

「プログラミング言語」課題

1029-24-9540 山崎啓太郎

July 3, 2013

1 Ex. 4.27

1.1 考え方

式の値がサunkの時は、それがサunkでないものに達するまで強制を繰り返すため、`w` を定義した際に、外側の `id` が実行され、内側の `id` は実行されない。

そのため、一回目の `count` の評価結果は 1 となる。

`w` を評価すると内側の `id` が実行され 10 が評価値となる。

さきほどの `w` の評価により、二度目の `count` の評価結果は 2 となる。

1.2 実行例

```
;;; L-Eval input:
count
;;; L-Eval value:
1
```

```
;;; L-Eval input:
w
;;; L-Eval value:
10
```

```
;;; L-Eval input:
count
;;; L-Eval value:
2
```

2 Ex. 4.33

2.1 考え方

l-eval.scm を編集して、text-of-quotation 関数を呼び出す際に env を引数として渡すようにしたほか、text-of-quotation 関数を以下のように変更した。

```
(define (text-of-quotation exp env)
  (if (pair? (cadr exp))
      (eval (make-quotation (cadr exp)) env)
      (cadr exp)))

(define (make-quotation body)
  (if (null? body)
      '()
      (list 'cons (list 'quote (car body)) (make-quotation (cdr body))))))
```

これにより、クォートしたリストも遅延リストを生じるようになった。

2.2 実行例

```
;;; L-Eval input:
(define (cons x y) (lambda (m) (m x y)))
;;; L-Eval value:
ok

;;; L-Eval input:
(define (car z) (z (lambda (p q) p)))
;;; L-Eval value:
ok

;;; L-Eval input:
(define (cdr z) (z (lambda (p q) q)))
;;; L-Eval value:
ok

;;; L-Eval input:
(define (length l) (if (null? l) 0 (+ 1 (length (cdr l)))))
;;; L-Eval value:
ok

;;; L-Eval input:
(length (cons (/ 1 0) (cons 3 '())))
```

```
;;; L-Eval value:
```

```
2
```

```
;;; L-Eval input:
```

```
(car (cons 3 (/1 0)))
```

```
;;; L-Eval value:
```

```
3
```