

Paradigmas de Programação – Exercícios (3)

14 de junho de 2010

```
(display (op a b))  
(newline)  
(loop (+ 1 i))))))
```

=>

```
7  
3  
10  
2.5
```

Ex. 1 — Imagine uma função que mude o valor de uma variável para #f:

```
(define faz-falso!  
  (lambda (x)  
    (set! x #f)))
```

Porque ela não funciona? Como conseguir que (faz-falso! variavel) funcione de forma correta?

Ex. 2 — Escreva uma macro que troque os valores de duas variáveis, e diga porque ela não pode ser um procedimento.

Ex. 3 — Faça duas macros:

- (cancelado, porque não falei de define-macro nem de explicitly-renaming macros): Uma que receba várias e expanda uma deals aleatoriamente
- (este sim podem fazer) Ourta (mais útil) que deixa a escolha aleatória para o tempo de execução

Discuta a diferença entre elas.

Ex. 4 — Faça um procedimento faz-ciclo que receba uma lista de procedimentos procs e devolva um procedimento que devolve um dos procedimentos na lista procs (eles devem ser devolvidos em ordem, e quando chegar ao último voltar ao primeiro):

```
(let ((ciclo (faz-ciclo + - * /)))  
  (let ((a 5)  
        (b 2))  
    (let loop ((i 0))  
      (if (< i 4)  
          (let ((op (ciclo)))
```

Ex. 5 — Se você já cursou Linguagens Formais e Automata, diga (não precisa implementar) como poderia implementar um autômato finito determinístico que, em cada estado, chama uma função. (Você pode usar parte da idéia do exercício anterior)

Ex. 6 — Faça uma função que implemente um *closure* sobre as seguintes variáveis:

- Saldo: um valor numérico;
- Itens: uma lista de itens e quantidades. Por exemplo, ((banana 10) (notebooks 2) (saco-batata 35)).

O criador do fecho deve aceitar valores iniciais para estas variáveis.

Este fecho pode representar um agente negociador. As funções que devem ser retornadas, e que usam as variáveis do fecho, são:

- Uma para comprar um item (se já existe na lista, some a quantidade; senão, adicione à lista);
- Uma para vender um item;
- Uma para verificar a lista de itens;
- Uma para verificar o saldo.

Ex. 7 — Faça um programa que instancie dois ou três fechos do exercício anterior, inicialize cada um diferentemente e depois faça cada um comprar ou vender aleatoriamente para o outro. Depois de k rodadas, mostre o estado de cada um. Há o problema de determinar o preço de cada item. use uma tabela global, e faça os agentes comprarem e venderem pelo preço global $\pm\delta$, onde δ é um número aleatório entre -10% e +10% do preço.