Analisador Léxico para a Linguagem C- utilizando Máquina de Moore

João Vitor Moraski Lunkes

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

[2]Departamento 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 lexico em alguma linguagem utilizando um autómato.*

1. Descrição do trabalho

Como é requisitado no arquivo sobre o trabalho, ele se trata da implementação de um analisador léxico utilizando alguma linguagem de programação e que ele consiga analisar a 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.

2. Implementação

Linguagem: PHP 7.4.3
IDE: PHP Storm - Jetbrains Student License
O arquivo do projeto possuí um arquivo chamado README.md onde é possivel encontrar tutoriais de instalação e como rodar o codigo no computador.

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

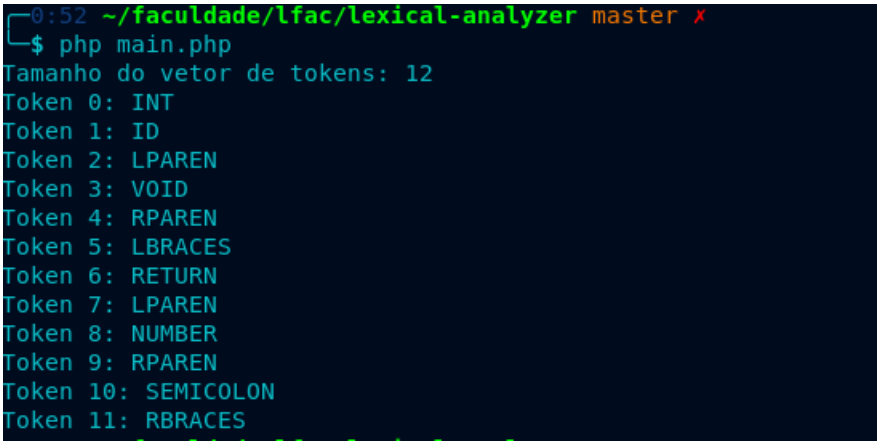

Figure 2. Resposta do primeiro caso de teste

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

Como a resposta do caso 02 é muito grande foi colocado em um arquivo externo para que fosse mais fácil a visualização **Link para acessar as respostas do caso 02 e 03**

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 4. Terceiro caso de teste

Como a resposta do caso 03 é muito grande foi colocado em um arquivo externo para que fosse mais fácil a visualização **Link para acessar as respostas do caso 02 e 03**