Foundations for Programming Languages
Answers for small examination 2

Problem 1 Show the type consistency of the following program fragment, which is written in the subset of C language presented in the lecture, according to (1) and (2).

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

(1) Rewrite the variable declarations `int *p;` and `int x[3];` in the postfix notation presented in the lecture.

(Answer)

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

(2) Show the type consistency of the assignment expression `p=x` by applying the inference rules to the declarations of `p` and `x` in the postfix notation obtained in (1).

(Answer)

$$\frac{\texttt{p} \;:\; \texttt{int} \,* \qquad \dfrac{\texttt{x} \;:\; \texttt{int}\;[3]}{\texttt{x} \;:\; \texttt{int}\;\&}}{\texttt{p} = \texttt{x} \;:\; \texttt{int}\;\&}$$

Problem 2 A lambda expression $(\lambda x.\ \lambda y.\ x)\ ((\lambda z.\ z)\ w)$ can be transformed to $(\lambda y.\ w)$ by applying the β reductions. Write the each step of the β reductions. (Although there are more than one sequences of β reductions, write one of them.)

(Answer 1)

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

(Answer 2)

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

Problem 3 Write the output to the display when executing the following program in

C++.

```
#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;
}
```

(Answer)

```
Box
```

Problem 4

Show the output produced by executing the following Pascal program. When the keyword `var` is attached to a formal parameter, it designates the parameter as call-by-reference. The procedure `writeln` writes out to the standard output the value of the parameter and a new line character.

```
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.
```

(Answer)

```
4
3
```

Problem 5

Show the output produced by executing the following Pascal program. Note that Pascal is statically (lexically) scoped.

```
program P;
var n : char;
procedure W;
begin
   writeln(n)
end;
```

```
procedure D;
var n : char;
begin
   n := 'D';
   W
end;
```

```
begin
   n := 'L';
   W;
   D
end.
```

(Answer)

```
L
L
```

Problem 6

Show the meaning of the following programs (1) and (2) by using the rules presented in the lecture. Note that the programs are in the small subset of C presented in the lecture. Let the states before executing the programs both to be $\sigma = \{(\mathtt{X}, 3), (\mathtt{Y}, 1), (\mathtt{Z}, 0)\}$.

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

$$\cfrac{\cfrac{<\mathtt{X}, \sigma> \to 3 \quad <4, \sigma> \to 4}{<(\mathtt{X}+4), \sigma> \to 7}}{<\mathtt{Z} = (\mathtt{X}+4);, \sigma> \to \sigma[7/Z]}$$

So in the state σ, after executing the program `Z=(X+4);` the state becomes as follows.

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

(2) `while(Y){Y=(Y-1);}`

$$\cfrac{<\mathtt{Y}, \sigma> \to 1 \quad \cfrac{\cfrac{<\mathtt{Y}, \sigma> \to 1 \quad <1, \sigma> \to 1}{<(\mathtt{Y}-1), \sigma> \to 0}}{<\mathtt{Y} = (\mathtt{Y}-1);, \sigma> \to \sigma[0/\mathtt{Y}]} \quad \cfrac{<\mathtt{Y}, \sigma[0/\mathtt{Y}]> \to 0}{<\mathtt{while(Y)}\{\mathtt{Y} = (\mathtt{Y}-1);\}, \sigma[0/\mathtt{Y}]> \to \sigma[0/\mathtt{Y}]}}{<\mathtt{while(Y)}\{\mathtt{Y} = (\mathtt{Y}-1);\}, \sigma> \to \sigma[0/\mathtt{Y}]}$$

So in the state σ, after executing the program `while(Y){Y=(Y-1);}` the state becomes as follows.

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