Notes on Discrete Mathematics

Sameer Rahmani

March 25, 2021

Contents

1 Language of Mathematics

1.1 Types of formal mathematical statements:

- Universal statements: These types of sentences usually starts with `for every` or `all` and state a fact. For example: **All** positive numbers are greater than zero
- Conditional statements: The type of sentences state two conditions and state that if the the first condition is true the second one is true as well. For example: `if 387 is divisible by 18, 387 is divisible by 6`
- Existential statements: Given a property that **may or may not** be true, then there is at least one thing true for which the property is true. For example: The is a prime number a which is even.

1.2 Compound statements:

Usually these statements either are fully explicit or one the statement types are implicit for example a **universal conditional statement** might be explicitly universal and conditional, or it might be explicitly conditional but implicitly universal and vice versa.

- Universal Conditional statements:
  - `For every` animal a, If a is a dog, a is a mammal
  - `All` dogs are mammals
- Universal Existential statements: It's an existential statement which the property of it is a true universal statement. For example: `For every` real number X, there is a an additive reverse[1] R.
- Existential Universal statements: In this type of statements the first part states existence of an object and the second part states that that object satisfies a certain property. For example: There is a positive integer *m*, that `for every` positive integer *n*, *m* <= *n*

Some of the most important mathematical concepts can only be defined using all three types of statement (universal, conditional and existential).

1.3 Sets

The term *set* coined By George Cantor on 1879.

Set Roaster Notation

We define the elements of a set by using the set roaster syntax which is simply put a the elements in a braces separated by a comma. For very large sets we can use ellipsis.

[1] Additive reverse of a is b which defines like $a + b = 0$

The **axiom of extension** says a set is determined completely by it's elements no matter what the order is or whether some of them appear more than once or not.

The set of real numbers ($\mathbb{R}$) is a set of numbers on a line (which is called **continuous**) that number 0 divides the line into positive numbers and negative numbers. Zero itself is called *origin* and numbers to the right of it are positive real numbers and to the left are negative real numbers. Real numbers are on continuous in compare to the set of integer numbers ($\mathbb{Z}$) which are **discrete** and are not continuous. For example in the line of continuous (real numbers) there's no hole but on integers are just fixed point separated by big holes.

Discrete mathematics comes from the distinction between continuous and discrete numbers.

Set Builder Notation

Another way of defining a new set is via the builder notation. Give S is a set and $P(x)$ is a property which may or may not hold true for elements of S. We can build a new set from S which property P holds true for all the members of S. So:

$$S' = x \in S | P(x)$$

S' is a new set that contains all the elements of S such that (| reads as "such that") $P(x)$ is true. The whole thing reads as, **the set of all elements** x **in** S**, such that** $P(x)$ **is true**.
Sometimes people use $x|P(x)$ without mentioning the source set of x which caused some contradiction in the set theory.

1.3.1 Subsets

Subset is a basic relation on sets. If A and B are sets, then A is said to be a subset of B if and only if all the elements of A are also elements of B. It writes as $A \subseteq B$ which means for every element x, if $x \in A$ then $x \in B$.

Alternative way of reading subsets are:

- A is contained in B.
- B contains A.

Obviously $A \nsubseteq B$ means that there is an x such that $x \in A$ and $x \notin B$ (there is at least on element that is in A and not in B).

A is a **Proper subset** of B if there is an element in B which is not in A.

1.3.2 Cartesian Product

To begin order to sets we can create an **ordered pair** of elements of usings sets like $\{\{a\}, \{a, b\}\}$ which is like element a is the first element (Since it is in a set alone) and b is the second element since it only exists in a set with a which was the first element. if $a = b$ then the ordered pair for be $\{\{a\}, \{a, a\}\}$ which is $\{\{a\}\}$.

The ordered pair can be simplifes syntatically to (a, b). Two ordered pair are equal if and only if the first element of both are equal and the second elements are equal.

Using the notation of the **ordered pair** we can get to the notation of *n-tuple*. Let $n \in \mathbb{Z}^+$ and x_1 through x_n (not necessarily distinct) elements of a set, then $(x_1, x_2, ..., x_n)$ is a *n*-tuple which x_1 is the first element and x_n the nth element.

A *2-tuple* is an **ordered pair** and a *3-tuple* is an **ordered triple**.

Two *n*-typles $(x_1, x_2, ..., x_n)$ and $(y_1, y_2, ..., y_n)$ are **equal** if and only if:

$$x_1 = y_1, x_2 = y_2 ... x_n = y_n$$

For the given sets $A_1, A_2, ..., A_n$ the **Cartesian Product** of $A_1, A_2, ..., A_n$, denoted $A_1 \times A_2 \times, ... \times A_n$ is a set of all ordered *n*-tuples $(a_1, a_2, ..., a_n)$ where $a_1 \in A_1, a_2 \in A_2, ..., a_n \in A_n$ which symbolically will be:

$$A_1 \bigtimes A_2, ... \bigtimes A_n = \{(a_1, a_2, ..., a_n) | a_1 \in A_1, a_2 \in A_2, ..., a_n \in A_n\}$$

In particular:

$$A_1 \bigtimes A_2 = \{(a_1, a_2) | a_1 \in A_1, a_2 \in A_2\}$$

is the Cartesian product of A_1 and A_2.

For example the Cartesian product of $A = x, y$ and $B = 1, 2, 3$ is:

$$A \bigtimes B = \{(x, 1), (x, 2), (x, 3), (y, 1), (y, 2), (y, 3)\}$$

We need to differentiate between $(A \times B) \times C$ and $A \times B \times C$. In the first case the Cartesian product is an order pair which the first element itself is again an ordered pair (in a lisp sense `((x . y) z)`). But in the second case the result is an ordered triple.

A **String** is a *n*-tuple of elements of a finit set A that are written with no paranthesis and comma. n is a nonnegative integer. Elements in this form are called **Characters**. A **null string** over A is a string with no character which often denoted as λ. A **bit string** is a string over $A = 0, 1$.

n-tuples are **ordered**.

1.3.3 Relations

If A and B are sets, then a **relation R from A to B** is defined as a subset of $A \times B$ where for every (x, y) in $A \times B$, x **is related to** y **by R** if and only if (x, y) is in **R**. A is called the **domain** and B is called the **co-domain**. It symbolically writes as xRy means that $(x, y) \in R$

NOTE: If we call a relation **H** for example it will write as xHy.

1.3.4 Functions

In terms of sets, A function is a relation with specific properties. Given a function F is a relation from A to B with it's domain being A and its co-domain being B that satisfies the following properties:

1. For every element x in A and y in B, $(x, y) \in F$
2. For all elements x in A and y, z in B if $(x, y) \in F$ and $(x, z) \in F$ then $y = z$.

Basically it has to be deterministic.

If f and g are functions (relations with special properties) from A to B we can write:

$$f = \{(x, y) \in A \bigtimes B \mid y = f(x)\}$$
$$g = \{(x, y) \in A \bigtimes B \mid y = g(x)\}$$

Function equality

f is equal to g, written as $f = g$ if and only if, $f(x) = g(x)$ for every x in A.

1.4 Graphs

A graph G consists of two finite sets: a nonempty set $V(G)$ of vertices and a set $E(G)$ of edges, where each edge is associated with a set consisting of either one or two vertices called its **endpoints**. The correspondence from edges to endpoints is called the **edge-endpoint function**. An edge with just one endpoint is called a **loop**, and two or more distinct edges with the same set of endpoints are said to be **parallel**. An edge is said to **connect** its endpoints; two vertices that are connected by an edge are called **adjacent**; and a vertex that is an endpoint of a loop is said to be **adjacent to itself**. An edge is said to be **incident** on each of its endpoints, and two edges incident on the same endpoint are called **adjacent**. A vertex on which no edges are incident is called **isolated**.

A **directed graph**, or **digraph**, consists of two finite sets: a nonempty set $V(G)$ of vertices and a set $D(G)$ of **directed edges**, where each is associated with an **ordered** pair of vertices called its endpoints. If edge e is associated with the pair (y, w) of vertices, then e is said to be the (directed) edge from y to w.

Let G be a graph and y a vertex of G. The degree of v, denoted $deg(v)$, equals the number of edges that are incident on y, **with an edge that is a loop counted twice**.

2 THE LOGIC OF COMPOUND STATEMENTS

An argument is a sequence of statements aimed at demonstrating the truth of an assertion. The assertion at the end of the sequence is called the **conclusion**, and the preceding state- ments are called **premises**. To have confidence in the conclusion that you draw from an argument, you must be sure that **the premises are acceptable on their own merits or follow from other statements that are known to be true**.

In logic, the form of an argument is distinguished from its content. Logical analysis won't help you determine the intrinsic merit of an argument's content, but it will help you analyze an argument's form to determine whether the truth of the conclusion follows necessarily from the truth of the premises. For this reason logic is sometimes defined as the science of necessary inference or the science of reasoning.

2.1 Statements

In any mathematical theory, new terms are defined by using those that have been previously defined. However, this process has to start somewhere. A few initial terms necessarily remain undefined. In logic, the words **sentence**, **true**, and **false** are the initial undefined terms.

Definition

A **statement** (or **proposition**) is a sentence that is true or false but not both.

2.2 Compound Statements

- $\neg p$ means "Not p"
- $p \wedge q$ means "p and q", a **conjuction**.
- $p \vee q$ means "p or q", a **disjunction**.

Definition

A **statement form** (or **propositional form**) is an expression made up of statement variables (such as p, q, and r) and logical connectives (such as $\vee$, $\wedge$, and $\neg$) that becomes a statement when actual statements are substituted for the component state- ment variables. The truth table for a given statement form displays the truth values that correspond to all possible combinations of truth values for its component state- ment variables.

2.3 Logical Equivalence

For each combination of truth values for p and q, $p \wedge q$ is true when, and only when, $q \wedge p$ is true. In such a case, the statement forms are called logically equivalent, and we say that p and q are **logically equivalent** statements.

Definition

Two statement forms are called **logically equivalent** if, and only if, they have identical truth values for each possible substitution of statemen1ts for their statement variables. The logical equivalence of statement forms P and Q is denoted by writing $P \equiv Q$. Two statements are called logically equivalent if, and only if, they have logi- cally equivalent forms when identical component statement variables are used to replace identical component statements.

2.3.1 Testing Whether Two Statement Forms P and Q Are Logically Equivalent

1. Construct a truth table with one column for the truth values of P and another column for the truth values of Q.
2. Check each combination of truth values of the statement variables to see whether the truth value of P is the same as the truth value of Q.
   - If in each row the truth value of P is the same as the truth value of Q, then P and Q are logically equivalent.
   - If in some row P has a different truth value from Q, then P and Q are not logically equivalent.

2.3.2 De Morgan's law

Symbolically:

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

2.3.3 Tautologies and Contradictions

A **tautology** is a statement form that is always true regardless of the truth values of the individual statements substituted for its statement variables. A statement whose form is a tautology is a **tautological statement**.

A **contradication** is a statement form that is always false regardless of the truth values of the individual statements substituted for its statement variables. A statement whose form is a contradiction is a **contradictory statement**.

2.3.4 Logical Equivalences

Given any statement variables p, q, and r, a tautology t and a contradiction c, the following logical equivalences hold:

1. Commutative laws

$$p \wedge q \equiv q \wedge p$$
$$p \vee q \equiv q \vee p$$

2. Associative laws:

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

3. Distributive laws:

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

4. Identity laws:

$$p \wedge t \equiv p$$
$$p \vee c \equiv p$$

5. Negation laws:

$$p \vee \neg p \equiv t$$
$$p \wedge \neg p \equiv c$$

6. Double negative law:

$$\neg(\neg p) \equiv p$$

7. Idempotent laws:

$$p \wedge p \equiv p$$
$$p \vee p \equiv p$$

8. Universal bound laws:

$$p \vee t \equiv t$$
$$p \wedge c \equiv c$$

9. De Morgan's laws:

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

10. Absorption laws:

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

2.4 Conditional Statements

Definition

If p and q are statement variables, the conditional of q by p is "If p then q" or "p implies q" and is denoted $p \to q$. It is false when p is true and q is false; other- wise it is true. We call p the **hypothesis** (or **antecedent**) of the conditional and q the **conclusion** (or **consequent**).

Order of operations is: $\neg$ and then $\wedge$ and $\vee$ and finally $\to$.

- fact: $p \vee q \to r \equiv (p \to r) \wedge (q \to r)$
- fact: $p \to q \equiv \neg p \vee q$

A conditional statement is only false if and only if the hypothesis is true and the conclusion is false. The negation of a conditional statement is logically equivalent to p (hypothesis) and not q (conclusion) ($\neg(p \to q) \equiv p \wedge \neg q$).

2.4.1 The Contrapositive of a Conditional Statement

If $L = p \to q$ then the cotrapositive of L would be $\neg q \to \neg p$.

Definition

One of the most fundamental laws of logic is the equivalence between a conditional statement and its contrapositive.

The **inverse** of L is $\neg p \to \neg q$ which is **NOT** logically equivalent to L.
The **converse** of L is $q \to p$ which is **NOT** logically equivalent to L.

2.4.2 Only If and the Biconditional

If p and q are statements, p only if q means "if not q then not p," or, equivalently, "if p then q."

p only if q DOES NOT MEAN p if q. Note that it is possible for "p only if q" to be true at the same time that "p if q" is false. For instance, to say that John will break the world's record only if he runs the mile in under four minutes does not mean that John will break the world's record if he runs the mile in under four minutes. His time could be under four minutes but still not be fast enough to break the record.

Definition

Given statement variables p and q, the biconditional of p and q is **"p if, and only if, q"** and is denoted $p \iff q$. It is true if both p and q have the same truth values and is false if p and q have opposite truth values. The words if and only if are sometimes abbreviated **iff**. So it would be true if p and q are both true or both are false and it would false otherwise.

iff has the same priority as **if** in the order of operations.
$p \iff q \equiv (p \to q) \wedge (q \to p)$

2.4.3 Necessary and Sufficient Conditions

Definition

If r and s are statements:
r is a **sufficient condition** for s, means "if r then s."
r is a **necessary condition** for s, means "if not r then not s."

In other words, to say "r is a sufficient condition for s" means that the occurrence of r is sufficient to **guarantee** the occurrence of s. On the other hand, to say "r is a necessary condition for s" means that if r does not occur, then s cannot occur either (and occurance of r does not guarantee the occurance of s): The occurrence of r is necessary to obtain the occurrence of s. Note that because of the equivalence between a statement and its contrapositive, r **is a necessary condition for** s **also means "if** s **then** r**."**

Consequently,

r is a **necessary and sufficient** condition for s means "r if, and only if, s."

2.5 Valid and Invalid Arguments

Definition

An argument is a sequence of statements, and an **argument form** is a sequence of statement forms. All statements in an argument and all statement forms in an argument form, except for the final one, are called **premises** (or **assumptions** or **hypotheses**). The final statement or statement form is called the conclusion. The symbol [, which is read "therefore," is normally placed just before the conclusion.
To say that an argument form is valid means that no matter what particular statements are substituted for the statement variables in its premises, if the resulting premises are all true, then the conclusion is also true. To say that an argument is valid means that its form is **valid**.

The crucial fact about a valid argument is that the truth of its conclusion follows neces- sarily or *inescapably* or by *logical form* alone from the truth of its premises. It is impos- sible to have a valid argument with all true premises and a false conclusion. When an argument is valid and its premises are true, the truth of the conclusion is said to be *inferred* or *deduced* from the truth of the premises. If a conclusion "ain't necessarily so," then it isn't a valid deduction.

2.5.1 Testing an Argument Form for Validity

1. Identify the premises and conclusion of the argument form.
2. Construct a truth table showing the truth values of all the premises and the conclusion.
3. A row of the truth table in which all the premises are true is called a critical row. If there is a critical row in which the conclusion is false, then it is

possible for an argument of the given form to have true premises and a false conclusion, and so the argument form is invalid. If the conclusion in every critical row is true, then the argument form is valid.

Caution! If at least one premise of an argument is false, then we have no information about the conclusion: It might be true or it might be false.

2.5.2 Modus Ponens and Modus Tollens

An argument form consisting of two premises and a conclusion is called a **syllogism**. The first and second premises are called the **major premise** and **minor premise**, respectively. The most famous form of syllogism in logic is called *modus ponens*. It has the following form:

$$\begin{array}{l} \text{if } p \text{ then } q \\ p \\ \therefore q \end{array}$$

Here is an example of modus tollens:

$$\begin{array}{l} \text{If Zeus is human, then Zeus is mortal.} \\ \text{Zeus is not mortal.} \\ \therefore \text{Zeus is not human.} \end{array}$$

modus tollens is another **syllogism** which is quite useful. It goes like:

$$\begin{array}{l} \text{if } p \text{ then } q \\ \neg q \\ \therefore \neg p \end{array}$$

We can prove *modus tollens* either via a truth table or via contrapositive law. *modus ponens* and *modus tollens* are equally important in mathematics.

A **rule of inference** is a form of argument that is valid. Thus modus ponens and modus tollens are both rules of inference.

1. Generalization

   The following forms are valid:

$$\begin{array}{l} p \\ \therefore p \vee q \end{array}$$

$$\begin{array}{l} q \\ \therefore p \vee q \end{array}$$

2. Specialization

   The following forms are valid:

$$\begin{array}{l} p \wedge q \\ \therefore p \end{array}$$

$$\begin{array}{l} p \wedge q \\ \therefore q \end{array}$$

   Specialization is usually used when we want to categorize some stuff based on a property. Both Generalization and Specialization are used frequently in mathematics to tailor facts to fit into hypotheses of known theorems in order to draw further conclusions. Elimination, transitivity, and proof by division into cases are also widely used tools.

3. Elimination

   The following argument forms are valid:

$$\begin{array}{ll} p \vee q & \\ \neg q & \therefore p \end{array}$$

$$\begin{array}{ll} p \vee q & \\ \neg p & \therefore q \end{array}$$

4. Transitivity

   The following argument forms are valid:

$$\begin{array}{l} p \rightarrow q \\ q \rightarrow r \\ \therefore p \rightarrow r \end{array}$$

5. Proof by Division into Cases

   The following argument form is valid:

$$\begin{array}{l} p \vee q \\ p \rightarrow r \\ q \rightarrow r \\ \therefore r \end{array}$$

2.5.3 Fallacies

A **fallacy** is an error in reasoning that results in an invalid argument. Three common fallacies are using ambiguous premises, and treating them as if they were unambiguous, circular reasoning (assuming what is to be proved without having derived it from the premises), and jumping to a conclusion (without adequate grounds).

For an argument to be valid, every argument of the same form whose premises are all true must have a true conclusion. It follows that for an argument to be invalid means that there is an argument of that form whose premises are all true and whose conclusion is false.

1. Converse error The following form is not valid:

$$\begin{array}{l} p \to q \\ q \\ \therefore p \end{array}$$

The fallacy underlying this invalid argument form is called the **converse error** because the conclusion of the argument would follow from the premises if the premise p S q were replaced by its converse. Such a replacement is not allowed, however, because a conditional statement is not logically equivalent to its converse. Converse error is also known as the **fallacy of affirming the consequent**.

2. Inverse error The following form is not valid:

$$\begin{array}{l} p \to q \\ \neg p \\ \therefore \neg q \end{array}$$

The fallacy underlying this invalid argument form is called the inverse error because the conclusion of the argument would follow from the premises if the premise p S q were replaced by its inverse. Such a replacement is not allowed, however, because a conditional statement is not logically equivalent to its inverse. Inverse error is also known as the **fallacy of denying the antecedent**.

Note that the validity and thruthy of an argument are two different concepts. An argument can be valid but it might have a false premise and a false conclusion and also an argument can be invalid despite the fact the it's premises and conclusion hold truthy.

Definition

An argument is called **sound** if, and only if, it is valid and all its premises are true. An argument that is not sound is called **unsound**.

2.5.4 Contradictions and Valid Arguments

Contradiction Rule

If you can show that the supposition that statement p is false leads logically to a contradiction, then you can conclude that p is true.

The contradiction rule is the logical heart of the method of proof by contradiction. A slight variation also provides the basis for solving many logical puzzles by eliminating contradic- tory answers: **If an assumption leads to a contradiction, then that assumption must be false.**

2.5.5 Summary

- Modus Ponens:

$$\begin{array}{l} p \to q \\ p \\ \therefore q \end{array}$$

- Modus Tollens:

$$\begin{array}{l} p \to q \\ \neg q \\ \therefore \neg p \end{array}$$

- Generalization:

$$\begin{array}{l} p \\ \therefore p \vee q \end{array}$$

$$\begin{array}{l} q \\ \therefore p \vee q \end{array}$$

- Specialization:

$$\begin{array}{l} p \wedge q \\ \therefore p \end{array}$$

$$\begin{array}{l} p \wedge q \\ \therefore q \end{array}$$

- Conjunction:
- Elimination:
- Transitivity:
- Proof by Division into Cases:
- Contradiction Rule:

2.6 Logical circuits

- NAND: $P|Q \equiv \neg(P \wedge Q)$ (Sheffer's strock)
- NOR: $P \downarrow Q \equiv \neg(P \vee Q)$ (Pierce arrow)

3 The logic of quantified statements

The symbolic analysis of predicates and quantified statements is called the **predicate calculus**. For example analyzing the meaning of the word "all" or "some" in a statement. The symbolic analysis of ordinary compound statements (like normal logical statement. e.g $p \wedge q$) is called the **statement calculus** (or the **propositional calculus**).

In logic, predicates can be obtained by removing some or all of the nouns from a statement. For instance, let P stand for "is a student at Bedford College" and let Q stand for "is a student at." Then both P and Q are predicate symbols. The sentences "x is a student at Bedford College" and "x is a student at y" are symbolized as P(x) and as Q(x, y), respectively, where x and y are *predicate variables* that take values in appropriate sets. When concrete values are substituted in place of predicate variables, a statement results. For simplicity, we define a predicate to be a predicate symbol together with suitable predicate variables. In some other treatments of logic, such objects are referred to as **propositional functions** or **open sentences**.

Definition

A **predicate** is a sentence that contains a finite number of variables and becomes a **statement** when specific values are substituted for the variables. The **domain** of a predicate variable is the set of all values that may be substituted in place of the variable.

Definition

If $P(x)$ is a predicate and x has domain D, the **truth set** of $P(x)$ is the set of all elements of D that make $P(x)$ true when they are substituted for x. The truth set of $P(x)$ is denoted.

$$\{x \in D | P(x)\}$$

3.1 The Universal Quantifier:

Definition

Let $Q(x)$ be a predicate and D the domain of x. A **universal statement** is a statement of the form "$\forall$ x $\in$ D, Q(x).$" It is defined to be true if, and only if, $Q(x)$ is true for each individual x in D. It is defined to be false if, and only if, $Q(x)$ is false for at least one x in D. A value for x for which Q(x) is false is called a **counterexample** to the universal statement.

The **method of exhaustion** is a technique to find the truthy value of a universal statement by finding the truth for every single element in the Domain. Obviously it works only for finite sets.

3.2 The Existential Quantifier:

The symbol ∃ denotes "there exists" and is called the **existential quantifier**.

Definition

Let $Q(x)$ be a predicate and D the domain of x. An existential statement is a statement of the form "$∃ x ∈ D such that Q(x).$" It is defined to be true if, and only if, $Q(x)$ is true for at least one x in D. It is false if, and only if, $Q(x)$ is false for all x in D.

3.3 Universal Conditional Statements

A reasonable argument can be made that the most important form of statement in math- ematics is **the universal conditional statement**:

$$\forall x, if P(x) then Q(x)$$

The definition of valid argument is a universal conditional statement:

For every combination of truth values for the component statements, if the premises are all true then the conclusion is also true.

3.4 Bound Variables and Scope

Assume the following:

$$\forall x, x^2 > x$$

We say that the variable x is bound by the **quantifier** that controls it and that its **scope** begins when the quantifier introduces it and ends at the end of the quantified statement.

3.5 Implicit qualification

We don't use explicit qualifications all the time, in fact most of the time we use implicit qualifications for example:

- If a number is an integer, then it is a rational number: The article "a" is implicitly implies "for every"
- The number 24 can be written as a sum of two even integers: Can be expressed as "∃ even integers m and n such that 24 = m + n"
- $(x + 1)^2 = x^2 + 2x + 1$: Can be expressed as "∀ real number x, $(x + 1)^2 = x^2 + 2x + 1$"
- Solve $5x - 1 = 2$: Is like prove that "∃ a real number x, such that %5x - 1 = 2$"

Definition

Let $P(x)$ and $Q(x)$ be predicates and suppose the common domain of x is D.

- The notation $P(x) \implies Q(x)$ means that every element in the truth set of $P(x)$ is in the truth set of $Q(x)$, or, equivalently, $\forall x, P(x) \to Q(x)$.
- The notation $P(x) \iff Q(x)$ means that $P(x)$ and $Q(x)$ have identical truth sets, or, equivalently, $\forall x, P(x) \to Q(x)$.

The quantification of a statement—whether universal or existential—crucially determines both how the statement can be applied and what method must be used to establish its truth. Thus it is important to be alert to the presence of hidden quantifiers when you read mathematics so that you will interpret statements in a logically correct way.

3.6 Negations of Quantified Statements

3.6.1 Negation of a universal statement

The negation of a statement of the form

$$\forall x \in D, Q(x)$$

is logically equivalent to a statement of the form

$$\exists x \in D, Q(x).$$

Symbolically,

$$\neg(\forall x \in D, Q(x)) \equiv \exists x \in D, \neg Q(x).$$

Thus **The negation of a universal statement ("all are") is logically equivalent to an existential statement ("some are not" or "there is at least one that is not").**

3.6.2 Negation of an Existential Statement

The negation of a statement of the form

$$\exists x \in D, Q(x)$$

is logically equivalent to a statement of the form

$$\forall x \in D, \neg Q(x).$$

Symbolically,

$$\neg(\exists x \in D, Q(x)) \equiv \forall x \in D, \neg Q(x).$$

Thus **The negation of an existential statement ("some are") is logically equivalent to a universal statement ("none are" or "all are not").**

3.6.3 Negations of Universal Conditional Statements

By definition of the negation of a for all statement:

$$\neg(\forall x, P(x) \to Q(x)) \equiv \exists x, \neg(P(x) \to Q(x)).$$

We can negate the conditional statement on the right side and rewrite the equation like:

$$\neg(\forall x, P(x) \to Q(x)) \equiv \exists x, P(x) \land \neg Q(x)).$$

3.6.4 The relation among $\forall, \exists, \land, \lor$

According to the De Morgan's law, the negation of $\land$ is $\lor$ and the negation for $\lor$ is $\land$, just like the relationship between $\forall$ and $\exists$. In fact we can write $\forall$ as a sequence of $\land$ statements using predicates on each element of the domain set. Similarly we can write $\exists$ as a sequence of $\lor$ statements using the predicate function on all the elements of the domain.

So (if D is a domain set like $\1, 2, 3}):

$$\forall x \in D, P(x) \equiv P(1) \land P(2) \land P(3)$$
$$\exists x \in D, P(x) \equiv P(1) \lor P(2) \lor P(3)$$

3.6.5 Vacuous Truth of Universal Statements

In general a statement in the form of $\forall x \in D, P(x) \to Q(x)$ is **vacuously true** or **true by default** if, and only if $P(x)$ is false for every x in D.

In mathematics, the words in general signal that what is to follow is a generalization of some aspect of the example that always holds true.

When we're trying to find the truth about a statement which might sound ambigious we can rely on it's negation. If the negation is false then the statement has to be true

3.6.6 Other Variants of Universial Conditional Statements

Definition

Consider the following statement: $\forall x \in D, P(x) \to Q(x)$.

1. **Contrapositive**: $\forall x \in D, \neg Q(x) \to \neg P(x)$.
2. **Converse**: $\forall x \in D, Q(x) \to P(x)$
3. **Inverse**: $\forall x \in D, \neg P(x) \to \neg Q(x)$

According to the rules of conditionals:

$$\forall x \in D, P(x) \to Q(x) \equiv \forall x \in D, \neg Q(x) \to \neg P(x)$$
$$\forall x \in D, P(x) \to Q(x) \not\equiv \forall x \in D, Q(x) \to P(x)$$
$$\forall x \in D, P(x) \to Q(x) \not\equiv \forall x \in D, \neg P(x) \to \neg Q(x)$$

3.6.7 Necessary and Sufficient Conditions, Only If

The definitions of necessary, sufficient, and only if can also be extended to apply to universal conditional statements.

Definition

- "$\forall$ x, r(x) is a **sufficient condition** for s(x)" means:

$$\forall x, r(x) \rightarrow s(x)$$

- "$\forall$ x, r(x) is a **necessary condition** for s(x)" means:

$$\forall x, \neg r(x) \rightarrow \neg s(x)$$

  or, equivalently:

$$\forall x, s(x) \rightarrow r(x)$$

- "$\forall$ x, r(x) only if s(x)" means

$$\forall x, \neg s(x) \rightarrow \neg r(x)$$

  or, equivalently:

$$\forall x, r(x) \rightarrow s(x).$$

3.7 Statements With Multiple Quantifiers

We apply the quantifiers logic based on their order of appearance.For example consider the following statement:

$$\forall x \in D, \exists y \in G, P(x, y)$$

In order to find the truth about the statement above you need to pick an element from set D and call it x, then find an element in G and call it y such that predicate P holds true with those two. You can only pick y after you picked x.

Since the we need to apply the quantifiers based on their order, we need to keep in mind that **the order of quantifiers significantly changes the meaning of the statements**.

If a quantifier immediately follows another quantifier of the same type (two $\forall$ for example), then the order of them doesn't matter. For example, we can say "for every" integer x and "for every" integer y $x.y = y.x$. The order of quantifiers of this statement is not important since both quantifiers are the same we can simplify the statement to: "For every" integer x **and** y, $x.y = y.x$.

3.8 Negations of Statements with Multiple Quantifier

In order to negate a statement with several quantifiers we can break it down to smaller pieces. For example consider the following statement:

$$\forall x \in D, \exists y \in G, \text{ such that } Q(x, y)$$

we can rewrite it as (This is not a correct form but it will give you an idea):

$$\forall x \in D, P(x)$$

which $P(x)$ would be $\exists y \in G, Q(x, y)$, so the negation of it would be:

$$\exists x \in D, \text{ such that } \neg P(x)$$

so by replacing the $P(x)$ with its original form we will have:

$$\exists x \in D, \text{ such that } \neg(\exists y \in G, \text{ such that } Q(x, y)) \equiv \exists x \in D, \text{ such that } \forall y \in G, \neg Q(x, y)$$

3.9 Formal Notation

From now on we will use the formal logic notation. In formal notation instead of using words we use logic notations and predicates to write statements. For example "Every bird has wings" can be written as:

$$\forall x(Bird(x) \rightarrow hasWings(x))$$

"Bird(x)" is a predicate function that is true when the given x is a bird and "hasWings(x)" is a predicate that is true if x has wings. We can read this statement with conditionals like "For every x, if x is a bird then it has wings" (note the conditional notation $\rightarrow$).

For existential quantifiers we can follow the same pattern. Consider the statement "This is a bird such that it is blue", we can write this statement in formal notation like:

$$\exists x(Bird \wedge Blue(x))$$

For statements with multiple quantifiers, we just have to follow what we have learn up until now. Here is an example:

$$\forall x(Integer(x) \rightarrow \exists y(Integer(y) \wedge (x + y = 100)))$$

We can extract $x + y = 100$ to a predicate and use the predicate instead. To be more clear the general form of quantified statements in formal notations goes as follows:

$$\forall x(A \rightarrow B)$$
$$\exists x(A \wedge B)$$

which A and B can be any statement.

By putting everything that we have learned up until now we can have a language to reason about different things. This new language is called **the language of first order logic**.

3.10 Arguments with Quantified Statements

Universal instantiation

If a property is true of *everything* in a set, then it is true of *any particular* thing in the set.

Universal instantiation is *the* most important and fundamental tool of deductive reasoning. With it, we can indicate that the truth of particular case follows as a special case of a general or **universal** truth. In simpler terms, in order to find the truth of a special case or an statement we can rely on the general case of it. For example, a theorem says that A and B are true in a certain condition and for all the things in a certain type T. In order to find the truth about a related statement about a specific object of T, all we have to do is to rely on **universal instantiation** and show that the object in question is in T. Also A and B is true for that object. Simply to conclude that the object is a special case of a general case which is the theorem.

3.10.1 Univeral Modus Ponens

We can combine universal instantiation and modus ponens together we can create a new form of valid arguments which is called **universal modus ponens**.

Universal Modus Ponens

Formally:

$$\begin{aligned} &\forall x(P(x) \to Q(x)) \\ &P(y) \text{ for a particular } y \\ &\therefore Q(y) \end{aligned}$$

We can read the universial modus ponens as, For every x that makes $P(x)$ true, then it makes $Q(x)$ true as well (If $P(x)$ then $Q(x)$). y makes $P(y)$ true, therefore it makes $Q(y)$ true as well.

The conclusion about y making $Q(y)$ true comes from the **universal instantiation**. y is a particular case of a general case.

3.10.2 Universal Modus Tollens

Modus tollens which is one the most important tools of proof of contradiction, is a combination of universal instantitation and modus tollens.

Universal Modus Tollens

Formally:

$$\forall x(P(x) \rightarrow Q(x))$$
$$\neg Q(y) \text{ for a particular } y$$
$$\therefore \neg P(y)$$

We can read the universial modus tollens as, For every x that makes $P(x)$ true, then it makes $Q(x)$ true as well (If $P(x)$ then $Q(x)$). y doesn't make $Q(y)$ true, therefore $P(y)$ must not be true.

3.10.3 Proving Validity of Arguments with Quantified Statements

The intuitive definition of validity for arguments with quantified statements is the same as for arguments with compound statements. An argument is valid if, and only if, the truth of its conclusion follows necessarily from the truth of its premises. The formal definition is as follows:

Definition

To say that an argument form is **valid** means the following: No matter what particular predicates are substituted for the predicate symbols in its premises, if the resulting premise statements are all true, then the conclusion is also true. An argument is called valid if, and only if, its form is valid. It is called **sound** if, and only if, its form is valid and its premises are true.

1. Converse and Inverse Errors Validating an argument can be tricky. We might end of **exhabiting** a **converse** or **inverse** error. For example consider the following:

   All human beings are mortal
   Jim is mortal
   $\therefore$ Jim is human

   The above argument is **invalid**. Since we can write the major premis as:

   $$\forall x(Human(x) \rightarrow Mortal(x))$$

   and the argument as:

   $$\forall x(Human(x) \rightarrow Mortal(x))$$
   $$Mortal(Jim)$$
   $$\therefore Human(Jim)$$

Since conditional statement is not logically equivalent to its converse, then the above argument is invalid because we can't drive the conclusion from the minor premis (Mortal(jim)) because it's the converse of the major premis, thus the argument is invalid. If we say this argument is valid, we are **exibiting a converse error**.

Same reasoning applies to the **inverse error as well**. For example:

All human beings are mortal
Lion is not human
$\therefore$ Lion is immortal($\neg Mortal(Lion)$)

That above argument is invalid and we are **exibiting an inverse error**.

Many people usually make converse or inverse errors due to the fact that they confuse the **conditional** statements with **biconditional** statements.

3.10.4 Creating Additional Forms of Argument

Universal modus ponens and modus tollens were obtained by combining universal instan- tiation with modus ponens and modus tollens. In the same way, additional forms of argu- ments involving universally quantified statements can be obtained by combining universal instantiation with other of the valid argument forms given in chapter 2. For example, consider the trasivity law:

$$p \to q$$
$$q \to r$$
$$\therefore p \to r$$

We can combine transivity law or (other laws) with universial instantiation to build new forms of valid arguments.

Universal Transivity

$$\forall x(P(x) \to Q(x))$$
$$\forall x(Q(x) \to R(x))$$
$$\therefore \forall x(P(x) \to R(x))$$

4 ELEMENTARY NUMBER THEORY AND METHODS OF PROOF

One of the best ways to think of a mathematical proof is as a carefully reasoned argument to convince a skeptical listener (often yourself) that a given statement is true. Imagine the listener challenging your reasoning every step of the way, constantly asking, "Why is that so?" If you can counter every possible challenge, then your proof as a whole will be correct.

> **Note**: We assume that there is no integer between 0 and 1 and that the set of all integers is closed under addition, subtraction, and multiplication. This means that sums, differences, and products of integers are integers.

4.1 Constructive Proof of Existence

One way to prove an existential statement such as:

$$\exists x(x \in D \wedge Q(x))$$

is to find an element in the set D which makes $Q(x)$ true. Another way is to give a set of directions to find such an element. Both of these are called **constructive proof of existence** which relies on more fundamental principle called **existential generalization**. According to it, if you know that a certain property is true for a particular object, then you may conclude that "There exist and object for which the property is true".

In the other hand the **nonconstructive proof of existence** is about either showing that the existence of the element x is guaranteed by an axiom or a previously proved theorem or the assumption that **there is no** such element x which the given properties leads to contradiction.

The disadvantage of a nonconstructive proof of existence is it might not give any clue about where and how to find x. This importance of this issue raised by wide spreading computers and advancements in the computer science world.

4.2 Disproving Universal Statements by Counter Example

In order to disprove an universal statement all we need to do is to prove that the negation of the universal statement which is an existential statement is true. For example, in order to disprove the statement $\forall x(P(x) \rightarrow Q(x))$ we need to show that $\exists x(P(x) \wedge \neg Q(x))$ is true by finding x such that $P(x)$ is true but $Q(x)$ is false. We call such a value, the **counter example**.

4.3 Proving Universal Statements

On way of proving universal statements to be true is the **method of exhaustion** which means showing that the statement is true for all the elements in the domain of it. While it seems simple enough but it's impossible to use the method of exhaustion for infinite domains or even domains with large number of elements.

The most powerful technique for proving a universal statement is one that works regardless of the size of the domain over which the statement is quantified. It is based on a logical principle sometimes called *universal generalization*. A more descriptive name is *generalizing from the generic particular*.

Generalizing from the Generic Particular

To show that **every** element of a set satisfies a certain property, suppose x is a **particular** but **arbitrarily** chosen element of the set, and show that x satisfies the property.