**IIT CS536: Science of Programming**

Homework 1: Course Basics, Logic

Prof. Stefan Muller

Submitted by: Abdurakhmon Urazboev

**Task 1.1**

1. Testing provides more of a guarantee that a program is correct than verification, because you actually run the program.

**ANSWER:** False, because testing only runs the program for multiple specified cases and checks if it is giving expected results. Verification, however, makes sure that the program works correctly by checking it matches specifications for the program assuming all the requirements are well defined.

1. It is useful to do both testing and verification on a program before deploying it.

**ANSWER:** True, because verification and testing do complementary job. Verification makes sure that program does what it was logically designed to do. Testing on the other hand, actually runs the program with testing cases and tests how running program handles given input.

**Task 1.2**

1. and are neither syntactically not semantically equal.
2. and are neither syntactically not semantically equal.
3. and are semantically equal but not syntactically.
4. and are neither syntactically not semantically equal.
5. and areneither syntactically not semantically equal.

**Task 2.1**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
| T | T | F | F | T | T | T |
| T | F | F | T | F | T | F |
| F | T | T | F | T | F | F |
| F | F | T | T | T | T | T |

From the truth table we can say that this proposition is contingency as it has both true and false cases.

**Task 2.2**

T => P -> Q \/ P

T

T \/ Q Dom

(~P \/ P) \/ Q LEM

~P \/ (P \/ Q) Assoc

~P \/ (Q \/ P) Comm

P -> Q \/ P DC

**Task 2.3**

As we saw in the lecture logical implication means that whenever Q is true, so is the P . Proving means that P is always true as value true on the left side of the logical implication means that the proposition on the right is always true. This means that this proposition is a Tautology. If we reverse the positions

**Task 2.4**

P -> (Q -> R) => (P /\ Q) -> R

P -> (Q -> R)

~P \/ (Q -> R) DC

~P \/ (~Q \/ R) DC

(~P \/ ~Q) \/ R Assoc

~(P /\ Q) \/ R DM

(P /\ Q) -> R DC

**Task 3.1**

In words, for all integers z which is the double of another integer p and bigger than 2 there exists such x and y in a set of prime numbers such that their sum is equal to z.

**Task3.2**

1. this is false as for all integers x not all integers are bigger than x. There are some which are bigger and some which are smaller than x.
2. This statement is false because as we know certain numbers are equal to the multiplication of the 2 other integers. For example, we take . As we can see for this case the statement do not hold as it says: For any all integers there doesn’t exist 2 integers which are bigger than 1 and whose multiplication is equal to that integer.
3. This statement is true as for any integer x, there exist another integer which is equal to .

**Task 3.3**

(Forall x. P(x) -> Q(x)) => ~(Exists x. P(x) /\ ~Q(x))

(Forall x. P(x) -> Q(x))

(Forall x. ~P(x) \/ Q(x)) DC

~~(Forall x. ~P(x) \/ Q(x)) DNE

~(Exists x. ~(~P(x) \/ Q(x))) DM

~(Exists x. (~~P(x) /\ ~Q(x))) DM

~(Exists x. (P(x) /\ ~Q(x))) DNE

**Task 4.1**

I have spent approximately 3-4 hours on this homework