CS 536 – Science of Programming

Homework 1 – Course Basics, Logic

Girish Rajani

A20503736

1. Course Basics

Task 1.1 (Written, 10 points)

a) **False**, even if you run the program to test it, it is not enough because it does not cover factors like unexpected inputs, unexpected user behavior, concurrency errors, changes in code nor changes in requirements. However, verification meets a specification and formally checks if a program is correct, therefore providing more of a guarantee.

b) **True**, both testing and verification are useful before deploying as testing involves the problems in functionality and the code while verification proves that the program meets the specification, both of which are important.

Task 1.2 (Written, 10 points)

a) (I)
b) (III)
c) (II)
d) (III)
e) (III)

2. Propositional Logic

Task 2.1 (Written, 5 points)

$(P \rightarrow Q) \rightarrow (\neg P \rightarrow \neg Q)$

$\boldsymbol{P}$	$\boldsymbol{Q}$	$\boldsymbol{\neg P}$	**¬Q**	$(\boldsymbol{P \rightarrow Q})$	$(\boldsymbol{\neg P \rightarrow \neg Q})$	$(\boldsymbol{P \rightarrow Q}) \rightarrow (\boldsymbol{\neg P \rightarrow \neg Q})$
T	**T**	**F**	**F**	**T**	**T**	**T**
T	**F**	**F**	**T**	**F**	**T**	**T**
F	**T**	**T**	**F**	**T**	**F**	**F**
F	**F**	**T**	**T**	**T**	**T**	**T**

It is a contingency because there does exist σ_1 and σ_2 such that $\sigma_1 \vDash (P \rightarrow Q) \rightarrow (\neg P \rightarrow \neg Q)$ and $\sigma_2 \nvDash (P \rightarrow Q) \rightarrow (\neg P \rightarrow \neg Q)$

Task 2.2 (Programming, 5 points)

See attached tauto.log file.

Task 2.3 (Written, 2 points)

As proven in 2.2, $T \Rightarrow P$ shows that P is a tautology. If we instead prove $P \Rightarrow T$, we can say that it will also show a tautology. We don't know the state of P but whichever state P may be, $P \Rightarrow T$ will return T hence making it a Tautology.

Task 2.4 (Programming, 5 points)

See attached uncurry.log file.

3. Predicate Logic

Task 3.1 (Written, 4 points)

Predicate Logic Notation:

$\forall x \in \mathrm{Z}. (E(x) \wedge \mathrm{x} > 2 \rightarrow \exists y \in M. \exists z \in M. x = y + z)$

$E(x)$ is an even number.

The above logic states that for all integer numbers where that number is even and greater than 2, there does exist two numbers (y and z) that are prime numbers whose sum is equal to that even integer number greater than 2.

Task 3.2 (Written, 9 points)

a. False – There exist an x and y where x > y for example x = 2 and y = 1 therefore making this statement false.
b. False – There is a counterexample where there does exist an a > 1 and b > 1 and where a*b = x. For example x = 12, then a = 6 and b = 2. 6 > 1 and 2 > 1 and 6*2 = 12.
c. True – For every single x, we can indeed find a y such that $y = x * 2$. For example if x = 8 then y can be 16.

Task 3.3 (Programming, 10 points)

See attached pred.log file.

4. One more wrap-up question

Task 4.1 (Written, 0 points)

6 hours (But it was over a long period of days so I can’t remember exactly)