SE 2A04 Fall 2001

Midterm Test 1 Answer Key

100 pts.

Instructor: W. M. Farmer
Revised: 4 October 2001

(1) [5 pts.] The Oberon type `REAL` denotes the set of real numbers. Is this statement true or false?

(a) True.

(b) $\boxed{\text{False.}}$ `REAL` only denotes a finite set of rational numbers.

(2) [5 pts.] The method presented in class for showing that a loop terminates can also be used to show that a recursive procedure terminates. Is this statement true or false?

(a) $\boxed{\text{True.}}$

(b) False.

(3) [5 pts.] In an Oberon constant declaration of the form

`CONST` $x = E$;

E can be any Oberon expression. Is this statement true or false?

(a) True.

(b) $\boxed{\text{False.}}$ E must be an expression whose value can be computed at compile time.

(4) [5 pts.] Suppose that P is a guarded program list of the form

$$(G_1 \rightarrow P_1 | \cdots | G_n \rightarrow P_n).$$

If for all i with $1 \leq i \leq n$ the program P_i satisfies the specification S when G_i is true, then P satisfies S. Is this statement true or false?

(a) True.

(b) $\boxed{\text{False.}}$ It is also necessary to check that $G_1, \ldots, G_n$ cover all cases.

(5) [10 pts.] The Oberon declarative

```
VAR ajax: ARRAY 4 OF INTEGER;
```

constructs an integer array named `ajax`. What are the Oberon *selectors* and *mutators* for this array?

Answer:

Selectors: `ajax[0]`, `ajax[1]`, `ajax[2]`, `ajax[3]`.
Mutators: `ajax[0] :=` , `ajax[1] :=` , `ajax[2] :=` , `ajax[3] :=` .

(6) [15 pts.] According to the specification given in Lab 1, what value should

```
Rectangle.overlap(0,1,1,0,0,2,2,0)
```

return? Explain your reasoning and what assumptions you are making.

Answer: There are three reasonable output values:

1: All the points of A are within B.
4: A and B overlap.
6: A and B are tangent.

Hence there are several possible answers depending on the assumptions that are made.

(7) [15 pts.] Let `w` be a variable of type `ARRAY 100 OF INTEGER` (indexed from 0 to 99) and `a` and `b` be variables of type `INTEGER`. Consider the following loop written in the planning language presented in class:

```
a := 0;
b := 0;
it
  (b < 0 -> stop |
   0 <= b and b < 100 -> a := a + w[b]; b := b + 1; go |
   100 <= b -> stop))
ti
```

Does the loop terminate? If so, what is the strictly decreasing natural number value for the loop?

Answer: Yes, the loop terminates because $100 - b$ is a natural number value that strictly decreases after each iteration of the loop.

(8) [20 pts.] Consider the following piece of code written in the planning language. `x` and `y` are variables of type `INTEGER`.

```
(x < -3 -> x := 17 |
 x = -3 -> x := x + y |
 x > -3 -> y := x)
```

Write an Oberon statement equivalent to this piece of code.

Answer:

```
IF x < -3 THEN
  x := 17;
ELSIF x = -3 THEN
  x := x + y;
ELSE (* x > -3 *)
  y := x;
END;
```

(9) [20 pts.] Consider the following Oberon procedure shown in class:

```
PROCEDURE FactTailRecAux(x: LONGINT; accum: LONGINT): LONGINT;
  BEGIN
    IF x < 0 THEN
      Out.String("Input must be nonnegative (ignore output).");
      RETURN -1;
    ELSIF x = 0 THEN
      RETURN accum;
    ELSE
      RETURN FactTailRecAux(x - 1, accum * x);
    END;
  END FactTailRecAux;
```

Rewrite the body of the procedure (i.e., the part starting with `BEGIN` and ending with `END;`) in the planning language using the **it** P **ti** constructor so that the new version of the procedure computes the same function as the old version.

Answer 1:

```
it
  ((x < 0 -> Out.String("Input must be nonnegative (ignore output).");
             RETURN -1) |
   (x = 0 -> RETURN accum) |
   (x > 0 -> accum := accum * x; x := x - 1; go))
ti
```

Answer 2:

```
it
  ((x < 0 -> Out.String("Input must be nonnegative (ignore output).");
             accum := -1; stop) |
   (x = 0 -> stop) |
   (x > 0 -> accum := accum * x; x := x - 1; go))
ti
RETURN accum;
```