Exam #1

Thursday, August 13, 2020

- This exam has 12 questions, with 100 points total.

- You have **two hours**.

- **You should submit your answers in the Gradescope platform** (not on NYU Classes).

- **It is your responsibility to take the time for the exam** (You may use a physical timer, or an online timer: https://vclock.com/set-timer-for-2-hours/).
Make sure to upload the files with your answers to gradescope BEFORE the time is up, while still being monitored by ProctorU.
We will not accept any late submissions.

- In total, you should upload 3 '.cpp' files:
    - One '.cpp' file for questions 1-10.
      Write your answer as one long comment (/* … */).
      Name this file 'YourNetID_q1to10.cpp'.
    - One '.cpp' file for question 11, containing your code.
      Name this file 'YourNetID_q11.cpp'.
    - One '.cpp' file for question 12, containing your code.
      Name this file 'YourNetID_q12.cpp'.

- **Write your name, and netID at the head of each file.**

- This is a closed-book exam. However, you are allowed to use
CLion or Visual-Studio. You should create a new project, and work ONLY in it.
You may also use two sheets of scratch paper.
Besides that, no additional resources (of any form) are allowed.

- You are not allowed to use C++ syntactic features that were not covered in the Bridge program so far.

- Calculators are **not** allowed.

- Read every question completely before answering it.
Note that there are 2 programming problems at the end.
Be sure to allow enough time for these questions

Table 1.5.1: Laws of propositional logic.

Idempotent laws:	$p \lor p \equiv p$	$p \land p \equiv p$
Associative laws:	$(p \lor q) \lor r \equiv p \lor (q \lor r)$	$(p \land q) \land r \equiv p \land (q \land r)$
Commutative laws:	$p \lor q \equiv q \lor p$	$p \land q \equiv q \land p$
Distributive laws:	$p \lor (q \land r) \equiv (p \lor q) \land (p \lor r)$	$p \land (q \lor r) \equiv (p \land q) \lor (p \land r)$
Identity laws:	$p \lor F \equiv p$	$p \land T \equiv p$
Domination laws:	$p \land F \equiv F$	$p \lor T \equiv T$
Double negation law:	$\neg\neg p \equiv p$	
Complement laws:	$p \land \neg p \equiv F$ $\neg T \equiv F$	$p \lor \neg p \equiv T$ $\neg F \equiv T$
De Morgan's laws:	$\neg(p \lor q) \equiv \neg p \land \neg q$	$\neg(p \land q) \equiv \neg p \lor \neg q$
Absorption laws:	$p \lor (p \land q) \equiv p$	$p \land (p \lor q) \equiv p$
Conditional identities:	$p \rightarrow q \equiv \neg p \lor q$	$p \leftrightarrow q \equiv (p \rightarrow q) \land (q \rightarrow p)$

Table 1.12.1: Rules of inference known to be valid arguments.

Rule of inference	Name
p $p \rightarrow q$ $\therefore q$	Modus ponens
$\neg q$ $p \rightarrow q$ $\therefore \neg p$	Modus tollens
p $\therefore p \lor q$	Addition
$p \land q$ $\therefore p$	Simplification

Rule of inference	Name
p q $\therefore p \land q$	Conjunction
$p \rightarrow q$ $q \rightarrow r$ $\therefore p \rightarrow r$	Hypothetical syllogism
$p \lor q$ $\neg p$ $\therefore q$	Disjunctive syllogism
$p \lor q$ $\neg p \lor r$ $\therefore q \lor r$	Resolution

Table 1.13.1: Rules of inference for quantified stateme

Rule of Inference	Name
c is an element (arbitrary or particular) $\forall x\, P(x)$ $\therefore P(c)$	Universal instantiation
c is an arbitrary element $P(c)$ $\therefore \forall x\, P(x)$	Universal generalization
$\exists x\, P(x)$ $\therefore$ (c is a particular element) $\wedge P(c)$	Existential instantiation*
c is an element (arbitrary or particular) $P(c)$ $\therefore \exists x\, P(x)$	Existential generalization

Table 3.6.1: Set identities.

Name	Identities	
Idempotent laws	$A \cup A = A$	$A \cap A = A$
Associative laws	$(A \cup B) \cup C = A \cup (B \cup C)$	$(A \cap B) \cap C = A \cap (B \cap C)$
Commutative laws	$A \cup B = B \cup A$	$A \cap B = B \cap A$
Distributive laws	$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$	$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$
Identity laws	$A \cup \varnothing = A$	$A \cap U = A$
Domination laws	$A \cap \varnothing = \varnothing$	$A \cup U = U$
Double Complement law	$\overline{\overline{A}} = A$	
Complement laws	$A \cap \overline{A} = \varnothing$ $\overline{U} = \varnothing$	$A \cup \overline{A} = U$ $\overline{\varnothing} = U$
De Morgan's laws	$\overline{A \cup B} = \overline{A} \cap \overline{B}$	$\overline{A \cap B} = \overline{A} \cup \overline{B}$
Absorption laws	$A \cup (A \cap B) = A$	$A \cap (A \cup B) = A$

Part I – Theoretical:

- ***You don't need to justify your answers to the questions in this part.***
- *For all questions in this part of the exam (questions 1-10), you should submit a* ***single*** *'.cpp' file. Write your answers as one long comment (`/* … */`).*
  Name this file 'YourNetID_q1to10.cpp'.

Question 1 (5 points)

a. Convert the decimal number $(148)_{10}$ to its base-2 representation.
b. Convert the 8-bits two's complement number $(10010100)_{\text{8-bit two's complement}}$ to its decimal representation.

Question 2 (5 points)

Select the proposition that is logically equivalent to $\neg(p \rightarrow \neg q)$.

a. $p \wedge q$
b. $p \wedge \neg q$
c. $\neg p \wedge q$
d. $\neg p \wedge \neg q$
e. None of the above

Question 3 (5 points)

The domain for variables x and y is the set {1, 2, 3}.
The table below gives the values of P(n, m) for every pair of elements from the domain.
For example, P(2, 3) = F because the value in row 2, column 3, is F.

P	1	2	3
1	T	T	T
2	T	F	F
3	F	T	F

Select the statement that is **true**.

a. $\forall x\ \forall y\ [P(x,y) \neq P(y,x)]$
b. $\forall x\ \exists y\ [P(x,y) \neq P(y,x)]$
c. $\forall x\ \forall y\ [P(x,y) = P(y,x)]$
d. None of the above

Question 4 (5 points)

The domain of discourse is the set of all positive integers.

Select the logical expression that is equivalent to: “There is a smallest positive integer.”

a. $\forall x \exists y\ [(x \neq y) \rightarrow (x < y)]$

b. $\exists x \forall y\ [(x \neq y) \rightarrow (x < y)]$

c. $\forall x \exists y\ [(x \neq y) \land (x < y)]$

d. $\exists x \forall y\ [(x \neq y) \land (x < y)]$

Question 5 (5 points)

$A = \{x \in Z : x\ is\ even\}$

$B = \{x \in Z : x\ is\ a\ prime\ number\}$

$D = \{5, 7, 8, 12, 13, 15\}$

Select the set corresponding to $D - (A \cup B)$.

a. $\{15\}$

b. $\{8, 12, 15\}$

c. $\{5, 7, 13, 15\}$

d. $\{5, 7, 8, 12, 13, 15\}$

Question 6 (5 points)

Which rule is used in the argument below?

“Alice is a patient that shows COVID-19 symptoms. Alice got a COVID-19 test but the result was negative. Therefore, there is a symptomatic patient who got a COVID-19 test that resulted negative.”

a. Universal instantiation

b. Universal generalization

c. Existential instantiation

d. Existential generalization

Question 7 (5 points)

Theorem: Consider a group of 3 families that have a total of 14 kids. Then, at least one of the families has at least 5 kids.

A proof by contradiction of the theorem starts by assuming which fact?

a. There is a family with 5 or fewer kids.

b. There is a family with fewer than 5 kids.

c. All the families have 5 or fewer kids.

d. All the families have less than 5 kids.

Question 8 (10 points)

$A = \{1, 2, 3, \{1\}, \{2\}, \{3, 4\}\}$.

For each of the following statements, state if they are true or false (no need to explain your choice).

a. $1 \in A$

b. $1 \subseteq A$

c. $\{1, 2\} \in A$

d. $\{1, 2\} \subseteq A$

e. $\{1, 2\} \in P(A)$

f. $\{1, 2\} \subseteq P(A)$

g. $\{3, 4\} \in A$

h. $\{3, 4\} \subseteq A$

i. $\emptyset \in A$

j. $\emptyset \subseteq A$

Question 9 (5 points)

Consider the following function:

$f: \{0,1\}^4 \rightarrow \{0,1\}^4$. f(x) is obtained from x by swapping the value of the ones bit (if the ones bit is 0, it would be changed to1, and if the ones bit is 1, it would be changed to 0).

For example, f(100**0**)=100**1**, and , f(110**1**)=110**0**.

Select the correct description of the function f.

a. One-to-one and onto

b. One-to-one but not onto

c. Onto but not one-to-one

d. Neither one-to-one nor onto

Question 10 (5 points)

Let A={1, 2, 3, 4}.

The function $f: P(A) \rightarrow P(A)$ is defined as: for every $X \in P(A)$, $f(X) = (X \cap \{1,2\}) \cup \{4\}$.

For example, $f(\{1, 3\}) = \{1, 4\}$.

Note: $P(A)$ is the powerset of A (a set with all A's subsets)

Find the range of f.

Part II – Coding:

- *For **each** question in this part (questions 11-12), you should submit a '.cpp' file, containing your code.*
- *Pay special attention to the style of your code. Indent your code correctly, choose meaningful names for your variables, define constants where needed, choose most suitable control statements, etc.*
- *In all questions, you may assume that the user enters inputs as they are asked.*
  For example, if the program expects a positive integer, you may assume that user will enter positive integers.
- *No need to document your code. However, you may add comments if you think they are needed for clarity.*

Question 11 (20 points)

Write a program that reads a positive integer, n, and prints a nxn square chess board shape. That is a nxn square with an alternating pattern of squares in two colors.
In this question, instead of two colors, we will use two symbols 'X' and 'O'.

Your program should interact with the user **exactly** as demonstrated in the following two executions:

Execution example 1:
```
Please enter a positive integer:
3
XOX
OXO
XOX
```

Execution example 2:
```
Please enter a positive integer:
8
XOXOXOXO
OXOXOXOX
XOXOXOXO
OXOXOXOX
XOXOXOXO
OXOXOXOX
XOXOXOXO
OXOXOXOX
```

Question 12 (25 points)
Consider the following definition:
Definition: A positive integer is called a ***more-even number*** if it has more even digits than odd ones (in the decimal representation of the number).
For example, 410 is a *more-even number*, while 4123 is not.

Write a program that reads from the user a sequence of positive integers and tells how many of them were *more-even numbers*.

Implementation requirement: The user should enter their numbers, each one in a separate line, and type -1 to indicate the end of the input.

Your program should interact with the user **exactly** the same way, as demonstrated bellow:

```
Please enter a sequence of positive integers, each one in a separate line.
End your sequence by typing -1:
3203
44
21168
5466
758
-1
You entered 3 more-even numbers
```