

# プログラミング言語論 2019年度 第2回小テスト解答例

問1 以下のプログラム断片の型の整合性を、手順(1)-(2)に従って示せ。ただし、これは講義中に提示した、Cのサブセットによるプログラムの断片である。

```
int *p;
int x[3];
p = x;
```

(1) 変数宣言 `int *p; int x[3];` を、講義で説明した型の後置記法による宣言に直せ。

(解答)

```
p : int *
x : int [3]
```

(2) (1) で得られた、`p, x` の型の後置記法による宣言から、講義中に提示した規則に従って、代入式 `p=x` が型に関して整合性を持つことを示せ。

(解答)

$$\frac{\frac{p : \text{int } * \quad x : \text{int } [3]}{p : \text{int } * \quad x : \text{int } \&}}{p = x : \text{int } \&}$$

問2 ラムダ式  $(\lambda x. \lambda y. x) ((\lambda z. z) w)$  は何度か  $\beta$  変換を行うことによってラムダ式  $(\lambda y. w)$  に変換できるが、その変換過程を示せ。(変換過程は複数あるが、そのうちの一つでよい。)

(解答 1)

$$(\lambda x. \lambda y. x) ((\lambda z. z) w) \xrightarrow{\beta} (\lambda x. \lambda y. x) w \xrightarrow{\beta} \lambda y. w$$

(解答 2)

$$(\lambda x. \lambda y. x) ((\lambda z. z) w) \xrightarrow{\beta} \lambda y. ((\lambda z. z) w) \xrightarrow{\beta} \lambda y. w$$

問3 以下の C++ 言語で書かれたプログラムを実行したときの画面への出力結果を示せ。

```
#include <stdio.h>
class Shape {
public:
    virtual void draw (void) {
        printf ("Shape\n");
    }
};
class Box : public Shape {
    void draw (void) {
        printf ("Box\n");
    }
};
```

```
int main (void) {
    Shape *s;
    s = new Box ();
    s->draw();
    return 0;
}
```

(解答)

Box

問4 以下の(1), (2)のプログラムの意味(状態の変化)を、講義中に提示した規則にしたがって示せ。ただし、これらのプログラムは、講義中で意味の定義を紹介するときに定義した、Cの非常に小さなサブセットによるプログラムである。(1)、(2)のプログラムの実行前の状態は、いずれも  $\sigma = \{(X, 3), (Y, 1), (Z, 0)\}$  とする。

(1)  $Z=(X+4);$

(解答)

$$\frac{\frac{\langle X, \sigma \rangle \rightarrow 3 \quad \langle 4, \sigma \rangle \rightarrow 4}{\langle (X+4), \sigma \rangle \rightarrow 7}}{\langle Z = (X+4);, \sigma \rangle \rightarrow \sigma[7/Z]}$$

上記より、状態  $\sigma$  においてプログラム  $Z=(X+4);$  を実行すると、状態は

$$\sigma[7/Z] = \{(X, 3), (Y, 1), (Z, 7)\}$$

になる。

(2)  $\text{while}(Y)\{Y=(Y-1);\}$

(解答)

$$\frac{\frac{\frac{\langle Y, \sigma \rangle \rightarrow 1 \quad \langle 1, \sigma \rangle \rightarrow 1}{\langle (Y-1), \sigma \rangle \rightarrow 0}}{\langle Y = (Y-1);, \sigma \rangle \rightarrow \sigma[0/Y]} \quad \frac{\langle Y, \sigma[0/Y] \rangle \rightarrow 0}{\langle \text{while}(Y)\{Y = (Y-1);\}, \sigma[0/Y] \rangle \rightarrow \sigma[0/Y]}}{\langle \text{while}(Y)\{Y = (Y-1);\}, \sigma \rangle \rightarrow \sigma[0/Y]}$$

上記より、状態  $\sigma$  においてプログラム  $\text{while}(Y)\{Y=(Y-1);\}$  を実行すると、状態は

$$\sigma[0/Y] = \{(X, 3), (Y, 0), (Z, 0)\}$$

になる。

問5 以下の Pascal プログラムを実行したときの出力結果を示せ。手続きの仮引数に var がついている場合、call by reference であることを表す。writeln は引数の値を出力後改行する。

```
program test;
var x : integer;
var y : integer;
procedure swap
  (var x: integer;
   var y : integer);
var z : integer;
begin
  z := x; x := y; y := z
end;
begin
  x := 3;
  y := 4;
  swap (x,y);
  writeln (x);
  writeln (y)
end.
```

(解答)

4

3

問6 以下の Pascal プログラムを実行したときの出力結果を示せ。Pascal における変数の有効範囲 (scope) の定め方は static scope であることに注意せよ。

```
program P;          procedure D;          begin
var n : char;      var n : char;          n := 'L';
procedure W;      begin          W;
begin              n := 'D';          D
  writeln(n)      W          end.
end;              end;
```

(解答)

L

L