

Linguagens Formais e Autômatos
Prof. Andrei Rimsa Álvares

Trabalho Prático

1. Objetivo

O objetivo desse trabalho é permitir que os alunos apliquem os conceitos assimilados na disciplina em um trabalho prático de implementação. A ideia é desenvolver um dos algoritmos em uma das máquinas/gramáticas vistas na disciplina em um programa de computador.

2. Descrição

Implementar um programa que recebe uma gramática livre de contexto (GLC) G e a transforme em uma GLC G' equivalente na forma normal de Chomsky. O programa pode ser desenvolvido em qualquer linguagem de programação, desde que exista compilador/interpretador gratuito disponível para ela.

3. Instruções

O programa deve receber uma especificação de uma gramática livre de contexto $G = (V, \Sigma, R, P)$ no formato JSON conforme a seguinte especificação:

```
{ "glc": [  
  [v, ∀v ∈ V],  
  [a, ∀a ∈ Σ],  
  [ [X, w], X ∈ V, w ∈ (V ∪ Σ)* ],  
  P ∈ V  
] }
```

Considere a gramática livre de contexto G dada a seguir à esquerda com sua representação em formato JSON à direita, onde # representa λ :

$P \rightarrow ABC \mid bCC$
$A \rightarrow aAA \mid BB$
$B \rightarrow \lambda$
$C \rightarrow ABC \mid b$

G

```
{ "glc": [  
  ["P", "A", "B", "C"],  
  ["a", "b"],  
  [  
    ["P", "ABC"],  
    ["P", "bCC"],  
    ["A", "aAA"],  
    ["A", "BB"],  
    ["B", "#"],  
    ["C", "ABC"],  
    ["C", "b"]  
  ],  
  "P"  
] }
```

G_{json}

O programa deve receber um arquivo de entrada com uma GLC no formato especificado e deve exibir uma GLC equivalente na forma normal de Chomsky. Para colocar uma gramática nessa forma normal, deve-se seguir os seguintes passos. Não é preciso considerar o caso onde a regra de partida é recursiva.

**Linguagens Formais e Autômatos**

Prof. Andrei Rimsa Álvares

1) Eliminar regras λ .

$P \rightarrow ABC \mid bCC$
$A \rightarrow aAA \mid BB$
$B \rightarrow \lambda$
$C \rightarrow ABC \mid b$

Antes

$P \rightarrow ABC \mid BC \mid AC \mid C \mid bCC$
$A \rightarrow aAA \mid aA \mid a \mid BB \mid B$
$B \rightarrow$
$C \rightarrow ABC \mid BC \mid AC \mid b$

Depois

2) Eliminar regras unitárias.

$P \rightarrow ABC \mid BC \mid AC \mid C \mid bCC$
$A \rightarrow aAA \mid aA \mid a \mid BB \mid B$
$B \rightarrow$
$C \rightarrow ABC \mid BC \mid AC \mid b$

Antes

$P \rightarrow ABC \mid BC \mid AC \mid bCC \mid b$
$A \rightarrow aAA \mid aA \mid a \mid BB$
$B \rightarrow$
$C \rightarrow ABC \mid BC \mid AC \mid b$

Depois

3) Eliminar variáveis inúteis.

$P \rightarrow ABC \mid BC \mid AC \mid bCC \mid b$
$A \rightarrow aAA \mid aA \mid a \mid BB$
$B \rightarrow$
$C \rightarrow ABC \mid BC \mid AC \mid b$

Antes

$P \rightarrow AC \mid bCC \mid b$
$A \rightarrow aAA \mid aA \mid a$
$C \rightarrow AC \mid b$

Depois

4) Modificar cada regra $X \rightarrow w$, $|w| \geq 2$, de forma que ela contenha apenas variáveis.

$P \rightarrow AC \mid bCC \mid b$
$A \rightarrow aAA \mid aA \mid a$
$C \rightarrow AC \mid b$

Antes

$P \rightarrow AC \mid BCC \mid b$
$A \rightarrow DAA \mid DA \mid a$
$C \rightarrow AC \mid b$
$B \rightarrow b$
$D \rightarrow a$

Depois

5) Substituir cada regra $X \rightarrow Y_1 Y_2 \dots Y_n$, $n \geq 3$, em que cada Y_i é uma variável, pelo conjunto de regras: $X \rightarrow Y_1 Z_1$, $Z_1 \rightarrow Y_2 Z_2$, ..., $Z_{n-2} \rightarrow Y_{n-1} Y_n$, em que Z_1, Z_2, \dots, Z_{n-2} são novas variáveis.

$P \rightarrow AC \mid BCC \mid b$
$A \rightarrow DAA \mid DA \mid a$
$C \rightarrow AC \mid b$
$B \rightarrow b$
$D \rightarrow a$

Antes

$P \rightarrow AC \mid BE \mid b$
$A \rightarrow DF \mid DA \mid a$
$C \rightarrow AC \mid b$
$B \rightarrow b$
$D \rightarrow a$
$E \rightarrow CC$
$F \rightarrow AA$

Depois

A seguir, é mostrado um exemplo de execução. Para a gramática G de entrada, é exibida a gramática G' na forma normal de Chomsky. Cada variável é formada por uma única letra maiúscula. Dada essa restrição, note que o alfabeto de símbolos não pode conter letras maiúsculas. **Nenhuma outra saída deve ser exibida, como mensagens de depuração por exemplo.**

Linguagens Formais e Autômatos
Prof. Andrei Rimsa Álvares

```
$ ./chomsky
Usar: ./chomsky [GLC]
$ ./chomsky G.json
{ "glc": [
  ["P", "A", "C", "B", "D", "E", "F"],
  ["a", "b"],
  [
    ["P", "AC"],
    ["P", "BE"],
    ["P", "b"],
    ["A", "DF"],
    ["A", "DA"],
    ["A", "a"],
    ["C", "AC"],
    ["C", "b"],
    ["B", "b"],
    ["D", "a"],
    ["E", "BC"],
    ["F", "AA"]
  ],
  "p"
]}
```

4. Avaliação

O trabalho deve ser feito em grupo de até dois alunos, sendo esse limite superior estrito. O trabalho será avaliado em 10 pontos, onde essa nota será multiplicada por um fator entre 0.0 e 1.0 para compor a nota de cada aluno individualmente. Esse fator poderá estar condicionado a apresentações presenciais a critério do professor. A avaliação é feita exclusivamente executando casos de testes criados pelo professor. Portanto, códigos que não compilam ou não funcionam serão avaliados com nota zero.

Trabalhos copiados, parcialmente ou integralmente, serão avaliados com nota **ZERO** do valor da prática, sem direito a contestação. Você é responsável pela segurança de seu código, não podendo alegar que outro grupo o utilizou sem o seu consentimento.

5. Submissão

O trabalho deverá ser submetido até as 23:59 do dia 07/02/2022 (segunda-feira) via sistema acadêmico em pasta específica. Não serão aceitos, em hipótese alguma, trabalhos enviados por e-mail ou por quaisquer outras fontes. Para trabalhos feitos em dupla, deve-se criar um arquivo README na raiz do projeto com os nomes dos integrantes da dupla. **A submissão deverá ser feita por apenas um dos integrantes da dupla.**