

「プログラミング言語」課題

1029-24-9540 山崎啓太郎

June 26, 2013

1 Ex. 4.6

1.1 考え方

(let ((<var1> <exp1>) (<var2> <exp2>)) <body>) と
((lambda (<var1> <var2>) body) <exp1> <exp2>) は等価であるので、let
構文を lambda 式を使った構文に変換する。

1.2 実行例

```
;;; M-Eval input:  
(let ((hoge 12)) hoge)
```

```
;;; M-Eval value:  
12
```

```
;;; M-Eval input:  
(let ((fuga 1) (hoga 2)) hoga)
```

```
;;; M-Eval value:  
2
```

2 Ex. 4.16

2.1 考え方

a.lookup-variable-value で value が*unassigned*であればエラーを返すように
する。

b. 内部の define 文を探し、let 構文で置き換えて返す。

内部の define を全て探して、変数名と値を取得し、一つの let 構文で変数名を定義し、*unassigned*で初期化した後に、set!を使用して変数に値を定義する。

c.scan-out-defines は等価なものを返すものではあるが、今回は eval 時に渡される exp 内で使用すべきである。

その点を考慮すると、make-procedure は eval のみから呼び出される関数であるが、procedure-body は user-print から呼び出されているため、make-procedure のほうが適切であると思われる。

2.2 実行例

```
;;; M-Eval input:  
((lambda ()  
  (define a 1)  
  a))
```

```
;;; M-Eval value:  
1
```