UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
DEPARTAMENTO DE INFORMÁTICA TEÓRICA

JOAO LUIZ GRAVE GROSS
JONAS RIBEIRO FLORES
RODRIGO LEITE

TURMA B – G16

Exercícios do Capítulo III

Trabalho da Disciplina de Teoria da Computação N

Prof. Dr. Tiarajú Asmuz Diverio

Porto Alegre, 11 de abril de 2011.

SUMÁRIO

1 EXERCÍCIOS DO CAPÍTULO III

1.3.1. Relativamente aos seguintes corolários e teoremas:

a) Corolário 3.4 Equivalência forte de Programas ⇔ Equivalência de programas em máquinas de Traços;

a.1) Justifique a afirmação de que a prova (→) é imediata;

É imediata porque P = Q se, e somente se, para qualquer máquina M as correspondentes funções parciais computáveis são iguais. Logo, como a afirmação é válida para qualquer máquina, então será válida para a máquina de traços em específico, ou seja, P e Q são fortemente equivalentes para máquina de traços.

a.2) Esboce a prova (←) para programas iterativo e recursivos;

Se dois prograas P e Q são equivalentes na máquina de traços isso significa que ambos apresentam a mesma sequência de execução de instruções, ou seja, independente da estrutura dos programas eles além de fazerrem a mesmo coisa, a realizam na mesma ordem. Isso mostra que P e Q são fortemente equivalentes, visto que a própria definição de equivalência forte de programas é englobada pela definição de equivalência d dois programas em uma máquina de traços.

b) Justifique a afirmação de que a prova do Lema 3.8 Equivalência forte: fluxogramas → rotuladas compostas é imediata;

Pois a composição das instruções rotuladas compostas não precisa de nenhuma premissa a não ser o fluxograma, como é mostrado no algoritmo da definição 3.7 (dos slides de aula).

c) Por quê o Lema 3.8 Equivalncia forte: fluxogramas → rotuladas compostas garante que $P_q \equiv P_r$ se, e somente se, $Q \equiv R$?

Pois se Q for diferente de R, ao montarmos as instruções rotuladas compostas de Q e de R a união disjunta de Iq e Ip, ou seja I, comum aos programas Pq e Pr, não será válida para Pq e nem para Pr, visto que a premissa base de que Q e R são equivalente já é falsa.

d) Justifique Corolário 3.4 Equivalência forte de Programas ⇔ Equivalência de programas em máquinas de Traços;

Respondido no item a).

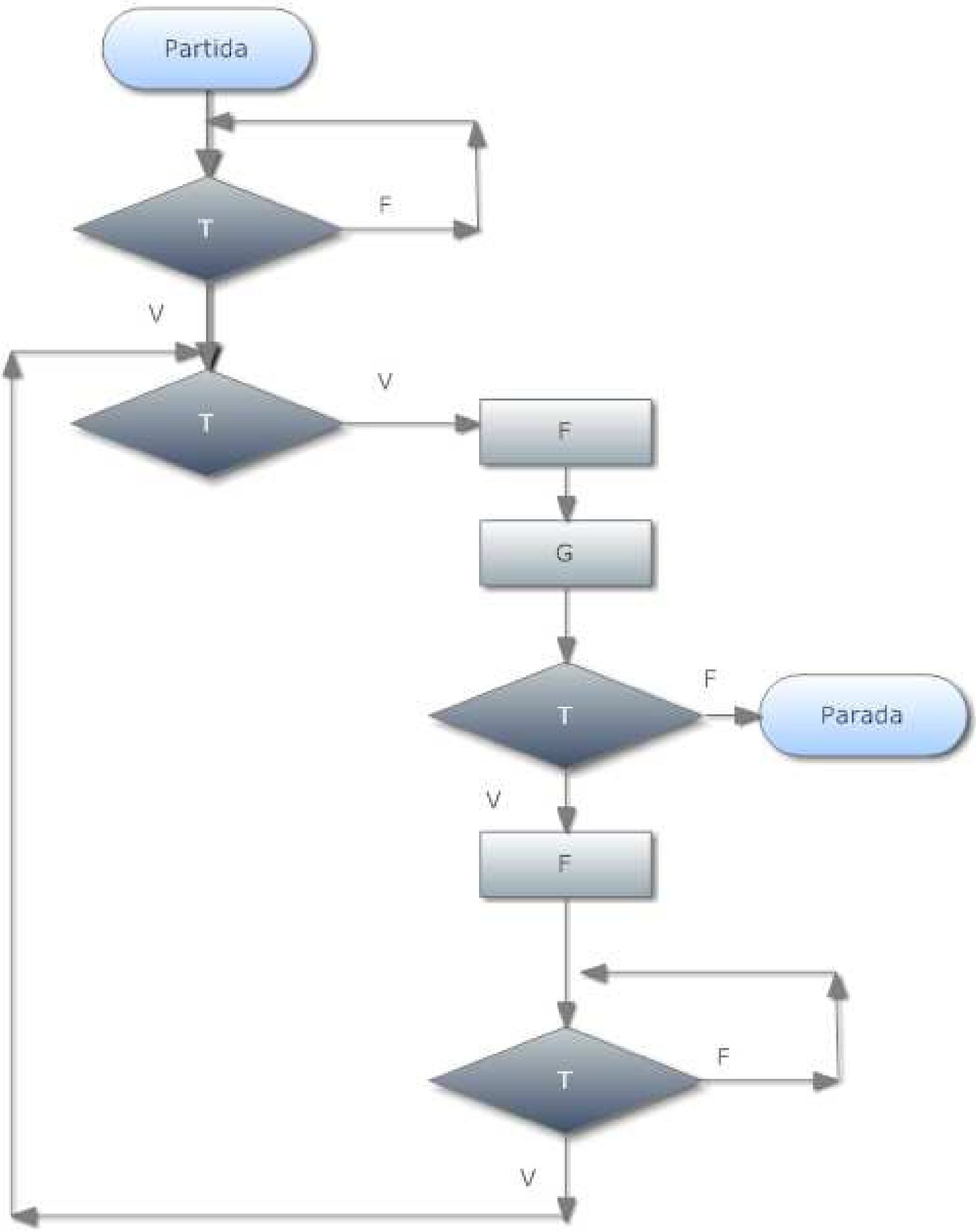
1.3.2. Mostre que os seguintes programas P e Q representados na Figura 3.7 são fortemente equivalentes.

Programa Iterativo P

```
até       T
faça      (√);
enquanto T
faça      (F;G;(se T
                 então    F;
                          até      T
                          faça (√)
                 senão    √))
```

	Programa Monolítico Q
1:	se T então vá_para 2 senão vá_para l
2:	faça F vá_para 3
3:	faça G vá_para 4
4:	se T então vá_para 5 senão vá_para 6
5:	faça F vá_para 1

Figura 3.7 Programas iterativo P e monolítico Q



Fluxograma - Programa Iterativo P – Questão 1.3.2

Instruções rotuladas compostas:

1: (F,2)(ciclo,w)

2: (G,3)(G,3)

3: (F,4)(parada,e)

4: (F,2)(ciclo,w)

w: (ciclo,w) (ciclo,w)

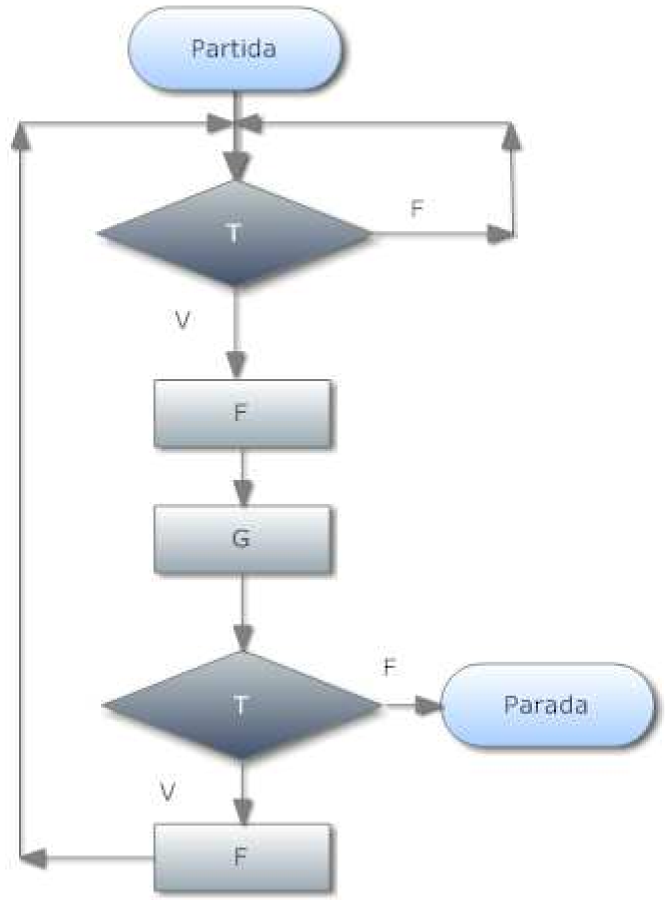
Identificação de ciclos infinitos:

$A_0 = \{\varepsilon\}$

$A_1 = \{3, \varepsilon\}$

$A_2 = \{2, 3, \varepsilon\}$

$A_3 = \{4, 1, 2, 3, \varepsilon\}$: nenhum ciclo infinito encontrado além dos já existentes



Fluxograma - Programa Monolítico Q – Questão 1.3.2

Instruções rotuladas compostas:

5: (F,6)(ciclo,w)

6: (G,7)(G,7)

7: (F,8)(parada,e)

8: (F,6)(ciclo,w)

w: (ciclo,w) (ciclo,w)

Identificação de ciclos infinitos:

$A_0 = \{\varepsilon\}$

$A_1 = \{7, \varepsilon\}$

$A_2 = \{6, 7, \varepsilon\}$

$A_3 = \{8, 5, 6, 7, \varepsilon\}$: nenhum ciclo infinito encontrado além dos já existentes

União disjunta Ip $\cup$ Im:

1: (F,2)(ciclo,w)

2: (G,3)(G,3)

3: (F,4)(parada,e)

4: (F,2)(ciclo,w)

5: (F,6)(ciclo,w)

6: (G,7)(G,7)

7: (F,8)(parada,e)

8: (F,6)(ciclo,w)

w: (ciclo,w) (ciclo,w)

Determinação de rótulos fortemente equivalentes:

$B_0 = \{(1,5)\}$

$B_1 = \{(2,6), (w,w)\}$

$B_2 = \{(3,7)\}$

$B_3 = \{(4,8), (e,e)\}$

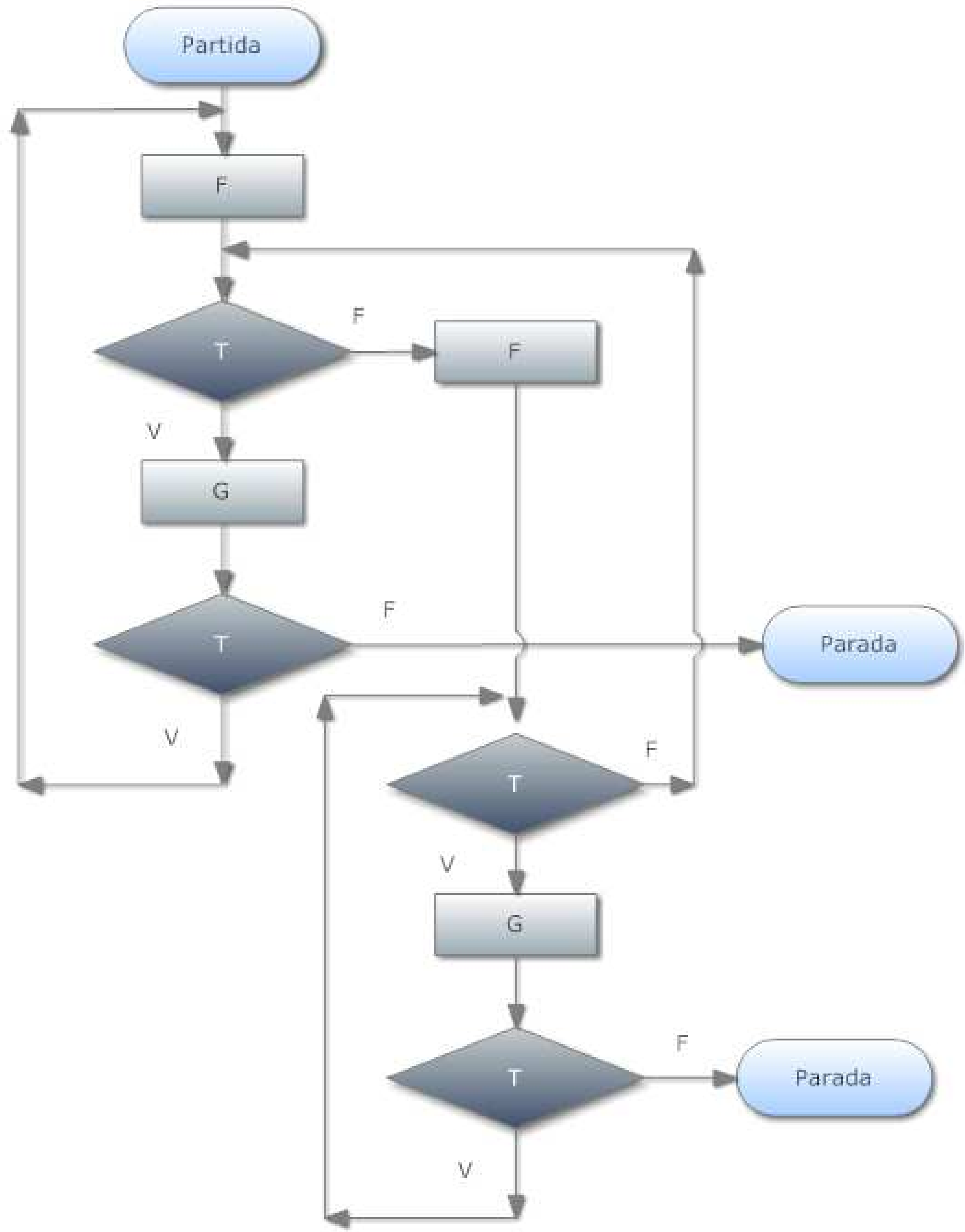
$B_4 = \{(2,6), (w,w)\} = B_1$: **P e Q são fortemente equivalente**

1.3.3. Verifique se os programas monolíticos M_1 e M_2 representados na Figura 3.8 são fortemente equivalentes.

Programa Monolítico M_1	
1:	faça F vá_para 2
2:	se T então vá_para 3 senão vá_para 5
3:	faça G vá_para 4
4:	se T então vá_para 1 senão vá_para 0
5:	faça F vá_para 6
6:	se T então vá_para 7 senão vá_para 2
7:	faça G vá_para 8
8:	se T então vá_para 6 senão vá_para 0

Programa Monolítico M_2	
1:	faça F vá_para 2
2:	se T então vá_para 3 senão vá_para 1
3:	faça G vá_para 4
4:	se T então vá_para 1 senão vá_para 0

Figura 3.8 Programas monolíticos M1 e M2



Fluxograma - Programa Monolítico M1 – Questão 1.3.3

Instruções rotuladas compostas:

1: (F,2)(F,2)

2: (G,3)(F,4)

3: (F,2)(parada,e)

4: (G,5)(F,4)

5: (G,5)(parada,e)

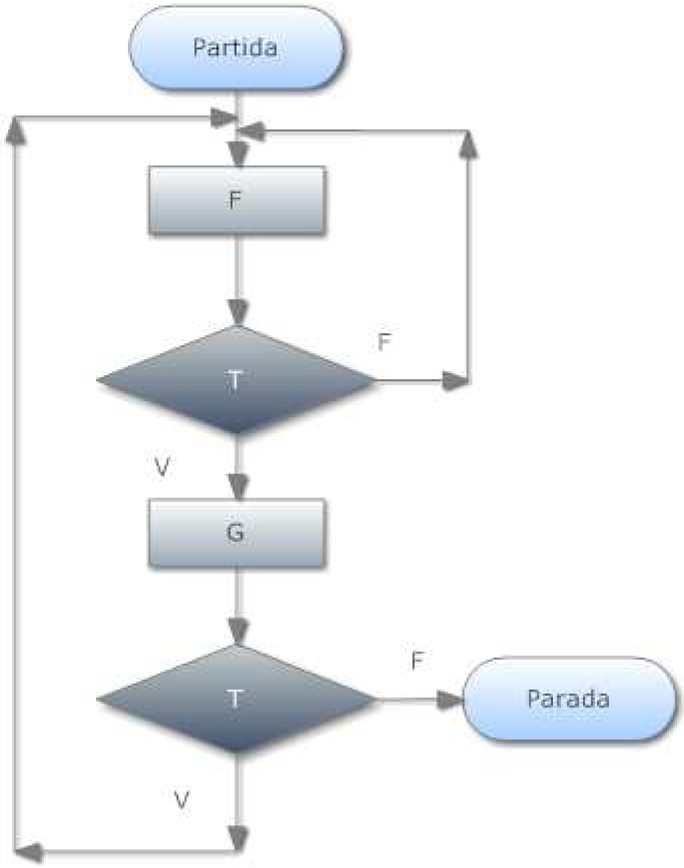
Identificação de ciclos infinitos:

$A_0 = \{\varepsilon\}$

$A_1 = \{5, 3, \varepsilon\}$

$A_2 = \{4, 2, 5, 3, \varepsilon\}$

$A_3 = \{1, 4, 2, 5, 3, \varepsilon\}$: nenhum ciclo infinito encontrado além dos já existentes



Fluxograma - Programa Monolítico M2 – Questão 1.3.3

Instruções rotuladas compostas:

6: (F,7)(F,7)

7: (G,8)(F,7)

8: (F,7)(parada,e)

Identificação de ciclos infinitos:

$A_0 = \{\varepsilon\}$

$A_1 = \{8, \varepsilon\}$

$A_2 = \{7, 8, \varepsilon\}$

$A_3 = \{6, 7, 8, \varepsilon\}$: nenhum ciclo infinito encontrado além dos já existentes

União disjunta $I_{M1} \cup I_{M2}$:

1: (F,2)(F,2)

2: (G,3)(F,4)

3: (F,2)(parada,e)

4: (G,5)(F,4)

5: (G,5)(parada,e)

6: (F,7)(F,7)

7: (G,8)(F,7)

8: (F,7)(parada,e)

Determinação de rótulos fortemente equivalentes:

$B_0 = \{(1,6)\}$

$B_1 = \{(2,7)\}$

$B_2 = \{(3,8), (4,7)\}$

$B_3 = \{(2,7), (e,e), (5,8), (4,7)\}$: falta analisar (5,8)

$B_4 = \{(5,7), (e,e)\}$: considerando T verdadeiro, a instrução 5 vai para a instrução 5, executando G, enquanto a instrução 8 vai para a instrução 7, executando F, logo os programas **M1 e M2 não são fortemente equivalentes.**

1.3.4. Qual a importância da relação de Equivalência Forte de Programas?

A importância da Relação Equivalência Forte de programas se deve a várias razões:

- Permite identificar diferentes programas em uma mesma classe equivalência, ou seja, identificar diferentes programas cujas funções computadas coincidem, para qualquer máquina;
- As funções computadas por programas equivalentes fortemente têm a mesma propriedade de que os mesmos testes e as mesmas operações são efetuados na mesma ordem, independentemente do significado dos mesmos;
- Fornece subsídios para analisar a complexidade estrutural de programas.

1.3.5. Verifique se os programas iterativos W_1 e W_2 definidos na Figura 3.9 e Figura 3.10, respectivamente, são fortemente equivalentes.

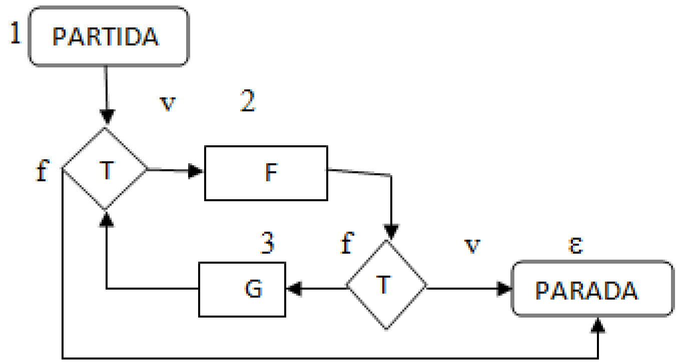
Programa Iterativo W_1
enquanto T faça (F;(se T então faça √ senão faça G))

Figura 3.9 Programa iterativo W1

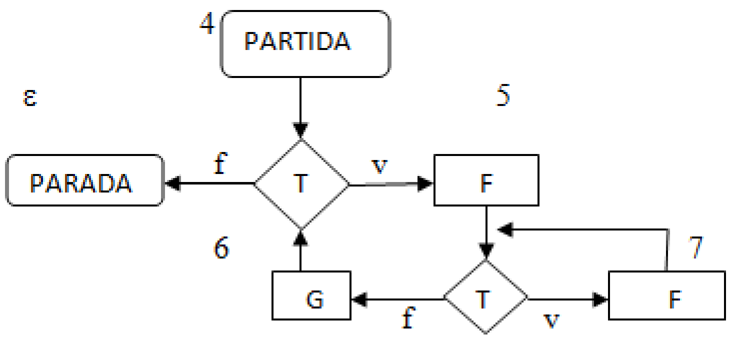
Programa Iterativo W_2
enquanto T faça (F; enquanto T faça (F);G)

Figura 3.10 Programa iterativo W2

Fluxograma W1:



Fluxograma W2:



Instruções Rotuladas do programa W1:

1: (F,2) , (parada, ε)

2: (parada, ε) , (G,3)

3: (F,2) , (parada, ε)

Instruções Rotuladas do programa W2:

4: (F,5) , (parada, ε)

5: (F,7) , (G,6)

6: (F,5) , (parada, ε)

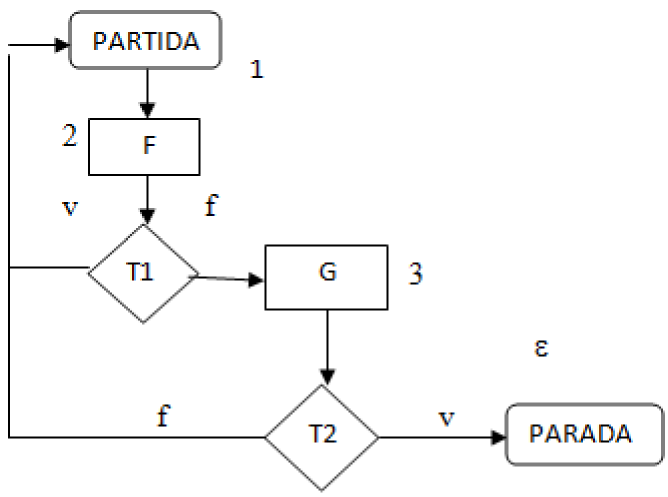
7: (F,7) , (G,6)

Conclusão: Os programas W1 e W2 não são fortemente equivalentes, pois as instruções compostas não tem o mesmo formato.

1.3.6. Traduza o programa monolítico da Figura 3.11 na forma de instruções rotuladas compostas. Como existem dois testes, cada instrução rotulada composta terá quatro possíveis sucessores, um para cada possível combinação de valores-verdade dos testes T_1 e T_2.

Programa Monolítico dois_testes	
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 vá_para 0 senão vá_para 1

Figura 3.11 Programa Monolítico dois_testes



Instruções rotuladas Compostas:

1: (F,2) , (F,2) , (F,2) , (F,2)

2: (F,2) , (F,2) , (G,3) , (G,3)

3: (parada, ε) , (F,2) , (parada, ε) , (F,2)

1.3.7. Adapte para o caso do programa monolítico da Figura 3.11, os seguintes itens:

a) Lema 3.11 Identificação de ciclos infinitos em programas monolíticos;
O lema continua o mesmo, partimos da parada e analisamos seus antecessores aqueles que nao forem são ciclos infinitos.

b) Teorema 3.15 Determinação de rótulos fortemente equivalentes;

Se as instruções compostas e simplificadas do programa forem consistentes.

c) Definição 3.16 Algoritmo de equivaencia forte de programas monolíticos.
Os correspondentes conjuntos de instruções rotuladas compostas são iguais, a menos dos rótulos.

1.3.8. Generalize a definição de instruções rotuladas compostas para o caso de três testes distintos.

Uma Instrução Rotulada Composta é uma seqüência de símbolos da seguinte forma (suponha que F e G são identificadores de operação e que T1, T2 e T3 são identificadores de teste):

- R1:se T1 então faça F vá_para R2 senão faça G vá_para R3
- R2: se T2 entao faça F vá_para R4 senão vá_para R5
- R3 : se T3 entao faça F vá_para R5 senão vá_para R6

1.3.9. Traduza os fluxogramas da Figura 3.12 e da Figura 3.13 em instruções rotuladas compostas.

1.3.10.

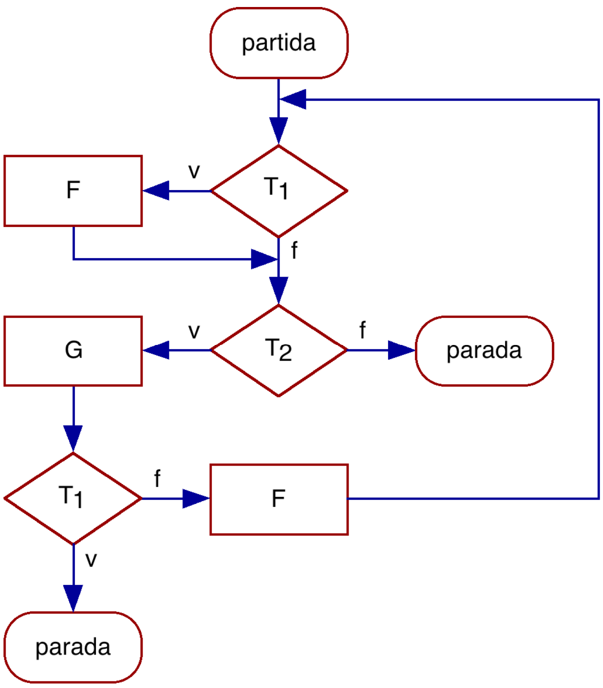


Figura 3.12 Fluxograma

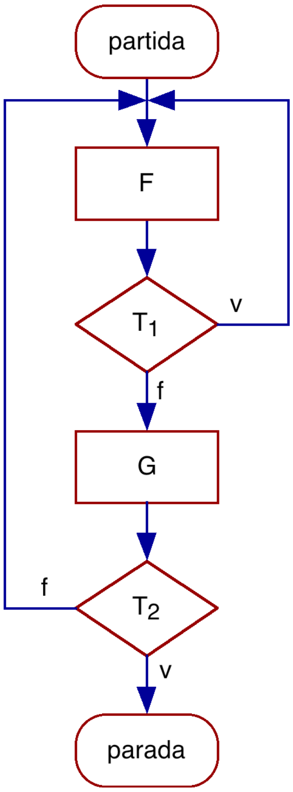


Figura 3.13 Fluxograma

Instruções Rotuladas Compostas 3.12:

1: (F,2) , (F,2) , (G,3) , (parada ε)

2: (G,3), (G,3) , (parada ε) , (parada ε)

3: (parada ε) , (parada ε) , (F,4) , (F,4)

4: (F,2) , (F,2) , (G,3) , (parada ε)

Instruções Rotuladas Compostas 3.13:

1: (F,2) , (F,2) , (F,2) , (F,2)

2: (F,2) , (F,2) , (G,3) , (G,3)

3: (parada ε), (parada ε) , (F,2) , (f,2)

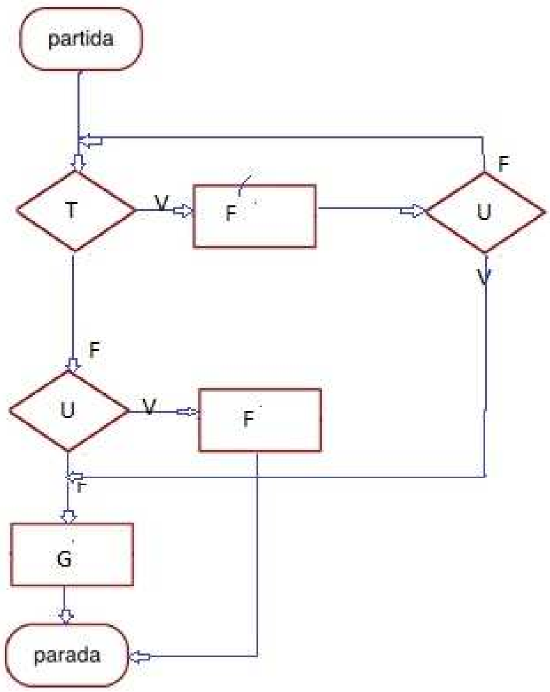
1.3.11. Reescreva o programa monolítico da Figura 3.14 na forma de Instruções Rotuladas Compostas. Apresente o Fluxograma com os nós identificados.

Oservação: Atenção ao número de testes!

Programa Monolítico	
1:	se T então vá-para 2 senão vá-para 3
2:	faça F vá-para 6
3:	se U então vá-para 5 senão vá-para 4
4:	faça G vá-para 0
5:	faça F vá-para 0
6:	se U então vá-para 4 senão vá-para 1

Figura 3.14 Programa Monolítico

Fluxograma:



Instruções Rotuladas Compostas:

1: (F,2), (F,2), (F,3), (G,4)

2: (G,4), (F,2), (G,4), (G,4)

3: (parada ε), (parada ε),(parada ε), (parada ε)

4: (parada ε), (parada ε),(parada ε), (parada ε)

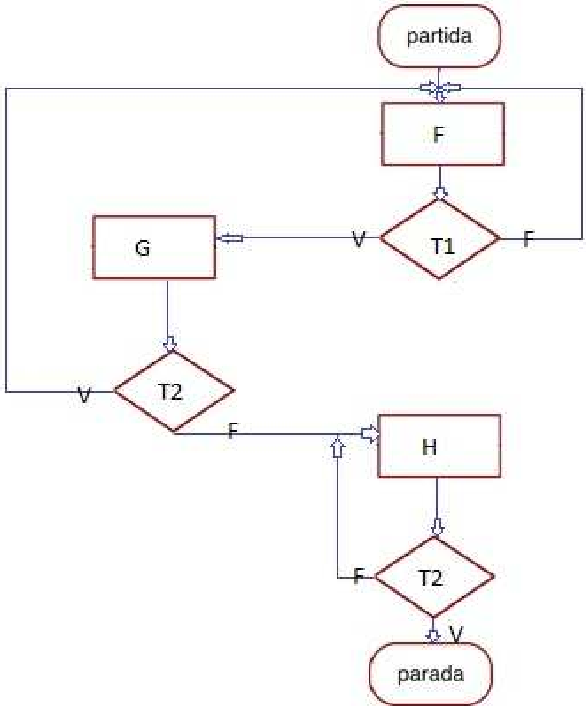
1.3.12. Reescreva o programa monolítico da Figura 3.15 na forma de Instruções Rotuladas Compostas. Apresente o Fluxograma com os nós identificados.

Oservação: Atenção ao número de testes!

Programa Monolítico	
1:	faça F vá-para 2
2:	se T_1 então vá-para 3 senão vá-para 1
3:	faça G vá-para 4
4:	se T_2 então vá-para 1 senão vá-para 5
5:	faça H vá-para 6
6:	se T_2 então vá-para 0 senão vá-para 5

Figura 3.15 Programa Monolítico

Fluxograma:



Instruções Rotuladas Compostas:

1: (F,2), (F,2), (F,2), (F,2)

2: (G,3), (G,3), (F,2), (F,2)

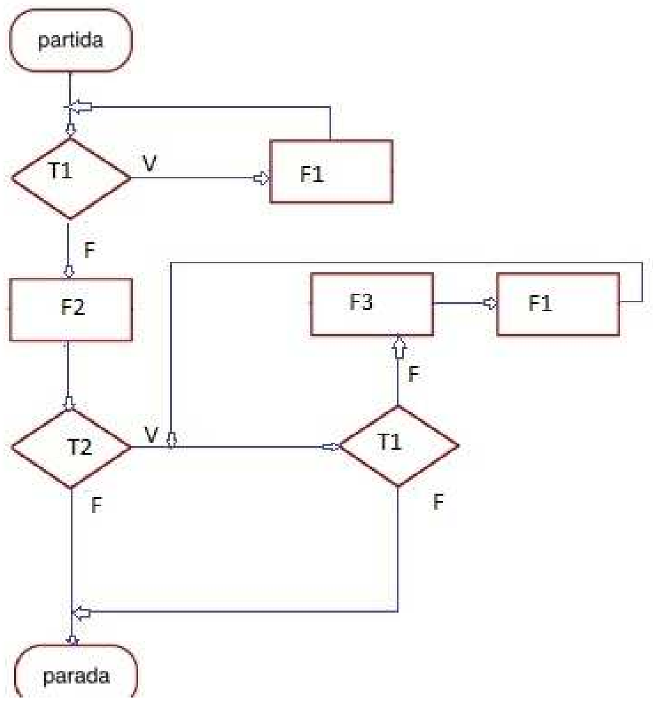
3: (F,2), (H,4), (F,2) (H,4)

4: (parada ε), (H,4),(parada ε), (H,4)

1.3.13. Reescreva o programa iterativo abaixo na forma de Instruções Rotuladas Compostas. Apresente o Fluxograma com os nós identificados. Atenção ao número de testes!

Oservação: Atenção ao número de testes!

enquanto T_1 faça (F_1);
F_2;
(se T_2 então faça até T_1 faça (F_3; F_1);√ senão faça √)



Instruções Rotuladas Compostas:

1: (F1,2), (F1,2), (F2,3), (F2,3)

2: (F1,2), (F1,2), (F2,3), (F2,3)

3: (parada ε), (parada ε),(F3,4), (parada ε)

4: (F1,5), (F1,5), (F1,5), (F1,5)

5:(parada ε), (parada ε),(F3,4), (F3,4)

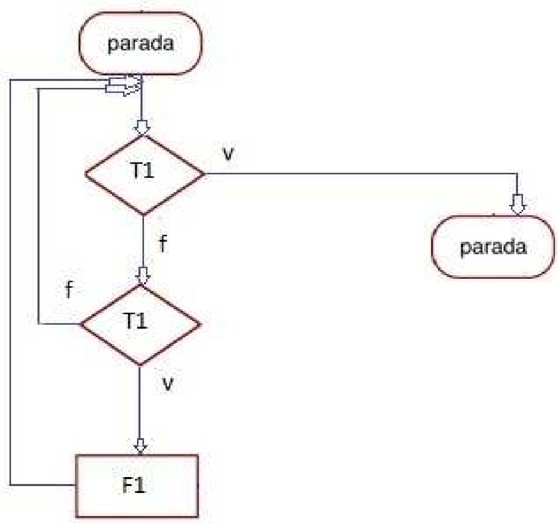
1.3.14. Reescreva o programa monolítico da Figura 3.16 na forma de Instruções Rotuladas Compostas. Apresente o Fluxograma com os nós identificados.

Oservação: Atenção ao número de testes!

Programa Monolítico	
1:	se T_1 então vá-para 4 senão vá-para 3
2:	faça F_1 vá-para 1
3:	se T_1 então vá-para 2 senão vá-para 1

Figura 3.16 Programa Monolítico

Fluxograma:



Instruções rotuladas compostas:

1: (parada ε), (ciclo w)

2: (parada ε), (ciclo w)

w:(ciclo w), (ciclo w)

REFERÊNCIAS

[TAD] Diverio, Tiarajú Asmuz. Teoria da computação: máquinas universais e computabilidade / Tiarajú Asmuz Diverio, Paulo Blauth Menezes. – Porto Alegre: Instituto de Informática da UFRGS : Bookman, c2011. 3ª edição.