

Implementação de Analisador Léxico com o Flex

Gabryella Mika Tanigawa¹ e Júlia Mendes¹

¹Ciência da Computação - Instituto de Biociências, Letras e Ciências Exatas

Agosto de 2024

Relatório apresentado à disciplina de Compiladores do curso de Ciência da Computação da Universidade Estadual Paulista.

INTRODUÇÃO

O compilador é um programa que mapeia um programa fonte para um programa objeto. E a análise léxica é crucial para o processo de compilação, permitindo o agrupamento dos caracteres do programa fonte em tokens. Diante desse cenário, o objetivo é desenvolver um analisador léxico com o uso da ferramenta FLEX capaz de identificar tokens na linguagem de programação C, como parte de um compilador.

ALFABETO

O alfabeto da linguagem analisada pelo analisador léxico é composto pelos seguintes elementos:

1. Letras Maiúsculas e Minúsculas:

- A, B, C, ..., Z.
- a, b, c, ..., z.

2. Dígitos:

- 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

3. Delimitadores:

- Parênteses: (,)
- Chaves: { , }
- Pontos e vírgulas: ;
- Vírgulas: ,
- Aspas: " , "

4. Símbolos:

- + , - , * , / , % , | , & , = , . , ! , < , >

5. Espaços em branco e quebras de linha:

- ' ' , \n

EXPRESSÕES REGULARES

Identificadores

- (A|B|...|Z|a|b|...|z)(A|B|...|Z|a|b|...|z|0|1|...|9)*

Expressão regular para identificadores, sendo que, eles começam obrigatoriamente com uma letra maiúscula ou minúscula seguida de uma quantidade qualquer de letras (também maiúsculas ou minúsculas), e/ou números.

0.0.1 Exemplos

Aceitos:

- AabbCd
- algo002
- n0Me
- f

Erro Léxico:

- 35aBAj
- ??
- @@

Números Inteiros

- $(0|1|\dots|9)(0|1|\dots|9)^*$

A expressão regular para os números inteiros se caracteriza por denotar palavras com pelo menos um dígito de 0 à 9 seguido por uma quantidade de dígitos de 0 à 9.

Exemplos

Aceitos:

- 1678
- 1
- 000771

Erro Léxico:

- ??@
- @@

Obs:

Para números inteiros negativos, o analisador irá identificar o operador aritmético "-" e um inteiro.

Números Reais

- $(0|1|\dots|9)(0|1|\dots|9)^*(.) (0|1|\dots|9)(0|1|\dots|9)^*$

A expressão regular para os números reais denota palavras com um ou mais dígitos de 0 à 9 seguidos por um ponto e um ou mais dígitos de 0 à 9.

Exemplos

Aceitos:

- 94.77
- 1.0
- 000.780

Erro Léxico:

- .897
- .

Obs:

Para números reais negativos, o analisador irá identificar o operador aritmético "-" e um real.

Operadores Aritméticos

- $(+|-|*|/|\%)$

A expressão regular dos operadores aritméticos denota que eles são compostos por apenas um símbolo podendo ser: +, -, *, /, ou %.

Exemplos

Aceitos:

- +
- -
- *

Erro Léxico:

- ÷

Operadores Lógicos

- (&& | | | | !)

A expressão regular dos operadores denota &&, ||, ou!.

Exemplos

Aceitos:

- &&
- | |
- !

Erro Léxico:

- ! ! !
- && && &&

Operadores Relacionais

- (== | > | < | >= | <= | !=)

Expressão regular dos operadores relacionais com palavras formadas por um ou dois caracteres: ==, >, <, >=, <= ou !=.

Exemplos

Aceitos:

- ==
- <
- >=

Erro Léxico:

- =!
- =>
- =<

Atribuição

- (=)

A expressão regular de atribuição denota apenas a palavra ”=”.

Exemplos

Aceitos:

- =

Erro Léxico:

- ===

Operadores Unários

- `(-- | ++ | &)`

A expressão regular de operadores unários denota palavras com um ou dois caracteres, sendo eles: `--`, `++` ou `&`.

Exemplos

Aceitos:

- `--`
- `++`
- `&`

Erro Léxico:

- `??@`
- `@@`

Separadores

- `((|) | { | } | ; | , | . | ")`

Expressão regular que denota que as palavras tem apenas um caracter sendo elas: `(`, `)`, `{`, `}`, `;`, `,`, `.` ou `"`.

Exemplos

Aceitos:

- `)`
- `{`
- `,`

Erro Léxico:

- `??@`
- `@@`

Comentário

- `(//)`

A Expressão regular denota q o símbolo para identificar comentários é `//`.

Exemplos

Aceitos:

- `//`

Erro Léxico:

- `??@`
- `@@`

Observações

Espaços e quebras de linha são apenas utilizados na hora de ler o arquivo de entrada do analisador, depois disso eles são ignorados.

COMO COMPILAR

Para compilar e executar o analisador léxico descrito no arquivo `'analisadorlexico.l'`, a ferramenta FLEX e o GCC devem estar instalados no sistema. No terminal Ubuntu:

1. Gerar o Código C com Flex

- Use o comando `flex analisadorlexico.l` para gerar o código C a partir do arquivo de definição do analisador léxico.

2. **Compilar o Código C com GCC**

- Compile o código C gerado usando o comando `gcc lex.yy.c -o analisador -lfl`. Isso criará o executável `analisador`.

3. **Executar o Analisador Léxico**

- Execute o analisador léxico com o comando `./analisador codigo_entrada.txt`. O arquivo `codigo_entrada.txt` corresponde ao arquivo de entrada enviado junto ao arquivo de definição do analisador léxico.