Teoria da Computação

Programas Monolíticos com Instruções Rotuladas

Cristiano Lehrer, M.Sc.

Copyright © 2009/2025 • ybadoo.com.br

ybadoo

Instruções Rotuladas

- Além da representação diagramática, um fluxograma pode ser denotado na forma de texto, usando instruções rotuladas.
- Como o próprio nome indica, cada instrução rotulada é identificada por um rótulo.
- Uma instrução rotulada pode ser como segue:
  - Operação:
    - Indica a operação a ser executada seguida de um desvio incondicional para a instrução subsequente.
  - Teste:
    - Determina um desvio condicional, ou seja, que depende da avaliação de um teste.

Copyright © 2009/2025 • ybadoo.com.br

ybadoo

Rótulo

- Um rótulo ou etiqueta é uma cadeia de caracteres finita (palavra) constituída de letras ou dígitos.
- Uma Instrução Rotulada é uma cadeia de caracteres finita (palavra) de uma das duas formas a seguir (suponha que F e T são identificadores de operação e teste, respectivamente e que R1, R2 e R3 são rótulos):
  - Operação:
    - `R1: faça F vá_para R2;`
  - Teste:
    - `R2: se T1 então vá_para R1 senão vá_para R3;`

Copyright © 2009/2025 • ybadoo.com.br

ybadoo

Programa Monolítico

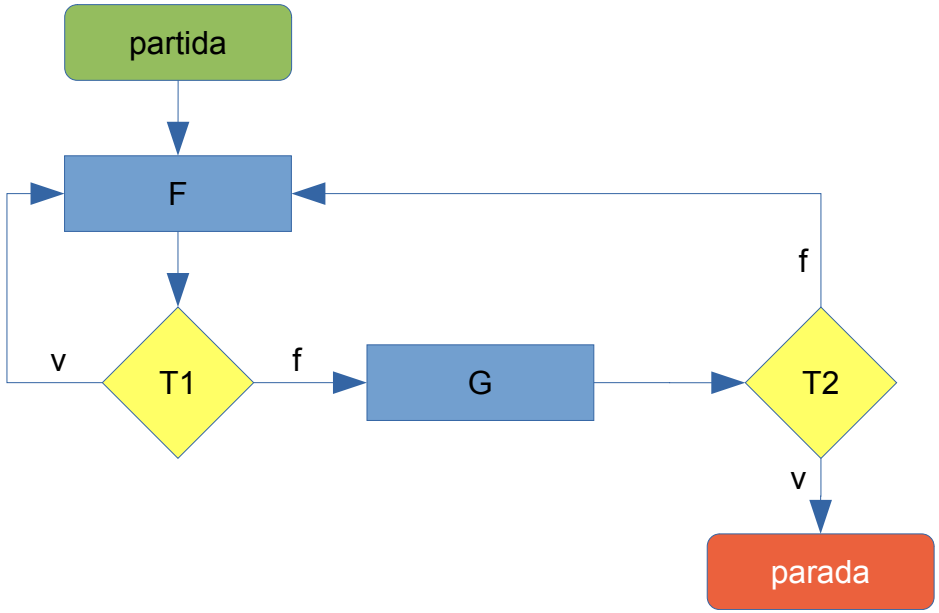
- Um Programa Monolítico é um par ordenado P = (I, R), onde:
  - I - Conjunto de Instruções Rotuladas o qual é finito.
  - R - Rótulo Inicial o qual distingue a instrução rotulada inicial em I.
- Observações:
  - Não existem duas instruções diferentes com um mesmo rótulo.
  - Um rótulo referenciado por alguma instrução o qual não é associado a qualquer instrução rotulada é dito um Rótulo Final.

Copyright © 2009/2025 • ybadoo.com.br

ybadoo

Exemplo

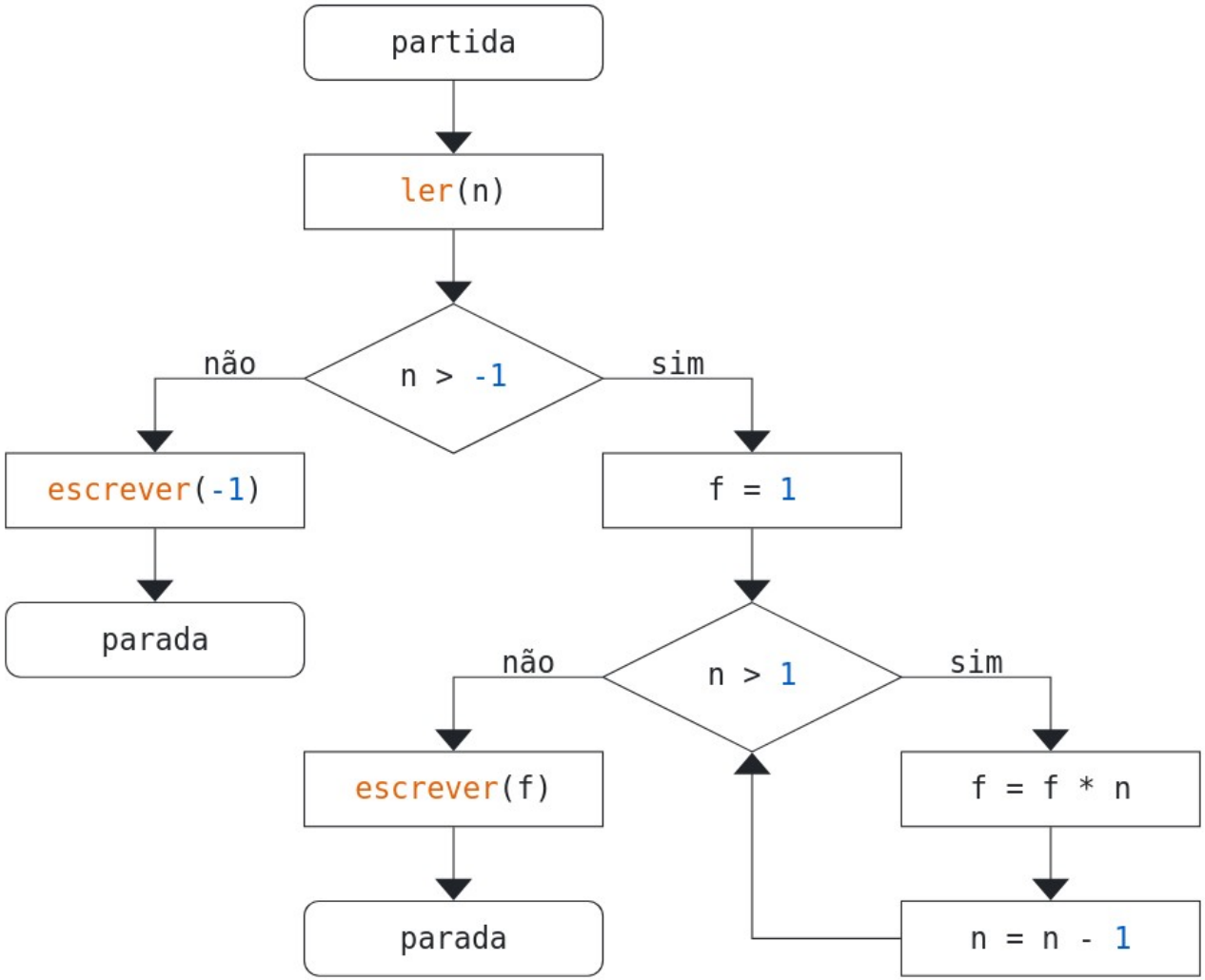
```
R1: faça F vá_para R2;
R2: se T1 então vá_para R1 senão vá_para R3;
R3: faça G vá_para R4;
R4: se T2 então vá_para R5 senão vá_para R1;
```



Copyright © 2009/2025 • ybadoo.com.br

ybadoo

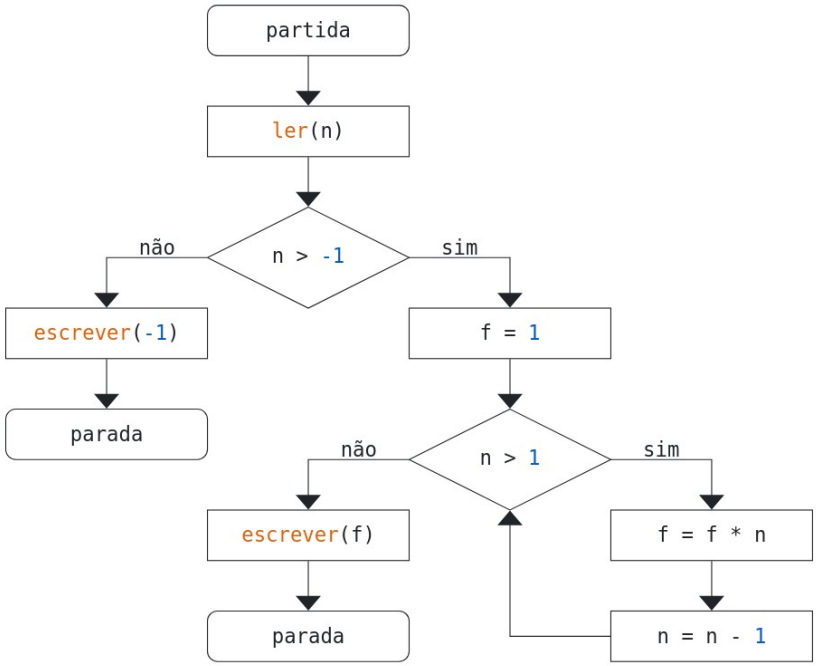
Fatorial - Fluxograma



Copyright © 2009/2025 • ybadoo.com.br

Fatorial - Instruções Rotuladas

```
R1: faça ler(n) vá_para R2;
R2: se n > -1 então vá_para R3 senão vá_para R8;
R3: faça f = 1 vá_para R4;
R4: se n > 1 então vá_para R5 senão vá_para R7;
R5: faça f = f * n vá_para R6;
R6: faça n = n - 1 vá_para R4;
R7: faça escrever(f) vá_para R0;
R8: faça escrever(-1) vá_para R0;
```



Copyright © 2009/2025 • ybadoo.com.br

Fatorial - Assembly (Intel)

```
        mov eax, 5
        mov ebx, 1
L1:     cmp eax, 0          // compara 0 com o valor em eax
        je L2               // pula para L2 se 0 == eax
        imul ebx, eax       // ebx = ebx * eax
        dec eax             // decrementa eax
        jmp L1              // desvio incondicional para L1
L2:     ret
```


Copyright © 2009/2025 • ybadoo.com.br

ybadoo

Ofuscação de código (1/3)

- Modificar o código do software para aumentar a complexidade e reduzir a legibilidade para impedir a compreensão e a análise.
  - Proteger o código de ser submetido a engenharia reversa ou adulterado.
  - Técnicas:
    - Renomeação de variáveis e métodos
    - Alteração do fluxo de controle
    - Criptografia de strings
    - Inserção de código supérfluo
    - Técnicas antidepuração
    - Virtualização de código

Copyright © 2009/2025 • ybadoo.com.br

ybadoo

Ofuscação de código (2/3)

- Por que usar um ofuscador de código:
  - Proteger a propriedade intelectual
  - Prevenir a engenharia reversa
  - Aumentar a segurança
  - Evitar adulteração do código
  - Melhorar o desempenho
  - Blindar a lógica do código
  - Controle de licenciamento
  - Proteção contra ataques automatizados

Copyright © 2009/2025 • ybadoo.com.br

ybadoo

Ofuscação de código (3/3)

```
R1: faça ler(n) vá_para R2;
R2: se n > -1 então vá_para R3 senão vá_para R8;
R3: faça f = 1 vá_para R4;
R4: se n > 1 então vá_para R5 senão vá_para R7;
R5: faça f = f * n vá_para R6;
R6: faça n = n - 1 vá_para R4;
R7: faça escrever(f) vá_para R0;
R8: faça escrever(-1) vá_para R0;
```

Normal

Ofuscado

```
R1: faça ler(a) vá_para R8;
R2: faça escrever(-1) vá_para R0;
R3: faça a = a - 1 vá_para R7;
R4: faça b = b * a vá_para R3;
R5: faça b = 1 vá_para R7;
R6: faça escrever(b) vá_para R0;
R7: se a > 1 então vá_para R4 senão vá_para R6;
R8: se a > -1 então vá_para R5 senão vá_para R2;
```


Copyright © 2009/2025 • ybadoo.com.br

ybadoo