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 4 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 Swedish or English. The solutions to the exercises will be available after the examination through the course page. **Good luck!**

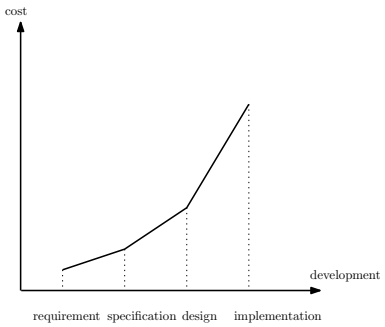
Responsible teacher: Mohammad Mousavi, Phone number: 072 977 35 83.

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).

Solution.

1. Static analysis is a technqiue that builds mathematical abstractions from programs in order to verify certain (fixed) properties without executing the program.
2. Nearest inverse denominator of a node n is the nearest node in the graph that appears in every path from n to termination.
3. Boehm's curve depicted below shows how the cost of fixing bugs increases with an order of magnitude as the detection of faults and fixing them is delayed from one development phase to the next:



4. Bezier testing levels:

   L0 debugging (ad hoc, few input/outputs)

   L1 showing that software works (validating some behavior)

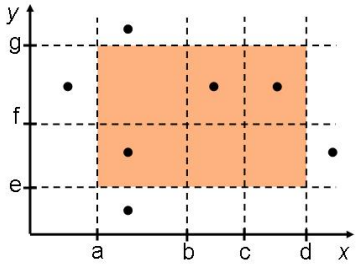
L2 showing that software does not work (scrutinizing corner cases)

L3 reducing risks (organizing and prioritizing test goals)

L4 mental discipline for quality (central to development)

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 classed and the domain of y is partitioned into 3 equivalence classes. What is the minimum number of test cases required in this case?

Solution. Consider the following picture illustrating the weak normal equivalence class based testing technique:



According to this picture, the minimum number of required test cases is 7.

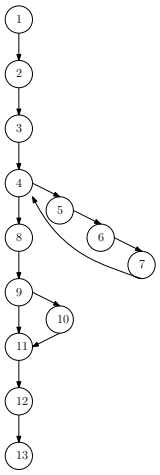
Exercise 3 (40 points) Consider the following program.

```
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)

Solution.

1. The control flow of the program is depicted below.



2. The cyclomatic number of a CFG is the number of regions created by it in the plane, which is 3 in this case.

3. The prime paths of the graph are given below:

   [1, 2, 3, 4, 8, 9, 11, 12, 13]

   [1, 2, 3, 4, 8, 9, 10, 11, 12, 13]

   [1, 2, 3, 4, 5, 6, 7]

   [4, 5, 6, 7, 4]

   [5, 6, 7, 4, 5]

   [6, 7, 4, 5, 6]

   [7, 4, 5, 6, 7]

   [5, 6, 7, 4, 8, 9, 11, 12, 13]

   [5, 6, 7, 4, 8, 9, 10, 11, 12, 13]

4. The slice of the program is calculated using the following table:

m	**DEF(m)**	**Relevant$_0$(m)**	**Slice$_0$**	**Cond$_1$**	**Rel$_1$**	**Slice$_1$**	**Slice**
1	$\{x\}$	$\emptyset$	$\surd$	$\times$	$\emptyset$	$\surd$	$\surd$
2	$\{y\}$	$\{x\}$	$\surd$	$\times$	$\{x\}$	$\surd$	$\surd$
3	$\{z\}$	$\{x, y\}$	$\times$	$\times$	$\{x, y\}$	$\surd$	$\surd$
4	$\emptyset$	$\{x, y\}$	$\times$	$\surd$	$\{x, y, z\}$	$\surd$	$\surd$
5	$\{y\}$	$\{x\}$	$\surd$	$\surd$	$\{y, z\}$	$\surd$	$\surd$
6	$\{x\}$	$\{x, y\}$	$\surd$	$\times$	$\{x, y\}$	$\surd$	$\surd$
7	$\{z\}$	$\{x, y\}$	$\times$	$\times$	$\{x, y\}$	$\surd$	$\surd$
8	$\emptyset$	$\{y\}$	$\times$	$\times$	$\{x, y\}$	$\times$	$\times$
9	$\emptyset$	$\{y\}$	$\times$	$\surd$	$\{x, y\}$	$\surd$	$\surd$
10	$\{y\}$	$\emptyset$	$\surd$	$\times$	$\{x, y\}$	$\surd$	$\surd$
11	$\emptyset$	$\{y\}$	$\times$	$\times$	$\{y\}$	$\times$	$\times$
12	$\{x\}$	$\{y\}$	$\surd$	$\times$	$\{y\}$	$\surd$	$\surd$
13	$\emptyset$	$\{x\}$	$\times$	$\times$	$\{x\}$	$\times$	$\times$
		$\{x\}$			$\{x\}$		

   Lines 8 and 11 should be added to allow for compiling the calculated slice.

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).

Solution.

1. Speed: it is much faster than manual method,
2. Flexibility: it is robust against changes in the layout,
3. Precision: it captures more bugs (at the graphical interface level) than both GUI model- and manual GUI testing).
4. Ease of use: it is relatively easy to build test models and use them.