Cap. 4 - Expressões Condicionais e Funções

INF05008 - Fundamentos de Algoritmos

Instituto de Informática
Universidade Federal do Rio Grande do Sul
Porto Alegre, Brasil
http://www.inf.ufrgs.br

Expressões Condicionais e Funções

- Para diversos problemas, o programa deve lidar com **situações diferentes** de **formas diferentes**, como por exemplo:
  - Jogo deve determinar **se** a velocidade de um objeto está dentro de um intervalo ou **se** um objeto está em determinada posição do vídeo
  - **Se** estamos lidando com frações ou com racionais
  - Em controle de processos, uma determinada condição determina **se** válvula deve ou não ser aberta
  - ...
- Aqui precisamos de uma nova classe de valores: os **Booleanos**

Expressões Condicionais e Funções

- **Condições** são comuns em Matemática
- Exemplos: um número pode ser igual, menor ou maior do que outro número
- Se x e y são números:
  1. $x = y$,
  2. $x < y$ ou
  3. $x > y$
- Casos não-numéricos serão vistos posteriormente

Booleanos e Relações

- Para cada par x, y de números, somente uma das afirmações anteriores é verdadeira:
  - Se $x = 4$ e $y = 5$, a segunda afirmação slide anterior é **verdadeira** (*true*) e as outras são **falsas** (*false*)
  - Se $x = 5$ e $y = 4$, a terceira afirmação é **verdadeira** e as outras são **falsas**
- Em geral, uma afirmação é **verdadeira** para alguns valores e **falsa** para outros

Booleanos e Relações

- Afirmações **atômicas** podem ser combinadas em afirmações **compostas**
- Exemplos de combinações das afirmações atômicas anteriores:
  1. $x = y$ **e** $x < y$ **e** $x > y$
  2. $x = y$ **ou** $x < y$ **ou** $x > y$
  3. $x = y$ **ou** $x < y$

Booleanos e Relações

- $x = y$ **e** $x < y$ **e** $x > y$ é sempre **falsa** (false)
- $x = y$ **ou** $x < y$ **ou** $x > y$ é sempre **verdadeira** (true)
- $x = y$ **ou** $x < y$ **verdadeira** em alguns casos e **falsa** em outros casos
  - A terceira expressão acima é:
    - **verdadeira** quando $x = 4$ **e** $y = 4$
    - **verdadeira** quando $x = 4$ **e** $y = 5$
    - **falsa** quando $x = 5$ **e** $y = 3$

Booleanos e Relações

- Em Racket:
  - `true` *verdadeiro*
  - `false` *falso*
  - `(= x y)` *x é igual a y*
  - `(< x y)` *x é menor do que y*
  - `(> x y)` *x é maior do que y*
  - `(<= x y)` *x é menor ou igual a y*
  - `(>= x y)` *x é maior ou igual a y*
  - `(not expressão)` *negação de expressão Booleana*

Booleanos e Relações

- Exemplos de condições em Racket:
  - `(< 4 5)`
  - `(and (= 5 5) (< 5 6))`
  - `(and (= x y) (< y z))`
  - `(or (= x y) (< y z))`
  - `(not (and (= 5 5) (< 5 6)))`

Exercícios

Exercício 4.1.1. Qual o resultado das seguintes condições em Racket?

1. `(and (> 4 3) (<= 10 100))`
2. `(or (> 4 3) (<= 10 100))`
3. `(not (= 2 3))`

Exercício 4.1.2. Qual o resultado de

1. `(> 4 3)`
2. `(and (> 4 x) (> x 3))`
3. `(= (* x x) x)`

para:

(a) $x = 4$,

(b) $x = 2$,

(c) $x = \frac{7}{2}$

Funções que Testam Condições

- Exemplo de função que testa uma condição sobre um número:

```
;; é-5? : Número -> Booleano
;; Dado um valor numérico,
;; determina se ele é igual a 5

;; exemplos:
;; (é-5? 5) deve retornar true
;; (é-5? 9) deve retornar false
```

- Que valor a função retorna quando recebe o valor -5 ?
- A função produz `true` **se e somente se** sua entrada é igual a 5
- ATENÇÃO: aqui são necessários DOIS exemplos, onde cada um cobre um caso distinto de uso da função !!!

Funções que Testam Condições

- Exemplo de função que testa uma condição sobre um número:

```
;; é-5? : Número -> Booleano
;; Dado um valor numérico,
;; determina se ele é igual a 5

(define (é-5? n)
(= n 5))
```

Funções que Testam Condições

- Outros exemplos de funções que testam condições:

```
;; entre-5-6? : Número -> Boolean
;; Dado um número, determina se
;; ele está entre 5 e 6 (exclusivo)

;; exemplos:
;; (entre-5-6? 5) deve retornar false
;; (entre-5-6? 5.2) deve retornar true
;; (entre-5-6? 6) deve retornar false
```

- ATENÇÃO: os exemplos devem cobrir os casos LIMITE (aqueles onde pode haver dúvida)

Funções que Testam Condições

- Outros exemplos de funções que testam condições:

```
;; entre-5-6-ou-acima-10? : Número -> Boolean
;; Dado um número, determina se
;; ele está entre 5 e 6 (exclusivo)
;; ou é maior do que 10

;; exemplos: (complete agora!)
```

Funções que Testam Condições

- Outros exemplos de funções que testam condições:

```
;; entre-5-6? : Número -> Boolean
;; Dado um número, determina se
;; ele está entre 5 e 6 (exclusivo)

(define (entre-5-6? n)
(and (< 5 n) (< n 6)))

;; entre-5-6-ou-acima-10? : Número -> Boolean
;; Dado um número, determina se
;; ele está entre 5 e 6 (exclusivo)
;; ou é maior do que 10

(define (entre-5-6-ou-acima-10? n)
(or (entre-5-6? n) (>= n 10))) ;; atente para o reuso de função
```

Exercícios

Exercícios 4.2.1. Traduza os intervalos abaixo para funções em Racket que consomem (aceitam) um número e produzem `true` se o número está no intervalo e `false`, caso contrário.

1. $(3, 7]$
2. $[3, 7]$
3. $[3, 9)$
4. União de $(1, 6)$ e $(9, 14)$
5. Na parte de fora de $[1, 3]$

Exercícios (cont.)

Exercícios 4.2.2.

(a) Traduza as funções abaixo para intervalos:

```
(define (no-intervalo-1? x)
(and (< -3 x) (< x 0)))

(define (no-intervalo-2? x)
(or (< x 1) (> x 2)))

(define (no-intervalo-3? x)
(not (and (<= 1 x) (<= x 5))))
```

(b) Escreva os contratos e objetivos para cada uma das funções acima.

Condicionais e Funções Condicionais

- Formato geral de **EXPRESSÕES CONDICIONAIS**:

```
(cond
 [pergunta resposta]
 ...
 [pergunta resposta])
```

ou

```
(cond
 [pergunta resposta]
 ...
 [else resposta])
```

Expressões Condicionais e Funções

- Exemplo:

```
(cond
  [(< n 10) 5.0]
  [(< n 20) 5]
  [(< n 30) true])
```

- Exemplo de expressão condicional **mal formada** (porquê???):

```
(cond
  [(< n 10) 30 12]
  [(> n 25) false]
  [(> n 20) 0])
```

Expressões Condicionais e Funções

- REPETINDO: Template de **EXPRESSÕES CONDICIONAIS**:

```
(cond
 [pergunta resposta]
 ...
 [pergunta resposta])
```

ou

```
(cond
 [pergunta resposta]
 ...
 [else resposta])
```

- Ou seja cada linha da expressão *cond* só pode conter DUAS expressões (uma pergunta e uma resposta)

Expressões Condicionais e Funções

- Racket determina o **valor de cada condição**
- Uma condição avalia para `true` ou `false`
- Para a **primeira que avaliar para `true`**, Racket avalia a *resposta* correspondente
- O valor desta resposta é o **valor final** da expressão condicional
- Se a última condição é um `else` e todas as demais falham, a **última resposta** é o valor da expressão condicional

Expressões Condicionais e Funções

- Dois exemplos:

```
(cond                          (cond
 [(<= n 1000)  .040]            [(<= n 1000)  .040]
 [(<= n 5000)  .045]            [(<= n 5000)  .045]
 [(<= n 10000)  .055]           [(<= n 10000)  .055]
 [(> n 10000)  .060])           [else .060])
```

- Avalie as expressões para $n = 10000$ e $n = 20000$.

Expressões Condicionais e Funções

- Qual das duas expressões abaixo é **legal**?

```
(cond
  [(< n 10) 20]
  [(> n 20) 0]
  [else 1])
```

```
(cond
  [(< n 10) 20]
  [(and (> n 20) (<= n 30))]
  [else 1])
```

Expressões Condicionais e Funções

- Porquê a seguinte expressão condicional é **ilegal**?

```
(cond
  [(< n 10) 20]
  [* 10 n]
  [else 555])
```

"Suponha que um banco pague juros de 4% para depósitos de até R$ 1000 (inclusive), 4.5% para depósitos de até R$ 5000 (inclusive) e de 5% para depósitos de mais de R$ 5000. Escreva um programa que, dado o valor a ser depositado, produza a taxa de juros correspondente a esse valor."

Expressões Condicionais e Funções

- Contrato, objetivo e cabeçalho

```
; taxa-de-juros : Número  ->  Número
; Determina a taxa de juros, dada uma
; quantia numérica

(define (taxa-de-juros quantia)
...)
```

- Exemplos (relevantes) de uso:

```
(= (taxa-de-juros 1000) .040)

(= (taxa-de-juros 5000) .045)

(= (taxa-de-juros 8000) .050)
```

Expressões Condicionais e Funções

- O corpo da função deve ser uma **expressão condicional** que distingue os **três casos** mencionados no enunciado do problema.

```
(cond
  [(<= quantia 1000) ...]
  [(<= quantia 5000) ...]
  [(> quantia 5000) ...])
```

Expressões Condicionais e Funções

- Usando os exemplos e o *rascunho* da expressão condicional, a resposta é fácil:

```
(define (taxa-de-juros quantia)
  (cond
    [(<= quantia 1000) 0.040]
    [(<= quantia 5000) 0.045]
    [(> quantia 5000) 0.050]))
```

Expressões Condicionais e Funções

- Como sabemos que a função só precisa de três casos, podemos trocar a última condição por um `else`:

```
(define (taxa-de-juros quantia)
  (cond
    [(<= quantia 1000) 0.040]
    [(<= quantia 5000) 0.045]
    [else 0.050]))
```

Expressões Condicionais e Funções

- Quando aplicamos *taxa-de-juros* para uma quantia (4000, por exemplo), o cálculo segue como esperado: Racket primeiro procura uma definição da função em questão, copia o corpo desta função e, depois, instancia a variável *quantia* que toma o valor numérico de 4000:

```
   (taxa-de-juros 4000)
=  (cond
     [(<= 4000 1000) 0.040]
     [(<= 4000 5000) 0.045]
     [else 0.050])
= 0.045
```

Resultado Final do Processo

```
;; taxa-de-juros : Número  ->  Número
;; Determina a taxa de juros, dada uma
;; quantia numérica
;; Exemplos de uso:
;;    (taxa-de-juros 1000) retorna .040
;;    (taxa-de-juros 5000) retorna .045
;;    (taxa-de-juros 8000) retorna .050

(define (taxa-de-juros quantia)
 (cond
  [(<= quantia 1000) 0.040]
  [(<= quantia 5000) 0.045]
  [else 0.050]))
```

Projetando Funções Condicionais
Etapas

- **Fase 1**: Análise de dados
  - **Objetivo:** Determinar as situações **distintas** com as quais a função deve lidar
  - **Atividade:** Inspecionar o enunciado do problema para identificar situações distintas; listar estas situações

- **Fase 2:** Exemplos
  - **Objetivo:** Fornecer um exemplo **para cada situação**
  - **Atividade:** Escolher pelo menos um exemplo para cada situação para intervalos ou enumerações. Os exemplos devem contemplar casos em **limites**

- **Fase 3:** Corpo da função (1)
  - **Objetivo:** Formular a **expressão condicional**
  - **Atividade:** Escrever o **esqueleto** da expressão condicional, com **uma cláusula por situação**

Expressões Condicionais e Funções

- **Fase 4:** Corpo da função (2)
  - **Objetivo:** Formular as **respostas** para a expressão condicional
  - **Atividade:** Lidar com **cada linha** da expressão condicional em separado e desenvolver a expressão Racket que computa a **resposta apropriada** para cada caso

Lembrete

1. Leia os capítulos 1 a 5 do livro `www.htdp.org`

2. Teste os exemplos do livro no DrRacket

3. Faça os exercícios da lista !