

Analizador Léxico para a Linguagem C- utilizando Máquina de Moore

João Vitor Moraski Lunkes

¹Universidade Tecnológica Federal Do Paraná(UTFPR)
R. Rosalina Maria Ferreira, 1233 - Vila Carolo, Campo Mourão - PR, 87301-899

²Departamento Acadêmico de Computação (DACOM) - UTFPR - CM

joaolunkes@alunos.utfpr.edu.br

Abstract. *This document presents the technical report of the first practical work on the subject of Formal Languages, Automata and Computability, taught by Professor Rogério Aparecido Gonçalves, the work asks for the implementation of a lexical analyzer in some language using an automaton.*

Resumo. *Este documento apresenta o relatório técnico do primeiro trabalho prático da matéria de Linguagens Formais, Autômatos E Computabilidade, ministrada pelo professor Rogério Aparecido Gonçalves, o trabalho pede a implementação de um analisador léxico em alguma linguagem utilizando um automato.*

1. Descrição do trabalho

O trabalho da matéria de Linguagens Formais, Autômatos e Computabilidade é para que seja implementado um analisador léxico utilizando qualquer linguagem, nesse caso foi utilizado PHP, para que ele reconheça os tokens da linguagem C-.

1.1. Linguagem C-

A escolha dessa linguagem foi pelo fato de grande parte das disciplinas introdutórias à programação, como Algoritmos dos cursos de Computação serem ministradas utilizando C, assim teríamos a vantagem de ser uma linguagem conhecida pelos alunos.

1.2. Análise Léxica

A Análise Léxica, ou tokenização é o processo que é feito para converter uma sequência de caracteres em um arquivo em uma sequência de tokens. Um programa que realiza isso é conhecido como lexer, tokenizador ou scanner, porém scanner também pode ser o nome do primeiro estágio de um lexer. O lexer faz a varredura do programa caractere por caractere e traduz isso em uma sequência de símbolos ou tokens. Por conta disso que cada linguagem possui suas palavras reservadas, constantes e identificadores.

2. Implementação

A biblioteca automata-lib, sugerida pelo professor no documento de especificação do trabalho teve algumas limitações para implementação e então, após algumas pesquisas foi decidido usar a linguagem PHP para o trabalho.

Foi utilizado a lógica de passagem de estados para estados em forma de função para que fosse validado letra a letra, símbolo a símbolo e número a número.

Em todo caso das palavras reservadas é verificado se é um id com o nome forVariavel.

Linguagem: PHP 7.4.3

IDE: PHP Storm - JetBrains Student License

O arquivo do projeto possui um arquivo chamado README.md onde é possível encontrar tutoriais de instalação e como rodar o código.

O código está separado em mais de um arquivo.

- entrada0*.cm Onde ficam os casos de teste criados pelo aluno.
- Estados.php Classe que possui todos os estados e as lógicas para passar de um para o outro.
- Token.php Entidade do token que possui várias variáveis estáticas para salvar os valores como constantes na entidade para ser mais fácil utilizar nos estados.
- main.php Arquivo principal para rodar o código, lê um arquivo e após isso executa as verificações linha a linha.
- testes/teste-00*.cm Testes fornecidos pelo professor

3. Casos de teste

3.1. Caso 01

Primeiro caso de teste

```
# Primeiro caso teste
int main(void){
    return(0);
}
```

Figure 1. Primeiro caso de teste

```

0:52 ~/faculdade/lfac/lexical-analyzer master x
$ php main.php
Tamanho do vetor de tokens: 12
Token 0: INT
Token 1: ID
Token 2: LPAREN
Token 3: VOID
Token 4: RPAREN
Token 5: LBRACES
Token 6: RETURN
Token 7: LPAREN
Token 8: NUMBER
Token 9: RPAREN
Token 10: SEMICOLON
Token 11: RBRACES

```

Figure 2. Resposta do primeiro caso de teste

3.2. Caso 02

Segundo caso de teste

```

# Segundo caso teste
int main(void) {
    if(1 < 2){
        return 0;
    }
    while (1){
        void;
    }
    printf(,);
    int numero;
    + - ** / <
    <=
    >
    >=
    ==
    !=
    ()
    []
    {}
    =
    ;
}

```

Figure 3. Segundo caso de teste

```
9:00 ~/faculdade/lfac/lexical-analyzer master x
$ php main.php
Tamanho do vetor de tokens: 58
Token 0: INT          Token 1: ID          Token 2: LPAREN      Token 3: VOID        Token 4: RPAREN
Token 5: LBRACES       Token 6: IF          Token 7: LPAREN      Token 8: NUMBER       Token 9: LESS
Token 10: NUMBER       Token 11: RPAREN     Token 12: LBRACES    Token 13: RETURN      Token 14: NUMBER
Token 15: SEMICOLON    Token 16: RBRACES   Token 17: WHILE      Token 18: LPAREN      Token 19: NUMBER
Token 20: RPAREN       Token 21: LBRACES   Token 22: ID          Token 23: SEMICOLON   Token 24: RBRACES
Token 25: ID           Token 26: LPAREN    Token 27: COMMA      Token 28: SEMICOLON   Token 29: RPAREN
Token 30: SEMICOLON    Token 31: INT       Token 32: ID          Token 33: SEMICOLON   Token 34: PLUS
Token 35: MINUS        Token 36: TIMES     Token 37: TIMES      Token 38: DIVIDE      Token 39: LESS
Token 40: LESS EQUAL   Token 41: ATTRIBUTION Token 42: GREATER     Token 43: GREATER EQUAL Token 44: ATTRIBUTION
Token 45: EQUALS       Token 46: ATTRIBUTION Token 47: DIFFERENT   Token 48: ATTRIBUTION   Token 49: LPAREN
Token 50: RPAREN       Token 51: LBRACKETS Token 52: RBRACKETS   Token 53: LBRACES     Token 54: RBRACES
Token 55: ATTRIBUTION  Token 56: SEMICOLON Token 57: RBRACES
9:00 ~/faculdade/lfac/lexical-analyzer master x
$
```

Figure 4. Resposta do segundo caso de teste

3.3. Caso 03

Terceiro caso de teste

```
## Terceiro caso teste
if (candidato1 >= candidato2){
    if (candidato1 > candidato3){
        if (candidato1 > candidato4){
            candidato1+=branco;
        } else{
            candidato4 += branco;
        }
    } else if (candidato3 > candidato4){
        candidato3+=branco;
    } else{
        candidato4+=branco;
    }
} else{
    candidato2 += branco;
}
```

Figure 5. Terceiro caso de teste

```

13:00 ~/faculdade/lfac/lexical-analyzer master x
$ php main.php
Tamanho do vetor de tokens: 48
Token 0: IF          Token 1: LPAREN      Token 2: ID          Token 3: GREATER_EQUAL Token 4: ATTRIBUTION
Token 5: ID          Token 6: IF          Token 7: LPAREN      Token 8: ID          Token 9: GREATER
Token 10: ID         Token 11: IF         Token 12: LPAREN      Token 13: ID          Token 14: GREATER
Token 15: ID         Token 16: ID         Token 17: SEMICOLON   Token 18: RBRACES     Token 19: ID
Token 20: ID         Token 21: PLUS       Token 22: ATTRIBUTION Token 23: ID          Token 24: SEMICOLON
Token 25: RBRACES    Token 26: RBRACES    Token 27: ELSE        Token 28: ID          Token 29: LPAREN
Token 30: ID         Token 31: GREATER    Token 32: ID          Token 33: ID          Token 34: SEMICOLON
Token 35: RBRACES    Token 36: ID         Token 37: ID          Token 38: SEMICOLON   Token 39: RBRACES
Token 40: RBRACES    Token 41: ID         Token 42: ID          Token 43: PLUS        Token 44: ATTRIBUTION
Token 45: ID         Token 46: SEMICOLON  Token 47: RBRACES
13:00 ~/faculdade/lfac/lexical-analyzer master x
$

```

Figure 6. Resposta do terceiro caso de teste

3.4. Caso 04

Quarto caso de teste

```

15:00 ~/faculdade/lfac-utfpr/lexical-analyzer master x
$ php main.php testes/teste-001.cm
Tamanho do vetor de tokens: 12
Token 0: INT          Token 1: ID          Token 2: LPAREN      Token 3: VOID          Token 4: RPAREN
Token 5: LBRACES      Token 6: RETURN      Token 7: LPAREN      Token 8: NUMBER        Token 9: RPAREN
Token 10: SEMICOLON   Token 11: RBRACES

```

Figure 7. Quarto caso de teste

3.5. Caso 05

Quinto caso de teste

```

15:01 ~/faculdade/lfac-utfpr/lexical-analyzer master x
$ php main.php testes/teste-002.cm
Tamanho do vetor de tokens: 32
Token 0: FLOAT        Token 1: ID          Token 2: LPAREN      Token 3: ID            Token 4: ID
Token 5: FLOAT        Token 6: ID          Token 7: RETURN      Token 8: LPAREN        Token 9: ID
Token 10: PLUS        Token 11: ID         Token 12: SEMICOLON   Token 13: RBRACES      Token 14: INT
Token 15: ID          Token 16: LPAREN     Token 17: VOID        Token 18: RPAREN       Token 19: LBRACES
Token 20: FLOAT       Token 21: ID         Token 22: ATTRIBUTION Token 23: ID            Token 24: LPAREN
Token 25: NUMBER      Token 26: RETURN     Token 27: LPAREN      Token 28: NUMBER       To
ken 29: RPAREN
Token 30: SEMICOLON   Token 31: RBRACES

```

Figure 8. Quinto caso de teste

3.6. Caso 06

Sexto caso de teste

```
15:02 ~/faculdade/lfac-utfpr/lexical-analyzer master x
$ php main.php testes/teste-003.cm
Tamanho do vetor de tokens: 77
Token 0: INT          Token 1: ID          Token 2: LPAREN      Token 3: ID          Token 4: ID
Token 5: INT          Token 6: ID          Token 7: ID          Token 8: ATTRIBUTION Token 9: NUMBER
Token 10: COMMA       Token 11: ID         Token 12: ATTRIBUTION Token 13: NUMBER      Token 14: COMMA
Token 15: ID          Token 16: SEMICOLON  Token 17: ID         Token 18: LPAREN     Token 19: ATTRIBUT
ION
Token 20: NUMBER      Token 21: SEMICOLON  Token 22: LESS_EQUAL Token 23: ATTRIBUTION To
ken 24: ID
Token 25: SEMICOLON   Token 26: ATTRIBUTION Token 27: PLUS       Token 28: NUMBER     Token 29:
RPAREN
Token 30: LBRACES     Token 31: ID         Token 32: ATTRIBUTION Token 33: ID          Token 34: PLUS
Token 35: ID          Token 36: SEMICOLON  Token 37: ID         Token 38: ATTRIBUTION Token 39: ID
Token 40: SEMICOLON   Token 41: ID         Token 42: ATTRIBUTION Token 43: ID          Token 44: SEMICOLON
N
Token 45: RBRACES     Token 46: RETURN     Token 47: ID         Token 48: SEMICOLON  Token 49:
RBRACES
Token 50: INT         Token 51: ID         Token 52: LPAREN     Token 53: VOID       Token 54: RPAREN
Token 55: LBRACES     Token 56: INT        Token 57: ID         Token 58: ATTRIBUTION Token 59: NUMBER
Token 60: NUMBER      Token 61: NUMBER     Token 62: NUMBER     Token 63: NUMBER     To
ken 64: NUMBER
Token 65: NUMBER      Token 66: INT        Token 67: ID         Token 68: ATTRIBUTION Token 69: ID
Token 70: LPAREN      Token 71: ID         Token 72: SEMICOLON  Token 73: RETURN     Token 74:
NUMBER
Token 75: SEMICOLON   Token 76: RBRACES
```

Figure 9. Sexto caso de teste

3.7. Caso 07

Sétimo caso de teste

```
15:03 ~/faculdade/lfac-utfpr/lexical-analyzer master x
$ php main.php testes/teste-004.cm
Tamanho do vetor de tokens: 69
Token 0: INT          Token 1: ID          Token 2: LPAREN      Token 3: ID          Token 4: ID
Token 5: LBRACES      Token 6: IF          Token 7: LPAREN      Token 8: ID          Token 9: EQUALS
Token 10: ATTRIBUTION Token 11: NUMBER     Token 12: RPAREN     Token 13: RETURN     To
ken 14: NUMBER
Token 15: SEMICOLON   Token 16: ID         Token 17: IF         Token 18: LPAREN     Token 19: ID
Token 20: EQUALS      Token 21: ATTRIBUTION Token 22: NUMBER     Token 23: RPAREN     To
ken 24: RETURN
Token 25: NUMBER      Token 26: SEMICOLON  Token 27: ELSE       Token 28: RETURN     Token 29:
ID
Token 30: LPAREN      Token 31: ID         Token 32: MINUS      Token 33: NUMBER     Token 34: RPAREN
Token 35: PLUS        Token 36: ID         Token 37: LPAREN     Token 38: ID         Token 39: MINUS
Token 40: NUMBER      Token 41: RPAREN     Token 42: SEMICOLON  Token 43: RBRACES    To
ken 44: INT
Token 45: ID          Token 46: LPAREN     Token 47: VOID       Token 48: RPAREN     Token 49: LBRACES
Token 50: INT         Token 51: ID         Token 52: SEMICOLON  Token 53: ID         Token 54: ATTRIBUTION
Token 55: NUMBER      Token 56: NUMBER     Token 57: NUMBER     Token 58: INT        Token 59:
ID
Token 60: ATTRIBUTION Token 61: ID         Token 62: LPAREN     Token 63: ID         Token 64: SEMICOLON
N
Token 65: RETURN      Token 66: NUMBER     Token 67: SEMICOLON  Token 68: RBRACES
```

Figure 10. Sétimo caso de teste