Midterm Test

COSC 4313 3.0 Software Engineering Testing

Section M, Winter 2005

Family Name: ______________________________

Given Name(s): ______________________________

Student Number: |___ |___ |___ |___ |___ |___ |___ |___ |___ |

Question	Out of	Mark
Q1	9	
Q2	10	
Q3	6	
Q4	7	
Q5	18	
Q6	25	
Q7	25	
Total	100	
Letter grade		

1. **[9 marks]** Describe the three different types of follow-up testing that a tester can employ once a bug is discovered. Give an example for each one.

   ANSWER:
   Module on "Reporting and Analyzing Bugs", slides 19–24
   3 marks per type of follow-up testing.

2. **[10 marks]** Give five different reasons that could explain the non-reproducability of a bug.

   ANSWER:
   Module on "Reporting and Analyzing Bugs", slides 35–44
   2 marks per reason

3. **[6 marks]** Explain the importance of representing states as boolean expressions in the FREE State Model.

   ANSWER:
   Gist of pp. 205-209 in the textbook
   It is executable, i.e. unambiguous and evaluated at machine speed
   Necessary for effective automated testing

4. **[7 marks]** Explain how the slogan "Require no more, ensure no less" applies to states and inheritance.

   ANSWER:
   Page 215 in the textbook

5. **[18 marks]** Consider a logic function Z of three boolean variables A, B, C. The formula for Z is

$$Z = AB + BC$$

Describe the process of applying the Variable Negation test strategy. What is the minimum test suite suggested by it?

ANSWER:
3 marks for the following truth table (not necessary to draw for full marks).

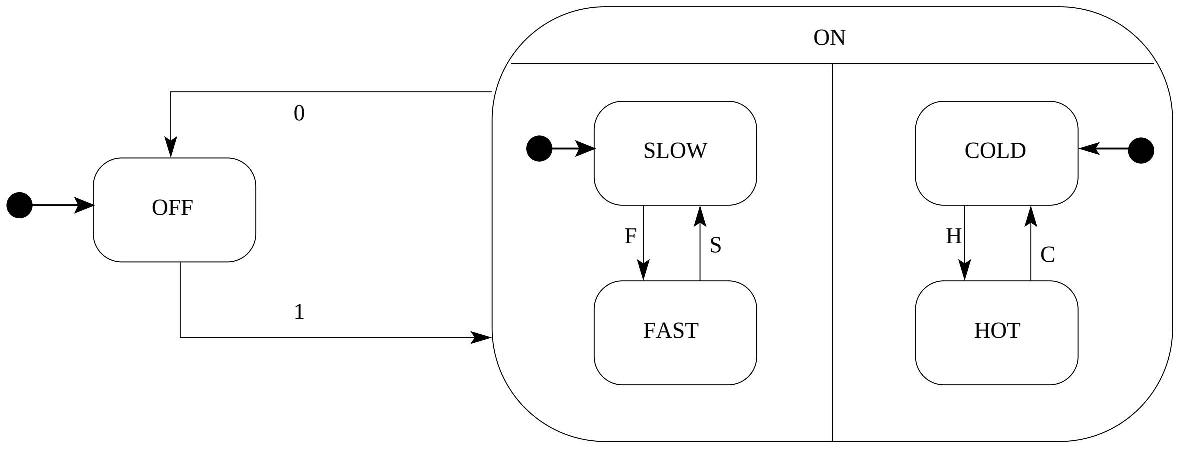
Variant	*A*	*B*	*C*	*Z*
0	*0*	*0*	*0*	*0*
1	*0*	*0*	*1*	*0*
2	*0*	*1*	*0*	*0*
3	*0*	*1*	*1*	*1*
4	*1*	*0*	*0*	*0*
5	*1*	*0*	*1*	*0*
6	*1*	*1*	*0*	*1*
7	*1*	*1*	*1*	*1*

Each row in the following table is worth 2 marks.

Candidate set #	*Term*	*Type*	*Set*
1	*AB*	*Unique true point*	*{6}*
2	*AB*	*Near false point* $A\tilde{B}$	*{4,5}*
3	*AB*	*Near false point* $\tilde{A}B$	*{2}*
4	*BC*	*Unique true point*	*{3}*
5	*BC*	*Near false point* $B\tilde{C}$	*{2}*
6	*BC*	*Near false point* $\tilde{B}C$	*{1,5}*

The minimum test suite for this function includes variants 2,3,5,6 (variant 5 covers both two-member candidate sets) [3 marks].

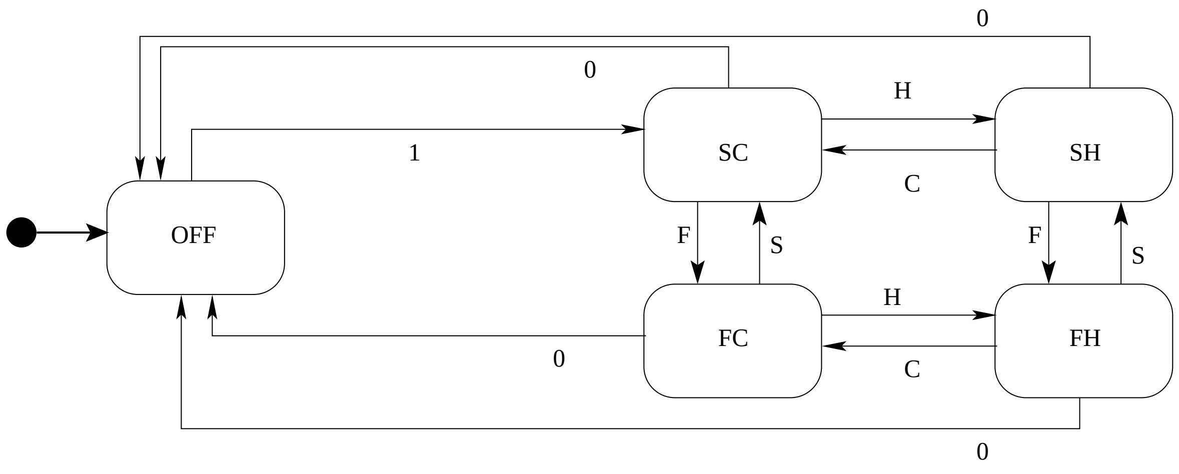
6. **[25 marks]** Consider the following statechart that describes a fan's behaviour. This kind of fan can produce an air stream that is either cold or hot in terms of temperature, and either slow or fast in terms of speed.



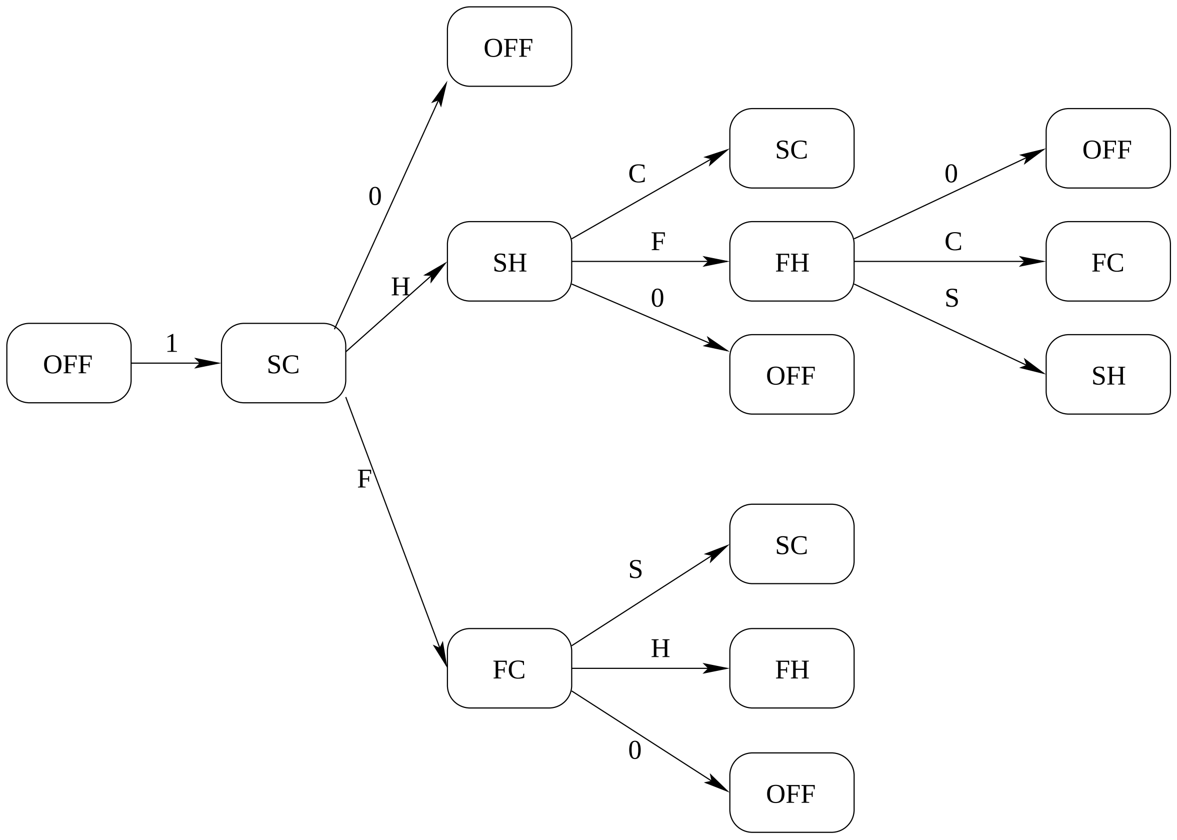
For this question, you can ignore output actions, i.e. it is assumed that if the fan is in the cold state, then its output is indeed a cold stream. You can also assume that the response matrix for this system indicates only **Ignore** responses.

In the following, derive the conformance test suite and the sneak path test suite for this system. Use breadth-first traversal for the construction of the round-trip path tree. Briefly explain the steps of the derivation process.

ANSWER:
The first step involves expanding the statechart into a state transition diagram. [8 marks]



ANSWER:
Next, we create the round-trip path tree (some variation is possible here based on the order one takes the various transitions out of a node). [9 marks]



Each one of the paths from the alpha state to a leaf node in the tree is one test case.
The sneak path test suite covers all possibilities not mentioned in the state transition diagram [8 marks]

State	*Event*	*Response*
OFF	*0*	*Ignore*
OFF	*F*	*Ignore*
OFF	*S*	*Ignore*
OFF	*H*	*Ignore*
OFF	*C*	*Ignore*
SC	*1*	*Ignore*
SC	*S*	*Ignore*
SC	*C*	*Ignore*

State	*Event*	*Response*
SH	*1*	*Ignore*
SH	*S*	*Ignore*
SH	*H*	*Ignore*
FC	*1*	*Ignore*
FC	*F*	*Ignore*
FC	*C*	*Ignore*
FH	*1*	*Ignore*
FH	*F*	*Ignore*
FH	*H*	*Ignore*

7. **[25 marks]** Consider the following specification for a method called `impl`.

   Let us consider an integer N that has at least two digits. By removing N's least significant digit, we obtain a new number M. The input to `impl` is N-M. It is guaranteed to be between 10 and 10^{18} inclusive. The output produced by `impl` to standard output must be all possible values for N separated by spaces.

   [1 mark] Which of the method-scope test patterns that we discussed in class would apply to this method?

   ANSWER:
   Category-Partition

   [8 marks] In the following, create test cases for this method based on the pattern you identified above. You may look at the implementation in the next page for hints.

   ANSWER:
   The only function of `impl` is to compute N [1 mark]
   There is one input (N-M), and one output (all possible values of N are printed to standard output) [1 mark]
   Categories and choices (2 marks each)

   - *Regular values: 17, 19, 64, 1000, 7362100.*
   - *Multiples of 9 (a multiple of 9 causes two values to be output): 18, 99, 123123123.*
   - *Special values (e.g. vary large to test overflow issues): 10, 999999999999999999, 1000000000000000000.*

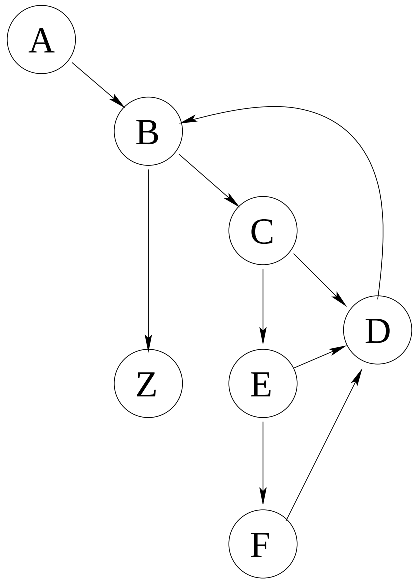
   The above values are examples and could also be divided differently.

Following is a Java implementation of `impl`.

```
void impl(long nm)
{
  for (int i=9;i>=0;i--)
  {
    long a = nm - i;
    if (a % 9 == 0)
    {
      System.out.print(a / 9 + nm);
      if (i == 9) System.out.print(" ");
    }
  }
  System.out.println();
}
```

[8 marks] Draw the Control Flow Graph for this method. Identify clearly the various code segments.

ANSWER:



1 mark per segment
1 for getting the edges right

Segment	*Code*
A	*int i=9;*
B	*i>=0;*

C	*long a = nm - i;*
	 if (a % 9 == 0)

D	*i--*

E	*System.out.print*
	 (a / 9 + nm);
	 if (i == 9)

F	*System.out.print(" ");*
Z	*System.out.println();*

[4 marks] What is a minimal set of test cases that will achieve segment coverage? Explain why.

ANSWER:
*Any test case that is a multiple of 9 will achieve segment coverage. That's because the following path will be followed ABCEFD(BCD)*BCEFDBZ.*

[4 marks] What is a minimal set of test cases that will achieve branch coverage? Explain why.

ANSWER:
*The path for the test case described above covers all branches except the one from E to D. A test case that is not a multiple of 9 will cover this branch as well, since the path followed will be A(BCD)*BCED(BCD)*BZ.*

[Extra answer space for any question]