Advanced Topics in Software Engineering (A.Y. 2019/2020) Mock exam

July 8^{th}, 2020

Exercise 1

You are building a software systems that has to control a physical system that can assume four different states σ_i with $i \in \{1, 2, 3, 4\}$, represented by the values assumed by two different boolean variables x and y, and according to specific conditions of the environment that can be represented by six boolean variables b_j with $j \in \{1, 2, 3, 4, 5, 6\}$. In particular the rules driving the behavious are the following ones:

$$\begin{array}{ll} b_1 \Rightarrow b_4 & b_1, b_2, b_3 \text{ cannot hold at the same time} \\ b_1 \vee b_3 \rightarrow x & (b_5 \wedge b_4) \vee ((b_2 \wedge b_6) \vee \neg b_1) \rightarrow y \\ b_4 \rightarrow y & \end{array}$$

- Which strategy would you apply to derive relevant test cases?
- Derive the tests according to the strategy you selected

Exercise 2

Consider the BOR, BRO, BRE criteria for testing predicates including expressions and relational operator, and shortly introduce their objectives and differences. Use the most appropriate criteria to generate a test set, able to discover logical and relational fault, for the following compound predicate:

$$((a \cdot c) \geq (b + c) \vee q) \wedge ((c = d) \vee q) \tag{1}$$

Exercise 3

Consider the development of a buffering system that has three positions and that has to abide by the following specification:

- In the initial state (state 1) the buffer is empty and it can accept the message `write` generating in output the message `ok` and then it moves to state 2. Instead in case the message received is read the buffering systems replies with a nok message and it stays in state 1.

- In state 2 and 3 the system behave similarly and in case it receives a write it replies with an ok message moving to state 3 and 4 respectively. In case the system receives a read message it replies with an ok message and it moves back to state 1 and 2 respectively.
- In the final state (state 4) the buffer is full, so in case it receives a write message it replies with a nok message and it stays in state 4. Instead in case it receives the message read it replies with an ok message and it moves to state 3.

After having modeled the system using a Finite State Machine generate a test set according to the $\mathscr{W}-method$ strategy assuming an implementation with one additional state[1]

[1] For convenience it is possible to abbreviate the input and output alphabet in the representation of the machine with write=w,read=r,ok=o,nok=n