Laboratório 1 - Compiladores

Disciplina CES-41

Prof. Mokarzel

14/03/2019

Isabelle Ferreira de Oliveira[1]

[1]Aluno de Graduação em Engenharia do Instituto Tecnológico de Aeronáutica (ITA).

E-mail: Isabelle.ferreira3000@gmail.com

Os códigos elaborados foram escritos em Flex baseados nos enunciados do roteiro e podem ser consultados no repositório do Github abaixo:

- **https://github.com/isabelleferreira3000/ces-41/tree/master/lab1.**

Abaixo seguem as imagens dos resultados obtidos por meio do código implementado.

OBJETIVO:

1. Escrever um programa em Flex reconhecedor de cadeias sobre o alfabeto {0, 1}, tais que o número de dígitos 0 seja par ou o número de dígitos 1 seja par.

2. Escrever um programa em Flex reconhecedor de cadeias sobre o alfabeto {0, 1}, tais que o número de dígitos 0 seja ímpar e o número de dígitos 1 seja ímpar.

3. Escrever um programa em Flex reconhecedor de cadeias sobre o alfabeto {0, 1, 2}, tais que o número de dígitos 2 seja divisível por 5 (Obs: zero é divisível por 5).

4. Escrever um programa em Flex reconhecedor de cadeias sobre o alfabeto {0, 1}, com no mínimo cinco caracteres, tais que qualquer bloco de cinco caracteres consecutivos contenha no mínimo três dígitos 1.

5. Escrever um programa em Flex para fazer análise léxica de uma mini-linguagem que contenha os seguintes átomos: identificadores (ID), constantes inteiras (CTINT), constantes reais (CTREAL), operadores aditivos (OPAD), operadores multiplicativos (OPMULT), abre e fecha-parentesis (ABPAR e FPAR), abre e fecha-chaves (ABCHAV e FCHAV), sinal de atribuição (ATRIB), vírgula e ponto-e-vírgula (VIRG e PVIRG) e ainda as palavras reservadas program, var, int e real. Alguns atributos também foram padronizados no roteiro.

RESULTADOS:

Os resultados saíram conforme o esperado dado os objetivos do roteiro e podem ser vistos abaixo.

- **1ª Questão:**

Entrada:
0011 00111 00011 000111
111 001 00101 11 00 100100 100100001 1 000001 00110001 01110001100

Saída:

output1.txt

```
0011                : Aceita
00111               : Aceita
00011               : Aceita
000111              : Rejeitada
111                 : Aceita
001                 : Aceita
00101               : Aceita
11                  : Aceita
00                  : Aceita
100100              : Aceita
100100001           : Aceita
1                   : Aceita
000001              : Rejeitada
00110001            : Rejeitada
01110001100         : Aceita
```

- **2ª Questão:**

Entrada:
0011 00111 00011 000111
111 001 00101 11 00 100100 100100001 1 000001 00110001 01110001100

Saída:

output2.txt

```
0011                : Rejeitada
00111               : Aceita
00011               : Rejeitada
000111              : Rejeitada
111                 : Aceita
001                 : Aceita
00101               : Rejeitada
11                  : Rejeitada
00                  : Rejeitada
100100              : Rejeitada
100100001           : Aceita
1                   : Aceita
000001              : Rejeitada
00110001            : Rejeitada
01110001100         : Rejeitada
```

- **3ª Questão:**

Entrada:
2 22 222 2222 22222
012 0012201 011122102
100210222 1010122222

Saída:

output3.txt

```
2                    : Rejeitada
22                   : Rejeitada
222                  : Rejeitada
2222                 : Rejeitada
22222                : Aceita
012                  : Rejeitada
0012201              : Rejeitada
011122102            : Rejeitada
100210222            : Rejeitada
1010122222           : Aceita
```

- **4ª Questão:**

Entrada:
11100111
11100000111
10101010101
110010110
10101101

Saída:

output4.txt

```
11100111             : Aceita
11100000111          : Rejeitada
10101010101          : Rejeitada
110010110            : Rejeitada
10101101             : Aceita
```

- **5ª Questão:**

Entrada:
program var int real ( ) { }
= , ; aaa2 abc3abc 123
12.3 + - * /

Saída:

output5.txt

```
  texto  |   tipo   | atributo
---------------------------------
  program|         1|
      var|         2|
      int|         3|
     real|         4|
        (|        10|
        )|        11|
        {|        12|
        }|        13|
        =|        14|
        ,|        15|
        ;|        16|
     aaa2|         5|
  abc3abc|         5|
      123|         6|       123
     12.3|         7|     12.30
        +|         8|         1
        -|         8|         2
        *|         9|         3
        /|         9|         4
```