MATH 2305 Notes

Stasya

Fall 2024

A course designed to prepare math, computer science, and engineering majors for a background in abstraction, notation, and critical thinking for the mathematics most directly related to computer science. Topics include: logic, relations, functions, basic set theory, countability and counting arguments, proof techniques, mathematical induction, combinatorics, discrete probability, recursion, sequence and recurrence, elementary number theory, graph theory, and mathematical proof techniques.

Contents

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 precendence 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

A nested quantifier is when one quantifier is in the scope of another.

Here is table that summarizes different quantifications involving two variables.

TABLE 1 Quantifications of Two Variables.

Statement	*When True?*	*When False?*
$\forall x \forall y P(x, y)$ $\forall y \forall x P(x, y)$	$P(x, y)$ is true for every pair x, y.	There is a pair x, y for which $P(x, y)$ is false.
$\forall x \exists y P(x, y)$	For every x there is a y for which $P(x, y)$ is true.	There is an x such that $P(x, y)$ is false for every y.
$\exists x \forall y P(x, y)$	There is an x for which $P(x, y)$ is true for every y.	For every x there is a y for which $P(x, y)$ is false.
$\exists x \exists y P(x, y)$ $\exists y \exists x P(x, y)$	There is a pair x, y for which $P(x, y)$ is true.	$P(x, y)$ is false for every pair x, y.

1.5 Rules of Inference

Rules of inference are our basic tools for establishing the truth of statements.

An argument in propositional logic is a sequence of propositions. All but the final proposition in the argument are called premises and the final proposition is called the conclusion. An argument is valid if the truth of all its premises implies the conclusion is true.

An argument form in propositional logic is a sequence of compound propositions involving propositional variables. An argument form is valid if no matter what particular propositions are substituted for the propositional variables in the premises, the conclusion is true if the premises are all true.

The tautology $(p \wedge (p \rightarrow q)) \rightarrow q$ is the basis of the rule of inference modus ponens, or the law of detachment, basically: if p and $p \rightarrow q$ $\therefore q$.

Yet again another chart I took from Rosen.

TABLE 1 Rules of Inference.

Rule of Inference	*Tautology*	*Name*
p $p \to q$ $\therefore\ q$	$(p \wedge (p \to q)) \to q$	Modus ponens
$\neg q$ $p \to q$ $\therefore\ \neg p$	$(\neg q \wedge (p \to q)) \to \neg p$	Modus tollens
$p \to q$ $q \to r$ $\therefore\ p \to r$	$((p \to q) \wedge (q \to r)) \to (p \to r)$	Hypothetical syllogism
$p \vee q$ $\neg p$ $\therefore\ q$	$((p \vee q) \wedge \neg p) \to q$	Disjunctive syllogism
p $\therefore\ p \vee q$	$p \to (p \vee q)$	Addition
$p \wedge q$ $\therefore\ p$	$(p \wedge q) \to p$	Simplification
p q $\therefore\ p \wedge q$	$((p) \wedge (q)) \to (p \wedge q)$	Conjunction
$p \vee q$ $\neg p \vee r$ $\therefore\ q \vee r$	$((p \vee q) \wedge (\neg p \vee r)) \to (q \vee r)$	Resolution

Fallacies resemble rules of inference, the difference lies in that they are based in contingencies rather than tautologies.

There are also some rules of inference for statements involving quantifiers. This is what Rosen summarized:

TABLE 2 Rules of Inference for Quantified Statements.

Rule of Inference	*Name*
$\forall x P(x)$ $\therefore\ P(c)$	Universal instantiation
$P(c)$ for an arbitrary c $\therefore\ \forall x P(x)$	Universal generalization
$\exists x P(x)$ $\therefore\ P(c)$ for some element c	Existential instantiation
$P(c)$ for some element c $\therefore\ \exists x P(x)$	Existential generalization

Because universal instantiation and modus ponens are so often used together, the combination of rules is called universal modus ponens. This rule tells us if $\forall x(P(x) \to Q(x))$ is true, and if $P(a)$ is true for a particular element a in the domain of the universal quantifier, then $Q(a)$ must also be true. Note that by universal instantiation $P(a) \to Q(a)$ is true. By modus ponens, $Q(a)$ must also be true.

This is how it is described:

$$\begin{array}{l} \forall x\,(P(x) \to Q(x)) \\ P(a)\text{, where } a \text{ is a particular element in the domain} \\ \hline \therefore\ Q(a) \end{array}$$

There is also universal modus tollens:

$$\begin{array}{l} \forall x\,(P(x) \to Q(x)) \\ \neg Q(a) \text{ where } a \text{ is a particular element in the domain} \\ \hline \therefore\ \neg P(a) \end{array}$$

1.6 Introduction to Proofs

A proof is a valid argument that establishes the truth of a mathematical statement.

A theorem is a statement that can be shown to be true. We show the truth using a proof.

A less important theorem that is helpful in the proof of other results is called a lemma.

A corollary is a theorem that can be established directly from a theorem that has been proved.

A conjecture is a statement that is being proposed to be a true statement.

A direct proof of a conditional statement $p \to q$ is constructed with the assumption that p is true with the goal for the combination of p being true and q being false to never happen.

Definition

The integer n is even if there exists an integer k such that $n = 2k$, and n is odd if there exists an integer k such that $n = 2k + 1$. Two integers have the same parity when both are even or both are odd; they have opposite parity when one is even and the other is odd.

An indirect proof is a type of proof that is not direct. One example is proof by contraposition. P roofs by contraposition use the fact that $p \to q$ is equal to $\neg q \to \neg p$. We have to show $\neg q \to \neg p$ is true to show that $p \to q$ is true as well.

We can prove that $p \to q$ is true if p is false as well. This is called a vacuous proof.

If we show that q is true in $p \to q$ this is called a trivial proof.

Definition

The real number r is rational if there exist integers p and q with $q \neq 0$ such that $r = p/q$. A real number that is not rational is irrational.

Suppose we want to prove a statement p is true. Suppose that we can find a contradiction q such that $\neg p \to q$ is true. Because q is false, but $\neg p \to q$ is true, we can conclude $\neg p$ is false, meaning p is true.

To find a contradiction q that might help us find that p is true, we can show that $\neg p \to (r \wedge \neg r)$ is true for some proposition r. This is called a proof by contradiction.

To prove a theorem in the form $p \leftrightarrow q$, we show that both $p \to q$ and $q \to p$ are both true.

Many incorrect arguments are based on a fallacy called begging the question or circular reasoning. This fallacy occurs when one or more steps of a proof are based on the truth of the statement being proved.

1.7 Proof Methods and Strategy

Suppose we have a conditional $(p_1 \vee p_2 \vee \cdots \vee p_n) \to q$. We can separate the proof into different cases, called proof by cases by proving each of the n conditional statements.

Some theorems can be proved by examining a relatively small number of examples. This is called proof by exhaustion.

The phrase "without loss of generality" (WLOG) means that we assert by proving one case of a theorem, no additional argument is required to prove other specified cases.

Many theorems are assertions that objects of a particular type exist. A theorem of this type is a proposition in the form $\exists x P(x)$, where P is a predicate. A proof of a proposition in this form is called an existence proof. Sometimes an existence proof can be given by finding an element a, called a witness, such that $P(a)$ is true. This existence proof is called constructive. If we prove $\exists x P(x)$ is true in another way it can be nonconstructive.

For a uniqueness proof, we need two parts:

- Existence: We show that an element x with the desired property exists.
- Uniqueness: We show that if x and y both have the desired property, $x = y$

Forward reasoning is a proof when you start with your premises. You construct a proof with steps leading to the conclusion. It may sometimes be helpful to use backwards reasoning, we find a statement p that we can prove $p \rightarrow q$.

2 Sets, Functions, Sequences, Sums, and Matrices

2.1 Sets

Sets are used to group objects together.

A set is an unordered collection of objects, called elements or members of the set. A set contains its elements. We write $a \in A$ to denote a is in an element of set A. The notation $a \notin A$ denotes a is not an element of set A.

Here are some sets to remember:

- $\mathbb{N}$ is the set of all natural numbers
- $\mathbb{Z}$ is the set of all integers
- $\mathbb{Z}^+$ is the set of all positive integers
- $\mathbb{Q}$ is the set of all rational numbers
- $\mathbb{R}$ is the set of all real numbers
- $\mathbb{R}^+$ is the set all positive real numbers
- $\mathbb{C}$ is the set of all complex numbers

Two sets are equal only if they contain the same elements.

An empty set is notated as $\emptyset$.

A set with one element is a singleton set.

Set A is a subset of set B and set B is the superset of set A if every element of A is also an element of B. To indicate A is a subset of B we write $A \subseteq B$. For the equivalent superset, we write $B \supseteq A$.

For every nonempty set S, there is a guarantee to have at least two subsets, the empty set and the set S itself.

When we want to say that A is a subset of B, but $A \neq B$, we can write $A \subset B$.

If there are n distinct elements in a set S, then the set is finite and n is the cardinality of S. The cardinality of S is denoted as $|S|$.

Otherwise, the set is infinite if it is not finite.

Definition

Given a set S, the power set of S is the set of all subsets of the set S. The power set of S is defined as $\mathcal{P}(S)$.

The power set of a set has 2^n elements. Because sets are unordered, we need to represent ordered collections using ordered n-tuples.

Definition

The ordered n-tuple $(a_1, a_2, \cdots, a_n)$ is the ordered collection that has a_1 as its first element,

2.2 Set Operations

If we let A and B be sets, the union of the sets, $A \cup B$, is the set that contains those elements that are either in A or B, or in both.

The intersection of the sets, $A \cap B$, is the set containing those elements in both A and B.

Two sets are called disjoint if the intersection of the sets is an empty set.

The difference of sets A and B, or $A - B$ is the set containing those elements that are in A but not in B. It is also called the complement of B with respect to A.

Definition

Let U be the universal set. The complement of set A denoted as $\overline{A}$ is the complement of A with respect to U. Therefore the complement of the set A is $U - A$.

Much like the last chapter, there are some set identities and properties

TABLE 1 Set Identities.

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

Credits to Rosen again.

The union of a collection of sets is the set that contains those elements that are members of at least one set in the collection.

The intersection of a collection of sets is the set that contains those elements that are members of all sets in the collection.

2.3 Functions

Definition

Let A and B be empty sets. A function f from A to B is an assignment of exactly one element of B to each element of A. We write $f(a) = b$ if b is the unique element of B assigned by the function f to the element a of A. If f is a function from A to B, we write $f : A \to B$.

If f is a function from A to B, we say A is the domain of f and B is the codomain of f. Also if $f(a) = b$, we can say b is the image of a and a is a preimage of b. The range, or image, is the set of all images of elements of A. Also, if f is a function from A to B, we say that f maps from A to B.

A function is called real-valued if its codomain is the set of real numbers and integer-valued if the codomain is the set of integers.

Some functions never assign the same value to two different domain elements. These are called one-to-one functions, or injective functions.

A function is called surjective or onto when the range and codomain are equal.

If a function is both surjective and injective, then it is bijective.

Only a one-to-one function can be invertible because the inverse of a one-to-one function exists.

2.4 Sequences and Summations

Sequences are ordered lists of elements. The terms of a sequence can be specified by providing a formula for each term of the sequence.

A sequence is used to represent an ordered list. We use the notation a_n to denote the image of the integer n. We call a_n a term of the sequence.

A geometric progression is a sequence in the following form:

$$a, ar, ar^2, \cdots, ar^n, \cdots$$

where the initian term a and common ratio r are real numbers.

An arithmetic progression is a seuqnece in the form:

$$a, a + d, a + 2d, \cdots, a + nd, \cdots$$

where the initial term a and the common difference d are real numbers.

A recurrence relation for the sequence a_n is an equation that expresses a_n in terms of one or more of the previous terms in the sequence.

A sequence is called a solution of a recurrence relation if its terms satisfy the recurrence relation.

Definition

The Fibonacci sequence, $f_0, f_1, f_2, \cdots$, is defined by the initial conditions $f_0 = 0, f_1 = 1$, and the recurrence relation:

$$f_n = f_{n-1} + f_{n-2}$$

for $n = 2, 3, 4, \cdots$.

Now we introduce the summation notation.

We use the notation $\sum_{j=m}^{n} a_j$ to represent $a_m + a_{m+1} + \cdots + a_n$.

Here j is the index of summation and is abitrary.

Sums of terms of geometric progressions commonly arise.

$$\sum_{j=0}^{n} ar^j = \frac{ar^{n+1} - a}{r - 1}$$

when $r \neq 1$ and $(n + 1)a$ when $r = 1$.

Here is some formulae for commonly occurring sums:

TABLE 2 Some Useful Summation Formulae.

Sum	*Closed Form*
$\sum_{k=0}^{n} ar^k \ (r \neq 0)$	$\frac{ar^{n+1} - a}{r-1}, r \neq 1$
$\sum_{k=1}^{n} k$	$\frac{n(n+1)}{2}$
$\sum_{k=1}^{n} k^2$	$\frac{n(n+1)(2n+1)}{6}$
$\sum_{k=1}^{n} k^3$	$\frac{n^2(n+1)^2}{4}$
$\sum_{k=0}^{\infty} x^k, \lvert x \rvert < 1$	$\frac{1}{1-x}$
$\sum_{k=1}^{\infty} kx^{k-1}, \lvert x \rvert < 1$	$\frac{1}{(1-x)^2}$

Credits to Rosen again.

3 Algorithms

An algorithm is a finite sequence of precise instructions for performing a computation or for solving a problem.

There are many properties of algorithms to keep in mind:

- Input - input values from a specified set
- Output - from each set of input values an algorithm produces output values
- Definiteness - the steps of an algorithm must be defined precisely
- Correctness - an algorithm should produce correct output values for each set of input values
- Finiteness - an algorithm should produce the desired output after a finite number of steps for any input
- Effectiveness - it must be possible to perform each step of an algorithm exactly and in a finite amount of time
- Generality - the procedure should be applicable for all problems of the desired form

The first algorithm to present is the linear search, or sequential search. This search begins by comparing a x and a_1 and continues with each a_n until a match is found.

The binary search works when the list is sorted. We first split the list into two smaller sublists of the same size, and keep splitting it up based on the comparison to the term to be found the middle term.

The bubble sort is one of the simplest sorting algorithms. It puts a list into increasing order by successively comparing adjacent elements.

The insertion sort begins with the second element and compares it with the the first element. Then the third element is compared with the first and then the second if it is larger than the first.

Many algorithms are designed to solve optimization problems. This means they want to maximize or minimize some parameter. Algorithms that make what seems to be the "best" choice at each step are called greedy algorithms.

An example would be the cashier's algorithm which makes changes using the fewest coins possible when change is made from quarters, dimes, nickels, and pennies.

The halting problem is a interesting problem. It asks whether there is a procedure that can input a computer program and determine whether the program will eventually stop.

The reason there isn't one is because you do not know if it will never halt or you haven't waited long enough for it to terminate.

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

8 Advanced Counting Techniques

8.1 Applications of Recurrence Relations

8.2 Inclusion-Exclusion

9 Relations

10 Graphs

10.1 Graphs and Graph Models

10.2 Graph Terminology and Special Types of Graphs

11 Trees