CIS 301: Logical Foundations of Programming

Fall 2024

Exam 1 – 100 points

**This test is closed-notes and closed-computers.**

This exam has 9 problems.

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Solution\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Score: \_\_\_\_\_\_\_\_\_\_\_\_

1. (11 pts) Consider the following code fragment:

num = 0;

if (test > 4 && test <= 10) {

num = 1;

}

else if (test <= 4) {

num = 2;

}

else {

num = 3;

}

// 🡨 suppose we are right here

Suppose *num* has the value 3 immediately after the above code fragment finishes (i.e., at the point marked with the 🡨 just after the brackets). What can we conclude about *test?* **Put a box around what you know about *test*** (being as concise as you can) and explain your answer.

1. (8 pts) Draw a circuit for the following logical formula: **NOT(p OR q) OR (p AND NOT r).** Use only a combination of AND, OR, and NOT gates.
2. (8 pts) Draw a parse tree for the following logical formula:

a ∧ b → c ∧ d V ¬e ∧ f

1. (14 pts) Consider the following pairs of propositional logic statements. For each pair, write either “equivalent” or “not equivalent”. Use either truth tables or a truth assignment to defend your answer, and write a sentence explaining why your truth tables/truth assignment demonstrates that the statements are or are not equivalent. (If you use truth tables, you may use one “combined” truth table that has a section for each statement.)
2. p → ¬ q

q ∨ ¬p

truth assignment

This truth assignment makes the first statement (?) and the second statement (T)

1. ¬(a ∨ b)

¬a ∧ ¬b

truth table – clear what the top-level operators were

For every possible truth assignment, the two statements have the same output so they are equivalent.

1. (8 pts) Apply DeMorgan's laws to write an if-statement whose condition is the negation of the condition in the if-statement below. Write your if-statement in such a way that it does not use any ! (not) symbols. Remember to consider order of operations.

if (a <= 7 && x >= 4 || list.contains(item)) {

//statements

}

Write your negated if-statement below:

if ((a > 7 || x < 4) && list.contains(item) == false) {

//statements

}

1. (8 pts) Is the statement (p ∧ q ∧ ¬r) → ¬(p V q) satisfiable? Defend your answer, using the definition of satisfiability.

Need ONE truth assignment that makes the overall statement true.

1. (15 pts) Consider the following questions about arguments.

a) (3 pts) Find a truth assignment that demonstrates that the argument below is invalid. You do not need an explanation – just a truth assignment.

Premises: p → q, ¬p

Conclusion: ¬q

p = F

q = T

b) (3 pts) Find a truth assignment that demonstrates that the argument below is invalid. You do not need an explanation – just a truth assignment.

Premises: (p ∨ q) → r, (p → q ) ∨ r, ¬r

Conclusion: p ∨ q

p = F

q = F

r = F

c) (5 pts) How did your truth assignments in (a) and (b) demonstrate that those arguments were invalid? What happened in each case with the premises and conclusion?

We found a truth assignment where all premises were true but the conclusion was false, which means the argument was invalid

d) (4 pts) Write one sentence to explain how you would use truth tables to prove that an argument ***is*** valid.

We would need a truth table that shows that every truth assignment that makes all the premises true also makes the conclusion true.

1. (12 pts) Complete the following natural deduction proof. (I know “h” doesn’t appear as a propositional atom in the list of premises, but that is intentional. Think about why you know the conclusion must be true based on the premises.)

( p ∧ q, m ∧ n) ⊢ ( h ∨ n ∧ q ∧ p)

Proof(

1 ( p ∧ q ) by Premise,

2 ( m ∧ n ) by Premise,

//FINISH THE PROOF BELOW

)

1. (16 pts) Complete the following natural deduction proof:

(p ∨ q, a, a ∧ p → z, q ∧ a → z ) ⊢ ( z )

Proof(

1 ( p ∨ q ) by Premise,

2 ( a ) by Premise,

3 ( a ∧ p → z ) by Premise,

4 ( q ∧ a → z ) by Premise,

//FINISH THE PROOF BELOW

)