CIS 301: Logical Foundations of Programming

Practice Exam 1

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

(This exam is similar to a previous exam, but has been edited to match the new Logika format.)

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

1. (15 pts) Consider the following questions about logical analysis of conditional statements.

a) (7 pts) Consider the following code fragment:

num = 4;

if (x < 10 || (y > 100 && y < 200)) {

if (x >= 10) {

num = 18;

}

}

// 🡨 suppose we are right here

Suppose *num* has the value 18 immediately after the above code fragment finishes (i.e., at the point marked with the 🡨 just after the brackets). What can we conclude about *x* and *y*? Explain as concisely as you can.

We know num was changed from 4 to 18, so it must have gone inside both if statements.

So both true:

x < 10 || (y > 100 && y < 200)

x >= 10

can’t have x < 10 since conflicts with x >=10, so (y > 100 && y < 200) must be true.

In summary: x >=10, y is between 100 and 200, exclusive

b) (8 pts) Consider the following code fragment:

num = 4;

if (val1 > 0 || val2 < 10) {

num = 7;

}

else if (val1 < 0) {

num = 10;

}

else {

num = 20;

}

// 🡨 suppose we are right here

Suppose *num* has the value 20 immediately after the above code fragment finishes (i.e., at the point marked with the 🡨 just after the brackets). What can we conclude about *val1* and *val2*? Explain as concisely as you can.

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

p → q V ¬r ∧ p

1. (13 pts) Use two truth tables to demonstrate that the following two statements are logically equivalent:

(p → q) ∧ (q → p)

(p V ¬q) ∧ (q V ¬p)

Afterwards, give a brief explanation about why your truth tables demonstrate that the statements are equivalent.

\* \*

------------------------------------------------------------------

p q # (p → q) ∧ (q → p) (p V ¬q) ∧ (q V ¬p)

--------------------------------------------------------------------

We can see that the output of both statements is the same for every input, 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.

if ((total >= 100 && Character.isDigit(ch) == false) || num < 10) {

//statements

}

Write your negated if-statement below:

1. (8 pts) Is the statement (p V ¬q) ∧ (¬p V ¬q) ∧ (p → q) satisfiable? How do we know?

p = F

q = F

This truth assignment makes the statement true, so it is satisfiable.

1. (11 pts) Consider the following (invalid) argument:

**Premises:**

If I order fries, then I get ketchup.

If I get ketchup, then I get a cheeseburger.

I don’t get ketchup.

**Conclusion**:

I don’t get a cheeseburger.

a) (4 pts) Translate each premise and conclusion to propositional logic. Start by identifying each propositional atom.

p: I order fries

q: I get ketchup

r: I get a cheeseburger

premises: p -> q, q -> r, !q

conclusion: !r

b) (7 pts) Provide a truth assignment for your translations in (a) that demonstrates that the argument is NOT valid. How do you know that truth assignment makes the argument invalid? Explain.

premises: p -> q, q -> r, !q

conclusion: !r

p =F

q= F

r = T

This truth assignment makes every premise true but makes the conclusion false, so the argument is invalid.

1. (14 pts) Complete the following natural deduction proof:

( a ∧ b ∧ d, e ) ⊢ ( (b ∧ e ) ∨ (d ∧ a ) )

Proof(

//YOUR PROOF GOES HERE

)

1. (15 pts) Complete the following natural deduction proof:

( a ∨ b, a  → c ∧ d, b → g, d → g ) ⊢ ( g )

Proof(

//YOUR PROOF GOES HERE

1 ( a ∨ b ) by Premise,

2 ( a  → c ∧ d ) by Premise,

3 ( b → g ) by Premise,

4 ( d → g ) by Premise,

5 SubProof(

6 Assume (a),

7 ( c ∧ d ) by ImplyE(2, 6),

8 ( d) by AndE2(7),

9 ( g )by ImplyE(4, 8)

),

10 SubProof(

11 Assume(b),

12 (g) by ImplyE(3, 11)

),

13 (g) by OrE(1, 5, 10)

)