Lógica da Computação - 2021/1

Aula 02/T01 - 24/Feb/2021

Maciel Calebe Vidal - macielcv@insper.edu.br

# Objetivos

1. Gramáticas e Linguagens
2. Teoria dos Conjuntos

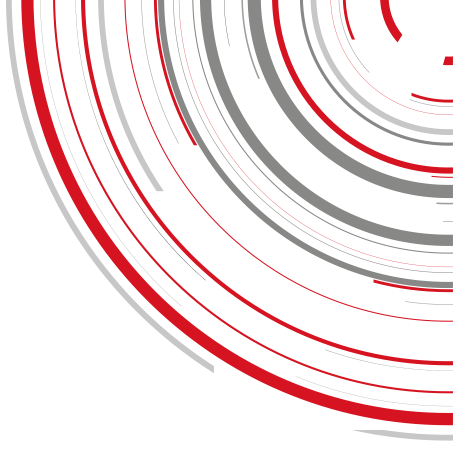
# Apresentação do Curso

Apresentação PDF publicada no Blackboard.

* Objetivos de Aprendizagem
* Conteúdos
* Critérios de Avaliação
* Composição da Nota

**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 Σ = {0,1,...,9,+,−,𝑖𝑓,𝑤ℎ𝑖𝑙𝑒,𝑑𝑒𝑓}, as possíveis palavras do alfabeto são:

* 0123 Resp: 4
* 0+if Resp: 3
* ifwhiledef- Resp: 4

# Passo II: Palavras

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

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

{ 𝜆, 0, 1, 00, 01, 10, 11, …}

Ou seja, existem infinitas cadeias.

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**: Σ∗ = Σ+ + 𝜆

# 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 = {} = 𝜙

* **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| = |𝜙| = 0

* ∈ indica quando um elemento **pertence** a um conjunto:

𝑎 ∈ 𝐴

𝑐 ∉ 𝐴

{𝑎} ∉ 𝐴

* ⊆ indica quando um conjunto está **contido ou é igual** a outro conjunto:

𝐴 ⊆ 𝐵

𝑐 ⊈ 𝐵 {𝑐} ⊆ 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 ↔ 𝐴 ⊆ 𝐵 𝑒 𝐵 ⊆ 𝐴

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

𝐴 ∪ 𝐵 = {𝑥|𝑥 ∈ 𝐴 𝑜𝑢 𝑥 ∈ 𝐵}

𝐴 ∪ 𝐵 = {𝑎,𝑏,𝑐} = 𝐵

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

𝐴 ∩ 𝐵 = {𝑥|𝑥 ∈ 𝐴 𝑒 𝑥 ∈ 𝐵}

𝐴 ∩ 𝐵 = {𝑎,𝑏} = 𝐴

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

𝐴 − 𝐵 = {𝑥|𝑥 ∈ 𝐴 𝑒 𝑥 ∉ 𝐵}

𝐴 − 𝐵 = 𝜙

𝐵 − 𝐴 = {𝑐}

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

𝐴 × 𝐵 = {(𝑥,𝑦)|𝑥 ∈ 𝐴 𝑒 𝑦 ∈ 𝐵}

𝐴 × 𝐵 = {(𝑎,𝑎),(𝑎,𝑏),(𝑎,𝑐),(𝑏,𝑎),(𝑏,𝑏),(𝑏,𝑐)}

|𝐴 × 𝐵| = |𝐴||𝐵|

𝐴 × 𝐵 ≠ 𝐵 × 𝐴

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

𝐴0 = 𝜙

𝐴2 = 𝐴 × 𝐴 𝐴𝑛 = 𝐴𝑛−1 × 𝐴

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

2𝐴 = {𝑍|𝑍 ⊆ 𝐴}

2𝐴 = {𝜙,{𝑎},{𝑏},{𝑎,𝑏}}

|2𝐴| = 2|𝐴| = 4

# Paradoxo de Russell

𝐴 = {𝐵|𝐵 𝑒́ 𝑢𝑚 𝑐𝑜𝑛𝑗𝑢𝑛𝑡𝑜 𝑒 𝐵 ∉ 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.

1. Então 𝜆 não é um conjunto vazio (𝜙)?

R: Exato, o 𝜆 é uma cadeia vazia (elemento), já 𝜙 é o conjunto vazio.

1. Quais são as operações que podemos realizar com cadeias?

R: Sejam: 𝑠 = 111 e 𝑡 = 00

* + Tamanho: |𝑠| = 3; |𝑡| = 2; |𝜆| = 0
  + **Concatenação**: 𝑠𝑡 = 11100 = 𝜆𝑠𝑡 = 𝑠𝑡𝜆 ≠ 𝑡𝑠 = 00111
  + Potência: 𝑠 = 13 = 112 = 1310

Exemplos abstratos de linguagem:

* + 𝐿1 = {𝑎𝑛𝑏𝑛|𝑛 ≥ 0} = {𝜆 , ab, aabb, aaabbb, …}
  + 𝐿2 = {𝑎𝑏𝑖|𝑖 ≥ 0} = {a, ab, abb, abbb, …}
  + 𝐿3 = {𝑠 ∈ {𝑎,𝑏}∗|0 ≤ |𝑠| ≤ 2} = {𝜆, a, b, ab, ba, aa, bb}

**Concluindo:** para obter uma linguagem basta misturar um alfabeto?

Não. A linguagem deve possuir um conjunto de regras para que ela seja construída também.

# Passo IV: Gramática

𝐺 = (𝑉 ,Σ,𝑃,𝑆)

Onde:

* 𝑉 é o vocabulário da linguagem, onde 𝑉 = 𝑁 ∪ Σ 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},Σ = {0,1},𝑃,𝑆 = 𝐴)

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 Σ = {+,−} Escreva:

* Σ∗
* Σ+
* Σ0
* Σ2

1. Seja 𝐿(𝐺) = {𝑠 ∈ {𝑎,𝑏}∗|𝑠 = 𝑎𝑛𝑏,𝑛 ≥ 0}. Escreva uma gramática que represente L(G).
2. Seja 𝐿(𝐺) = {𝑠 ∈ {𝑎,𝑏,𝑐}∗|𝑠 = 𝑎𝑚𝑏𝑐𝑛,𝑚 ≥ 0,𝑛 ≤ 1}

* ′𝑎𝑏′ ∈ 𝐿(𝐺)?
* ′𝑎𝑏𝑏𝑐′ ∈ 𝐿(𝐺)?
* Qual a menor cadeia?
* Escreva uma gramática para a linguagem.

4. (Ramos et al Pag 137) Considere o alfabeto Σ = {𝑎,𝑏}. Proponha gramáticas diferentes 𝐺1 e 𝐺2 que gerem linguagens sobre esse alfabeto, de tal forma que:

* 𝐺1 ≠ 𝐺2
* 𝐿1(𝐺1) ⊆ Σ∗
* 𝐿2(𝐺2) ⊆ Σ∗
* 𝐿1 seja infinita
* 𝐿2 seja infinita

Adicionalmente:

* 1. 𝐿1 ∩ 𝐿2 = 𝜙
  2. 𝐿1 ⊆ 𝐿2 e 𝐿1 ≠ 𝐿2
  3. 𝐿1 = Σ∗ − 𝐿2
  4. 𝐿1 = 𝐿2 = Σ∗
  5. 𝐿1 ∩ 𝐿2 = (𝑎𝑏)∗
  6. 𝐿1 − 𝐿2 = {𝑎,𝑎𝑏,𝑏}
  7. 𝐿1 ∪ 𝐿2 = Σ∗ e 𝐿1 ∩ 𝐿2 = 𝜙

**Próxima aula:**

* Diagrama Sintático
* Léxico: tokenização
* Embrião do Sintático

Referências:

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