

SOFTWARE VERIFICATION AND VALIDATION

CDT414

EXAMINATION

Tuesday, January 15, 2019
Duration – 4 Hours

Course Examiners: Adnan Causevic
Sasikumar Punnekatt
Daniel Sundmark

Max score: 100 points

Preliminary grades
(ECTS): 3 60p
 4 75p
 5 85p

Course pass (G): **60p** minimum

Notes:

Write on one side of the sheet only.

Assumptions must be made when there is not enough information provided to solve an assignment, and all assumptions must be specified and explained to achieve maximum points.

Questions 1-10 (5 points each)

1. What is the difference between executing, debugging and testing the software? Please elaborate your answer.
2. What is meant by the "Pareto principle of fault distribution"?
3. Which of the following statements are correct and which are not?
Explain your answer in detail.
 - a. The goal of software testing is to prove that all defects are identified.
 - b. The goal of software testing is to detect as many failures as possible so that defects can be identified and corrected.
 - c. The goal of software testing is to prove that any remaining defect will not cause any failures.
4. Describe the Beizer's Maturity Model.
5. What is "the oracle" in a test case, and how are the terms "expected", "actual" and "verdict" connected? Please elaborate your answer.
6. In which group of test design techniques does TDD (test-driven development) belong?
7. Explain the principle behind Restrictive Active Clause Coverage (MC/DC variant) coverage criteria.
8. How can one use coverage or mutation analysis information for designing test cases? Try to elaborate your answer.
9. Describe the concepts of **safeness** and **false positives** in program analysis. How do they affect the usage of static and dynamic analysis?
10. Where in the development phase would you use static and dynamic analysis respectively? Please motivate your answer.

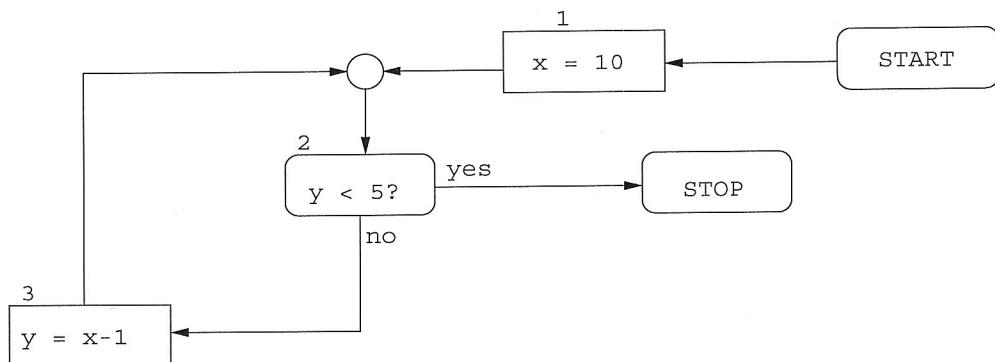
Questions 11-15 (10 points each)

11. Given the following sorting function pseudocode (source: Wikipedia)

```
procedure cocktailSort( A : list of sortable items ) defined as:  
begin := -1  
end := length( A ) - 2  
do  
    swapped := false  
    begin := begin + 1  
    for each i in begin to end do:  
        if A[ i ] > A[ i + 1 ] then  
            swap( A[ i ], A[ i + 1 ] )  
            swapped := true  
        end if  
    end for  
    if swapped = false then  
        break do-while loop  
    end if  
    swapped := false  
    end := end - 1  
    for each i in end to begin do:  
        if A[ i ] > A[ i + 1 ] then  
            swap( A[ i ], A[ i + 1 ] )  
            swapped := true  
        end if  
    end for  
    while swapped  
end procedure
```

- a. Derive the **control flow graph** of the function.
- b. Based on the control flow graph, derive the set of coverage items that need to be exercised during testing to meet the **statement (or node) coverage** adequacy criterion.
- c. Derive a **set of test inputs** that, when executed, satisfy the statement coverage criterion.
- d. Based on the control flow graph, derive the set of coverage items that need to be exercised during testing to meet the **branch coverage** adequacy criterion.

12. Perform a reaching definitions analysis for the program represented by the control flow graph below! Set up the equations, and solve them using fixed-point iteration. Show a few steps in the iterations (enough so we can see that you have understood the principle), and give the resulting sets of reaching definitions for the different program points (edges in the graph). Are there any statements where there is a risk that the value of an uninitialized variable is used?



13. Answer the following questions about the following program segment:

```

if boarded & (temp < 0) & (take-off < 30 min)
then spray_plane_with_chem;
else clear_for_takeoff;
    
```

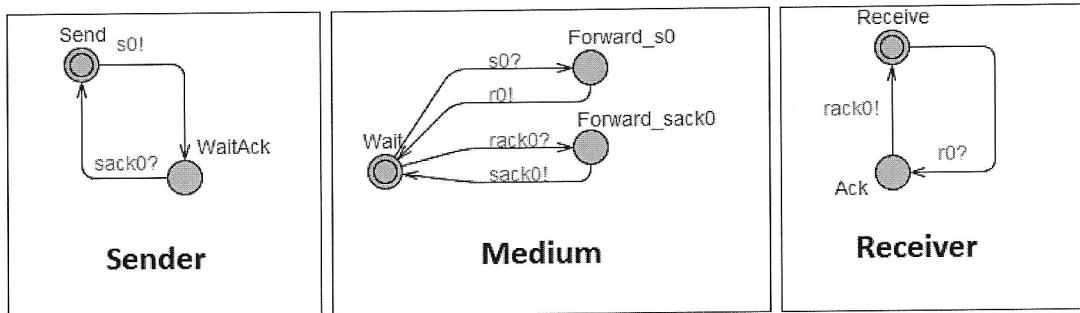
- (1) What is the minimum number of test cases you need, to fulfill Clause coverage?
- (2) Give an example of values that fulfill Clause coverage?
- (3) What is the minimum number of test cases you need, to fulfill 100% Active Clause Coverage?
- (4) Give examples of values that fulfill 100% Active Clause Coverage?

14. Take a look at the three automata that are provided below and answer the following questions:

a. What do we achieve with the following verification query:

E<> Sender.WaitAck

- b. Write a query that will check whether the model is deadlock free.
- c. Write a query to check whether every time the sender starts sending, the receiver will eventually receive the message.



15. Derive equivalence partitioning classes and write a set of tests needed for the **Triangle** problem:

Consider a program, which reads three integer values as inputs.

The three values represent the lengths of the sides of a triangle.

The program displays a message that states whether the triangle is scalene, isosceles, or equilateral.