UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA

TEORIA DA COMPUTAÇÃO N

INF05501
Turma B

Prof. Dr. Tiarajú Asmuz Diverio

Gabarito Capítulo 2

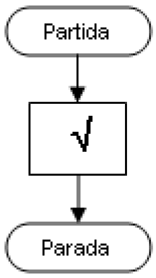
Alunos: Filipe da Silva Echevengua 00194713
Arthur Remuzzi Foscarini 00193023

Respostas:

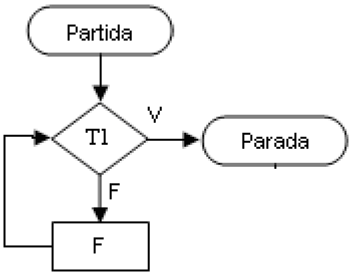
2.1) No caso da linguagem C, um programa monolítico receberia desvios condicionais e incondicionais (if, else, goto...). Um programa iterativo não aceitaria mais os desvios incondicionais e permitiria laços em sua forma estruturada (for, while, do while...) e os programas recursivos seriam organizados em partes (subrotinas) responsáveis pelo controle da recursividade e execução de operações.

2.2)

a)P2 = ({ r1: faça √ vá_para r2 }, r1)



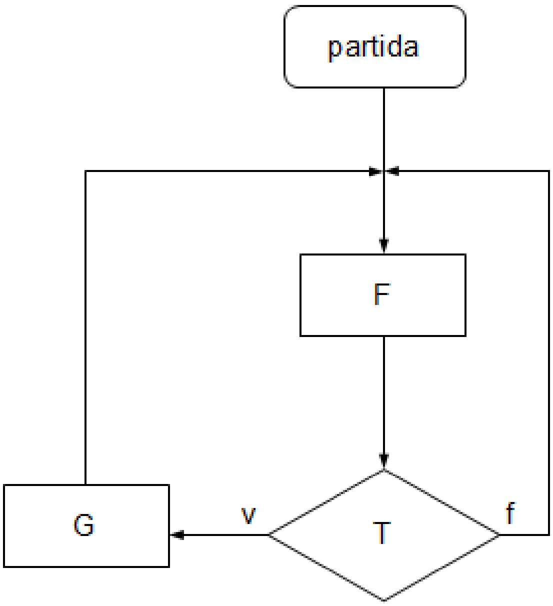
b) Composição até (programa iterativo);



c) Programa sem instrução alguma;

Não existe.

d) Programa sem instrução de parada.



2.3)

a) V não será executado somente quando a resposta do teste T for falsa na primeira vez em que T for executado.

b) Por definição de Programa iterativo, temos que cada identificador de operação e a operação √ constitui um programa iterativo. Devido a operação √ ser a base da indução para a definição de programa iterativo, e quando se chega na base significa que acabou.

c) Como um programa monolítico é representado claramente em fluxogramas, e também é possível transformar todas instruções de um programa iterativo para fluxograma, por isso conseguimos fazer uma tradução de um programa iterativo para monolítico sem muitas dificuldades.

2.4) Para as três estruturas de programas (monolítico, iterativo e recursivo), dado um valor inicial, em uma mesma máquina, a computação será sempre a mesma (a seqüência de instruções executadas e os valores intermediários de memória, em diversas execuções do mesmo programa, para o mesmo valor de entrada, será sempre a mesma). O resultado da execução do mesmo programa, na mesma máquina, com mesmo valor inicial, será sempre o mesmo.

2.5) Computação é a sequência de instruções executadas para um programa e os diferentes valores da memória ao longo de sua execução, basicamente é o histórico do funcionamento da máquina para o programa. Função computada é o resultado obtido ao fim de uma computação, desde que a computação seja finita.

2.6)

Sejam $M = (V, X, Y, \pi_X, \pi_Y, \Pi_F, \Pi_T)$ uma máquina e P um programa iterativo para M.

A **computação do programa iterativo P na Máquina M** é uma cadeia de pares da forma: $(X_0, v_0)\ (X_1, v_1)\ (X_2, v_2)$... onde

- (X_0, v_0) é tal que $\mathbf{X_0;}$□ é todo o programa iterativo inicial P concatenado ao programa vazio e v_0 é o valor inicial armazenado na memória;
- para cada par $(\mathbf{X_j, v_j})$ da cadeia, onde $j \in \{0, 1, 2, ...\}$, tem-se que X_j são programas Iterativos (suponha que F é um identificador de operação, T é um identificador de teste e $\mathbf{W}$, $\mathbf{W_1}$, $\mathbf{W_2}$ são programas iterativos):

a) Se X_j é da forma: $\mathbf{X_j}$ = □; $\mathbf{W}$

então, tem-se que:

$X_{j+1} = \mathbf{W}$ e

$\mathbf{v_{j+1} = v_j}$

b) Se X_j é da forma: $\mathbf{X_j = F; W}$

então, tem-se que:

$X_{j+1} = \mathbf{W}$ e

$\mathbf{v_{j+1}} = \pi_F\ (\mathbf{v_j})$

c) Se X_j é da forma $\mathbf{X_j = (W_1; W_2); W}$

então, tem-se que:

$X_{j+1} = \mathbf{W_1;(W_2; W)}$ e

$\mathbf{v_{j+1} = v_j}$

d) Se X_j é da forma $\mathbf{X_j = (}$se $\mathbf{T}$ então $\mathbf{W_1}$ senão $\mathbf{W_2); W}$

então, tem-se que:

$X_{j+1} = \mathbf{W_1; W}$ se $\pi_T(v_j)$ = verdadeiro

$\mathbf{W_2; W}$ se $\pi_T(v_j)$ = falso

e $\mathbf{v_{j+1} = v_j}$

e) Se X_j é da forma $\mathbf{X_j}$ = enquanto $\mathbf{T}$ faça $\mathbf{(W_1); W}$

então, tem-se que:

$X_{j+1} = \mathbf{W_1}$; enquanto $\mathbf{T}$ faça $\mathbf{(W_1);W}$ se $\pi_T(v_j)$= verdadeiro

$\mathbf{W}$ se $\pi_T(v_j)$ = falso

e $\mathbf{v_{j+1} = v_j}$

f) Se X_j é da forma $\mathbf{X_j}$ = até $\mathbf{T}$ faça $\mathbf{(W_1); W}$

então, tem-se que:

$X_{j+1} = \mathbf{W_1}$; até $\mathbf{T}$ faça $\mathbf{(W_1);W}$ se $\pi_T(v_j)$= *falso*

$\mathbf{W}$ se $\pi_T(v_j)$ = *verdadeiro*

e $\mathbf{v_{j+1} = v_j}$

2.7)

Sejam $M = (V, X, Y, \pi_X, \pi_Y, \Pi_F, \Pi_T)$ uma máquina e P um programa iterativo para M.

A ***função computada pelo programa iterativo* P *na máquina* M** denotada por:

$\langle P, M\rangle: X \rightarrow Y$

é uma função parcial definida para $x \in X$ se a cadeia:

$$\mathbf{(X_0, v_0)(X_1, v_1)...(X_n, v_n)}$$

é uma computação finita de P em M, onde $X_n = \square$, e o valor inicial da memória é dado pela função de entrada, ou seja, $v_0 = \pi_X(x)$. Nesse caso, a imagem de x é dada pela função de saída aplicada ao último valor da memória na computação, ou seja:

$$\mathbf{\langle P, M\rangle(x) = \pi_Y(v_n)}$$

2.8) A definição de computação para programas iterativos está feita no exercício 1.2.6. Precisamos apenas considerar que a cadeia de pares é finita:

$$(X_0, v_0)\ (X_1, v_1)...(X_s, v_s)$$

2.9)

Na máquina um_reg, o programa a seguir tem computação infinita:

ad;(até zero faça ad)

2.10)

a) Sim, o programa retorna um valor para qualquer entrada do conjunto de entrada.
b) Nenhum, pois para todos eles a computação é infinita.
c) Pois independentemente da entrada e independentemente da máquina, tudo o que o programa faz é chamar recursivamente sua única sub-rotina infinitamente.

2.11)

a) Fluxograma 1 representado na **Figura 2.24**;

P é R1 onde

R1 def (se T1 então R2 senão R3)

R2 def (F; R3)

R3 def (se T2 então R4 senão R7)

R4 def (G; R5)

R5 def (se T1 então R7 senão R6)

R6 def (F; R1)

R7 def (√)

Simplificando:

P é R1 onde

R1 def (se T1 então F; R2 senão R2)

R2 def (se T2 então G; R3 senão √)

R3 def (se T1 então √ senão F; R1)

b) Fluxograma 2 representado na **Figura 2.25**;

P é R1 onde

R1 def (se T então R2 senão R3)

R2 def (G; R6)

R3 def (F;R4)

R4 def (se T então R6 senão R5)

R5 def (F; R1)

R6 def (√)

Simplificando:

P é R1 onde

R1 def (se T então G; √ senão F;R2)

R2 def (se T então √ senão F; R1)

c) Fluxograma 3 representado na **Figura 2.26;**

P é R1 onde

R1 def (F;R2)

R2 def (se T1 então R1 senão R3)

R3 def (G;R4)

R4 def (se T2 então R5 senão R1)

R5 def (√)

Simplificando:

P é R1 onde

R1 def F;(se T1 então R1 senão G;R2)

R2 def (se T2 então √ senão R1)

d) Fluxograma 4 representado na **Figura 2.27;**

P é R1 onde

R1 def (se T então R2 senão R3)

R2 def (F; R1)

R3 def (√)

Simplificando:

R1 def (se T então F; R1 senão √)

e) Fluxograma 5 representado na **Figura 2.28.**

P é R1 onde

R1 def (se T então R2 senão R4)

R2 def (F; R3)

R3 def (se T então R1 senão R4)

R4 def (√)

Simplificando:

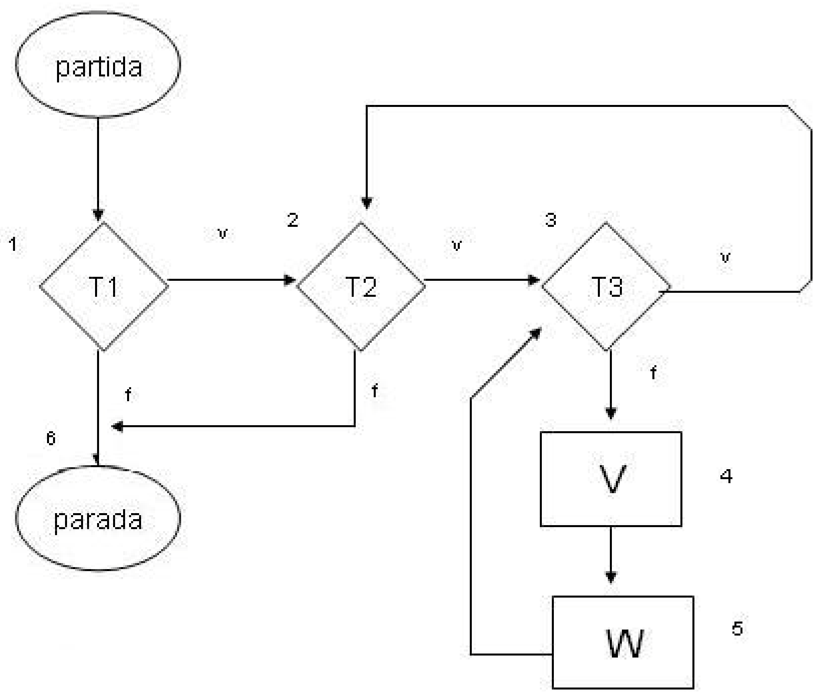
P é R1 onde

R1 def (se T então F; R2 senão √)

R2 def (se T então R1 senão √)

2.12)

a)



b) 1: Se T1 então vá-para 2 senão vá-para 6
2: Se T2 então vá-para 3 senão vá-para 6
3: Se T3 então vá-para 2 senão vá-para 4
4: Faça V vá-para 5
5: Faça W vá-para 3

2.13) Duas soluções:

Solução 1:
até T faça √
enquanto T faça (F; G; (se T então (F; até T faça √) senão faça √))

Solução 2:
(até T faça √);
F;
G;

(enquanto T
faça
(F;
(até T faça √);
F;G;))

2.14) Se a máquina M_2 simula a máquina M_1, sabe-se que qualquer programa P' que tenha uma computação finita em M_2 também o terá em M_1. Logo, suas funções computadas em ambas as máquinas serão iguais, se P' for construído a partir de P em M_2. Pode ocorrer, entretanto, que um programa Q tenha uma computação finita em M_2, mas não em M_1. Neste caso, a computação de Q em M_1 pode não ser definida ou estar em loop. O poder computacional de M_1 não necessariamente é menor do que M_2. Os novos testes e operações de M_1 simplesmente facilitam a programação, mas não possibilitam a resolução de um novo problema.

2.15) Sejam $\mathbf{M=(V, X, Y, \pi_X, \pi_Y, \Pi_F, \Pi_T)}$ uma máquina e **P** um programa monolítico para **M**. A função computada por **P** em **M** é uma função parcial definida para $x \in \mathbf{X}$ se a cadeia $(s_0, v_0)(s_1, v_1)...(s_n, v_n)$ é uma computação finita de **P** em **M** e todo v_k definido por $v_k = \pi_F(v_{k-1})$, sendo $\pi_F \in \Pi_F$, tem π_F definida em v_{k+1} – sendo $v_0 = \pi_X(x)$. Nesse caso, a imagem de x é dada por $\pi_Y(v_n)$.

2.16)

W1:
P é R1 onde
R1 def (Se T então F; R2 senão √)
R2 def (Se T então R1 senão G; R1)

W2:

P é R1 onde
R1 def (Se T então F; R2 senão √)
R2 def (Se T então F; R2 senão G; R1)

Gabarito das questões 2.17 até 2.30

Exercício 2.17	D
Exercício 2.18	E
Exercício 2.19	D
Exercício 2.20	A
Exercício 2.21	D
Exercício 2.22	B
Exercício 2.23	B
Exercício 2.24	D
Exercício 2.25	D
Exercício 2.26	B
Exercício 2.27	C
Exercício 2.28	D
Exercício 2.29	B
Exercício 2.30	B