Testing and Verification (DIT085)
Final Examination - March 16, 2016

Important Notes. It is not allowed to use study material, computers, and calculators during the examination. The examination comprises 5 questions 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 Swedish or English. The solutions to the exercises will be available after the examination through the course page. **Good luck!**

Reponsible teacher: Mohammad Mousavi, Phone number: 072 977 35 83.

Exercise 1 (20 points) Define the following concepts:

1. static analysis,
2. nearest inverse denominator,
3. Boehm's curve, and
4. Bezier testing levels (please mention the general idea and name and explain all levels).

Exercise 2 (10 points) Assume that we would like to test a method with 2 input parameters x and y using the weak robust equivalence class testing method. Moreover, assume that the domain of variable x is partitioned into 2 equivalence classes and the domain of y is partitioned into 3 equivalence classes. What is the minimum number of test cases required in the test suite?

Exercise 3 (40 points) Consider the following program.

```
1: Input(x);
2: Input(y);
3: Input(z);
4: while z < y then
5:     y := x;
6:     x := x + 1;
7:     z := x;
8: end while
9: if y <= 20 then
10:    y := 2;
11: end if
12: x := 2 * y;
13: Output(x);
```

1. Draw the control-flow graph of the program. (5 pts)
2. Calculate its cyclomatic number. (5 pts)
3. Calculate all prime paths of the control-flow graph. (10 pts)

4. Calculate $Slice(13, \{x\})$. The final solution is not sufficient; you need to elaborate on the steps towards the final solution (include the relevant variables and the intermediate steps towards the final slice). (20 pts)

Exercise 4 (20 points) Specify the following properties in Timed Computational Tree Logic:

- There is no deadlocking state in any execution. (5 points)
- In all executions, if automata a is in state *sending*, it (i.e., automata a) will eventually be in state *sent*. (7.5 points)
- All executions will eventually arrive in a state where variable x is greater than 10. (7.5 points)

Exercise 5 (10 points) Regarding Visual GUI testing, explain the benefits of VGT (versus GUI model testing and manual GUI testing).