Practice Exam 1

Tables Provided on Exam 1

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) $\underline{\forall x\ P(x)}$ $\therefore$ P(c)	Universal instantiation
c is an arbitrary element $\underline{P(c)\quad}$ $\therefore \forall x\ P(x)$	Universal generalization
$\underline{\exists x\ P(x)}$ $\therefore$ (c is a particular element) $\wedge$ P(c)	Existential instantiation*
c is an element (arbitrary or particular) $\underline{P(c)\quad}$ $\therefore \exists x\ P(x)$	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	$A \cup (A \cap B) = A$	$A \cap (A \cup B) = A$

Set Theory

Given:

A = { 1, {2}, {{3, 4}} }
For each of the following statements, state whether they are **true** or **false.**

a. 1 ∈ A
b. 1 ⊆ A
c. {2} ∈ A
d. {2} ⊆ A
e. {3, 4} ∈ A
f. {3, 4} ⊆ A
g. {{3,4}} ∈ A
h. {{3,4}} ⊆ A
i. ∅ ∈ A
j. ∅ ⊆ A

Solutions:

a. T, the number 1 is an element of set A

b. F, the number 1 is not contained within a set of braces, thus is not a subset of A

c. T, the set containing 2 is an element of A (the number 2 is contained within a set of { } within A)

d. F, and would only be T if {{2}} was contained within A

e. F, because {3,4} would be need to contained within {{ }} rather than within just { }

f. F, because {{3,4}} would need to be an element contained within {{{ }}}

g. T, the set containing 3 & 4 ({3, 4}) is an element of set A

h. F

i. F, the empty set is always a subset and not an element of the larger set

j. T, the empty set is only a subset (always just a subset) of the larger set

1.2 Let A = {1, 2, 3, 4}. Select the statement that is **false**.

a. ∅ ∈ P(A)
b. ∅ ⊆ P(A)
c. {2, 3} ∈ A
d. {2, 3} ⊆ A

Solution:

C

Functions

Choose the property for which the function satisfies if well defined.

a. Neither one-to-one, nor onto

b. One-to-one, but not onto

c. Onto, but not one-to-one

d. one-to-one and onto

e. not well defined

Given a function whose domain is the set of all integers and whose target is the set of all positive integers:

a)

$f(x) = \{x >= 0 : 2x + 1 \wedge x < 0 : -2x\}$

Solution: d

b)

$f(x) = |x| + 1$

Solution: c

c)

$f(x) = x\text{^}2 + 1$

Solution: a

d)

$f(x) = \{x > 0 : 2x + 1 \wedge x <= 0 : -2x\}$

Solution: e

e)

$f(x) = x >= 0 : 2x + 1 \wedge x < 0 : -2x + 2$

Solution: b

Proofs

4.1 Direct Proof

Prove that the product of two odd integers is an odd integer.

Proof.

Assume that integers m and n are odd integers. We will show that m*n is also an odd integer.

Integers m and n are odd integers

$\Rightarrow$ m = 2k+1 for some integer k and n = 2j + 1 for some integer j

$\Rightarrow$ m*n = (2k+1)(2j+1) for some integers k and j

$\Rightarrow$ = 4kj + 2k + 2j + 1

$\Rightarrow$ = 2(kj + k + j) + 1

Since j and k are both integers, then 2(kj + k + j) is also an integer. Since m*n can be expressed as 2 times an integer plus 1, m*n is an odd integer.

4.2 Proof by Contrapositive

Prove that if n^2 is even, then n is even.

Consider the contraposition of the proposition which is ~q→~p. Show that if n is odd, then n^2 is odd.

Proof.

Assume the hypothesis of the implication to be true.

Assume that *n* is an odd integer.

Show the implication.

n is an odd integer

$\Rightarrow$ n = 2q+1 for some integer q

$\Rightarrow$ $n^2 = (2q + 1)^2$ for some integer q

$\Rightarrow$ $n^2 = 4q^2 + 4q + 1$ for some integer q

$\Rightarrow$ $n^2 = 2(2q^2 + 2q) + 1$ for some integer q and because q is an integer, then $(2q^2 + 2q)$ is also an integer,

$\Rightarrow$ $n^2 = 2j + 1$ for some integer j

Therefore n^2 is an odd integer that can be represented as 2j+1.

Having used a direct proof of the contrapositive statement, we can conclude that if n^2 is even, then n is even.

4.2 Proof by Contradiction
Prove by contradiction that if 3n+5 is odd, then n is even.

Proof.
Begin by assuming that 3n+5 is odd and n is *odd*.
Integer n is odd therefore n=2k+1 for some integer k.

3n+5 = 3(2k+1) + 5 for some integer k, substituting n=2k+1 for n in 3n+5
= 6q +8 for some integer k, algebra
= 2(3k +4) for some integer k, and since k is an integer, then (3k+4) is also an integer. Further, since (3k+4) is an integer, and 2 multiplied by anything is an even number, then 2(3k+4) is even.

Thus, 3n+5 is even and because this **contradicts** the hypothesis that 3n+5 is odd then n is even and we have proven our conclusion.

Number Systems Conversion

5.1 Decimal to 8-bit Two's Complement

$(-43)_{10} = (11010101)_{\text{8-bit Two's Complement}}$

5.2 Binary to Hexadecimal

$(110011100)_2 = (19C)_{16}$

Coding

```
    a
   b b
  c   c
 d     d
e       e
 d     d
  c   c
   b b
    a
```

Make a hollowed-out diamond made up of in-order alphabet letters. An example is below when `n=5`.

Solution:

```
#include <iostream>
using namespace std;

int main(){
   int userInput = 7;
   char letter = 'a';
   char space = ' ';

   int outsideSpaceLimit = userInput - 1;
   int insideSpaceLimit = 0;

   for (int i = 1; i < (userInput * 2); i++) {
       // outside spaces
       for (int j = 0; j < outsideSpaceLimit; j++) {
           cout << space;
       }
       // first letter

       cout << letter;

       //inside spaces
       for (int j = 0; j < insideSpaceLimit; j++) {
           cout << space;
       }

       //second letter
       if (i != 1 && i != ((userInput * 2) -1)){
           cout << letter;
       }

       if(i < userInput){
           letter++;
           insideSpaceLimit += 2;
           outsideSpaceLimit--;
       }
```

```
        else{
            letter--;
            insideSpaceLimit -= 2;
            outsideSpaceLimit++;
        }

        cout << endl;
    }


    return 0;
}
```

Challenge Question

For this question, you have to create the following Barn Door shape with `n=10`:

```
##########
#$      $#
# $    $ #
#  $  $  #
#   $$   #
#   $$   #
#  $  $  #
# $    $ #
#$      $#
##########
```

ONE answer (again, there are **many** ways to do this!):

```
#include <iostream>
using namespace std;

int main(){
    int input;

    char frame = '#';
    char planks = '$';
    char space = ' ';

    cout << "Enter positive integer: ";
    cin >> input;

    for (int i = 0; i < input; i++) { //rows
        for (int j = 0; j < input; j++) {
            // left or right or top or bottom
            if (j == 0 || j == input - 1 || i == 0 || i == input - 1){
                cout << frame;
            }
            // left diagonal going right || right diagonal going left
            else if(i == j || i + j == input - 1){
                cout << planks;
            }
            else{
                cout << space;
            }

        }
        cout << endl;
    }
```

```
    return 0;
}
```