

UESC – Universidade Estadual de Santa Cruz

Relatório de Projeto

Relatório do Projeto1 da disciplina Conceitos de Linguagens de
Programação (CET087)

Giovanne Almeida Messias (201120035)
26/3/2015

Tabela de conteúdo

1. Instruções para compilar o projeto (usando GCC)	2
2. Instruções para executar o projeto	2
3. Links pra download	3

1. Instruções para compilar o projeto (usando GCC)

Abra o **Prompt de Comando**, navegue até a pasta onde se encontra o arquivo “`analizador.c`” e digite:

```
gcc analisador.c -o analisador.exe [Enter]
```

2. Instruções para executar o projeto

Para usar executar, basta acionar o programa e selecionar a gramática base.

Para isso, ainda no **Prompt de Comando**, no mesmo diretório digite:

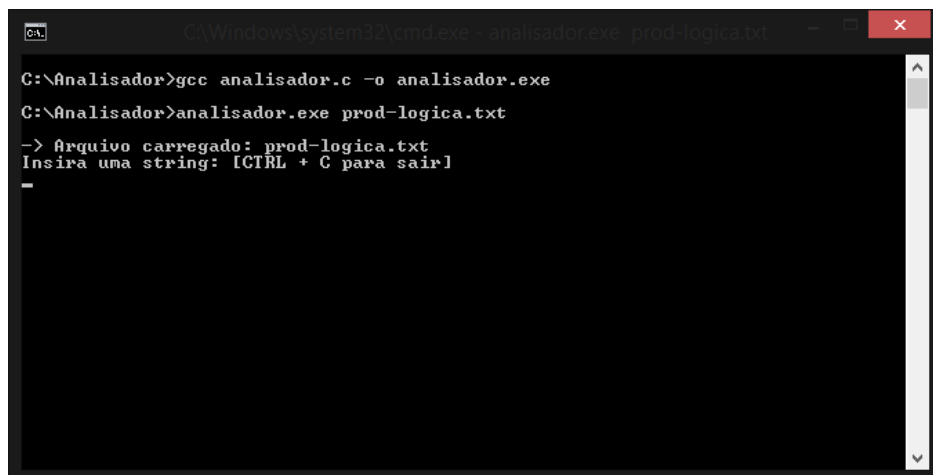
```
analizador.exe <nome-do-arquivo-de-producoes.txt> [Enter]
```

A. Exemplo:

Se deseja analisar a cadeia **!p**, devemos carregar as produções lógicas (pois a cadeia se trata de uma expressão lógica).

Digitaremos então:

```
analizador.exe prod-logica.txt [Enter]
```



```
C:\Windows\system32\cmd.exe - analizador.exe prod-logica.txt
C:\Analizador>gcc analisador.c -o analisador.exe
C:\Analizador>analizador.exe prod-logica.txt
-> Arquivo carregado: prod-logica.txt
Insira uma string: [CTRL + C para sair]
-
```

A partir daqui o programa já recebe a cadeia que se deseja analisar.

Por conta da dificuldade em representar as árvores de transições num arquivo de saída de análise de amostras, este programa somente exibe as árvores de transições na tela após a entrada de cadeias uma a uma, ou seja, a cada cadeia digitada e confirmada, a análise será feita e a árvore aparecerá ao fim da análise caso sejam bem sucedidas.

A partir daqui já podemos digitar as cadeias que queremos analisar. Como amostra digitamos então a cadeia **!p** e confirmamos com **[Enter]**.

```

C:\Windows\system32\cmd.exe - analisador.exe prod-logica.txt
C:\Users\Giovanne\Documents\DevCppProjects\Analizador - Proj1c>analizador.exe pr
od-logica.txt
-> Arquivo carregado: prod-logica.txt
Insira uma string: [CTRL + C para sair]
!p
FÓRMULA BEM FORMADA ACEITA!
Árvore!
+---+---+
| 0 | S |
+---+---+
| 1 | ! |
+---+---+
| 2 | S |
+---+---+
| 5 | p |
+---+---+
Insira uma string: [CTRL + C para sair]

```

Como dito, ao fim de toda análise bem sucedida (em que a entrada corresponda às regras de produções selecionadas) será exibida a árvore de transições. Somente as posições relevantes serão exibidas.

Como a árvore foi abstraída usando um vetor, decidi exibí-la por meio de uma tabela com duas colunas.

Na coluna da esquerda estarão os índices ou posições da árvore e na coluna da direita estarão os elementos contidos em cada posição.

No caso da última amostra (!p) temos como resultado a tabela:

```

+---+---+
| 0 | S |
+---+---+
| 1 | ! |
+---+---+
| 2 | S |
+---+---+
| 5 | p |
+---+---+

```

onde a raiz da árvore é o símbolo inicial **S**, que foi substituído pela produção **!S** e que, por fim, teve o **S** substituído por **p** compondo a cadeia.

3. Links pra download

GitHub: <<https://github.com/giovannealmeida/projc/tree/dev>>