Testing and Verification (DIT085)
Solutions to Model Examination - March 2015

Important Notes. It is not allowed to use study material, computers, and calculators during the examination. The examination comprises 5 question in 2 pages. Please check beforehand whether your copy is properly printed. In order to obtain a VG you need to obtain 80/100, for a G you need to obtain 60/100. Give complete explanation and do not confine yourself to giving the final answer. The answers may be given in Dutch or English. **Good luck!**

Exercise 1 (20 points) Define the following concepts:

1. Fault, Error, Failure

2. Robust Equivalence-Class Testing

3. Regression Testing

4. All-Uses Coverage Criterion

Exercise 2 (25 points) Consider the following program.

```
 1: read(x);
 2: read(y);
 3: if x < 10 then
 4:     x := 10;
 5: end if
 6: while y < x then
 7:     y := y + 1;
 8: end while
 9: x := x + 1
10: write(x);
```

1. Draw the control-flow graph of the program (5 pts),

2. Calculate all prime paths of the CFG (10 pts),

3. Define a set with the fewest number of test cases that satisfies the all-prime-path coverage criterion (10 pts).

Exercise 3 (20 points) Explain the meaning of the following formulas in English.

1. `E <> deadlock` (5 pts),

2. `A <> (a.lora.lp)andv <= 2` (5 pts),

3. `a.l −− >a.lp` (10 pts).

Exercise 4 (25 points) Calculate $Slice(9, \{x\})$ for the following program. The final solution is not sufficient; elaborate on the steps towards the final solution. (20 pts)

```
1: read(x);
2: read(y);
3: z := y;
4: while y < 10 then
5:      z := z + 1;
6: if z < y then
7:      x := 10;
8: x := x + 1
9: write(x);
```

Is the calculated slice optimal? Motivate your answer. (5 pts)

Exercise 5 (10 points) Consider the following procedure, which is supposed to take an array of integers and its size and write the average of the numbers in the array (up to size) on the screen. It turns out that it outputs '8.0' when input $arr= [2, 2, 2, 14, 8]$, $size=5$ is given (which is incorrect). Simplify the test-case using simplification. (Assume that you can manually check whether the outcome of each test is really correct or not.)

```
 1: const MAX = 100;
 2: procedure writeAvg(arr: array of integer; size: integer);
 3: var i : integer;
 4:     avg : real;
 5: begin
 6:     avg := 0;
 7:     for i := 0 to size - 1 do
 8:         avg := avg + arr[size - 1];
 9:     write(avg/size);
10: end;
```