




Plano de Ensino



- Apresentação e Revisão
- Introdução à Teoria da Computação.
- Conceitos Básicos de Teoria da Computação.
- Programas.
- Máquinas e Computações.
- **Classes de Programas.**
- Função Computada e Equivalência.
- Modelos Computacionais.
- Máquinas Universais.
- Tese de Church.
- Máquina de Turing.



Livro-Texto



- Bibliografia Básica:
 - » LEWIS, Harry R.; PAPADIMITRIOU, Christos H. **Elementos da Teoria da Computação**. 2ª ed. Porto Alegre: Bookman, 2000.
 - » SIPSER, Michael. **Introdução à Teoria da Computação**. 2ª ed. São Paulo: Cengage Learning, 2011.

4. Classes de Programas – Hierarquia

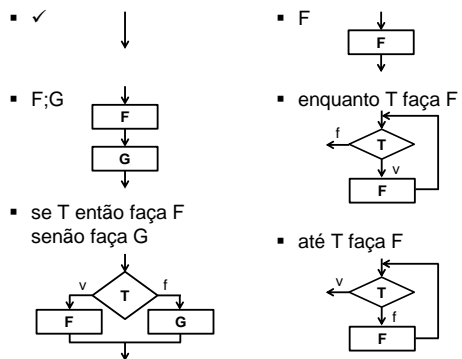


4. Classes de Programas – Conversão



- **Teorema 1:** seja P_I um programa iterativo; então existe um programa monolítico P_M tal que
$$P_M \equiv P_I$$
- Tal obtenção é feita diretamente pelo mapeamento das construções elementares do programa iterativo em sequências de construções equivalentes do programa monolítico.

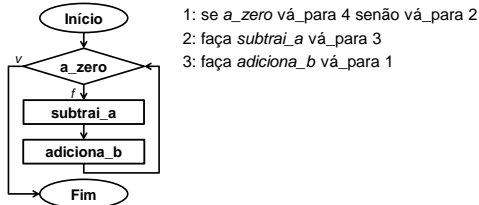
4. Classes de Programas – Conversão



4. Classes de Programas – Exemplo 1



- Considere P_I o programa iterativo a seguir:
até a_zero (
 subtrai_a; adiciona_b)
- O mapeamento para um programa monolítico P_M é feito de forma direta:



4. Classes de Programas – Exemplo 2



$P_{ITER} =$ (
 A; B; enquanto T (C); D
)

4. Classes de Programas – Conversão



- Teorema 2:** seja P_M um programa monolítico; então existe um programa recursivo P_R tal que
$$P_R \equiv P_M$$
- Suponha que $L = \{r_1, r_2, \dots, r_n\}$ é o conjunto de rótulos de P_M , r_1 sendo o rótulo inicial e r_n seja o único rótulo final. Então:
 P_R é R_1 onde $R_1 \text{ def } E_1, R_2 \text{ def } E_2, \dots, R_n \text{ def } \checkmark$
- E $\forall k, 1 \leq k \leq n$, E_k é definido como:
 - r_k : faça F vá_para r_i
 $E_k = F; R_i$
 - r_k : se T então vá_para r_i senão vá_para r_j
 $E_k = (\text{se T então } R_i \text{ senão } R_j)$

4. Classes de Programas – Exemplo 3



- Considere P_M o programa monolítico abaixo:
 - se a_zero vá_para 4 senão vá_para 2
 - faça subtrai_a vá_para 3
 - faça adiciona_b vá_para 1
- O mapeamento para um programa recursivo P_R , obtido por mapeamento direto é:

R é R_1 onde

R_1 def (se a_zero R_4 senão R_2)

R_2 def (subtrai_a; R_3)

R_3 def (adiciona_b; R_1)

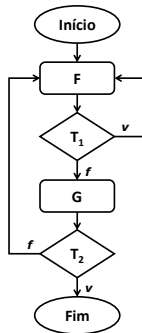
R_4 def ✓

» De forma simplificada, substituindo os termos:

R é R_1 onde

R_1 def (se a_zero ✓ senão subtrai_a; adiciona_b; R_1)

4. Classes de Programas – Exemplo 4



4. Classes de Programas – Conversão



- Teorema 3:** seja P_I um programa iterativo; então existe um programa recursivo P_R tal que

$$P_R \equiv P_I$$
- Os passos para obtenção desta tradução seguem os teoremas 1 e 2 já apresentados.



**Teoria da Computação –
Aula 04**

Ciência da Computação

clayton.valdo@anhanguera.com