática recursiva baseada em regras livres de contexto, permitindo lidar com precedência de operadores (* e / antes de + e -).
- Interpretação: cálculo do valor da expressão a partir da árvore sintática gerada.

Durante o desenvolvimento, foi aplicado o uso de versionamento de código com Git, garantindo o controle das versões e a organização do projeto. O compilador foi dividido em três módulos principais: Léxico, Sintático e Interpretador. Além disso, foram aplicados princípios de engenharia de software como modularização, documentação e testes automatizados para verificar o comportamento do compilador diante de diferentes expressões, incluindo casos inválidos.

4. Resultados Esperados

Espera-se que o compilador desenvolvido seja capaz de ler, analisar e avaliar corretamente expressões matemáticas de diferentes níveis de complexidade, respeitando a precedência e o agrupamento de operações.

O sistema deverá identificar erros léxicos e sintáticos, como caracteres inválidos ou uso incorreto de parênteses, emitindo mensagens claras e instrutivas. Como resultado final, o compilador deve possibilitar que o usuário digite uma expressão e receba o valor calculado em tempo real, funcionando como uma ferramenta didática para demonstrar o papel das etapas internas de um compilador.

5. Conclusão

O desenvolvimento de um compilador didático para expressões matemáticas em Python proporcionou uma compreensão prática e integrada das etapas fundamentais de um processo de compilação.

A aplicação de conceitos teóricos de linguagens formais e autômatos demonstrou como é possível traduzir regras abstratas em algoritmos concretos, reforçando a importância desses tópicos na formação do cientista da computação.

O projeto confirmou que, mesmo com recursos simples, é possível construir um compilador funcional e educativo, capaz de contribuir para o aprendizado dos processos de análise léxica, parsing e interpretação.

Além de consolidar o entendimento sobre o funcionamento de compiladores reais, o trabalho reforça o valor de projetos práticos como meio de aprendizado ativo e significativo.

Referências

Aho, A. V., Lam, M. S., Sethi, R., & Ullman, J. D. (2006). Compilers: Principles, Techniques, and Tools. Addison Wesley.
Nunez, E. (2020). Construção de Compiladores com Python. São Paulo: Novatec.
Silva, R. L. (2018). Linguagens Formais e Autômatos. LTC Editora.
Grune, D., & Jacobs, C. (2008). Parsing Techniques: A Practical Guide. Springer.