Paradigmas de Programação – Exercícios (5)

28 de junho de 2010

Ex. 1 — Mostre como implementar `quote` como uma macro (sem usar `quote` dentro da macro, é claro!)

Ex. 2 — Sem usar escopo dinâmico, crie um procedimento `log-ize` que aceite como parâmetro procedimentos `proc`, `log-got-in`, `log-got-out` e devolva outro procedimento. Este novo procedimento é idêntico ao procedimento original, exceto que chama `log-got-in` quando `proc` é chamado, e `log-got-out` quando `proc` retorna.

Ex. 3 — Como obter o mesmo efeito de `log-ize` usando escopo dinâmico? (Implemente!)

Ex. 4 — Implemente a macro `xor`, que recebe n parâmetros e retorna `#f`, a não ser que *exatamente* um de seus argumentos seja diferente de `#f`:

```
(xor 1 #f #f)  ;; => 1
(xor #t 10 20) ;; => #f
(xor #f #f #f) ;; => #f
```

Ex. 5 — (médio) Mude sua macro `xor` para que ela receba k, o número exato de parâmetros que devem ser não-`#f` para que ela retorne `#t`:

```
(xor 2 'x #f #f)  ;; => #f (k=2)
(xor 1 'x #f #f)  ;; => 'x (k=1)
(xor 1 #t 10 20)  ;; => #f (k=1)
(xor 0 #f #f #f)  ;; => #t (k=0)
```

Ex. 6 — (difícil) Considere o seguinte trecho de código:

```
(with-modulo 10
  (set! resultado
        (+ 1 (* 7 x)
             (* 5 y))))
```

Neste exemplo, `(with-modulo ...)` muda toda a aritmética em seu escopo para módulo 10. O valor de resultado será, então, $(1 + 7x + 5y) \mod 10$.

a) Diga como `with-modulo` pode ser implementado (macro ou função?) e porque.

b) Afora decidir entre macro ou função, há também outra questão: como mudar o significado das operações aritméticas? Grosso modo, isso pode ser feito sintaticamente ou semanticamente; discuta como as duas abordagens funcionariam.

c) Há diferença na prática entre mudar o significado de `+`, `-`, `*` `/` sintaticamente ou semanticamente? Porque? (Leve em consideração o exemplo abaixo)

```
(define afim
  (lambda (a b x)
    (+ (* a x) b)))

(with-modulo 7
  (set! resultado
        (+ 1 (afim var1
                   var2
                   var3))))
```

Ex. 7 — (difícil) Implemente `with-modulo`.