

TEORIA DA COMPUTAÇÃO

Prof. Marcus Ramos – P1 – 16/10/2025

1. (1,0 ponto) Por que os programas monolíticos são mais difíceis de serem mantidos do que os programas iterativos?

Porque a distância entre a representação estática e o comportamento dinâmico é maior nos programas monolíticos do que nos programas iterativos. A programação iterativa visa justamente diminuir essa distância.

2. (1,5 ponto) Defina, algebricamente, uma Máquina de Três Registradores (escolha as operações e testes que você quiser).

$M_T = (V, X, Y, \pi_X, \pi_Y, \Pi_O, \Pi_T)$, com 3 registradores a , b e c .

$V = \mathbb{N}^3$

$X = \mathbb{N}$

$Y = \mathbb{N}$

$\pi_X = \text{armazena_a}$

$\pi_Y = \text{retorna_c}$

$O_M = \{\text{subtrai_a}, \text{adicona_b}, \text{adiciona_c}\}$

$T_M = \{a_zero, c_zero\}$

- $\text{armazena_a}: \mathbb{N} \rightarrow \mathbb{N}^3$
 $\forall n \in \mathbb{N}, \text{armazena_a}(n) = (n, 0, 0)$
- $\text{retorna_c}: \mathbb{N}^3 \rightarrow \mathbb{N}$
 $\forall (p, q, r) \in \mathbb{N}^3, \text{retorna_c}(p, q, r) = r$
- $\text{subtrai_a}: \mathbb{N}^3 \rightarrow \mathbb{N}^3$
 $\forall (p, q, r) \in \mathbb{N}^3, \text{subtrai_a}(p, q, r) = (p - 1, q, r)$
Se $p = 0$, então (p, q, r) .
- $\text{adicona_b}: \mathbb{N}^3 \rightarrow \mathbb{N}^3$
 $\forall (p, q, r) \in \mathbb{N}^3, \text{adicona_b}(p, q, r) = (p, q + 1, r)$
- $\text{adiciona_c}: \mathbb{N}^3 \rightarrow \mathbb{N}^3$
 $\forall (p, q, r) \in \mathbb{N}^3, \text{adiciona_c}(p, q, r) = (p, q, r + 1)$
- $a_zero: \mathbb{N}^3 \rightarrow \{\text{verdadeiro}, \text{falso}\}$
 $\forall (p, q, r) \in \mathbb{N}^3$, se $p = 0$, então $a_zero(p, q, r) = \text{verdadeiro}$
 $\forall (p, q, r) \in \mathbb{N}^3$, se $p \neq 0$, então $a_zero(p, q, r) = \text{falso}$
- $c_zero: \mathbb{N}^3 \rightarrow \{\text{verdadeiro}, \text{falso}\}$
 $\forall (p, q, r) \in \mathbb{N}^3$, se $r = 0$, então $c_zero(p, q, r) = \text{verdadeiro}$
 $\forall (p, q, r) \in \mathbb{N}^3$, se $r \neq 0$, então $c_zero(p, q, r) = \text{falso}$

3. (1,0 ponto) Defina, genericamente, o termo “computação de um programa”.

“Computação de um programa” é a sucessão de estados que o programa assume durante a sua execução. É definida conforme o tipo de programa. Para programas monolíticos, é uma sequência de elementos de $R \times V$, onde R é o

conjunto de rótulos do programa e V é o conjunto de valores de memória. Para programas iterativos, é uma sequência de elementos de $I \times V$, onde I é o conjunto de programas iterativos. E para programas recursivos é uma sequência de elementos de $J \times V$, onde J é o conjunto de expressões.

4. (1,0 ponto) Qual a importância da Máquina de Traços na formulação do algoritmo de verificação da equivalência forte de programas?

Máquinas de Traços nos informam que dois programas são fortemente equivalentes se eles executam as mesmas operações na mesma ordem. O algoritmo de verificação da equivalência forte de programas verifica se dois programas executam as mesmas operações na mesma ordem. Assim, Máquinas de Traços são a base teórica do algoritmo em questão.

5. (1,0 ponto) Qual a importância do Teorema Fundamental da Aritmética na codificação de dados estruturados?

O objetivo da codificação de dados estruturados é obter uma codificação unívoca para cada elemento considerado. Ele pode ser alcançado através do uso do Teorema Fundamental da Aritmética (TFA), que garante que todo número maior do que 1 pode ser decomposto de maneira unívoca (a menos de permutações) em produtos de números primos. Em outras palavras, o TFA possibilita a codificação unívoca de dados estruturados.

6. (1,5 ponto) Obtenha um programa iterativo que implemente a operação de potenciação $A := A^B$ na Máquina Norma, onde A e B são registradores (suponha que a multiplicação é dada).

$A := A^B$

Se $B = 0$ então $A := 1$

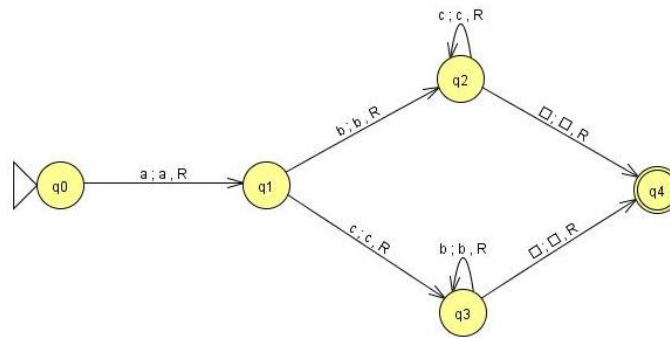
senão $B := B - 1$

até $(B = 0)$ faça

$A := A * A$

$B := B - 1$

7. (1,5 ponto) Obtenha uma Máquina de Turing que reconheça a linguagem $abc^*|acb^*$ sobre o alfabeto $\{a, b, c\}$.



8. (1,5 ponto) Descreva duas estratégias que podem ser usadas para provar que uma máquina M é universal.

Evidências internas é uma estratégia que procura mostrar que M é universal a partir das operações e testes primitivos de M , mostrando como construir programas que implementam operações e testes mais complexos. *Evidências externas* é uma estratégia alternativa, que procura estabelecer a equivalência de M com a Máquina de Turing (MT) através de dois teoremas: (i) M simula MT e (ii) MT simula M .