Lógica da Computação - 2020/1

Aula 02/T01 - 17/Feb/2020

Raul Ikeda - rauligs@insper.edu.br

Objetivos

1. Gramáticas e Linguagens
2. Teoria dos Conjuntos

Discussão - 15 min.

Artigo: Neto, J. J. A Teoria da Computação e o profissional de informática. Revista de Computação e Tecnologia da PUC-SP, vol. 1, 2009.

- Apresentação de fatos históricos
- Introdução de conceitos.
- Conclusão.

Linguagem de Programação?

Podemos definir uma LP como um conjunto estruturado de instruções para um computador.

- Assim como uma Linguagem Natural (LN), existem regras gramaticais?
- Como é definida essa gramática?
- O que diferencia uma LP de uma LN?

Vamos aprender uma nova Linguagem

Passo I: Alfabeto

> **Definição**: O **Alfabeto** é um conjunto de símbolos (aka átomos ou tokens) que deve ser ***finito*** e ***não vazio***, normalmente representado pela letra Σ.

Exemplo: Se $\Sigma = \{0, 1, ..., 9, +, -, if, while, def\}$, as possíveis palavras do alfabeto são:

- 0123
- 0+if
- ifwhiledef-

Passo II: Palavras

> **Definição**: Palavras ou **Cadeias** são ***concatenações*** finitas de símbolos de um determinado alfabeto.

1. Considerando $\Sigma = \{0, 1\}$, quais são todas as cadeias desse alfabeto?

Outras definições importantes:

- λ ou ϵ representa uma cadeia vazia.
- Σ^* é chamado de Fechamento Reflexivo e Transitivo, ou **Fecho de Kleene**.
- Σ^+ é chamado de Fechamento Transitivo.

  Definição: $\Sigma^* = \Sigma^+ + \lambda$

Passo III: Linguagem

Definição: Dado certo alfabeto Σ, uma **Linguagem** é um subconjunto de Σ^*, ou seja, um **conjunto de cadeias**.

Passo III.a: Revisando Teoria dos Conjuntos

Considere os seguintes conjuntos:

$$A = \{a, b\}$$
$$B = \{a, b, c\}$$
$$C = \{\} = \phi$$

- **Conjunto** é uma coleção de elementos não indexados. Não pode haver elementos repetidos.
- ϕ representa o conjunto vazio, ou seja, sem elementos.
- O tamanho de um conjunto é dado pela quantidade de elementos:

$$|A| = 2$$
$$|B| = 3$$
$$|C| = |\phi| = 0$$

- $\in$ indica quando um elemento **pertence** a um conjunto:

$$a \in A$$
$$c \notin A$$
$$\{a\} \notin A$$

- $\subseteq$ indica quando um conjunto está **contido ou é igual** a outro conjunto:

$$A \subseteq B$$
$$c \nsubseteq B$$
$$\{c\} \subseteq B$$

  - A está contido em B, ou ainda B contém A.
  - ϕ está contido em qualquer conjunto.

- Um conjunto é **igual** ao outro sse:

$$A = B \leftrightarrow A \subseteq B \ e \ B \subseteq A$$

- A **união** de dois conjuntos é dada por:

$$A \cup B = \{x | x \in A \ ou \ x \in B\}$$
$$A \cup B = \{a, b, c\} = B$$

- A **intersecção** de dois conjuntos é dada por:

$$A \cap B = \{x | x \in A \ e \ x \in B\}$$
$$A \cap B = \{a, b\} = A$$

- A **diferença** de dois conjuntos é dada por:

$$A - B = \{x | x \in A \ e \ x \notin B\}$$
$$A - B = \phi$$
$$B - A = \{c\}$$

- O **produto cartesiano** de dois conjuntos é dado por:

$$A \times B = \{(x, y)|x \in A \ e \ y \in B\}$$
$$A \times B = \{(a, a), (a, b), (a, c), (b, a), (b, b), (b, c)\}$$
$$|A \times B| = |A||B|$$
$$A \times B \neq B \times A$$

- A **potência** de um conjunto é dada por:

$$A^0 = \phi$$
$$A^2 = A \times A$$
$$A^n = A^{n-1} \times A$$

- O **power set** de um conjunto é dado por:

$$2^A = \{Z|Z \subseteq A\}$$
$$2^A = \{\phi, \{a\}, \{b\}, \{a, b\}\}$$
$$|2^A| = 2^{|A|} = 4$$

Paradoxo de Russell

$$A = \{B|B \ \acute{e} \ um \ conjunto \ e \ B \notin B\}$$

`>> Ver Pag. 31 Hammack`

Q&A:

1. Espera um pouco, cadeias são conjuntos?
   R: Não, cadeias são elementos de uma linguagem, que é um conjunto.
2. Então λ não é um conjunto vazio (ϕ)?
   R: Exato, o λ é uma cadeia vazia (elemento), já ϕ é o conjunto vazio.
3. Quais são as operações que podemos realizar com cadeias?
   R: Sejam: $s = 111$ e $t = 00$
   - Tamanho: $|s| = 3$; $|t| = 2$; $|\lambda| = 0$
   - **Concatenação**: $st = 11100 = \lambda st = st\lambda \neq ts = 00111$
   - Potência: $s = 1^3 = 11^2 = 1^3 1^0$

Exemplos abstratos de linguagem:

- $L_1 = \{a^n b^n | n \geq 0\} =$
- $L_2 = \{ab^i | i \geq 0\} =$
- $L_3 = \{s \in \{a, b\}^* | 0 \leq |s| \leq 2\} =$

Concluindo: para obter uma linguagem basta misturar um alfabeto?

Passo IV: Gramática

$$G = (V, \Sigma, P, S)$$

Onde:

- V é o vocabulário da linguagem, onde $V = N \cup \Sigma$ e N é o conjunto de **símbolos não terminais** usados para representar estados intermediários nas regras de produção. N é composto por letras maiúsculas.
- Σ é o alfabeto de **símbolos terminais**. São os símbolos na qual as cadeias da linguagem são construídas.
- P é o conjunto das regras de produção
- S é o símbolo não terminal inicial da gramática

Exemplo:

$$G = (V = \{A, B, 0, 1\}, \Sigma = \{0, 1\}, P, S = A)$$

$$P = \begin{cases} A \rightarrow B \\ B \rightarrow 1A \\ B \rightarrow 0 \end{cases}$$

1. Quais as possíveis construções (cadeias) dessa gramática?

2. Escreva uma representação de uma linguagem definida pela gramática.

E o Compilador no fim das contas?

- O compilador ou interpretador reconhece uma linguagem através da análise sintática. Ele verifica se a cadeia pertence ou não a uma certa gramática.
- Ele realiza uma análise léxica para separar as cadeias da linguagem. Essas cadeias irão alimentar o analisador sintático. Essa etapa é também chamada de tokenização.
- O compilador verifica se o programa faz sentido, ou seja, apesar de impecável sintaticamente, ele pode não fazer sentido (tipo errado, variável não declarada, etc). Essa etapa é chamada de análise semântica.
- Ele traduz uma linguagem de programação para instruções de máquina (mneumônicos) ou instruções intermediárias para uma Virtual Machine (JVM, CIL.net, LLVM, etc).

Lista de Exercícios

1. Seja $\Sigma = \{+, -\}$ Escreva:

- Σ^*
- Σ^+
- Σ^0
- Σ^2

2. Seja $L(G) = \{s \in \{a, b\}^* | s = a^n b, n \geq 0\}$. Escreva uma gramática que represente L(G).
3. Seja $L(G) = \{s \in \{a, b, c\}^* | s = a^m bc^n, m \geq 0, n \leq 1\}$

- $'ab' \in L(G)$?
- $'abbc' \in L(G)$?
- Qual a menor cadeia?
- Escreva uma gramática para a linguagem.

4. (Ramos et al Pag 137) Considere o alfabeto $\Sigma = \{a, b\}$. Proponha gramáticas diferentes G_1 e G_2 que gerem linguagens sobre esse alfabeto, de tal forma que:

- $G_1 \neq G_2$
- $L_1(G_1) \subseteq \Sigma^*$
- $L_2(G_2) \subseteq \Sigma^*$
- L_1 seja infinita
- L_2 seja infinita

Adicionalmente:

a) $L_1 \cap L_2 = \phi$
b) $L_1 \subseteq L_2$ e $L_1 \neq L_2$
c) $L_1 = \Sigma^* - L_2$
d) $L_1 = L_2 = \Sigma^*$
e) $L_1 \cap L_2 = (ab)*$
f) $L_1 - L_2 = \{a, ab, b\}$
g) $L_1 \cup L_2 = \Sigma^*$ e $L_1 \cap L_2 = \phi$

Próxima aula:

- Diagrama Sintático
- Léxico: tokenização
- Embrião do Sintático

Referências:

- J. J. Neto - Pag 78-82, Pag 159