MATH 2305: Discrete Mathematics Janak Joshi, Dallas College, Fall 2024

1 Logic and Proofs

1.1 Propositional Logic

A proposition is a declarative statement that is either true or false, but not both. We use letters to denote propositional variables. The conventional letters for propositional variables are p, q, r, s. True is notated as T and false is notated as F. An atomic proposition is a proposition that cannot be expressed in terms of simpler propositions.

> **Definition**
>
> Let p be a proposition. The negation of p, denoted by $\neg p$ (or $\bar{p}$) means "It is not the case that p".
>
> The proposition $\neg p$ is read as "not p". The truth value of the negation of p, $\neg p$ is the opposite of the truth value of p.

The negation of a proposition can also be considered the result of the operation of the negation operator on a proposition.

The next operator is a connective and it is used to form new propositions from two or more existing propositions.

> **Definition**
>
> Let p and q be propositions. The conjunction of p and q, denoted by $p \wedge q$, is the proposition of "p and q". The conjunction $p \wedge q$ is true when both are true, and false when both are false.

> **Definition**
>
> Let p and q be propositions. The disjunction of p and q, denoted by $p \vee q$, is the proposition "p or q". The disjunction $p \vee q$ is false when both p and q are false, otherwise it is true.

> **Definition**
>
> Let p and q be propositions. The exclusive or of p and q, denoted by $p \oplus q$ is the proposition that is true when exactly one of p and q is true and is false otherwise.

There are other ways propositions can be combined.

> **Definition**
>
> Let p and q be propositions. The conditional statement $p \rightarrow q$, is the proposition, "if p, then q". The conditional statement $p \rightarrow q$ is false when p is true and q is false, and true otherwise. p is called the hypothesis and q is called the conclusion.

A conditional statement is also called an implication.

With $p \rightarrow q$, we can form three related conditional statements.

The first is the proposition $q \rightarrow p$, which is the converse of $p \rightarrow q$.

The contrapositive of $p \rightarrow q$ is $\neg q \rightarrow \neg p$.

The inverse of $p \rightarrow q$ is $\neg p \rightarrow \neg q$.

The contrapositive of a conditional statement is equal to it. We all two compound propositions equivalent when they always have the same truth values. The converse and inverse of a conditional statement are equivalent as well.

There is another way to combine propositions that expresses that two propositions have the same truth value.

Definition

Let p and q be propositions. The biconditional statement $p \leftrightarrow q$ is the proposition "p if and only q". This statement is true if p and q have the same truth values, and is false otherwise.

The negation operator is applied before all logical operators. Another general rule is that the conjunction operator takes precedence over the disjunction operator. It is an accepted rule that conditional and biconditional operators have lower precendence than the conjuction and disjunction operators.

A bit is a symbol with two values, 0 and 1. 1 is true, and 0 is false.

1.2 Propositional Equivalences

Definition

A compound proposition that is always true, no matter what the truth values of the propositional values that occur in it, is called a tautology. A compound proposition that is always false is called a contradiction. If it is neither a tautology or a contradiction, it is called a contingency.

Compound propositions that have the same truth values in all possible cases are called logically equivalent.

Definition

The compound propositions p and q are called logically equivalent if $p \leftrightarrow q$ is a tautology. The notation $p \equiv q$ denotes that p and q are logically equivalent.

We can establish logical equivalence of more than two compound propositions. Generally 2^n rows are required if a compound proposition involves n propositional variables.

TABLE 6 Logical Equivalences.	
Equivalence	*Name*
$p \wedge \mathbf{T} \equiv p$ $p \vee \mathbf{F} \equiv p$	Identity laws
$p \vee \mathbf{T} \equiv \mathbf{T}$ $p \wedge \mathbf{F} \equiv \mathbf{F}$	Domination laws
$p \vee p \equiv p$ $p \wedge p \equiv p$	Idempotent laws
$\neg(\neg p) \equiv p$	Double negation law
$p \vee q \equiv q \vee p$ $p \wedge q \equiv q \wedge p$	Commutative laws
$(p \vee q) \vee r \equiv p \vee (q \vee r)$ $(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$	Associative laws
$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$ $p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$	Distributive laws
$\neg(p \wedge q) \equiv \neg p \vee \neg q$ $\neg(p \vee q) \equiv \neg p \wedge \neg q$	De Morgan's laws
$p \vee (p \wedge q) \equiv p$ $p \wedge (p \vee q) \equiv p$	Absorption laws
$p \vee \neg p \equiv \mathbf{T}$ $p \wedge \neg p \equiv \mathbf{F}$	Negation laws

TABLE 7 Logical Equivalences Involving Conditional Statements.

$p \to q \equiv \neg p \vee q$

$p \to q \equiv \neg q \to \neg p$

$p \vee q \equiv \neg p \to q$

$p \wedge q \equiv \neg(p \to \neg q)$

$\neg(p \to q) \equiv p \wedge \neg q$

$(p \to q) \wedge (p \to r) \equiv p \to (q \wedge r)$

$(p \to r) \wedge (q \to r) \equiv (p \vee q) \to r$

$(p \to q) \vee (p \to r) \equiv p \to (q \vee r)$

$(p \to r) \vee (q \to r) \equiv (p \wedge q) \to r$

TABLE 8 Logical Equivalences Involving Biconditional Statements.

$p \leftrightarrow q \equiv (p \to q) \wedge (q \to p)$

$p \leftrightarrow q \equiv \neg p \leftrightarrow \neg q$

$p \leftrightarrow q \equiv (p \wedge q) \vee (\neg p \wedge \neg q)$

$\neg(p \leftrightarrow q) \equiv p \leftrightarrow \neg q$

Credit: Rosen's Discrete Mathematics 8e

When using De Morgan's Laws, remember to change the logical connective after you negate.

De Morgan's Laws tell us how to negate conjunctions and how to negate disjunctions. The equivalence $\neg(p \vee q) \equiv \neg p \wedge \neg q$ tells us that the negation of a disjunction is formed from the conjunction of the negations of the component propositions. The negation of a conjunction also gives you the disjunction of the negations of component propositions.

A compound proposition is satisfiable if there is an assignment of truth values to its variables that make it true. If no such assignments exist, then it is unsatisfiable.

1.3 Predicates and Quantifiers

We will introduce predicate logic now. For example in the statement x is greater than 3, the variable is x and the predicate is "is greater than 3".

In general a statement involving n variables $x_1, x_2, \cdots, x_n$ can be denoted as $P(x_1, x_2, \cdots, x_n)$.

Definition

The universal quantification of $P(x)$ is the statement:

"$P(x)$ for all values of x in the domain."

The notation $\forall x P(x)$ denotes the universal quantification of $P(x)$. Here $\forall$ is called the universal quantifier. An element for which $P(x)$ is false is called a counterexample to $\forall x P(x)$.

Definition

The existential quantification of $P(x)$ is the proposition:

"There exists an element in x in the domain such that $P(x)$"

We use the notation $\exists x P(x)$ for the existential quantification of $P(x)$.

Remember the truth value for both $\forall x P(x)$ and $\exists x P(x)$ depends on the domain.

TABLE 2 De Morgan's Laws for Quantifiers.			
Negation	*Equivalent Statement*	*When Is Negation True?*	*When False?*
$\neg \exists x P(x)$	$\forall x \neg P(x)$	For every x, $P(x)$ is false.	There is an x for which $P(x)$ is true.
$\neg \forall x P(x)$	$\exists x \neg P(x)$	There is an x for which $P(x)$ is false.	$P(x)$ is true for every x.

Credit: Rosen's Discrete Mathematics 8e

1.4   Nested Quantifiers

1.5   Rules of Inference

1.6   Introduction to Proofs

1.7   Proof Methods and Strategy

2 Sets, Functions, Sequences, Sums, and Matrices

2.1 Sets

2.2 Set Operations

2.3 Functions

2.4 Sequences and Summations

3   Algorithms

3.1   Algorithms

4 Number Theory and Cryptography

4.1 Divisibility and Modular Arithmetic

4.2 Integer Representation and Algorithms

4.3 Primes and Greatest Common Divisors

4.4 Solving Congruences

5 Induction and Recursion

5.1 Mathematical Induction

5.2 Strong Induction and Well-Ordering

5.3 Recursive Defintions and Structural Induction

5.4 Recursive Algorithms

6 Counting

6.1 The Basics of Counting

6.2 The Pigeonhole Principle

6.3 Permuations and Combinations

6.4 Binomial Coefficients and Identities

6.5 Generalized Permuations and Combinations

7 Discrete Probability

7.1 An Introduction to Discrete Probability

8 Advanced Counting Techniques

8.1 Applications of Recurrence Relations

8.2 Inclusion-Exclusion

9 Relations

9.1 Relations and Their Properties

10 Graphs

10.1 Graphs and Graph Models

10.2 Graph Terminology and Special Types of Graphs

11 Trees

11.1 Introduction to Trees