Universidade Federal de Viçosa

Campus Florestal

Compiladores CCF 441

Relatório do Trabalho Prático 1

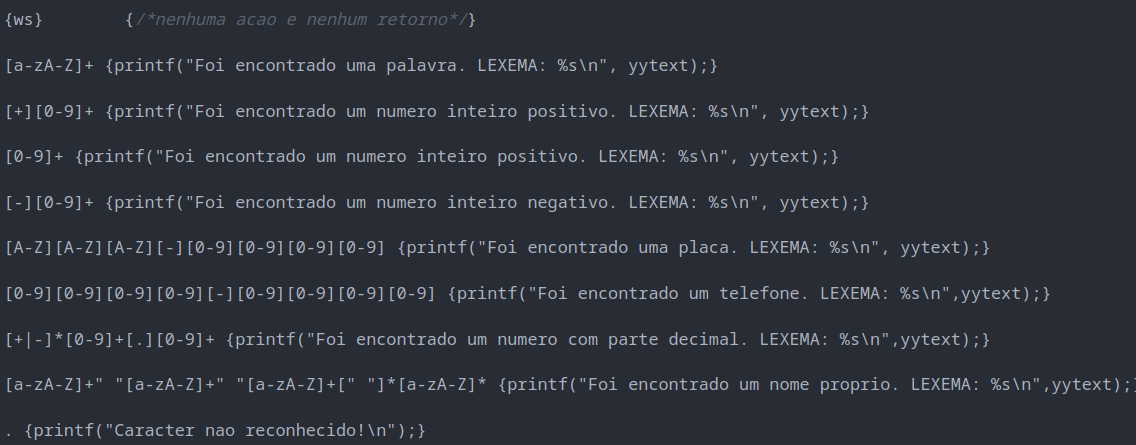
Samuel Sena - 3494

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



Sendo que cada expressão realiza o casamento dos seguintes padrões:

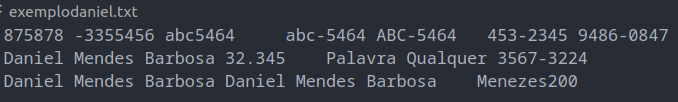
* Palavra: **[a-zA-Z]+**
* Número inteiro positivo: **[+][0-9]+** e **[0-9]+**
* Número inteiro negativo: **[-][0-9]+**
* Placa: **[A-Z][A-Z][A-Z][-][0-9][0-9][0-9][0-9]**
* Telefone: **[0-9][0-9][0-9][0-9][-][0-9][0-9][0-9][0-9]**
* Número decimal (positivo ou negativo) : **[+|-]\*[0-9]+[.][0-9]+**
* 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

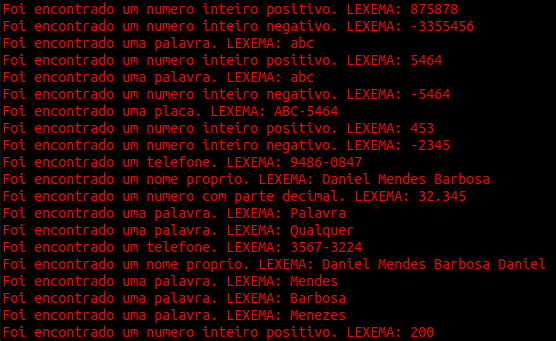
A execução do analisador léxico gerado após o processo de compilação do arquivo “lex.l” para o arquivo “lex.yy.c”, e em seguida para o arquivo “a.out” foi realizada utilizando o arquivo de entrada passado pela descrição do trabalho (nomeado aqui de “exemplodaniel.txt”) e também com um arquivo de entrada 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



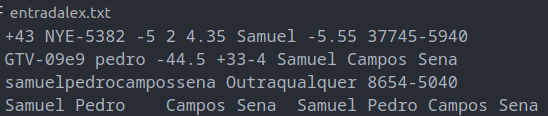
Arquivo de entrada “exemplodaniel.txt”

Figura 3



Execução do arquivo de entrada “exemplodaniel.txt”

Figura 4



Arquivo de entrada “entradalex.txt”

Figura 5



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.

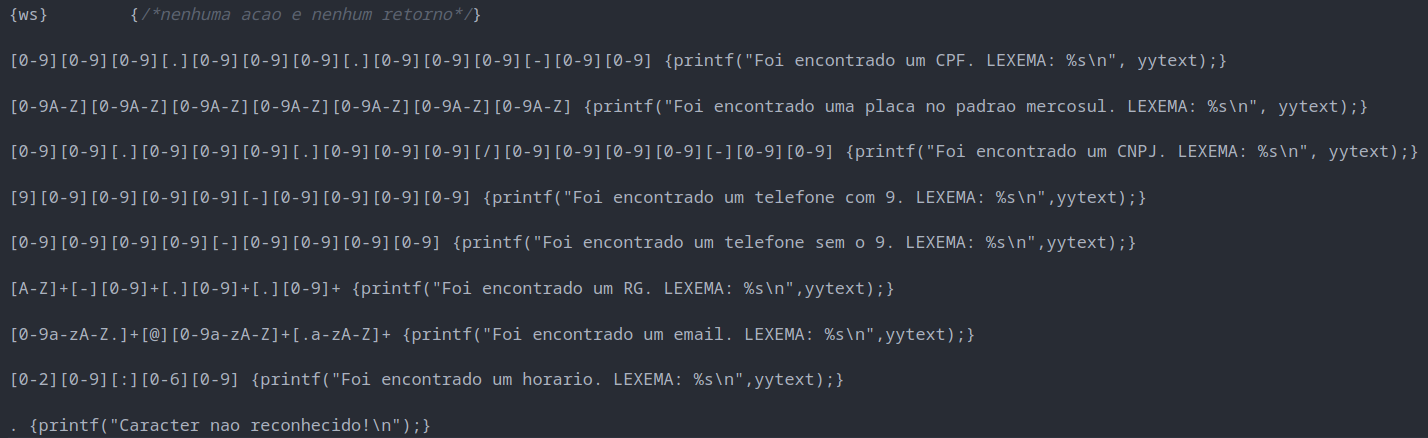
lex2.l

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



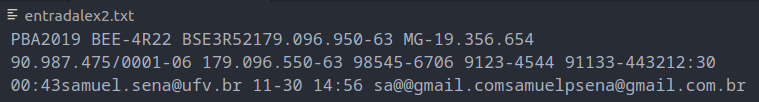
Sendo que cada expressão realiza o casamento dos seguintes padrões:

* 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]**
* Números de celular sem o prefixo 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.l

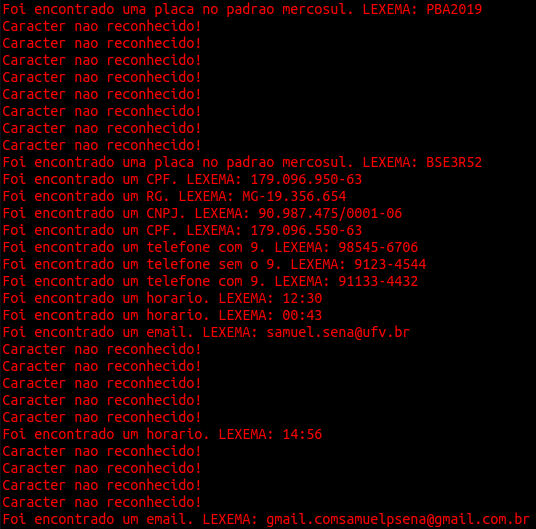
A execução do analisador léxico gerado após o processo de compilação do arquivo “lex2.l” para o arquivo “lex.yy.c”, e em seguida para o arquivo “a.out” foi realizada utilizando um arquivo de entrada criado por mim (nomeado “entradalex2.txt”). A execução de arquivo de entrada e sua respectiva saída (através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

Figura 7



Arquivo de entrada “entradalex2.txt”

Figura 8



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.