Insper

Lógica da Computação - 2019/2

Aula 04/T02 - 14/Aug/2019

Raul Ikeda - rauligs@insper.edu.br

Objetivos

1. Gramáticas Regulares
2. Autômatos Finitos

Retomando a gramática do compilador:

Qual é a linguagem?

Que outras características podemos observar nas regras de produção?

Definindo uma Gramática Regular

Definição: Uma gramática é dita regular se as regras de produção possuem a seguinte forma:

$$\alpha \rightarrow \beta\gamma \ ou \ \alpha \rightarrow \gamma\beta$$

Onde:

- $\alpha \in N = V \cap \bar{\Sigma}$
- $\beta \in \Sigma^*$
- $\gamma \in N \cup \phi$

Ou seja, um único **não terminal** só pode ser substituído por uma **cadeia de terminais** e até um **único não terminal**.

- Regras $\alpha \rightarrow \beta\gamma$ produzem uma **Gramática Linear à Direita**.
- Regras $\alpha \rightarrow \gamma\beta$ produzem uma **Gramática Linear à Esquerda**.

Portanto a gramática do compilador, por enquanto, é:

Exemplos de Aplicações:

- Verificação do comportamento de circuitos digitais
- O próprio analisador léxico em que estamos trabalhando
- Procura de palavras ou frases em um texto ou string em geral
- Protocolo de comunicação de dispositivos (por exemplo IoT)

Outros exemplos (não tão aplicados):

- $L(G_1) = \{a^n | n \geq 0\}$

- $L(G_2) = \{a^m b^n | m \geq 0, n \geq 0\}$

- $L(G_3) = \{a^m b^m | m \geq 0\}$

Equivalências entre GR

> **Teorema**: "Se G_1 é uma gramática linear à direita, então existe uma gramática linear à esquerda G_2 tal que $L(G_1) = L(G_2)$, e vice versa". Ver: Ramos et al Pag 142.

Ideia:

```
>> Ver Pag. 142 Ramos et al
```

Exemplo Prático: RegExp - Regular Expression

Muito utilizado para pesquisar em um texto ou validar se uma substring segue um determinado padrão.

Exemplo: Simples validador de E-mail.

testar: https://www.regextester.com/
Mais detalhes: https://en.wikipedia.org/wiki/Regular_expression

Gramática vs RegExp

Escreva uma gramática regular equivalente à expressão regular anterior.

Será que as expressões regulares podem nos ajudar a representar uma linguagem de programação?

Validando uma cadeia

1. Como faço para reconhecer automaticamente se uma cadeia pertence a uma gramática regular? Será que é preciso fazer um Analisador Sintático para validar?

Autômatos Finitos Determinísticos - AFD

Um AFD atesta que uma cadeia é aderente a uma gramática se ao terminar de processar a cadeia o estado atual pertence ao conjunto de estados finais. Implicações:

- Todo estado deve ser capaz de processar todos os símbolos do alfabeto
- Cada símbolo só pode estar atrelado apenas à uma regra de transição
- Não pode haver transições em vazio (λ)

`>> Ver JFLAP: jflap.org`

Formalizando

Um autômato finito é uma quíntupla:

$$M = (Q, \Sigma, \delta, q_0, F)$$

Onde:

- Q é um conjunto finito de estados.
- Σ é o alfabeto (finito e não vazio) de entrada.
- δ é um conjunto de funções de transição. $\delta : Q \times \Sigma \rightarrow Q$
- q_0 é o estado inicial. $q_0 \in Q$
- F é conjunto de estados finais. $F \subseteq Q$

Enfim, autômato é ou não um computador?

Equivalência entre GR e AF

> **Teorema**: "Seja G uma gramática linear à direita. Então é possível definir um autômato finito M de tal modo que L(G) = L(M)" Ver: Ramos et al Pag. 202.

Algoritmo:

1. Cada símbolo não terminal vira um estado do autômato.
2. O símbolo raiz da gramática vira o estado inicial.
3. Adicionar um estado final Z ao conjunto de estados.
4. Para cada regra de produção, fazer a seguinte transformação:
   - Se $X \rightarrow aY$ então $(X, a) \rightarrow Y$
   - Se $X \rightarrow Y$ então $(X, \lambda) \rightarrow Y$
   - Se $X \rightarrow a$ então $(X, a) \rightarrow Z$
   - Se $X \rightarrow \lambda$ então $(X, \lambda) \rightarrow Z$

Exemplo: Usando a gramática do compilador.

Wait! Esse autômato tem transições em vazio?
R: É possível (e saudável) eliminá-los. Ver: Ramos et al Pag. 169.
Pode haver estados inacessíveis ou inúteis?
R: Ver Ramos et al Pag. 179.

Autômato melhorado:

Quais as conclusões desse exemplo?

TGF: Minimização de Autômatos. Ramos et al Pag. 216.

O Nosso Estado atual

Lista de Exercícios

1. Construa um AFD que aceite a seguinte expressão regular: $a(a|b)^*aaa$.
2. Escreva a gramática regular equivalente ao item 1.
3. Dada a gramática $G = (\{A, B, C, a, b\}, \{a, b\}, P, A)$, sendo:

$$P = \begin{cases} A \to aB \\ B \to bA \\ B \to C \\ C \to aC \\ C \to \lambda \end{cases}$$

a) Explique porque ela é regular.

b) Escreve uma gramática linear à esquerda equivalente.
c) Monte um Autômato que reconheça a gramática acima.
d) Transforme o autômato do item c em um AFD.
e) Escreve a linguagem e a expressão regular equivalente.

4. Como você demonstra que duas gramáticas são iguais?

Próxima aula: J. J. Neto - Cap. 4.2

1. Expansão do Compilador: novos operadores.
2. Erros sintáticos
3. Comentários