

# 「プログラミング言語」課題

1029-24-9540 山崎啓太郎

June 19, 2013

## 1 Ex. 4.4

### 1.1 考え方

and は左から右に評価され、false のものがあれば false、なければ true が返される。

or は左から右に評価され、true のものがあれば true、なければ false が返される。

そのため、and ・ or の評価は iterator を使えばよい。

### 1.2 実行例

```
;;; M-Eval input:  
(and 1 2 false)
```

```
;;; M-Eval value:  
#f
```

```
;;; M-Eval input:  
(and 1 false 2)
```

```
;;; M-Eval value:  
#f
```

```
;;; M-Eval input:  
(and 1 2 3 4 5)
```

```
;;; M-Eval value:  
#t
```

```
;;; M-Eval input:  
(or false false false)
```

```
;;; M-Eval value:  
#f
```

```
;;; M-Eval input:  
(or false false 1)
```

```
;;; M-Eval value:  
#t
```

```
;;; M-Eval input:  
(or 1 false false)
```

```
;;; M-Eval value:  
#t
```

```
;;; M-Eval input:  
(or)
```

```
;;; M-Eval value:  
#f
```

```
;;; M-Eval input:  
(and)
```

```
;;; M-Eval value:  
#t
```

## 2 Ex. 4.11

### 2.1 考え方

frame を list から key-value のペアにした。

それに伴い、set-variable-value!等の env を渡す関数も変更した。

'((key1 key2 key3) (1 2 3)) となっていた frame を'((key1 1) (key2 2) (key3 3)) という形に直せば良い。

## 2.2 実行例

test/4.11.scm

```
(use gauche.test)
(add-load-path ".")

(test-start "4.11")

(load "4.11")

(define frame (make-frame '(key0) '(0)))
(define env (list frame))

(test-section "frame test")
(test* "(frame-variables frame)" (frame-variables frame) '(key0))
(test* "(frame-values frame)" (frame-values frame) '(0))
(add-binding-to-frame! 'key1 1 frame)
(add-binding-to-frame! 'key2 2 frame)
(test* "(frame-variables frame)" (frame-variables frame) '(key0 key1 key2))
(test* "(frame-values frame)" (frame-values frame) '(0 1 2))

(test-section "env test")
(test* "(lookup-variable-value 'key0 env)" (lookup-variable-value 'key0 env) 0)
(test* "(lookup-variable-value 'key1 env)" (lookup-variable-value 'key1 env) 1)
(set-variable-value! 'key2 3 env)
(test* "(lookup-variable-value 'key2 env)"
(lookup-variable-value 'key2 env) 3)
(define-variable! 'key3 4 env)
(test* "(lookup-variable-value 'key3 env)"
(lookup-variable-value 'key3 env) 4)
```

### 結果

Testing 4.11 ...

```
<frame test>-----
test (frame-variables frame), expects (key0) ==> ok
test (frame-values frame), expects (0) ==> ok
test (frame-variables frame), expects (key0 key1 key2) ==> ok
test (frame-values frame), expects (0 1 2) ==> ok
<env test>-----
test (lookup-variable-value 'key0 env), expects 0 ==> ok
test (lookup-variable-value 'key1 env), expects 1 ==> ok
```

```
test (lookup-variable-value 'key2 env), expects 3 ==> ok
test (lookup-variable-value 'key3 env), expects 4 ==> ok
```