工学部専門科目「プログラミング言語」2014 年度 期末試験

実施日時: 2014 年 7 月 22 日 10:30〜12:00，担当: 五十嵐 (淳)

問 1(解答用紙 1 枚目表に解答) 以下のプログラムについて，各行の実行後に変数 x, y が指す構造を，ペアを表す箱とポインタを表す矢印を使って示すとともに，プログラムの挙動を説明せよ．

Illustrate the structures to which variables x and y point (if any) after executing each line of the following program, by using boxes representing pairs and arrows representing pointers and explain the behavior of the program.

```
(define x (list 3 5))
(set-cdr! (cdr x) x)
(define y (cons 4 (cdr x)))
(set-cdr! x y)
```

問 2(解答用紙 1 枚目裏に解答)

(1) let 式 (let ((\(x_1\) \(e_1\)) ⋯ (\(x_n\) \(e_n\)) \(e_0\)) がどのように評価されるかを環境を使って説明せよ．ただし，\(x_i\) は変数，\(e_i\) は式である．

Explain how a let expression of the form (let ((\(x_1\) \(e_1\)) ⋯ (\(x_n\) \(e_n\)) \(e_0\)) is evaluated, by using the evaluation model based on environments. Here, \(x_i\) stands for a variable and \(e_i\) for an expression.

(2) 以下の定義を入力・実行した後，環境がどのようになっているか図を描き説明せよ．

Illustrate and explain the environment after processing the following definition.

```
(define f
  (let ((x 0))
    (lambda (y) (set! x (+ x y)) x)))
```

(3) (2) の環境のもとで (f 4) がどのように評価されるか説明せよ．

Explain how (f 4) is evaluated under the environment in (2).

問 3(解答用紙 2 枚目表に解答) ストリームを使って，二乗和が平方数であるような非負整数の組 \((i, j)\) (ただし \(i \leq j\)) を列挙することを考える．

Consider enumerating as a stream of all the pairs of non-negative integers \((i, j)\) such that \(i \leq j\) and the sum of squares of \(i\) and \(j\) is a square number.

1. s を \(i \leq j\) なる非負整数のペアを列挙したストリームとする．s を使って題意のストリームを定義せよ．

   Let s be a stream of all the pairs of non-negative integers \((i, j)\) such that \(i \leq j\). Give the definition of the stream specified as above by using s.

2. 以下の s の定義の誤りを説明せよ．

   What's wrong with the following definition of s?

```
(define (integer-from n) (cons-stream n (integer-from (+ n 1))))

(define (stream-append s1 s2)
  (if (stream-null? s1) s2
      (cons-stream (stream-car s1) (stream-append (stream-cdr s1) s2))))

(define (pairs n)
  (cons-stream
    (cons n n)
    (stream-append
      (stream-map (lambda (x) (cons n x)) (integer-from (+ n 1)))
      (pairs (+ n 1)))))

(define s (pairs 0))
```

3. 正しい s の定義を与え，説明せよ．

   Give a correct definition of s and explain it.

問 4(解答用紙 2 枚目裏に解答) let* 式は let と同様に変数の局所定義をするための構文だが，let と違い左から順番に変数束縛を行う．例えば，

e = `(let* ((x 3) (y (+ x 4))) (* x y))`

の値は 21 となる．(y の値が先行する x の値を使って計算されることに注意せよ．)

この let* を derived expression としてメタサーキュラ評価器に実装することを考える．

Let* is a syntactic form for variable binding, similarly to let. Unlike let, however, let* binds variables from left to right. For example, the expression e above evaluates to 21. (Note that the value of y depends on the value of the preceding definition of x.)

Let's implement let* for a meta-circular evaluator as a derived expression.

(1) let* 式 (let* ((x_1 e_1) $\cdots$ (x_n e_n)) e_0) (ただし，x_i は変数，e_i は式である) と等価な式を，入れ子になった let 式を使って表せ．

Show a let*-free expression equivalent to a let* expression of the form (let* ((x_1 e_1) $\cdots$ (x_n e_n)) e_0), by using nested let.

(2) let* 式と等価な式を let* も let も使わずに表せ．

Show an expression equivalent to a let* expression of the same form above without even using let.

(3) (2) の変形を行う手続 (expand-let* exp) を定義せよ．入力 exp が let* 式 (の引用) であることは仮定してよい．

Give the definition of procedure (expand-let* exp) to perform the transformation of (2). You can assume exp is a (quoted) let* expression.