



Universidade Estadual de Maringá

Departamento de Informática

9926 - 31 - Introdução a Compilação

Professor: MSc. Felipe Fernandes da Silva

Análise Léxica, Sintática e Semântica

Henrique Yoshiharu Kajihara RA: 78607

Maringá, 18 de Outubro de 2023

Conteúdo

1	Introdução	1
2	Análise Léxica	2
3	Análise Sintática	2
4	Análise Semântica	2
5	Gramática	2
5.1	Tokens e símbolos	2
6	Árvore	3
7	Referências	4

1 Introdução

O compilador é um tipo de software que traduz o código-fonte de alto nível de um programa para um código de baixo nível ("linguagem de máquina"), para que possa ser executado diretamente pelo hardware do computador.

O compilador implementado realizará na linguagem Python, utilizando a biblioteca PLY, compilação de uma linguagem genérica semelhante as linguagens de programação Pascal e C++ (em sua syntaxe), e irá realizar as etapas de análise léxica (geração de tokens do código fonte), análise sintática (onde a estrutura do programa é analisada e inserida numa árvore de análise), e a análise semântica (onde são feitas as verificações para garantir que o código esteja de acordo com as regras).

2 Análise Léxica

Dificuldades: inicialmente fora tentado implementar um compilador da linguagem Python, utilizando a própria linguagem Python. Porém, durante o desenvolvimento do compilador, não foi possível encontrar um método para verificar a tabulação do código e encontrar o final de cada linha. Portanto, diante destas dificuldades, foi decidido implementar um compilador para uma sintaxe mais genérica análoga a Pascal e C++.

3 Análise Sintática

Dificuldades: após a implementação de diversos tokens para a linguagem Python, ao tentar implementar a árvore de análise com as regras gramaticais foram encontradas dificuldades para delimitar o que era uma expressão, o que seria uma palavra reservada ou um símbolo aritmético. Além disso, a documentação da biblioteca é falha, não descreve em detalhes o que cada método realiza e os diversos exemplos encontrados são discrepantes. Por esse motivo a maior parte dos erros foi solucionada por tentativa e erro.

4 Análise Semântica

Dificuldades: como a árvore de análise foi corretamente organizada, a maior dificuldade para implementar este método foi verificar quando não há folhas para o nó analisado, nestes casos acaba ocorrendo erro ao tentar verificar a variável nula.

5 Gramática

Na gramática do compilador foram implementadas operações binárias (adição, soma, subtração, and, or), precedência de operadores, laço de repetição (while), e a impressão dos dados do prompt (print).

5.1 Tokens e símbolos

Foram implementados os seguintes tokens:

'INTEIRO'
'FLOAT'
'STRING'
'SUBTRAI'
'SOMA'
'MULTIPLICA'
'DIVIDE'
'L_PARENTESES'
'R_PARENTESES'
'ATRIBUI'
'AND'
'OR'
'IDENTIFIER'
'MAIOR'
'MENOR'

'MAIOR_IGUAL'
'MENOR_IGUAL'
'DIFERENTE'
'EQUIVALENTE'
'PONTO_VIRGULA'
'PRINT'
'WHILE'
'ENDWHILE'

Palavras reservadas:

'AND'
'OR'
'PRINT'
'WHILE'
'ENDWHILE'
'TRUE'
'FALSE'

6 **Árvore**

A árvore gerada foi a árvore sintática abstrata, onde cada um de seus nós-folha contém o uma variável com o tipo do identificador, uma variável para armazenar a lista com os filhos gerados na gramática, e uma variável com um nó folha para armazenar regras que não são possíveis de serem simplificadas (numéricos, strings e identificadores).

7 Referências

PLY Python Lex-Tacc