



Universidade Federal de Viçosa - Campus Florestal
Compiladores - CCF 441
Trabalho Prático 0

Nome: Gabriel Vitor da Fonseca Miranda

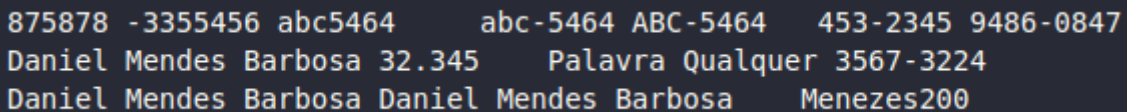
Matrícula: 3857

Professor: Daniel Mendes

Arquivo lex.l

No arquivo lex.l que foi o primeiro arquivo criado foram criadas as definições regulares que reconhecem o fluxo de caracteres passado na documentação do tp. Ou seja, cada lexema passado foi reconhecido pelo analisador léxico produzido pelo lex.

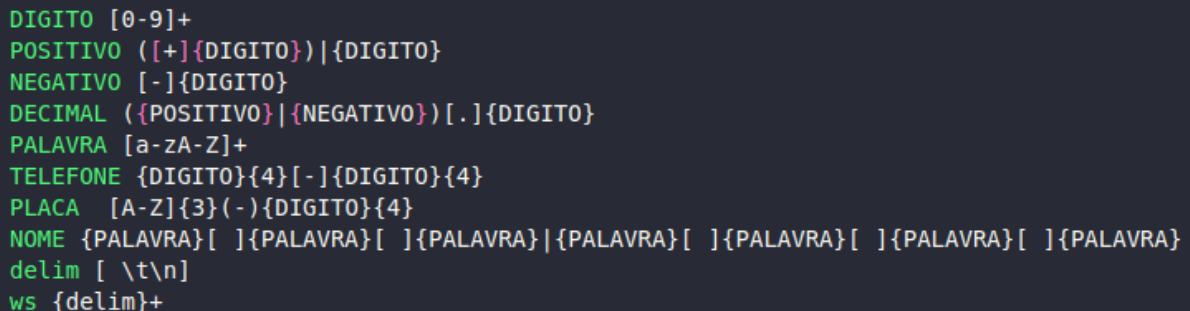
Veja na figura 1 o fluxo de caracteres passados.



```
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
```

figura 1.

Antes de passar o fluxo de caracteres para o analisador léxico, primeiro foram feitas as definições regulares, veja na figura 2 as definições:



```
DIGITO [0-9]+
POSITIVO ([+]{DIGITO})|{DIGITO}
NEGATIVO [-]{DIGITO}
DECIMAL ({POSITIVO}|{NEGATIVO})[.]{DIGITO}
PALAVRA [a-zA-Z]+
TELEFONE {DIGITO}{4}[-]{DIGITO}{4}
PLACA [A-Z]{3}(-){DIGITO}{4}
NOME {PALAVRA}[ ]{PALAVRA}[ ]{PALAVRA}|{PALAVRA}[ ]{PALAVRA}[ ]{PALAVRA}
delim [ \t\n]
ws {delim}+
```

figura 2.

Na figura 2 foram feitas as definições regulares para o fluxo de caracteres da figura 1, onde cada palavra que está na figura foi reconhecida, pois de antemão foi definido um padrão para estas palavras. Na figura 2, é possível ver a palavra “DIGITO” que está em verde, ele é um padrão, um padrão que expressa um expressão regular, cujo a expressão gerada são todos os números de 0 a 9 ou mais dígitos de 0 a 9. Já o padrão “POSITIVO” gera todos os numerais positivos. O padrão “NEGATIVO” também gera todos os numerais, uma diferença é que ele tem um prefixo “-”, que denota a negatividade de um número. O padrão “DECIMAL” irá gerar expressões como sufixo positivas ou negativas, seguidas de um “.” e depois um “DIGITO”. A expressão regular “PALAVRA”, gera todas as letras de A a Z, sejam elas maiúsculas ou minúsculas. O padrão “TELEFONE” tem como início 4 “DÍGITOS” seguidos de “-” e fim 4 “DÍGITOS”. O padrão “PLACA” vai gerar uma expressão em que o início é composto por 3 letras por isso a expressão tem: {3}, seguida de um “-” e com 4 “DÍGITOS”. E a expressão “NOME” vai ter geralmente uma “PALAVRA” e um espaço.

Logo, a partir destas regras definidas conseguiremos reconhecer o fluxo de caracteres da figura 1, usando as expressões regulares definidas no nosso analisador léxico. Veja na figura 3 o fluxo de caracteres sendo reconhecido, pelo analisador léxico gerado pelo lex, com as nossas definições regulares.

```
gabriel@gabriel:~/Documentos/Compiladores/TP0-Compiladores-CCF441$ flex lex.l
gabriel@gabriel:~/Documentos/Compiladores/TP0-Compiladores-CCF441$ gcc lex.yy.c
gabriel@gabriel:~/Documentos/Compiladores/TP0-Compiladores-CCF441$ ./a.out < entrada.txt
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. LEXEMA: Daniel Mendes Barbosa
Foi encontrado um numero 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. 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
```

figura 3.

Diante disso, foi criado outro arquivo (veja na figura 4) para o analisador léxico do arquivo lex.l, este arquivo tem padrões que não são reconhecidos, então vejamos na figura 5, como ficou a saída com o arquivo “entrada2.txt”.

```
gabriel leitura 5555-666 -+1/
compiladores CCF421 != <> -777+777
SAPIENS 10 em compiladores
/////7 *+55 fluxo 0dim
Miranda note++ c++ ?- 2567-
3724 bfc gabriel vitor da fonseca
Lexema14 -5248+4171 -15 -khn
```

figura 4.

Vejamos na figura 5 que alguns caracteres e palavras não foram reconhecidos pelo analisador léxico, ou seja, por que não foram padrões reconhecidos pelas nossas expressões regulares.

```

gabriel@gabriel:~/Documentos/Compiladores/TP0-Compiladores-CCF441$ flex lex.l
gabriel@gabriel:~/Documentos/Compiladores/TP0-Compiladores-CCF441$ gcc lex.yy.c
gabriel@gabriel:~/Documentos/Compiladores/TP0-Compiladores-CCF441$ ./a.out < entrada2.txt
Foi encontrado uma palavra. LEXEMA: gabriel
Foi encontrado uma palavra. LEXEMA: leitura
Foi encontrado um numero inteiro positivo. LEXEMA: 5555
Foi encontrado um numero inteiro negativo. LEXEMA: -666
-Foi encontrado um numero inteiro positivo. LEXEMA: +1
/Foi encontrado uma palavra. LEXEMA: compiladores
Foi encontrado uma palavra. LEXEMA: CCF
Foi encontrado um numero inteiro positivo. LEXEMA: 421
!=<>Foi encontrado um numero inteiro negativo. LEXEMA: -777
Foi encontrado um numero inteiro positivo. LEXEMA: +777
Foi encontrado uma palavra. LEXEMA: SAPIENS
Foi encontrado um numero inteiro positivo. LEXEMA: 10
Foi encontrado uma palavra. LEXEMA: em
Foi encontrado uma palavra. LEXEMA: compiladores
////Foi encontrado um numero inteiro positivo. LEXEMA: 7
*Foi encontrado um numero inteiro positivo. LEXEMA: +55
Foi encontrado uma palavra. LEXEMA: fluxo
Foi encontrado uma palavra. LEXEMA: Odim
Foi encontrado uma palavra. LEXEMA: Miranda
Foi encontrado uma palavra. LEXEMA: note
++Foi encontrado uma palavra. LEXEMA: c
++?-Foi encontrado um numero inteiro positivo. LEXEMA: 2567
-Foi encontrado um numero inteiro positivo. LEXEMA: 3724
Foi encontrado um nome. LEXEMA: bfc gabriel vitor da
Foi encontrado uma palavra. LEXEMA: fonseca
Foi encontrado uma palavra. LEXEMA: Lexema
Foi encontrado um numero inteiro positivo. LEXEMA: 14
Foi encontrado um numero inteiro negativo. LEXEMA: -5248
Foi encontrado um numero inteiro positivo. LEXEMA: +4171
Foi encontrado um numero inteiro negativo. LEXEMA: -15
-Foi encontrado uma palavra. LEXEMA: khn

```

figura 5.

Arquivo lex2.l

No arquivo lex2.l que foi o segundo arquivo criado foram criadas as definições regulares que reconhecem o fluxo de caracteres no arquivo de entrada “entrada3.txt” Ou seja, cada lexema passado foi reconhecido pelo analisador léxico produzido pelo lex.

Veja na figura 6 o fluxo de caracteres passados.

```

4+4 55-55 @Daniel arquivo.c 53:40 12:50
@gabriel 8-5 @Miranda lk.c 99:35 -999 45-66
58:44 @vitor fonseca lex.c -58+333 gabriel

```

Figura 6.”entrada3.txt”.

Antes de passar o fluxo de caracteres para o segundo analisador léxico, primeiro foram feitas as definições regulares, veja na figura 7 as definições:

```

/* definicoes regulares */

DIGITO [0-9]+
POSITIVO ([+]{DIGITO})|{DIGITO}
NEGATIVO [-]{DIGITO}
SOMA ({POSITIVO}|{NEGATIVO})[+]{DIGITO}
SUBTRACAO ({POSITIVO}|{NEGATIVO})[-]{DIGITO}
NOME [A-Za-z]+
INSTAGRAM [@]{NOME}
TEMPORIZADOR {DIGITO}{2}[:]{DIGITO}{2}
ARQUIVOC {NOME}[.][c]
delim      [ \t\n]
ws         {delim}+

```

Figura 7, “Definições regulares lex2.l”.

Na figura 7, foram feitas as definições regulares no arquivo lex2.l. No arquivo foi feita a expressão regular “SOMA” que identifica padrões de operação que estão fazendo uma soma. Já a expressão regular “SUBTRAÇÃO” encontra uma operação que entra dois números tenha um sinal “-” a mesma coisa do “SOMA”, porém muda o sinal. Na expressão “INSTAGRAM” é necessário achar um @ antes de uma expressão “NOME”. A expressão “TEMPORIZADOR” deve encontrar um relógio que marca até 99:99 “99 minutos e 99 segundos”. Por último temos a expressão regular que identifica se o arquivo é um arquivo.c, estamos falando da expressão “ARQUIVOC”.

Dessa forma, a partir destas regras definidas conseguiremos reconhecer o fluxo de caracteres da figura 6, usando as expressões regulares definidas no nosso analisador léxico. Veja na figura 8 o fluxo de caracteres sendo reconhecido, pelo analisador léxico gerado pelo lex, com as nossas definições regulares.

```

gabriel@gabriel:~/Documentos/Compiladores/TP0-Compiladores-CCF441$ flex lex2.l
gabriel@gabriel:~/Documentos/Compiladores/TP0-Compiladores-CCF441$ gcc lex.yy.c
gabriel@gabriel:~/Documentos/Compiladores/TP0-Compiladores-CCF441$ ./a.out < entrada3.txt
Foi encontrado uma soma. LEXEMA: 4+4
Foi encontrado um subtracao. LEXEMA: 55-55
Foi encontrado um @ de instagram. LEXEMA: @Daniel
Foi encontrado um arquivo.c. LEXEMA: arquivo.c
Foi encontrado um temporizador. LEXEMA: 53:40
Foi encontrado um temporizador. LEXEMA: 12:50
Foi encontrado um @ de instagram. LEXEMA: @gabriel
Foi encontrado um subtracao. LEXEMA: 8-5
Foi encontrado um @ de instagram. LEXEMA: @Miranda
Foi encontrado um arquivo.c. LEXEMA: lk.c
Foi encontrado um temporizador. LEXEMA: 99:35
Foi encontrado um numero inteiro negativo. LEXEMA: -999
Foi encontrado um subtracao. LEXEMA: 45-66
Foi encontrado um temporizador. LEXEMA: 58:44
Foi encontrado um @ de instagram. LEXEMA: @vitor
Foi encontrado um nome. LEXEMA: fonseca
Foi encontrado um arquivo.c. LEXEMA: lex.c
Foi encontrado uma soma. LEXEMA: -58+333
Foi encontrado um nome. LEXEMA: gabriel

```

Figura 8.

Agora como entrada para o arquivo lex2.l iremos colocar o arquivo de entrada "entrada4.txt" e verificar como fica o reconhecimento do fluxo de caracteres. Veja na figura 9 o arquivo de entrada.

```
4+4 55-55 lexam, @Daniel arquivo.c 53:40 12:50
,l, ***gabriel 85/85 Daniel/Daniel
2:3333 8-9* g/a/s if while/* /w/w/
```

Figura 9.

Veja na figura 10 que alguns caracteres não foram reconhecidos pelo analisador léxico, ou seja, por que não foram padrões reconhecidos pelas nossas expressões regulares.

```
gabriel@gabriel:~/Documentos/Compiladores/TP0-Compiladores-CCF441$ flex lex2.l
gabriel@gabriel:~/Documentos/Compiladores/TP0-Compiladores-CCF441$ gcc lex.yy.c
gabriel@gabriel:~/Documentos/Compiladores/TP0-Compiladores-CCF441$ ./a.out < entrada4.txt
Foi encontrado uma soma. LEXEMA: 4+4
Foi encontrado um subtracao. LEXEMA: 55-55
Foi encontrado um nome. LEXEMA: lexam
,Foi encontrado um @ de instagram. LEXEMA: @Daniel
Foi encontrado um arquivo.c. LEXEMA: arquivo.c
Foi encontrado um temporizador. LEXEMA: 53:40
Foi encontrado um temporizador. LEXEMA: 12:50
,Foi encontrado um nome. LEXEMA: l
,***Foi encontrado um nome. LEXEMA: gabriel
Foi encontrado um numero inteiro positivo. LEXEMA: 85
/Foi encontrado um numero inteiro positivo. LEXEMA: 85
Foi encontrado um nome. LEXEMA: Daniel
/Foi encontrado um nome. LEXEMA: Daniel
Foi encontrado um numero inteiro positivo. LEXEMA: 2
:Foi encontrado um numero inteiro positivo. LEXEMA: 3333
Foi encontrado um subtracao. LEXEMA: 8-9
*Foi encontrado um nome. LEXEMA: g
/Foi encontrado um nome. LEXEMA: a
/Foi encontrado um nome. LEXEMA: s
Foi encontrado um nome. LEXEMA: if
Foi encontrado um nome. LEXEMA: while
/*Foi encontrado um nome. LEXEMA: w
/Foi encontrado um nome. LEXEMA: w
```

Figura 10.