Guideline

Homework for week 01.

Please submit your solutions in a single PDF on Brightspace.

If you are writing by hand, please ensure your handwriting is legible.

Multiple submissions are possible before the due time; the last submission will be graded.

Exercise 1. (points = 10)

Check if the two statement forms below are logically equivalent. Produce a truth table and make a conclusion.

- p \/ q -> r
- (p -> r) /\ (q -> r)

Exercise 2 (points = 10)

Check if the two statement forms below are equivalent. Produce a truth table and make a conclusion.

- (p -> q) /\ (q -> r) /\ (r -> p)
- p /\ q /\ r

Exercise 3 (points = 30)

Consider six statement forms (a-f):

- (a) p -> q
- (b) q -> p
- (c) ~p \/ q
- (d) ~q \/ p
- (e) ~q -> ~p
- (f) ~p -> ~q

1. Find all statement forms that are equivalent to (a), except (a) itself.
2. Find all statement forms that are equivalent to (b), except (b) itself.

For this exercise, the explanation is not needed. Just show your results concisely.

Exercise 4 (points = 10)

Consider the proposition ~P /\ (Q -> P). What can you conclude about P and Q if you know the statement is true? For this exercise, the explanation is not needed. Just show your results concisely.

Exercise 5 (points = 20)

A tautology and a contradiction are terms used in logic to describe specific types of propositions.

- A *tautology* is a proposition that is always true, regardless of the truth values of its constituent parts. In other words, a proposition is a tautology if it is true in every row of a truth table. For example, p \/ ~p. An example in plain language: "It will either rain today or it won't."
- A *contradiction* is a proposition that is always false, regardless of the truth values of its constituent parts. In other words, a proposition is a contradiction if it is false in every row of a truth table. For example, p /\ ~p. An example in plain language: "I will finish the homework today and I will not."

For each statement form below, determine if it is a tautology, contradiction, or neither. For this exercise, the explanation is not needed. Just show your results concisely.

1. (~p \/ q) \/ (p /\ ~q)
2. (p /\ ~q) /\ (~p \/ q)
3. (p /\ q) \/ (~p \/ (p /\ ~q))

Exercise 6 (points = 20)

Verifying Simple Code with Hoare Logic

Hoare Logic is a formal system for reasoning about the correctness of computer programs. Tony Hoare introduced it in 1969, which contributed to his Turing Award. Hoare Logic helps in verifying that a program does what it is supposed to do. Interested students should read this: https://amturing.acm.org/award_winners/hoare_4622167.cfm

Hoare Logic revolves around the concept of Hoare Triple, which has the following format:

```
{P} Code {Q}
```

- P is the precondition. It's a proposition about the variables in your program that is true before the code runs.
- Q is the postcondition. It's a proposition that you expect to be true after the code runs.
- Code is the program or part of the program that you're analyzing.

Examples of correct Hoare Triples

- {x = 1} `y := x + 1` {y = 2 /\ x = 1}
- {x = 1} `y := x + 1` {y = 2}
- {x = 0} `while (x<=10) {x = x + 1}` {x = 11}

Note: Above y = 2 is weaker than y = 2 /\ x = 1, but it remains to be correct after `y := x + 1` is executed from precondition x = 1.

Examples of wrong Hoare Triple:

- {x = 1} `y := x + 1` {y - x = 2}
- {x > 0} `while (x<=10) {x = x + 1}` {x = 11}

Note: The second is incorrect because it states that: If x > 0, x will equal 11 at the end of the loop. This is flawed for two reasons: (1) If x starts as 0.5, then at the end of the loop, x will be 10.5, not 11. (2) If x is a value greater than 10, such as 42, then the loop will terminate without changing the value of x at all.

Questions: Determine whether the following Hoare triple is correct or not. For this exercise, the explanation is not needed. Just show your results concisely.

1. {x = 3 /\ y = 2} `Code` { x = 2 /\ y = 3}, where "Code" refers to the following

```
x := x + y
y := x − y
x := x − y
```

2. {x > 0} `while (x<=10) {x = x + 1}` {x > 10}