7. Estruturas condicionais

DIM0320

2015.1

Sumário

Constatação

- No momento, todos os algoritmos são uma sequência de comandos executados na ordem definida pela sequência.
  - É uma linha reta do início ao fim do algoritmo
- Essa aula introduz um **novo comando** que permite
  - "bifurcar" essa sequência linear,
  - dissociar várias possibilidades, vários comportamentos/ fluxos de dados dentro do mesmo algoritmo.

Um algoritmo problemático

```
algoritmo "Pessoa mais alta - incompleto"
var nome1, nome2 : caractere
    alt1, alt2 : real
inicio
escreva("Entre com o nome da primeira pessoa: ")
leia(nome1)

escreva("Entre com a altura da primeira pessoa: ")
leia(alt1)

escreva("Entre com o nome da segunda pessoa: ")
leia(nome2)

escreva("Entre com a altura da segunda pessoa: ")
leia(alt2)

// Escreva o nome da pessoa de maior altura
fimalgoritmo
```

Soluções?

- **Hipótese**: alt1 <> alt2
- Tentativa
  - Se a primeira pessoa for mais alta, queremos escrever a saída
    - `escreva(nome1 , " é mais alto(a) do que " , nome2)` ou
  - caso contrário:
    - `escreva(nome2 , " é mais alto(a) do que " , nome1)`
- Não sabemos com antecedência quem é a pessoa mais alta porque depende de entradas desconhecidas no momento da escritura do algoritmo.
- A dependência às entradas proíbe que possamos escolher uma solução para **todas** ocorrências
- É preciso mais uma estrutura sintática para especificar essa situação de exclusão mútua.

Sintaxe

Definição (Forma)

```
se <expr> entao
   <cmdo_1>
   ..
   <cmdo_n>
senao
   <cmdo_1>
   ..
   <cmdo_m>
fimse
```

Observações

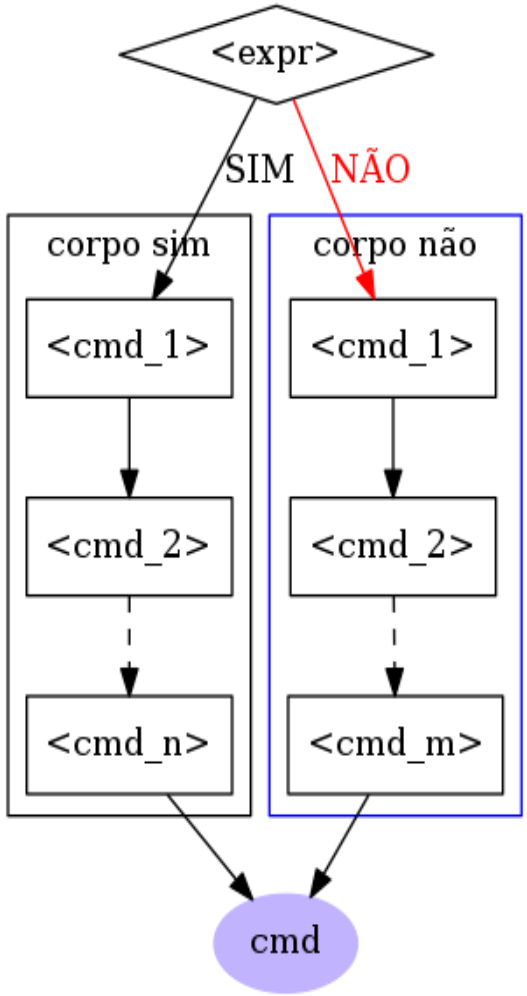
- <expr> deve ser do tipo `logico`
- Os comandos podem ser qualquer comandos a serem vistos
- Das 2 sequências de comandos só **uma** sera executada
- Por enquanto, assuma que a expressão lógica é uma expressão relacional.

Semântica

Avaliação

1. <expr> é avaliada
2. se <expr> for `verdadeiro`, o algoritmo executa a seção entre o `entao` e o `senao` (i.e. `cmd_1`, ..., <cmd_n>)
3. se <expr> for `falso`, o algoritmo executa a seção entre o `senao` e o `fimse` (i.e. `cmd_1`, ..., <cmd_n>)
4. depois da execução de 2 ou 3, o algoritmo executa o primeiro comando após `fimse`

Representação do fluxo de controle.



Voltando ao exemplo

```
algoritmo "Pessoa mais alta"
var nome1, nome2 : caractere
    alt1, alt2 : real
inicio
  escreva("Entre com o nome da primeira pessoa: ")
  leia(nome1)

  escreva("Entre com a altura da primeira pessoa: ")
  leia(alt1)

  escreva("Entre com o nome da segunda pessoa: ")
  leia(nome2)

  escreva("Entre com a altura da segunda pessoa: ")
  leia(alt2)

  se alt1 < alt2 entao
    escreva("A segunda pessoa e mais alta")
  senao
    escreva("A primeira pessoa e mais alta")
  fimse
fimalgoritmo
```

Aninhamento de comandos condicionais no exemplo inicial

Modificação

Tiramos a hipótese `alt1 <> alt2`, podemos agora ter `alt = alt2`

```
algoritmo "Pessoa mais alta 2"
var nome1, nome2 : caractere
    alt1, alt2 : real
inicio
  leia(nome1, alt1, nome2, alt2)     // Leituras condensadas
  se alt1 < alt2 entao
    escreva("A segunda pessoa e mais alta")
  senao
    se alt1 > alt2 entao
      escreva("A primeira pessoa e mais alta")
    senao
      escreva("Nenhuma pessoa e mais alta")
    fimse
  fimse
fimalgoritmo
```

- **Qualquer comando pode ser escrito dentro de uma parte do comando condicional**.
- Isto significa que pode ser um **outro** comando condicional.

Mais um exemplo

Assunto

Escrever um algoritmo que:

- lê as notas de 3 provas de um aluno,
- calcula a média aritmética e
- escreve a situação dele como saída.

A situação pode ser

- "aprovado" (média >= 6.8),
- "reprovado" (média <= 3.4) ou
- "em exame" (média > 3.4 e < 6.8).

Solução

```
algoritmo "Situacao escolar de aluno"
var n1, n2, n3, media : real
inicio
  escreva("Entre com a primeira nota: ")
  leia(n1)
  escreva("Entre com a segunda nota: ")
  leia(n2)
  escreva("Entre com a terceira nota: ")
  leia(n3)

  media <- (n1 + n2 + n3) / 3
  se media >= 6.8 entao
    escreva("aprovado")
  senao // media < 6.8
    se media > 3.4 entao
      escreva("em exame")
    senao // media <= 3.4 e media < 6.8
      escreva("reprovado")
    fimse
  fimse
fimalgoritmo
```

O comando se-então-fimse.

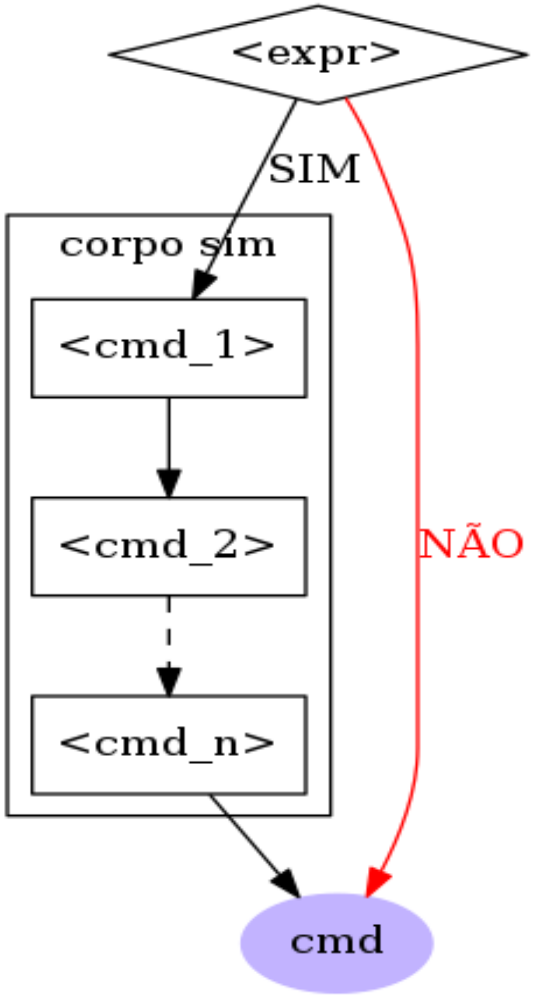
Sintaxs

```
se <expressao_logica> entao
   <comando_1>
   ..
   <comando_n>
fimse
```

Observação

- É uma forma simplificada do comando `se-entao-senao-fimse` onde segundo bloco `senao` é vazio.

Representação do fluxo de controle



Exemplo

Assunto

O salário líquido é 70% do salário bruto. Um funcionário recebe um acréscimo de R$ 100 se ele tem mais de 2 filhos.

Solução

```
algoritmo "Salario liquido mensal de funcionario"
var
 salbruto, salario : real
 filhos : inteiro
inicio
 escreva("Entre com o salario bruto mensal: ")
 leia (salbruto)

 escreva("Entre com o numero de filhos: ")
 leia (filhos)

 salario <- 0.7 * salbruto
 se filhos > 2 entao
   salario <- salario + 100
 fimse

 escreva ("O salario liquido e: ", salario)
fimalgoritmo
```

Trocar 2 variáveis

Objetivo

Sejam x, y 2 variáveis de tipo inteiro com valores respetivos de 4 e 5. Após a troca, x terá um valor de 5 e y de 4.

Trocar 2 variáveis

Objetivo

Sejam x, y 2 variáveis de tipo inteiro com valores respetivos de 4 e 5. Após a troca, x terá um valor de 5 e y de 4.

Versão 1

```
x <- y
y <- x
```

Trocar 2 variáveis

Objetivo

Sejam x, y 2 variáveis de tipo inteiro com valores respetivos de 4 e 5. Após a troca, x terá um valor de 5 e y de 4.

Versão 1

```
x <- y
y <- x
```

Versão 2

```
z <- y
y <- x
x <- z
```

Números em ordem crescente

Assunto

Escreva um algoritmo que lê 2 números reais, e escreve os números em ordem crescente (não estrita).
Após a leitura dos dados, pode usar apenas **um** comando escreva.

Números em ordem crescente

Assunto

Escreva um algoritmo que lê 2 números reais, e escreve os números em ordem crescente (não estrita).
Após a leitura dos dados, pode usar apenas **um** comando escreva.

```
algoritmo "Numeros em ordem nao-decrescente - primeira versao"
var
    num1, num2, num3 : real
inicio
  escreva( "Entre com o primeiro numero: " )
  leia( num1 )
  escreva( "Entre com o segundo numero: " )
  leia( num2 )
  se num1 > num2 entao
      num3 <- num1
      num1 <- num2
      num2 <- num3
  fimse
  escreva( "Os numeros em ordem nao-decrescente: " , num1 , " e " , num2 )
fimalgoritmo
```

Um novo comando

Observação

Aninhamento de `se-entao-senao-fimse` pode dificultar a leitura.

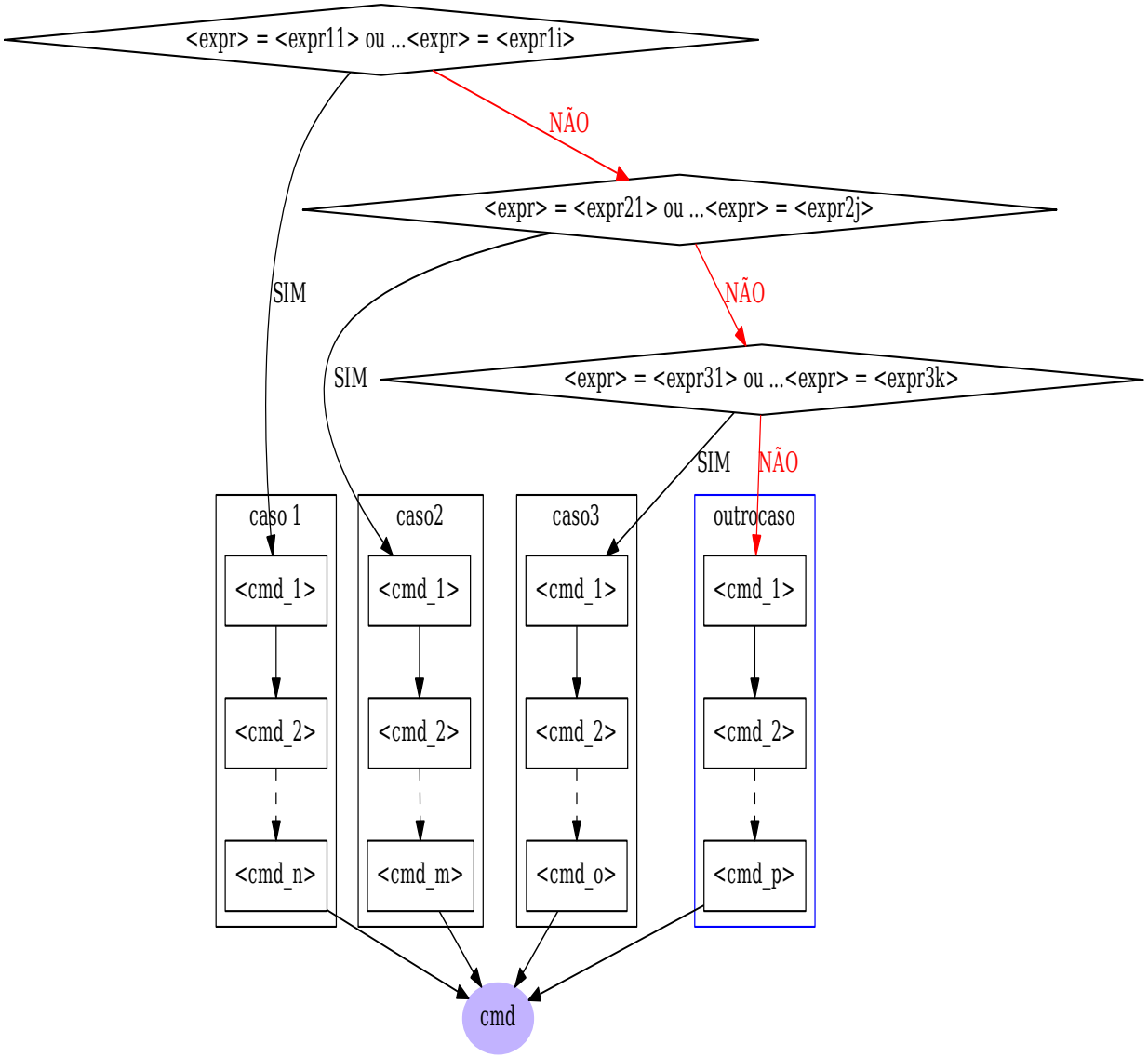
Definição (Sintaxe)

```
escolha <expr>
  caso <expr11>, <expr12>, .., <expr1i>
    <cmd11>
     ..
    <cmd1m>
  caso <expr21>, <expr22>, .., <expr2j>
    <cmd21>
    ..
    <cmd2n>
  caso <expr31>, <expr32>, .., <expr3k>
    <cmd31>
    ..
    <cmd3o>
  ..

  outrocaso
    <cmd1>
    ..
    <cmdp>

fimescolha
```

Semântica informal

- Avaliar `<expr>`, produzindo o valor `v`
- Se `v = <expr11> ou v = <expr12> ou ... ou v = <expr1i>`, execute `<cmd11> ... <cmd1m>`
- Senão se `v = <expr21> ou v = <expr22> ou ... ou v = <expr2j>`, execute `<cmd21> ... <cmd2n>`
- Senão se `v = <expr31> ou v = <expr32> ou ... ou v = <expr3k>`, execute `<cmd31> ... <cmd3o>`
- senão, execute `<cmd1> ... <cmdp>`

Fluxograma


<expr> = <expr11> ou ...<expr> = <expr1i>
NÃO
<expr> = <expr21> ou ...<expr> = <expr2j>
SIM
NÃO
SIM
<expr> = <expr31> ou ...<expr> = <expr3k>
SIM
NÃO
caso 1
caso2
caso3
outrocaso
<cmd_1>
<cmd_2>
<cmd_n>
<cmd_m>
<cmd_o>
<cmd_p>
cmd

Exemplo de uso

```
algoritmo "Escolha"
var jogador, time: caractere
inicio
  escreva("Entre com um nome de jogador: ")
  leia(jogador)
  escolha jogador
  caso "Reus", "Khedira", "Klose", "Kroos"
       time <- "Alemanha"
  caso "Benzema", "Valbuena", "Lloris"
       time <- "França"
  caso "Fred", "Neymar", "Oscar", "Paulinho"
       time <- "Brasil"
  outrocaso
       time <- "desconhecido"
  fimescolha
  escreval("O jogador escolhido é do time: ", time)
fimalgoritmo
```

Exemplo com intervalos

```
algoritmo "Escolha"
var media : inteiro
    resultado : caractere
inicio
  escreva("Entre com a media: ")
  leia(media)
  escolha media
  caso 0..3
       resultado <- "reprovado"
  caso 3..5
       resultado <- "em reposicao"
  caso 5..10
       resultado <- "aprovado"
  outrocaso
       resultado <- "desconhecido"
  fimescolha
  escreval("O aluno esta ", resultado)
fimalgoritmo
```

Resumo

Perguntas ?

http://dimap.ufrn.br/~richard/dim0320

Avaliação de programa

Escrever a saída (na tela) do seguinte algoritmo quando ele for executado nas seguintes entradas:

1. 2
2. 21

```
1  algoritmo "Misterioso"
2  var n : inteiro
3  inicio
4    escreva ("Entre com um número inteiro positivo:")
5    leia (n)
6    se (n % 2) = 1 entao
7      se (n % 7) = 0 entao
8        escreva ("Passei pelo ponto 1")
9      senao
10       escreva ("Passei pelo ponto 4")
11     fimse
12   senao
13     se (n % 4) = 0 entao
14       escreva ("Passei pelo ponto 2")
15     senao
16       escreva ("Passei pelo ponto 3")
17     fimse
18    fimse
19 fimalgoritmo
```

Soma de algarismos

Assunto

Escrever um algoritmo que lê um número inteiro positivo com quatro dígitos e escreve

- "sim" se a soma dos algarismos da centena e milhar do número é par
- "não" caso contrário.

Usar o comando `escolha`

Assunto

Crie um algoritmo que solicite o número de camisa de um jogador de futebol ao usuário e escreva a posição usual ligada a este número.
A posição é definida como :

- "goleiro" se o número dado for 1 ou 12
- "lateral" se for 2 ou 3
- "zagueiro" se for 4 ou 5
- "meio-campo" se for 6, 7, 8 ou 10
- "atacante" se for 9 ou 11
- "substituto" senão

Triângulos

Assunto

Escreva um algoritmo que leia três números reais e escreva

- "equilátero" se eles formam os lados de um triângulo equilátero,
- "isósceles" se eles formam os lados de um triângulo isósceles,
- "escaleno" se eles formam os lados de um triângulo escaleno e
- "não formam os lados de um triângulo" se eles não formam os lados de um triângulo.