MATH 220 Notes

Reese Critchlow

Sets

Sets: A **set** is a collection of objects. The objects are referred to as **elements** of **members** of the set.

For sake of notation, sets are generally described as capital letters A, B, C, X, Y, and members of a set are described as their respective lowercase letters a, b, c, x, y.

Integers are generally referred to as i, j, k, l, m, n and reals are generally referred to as x, y, z, w.

Set Notation: Sets can also describe membership by means of curly braces.

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

Ellipses can also be used to demonstrate that a pattern continues, however, this often can be hazardous as the pattern must be obvious to the reader.

$$B = \{1, 2, 3, \dots\}$$

The **empty set** is also important to note, which defines the set which contains no elements. It is denoted as $\emptyset$, where for any x, $x \notin \emptyset$, and $\emptyset = \{\}$.

Set Builder Notation: Set builder notation is the general approach to describing sets with a general rule. A standard form of a set described by set builder notation is as follows:

$$S = \{\text{some expression} \mid \text{some rule}\} \qquad S = \{\text{a function} \mid \text{a domain}\}.$$

For example, we can have:

$$\begin{aligned} E &= \{n : n \text{ is an even integer}\} \\ E &= \{n \mid n = 2k \text{ for some } k \in \mathbb{Z}\} \\ E &= \{2n \mid n \in \mathbb{Z}\}. \end{aligned}$$

Statements and Open Sentences

Statements: A statement is an declarative sentence that can be assigned a truth value.

Open Sentences: An open sentence is a sentence whose truth value depends on another “thing” inside of the sentence. For example, take the sentence/equation:

$$x^2 - 5x + 4 = 0.$$

Because no information on the value of x is provided, the truth value of the sentence is unknown.

Negations: Negating a statement requires a full inversion of the truth table. A negation of a statement P can be denoted as $\sim P$, $!P$ or $\sim P$.

Logical Operations

Disjunction (Or): The **disjunction** of two statements P and Q is the statement “P or Q” and is denoted $P \vee Q$. The disjunction is true if at least one of P and Q are true. The disjunction is only false if both P and Q are false.

Conjunction (And): The **conjunction** of two statements P and Q is the statement “P and Q” and is denoted $P \wedge Q$. The conjunction is true when both P and Q are true. It is false if at least one of P and Q are false.

The Implication: For statements P and Q, the **implications** or **conditional** is the statement: if P then Q, and is denoted $P \Rightarrow Q$. In this context, we define

- P is called the **hypothesis**

- Q is called the **conclusion**

Thus, the only case in which the implication is false is when the hypothesis is true, and the conclusion is false.

It is important to note that the implication shares the same truth table as the statement:

$$(\sim P) \vee Q.$$

It is also important to note that the implication is not symmetric. Thus, $P \Rightarrow Q$ is not the same as $Q \Rightarrow P$.

Implications of the Implication

Modus Ponens: By the truth table of the implication, there is one crucial entry that makes the basis for proofs. That is when both the hypothesis and the conclusion are true. Thus, we introduce **Modus Ponens**, which is the deduction that:

- P implies Q is true, and
- P is true
- Hence, Q must be true

Thus, if one can prove that Q is true under the assumption that P is true, then the implication is proven. Modus Ponens also opens the door to chaining conditionals together to prove the original implication.

Modus Tollens: Extrapolating off of Modus Ponens, we can introduce **Modus Tollens**, which is the deduction:

- P implies Q is true, and
- Q is false
- hence P must be false.

This will later be the foundation for the *contrapositive*.

Affirming the Consequent: A common logical error is the false deduction:

- P implies Q is true, and
- Q is true,
- and hence P must be true.

This is called **affirming the consequent** and it is wrong because the flow of logic is wrong.

Denying the Antecedent: Is another common logical error in which the **false** deduction is made that:

- P implies Q is true, and
- P is false,
- and hence Q must be false.

Negating the Conditional: Since the conditional has the same truth table as $(\sim P) \vee Q$, we can write the negation of the conditional as $P \wedge (\sim Q)$.

Contrapositive: The **contrapositive** of an implication is $(\sim Q) \Rightarrow (\sim P)$. It has the same truth table as the implication, and thus can be used as another method to prove an implication.

Converse: The **converse** of an implication is $Q \Rightarrow P$. It does not share the same truth table as the implication.

Inverse: The **inverse** of an implication is $(\sim P) \Rightarrow (\sim Q)$. It does not share the same truth table as the implication.

Biconditional: The **biconditional** is where for statements P and Q, P implies Q and Q implies P. This can be written as:

- P if and only if Q
- P iff Q
- $P \iff Q$.

General Overview of Direct Proofs

Most proofs start with the a conditional that must be proven. Thus, one must demonstrate that under the hypothesis, the conclusion holds. Many of these proofs have the following form:

1. Assume the original statement is true.
2. Provide an alternative form for the original statement.
3. Manipulate both sides of the equation until you reach a form that fits the conclusion.

There exist some common methods for proving such statements.

1. Proving the Contrapositive: Instead of proving $P \Rightarrow Q$, one can prove $(\sim Q) \Rightarrow (\sim P)$.
   Use Cases:
   - To escape having to prove non-divisibilities.
   - To escape having to prove something does not have membership.
   - For when the negation of a statement is simply easier to prove.
2. Proof by Cases: In the case that a statement cannot be proven directly, introducing multiple cases which fit the problem favourably can be of assistance.
   Use Cases:
   - Exploiting divisibilities (cases stem from varying remainder values)
   - Using all odd/even integers (cases for odd or even integers)
   - Having permanent signs in statements with **absolute values**
   - Proving cases for which a value is in different ranges (generally positive or negative)

There also exist some common practices which are common with many general proofs:

1. Using terms to complete the square (often useful with **inequalities** to demonstrate that a term is positive)
2. Performing "ghost operations" where equations are manipulated in seemingly counterintuitive ways $(3 + 2) = 5$ to achieve the desired form.
3. Exploiting subsets, for example, if you need to prove something for all primes, P but you can prove it for all odd numbers and 2 (S), since all primes are either odd or two, then $P \subset S$, so the statement holds for all primes.

Quantifiers

Quantifiers come in two main forms:

The Universal Quantifer: Denoted $\forall$ and is read "for all".

The Existential Quantifer: Denoted $\exists$ and is read "there exists" or "at least one"

Quantifiers help describe how many members of a set fit a certain condition.

Negation of Quantifiers: To negate a quantifier, it is important to note some main points

- The two main quantifiers, $\forall$ and $\exists$ swap.

- The domains do not change.
- The condition for the members is negated as well.

Bounding Proofs

A function is bounded if it satisfies the following statement:

$$\exists M \in \mathbb{R} \text{ s.t. } \forall x \in f : A \to \mathbb{R}, |f(x)| \leq M.$$

Bounded Proofs: A bounded proof generally follows the form of:

1. Choose an $M(x)$ such that $|f(x)| \leq M$ is satisfied.

Unbounded Proofs: An unbounded proof needs to satisfy the statement:

$$\forall M \in \mathbb{R}, \exists x \in f : A \to \mathbb{R} \text{ s.t. } |f(x)| > M.$$

Thus an unbounded proof takes the form of:

1. Let M be arbitrary, such that $M \in \mathbb{R}$.
2. Carefully choose an x value, generally dependant on M, such that $|f(x)| > M$.

Sequence Convergence Proofs

A sequence (x_n) is said to converge to the value L if it satisfies the following statement:

$$\forall \epsilon > 0, \exists N \in \mathbb{N} \text{ s.t. } \forall n \in \mathbb{N}, (n \geq N \Rightarrow |x_n - L| \leq \epsilon)\,.$$

Convergence Proofs: A convergence proof can generally take the following form:

1. Let $\epsilon > 0$ be arbitrary.
2. Carefully choose some N, which is generally a function of ϵ. This can generally be achieved by holding $|x_n - L| \leq \epsilon$ and solving for n.
3. Assume $n \geq N$ and substitute values to show that $|x_n - L| \leq \epsilon$.

Non-Convergence Proofs: The negation of the convergence statement is as follows:

$$\exists \epsilon > 0 \text{ s.t. } \forall N \in \mathbb{N}, \exists n \in \mathbb{N} \text{ s.t. } (n \geq N) \wedge (|x_n - L| > \epsilon).$$

Common proofs for non-convergence proofs generally resemble:

1. Let $N \in \mathbb{N}$ be arbitrary.
2. Carefully choose ϵ, which is generally a constant. This is usually obtained by solving $|x_n - L| > \epsilon$.
3. Choose some value for n such that it fits the inequality $|x_n - L| > \epsilon$. Since $n \geq N$ for all N, generally, a max function is required, generally of the form: $\max\{a, N\}$, where a is some number that minimally satisfies $|x_n - L| > \epsilon$.

Limit Proofs

A function $f : A \to R$ has the limit L as x goes to a, as denoted by:

$$\lim_{x \to a} f(x) = L.$$

Thus, to prove a limit, we need to prove the limit statement true:

$$\forall \epsilon > 0, \exists \delta > 0 \text{ s.t. } (0 < |x - a| < \delta) \Rightarrow (|f(x) - L| < \epsilon)$$

Limit Proofs: A limit proof can generally take the following form:

1. Let $\epsilon > 0$ be arbitrary.
2. Choose $\delta > 0$, which is generally a function of ϵ. It is usually case that to find your δ, one must:
   - First try to directly solve $|f(x) - L| < \epsilon$, such that the inequality takes the form of $|x - a| < \delta(\epsilon)$.
   - However, this is not always the case, generally, when δ is a function of both ϵ and x, so, it is possible to choose δ to be a minimum function such that we can manipulate x terms.

   So δ is generally either some function of ϵ, or some minimum function $\delta = \min\{c, g(\epsilon)\}$, where c is some value to make algebra easier.
3. Starting with $0 < |x - a| < \delta$, use the minimum and/or rearrange the expression to gain information on x.
4. From the minimum function, we can also extrapolate that $|x - a| < \delta \leq g(\epsilon)$.
5. Using both pieces of information from (3) and (4), $|f(x) - L| < \epsilon$ can be manipulated to prove the statement.

Triangle Inequality: For limit proofs, and many other proofs, it is important to note the triangle inequality, which states that:

$$|x + y| \leq |x| + |y|$$

for any real valued x, y.

Normal Induction

Normal Induction proofs rely on being able to prove a **base case** and then an **inductive step**. Normal Induction proofs come in the following forms:

- Bounding: use the base case and surrounding values to bound the inductive step.
- Explicit Statements: such as a Fibonacci Series use a recursive definition of a series. Using prior elements of these series generally are the solution to solving these problems.
- Series: generally use the original series plus an additional term to solve them.

Inductive Proofs: An inductive proof has the following form:

Statement:

$$\forall n > a, n \in \mathbb{Z}, P(n).$$

Where a is the base case.

1. Prove the Base Case: Let $n = a$. Show the reader that $P(a)$ is true.
2. Assume the hypothesis: Assume that $P(k)$ is true.
3. Prove the Inductive Step: Manipulate $P(k + 1)$ such that $P(k + 1)$ is true.

Remark: Often, it is easy to confuse yourself where you are in the proof. Say that for $P(n)$ to be true, $P(n) = f(n)$. It is important to note that you are going to be searching for $P(n+1) = f(n+1)$, not $P(n+1) = f(n)$. Thus, it is often helpful to write out what $f(n+1)$ is first, so that you know where you want to end up.

Strong Induction

Strong induction is where all prior cases imply the next case in an inductive proof.

Final Exam

Returning to Sets

Power Sets: A power set of a set A, denoted $\mathcal{P}(A)$ is the set of all subsets of A. We can also note that if $|A| = n$, then $|\mathcal{P}(n) = 2^n$.
Set Operations:

- Union: The union of two sets A and B are all of the elements that are in either A **or** B, denoted $A \cup B$.
- Intersection: The intersection of two sets A and B are the elements that are in both A **and** B, denoted $A \cap B$.
- Difference: The difference of two sets A and B is the elements which are in A, less the elements that are in B, denoted $A - B$.
- Complement: Given a set A which is a subset of a larger set, U, then the complement of A is all of the elements in U which are not in A, denoted $\overline{A}$.

Cartesian Product: The Cartesian product of two sets A and B is the set of ordered pairs containing elements of A and B.

Relations

We can say that two elements of a set A bear a relation to each other when they satisfy some rule outlined by the given relation.

Equivalence Relations: Most problems surrounding relations will be in the form of equivalence proofs. To prove that a relation is an equivalence relation, we must prove the following properties:

- Reflexivity: $\forall a \in A, aRa$ or $(a, a) \in R$
- Symmetry: $\forall a, b \in A, aRb \implies bRa$.
- Transitivity: $\forall a, b, c \in A, ((aRb) \wedge (bRc)) \implies aRc$.

Equivalence Classes: Equivalence classes are all of the elements in a given set A which bear an equivalence relation to a chosen element x. This is denoted $[x]$.

$$[x] = \{a \in A : aRx\}$$

Partitions

A partition of a set A is a **set** $\mathcal{P}$, of non-empty subsets of A, such that:

- If $x \in A$, then there is $X \in \mathcal{P}$ with $x \in X$.
- If $X, Y \in \mathcal{P}$, then either $X \cap Y = \emptyset$ or $X = Y$.

As a sidenote, the set of equivalence classes on a set A forms a set partition.

Integers modulo n: The equivalence relation $\equiv \pmod{n}$ describes the integers that have the same remainder when divided by n. Hence, we can devine interesting properties on these classes, given some $x \in [a]$ and $y \in [b]$, then:

$$x + y \in [a + b] \qquad\qquad xy \in [ab]$$

Often, such relations are written $[r]_n$, for $\equiv r \pmod{n}$.

Functions

A function between two sets, A and B is a non empty subset $f \subseteq A \times B$ such that:

- For every $a \in A$, there exists a $b \in B$ such that $(a, b) \in f$.
  Every input has an output.
- If $(a, b) \in f$ and $(a, c) \in f$, then $b = c$.
  Every input has a distinct output.

Function Definitions: Given a function $f : A \to B$:

- Domain: The domain, in this case, A, is the writer-defined set of valid inputs to the function.
- Codomain: The codomain, in this case, B, is the writer-defined set of valid outputs to the function.
- Range: The range is all of the actual outputs of the function yielded by the domain. We can formalize this by:

  $$\text{rng}(f) = \{b \in B \text{ s.t. } \exists a \in A \text{ s.t. } f(a) = b\}$$

  We note that $\text{rng}(f) \subseteq B$, or the range is a subset of the codomain.
  If we define some additional sets, $C \subseteq A$ and $D \subseteq B$, then we can add some additional definitions:
- Image: The image of C in B is $f(C) = \{f(x) \text{ s.t. } x \in C\}$. In simpler terms, the image is the range of some chosen subset of the domain.
- Preimage: The preimage of D in A are all of the elements in A which correspond to outputs in D. To formalize:

  $$f^{-1}(D) = \{x \in A \text{ s.t. } f(x) \in D\}$$

  It is important to note that the preimage is not the inverse.

Identities in Images and Preimages: I feel like these could come in handy.
Given some $f : A \to B$, and $C_1, C_2 \subseteq A$ and $D_1, D_2 \subseteq B$:

$$f(C_1 \cap C_2) \subseteq f(C_1) \cap f(C_2) \qquad\qquad f(C_1 \cup C_2) = f(C_1) \cup f(C_2)$$
$$f^{-1}(D_1 \cap D_2) = f^{-1}(D_1) \cap f^{-1}(D_2) \qquad\qquad f^{-1}(D_1 \cup D_2) = f^{-1}(D_1) \cup f^{-1}(D_2)$$

Properties of Functions and Function Operations

We can define some important properties on a given function $f : A \to B$, $a, a_1, a_2 \in A$ and $b, b_1, b_2 \in B$:

- Injective: We call a function "injective" if it follows that

$$(a_1 \neq a_2) \implies (f(a_1) \neq f(a_2))$$

  The contrapositive of this statement is often more useful.

- Surjective: We call a function "surjective" if it follows that

$$\forall b \in B, \exists a \in A \text{ s.t. } b = f(a).$$

- Bijective: We call a function "bijective" if it is both injective and surjective.

Function Compositions: Given some functions $f : A \to B$ and $g : B \to C$, we can obtain the composition of f and g as $g \circ f : A \to C$. This can be rewritten as:

$$(g \circ f)(a) = g(f(a)), \forall a \in A.$$

It is important to note that compositions are associative: $h \circ (g \circ f) = (h \circ g) \circ f$.

Compositions also preserve properties: if a f and g are both in/sur/bi-jective then $f \circ g$ is also in/sur/bi-jective.

We can also define the partial converses, which state that:

- If $g \circ f$ is an injection, then f is an injection.
- If $g \circ f$ is a surjection, then g is a surjection.

Proof by Contradiction

Proof by contradiction is a proof method that generally uses the following logic:

1. Assume, to the contrary, a given statemnt is false
2. Imply that some false statement is true.
3. By contradiction, the original assumption must be true, as required.

There are some common ways that "false statements" present themselves:

- Equality Contradictions: $0 = 1$ or something of the like.
- Divisibility Contradictions: $3 \mid 1$ or something of the like.
- Set Inclusion Contradictions: $1 \in \mathbb{I}$ or something of the like.

Inverse Functions

We can also define different inverses of functions, given some functions $f : A \to B$ and $g : B \to A$:

- If $g \circ f = i_A$, we say that g is a left-inverse of f.
- If $f \circ g = i_B$, we say that g is a right-inverse of f.
- If g is both a left-inverse and a right-inverse of f, then we can call g an inverse of f.

Note: i_A denotes the original element in A given to some function when passed through the function.

Inverses have nice relations to function properties, given some $f : A \to B$:

- f has a left-inverse iff f is injective.
- f has a right-inverse iff f is surjective.
- f has an inverse iff f is bijective.
- If f has a left-inverse, g, and a right-inverse, h, then $g = h$.

Cardinality of Sets

We can also discuss the size of sets, or their cardinalities.

As a very important result, we know that for two sets A and B, we can state that they have the same cardinality if $A = B = \emptyset$, or if there exists a bijection from A to B.

The properties of functions have important implications for cardinality. Given some function $f : A \to B$, then

- If f is injective, then $|A| \leq |B|$.
- If f is surjective, then $|A| \geq |B|$.
- If f is bijective, then $|A| = |B|$.

Denumerable Sets: We call a set B denumerable if we can list out its elements. Hence, we can state that there exists a bijection $f : \mathbb{N} \to B$. Hence, the list has some nice properties:

- The list does not repeat (f injective).
- Every entry appears at some finite position (f surjective).

From this, we can draw some important conclusions:

- If B is denumerable, then $A \subseteq B$ implies A is denumerable.
- If A and B are denumerable, then $A \cap B$ and $A \cup B$ are denumerable.
- If A and B are denumerable, then $A \times B$ is denumerable.
- The sets $\mathbb{N}, \mathbb{Z}, \mathbb{Q}$ are all denumerable.
- $|A| < |\mathcal{P}(A)|$

It is also known that the reals are uncountable.