Paradigma de Programação Funcional

2.0 - Fundamentos - Exercícios

2.0.1) [sicp 1.1] A seguir está uma lista de expressões. Qual é o resultado impresso pelo interpretador Racket em resposta a cada expressão? Assuma que as expressões serão avaliadas na ordem em que são apresentadas.

```
10

(+ 5 3 4)

(- 9 1)

(/ 6 2)

(+ (* 2 4) (- 4 6))

(define a 3)

(define b (+ a 1))

(+ a b (* a b))

(= a b)

(if (and (> b a) (< b (* a b)))
    b
    a)

(cond [(= a 4) 6]
      [(= b 4) (+ 6 7 a)]
      [else 25])

(+ 2 (if (> b a) b a))

(* (cond [(> a b) a]
         [(< a b) b]
         [else -1])
   (+ a 1))
```

2.0.2) [sicp 1.2] Traduza a seguinte expressão para a forma pré-fixa

$$\frac{5+4+(2-(3-(6+\frac{4}{5})))}{3(6-2)(2-7)}$$

2.0.3) [tspl 2.2.2] Experimente os procedimentos `+`, `-`, `*` e `/` e determine as regras do Racket para o tipo do valor de retorno para cada procedimento quando são dados diferentes tipos de argumentos numéricos.

2.0.4) [sicp 1.4] O modelo de avaliação visto em sala permite combinações em que os operadores são expressões compostas. Use está observação para descrever o comportamento do seguinte procedimento:

```
(define (a-plus-abs-b a b)
    ((if (> b 0) + -) a b))
```

2.0.5) [sicp 1.5] Ben Bitdiddle inventou um método para determinar se um interpretador está usando avaliação com ordem aplicativa ou avaliação com ordem normal. Ele definiu os seguintes procedimentos:

```
(define (p) (p))

(define (test x y)
  (if (= x 0)
      0
      y))
```

Então avaliou a seguinte expressão

```
(test 0 (p))
```

Qual é o comportamento que Ben irá observar com um interpretador que usa avaliação com ordem aplicativa? Qual é o comportamento que ele irá observar com um interpretador que usa avaliação com ordem normal? Explique a sua resposta.

2.0.6) Defina uma função que encontre o maior valor entre 2 valores dados.

2.0.7) [sicp 1.3] Defina uma função que receba 3 números como parâmetros e retorne a soma dos quadrados dos dois maiores números.

2.0.8) Defina uma função que calcule a distância de um ponto no plano cartesiano (representado por dois números) a origem.

2.0.9) Defina uma função que receba como parâmetro 3 números que representam os lados de um triângulo e classifique o triângulo como equilátero, isósceles ou escaleno. Veja a página sobre triângulos na Wikipédia.

2.0.10) Defina uma função que classifique o grau de obesidade de uma pessoa usando o IMC.

2.0.11) Faça a execução passo a passo das funções definidas nos exercícios anteriores para alguns exemplos. Para isto você deve criar um novo arquivo, mudar a linguagem para "Beginning Student" e copiar o código da função e da chamada de exemplo.

Referências

- [sicp]. Structure and Interpretation of Computer Programs
- [tspl]. The Scheme Programming Languague

Licença