

工学部専門科目「プログラミング言語」2013年度 期末試験

実施日時: 2013年7月31日 8:45~10:15, 担当: 五十嵐(淳)

問1(解答用紙1枚目表に解答) メタ・サーキュラー・インタプリタにおいて, 環境のフレームを, 変数とその値のペアのリストで表現することを考える. すなわち, 変数のリスト `variables`, 値のリスト `values` からフレームを作る手続きは

```
(define (make-frame variables values)
  (if (null? variables) '()
      (cons (cons (car variables) (car values))
            (make-frame (cdr variables) (cdr values)))))
```

であるとする.

この時, 環境 `env` に束縛された変数 `var` の値を `val` に更新する手続き (`set-variable-value! var val env`) を, 下に示す定義の ~ を埋めて完成させよ. そして, その動作を説明せよ. ただし, 与えられた環境 `env` から最初のフレームを除いた環境を取り出す手続き (`enclosing-environment env`) を使ってよい.

```
(define (set-variable-value! var val env)
  (define (env-loop env)
    (define (scan frame)
      (cond ((null? frame)
             )
            ((eq? var (caar frame))
             )
            (else ))
      (if (eq? env the-empty-environment)
          (error "Unbound variable -- SET!" var)
          (let ((frame (first-frame env)))
              (scan frame))))
    (env-loop env))
```

問2(解答用紙1枚目裏に解答) メタ・サーキュラー・インタプリタにおける内部定義の実現方式について, 方式1, 2を比較せよ. 具体的には, ふたつの方式が異なる挙動を示す具体的な例を示し, 方式1の問題点を指摘せよ.

| | | | | |
|-----|---|---|--|--------|
| | (lambda (<vars>) | | (lambda (<vars>) | |
| | (define x ₁ <e ₁ >) | | (let ((x ₁ <e ₁ >) | |
| 方式1 | : | を | : | に変換する. |
| | (define x _n <e _n >) | | (x _n <e _n >)) | |
| | <e ₀ >) | | <e ₀ >)) | |

| | | | | | |
|------|---|---|---|---|---------|
| | | (lambda <vars> | | (let ((x ₁ '*unassigned*) | |
| | | | | ⋮ | |
| | (lambda <vars> | (define x ₁ <e ₁ >) | | (x _n '*unassigned*)) | |
| 方式 2 | ⋮ | | を | (set! x ₁ <e ₁ >) | に変換する . |
| | (define x _n <e _n >) | | | ⋮ | |
| | <e ₀ >) | | | (set! x _n <e _n >) | |
| | | | | <e ₀ >)) | |

問 3(解答用紙 2 枚目表に解答) catch/throw 機構を持つ拡張 Scheme を考える . 以下のふたつの手続き prod-list, prod-list-catch はいずれも与えられた整数リスト lst の要素の積を求めるものである . 両者の動作を説明し , 実行される乗算の回数について比較せよ .

```
(define (prod-list lst)
  (if (null? lst)
      1
      (if (= 0 (car lst)) 0
          (* (car lst) (prod-list (cdr lst))))))
```

```
(define (prod-list-catch lst)
  (define (prod-list-aux lst)
    (if (null? lst)
        1
        (if (= 0 (car lst)) (throw 'zero 0)
            (* (car lst)
                (prod-list-aux (cdr lst))))))
  (catch 'zero (prod-list-aux lst)))
```

問 4(解答用紙 2 枚目裏に解答)

- 与えられたふたつの整数 n, m について $n \leq k \leq m$ なる k を非決定的に返す手続き (an-integer-between n m) を特殊形式 amb を使って書け .
- 友愛数とは , 異なる 2 つの自然数の組 (i, j) で ,

i の (i を除く) 約数の和 = j

j の (j を除く) 約数の和 = i

をみたすものをいう . 例えば (220, 284) は 220 の約数が 1, 2, 4, 5, 10, 11, 20, 22, 44, 55, 110, 220 で (220 を除く) 和が 284 であり , 284 の約数が 1, 2, 4, 71, 142, 284 で (284 を除く) 和が 220 なので友愛数である .

与えられたふたつの自然数 n, m に対し , $n \leq i \leq m$ かつ $n \leq j \leq m$ を満たす友愛数 (i, j) を非決定的に返す手続き (amicable-numbers n m) を書け .