

# **Analizador léxico para a linguagem C-**

**Guilherme Inoe Araújo**

Universidade Tecnológica Federal do Paraná (UTFPR) – Campo Mourão – PR –  
Brazil

Departamento Acadêmico de Computação  
Universidade Tecnológica Federal do Paraná (UTFPR) – Campo Mourão, PR –  
Brazil

`guilhermeinoe@alunos.utfpr.edu.br`

## **1. Descrição do trabalho**

O trabalho proposto consiste em um analisador léxico (lexer), que é um importante componente de um compilador. O mesmo é responsável por interpretar um Código fonte e retornar uma lista de tokens a partir de uma entrada, que seria o Código em C-.

Assim, o trabalho têm como objetivo a construção de um analisador léxico para a linguagem C- a partir da construção de um Autômato Finito Determinístico (AFD), que segue a ideia de uma máquina de Moore, onde os valores de saída de um autômato são determinados somente pelo estado atual.

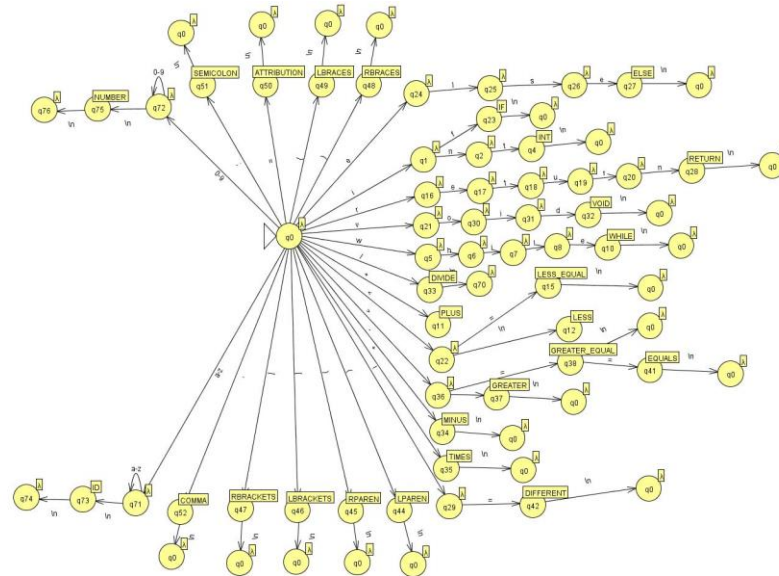
## **2. Linguagem utilizada**

A linguagem utilizada foi Python, em conjunto com a biblioteca “automata-python”

## **3. Processo de Análise Léxica e da implementação**

O Código fonte escrito em Python recebe como entrada um Código em C-, e o mesmo deve analisar cada caractere individualmente e retornar algum token. O Código foi baseado no autômato, e ao final de cada estado existe um espaço e uma quebra de linha, que servem para retornar ao estado inicial q0.

#### 4. Diagrama do autômato



**Figura 1. Diagrama do autômato**

## 5. Exemplos de entradas e saídas

```
Entrada:
int main ( void ) {
    return ( 0 ) ;
}

Lista de Tokens:
INT
ID
LPAREN
VOID
RPAREN
LBRACES
RETURN
LPAREN
NUMBER
RPAREN
SEMICOLON
RBRACES
```

### Figura 2. Exemplo de execução

```
Entrada:
int a ;
float b ;

int main ( void ) {
    int x ;
    float y ;

    return ( 0 ) ;
}

Lista de Tokens:
INT
ID
SEMICOLON
ID
ID
SEMICOLON
INT
ID
LPAREN
VOID
RPAREN
LBRACES
INT
ID
SEMICOLON
ID
ID
SEMICOLON
RETURN
LPAREN
NUMBER
RPAREN
SEMICOLON
RBRACES
```

**Figura 3. Exemplo de execução**