

ACH2043 - Introdução à Teoria da Computação
Exercício de Programação 2: **Simplificação de GLC'S**
Data de entrega: 31/07/2021

1 Objetivo

Desenvolver um programa para simplificação de Gramáticas Livres do Contexto (GLCs). Dada a especificação de uma ou mais GLC's, que podem ou não estar simplificadas, deve-se simplificá-las e imprimir sua nova especificação.

O algoritmo de simplificação de GLC's, conforme descrito na aula 17, consiste em quatro passos

1. Eliminação de as regras vazias;
2. Eliminação de produções unitárias;
3. Eliminação de símbolos inúteis;
4. Eliminação de símbolos inacessíveis.

O programa receberá um arquivo de entrada denominado `entrada.txt` e retornará um arquivo de saída denominado `saida.txt`. Esses dois nomes deverão ser *hard-coded*, e deverá ser assumido que ambos serão lidos e escritos no mesmo diretório do programa executável.

2 Formato dos arquivos de entrada e saída

Assuma que a entrada a ser lida do arquivo `entrada.txt` estará no seguinte formato:

- a primeira linha conterá um valor inteiro positivo (n) que indica o número de gramáticas a serem simplificadas;
- a segunda linha conterá os campos v_1 , t_1 , r_1 , que são, respectivamente, o número de variáveis ou símbolos não-terminais, número de símbolos terminais e número de regras de substituições da gramática G_1 , todos separados por espaço (note que todos esses parâmetros são estritamente positivos);
- a terceira linha deve conter a lista das v_1 variáveis ou símbolos não terminais da gramática G_1 , separados por espaço (cada variável sendo representada por um ou mais caracteres concatenados sem espaço);
- a quarta linha deve conter a lista dos t_1 símbolos terminais da gramática G_1 , separados por espaço (cada símbolo terminal sendo representada por um ou mais caracteres concatenados sem espaço e **não contendo o caracter &, que será utilizado como cadeia vazia**);
- as linhas 5 a $5 + r_1 - 1$ conterão as regras de substituição da gramática G_1 na forma:

$$Variavel \Rightarrow lado_direito$$

sendo *Variavel* um símbolo não terminal necessariamente contido na lista de símbolos não terminais definida acima, e *lado_direito* sendo a cadeia vazia (&) ou uma sequência de um ou mais símbolos terminais e não terminais separados por espaços. Exemplos:

$S \Rightarrow a S b$

$PREDICADO \Rightarrow VERBO \text{ uma nota } ADJETIVO$

Observação 1: Caso queira representar várias produções de um mesmo símbolo não terminal, cada alternativa será sempre descrita como uma regra, tanto no arquivo de entrada quanto no arquivo de saída. Exemplo: $S \Rightarrow a S b \mid S b$ será escrito como:

$S \Rightarrow a S b$

$S \Rightarrow S b$

Observação 2: Embora em aula e nos exemplos aqui apresentados utilizamos letras maiúsculas para símbolos não terminais e minúsculas para terminais, não há tal restrição. A única restrição é que a intersecção entre esses dois conjuntos de símbolos deve ser vazia.

- Se $n > 1$, as próximas gramáticas serão descritas de forma semelhante nas linhas imediatamente seguintes.

A saída deve ser impressa no arquivo `saida.txt` que deve seguir exatamente o mesmo formato do arquivo de entrada, mas contendo as gramáticas simplificadas na mesma ordem apresentada no arquivo de entrada.

Exemplo de entrada

Seja uma Gramática Livre de Contexto (GLC), uma quádrupla (V, Σ, S, P) , na qual V é o conjunto de símbolos não-terminais (variáveis), Σ é o conjunto de símbolos terminais, S é o símbolo inicial e P é o conjunto de produções. Considere um exemplo contendo duas gramáticas.

A primeira GLC é definida por $G_1 = (\{ S, A, B, C \}, \{ a, b, c \}, R)$ com R :

$S \Rightarrow A B \mid S C B$

$A \Rightarrow B$

$B \Rightarrow b B \mid b$

$C \Rightarrow c C \mid \&$

A segunda GLC é definida por $G_2 = (\{ X, Y, D, E \}, \{ x, y, d, e \}, R)$ com R :

$X \Rightarrow e D$

$Y \Rightarrow x Y x \mid X y D$

$D \Rightarrow y d y \mid \&$

$E \Rightarrow e$

A entrada será:

2

4 3 7

S A B C

a b c

$S \Rightarrow A B$

$S \Rightarrow S C B$

$A \Rightarrow B$

$B \Rightarrow b B$
 $B \Rightarrow b$
 $C \Rightarrow c C$
 $C \Rightarrow \&$
 $4 \ 4 \ 6$
 $X \ Y \ D \ E$
 $x \ y \ d \ e$
 $X \Rightarrow e \ D$
 $Y \Rightarrow x \ Y \ x$
 $Y \Rightarrow X \ y \ D$
 $D \Rightarrow y \ d \ y$
 $D \Rightarrow \&$
 $E \Rightarrow e$

3 Erros

Em caso de erros na Gramática deve ser impresso um código de erro conforme descrito na tabela abaixo:

| Erro | Código |
|---|--------|
| Os parâmetros n , v_i , t_i ou r_i não são positivos | ERRO 1 |
| Existe um símbolo que está tanto no conjunto dos símbolos terminais quanto no dos não-terminais | ERRO 2 |
| Quando do lado esquerdo da produção tem um símbolo terminal | ERRO 3 |
| Quando na regra de produção existe um símbolo que não foi definido | ERRO 4 |
| Quando do lado esquerdo existe mais de um símbolo (que não seja uma cadeia definida) | ERRO 5 |

Quando o erro for no parâmetro n , o programa deve sair imprimindo apenas “ERRO 1” na saída. Os demais erros, por serem específicos de uma gramática, deve inviabilizar apenas o processamento da gramática com descrição falha, com a devida saída de erro.

Assim, se por exemplo o arquivo de entrada apresentar 3 gramáticas e apenas a segunda contiver problemas de descrição, a saída deve conter algo como:

3
<linhas referentes à saída da gramática 1>
ERRO x
<linhas referentes à saída da gramática 3>

sendo que “x” é o código do primeiro erro observado na descrição de entrada (não precisa identificar os demais).