

UNIVERSIDADE ESTADUAL DO PARANÁ
CIÊNCIA DA COMPUTAÇÃO

GUILHERME HENRIQUE DE SOUZA
SARA SAORI SATAKE

DERIVAÇÃO DE GRAMÁTICAS REGULARES
RELATÓRIO TÉCNICO

APUCARANA – PR

2017

GUILHERME HENRIQUE DE SOUZA

SARA SAORI SATAKE

DERIVAÇÃO DE GRAMÁTICAS REGULARES

RELATÓRIO TÉCNICO

Relatório técnico do trabalho sobre
Derivação de Gramáticas Regulares,
apresentada na disciplina de
Linguagens Formais Autômatos para
o 2º ano de Ciência da Computação.

APUCARANA –PR

2017

Sumário

INTRODUÇÃO	4
OBJETIVOS.....	5
GRAMÁTICAS REGULARES	6
EXEMPLOS	7
DECISÕES DE PROJETO PARA A IMPLEMENTAÇÃO.....	9
REFERÊNCIAS	11

INTRODUÇÃO

A programação de computadores possui uma ampla e complexa cadeia de comandos que permite o funcionamento dessas máquinas de forma eficiente. A proximidade com as linguagens compreensíveis ao ser humano é então, reduzida, provocando o distanciamento desses dois universos.

Para diminuir esse impasse, foram criadas as linguagens de programação. Estas são linguagens formais, que visam eliminar ambiguidades, permitindo assim, que um comando e palavras reservadas tenham sempre o mesmo significado, independentemente de como estão colocadas em um programa.

Assim como a língua portuguesa, ou qualquer outro idioma, as linguagens de programação possuem uma gramática, que possibilitam a criação de programas válidos. A partir de uma gramática, é possível gerar todas as palavras de uma linguagem formal.

Neste trabalho, será apresentado a implementação de um programa que permite a derivação de uma palavras de uma Gramática Regular, ou seja, linguagens que podem ser reconhecidas por autômatos finitos.

As derivações poderão ser feitas em Gramática Regular Unitária à Direita (GLUD) e Gramática Regular Unitária à Esquerda (GLUE), sendo iminente também, observar o passo a passo das regras de produção tomadas para cada derivação.

OBJETIVOS

Desenvolver e implementar um programa que realize e mostre as derivações de uma palavra de uma Gramática Regular, havendo possibilidade de selecionar tanto a opção Gramática Linear Unitária à Direita (GLUD) quanto a Gramática Linear Unitária à Esquerda (GLUE).

Compreender de maneira prática os conhecimentos obtidos em sala de aula na disciplina de Linguagens Formais e Autômatos.

Analisar os conceitos teóricos e colocá-los em prática para aquisição de melhores resultados, conseguindo assim, um êxito na compreensão da matéria estudada.

Analisar as técnicas de programação e estrutura de dados responsáveis para o bom funcionamento do projeto.

GRAMÁTICAS REGULARES

Uma Gramática Regular é o mecanismo gerador para as linguagens regulares, ou seja, as linguagens que podem ser reconhecidas por Autômatos Finitos Determinísticos (AFD) ou Autômatos Finitos Não Determinísticos (AFND).

Uma gramática G é formada por uma quádrupla $\{V, T, P, S\}$, onde:

- ❖ V são variáveis utilizadas para a construção da gramática (maiúsculas);
- ❖ T são os terminais, ou o alfabeto da linguagem (minúsculas);
- ❖ P são regras de produção representadas por $(X \rightarrow Y)$;
- ❖ S é o símbolo de partida para as derivações ($S \in V$).

- GRAMÁTICA LINEAR UNITÁRIA À DIREITA (GLUD):

Uma Gramática Regular é dita GLUD se as regras de produção estão no formato

$$\begin{aligned} A &\rightarrow wB \\ A &\rightarrow w \end{aligned}$$

Tal que $|w| \leq 1$
 $A, B \in V$
 $w \in T^*$.

- GRAMÁTICA LINEAR UNITÁRIA À ESQUERDA (GLUE):

Uma Gramática Regular é dita GLUE se as regras de produção estão no formato

$$\begin{aligned} A &\rightarrow Bw \\ A &\rightarrow w \end{aligned}$$

Tal que $|w| \leq 1$
 $A, B \in V$
 $w \in T^*$.

EXEMPLOS

- GLUE:

```
Digite 0 para GLUE ou 1 para GLUD
0
Digite a quantidade de variáveis <U>:
3
Digite a 1ª variável:
S
Digite a 2ª variável:
A
Digite a 3ª variável:
B
Digite a quantidade de alfabeto da linguagem <T>:
2
Digite o 1º caractere do alfabeto:
a
Digite o 2º caractere do alfabeto:
b
Digite o símbolo de partida <S>:
S
Digite a quantidade de ordens de produções:
6
-----
=> Digite no padrão A>Bc
- Sendo A e B pertencentes a U e c pertencente a T
- Caso tenha algum branco, simbolizar com *
-----
Digite a 1ª ordem de produção:
S>Bb
Digite a 2ª ordem de produção:
S>Aa
Digite a 3ª ordem de produção:
A>Aa
Digite a 4ª ordem de produção:
A>**
Digite a 5ª ordem de produção:
B>Bb
Digite a 6ª ordem de produção:
B>Aa
Digite a palavra:
abbb
```

```
=====
= Palavra Aceita =
=====
Ordens de produção passadas pelo algoritmo:
<1>S>Bb
Bb
<5>B>Bb
Bbb
<5>B>Bb
Bbbb
<6>B>Aa
Aabbb
<4>A>**
abbb
```

- GLUD:

```

Digite 0 para GLUE ou 1 para GLUD
1
Digite a quantidade de variáveis <U>:
3
Digite a 1ª variável:
S
Digite a 2ª variável:
A
Digite a 3ª variável:
B
Digite a quantidade de alfabeto da linguagem <T>:
2
Digite o 1º caractere do alfabeto:
a
Digite o 2º caractere do alfabeto:
b
Digite o simbolo de partida <S>:
S
Digite a quantidade de ordens de produções:
6

```

```

-----
=> Digite no padrão A>bC
- Sendo A e C pertencentes a U e b pertencente a T
- Caso tenha algum branco, simbolizar com *
-----

```

```

Digite a 1ª ordem de produção:
S>aA
Digite a 2ª ordem de produção:
A>aA
Digite a 3ª ordem de produção:
A>bB
Digite a 4ª ordem de produção:
A>**
Digite a 5ª ordem de produção:
B>bB
Digite a 6ª ordem de produção:
B>**
Digite a palavra:
aaab

```

```

=====
= Palavra Aceita =
=====
Ordens de produção passadas pelo algoritmo:
<1>S>aA
aA
<2>A>aA
aaA
<2>A>aA
aaaA
<3>A>bB
aaabB
<6>B>**
aaab

```


DECISÕES DE PROJETO PARA A IMPLEMENTAÇÃO

A linguagem de programação escolhida foi Java, fazendo-se uso do compilador NetBeans IDE 8.2.

Ao iniciar-se o programa, foi implementado a opção de escolha por parte do usuário se o mesmo deseja optar por GLUE ou GLUD. Em seguida, deve-se informar a quantidade de variáveis (V), bem como quais serão elas; podendo assim, serem armazenadas em um array. O mesmo ocorre para o alfabeto da linguagem (T). O símbolo para representar o branco foi pré-definido como “*”, não havendo possibilidade de alterações. Deve-se informar também um símbolo de partida (esse deve estar contido em V), e a quantidade, seguido de quais serão as ordens de produção (P), as quais devem possuir tamanho 4. No caso do GLUD, o primeiro e o quarto índices devem estar contidos em V e o terceiro, em T. Já no GLUE, o primeiro e terceiro índices devem estar contidos em V e o quarto, em T. Em ambos os casos, o segundo índice deve ser obrigatoriamente “>”. Em situações de apenas uma variável (V) ou um caractere do alfabeto (T), deve-se completar com branco (*). Todas as opções passam por verificações antes de serem armazenadas e prosseguem para a próxima instrução.

Assim que a palavra a ser testada é inserida, verifica-se se possui alguma das variáveis anteriormente informadas. Caso possua, a palavra é automaticamente rejeitada, evitando percorrer desnecessariamente todo o algoritmo. No caso do GLUE, a palavra é primeiramente invertida antes de prosseguir no algoritmo propriamente dito. Após isso, verifica-se se pode ser aceita ou não.

Para verificar as possibilidades de derivações, é feita uma comparação entre o nó atual com o primeiro índice das ordens de produção e o índice da palavra com o segundo índice das ordens de produção, ou terceiro índice no caso do GLUE. Após a obtenção dessas possibilidades, armazenam-se em um array bidimensional, sendo um dos índices a altura do nó e o outro, a possibilidade. No caso da presença de mais de uma possibilidade, o algoritmo consta com variáveis destinadas ao backup de informações como altura do nó, quantidade de possibilidades, possibilidade atual, ordens de produção realizadas, e até mesmo o resultado até aquela interação. Com a posse desses dados, é possível o deslocamento na árvore em ambos os sentidos.

Os vetores de controle, armazenamento e auxílio foram pré-estabelecidas com tamanho 9999 e altura do nó em 500, fazendo-se assim, um limite para o tamanho das palavras. Foram utilizadas também, variáveis do tipo int, boolean, String ao decorrer do algoritmo. As opções GLUE e GLUD agem separadamente, não havendo interferências uma com a outra.

A cada ciclo, é verificado se o resultado gerado é igual à palavra. Caso seja, o ciclo é encerrado, e as ordens de produção são exibidas.

CONCLUSÃO

Com a realização deste trabalho, os conceitos sobre Gramática Regular foram melhores compreendidos, havendo melhor aproveitamento do conteúdo estudado no bimestre.

Foi possível o entendimento e análise da importância da manipulação de índices de vetores, bem como seu real funcionamento em conjunto de regras de produções, tornando-se possível alcançar os objetivos propostos.

A diferença da lógica aplicada em derivações GLUD e GLUE foram esclarecidas, assim como as limitações de cada uma.

A linguagem de programação foi, sem dúvida, um grande marco para a história da computação. Em adição, as gramáticas implementadas são de extrema importância para o bom funcionamento do mesmo.

As gramáticas possuem um poder de decodificação poderoso, que podem ser o diferencial para a variação de uma linguagem formal a outra. Os estudos aplicados à esta área são inacabáveis, visto que possui uma grande magnitude em relação às linguagens de programação e compiladores, as quais estão em constante desenvolvimento.

REFERÊNCIAS

BROOKSHEAR, J.G. **Ciência da computação**. Porto Alegre: Bookman, 2000.

SIPSER M. **Introdução à Teoria da Computação**. 2 ed. Cengage Learning.2007.

PUC-RIO. **Linguagens formais e autômatos (lfa)**. Disponível em:
<<http://www.inf.puc-rio.br/~inf1626/docs/2013/slides/lfa-aula10.pdf>>. Acesso em: 20 set. 2017.