

Compiladores
Trabalho Prático 0
Analisador léxico

Isabella Menezes Ramos - 3474

13 de junho de 2022

Conteúdo

1	Analisadores léxicos	2
1.1	Analisador léxico 1	2
1.2	Analisador léxico 2	6

1 Analisadores léxicos

O trabalho consiste na implementação de dois analisadores léxicos gerados por um programa gerador de analisadores léxicos, o Flex. O objetivo é construir analisadores léxicos que reconhece combinações de padrões, definidos pelo professor e pelo aluno. Para a implementação dos analisadores léxicos, foi utilizado como referência o livro "Compiladores: Princípios, técnicas e ferramentas" e o material complementar fornecido pelo professor.

1.1 Analisador léxico 1

Para a implementação do arquivo lex.l, foi utilizado as expressões regulares presentes na Figura 1, para reconhecer os tokens solicitados no trabalho. As expressões utilizadas são:

Para reconhecimento de números inteiros positivos, foi utilizado o seguinte padrão:

$$[+] * [0 - 9] +$$

Por o sinal de adição ser optativo, foi utilizado o símbolo "*", fecho de kleene, para indicar que ele pode ou não aparecer no arquivo de entrada. Para indicar os números possíveis, foi utilizado o intervalo de 0 a 9, seguido do símbolo "+" para indicar que poderá ter um ou mais números dentro do intervalo de números estabelecidos.

Para o reconhecimento de um número inteiro negativo, foi utilizado apenas o sinal de subtração dentro do colchete. Para indicar o intervalo de números possíveis foi utilizado o intervalo de 0 a 9, e para indicar que poderá haver um ou mais símbolos deste intervalo, foi utilizado o símbolo "+". A expressão completa está abaixo.

$$[-][0 - 9] +$$

Para o reconhecimento de um número flutuante, foi utilizado o símbolo de união entre os sinais de adição e subtração, indicando que poderá ser um número positivo ou negativo, seguido do fecho de kleene, indicando que poderá ou não haver este sinal. Para indicar o intervalo da parte inteira possível, foi utilizado o intervalo de 0 a 9, seguido do símbolo "+" para indicar que poderá ter um ou mais números dentro do intervalo de números estabelecidos. Para indicar a parte fracionada, foi utilizado o sinal de "." além do intervalo de 0 a 9. A expressão completa está abaixo.

$$[+|-] * [0 - 9] + [.] [0 - 9] +$$

Para o reconhecimento de uma placa de carro, foi utilizado três intervalos da letra A até Z, o símbolo de "-" e mais quatro intervalos de 0 a 9. A expressão completa está abaixo.

$$[A - Z][A - Z][A - Z][-][0 - 9][0 - 9][0 - 9][0 - 9]$$

Para o reconhecimento de uma palavra qualquer, foi utilizado dois intervalos, um de a-z, indicando que a palavra pode ter letras minúsculas e um intervalo de A-Z, indicando que a palavra pode ter letras maiúsculas, seguido do sinal de "+", indicando que a palavra poderá ter um ou mais caracteres dos intervalos. A expressão completa está abaixo.

$$[a - zA - Z] +$$

Para o reconhecimento de um número de telefone, foi utilizado quatro intervalos de 0 a 9 concatenados, juntamente de um sinal de "-" concatenado com mais quatro intervalos de 0 a 9 concatenados. A expressão completa está abaixo.

$$[0 - 9][0 - 9][0 - 9][0 - 9][-][0 - 9][0 - 9][0 - 9][0 - 9]$$

Para o reconhecimento de um nome próprio, tendo três ou quatro palavras, foi utilizado primeiramente a concatenação de dois intervalos, de A-Z, indicando que o nome deve começar com letra maiúscula, seguido do intervalo a-z, indicando que o restante da palavra deve ser em letra minúscula. Após o segundo intervalo, temos o sinal "+" indicando que poderá ter uma ou mais letras minúsculas. Após isso, temos a concatenação com um espaço em branco, antes de ter uma outra concatenação do intervalo de A-Z com o intervalo a-z. Temos também outra concatenação de A-Z com a-z seguido do símbolo "+" e concatenação de um espaço em branco com fecho de kleene. Após isso, Um intervalo de A-Z com fecho de kleene, concatenado com a-z com fecho de kleene, concatenado com um espaço em branco, também com fecho de kleene. A expressão completa está abaixo.

$$[A - Z][a - z] + " "[A - Z][a - z] + " "[A - Z][a - z]" " * [A - Z][a - z]" " *$$

O código completo, com todas as implementações citadas acima, estão presente na Figura 1.

```

%{
/*codigo colocado aqui aparece no arquivo gerado pelo flex*/
%}

/* This tells flex to read only one input file */
%option noyywrap

/* definicoes regulares */

delim      [ \t\n]
ws         {delim}+

%%

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

[+]*[0-9]+ {printf("\nFoi encontrado um número inteiro positivo. LEXEMA: %s\n",yytext);}

[-]*[0-9]+ {printf("\nFoi encontrado um número inteiro negativo. LEXEMA: %s\n",yytext);}

[+|-*][0-9]+.[0-9]+ {printf("\nFoi encontrado um número flutuante. LEXEMA: %s\n",yytext);}

[A-Z][A-Z][A-Z][-][0-9][0-9][0-9][0-9] {printf("\nFoi encontrado uma placa. LEXEMA: %s\n",yytext);}

[a-zA-Z]+ {printf("\nFoi encontrado uma palavra. LEXEMA: %s\n", yytext);}

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

[A-Z][a-z]+" "[A-Z][a-z]+" "[A-Z][a-z]+[" "]*[A-Z]*[a-z]*[" "]* {printf("\nFoi encontrado um nome
próprio. LEXEMA: %s\n",yytext);}

%%

/*codigo em C. Foi criado o main, mas podem ser criadas outras funcoes aqui.*/

int main(void)
{
    /* Call the lexer, then quit. */
    yylex();
    return 0;
}

```

Figura 1: Código de lex.l

Para testar o analisador léxico, foi utilizado como entrada primeiramente o arquivo de entrada fornecido pelo professor. O arquivo de entrada está presente na Figura 2.

Para executar o código lex, foi seguido os seguintes passos:

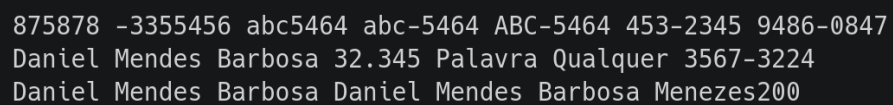
```

flex lex.l
gcc lex.yy.c

```

Para ler o arquivo de entrada foi usado:

```
./a.out < entrada.txt
```

A terminal window with a dark background and light green text. It displays the output of a program that reads from a file named 'entrada.txt'. The output consists of three lines of space-separated tokens.

```
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 2: Entrada 01 de lex.l

A saída obtida está presente na Figura 3.

```

Foi encontrado um número inteiro positivo. LEXEMA: 875878
Foi encontrado um número inteiro negativo. LEXEMA: -3355456
Foi encontrado uma palavra. LEXEMA: abc
Foi encontrado um número inteiro positivo. LEXEMA: 5464
Foi encontrado uma palavra. LEXEMA: abc
Foi encontrado um número inteiro negativo. LEXEMA: -5464
Foi encontrado uma placa. LEXEMA: ABC-5464
Foi encontrado um número inteiro positivo. LEXEMA: 453
Foi encontrado um número inteiro negativo. LEXEMA: -2345
Foi encontrado um telefone. LEXEMA: 9486-0847
Foi encontrado um nome próprio. LEXEMA: Daniel Mendes Barbosa
Foi encontrado um número flutuante. 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 próprio. LEXEMA: Daniel Mendes Barbosa Daniel
Foi encontrado um nome próprio. LEXEMA: Mendes Barbosa Menezes
Foi encontrado um número inteiro positivo. LEXEMA: 200

```

Figura 3: Saída 01 de lex.l

Outro arquivo de entrada foi gerado com o intuito de testar o analisador léxico que foi implementado. O outro arquivo de entrada está presente na Figura 4.

```

Isabella Menezes Ramos Ramos ramos 3398-9250 323232.2
-25021 +2325 AXT-2020 Uma palavra qualquer
Isabella2020Ramos Contagem-20 -30.2

Isabella Fulana De Tal

```

Figura 4: Entrada 02 de lex.l

A saída obtida está presente na Figura 5.

```
Foi encontrado um nome próprio. LEXEMA: Isabella Menezes Ramos Ramos
Foi encontrado uma palavra. LEXEMA: ramos
Foi encontrado um telefone. LEXEMA: 3398-9250
Foi encontrado um número flutuante. LEXEMA: 323232.2
Foi encontrado um número inteiro negativo. LEXEMA: -25021
Foi encontrado um número inteiro positivo. LEXEMA: +2325
Foi encontrado uma placa. LEXEMA: AXT-2020
Foi encontrado uma palavra. LEXEMA: Uma
Foi encontrado uma palavra. LEXEMA: palavra
Foi encontrado uma palavra. LEXEMA: qualquer
Foi encontrado uma palavra. LEXEMA: Isabella
Foi encontrado um número inteiro positivo. LEXEMA: 2020
Foi encontrado uma palavra. LEXEMA: Ramos
Foi encontrado uma palavra. LEXEMA: Contagem
Foi encontrado um número inteiro negativo. LEXEMA: -20
Foi encontrado um número flutuante. LEXEMA: -30.2
Foi encontrado um nome próprio. LEXEMA: Isabella Fulana De Tal
```

Figura 5: Saída 02 de lex.l

1.2 Analisador léxico 2

O segundo analisador léxico construído tem o intuito de reconhecer alguns domínios de sites comerciais de diversos países. Para que o analisador possa reconhecer, o site deve começar com "www", ter a extensão de domínio comercial (".com") e ter o código do país correspondente. Outra opção é terminar apenas com a extensão do domínio comercial (".com"), onde o analisador léxico reconhece apenas como um domínio comercial.

Os domínios de países que o analisador léxico reconhece são do Brasil, do Canadá, da Coreia do Sul, do Japão, do Reino Unido, da Argentina, da Itália, do Egito, da França, de Hong Kong, do México, da Austrália, da Nigéria, do Peru, e como já dito antes, um domínio comercial qualquer.

Uma observação a se fazer, é que o analisador léxico pode reconhecer duas expressões de domínio do Reino Unido, "uk" e "gb", sendo a última pouco utilizada mas foi também escolhida para ser reconhecida pelo analisador léxico.

O formato das expressões regulares são semelhantes. Primeiramente é atribuído uma constante, sendo esta "www". Logo em seguida, temos uma constante de ".", em seguida temos um intervalo de "[a-zA-Z]", que significa que podemos ter no endereço letras minúsculas e maiúsculas, pois isto não influencia no domínio. Após isto, temos mais um intervalo de "[a-zA-Z]" seguido do símbolo "+" e como sabemos, o símbolo "+" significa que podemos ter um ou mais símbolos na expressão. Foi feita esta escolha pois uma das regras de domínio de sites é ter dois ou mais caracteres no endereço. Logo em seguida temos novamente o símbolo "." seguido da constante "com", novamente o símbolo "." e por fim, o domínio dos países ditos anteriormente. Como já dito, caso o endereço não tenha o domínio

do país, ele será reconhecido como um domínio comercial. O código completo, com todas as implementações citadas, está presente na Figura 6.

O arquivo de entrada utilizado para ser reconhecido pelo analisador léxico está disponível na Figura 7. Ele tem mais o objetivo de encontrar erros no reconhecimento das expressões do analisador léxico. Lembrando que tanto na primeira implementação quanto na segunda, há mais de uma combinação de padrões disponíveis.

A saída obtida pelo analisador léxico 2, está disponível na Figura 8.

Como podemos observar na Figura 8, muitas palavras não foram reconhecidas pelo analisador léxico, apenas impressas na saída. Mas este foi o resultado esperado, já que a maioria das palavras compostas no arquivo de entrada não eram para ser reconhecidas.

```

%{

/*codigo colocado aqui aparece no arquivo gerado pelo flex*/

%}

/* This tells flex to read only one input file */
%option noyywrap

/* definicoes regulares */

delim      [ \t\n]
ws         {delim}+

%%

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

www\.[a-zA-Z][a-zA-Z]+\.[a-zA-Z]{2,3} {printf("\nFoi encontrado um dominio do Brasil. LEXEMA: %s\n",yytext);}

www\.[a-zA-Z][a-zA-Z]+\.[a-zA-Z]{2,3} {printf("\nFoi encontrado um dominio do Canadá. LEXEMA: %s\n",yytext);}

www\.[a-zA-Z][a-zA-Z]+\.[a-zA-Z]{2,3} {printf("\nFoi encontrado um dominio da Coreia do Sul. LEXEMA: %s\n",yytext);}

www\.[a-zA-Z][a-zA-Z]+\.[a-zA-Z]{2,3} {printf("\nFoi encontrado um dominio do Japão. LEXEMA: %s\n",yytext);}

www\.[a-zA-Z][a-zA-Z]+\.[a-zA-Z]{2,3} {printf("\nFoi encontrado um dominio do Reino Unido. LEXEMA: %s\n",yytext);}

www\.[a-zA-Z][a-zA-Z]+\.[a-zA-Z]{2,3} {printf("\nFoi encontrado um dominio do Reino Unido. LEXEMA: %s\n",yytext);}

www\.[a-zA-Z][a-zA-Z]+\.[a-zA-Z]{2,3} {printf("\nFoi encontrado um dominio da Argentina. LEXEMA: %s\n",yytext);}

www\.[a-zA-Z][a-zA-Z]+\.[a-zA-Z]{2,3} {printf("\nFoi encontrado um dominio da Itália. LEXEMA: %s\n",yytext);}

www\.[a-zA-Z][a-zA-Z]+\.[a-zA-Z]{2,3} {printf("\nFoi encontrado um dominio do Egito. LEXEMA: %s\n",yytext);}

www\.[a-zA-Z][a-zA-Z]+\.[a-zA-Z]{2,3} {printf("\nFoi encontrado um dominio da França. LEXEMA: %s\n",yytext);}

www\.[a-zA-Z][a-zA-Z]+\.[a-zA-Z]{2,3} {printf("\nFoi encontrado um dominio de Hong Kong. LEXEMA: %s\n",yytext);}

www\.[a-zA-Z][a-zA-Z]+\.[a-zA-Z]{2,3} {printf("\nFoi encontrado um dominio do México. LEXEMA: %s\n",yytext);}

www\.[a-zA-Z][a-zA-Z]+\.[a-zA-Z]{2,3} {printf("\nFoi encontrado um dominio da Austrália. LEXEMA: %s\n",yytext);}

www\.[a-zA-Z][a-zA-Z]+\.[a-zA-Z]{2,3} {printf("\nFoi encontrado um dominio da Nigéria. LEXEMA: %s\n",yytext);}

www\.[a-zA-Z][a-zA-Z]+\.[a-zA-Z]{2,3} {printf("\nFoi encontrado um dominio do Peru. LEXEMA: %s\n",yytext);}

www\.[a-zA-Z][a-zA-Z]+\.[a-zA-Z]{2,3} {printf("\nFoi encontrado um dominio comercial. LEXEMA: %s\n",yytext);}

%%

/*codigo em C. Foi criado o main, mas podem ser criadas outras funcoes aqui.*/

int main(void)
{
    /* Call the lexer, then quit. */
    yylex();
    return 0;
}

```

Figura 6: Código de lex2.1


```

www.isabella.com.brwww.isabella.com.br    ww.com.br
www.isabella.com.com.br    www.isabella.com.uk    www.isabella.com.jp    www.www.w
www.isabella.com.hk        isa.isa.isa.com    www.w.fr www.site.coom.br
mxmx.mxmx.com.mx.com    www.isabella.com.ng    australia.australia.com.au
www.italia.com.it    pe.pe.com.pe www.com.br www.italia.com.it.it

```

Figura 7: Entrada 01 do lex2.l

```

Foi encontrado um dominio do Brasil. LEXEMA: www.isabella.com.br
w
Foi encontrado um dominio do Brasil. LEXEMA: www.isabella.com.br
ww.com.br
Foi encontrado um dominio comercial. LEXEMA: www.isabella.com
.com.br
Foi encontrado um dominio do Reino Unido. LEXEMA: www.isabella.com.uk

Foi encontrado um dominio do Japão. LEXEMA: www.isabella.com.jp
www.www.w
Foi encontrado um dominio de Hong Kong. LEXEMA: www.isabella.com.hk
isa.isa.isa.comwww.w.frwww.site.coom.brmxmx.mxmx.com.mx.com
Foi encontrado um dominio da Nigéria. LEXEMA: www.isabella.com.ng
australia.australia.com.au
Foi encontrado um dominio da Itália. LEXEMA: www.italia.com.it
pe.pe.com.pewww.com.br
Foi encontrado um dominio da Itália. LEXEMA: www.italia.com.it

```

Figura 8: Saída 01 lex2.l