

TEORIA DA COMPUTAÇÃO I

Aula 01 – Introdução

Prof. Dr. Guilherme Pina Cardim

guilhermecardim@fai.com.br

- A teoria da computação é uma das bases fundamentais da ciência da computação;
- Foi desenvolvida independentemente do estudo do computador como conhecemos hoje;
- Ao longo do desenvolvimento da teoria da computação surgiram três questionamentos:
 - O que é uma solução computável?
 - Quais são os limites do que pode ser computado?
 - Existem problemas sem solução computacional?

(Diverio e Menezes, 2011)

- Há duas ênfases nos estudos de ciência da computação:
 - Fundamentos e modelos computacionais (ênfase teórica);
 - Projeto de sistemas computacionais (ênfase prática);
- Para a ênfase teórica o computador é apenas uma instância tecnológica onde a teoria é aplicada;
 - A ênfase teórica não depende de instrumentos ou máquinas, ou seja, ela é independente do computador utilizado (*hardware e software*).

(Diverio e Menezes, 2011)

- “A teoria da computação estuda modelos de computação suficientemente genéricos para especificar qualquer função computável, bem como explora os limites do que pode ser computado”.

(Diverio e Menezes, 2011)

- Linguagem

linguagem

lin·gua·gem

sf

- 1 **LING** Faculdade que tem todo homem de comunicar seus pensamentos e sentimentos.
- 2 **LING** Conjunto de sinais falados, escritos ou gesticulados de que se serve o homem para exprimir esses pensamentos e sentimentos.
- 3 **LING** Faculdade inata de toda a espécie humana de expressão audível e articulada, produzida pela ação de língua e dos órgãos vocais adjacentes.
- 4 **LING** Faculdade inata de todo indivíduo de aprender e usar uma língua.
- 5 Qualquer meio utilizado pelo homem para se comunicar.
- 6 **SOCIOLING** Uso da língua, segundo a situação, o meio social, o interlocutor etc.; registro.
- 7 Sistema de comunicação animal por meio de sons, cantos e outros meios: *Alguns estudos indicam que os golfinhos e as baleias têm linguagem própria.*
- 8 Sistema de sinais desenvolvido a partir de uma língua e que funciona como meio de comunicação, em circunstâncias específicas: *A linguagem dos sinais de trânsito nem sempre é compreendida por todos os motoristas.*
- 9 Conjunto de sinais convencionados que permitem a construção e a transmissão de uma mensagem, apenas por aqueles que os conhecem; código.

- **Linguagem:**

Forma precisa de expressar problemas, possibilitando o desenvolvimento formal adequado ao estudos da computabilidade.

(Diverio e Menezes, 2011)

Para melhor definir linguagem é necessário compreender outros conceitos, como *alfabeto* e *palavra*.

Alfabeto

Conjunto finito de símbolos ou caracteres

- Símbolo: Σ
- Um conjunto infinito **não é** um alfabeto;
- O conjunto vazio **é** um alfabeto;

Exemplos:

✓ $\{a, b, c\}$

✓ \emptyset (Conjunto vazio)

Não são exemplos:

× \mathbb{N} (Conjunto dos números naturais)

× $\{a, b, aa, ab, ba, bb, aaa, \dots\}$

Palavra (Cadeia de caracteres)

- *Uma palavra sobre Σ é uma sequência finita de símbolos de Σ*

- Símbolo: **w**
- Uma cadeia sem símbolos é uma cadeia válida (ε);

Exemplos: ($\Sigma = \{a, b, c\}$)

✓ abca

✓ ε (palavra vazia)

Não são exemplos:

× abcd

× eab

Comprimento

Número de símbolos que compõe a palavra

- Símbolo: $|w|$

Exemplos: ($\Sigma = \{a, b, c\}$)

$$\checkmark |abcb| = 4$$

$$\checkmark |\varepsilon| = 0$$

Subpalavra / Prefixo / Sufixo

Uma subpalavra é qualquer sequência de símbolos contíguos da palavra

Um prefixo/sufixo é qualquer subpalavra inicial/final

Exemplos: ($\Sigma = \{a, b, c\}$ e $w = abcb$)

- ✓ Prefixos: ε , a, ab, abc, abcb
- ✓ Sufixos: ε , b, cb, bcb, abcb
- ✓ Subpalavras: qualquer prefixo ou sufixo

$$\Sigma^* / \Sigma^+$$

Σ^* é o conjunto de todas as palavras de um alfabeto Σ

Σ^+ é o conjunto de todas as palavras de um alfabeto Σ , com exceção da palavra vazia (ε)

Considerando o alfabeto $\Sigma = \{a, b\}$, então:

$$\Sigma^* = \{\varepsilon, a, b, aa, ab, ba, bb, aaa, \dots\}$$

$$\Sigma^+ = \{a, b, aa, ab, ba, bb, aaa, \dots\}$$

Linguagem Formal

Uma linguagem L sobre um alfabeto Σ é um conjunto de palavras sobre Σ , ou seja:

$$L \subseteq \Sigma^*$$

- Tanto o conjunto vazio \emptyset , como o conjunto formado pela palavra vazia $\{\varepsilon\}$ são linguagens sobre qualquer alfabeto ($\emptyset \neq \{\varepsilon\}$)
- Os conjuntos Σ^* e Σ^+ são linguagens sobre qualquer alfabeto Σ

Concatenação de Palavras

É uma operação definida sobre uma linguagem L que associa a cada par de palavras uma palavra formada pela justaposição da primeira com a segunda

Considerando as palavras $v = \text{baaaa}$ e $w = \text{bb}$:

$$vw = \text{baaaabb}$$

$$v\varepsilon = \text{baaaa} = v$$

Propriedades:

- Associativa: $v(wt) = (vw)t$
- Elemento neutro: $\varepsilon w = w = w\varepsilon$
- Sucessiva: $w^n = ww^{n-1}$; onde $w^0 = \varepsilon$

Dúvidas?

1) Identifique quais dos seguintes conjuntos são alfabetos:

- a) Conjunto dos números inteiros;
- b) Conjunto dos números primos;
- c) Conjunto das letras do alfabeto brasileiro;
- d) Conjunto dos algarismos arábicos;
- e) Conjunto dos algarismos romanos;
- f) Conjunto { a, b, c, d };
- g) Conjunto das partes de { a, b, c };
- h) Conjunto das vogais;
- i) Conjunto das letras gregas.

2) Apresente todos os possíveis prefixos e sufixos de cada uma das seguintes palavras:

- a) aaa;
- b) aba;
- c) abccba;
- d) abcabc;
- e) teoria;
- f) unifai;
- g) universidade.

3) *Em relação à palavra **abccba**, avalie as afirmativas abaixo:*

- a) Os conjuntos de sufixos e prefixos são o mesmo;
- b) O alfabeto sobre o qual é gerada essa palavra só pode ser $\Sigma = \{a, b, c\}$
- c) Existem exatamente três subpalavras que são prefixos e sufixos ao mesmo tempo.

Sequência de instruções que descrevem como um computador deve realizar cada tarefa.

(Tanenbaum, 2007)

Conjunto estruturado de instruções que capacitam uma máquina a aplicar sucessivamente determinadas operações básicas e testes sobre dados iniciais fornecidos, com o objetivo de transformar estes dados numa forma desejável.

(Diverio e Menezes, 2011)

Instruções Rotuladas

- Utilizadas para representar programas;
- Cada instrução rotulada deve ser identificada por um rótulo;
- Pode ser de dois tipos:

Operação

Indica a operação a ser executada seguida de um desvio incondicional para a próxima instrução

- r_1 : faça **F** vá_para r_2
- r_1 : faça **✓** vá_para r_2

Instruções Rotuladas

- Utilizadas para representar programas;
- Cada instrução rotulada deve ser identificada por um rótulo;
- Pode ser de dois tipos:

Teste

Determina um desvio condicional. Realiza a avaliação de um teste para determinar qual a próxima instrução.

- r_1 : se **T** vá_para r_2 senão vá_para r_3

- A estrutura de controle e testes de um programa pode assumir três modos distintos:
 - Estruturação Monolítica;

- *Estruturação Monolítica:* O programa não faz uso explícito de mecanismos como iteração, subdivisão ou recursão. A estruturação é baseada em desvios condicionais e incondicionais.

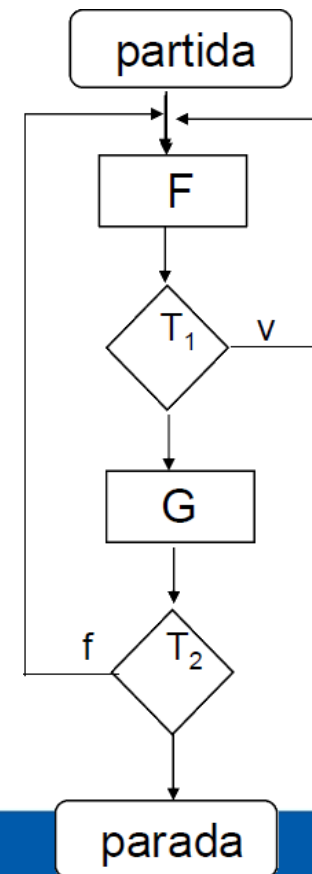
Instruções rotuladas:

1: faça F vá_para 2

2: se T_1 então vá_para 1 senão vá_para 3

3: faça G vá_para 4

4: se T_2 então 5 senão vá_para 1



Um programa monolítico é um par ordenado:

$$P = (I, r)$$

I: é o conjunto finito de instruções rotuladas. Não há duas instruções distintas com o mesmo rótulo;

r: é o rótulo inicial que identifica a instrução inicial em I;

Um rótulo é dito **rótulo final** se é referenciado por pelo menos uma instrução de I, mas não rotula qualquer instrução de I;

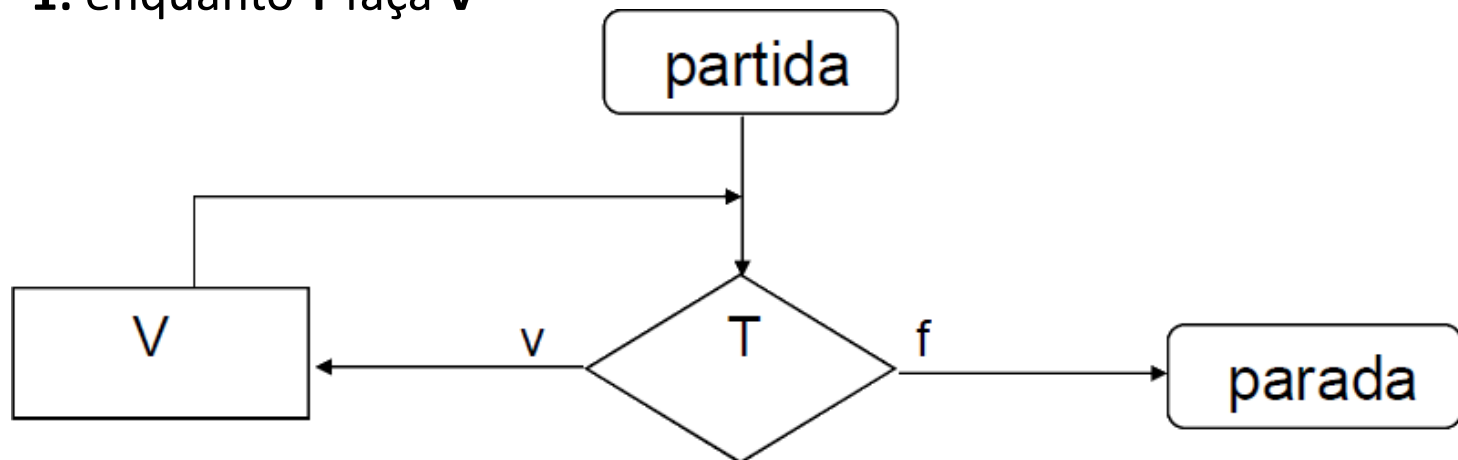
O conjunto I nunca é vazio, pois sempre existe pelo menos a instrução etiquetada pelo rótulo inicial.

- A estrutura de controle e testes de um programa pode assumir três modos distintos:
 - Estruturação Monolítica;
 - Estruturação Iterativa;

Programa Iterativo

- *Estruturação Iterativa*: O programa é escrito de forma a substituir os desvios incondicionais por estruturas de testes e/ou repetições melhorando a estruturação de desvios.

1: enquanto T faça V



Programa Iterativo

- Um programa iterativo **P** é definido como:
 1. A operação vazia \checkmark constitui um PI;
 2. Cada identificador de operação constitui um PI;
 3. Se **V** e **W** são PI, então a *composição sequencial* **V;W** é um PI com execução de **V** e após a execução de **W**;
 4. Se **V** e **W** são PI e se **T** é um teste, então a *composição condicional* (se **T** então **V** senão **W**) é um PI com execução de **V** se **T** é verdadeiro ou de **W** se **T** é falso;
 5. Se **V** é PI e **T** é um teste, então a *composição enquanto* (enquanto **T** faça **V**) é um PI que executa **V** repetidamente enquanto o teste for verdadeiro.

- A estrutura de controle e testes de um programa pode assumir três modos distintos:
 - Estruturação Monolítica;
 - Estruturação Iterativa;
 - Estruturação Recursiva.

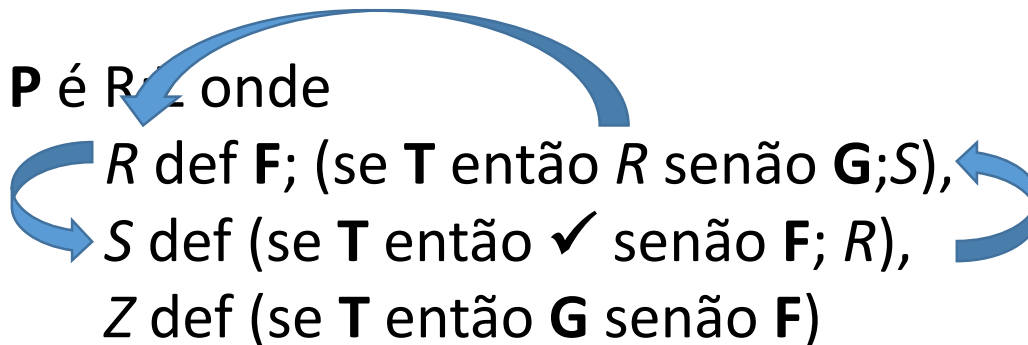
- *Estruturação Recursiva*: O programa admite a definição de sub-rotinas de forma recursiva. O uso de sub-rotinas permite a estruturação hierárquica de programas, possibilitando níveis diferenciados de abstração;

P é **R** onde

R def **F**; (se **T** então **R** senão **G**; **S**),

S def (se **T** então ✓ senão **F**; **R**),

Z def (se **T** então **G** senão **F**)



função fatorial(x)
se x==1 então 1
senão x*fatorial(x-1);

Material Referência

- DIVERIO, Tiarajú A. e MENEZES, Paulo B. **Teoria da computação: máquinas universais e computabilidade**. 3ed. Porto Alegre: Bookman, 2011.
- TANENBAUM, Andrew S. **Organização estruturada de computadores**. 5.ed. São Paulo: Pearson Prentice Hall, 2007.
- OLIVETE JR, Celso. **Teoria da computação: Aula 01**. Presidente Prudente: FCT/UNESP, 2019.