



Universidade Federal de Viçosa

Campus Florestal

Compiladores CCF 441

Relatório do Trabalho Prático 1

Samuel Sena - 3494

04/09/2021

Neste trabalho foi realizada a implementação de dois analisadores léxicos com o auxílio do gerador de analisador léxico Flex.

Inicialmente foi realizado a implementação do arquivo “lex.l”, as seguintes linhas com expressões e ações correspondentes para identificar os padrões requisitados na descrição do trabalho foram escritas no arquivo:

Figura 1

```
{ws}      { /*nenhuma acao e nenhum retorno*/}  
  
[a-zA-Z]+ {printf("Foi encontrado uma palavra. LEXEMA: %s\n", yytext);}  
  
[+][0-9]+ {printf("Foi encontrado um numero inteiro positivo. LEXEMA: %s\n", yytext);}  
  
[0-9]+ {printf("Foi encontrado um numero inteiro positivo. LEXEMA: %s\n", yytext);}  
  
[-][0-9]+ {printf("Foi encontrado um numero inteiro negativo. LEXEMA: %s\n", yytext);}  
  
[A-Z][A-Z][A-Z][-][0-9][0-9][0-9][0-9] {printf("Foi encontrado uma placa. LEXEMA: %s\n", yytext);}  
  
[0-9][0-9][0-9][0-9][-][0-9][0-9][0-9][0-9] {printf("Foi encontrado um telefone. LEXEMA: %s\n",yytext);}  
  
[+|-]*[0-9]+[.][0-9]+ {printf("Foi encontrado um numero com parte decimal. LEXEMA: %s\n",yytext);}  
  
[a-zA-Z]+" [a-zA-Z]+" [a-zA-Z]+[" "]*[a-zA-Z]* {printf("Foi encontrado um nome proprio. LEXEMA: %s\n",yytext);}  
  
. {printf("Caracter nao reconhecido!\n");}
```

Sendo que cada expressão realiza o casamento dos seguintes padrões:

- Palavra: **[a-zA-Z]+**
- Número inteiro positivo: **[+][0-9]+** e **[0-9]+**
- Número inteiro negativo: **[-][0-9]+**
- Placa: **[A-Z][A-Z][A-Z][-][0-9][0-9][0-9][0-9]**

- Telefone: `[0-9][0-9][0-9][0-9][-][0-9][0-9][0-9][0-9]`
- Número decimal (positivo ou negativo) : `[+|-]*[0-9]+[.][0-9]+`
- Nome próprio: `[a-zA-Z]+" "[a-zA-Z]+" "[a-zA-Z]+[" "]* [a-zA-Z]*`
- Demais caracteres não reconhecidos: `.`
- Tabulações, espaços em branco e quebras de linhas são ignorados através do uso das definições regulares `"ws"` e `"delim"`.

Sendo que para cada padrão casado, a ação correspondente imprime uma saída como requisitado na descrição do trabalho.

Execução

A execução do analisador léxico gerado após o processo de compilação do arquivo `"lex.l"` para o arquivo `"lex.yy.c"`, e em seguida para o arquivo `"a.out"` foi realizada utilizando o arquivo de entrada passado pela descrição do trabalho (nomeado aqui de `"exemplodaniel.txt"`) e também com um arquivo de entrada criado por mim (nomeado de `"entradalex.txt"`). A execução de cada arquivo de entrada e sua respectiva saída (através do comando `./a.out < nomearquivo.txt"`) são exibidos pelas figuras abaixo:

Figura 2

```
exemplodaniel.txt
875878 -3355456 abc5464      abc-5464 ABC-5464      453-2345 9486-0847
Daniel Mendes Barbosa 32.345      Palavra Qualquer 3567-3224
Daniel Mendes Barbosa Daniel Mendes Barbosa      Menezes200
```

Arquivo de entrada `"exemplodaniel.txt"`

Figura 3

```
Foi encontrado um numero inteiro positivo. LEXEMA: 875878
Foi encontrado um numero inteiro negativo. LEXEMA: -3355456
Foi encontrado uma palavra. LEXEMA: abc
Foi encontrado um numero inteiro positivo. LEXEMA: 5464
Foi encontrado uma palavra. LEXEMA: abc
Foi encontrado um numero inteiro negativo. LEXEMA: -5464
Foi encontrado uma placa. LEXEMA: ABC-5464
Foi encontrado um numero inteiro positivo. LEXEMA: 453
Foi encontrado um numero inteiro negativo. LEXEMA: -2345
Foi encontrado um telefone. LEXEMA: 9486-0847
Foi encontrado um nome proprio. LEXEMA: Daniel Mendes Barbosa
Foi encontrado um numero com parte decimal. LEXEMA: 32.345
Foi encontrado uma palavra. LEXEMA: Palavra
Foi encontrado uma palavra. LEXEMA: Qualquer
Foi encontrado um telefone. LEXEMA: 3567-3224
Foi encontrado um nome proprio. LEXEMA: Daniel Mendes Barbosa Daniel
Foi encontrado uma palavra. LEXEMA: Mendes
Foi encontrado uma palavra. LEXEMA: Barbosa
Foi encontrado uma palavra. LEXEMA: Menezes
Foi encontrado um numero inteiro positivo. LEXEMA: 200
```

Execução do arquivo de entrada `"exemplodaniel.txt"`

Figura 4

```
entradalex.txt
+43 NYE-5382 -5 2 4.35 Samuel -5.55 37745-5940
GTV-09e9 pedro -44.5 +33-4 Samuel Campos Sena
samuelpedrocampossena Outraqualquer 8654-5040
Samuel Pedro Campos Sena Samuel Pedro Campos Sena
```

Arquivo de entrada “entradalex.txt”

Figura 5

```
Foi encontrado um numero inteiro positivo. LEXEMA: +43
Foi encontrado uma placa. LEXEMA: NYE-5382
Foi encontrado um numero inteiro negativo. LEXEMA: -5
Foi encontrado um numero inteiro positivo. LEXEMA: 2
Foi encontrado um numero com parte decimal. LEXEMA: 4.35
Foi encontrado uma palavra. LEXEMA: Samuel
Foi encontrado um numero com parte decimal. LEXEMA: -5.55
Foi encontrado um numero inteiro positivo. LEXEMA: 37745
Foi encontrado um numero inteiro negativo. LEXEMA: -5940
Foi encontrado uma palavra. LEXEMA: GTV
Foi encontrado um numero inteiro negativo. LEXEMA: -09
Foi encontrado uma palavra. LEXEMA: e
Foi encontrado um numero inteiro positivo. LEXEMA: 9
Foi encontrado uma palavra. LEXEMA: pedro
Foi encontrado um numero com parte decimal. LEXEMA: -44.5
Foi encontrado um numero inteiro positivo. LEXEMA: +33
Foi encontrado um numero inteiro negativo. LEXEMA: -4
Foi encontrado um nome proprio. LEXEMA: Samuel Campos Sena
Foi encontrado uma palavra. LEXEMA: samuelpedrocampossena
Foi encontrado uma palavra. LEXEMA: Outraqualquer
Foi encontrado um telefone. LEXEMA: 8654-5040
Foi encontrado uma palavra. LEXEMA: Samuel
Foi encontrado uma palavra. LEXEMA: Pedro
Foi encontrado uma palavra. LEXEMA: Campos
Foi encontrado uma palavra. LEXEMA: Sena
Foi encontrado um nome proprio. LEXEMA: Samuel Pedro Campos Sena
```

Execução do arquivo de entrada “entradalex.txt”

A construção do arquivo de entrada “entradalex.txt” teve o intuito de testar se o casamento de padrões aconteceria na ordem correta (já que mais de um tipo de casamento é possível dependendo da ordem em que são declaradas as expressões). Tendo isso em vista, no arquivo de entrada foi testado a entrada de números inteiros e decimais (positivos e negativos), placas, palavras de diferentes tamanhos, números de telefone, nomes próprios. Além disso, tentativas de casamentos incorretos foram testadas, como: número de telefone com 9 dígitos, nome próprio com mais de um espaço entre palavras e placa de carro com uma letra no lugar de um número. Todas essas tentativas de casamentos incorretos não funcionaram, comprovando o funcionamento correto do analisador sintático.

lex2.l

Em seguida, foi implementado o arquivo “lex2.l”. Este realiza o reconhecimento dos seguintes padrões:

- CPF
- RG
- CNPJ
- Placa no padrão mercosul
- Números de celular com ou sem o prefixo 9
- Email
- Horário

Para isso, as seguintes linhas com expressões e ações correspondentes foram escritas no arquivo “lex2.l”:

Figura 6

```
{ws}          { /*nenhuma acao e nenhum retorno*/ }

[0-9][0-9][0-9][.][0-9][0-9][0-9][.][0-9][0-9][0-9][-][0-9][0-9] {printf("Foi encontrado um CPF. LEXEMA: %s\n", yytext);}

[0-9A-Z][0-9A-Z][0-9A-Z][0-9A-Z][0-9A-Z][0-9A-Z][0-9A-Z] {printf("Foi encontrado uma placa no padrao mercosul. LEXEMA: %s\n", yytext);}

[0-9][0-9][.][0-9][0-9][0-9][.][0-9][0-9][0-9[/][0-9][0-9][0-9][0-9][-][0-9][0-9] {printf("Foi encontrado um CNPJ. LEXEMA: %s\n", yytext);}

[9][0-9][0-9][0-9][0-9][0-9][-][0-9][0-9][0-9][0-9] {printf("Foi encontrado um telefone com 9. LEXEMA: %s\n",yytext);}

[0-9][0-9][0-9][0-9][0-9][-][0-9][0-9][0-9][0-9] {printf("Foi encontrado um telefone sem o 9. LEXEMA: %s\n",yytext);}

[A-Z]+[-][0-9]+[.][0-9]+[.][0-9]+ {printf("Foi encontrado um RG. LEXEMA: %s\n",yytext);}

[0-9a-zA-Z.]+@[0-9a-zA-Z]+[.a-zA-Z]+ {printf("Foi encontrado um email. LEXEMA: %s\n",yytext);}

[0-2][0-9][:][0-6][0-9] {printf("Foi encontrado um horario. LEXEMA: %s\n",yytext);}

. {printf("Caracter nao reconhecido!\n");}
```

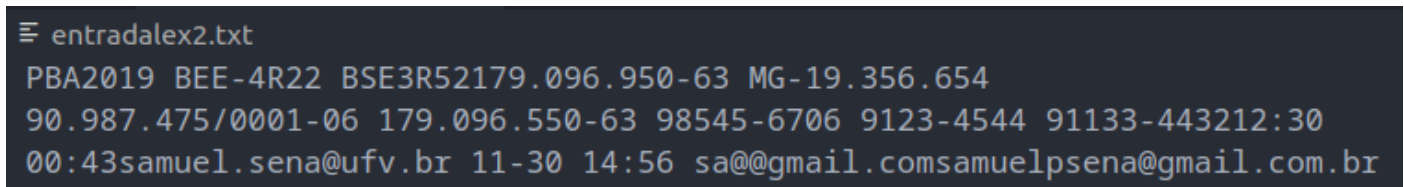
Sendo que cada expressão realiza o casamento dos seguintes padrões:

- CPF: `[0-9][0-9][0-9][.][0-9][0-9][0-9][.][0-9][0-9][0-9][-][0-9][0-9]`
- Placa no padrão mercosul: `[0-9A-Z][0-9A-Z][0-9A-Z][0-9A-Z][0-9A-Z][0-9A-Z][0-9A-Z]`
- CNPJ: `[0-9][0-9][.][0-9][0-9][0-9][.][0-9][0-9][0-9[/][0-9][0-9][0-9][0-9][-][0-9][0-9]`
- Números de celular com o prefixo 9: `[9][0-9][0-9][0-9][0-9][0-9][-][0-9][0-9][0-9][0-9]`
- Números de celular sem o prefixo 9: `[0-9][0-9][0-9][0-9][0-9][-][0-9][0-9][0-9][0-9]`
- RG: `[A-Z]+[-][0-9]+[.][0-9]+[.][0-9]+`
- Email: `[0-9a-zA-Z.]+@[0-9a-zA-Z]+[.a-zA-Z]+`
- Horário: `[0-2][0-9][:][0-6][0-9]`
- Demais caracteres não reconhecidos: `.`
- Tabulações, espaços em branco e quebras de linhas são ignorados através do uso das definições regulares “ws” e “delim”.

Execução lex2.1

A execução do analisador léxico gerado após o processo de compilação do arquivo “lex2.l” para o arquivo “lex.yy.c”, e em seguida para o arquivo “a.out” foi realizada utilizando um arquivo de entrada criado por mim (nomeado “entradalex2.txt”). A execução de arquivo de entrada e sua respectiva saída (através do comando “./a.out < entradalex2.txt”) são exibidos pelas figuras abaixo:

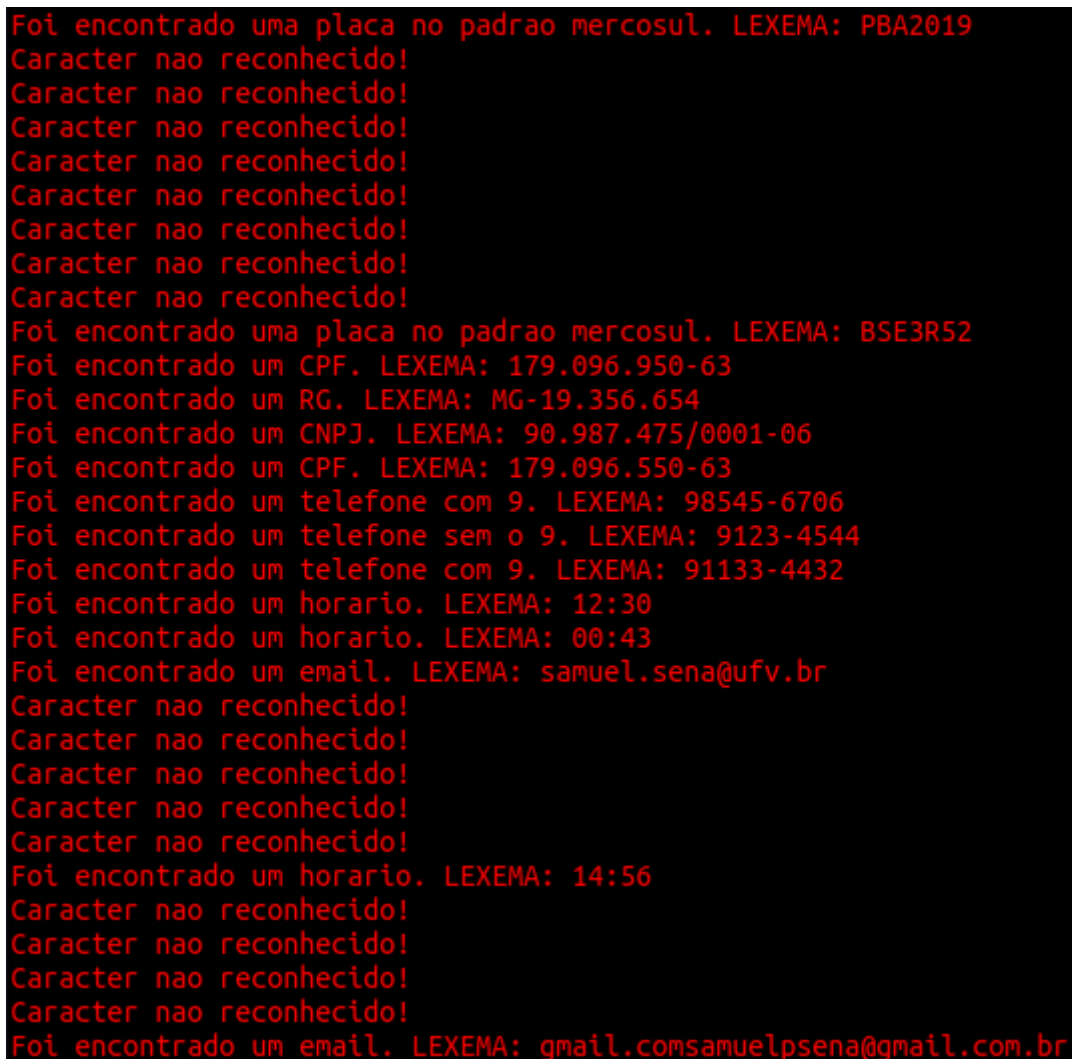
Figura 7



```
≡ entradalex2.txt
PBA2019 BEE-4R22 BSE3R52179.096.950-63 MG-19.356.654
90.987.475/0001-06 179.096.550-63 98545-6706 9123-4544 91133-443212:30
00:43samuel.sena@ufv.br 11-30 14:56 sa@@gmail.comsamuelpsena@gmail.com.br
```

Arquivo de entrada “entradalex2.txt”

Figura 8



```
Foi encontrado uma placa no padrao mercosul. LEXEMA: PBA2019
Caracter nao reconhecido!
Caracter nao reconhecido!
Caracter nao reconhecido!
Caracter nao reconhecido!
Caracter nao reconhecido!
Caracter nao reconhecido!
Caracter nao reconhecido!
Caracter nao reconhecido!
Foi encontrado uma placa no padrao mercosul. LEXEMA: BSE3R52
Foi encontrado um CPF. LEXEMA: 179.096.950-63
Foi encontrado um RG. LEXEMA: MG-19.356.654
Foi encontrado um CNPJ. LEXEMA: 90.987.475/0001-06
Foi encontrado um CPF. LEXEMA: 179.096.550-63
Foi encontrado um telefone com 9. LEXEMA: 98545-6706
Foi encontrado um telefone sem o 9. LEXEMA: 9123-4544
Foi encontrado um telefone com 9. LEXEMA: 91133-4432
Foi encontrado um horario. LEXEMA: 12:30
Foi encontrado um horario. LEXEMA: 00:43
Foi encontrado um email. LEXEMA: samuel.sena@ufv.br
Caracter nao reconhecido!
Caracter nao reconhecido!
Caracter nao reconhecido!
Caracter nao reconhecido!
Caracter nao reconhecido!
Foi encontrado um horario. LEXEMA: 14:56
Caracter nao reconhecido!
Caracter nao reconhecido!
Caracter nao reconhecido!
Caracter nao reconhecido!
Foi encontrado um email. LEXEMA: gmail.comsamuelpsena@gmail.com.br
```

Execução do arquivo de entrada “entradalex2.txt”

A construção do arquivo de entrada “entradalex2.txt” também teve o intuito de testar se o casamento de padrões aconteceria na ordem correta. Para isso, no arquivo de entrada foi testado a entrada da placa no

padrão Mercosul (sem o hífen e com letras e números), CPF, RG, CNPJ, horário, telefone com e sem prefixo 9 e email. Além disso, tentativas de casamentos incorretos foram testadas, como: placa com hífen, horário escrito de maneira errada, email escrito com dois “@”. Todas essas tentativas de casamentos incorretos não funcionaram (não foram reconhecidas como pertencentes às expressões), comprovando o funcionamento correto do analisador sintático.

Referências

MITTAL, Shivani. **Flex (Fast Lexical Analyzer Generator)**. Disponível em: <https://www.geeksforgeeks.org/flex-fast-lexical-analyzer-generator/>. Acesso em: 04 set. 2021.