

# Exercício-Programa: Implementação do algoritmo CYK para Gramáticas Livres do Contexto

## Objetivo:

Desenvolver um programa para processamento de Gramáticas Livres do Contexto (GLCs). Dada a especificação de uma GLC  $G$  na Forma Normal de Chomsky e uma cadeia  $w \in \Sigma^*$ , seu programa deve determinar se  $G$  gera  $w$ , e devolver a matriz *Tabela* (vide descrição do algoritmo) e o status (aceita / rejeita). Em outras palavras, o problema é decidir se  $w$  pertence a  $L(G)$ . O programa deverá ser desenvolvido em C ou em Java, e deverá ser executável via linha de comando do DOS.

## Entrada:

A chamada do programa será feita através de linha de comando, pelo nome:

Implementação em C:

glc.exe

Implementação em Java:

java glc

O programa terá os seguintes arquivos de entrada:

- **inp-glc.txt:**

Conterá a especificação da GLC (na Forma Normal de Chomsky).

A primeira linha conterá os campos:

$q$   $t$   $s$

onde  $q$  é o número de variáveis,  $t$  é o número de símbolos terminais e  $s$  é o número de regras de substituição. Note que todos esses parâmetros devem ser estritamente positivos.

A segunda linha conterá a lista de variáveis, separadas por espaços. A variável inicial deve aparecer em primeiro lugar na lista.

A terceira linha conterá a lista de terminais, separados por espaços.

As demais linhas (linhas 4 a  $s + 3$ ) conterão as regras de substituição, em uma das formas abaixo:

Variável  $>$  &

Variável  $>$  terminal

Variável  $>$  Variável Variável

Observações:

- Nas regras de substituição, as variáveis e símbolos serão separados por espaços;

- O símbolo reservado & representará a cadeia vazia.

O exemplo a seguir ilustra a especificação da GLC abaixo.

$$\begin{aligned}
 S_0 &\rightarrow \& \mid A T \mid B U \mid S S \mid A B \mid B A \\
 S &\rightarrow A T \mid B U \mid S S \mid A B \mid B A \\
 T &\rightarrow S B \\
 U &\rightarrow S A \\
 A &\rightarrow a \\
 B &\rightarrow b
 \end{aligned}$$

```

6 2 15
S0 S T U A B
a b
S0 > &
S0 > A T
S0 > B U
S0 > S S
S0 > A B
S0 > B A
S > A T
S > B U
S > S S
S > A B
S > B A
T > S B
U > S A
A > a
B > b

```

- **inp-cadeias.txt:**

Conterá as cadeias a serem avaliadas.

A primeira linha conterá um inteiro  $n$  representando o número de cadeias a serem processadas.

Cada uma das demais linhas conterá uma cadeia, onde os símbolos serão separados por espaços. Ex:

```

3
&
a
a b a a b b

```

Os arquivos de saída serão:

- **out-status.txt:**

Conterá uma única linha com os status (0=rejeita, 1=aceita) das cadeias de entrada. Esses dígitos serão separados por espaços.

Para o exemplo anterior, o arquivo conteria a linha:

```
1 0 1
```

- **out-tabela.txt:**

Conterá, para cada cadeia de entrada, a respectiva tabela de variáveis, conforme descrito no algoritmo.

A primeira linha conterá o número de cadeias de entrada. As demais linhas conterão os blocos referentes a cada cadeia.

Em cada bloco de cadeias, a 1ª linha conterá a cadeia original (onde os símbolos serão separados por espaços); para cada posição da tabela, será impressa uma linha contendo o formato:

*i j* <lista de variaveis separadas por espacos>

onde *i* e *j* representam, respectivamente, a linha e coluna da matriz. Note que apenas a porção superior da tabela precisa ser apresentada, e portanto  $j \geq i$ . Use espaços como separadores em cada linha.

Para o exemplo acima, o conteúdo desse arquivo será<sup>1</sup>:

```
3
&  <-- Para cadeia vazia, nao se imprime a matriz de variaveis
a  <-- Segunda cadeia
1 1 A
a b a b b  <-- Terceira cadeia
1 1 A      <-- Cada linha corresponde a uma posicao (i,j) da matriz
1 2 S0 S   <-- Separacao por espacos
1 3 U
1 4
1 5
1 6 S0 S
2 2 B
2 3 S0 S
2 4 U
2 5 S0 S
2 6 T
3 3 A
3 4
3 5
```

---

<sup>1</sup>Os comentários mostrados em out-tabela.txt são apenas para auxiliar a compreensão, e não deverão ser impressos

3 6 S0 S  
4 4 A  
4 5 S0 S  
4 6 T  
5 5 B  
5 6  
6 6 B

## Entrega do trabalho:

Condições da entrega:

- O trabalho poderá ser feito em grupos de ATÉ dois alunos, devidamente identificados na primeira linha do código-fonte.
- O prazo para entrega é 09/11/2014.
- Deverá ser entregue um diretório compactado (formato .zip) contendo o arquivo fonte e o executável (ou classe Java compilada). O diretório deve ser nomeado na forma *d<numerousp1>\_<numerousp2>.zip*. O módulo principal deve ter o nome glc.c ou glc.java.
- O código-fonte deverá ser compilável via comando gcc ou javac. Se desenvolver seu programa em IDEs como Eclipse ou Netbeans, certifique-se de que seu programa seja compilável sob as condições aqui expostas.
- Inclua também, no diretório de seu ep, um arquivo chamado LEIAME.TXT contendo o comando de chamada do compilador (gcc ou javac) e os argumentos necessários para compilação. Em outras palavras, o arquivo LEIAME.TXT deverá conter a linha de comando exata a ser digitada no prompt do sistema operacional para compilar seu programa.
- O trabalho deverá ser enviado para marcelolauretto@usp.br com o assunto ACH2043-EP1. No corpo da mensagem e nas 1as linhas do código fonte deverá constar os nomes e números USP dos membros do grupo.

Não é necessário que os dois alunos do grupo enviem o código-fonte, basta um e-mail por grupo.

- Dúvidas a respeito das especificações ou a respeito da implementação do trabalho serão sanadas até o dia 05/11. Dúvidas encaminhadas após este prazo serão ignoradas.
- Além da correção do programa, será considerada a qualidade da documentação do código fonte.
- Se houver evidência de plágio entre trabalhos de grupos distintos, os mesmos serão desconsiderados.