

Estruturas de Controle em Sentenças

Eduardo Piveta

Introdução

- Computações em LPs imperativas são realizados ao avaliar expressões e atribuir os valores resultantes a variáveis
- Poucos programas úteis podem ser escritos usando somente tais instruções
- Dois mecanismos linguísticos adicionais são necessários:
 - Estruturas condicionais
 - Estruturas de repetição
- Tais mecanismos são chamados de **sentenças de controle**

Introdução

- A pesquisa em torno de sentenças de controle focou basicamente nas décadas de 60 e 70, com as seguintes conclusões:
 - Um único mecanismo de controle é suficiente (desvio incondicional - goto), mas indesejável
 - Um conjunto pequeno de sentenças de controle são capazes de representar todos os algoritmos expressos em gráficos de fluxo:
 - Um desvio condicional
 - Um laço de repetição
 - As LPs modernas normalmente não fornecem um mecanismo de desvio incondicional

Estruturas de Controle

- Estrutura de Controle = Sentença de Controle + Sentenças Controladas

- Exemplo:

```
if (x > 20) {  
    x++;  
} else {  
    printf("Bla bla bla");  
    ...  
}
```

Sentenças de Seleção

- Uma sentença de seleção fornece os mecanismos para escolher entre dois ou mais fluxos de execução em um programa
- Elas se dividem em duas categorias gerais:
 - Sentenças de dois caminhos
 - Sentenças de n caminhos (seleção múltipla)

Sentenças de Seleção de Dois Caminhos

- Forma geral:
if expressão_de_controle
 cláusula **then**
 cláusula **else**
- Questões de projeto de LPs:
 - Qual é a forma e o tipo de expressão que controla a seleção? (C89, Python, C99, C++ vs. Ada, Java, Ruby e C#)
 - Como as cláusulas **then** e **else** são especificadas?
 - (parênteses vs. then explícito – Pascal, Ruby)
 - (sentenças simples vs. sentenças compostas)
 - (chaves, palavras ou indentação – begin...end - Pascal, if ... endif – Fortran, Ada, Ruby, indentação e vírgula - Python)
 - Como o significado de seletores aninhados deve ser especificado?

Lógica Booleana

- É um cálculo lógico que verifica a veracidade de expressões
 - Usa valores “verdadeiro” e “falso” para expressar essa veracidade.
- Existem diversas operações no cálculo proposicional (que avalia proposições lógicas).
 - Conjunção (e), Disjunção (ou), Negação (não), Condicional (se...então), etc.

Operadores Relacionais

- São avaliados como expressões booleanas
- Operadores Binários (sintaxe Lógica e Alg.):
 - Menor ($a < b$)
 - Maior ($a > b$)
 - Maior ou igual ($a \geq b$)
 - Menor ou igual ($a \leq b$)
 - Igual ($a = b$)
 - Diferente ($a \neq b$)

Operadores e Constantes Booleanas

- Operadores Binários
 - Conjunção (a **e** b)
 - Disjunção (a **ou** c)
- Operadores Unários
 - Negação (**não** a)
- Constantes Booleandas
 - verdadeiro (**V**)
 - falso (**F**)

Tabelas verdade

a	a	a e b
V	V	V
F	V	F
V	F	F
F	F	F

a	a	a ou b
V	V	V
F	V	V
V	F	V
F	F	F

a	não a
V	F
F	V

Expressões booleanas

- Expressões contendo operadores booleanos.

Exemplos:

$a < 5$

$(\text{n\~{a}o}(d)) \text{ ou } (a > b \text{ e } d)$

$(b \geq 10 \text{ e } c = 20) \text{ ou } (a > b)$

$(a > 10 \text{ e } b \neq 20) \text{ ou } (a \neq 2 \text{ ou } b = 5) \text{ ou } \text{n\~{a}o}(d)$

$\text{n\~{a}o}(d) \text{ e } (d = F) \text{ ou } (d \text{ e } b \leq 5)$

Expressões booleanas

- Considere, por ex., $a = 1$, $b = 12$, $c = 3$ e $d = V$

$a < 5$

$(\text{não}(d))$ ou $(a > b \text{ e } d)$

$(b \geq 10 \text{ e } c = 20)$ ou $(a > b)$

$(a > 10 \text{ e } b \neq 20)$ ou $(a \neq 2 \text{ ou } b = 5)$ ou $\text{não}(d)$

$\text{não}(d) \text{ e } (d = F) \text{ ou } (d \text{ e } b \leq 5)$

a	a	a e b
V	V	V
F	V	F
V	F	F
F	F	F

a	a	a ou b
V	V	V
F	V	V
V	F	V
F	F	F

a	não a
V	F
F	V

Expressões booleanas

- Considere, por ex., $a = 1$, $b = 12$, $c = 3$ e $d = V$

$1 < 5$

$(\text{não}(d))$ ou $(a > b \text{ e } d)$

$(b \geq 10 \text{ e } c = 20)$ ou $(a > b)$

$(a > 10 \text{ e } b \neq 20)$ ou $(a \neq 2 \text{ ou } b = 5)$ ou $\text{não}(d)$

$\text{não}(d) \text{ e } (d = F)$ ou $(d \text{ e } b \leq 5)$

a	a	a e b
V	V	V
F	V	F
V	F	F
F	F	F

a	a	a ou b
V	V	V
F	V	V
V	F	V
F	F	F

a	não a
V	F
F	V

Expressões booleanas

- Considere, por ex., $a = 1$, $b = 12$, $c = 3$ e $d = V$

V

(não(d)) ou ($a > b$ e d)

($b \geq 10$ e $c = 20$) ou ($a > b$)

($a > 10$ e $b \neq 20$) ou ($a \neq 2$ ou $b = 5$) ou não(d)

não(d) e ($d = F$) ou (d e $b \leq 5$)

a	a	a e b
V	V	V
F	V	F
V	F	F
F	F	F

a	a	a ou b
V	V	V
F	V	V
V	F	V
F	F	F

a	não a
V	F
F	V

Expressões booleanas

- Considere, por ex., $a = 1$, $b = 12$, $c = 3$ e $d = V$

V

(**n**ão(d)) **ou** ($a > b$ **e** d)

($b \geq 10$ e $c = 20$) ou ($a > b$)

($a > 10$ e $b \neq 20$) ou ($a \neq 2$ ou $b = 5$) ou não(d)

não(d) e ($d = F$) ou (d e $b \leq 5$)

a	a	a e b
V	V	V
F	V	F
V	F	F
F	F	F

a	a	a ou b
V	V	V
F	V	V
V	F	V
F	F	F

a	n ão a
V	F
F	V

Expressões booleanas

- Considere, por ex., $a = 1$, $b = 12$, $c = 3$ e $d = V$

V

(**n**ão(V)) **ou** ($1 > 12$ **e** V)

($b \geq 10$ e $c = 20$) ou ($a > b$)

($a > 10$ e $b \neq 20$) ou ($a \neq 2$ ou $b = 5$) ou não(d)

não(d) e ($d = F$) ou (d e $b \leq 5$)

a	a	a e b
V	V	V
F	V	F
V	F	F
F	F	F

a	a	a ou b
V	V	V
F	V	V
V	F	V
F	F	F

a	não a
V	F
F	V

Expressões booleanas

- Considere, por ex., $a = 1$, $b = 12$, $c = 3$ e $d = V$

V

(F) ou (F e V)

$(b \geq 10 \text{ e } c = 20) \text{ ou } (a > b)$

$(a > 10 \text{ e } b \neq 20) \text{ ou } (a \neq 2 \text{ ou } b = 5) \text{ ou não}(d)$

não(d) e $(d = F) \text{ ou } (d \text{ e } b \leq 5)$

a	a	a e b
V	V	V
F	V	F
V	F	F
F	F	F

a	a	a ou b
V	V	V
F	V	V
V	F	V
F	F	F

a	não a
V	F
F	V

Expressões booleanas

- Considere, por ex., $a = 1$, $b = 12$, $c = 3$ e $d = V$

V

(F) ou (F)

$(b \geq 10 \text{ e } c = 20) \text{ ou } (a > b)$

$(a > 10 \text{ e } b \neq 20) \text{ ou } (a \neq 2 \text{ ou } b = 5) \text{ ou não}(d)$

não(d) e $(d = F) \text{ ou } (d \text{ e } b \leq 5)$

a	a	a e b
V	V	V
F	V	F
V	F	F
F	F	F

a	a	a ou b
V	V	V
F	V	V
V	F	V
F	F	F

a	não a
V	F
F	V

Expressões booleanas

- Considere, por ex., $a = 1$, $b = 12$, $c = 3$ e $d = V$

V

F ou F

$(b \geq 10 \text{ e } c = 20) \text{ ou } (a > b)$

$(a > 10 \text{ e } b \neq 20) \text{ ou } (a \neq 2 \text{ ou } b = 5) \text{ ou não}(d)$

$\text{não}(d) \text{ e } (d = F) \text{ ou } (d \text{ e } b \leq 5)$

a	a	a e b
V	V	V
F	V	F
V	F	F
F	F	F

a	a	a ou b
V	V	V
F	V	V
V	F	V
F	F	F

a	não a
V	F
F	V

Expressões booleanas

- Considere, por ex., $a = 1$, $b = 12$, $c = 3$ e $d = V$

V

F

$(b \geq 10 \text{ e } c = 20) \text{ ou } (a > b)$

$(a > 10 \text{ e } b \neq 20) \text{ ou } (a \neq 2 \text{ ou } b = 5) \text{ ou não}(d)$

$\text{não}(d) \text{ e } (d = F) \text{ ou } (d \text{ e } b \leq 5)$

a	a	a e b
V	V	V
F	V	F
V	F	F
F	F	F

a	a	a ou b
V	V	V
F	V	V
V	F	V
F	F	F

a	não a
V	F
F	V

Expressões booleanas

- Considere, por ex., $a = 1$, $b = 12$, $c = 3$ e $d = V$

V

F

$(b \geq 10 \text{ e } c = 20) \text{ ou } (a > b)$

$(a > 10 \text{ e } b \neq 20) \text{ ou } (a \neq 2 \text{ ou } b = 5) \text{ ou não}(d)$

$\text{não}(d) \text{ e } (d = F) \text{ ou } (d \text{ e } b \leq 5)$

a	a	a e b
V	V	V
F	V	F
V	F	F
F	F	F

a	a	a ou b
V	V	V
F	V	V
V	F	V
F	F	F

a	não a
V	F
F	V

Expressões booleanas

- Considere, por ex., $a = 1$, $b = 12$, $c = 3$ e $d = V$

V

F

$(12 \geq 10 \text{ e } 3 = 20) \text{ ou } (1 > 12)$

$(a > 10 \text{ e } b \neq 20) \text{ ou } (a \neq 2 \text{ ou } b = 5) \text{ ou não}(d)$

$\text{não}(d) \text{ e } (d = F) \text{ ou } (d \text{ e } b \leq 5)$

a	a	a e b
V	V	V
F	V	F
V	F	F
F	F	F

a	a	a ou b
V	V	V
F	V	V
V	F	V
F	F	F

a	não a
V	F
F	V

Expressões booleanas

- Considere, por ex., $a = 1$, $b = 12$, $c = 3$ e $d = V$

V

F

(V e F) ou (F)

$(a > 10 \text{ e } b \neq 20) \text{ ou } (a \neq 2 \text{ ou } b = 5) \text{ ou não}(d)$

$\text{não}(d) \text{ e } (d = F) \text{ ou } (d \text{ e } b \leq 5)$

a	a	a e b
V	V	V
F	V	F
V	F	F
F	F	F

a	a	a ou b
V	V	V
F	V	V
V	F	V
F	F	F

a	não a
V	F
F	V

Expressões booleanas

- Considere, por ex., $a = 1$, $b = 12$, $c = 3$ e $d = V$

V

F

F ou **F**

$(a > 10 \text{ e } b \neq 20) \text{ ou } (a \neq 2 \text{ ou } b = 5) \text{ ou não}(d)$

$\text{não}(d) \text{ e } (d = F) \text{ ou } (d \text{ e } b \leq 5)$

a	a	a e b
V	V	V
F	V	F
V	F	F
F	F	F

a	a	a ou b
V	V	V
F	V	V
V	F	V
F	F	F

a	não a
V	F
F	V

Expressões booleanas

- Considere, por ex., $a = 1$, $b = 12$, $c = 3$ e $d = V$

V

F

F

$(a > 10 \text{ e } b \neq 20) \text{ ou } (a \neq 2 \text{ ou } b = 5) \text{ ou não}(d)$
 $\text{não}(d) \text{ e } (d = F) \text{ ou } (d \text{ e } b \leq 5)$

a	a	a e b
V	V	V
F	V	F
V	F	F
F	F	F

a	a	a ou b
V	V	V
F	V	V
V	F	V
F	F	F

a	não a
V	F
F	V

Expressões booleanas

- Considere, por ex., $a = 1$, $b = 12$, $c = 3$ e $d = V$

V

F

F

$(a > 10 \text{ e } b \neq 20) \text{ ou } (a \neq 2 \text{ ou } b = 5) \text{ ou não}(d)$

$\text{não}(d) \text{ e } (d = F) \text{ ou } (d \text{ e } b \leq 5)$

a	a	a e b
V	V	V
F	V	F
V	F	F
F	F	F

a	a	a ou b
V	V	V
F	V	V
V	F	V
F	F	F

a	não a
V	F
F	V

Expressões booleanas

- Considere, por ex., $a = 1$, $b = 12$, $c = 3$ e $d = V$

V

F

F

$(1 > 10 \text{ e } 12 \neq 20) \text{ ou } (1 \neq 2 \text{ ou } 12 = 5) \text{ ou não}(V)$

$\text{não}(d) \text{ e } (d = F) \text{ ou } (d \text{ e } b \leq 5)$

a	a	a e b
V	V	V
F	V	F
V	F	F
F	F	F

a	a	a ou b
V	V	V
F	V	V
V	F	V
F	F	F

a	não a
V	F
F	V

Expressões booleanas

- Considere, por ex., $a = 1$, $b = 12$, $c = 3$ e $d = V$

V

F

F

(F e V) ou (V ou F) ou F

não(d) e (d = F) ou (d e b <=5)

a	a	a e b
V	V	V
F	V	F
V	F	F
F	F	F

a	a	a ou b
V	V	V
F	V	V
V	F	V
F	F	F

a	não a
V	F
F	V

Expressões booleanas

- Considere, por ex., $a = 1$, $b = 12$, $c = 3$ e $d = V$

V

F

F

(F) ou (V) ou F

não(d) e (d = F) ou (d e b <=5)

a	a	a e b
V	V	V
F	V	F
V	F	F
F	F	F

a	a	a ou b
V	V	V
F	V	V
V	F	V
F	F	F

a	não a
V	F
F	V

Expressões booleanas

- Considere, por ex., $a = 1$, $b = 12$, $c = 3$ e $d = V$

V

F

F

V ou F

não(d) e (d = F) ou (d e b <=5)

a	a	a e b
V	V	V
F	V	F
V	F	F
F	F	F

a	a	a ou b
V	V	V
F	V	V
V	F	V
F	F	F

a	não a
V	F
F	V

Expressões booleanas

- Considere, por ex., $a = 1$, $b = 12$, $c = 3$ e $d = V$

V

F

F

V

não(d) e (d = F) ou (d e b <=5)

a	a	a e b
V	V	V
F	V	F
V	F	F
F	F	F

a	a	a ou b
V	V	V
F	V	V
V	F	V
F	F	F

a	não a
V	F
F	V

Expressões booleanas

- Considere, por ex., $a = 1$, $b = 12$, $c = 3$ e $d = V$

V

F

F

V

não(d) e (d = F) ou (d e b <=5)

a	a	a e b
V	V	V
F	V	F
V	F	F
F	F	F

a	a	a ou b
V	V	V
F	V	V
V	F	V
F	F	F

a	não a
V	F
F	V

Expressões booleanas

- Considere, por ex., $a = 1$, $b = 12$, $c = 3$ e $d = V$

V

F

F

V

não(V) e (V = F) ou (V e 12 <=5)

a	a	a e b
V	V	V
F	V	F
V	F	F
F	F	F

a	a	a ou b
V	V	V
F	V	V
V	F	V
F	F	F

a	não a
V	F
F	V

Expressões booleanas

- Considere, por ex., $a = 1$, $b = 12$, $c = 3$ e $d = V$

V

F

F

V

F e F ou (V e F)

a	a	a e b
V	V	V
F	V	F
V	F	F
F	F	F

a	a	a ou b
V	V	V
F	V	V
V	F	V
F	F	F

a	não a
V	F
F	V

Expressões booleanas

- Considere, por ex., $a = 1$, $b = 12$, $c = 3$ e $d = V$

V

F

F

V

F e F ou F

a	a	a e b
V	V	V
F	V	F
V	F	F
F	F	F

a	a	a ou b
V	V	V
F	V	V
V	F	V
F	F	F

a	não a
V	F
F	V

Expressões booleanas

- Considere, por ex., $a = 1$, $b = 12$, $c = 3$ e $d = V$

V

F

F

V

F ou F

a	a	a e b
V	V	V
F	V	F
V	F	F
F	F	F

a	a	a ou b
V	V	V
F	V	V
V	F	V
F	F	F

a	não a
V	F
F	V

Expressões booleanas

- Considere, por ex., $a = 1$, $b = 12$, $c = 3$ e $d = V$

V

F

F

V

F

a	a	a e b
V	V	V
F	V	F
V	F	F
F	F	F

a	a	a ou b
V	V	V
F	V	V
V	F	V
F	F	F

a	não a
V	F
F	V

Declarando e usando variáveis booleanas

- Variáveis booleanas podem ser criadas usando o tipo **booleano**. Exemplo:

```
b: booleano
```

```
b := V
```

```
se b
```

```
    escrever "b é verdadeiro"
```

```
c: booleano
```

```
c := não b
```

```
se c = b
```

```
    escrever "O programa nunca chega aqui"
```

Lógica e Algoritmos: Sintaxe da Seleção

se <expressão_booleana>

sentença_1

sentença_n

[então

sentença_1

sentença_n

]

Exemplo 1: Somente cláusula **se**

se $a \neq 0$

escrever "a não pode ser nulo"

Exemplo 2: Cláusulas **se** e **senão**

se $a \neq 0$

 escrever "a não pode ser nulo"

senão

$x := 10 / a;$

Exemplo 2: Múltiplas linhas em cada bloco

se $a \neq 0$

 escrever "a não pode ser nulo"

$a := 10$

senão

x : inteiro

$x := 10 / a$;

Seletores aninhados

- Podem causar ambiguidade. Tal ambiguidade deve ser resolvida pelo projetista da LP.
- Exemplo:

```
if (sum == 0)
    if (count == 0)
        result = 0;
else
    result = 1;
```
- A indentação parece indicar que o **else** é referente ao primeiro **if**.
 - No entanto, a maioria das linguagens ignora indentação
 - Normalmente o **else** casa com o **if** anterior mais próximo

Seletores aninhados - Perl

- Outras linguagens forçam o uso de sentenças compostas (e.g. Perl)

```
if (sum == 0){  
    if (count == 0) {  
        result = 0;  
    }  
} else {  
    result = 1;  
}
```

```
if (sum == 0){  
    if (count == 0) {  
        result = 0;  
    }  
    else {  
        result = 1;  
    }  
}
```

Seletores aninhados – Ada, Fortran95, Ruby

if a > b then

sum = sum + a

acount = acount + 1

else

sum = sum + b

bcount = bcount + 1

end

Seletores aninhados - Ruby

- Tal construção é mais regular que nas LPs baseadas em C:

```
if (sum == 0) then
  if (count == 0) then
    result = 0
  else
    result = 1
  end
end
```

```
if (sum == 0) then
  if (count == 0) then
    result = 0
  end
else
  result = 1
end
```

Lógica e Algoritmos – Sintaxe

Aninhamento

se $a \neq 0$

 escrever "a não pode ser nulo"

se $a > 1$ e $a < 10$

 escrever "A é maior que 1 e menor que 10"

senão

 escrever "A é menor que um ou maior que 10"

$a := 10$

senão

x : inteiro

$x := 10 / a$;

Lógica e Algoritmos – Sintaxe

Aninhamento

se a != 0

escrever "a não pode ser nulo"

se a > 1 e a < 10

escrever "A é maior que 1 e menor que 10"

senão

escrever "A é menor que um ou maior que 10"

a := 10

senão

x: inteiro

x := 10 / a;

Lógica e Algoritmos – Sintaxe

Aninhamento

se $a \neq 0$

escrever "a não pode ser nulo"

se $a > 1$ e $a < 10$

escrever "A é maior que 1 e menor que 10"

senão

escrever "A é menor que um ou maior que 10"

$a := 10$

senão

x : inteiro

$x := 10 / a$;

Construções de Seleção Múltipla

- Seleção de um ou mais sentenças ou grupos de sentenças
- A necessidade de escolher dentre mais de dois caminhos de controle em um programa é comum
- Tais construções podem ser construídas usando apenas seleções de dois caminhos ou desvios incondicionais
 - No entanto, são mais difíceis de escrever, ler e manter do que construções próprias para isso

Exemplos de Seletores Múltiplos

- C, Java, JavaScript:
switch (expressão) {
 case expressão_constante_1: sentença_1;
 ...
 case expressão_constante_n: sentença_n;
 [**default**: sentença_n+1]
}
- A expressão de controle e expressão constante são de algum tipo inteiro
- Não existem desvios implícitos no final dos segmentos
 - Isso permite que o controle flua através de mais de um segmento

Exemplos de Seletores Múltiplos

```
switch (indice){  
    case 1:  
    case 3: impar += 1;  
            somaimpar += indice;  
            break;  
    case 2:  
    case 4: par += 1;  
            somapar += indice;  
            break;  
    default: printf("Erro no switch, indice = %d\n", indice);  
}
```

Exemplos de Seletores Múltiplos – C#

- A semântica da LP não permite a execução implícita de mais de um segmento
 - Cada segmento selecionável deve terminar com uma sentença de desvio incondicional explícita: **break** ou **goto**

```
switch (valor){  
    case -1: negatives++; break;  
    case 0: zeros++; goto case 1;  
    case 1: ...  
    default: Console.WriteLine("Error");  
}
```

Seletores Múltiplos usando IFs

- Em Python:

```
if count < 10 :
```

```
    bag1 = True
```

```
elif count < 100 :
```

```
    bag2 = True
```

```
elif count < 1000 :
```

```
    bag3 = True
```

Lógica e Algoritmos - Sintaxe

Desvio Incondicional

- Desvios incondicionais transferem o controle da execução para uma localização específica em um programa
 - **GOTO**
- Tais sentenças tornam os programas bastante difíceis de ler, não confiáveis e caros para manter (manualmente).
- Algumas LPs não possuem GOTOs explícitos: Java, Python, Ruby.
- No entanto, desvios incondicionais serão usados em algumas disciplinas (Compiladores, por ex.)

Estruturas de Controle em Sentenças

Eduardo Piveta