IS 5311: Discrete Mathematics

Chapter 1: Introduction to Discrete Mathematics

Lecturer: MS. M.W.S. Randunu *Department of Interdisiplinary Studies,University of Ruhuna.*

1.1 Introduction

Discrete mathematics is a branch of mathematics that deals with countable, distinct, and separate objects rather than continuous ones. It provides the theoretical foundation for various fields, including computer science, cryptography, and operations research. Key areas of discrete mathematics include:

1. Sets and Relations
   Sets theory lays the groundwork for understanding collections of objects and their relationships. Relations describe connections between elements of sets, such as equivalence relations and partial orders.

2. Logic
   Logic forms the basis for reasoning and the study of propositions. Propositional logic deals with the logical relationships between propositions, while predicate logic extends this to handle quantifiers like "for all" ($\forall$) and "there exists" ($\exists$) .

3. Combinatorics
   Combinatorics focuses on counting, arrangements, and combinations of objects. It includes topics such as permutations, combinations, binomial coefficients, and the pigeonhole principle.

4. Graph Theory
   Graph theory studies networks of interconnected nodes (vertices) and edges. It encompasses concepts like paths, cycles, trees, connectivity, and graph coloring.

5. Number Theory
   Number theory explores the properties of integers, including divisibility, prime numbers, congruences, and modular arithmetic. It has significant applications in cryptography.

6. Discrete Probability
   Discrete probability deals with the likelihood of events that have a finite or countably infinite sample space. It includes topics such as probability distributions, random variables, expected value, and conditional probability.

Importance of Discrete Mathematics

1. Foundation for Computer Science
   Discrete mathematics forms the backbone of computer science, providing tools and techniques for analyzing algorithms, designing data structures, and solving computational problems efficiently.

2. Cryptography and Information Security
   Number theory, combinatorics, and discrete probability are essential in designing and analyzing cryptographic algorithms for secure communication and data protection.

3. Networks and Communication Systems
   Graph theory is instrumental in modeling and analyzing networks, such as social networks, communication networks, and transportation networks.

4. Operations Research and Optimization
   Discrete mathematics techniques are used in operations research to optimize processes and make informed decisions in areas like logistics, supply chain management, and resource allocation.

5. Artificial Intelligence and Machine Learning
   Discrete mathematics concepts are foundational to various aspects of artificial intelligence and machine learning, including optimization algorithms, graph-based models, and probabilistic reasoning.

6. Combinatorial Optimization
   Combinatorial optimization problems, such as the traveling salesman problem and the knapsack problem, are pervasive in real-world applications and are tackled using techniques from discrete mathematics.

7. Logic and Reasoning
   Discrete mathematics enhances critical thinking skills by providing tools for precise reasoning, logical argumentation, and problem-solving.

In summary, discrete mathematics plays a crucial role in various disciplines and real-world applications, providing the theoretical framework and analytical tools necessary for solving complex problems efficiently and effectively.

1.2 Foundations for Logic and Proofs

To understand mathematics, we must understand what makes up a correct mathematical argument, that is, a proof. Once we prove a mathematical statement is true, we call it a theorem.A collection of theorems on a topic organize what we know about this topic. To learn a mathematical topic, a person needs to actively construct mathematical arguments on this topic, and not just read exposition. Moreover, knowing the proof of a theorem often makes it possible to modify the result to fit new situations.

Everyone knows that proofs are important throughout mathematics, but many people find it surprising how important proofs are in computer science. In fact, proofs are used to verify that computer programs produce the correct output for all possible input values, to show that algorithms always produce the correct result, to establish the security of a system, and to create artificial intelligence. Furthermore, automated reasoning systems have been created to allow computers to construct their own proofs.

1.2.1 Propositional Logic

Definition 1. *A proposition is a declarative sentence (that is, a sentence that declares a fact) that is either true or false, but not both.*

Example

1. Sri Jayawardenepura Kotte is the capital of Sri Lanka.. **(T)**
2. Los Angeles is the capital of the United States of America. **(F)**
3. 1 + 1 = 2. **(T)**
4. 2 + 2 = 3. **(F)**

Some examples of sentences that are not propositions.

1. What time is it?
2. How are you feeling today?

   These sentences are not propositions because they do not express a statement that is either true or false. Instead, they may be questions, commands, expressions of feelings or opinions, or statements that lack a clear truth value.
3. x + 1 = 2.
4. x + y = z.

   So, while these statements are not propositions in their current form, they can become propositions when specific values are assigned to the variables, allowing for evaluation as true or false.

1.2.1.1 Key Concepts

- Propositional Variables:
  These are variables that represent propositions or statements. Just as letters are used to denote numerical variables, letters like $p, q, r, s, \ldots$ are used to denote propositional variables. Each propositional variable can stand for a statement that can be true or false.
- Truth Values:
  The truth value of a proposition is either true or false. True is typically denoted by T and false by F.These truth values indicate whether a proposition is true or false under a given interpretation or in a specific context.
- Propositional Calculus/Logic: This is the area of logic that deals with propositions, their combinations, and the rules governing their relationships. Propositional logic is concerned with the manipulation and analysis of propositions using logical operators like conjunction, disjunction, and negation.

Many mathematical statements are constructed by combining one or more propositions. New propositions, called **compound propositions**, are formed from existing propositions using logical operators.

1.2.1.2 Logical Operators

Definition 2. *Let p be a proposition. The negation of p, denoted by $\neg p$ (also denoted by $\bar{p}$), is the statement "It is not the case that p." The proposition $\neg p$ is read "**not** p."*

The truth value of the negation of p, , is the opposite of the truth value of p.

Example

1. Proposition: "The sun rises in the east."
   Negation: "It is **not** the case that the sun rises in the east."
   Simpler English: "The sun **does not** rise in the east."
2. Proposition: "The temperature outside is above 25 degrees Celsius."
   Negation: "It is **not** the case that the temperature outside is above 25 degrees Celsius."
   Simpler English: "The temperature outside is **not** above 25 degrees Celsius."

p	$\neg p$
T	F
F	T

Table 1.1: Truth table for the negation of a proposition p.

The negation operator constructs a new proposition from a single existing proposition. We will now introduce the logical operators that are used to form new propositions from two or more existing propositions. These logical operators are also called **connectives**.

Definition 3. *Let p and q be propositions. The conjunction of p and q, denoted by $p \wedge q$, is the proposition "p **and** q."*

Example Consider a scenario of deciding whether to go out for a walk based on weather conditions. We have two conditions:

- A: It's sunny outside.
- B: It's not raining.

We decide to go for a walk only if it's sunny outside **and** it's not raining.

Condition for going out for a walk: $A \wedge B$

p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

Table 1.2: The Truth Table for the Conjunction of Two Propositions.

The conjunction $p \wedge q$ is true when both p and q are true and is false otherwise.

Definition 4. *Let p and q be propositions. The disjunction of p and q, denoted by $p \vee q$, is the proposition "p **or** q."*

Example Referencing back to our earlier example of deciding whether to go out for a walk based on weather conditions.

Here we decide to go for a walk if either it's sunny outside **or** it's not raining (or both).

Condition for going out for a walk: $A \vee B$

p	q	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F

Table 1.3: The Truth Table for the Disjunction of Two Propositions.

The disjunction $p \vee q$ is false when both p and q are false and is true otherwise.

Definition 5. *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.*

Example Again the same scenario of deciding whether to go out for a walk based on weather conditions.

We decide to go for a walk if it's either sunny outside **or** it's not raining, but **not both**. If it's both sunny and raining, we stay inside.

Condition for going out for a walk: $A \oplus B$

p	q	$p \oplus q$
T	T	F
T	F	T
F	T	T
F	F	F

Table 1.4: The Truth Table for the Exclusive Or of Two Propositions

1.2.2 Conditional Statements

Definition 6. *Let p and q be propositions. The conditional statement $p \to q$ is the proposition **"if p, then q."***

p	q	$p \to q$
T	T	T
T	F	F
F	T	T
F	F	T

Table 1.5: The Truth Table for the Conditional Statement $p \to q$.

The conditional statement $p \to q$ is false when p is true and q is false, and true otherwise.
In the conditional statement $p \to q$, p is called the **hypothesis (or antecedent or premise)** and q is

called the **conclusion (or consequence)**.

The statement $p \to q$ is called a conditional statement because $p \to q$ asserts that q is true on the condition that p holds. A conditional statement is also called an **implication**.

Ways to express the conditional statement:
"if p, then q"
"p implies q"
"if p, q"
"p only if q"
"p is sufficient for q"
"a sufficient condition for q is p"
"q if p"
"q whenever p"
"q when p"
"q is necessary for p"
"a necessary condition for p is q"
"q follows from p"
"q unless $\neg p$"
A conditional statement, often referred to as an "if-then" statement, expresses a relationship between two propositions where the truth of one proposition depends on the truth of the other. Here's an example:

Example

1. "**If** it is raining outside, **then** I will bring an umbrella."

   The first part, "if it is raining outside," is the condition or the antecedent.
   The second part, "then I will bring an umbrella," is the consequence or the consequent.

   This statement asserts that if the condition (rain) is true, then the consequence (bringing an umbrella) will also be true. If it's not raining, the statement doesn't specify what will happen, so it's open-ended.

2. "**If** I am elected, **then** I will lower taxes."

3. "**If** you get 100 on the final, **then** you will get an A."

Exercises

1. Let p be the statement "I study hard for exams", and q be the statement "I will get good grades."Express $p \to q$ in English ?

2. What is the value of the variable x after the statement

   $$\text{if } 2 + 2 = 4 \text{ then } x := x + 1$$

   if $x = 0$ before this statement is encountered? (The symbol := stands for assignment. The statement $x := x + 1$ means the assignment of the value of $x + 1$ to x.)

We can form some new conditional statements starting with a conditional statement $p \to q$. In particular, there are three related conditional statements that occur so often that they have special names.

- The proposition $q \to p$ is called the **converse** of $p \to q$.

q	p	$q \to p$
T	T	T
T	F	F
F	T	T
F	F	T

Table 1.6: The Truth Table for the Conditional Statement $q \to p$.

- The **contrapositive** of $p \to q$ is the proposition $\neg q \to \neg p$.

$\neg q$	$\neg p$	$\neg q \to \neg p$
T	T	T
T	F	F
F	T	T
F	F	T

Table 1.7: The Truth Table for the Conditional Statement $\neg q \to \neg p$.

- The proposition $\neg p \to \neg q$ is called the **inverse** of $p \to q$.

$\neg p$	$\neg q$	$\neg p \to \neg q$
T	T	T
T	F	F
F	T	T
F	F	T

Table 1.8: The Truth Table for the Conditional Statement $\neg p \to \neg q$.

We will see that of these three conditional statements formed from $p \to q$, only the contrapositive always has the same truth value as $p \to q$.

When two compound propositions always have the same truth value we call them equivalent, so that a conditional statement and its contrapositive are equivalent. The converse and the inverse of a conditional statement are also equivalent.

Example: What are the contrapositive, the converse, and the inverse of the conditional statement

"The home team wins whenever it is raining."

Solution: Because "q whenever p" is one of the ways to express the conditional statement $p \to q$,

- The original statement can be rewritten as,

  "If it is raining, then the home team wins."

- The contrapositive of this conditional statement is

  "If the home team does not win, then it is not raining."

- The converse is

  "If the home team wins, then it is raining."

- The inverse is

  "If it is not raining, then the home team does not win."

Only the contrapositive is equivalent to the original statement.

1.2.3 Biconditional Statement

Definition 7. *Let p and q be propositions. The* ***biconditional statement*** $p \leftrightarrow q$ *is the proposition "p* ***if and only if*** *q."*

p	q	$p \leftrightarrow q$
T	T	T
T	F	F
F	T	F
F	F	T

Table 1.9: The Truth Table for the Biconditional $p \leftrightarrow q$.

The biconditional statement $p \leftrightarrow q$ is true when p and q have the same truth values, and is false otherwise. Biconditional statements are also called **bi-implications**.

Note that the statement $p \leftrightarrow q$ is true when both the conditional statements $p \rightarrow q$ and $q \rightarrow p$ are true and is false otherwise. That is why we use the words "if and only if" to express this logical connective and why it is symbolically written by combining the symbols $\rightarrow$ and $\leftarrow$. There are some other common ways to express $p \leftrightarrow q$:

- "p is necessary and sufficient for q"
- "if p then q, and conversely"
- "p iff q."

The last way of expressing the biconditional statement $p \leftrightarrow q$ uses the abbreviation "iff" for "if and only if." Note that $p \leftrightarrow q$ has exactly the same truth value as $(p \rightarrow q) \wedge (q \rightarrow p)$.

Example: Let p be the statement "You attend the meeting," and let q be the statement "You receive the meeting minutes."

Biconditional Statement ($p \leftrightarrow q$):

"You can attend the meeting if and only if you receive the meeting minutes."
Truth Conditions:

- The statement is true if:
  - You attend the meeting and you receive the meeting minutes.
  - You do not attend the meeting and you do not receive the meeting minutes.
- The statement is false if:
  - You attend the meeting but do not receive the meeting minutes.
  - You do not attend the meeting but receive the meeting minutes.

1.2.3.1 Implicit Use of Biconditionals in Natural Language:

In everyday language, we often do not explicitly use the "if and only if" construction. Instead, we might say something like:

- "If you attend the meeting, then you will receive the meeting minutes."
- "You will receive the meeting minutes only if you attend the meeting."

These statements imply the biconditional relationship, even though the "if and only if" is not explicitly stated.

Simplified Explanation:

1. Conditional Statement ($p \rightarrow q$):

   "If you attend the meeting, then you will receive the meeting minutes."

   This only covers one direction: attending the meeting leads to receiving the minutes.

2. Implicit Biconditional ($p \leftrightarrow q$):

   "You will receive the meeting minutes only if you attend the meeting."

   This implies the converse as well: not attending the meeting means not receiving the minutes.

As a Summary:

- Biconditional: "You can attend the meeting if and only if you receive the meeting minutes."
- True: Both statements are true or both are false.
- False: One statement is true and the other is false.
- Everyday Language: Often uses "if, then" or "only if" constructions to imply a biconditional relationship.

Example: Determine whether each of the following bi-conditional statement is true or false.

1. A figure is a rectangle if and only if it has four right angles. (T)
2. An integer is even if and only if it is divisible by 2. (T)
3. 1+2=3 if and only if 1+1=3 (F)
4. Dogs can fly if and only if 1+1=3 (T)

1.2.4 Truth Tables of Compound Propositions

We have introduced four important logical connectives: conjunctions, disjunctions, conditional statements, and biconditional statements, along with negations. These connectives allow us to construct complex compound propositions using multiple propositional variables. To determine the truth values of these compound propositions, we use truth tables. Each step in the compound proposition is evaluated in its own column, with the final column showing the truth value for every possible combination of truth values for the propositional variables.

Example: Construct the truth table for the compound proposition $(p\wedge) \vee r$.

p	q	r	$(p \wedge q)$	$((p \wedge q) \vee r)$
T	T	T	T	T
T	T	F	T	T
T	F	T	F	T
T	F	F	F	F
F	T	T	F	T
F	T	F	F	F
F	F	T	F	T
F	F	F	F	F

Table 1.10: The Truth Table of ((p $\wedge q) \vee r$).

Example: Construct the truth table for the compound proposition $(p \wedge \neg r) \vee (q \rightarrow r)$.

Steps:

1. Columns: We need columns for p, q, and r.
2. Negation: Include a column for $\neg r$.
3. Conjunction: Include a column for $p \wedge \neg r$.
4. Conditional: Include a column for $q \rightarrow r$.
5. Disjunction: Finally, include a column for $(p \wedge \neg r) \vee (q \rightarrow r)$.

p	q	r	$\neg r$	$p \wedge \neg r$	$q \rightarrow r$	$(p \wedge \neg r) \vee (q \rightarrow r)$
T	T	T	F	F	T	T
T	T	F	T	T	F	T
T	F	T	F	F	T	T
T	F	F	T	T	T	T
F	T	T	F	F	T	T
F	T	F	T	F	F	F
F	F	T	F	F	T	T
F	F	F	T	F	T	T

Table 1.11: The Truth Table of $(p \wedge \neg r) \vee (q \rightarrow r)$.

Explanation:

1. Columns 1-3: List all possible truth values for p, q, and r.
2. Column 4 ($\neg r$): Negate the values in column 3 (r).
3. Column 5 ($p \wedge \neg r$): Perform the conjunction operation between column 1 (p) and column 4 ($\neg r$).
4. Column 6 ($q \rightarrow r$): Evaluate the conditional $q \rightarrow r$ using columns 2 (q) and 3 (r).
5. Column 7 ($(p \wedge \neg r) \vee (q \rightarrow r)$): Finally, perform the disjunction between column 5 ($p \wedge \neg r$) and column 6 ($q \rightarrow r$).

This truth table helps to visualize the truth values of the compound proposition $(p \wedge \neg r) \vee (q \rightarrow r)$ for all possible combinations of p, q, and r.

1.2.4.1 Precedence of Logical Operators

We can construct compound propositions using negation and other logical operators. Parentheses specify the order of operations, for example, $(p \vee q) \wedge (\neg r)$ means the conjunction of $p \vee q$ and $\neg r$.

To reduce parentheses, we follow these rules:

1. **Negation** ($\neg$): Applies first. So, $\neg p \wedge q$ means $(\neg p) \wedge q$, not $\neg(p \wedge q)$.
2. **Conjunction** ($\wedge$): Takes precedence over disjunction ($\vee$). So, $p \wedge q \vee r$ means $(p \wedge q) \vee r$, not $p \wedge (q \vee r)$.

For clarity, we still use parentheses, especially when combining conjunctions, disjunctions, conditionals, and biconditionals. Here is the precedence of logical operators:

Operator	Precedence
$\neg$	1
$\wedge$	2
$\vee$	3
$\rightarrow$	4
$\leftrightarrow$	5

Table 1.12: Precedence of Logical Operators.

Conditional ($\rightarrow$) and biconditional ($\leftrightarrow$) operators have lower precedence than conjunction and disjunction. Thus, $p \vee q \rightarrow r$ is the same as $(p \vee q) \rightarrow r$. Parentheses clarify the order when needed.

1.2.5 Logic and Bit Operations

Computers represent information using bits. A bit is a symbol with two possible values: 0 (zero) and 1 (one). This term comes from "binary digit," as zeros and ones are the digits used in binary representations of numbers. A bit can represent a truth value because there are two truth values: true and false. We use 1 to represent true (T) and 0 to represent false (F). A variable is called a **Boolean variable** if its value is either true or false. Consequently, a Boolean variable can be represented using a bit.

Truth Value	Bit
T	1
F	0

Computer bit operations correspond to logical connectives. By replacing true with 1 and false with 0 in the truth tables for the operators $\wedge$, $\vee$, and $\oplus$, we obtain the following tables for the corresponding bit operations. We use the notation AND, OR, and XOR for the operators $\wedge$, $\vee$, and $\oplus$ in various programming languages.

p	q	AND($p \wedge q$)
1	1	1
1	0	0
0	1	0
0	0	0

p	q	OR($p \vee q$)
1	1	1
1	0	1
0	1	1
0	0	0

p	q	XOR($p \oplus q$)
1	1	0
1	0	1
0	1	1
0	0	0

Table 1.13: Tables for the Bit Operators OR, AND, and XOR.

Definition 8. *A **bit string** is a sequence of zero or more bits. The length of this string is the number of bits in the string.*

Example: 11011000 is a bit string of length eight.

We extend bit operations to bit strings by defining bitwise OR, bitwise AND, and bitwise XOR of two strings of the same length. These operations produce strings where each bit is the result of applying the OR, AND, or XOR operation to the corresponding bits in the two input strings. We use the symbols $\vee$, $\wedge$, and $\oplus$ to represent bitwise OR, bitwise AND, and bitwise XOR operations, respectively.

Example: Given two bit strings 01 1011 0110 and 11 0001 1101,find their bitwise OR, bitwise AND, and bitwise XOR.

The bitwise OR, bitwise AND, and bitwise XOR of these strings are obtained by taking the OR, AND, and XOR of the corresponding bits, respectively. This gives us:

$$\begin{array}{ll} 01\,1011\,0110 & \\ 11\,0001\,1101 & \\ \hline 11\,1011\,1111 & \text{bitwise OR} \\ 01\,0001\,0100 & \text{bitwise AND} \\ 10\,1010\,1011 & \text{bitwise XOR} \end{array}$$

Exercise:

1. Evaluate the expression $(101011 \oplus 110011) \wedge (011010 \vee 100110)$.
2. Simplify the expression $(10101 \wedge 11011) \vee (10010 \oplus 01010)$.
3. Compute the result of $(111001 \wedge 100011) \oplus (110110 \vee 101010)$.
4. Evaluate the expression $(1111 \vee 0101) \wedge (1010 \oplus 0110)$.

1.3 Applications of Propositional Logic

Logic is incredibly useful in many fields like math, computer science, and more. Often, statements in these areas or in everyday language are unclear or open to interpretation. To make them clearer and more precise, we can translate them into the language of logic.

For instance, when developing software or hardware, it's crucial to have precise specifications before starting. Logic helps in this process by providing a clear framework for defining requirements.

Additionally, propositional logic and its rules play a big role in various tasks. They're used to design computer circuits, create computer programs, verify program correctness, and even build expert systems.

Logic is also handy for solving puzzles and analyzing problems systematically. Some software systems are even built to automatically generate certain types of proofs based on logical rules.

1.3.1 Translating Statements

Translating English sentences into expressions with propositional variables and logical connectives serves several purposes. Once translated, we can analyze these expressions to determine their truth values.

Example: Translate the following English sentence into a logical expression:

"If it is raining, then I will bring an umbrella"

Solution: We can represent:

- r: "It is raining."
- u: "I will bring an umbrella."

The sentence implies a conditional relationship, where the action of bringing an umbrella depends on whether it is raining. So, we can translate the sentence into the logical expression

$$r \to u.$$

This expression reads as "If it is raining, then I will bring an umbrella." Using propositional variables and logical connectives helps us precisely capture the meaning of the English sentence in a logical form.

Example: Translate the following English sentence into a logical expression:

"If Alice buys a ticket and Bob accompanies her, then they will attend the concert together."

Solution:

Let's represent:

- a: "Alice buys a ticket."
- b: "Bob accompanies Alice."
- c: "They will attend the concert together."

The sentence implies a conditional relationship, where Alice buying a ticket and Bob accompanying her results in them attending the concert together. So, we can translate the sentence into the logical expression

$$(a \wedge b) \rightarrow c.$$

Example: Translate the English sentence

"You can enter the contest if you are a resident of the country and you are over 18 years old, or if you have won a previous contest"

into a logical expression.

Solution: Let's represent:

- e: "You can enter the contest."
- r: "You are a resident of the country."
- a: "You are over 18 years old."
- w: "You have won a previous contest."

The sentence implies a conditional relationship and an alternative condition. The condition is that you can enter the contest if you are a resident of the country and over 18 years old, represented as $(r \wedge a)$. Additionally, there is an alternative condition: you can enter the contest if you have won a previous contest, represented as w.

So, the logical expression for the sentence would be:

$$(r \wedge a) \vee w \rightarrow e$$

1.3.2 Logic Circuits

Propositional logic can be applied to the design of computer hardware.

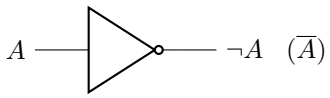
In a logic circuit, also known as a digital circuit, input signals $p_1, p_2, \ldots, p_n$ are received, each representing a bit (0 for "off" and 1 for "on"). The circuit processes these inputs and produces output signals $s_1, s_2, \ldots, s_n$, also represented as bits.

For simplicity, we'll focus on logic circuits with a single output signal. However, it's important to note that digital circuits can have multiple outputs.

1.3.2.1 Basic logic gates

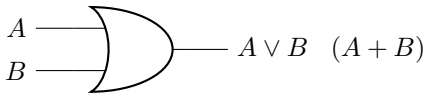
Complicated digital circuits can be constructed from **three basic circuits**, called **gates**.These gates are fundamental building blocks for constructing complex digital circuits. Here are the three basic types of gates typically used:

1. This gate, also known as an inverter, produces an output that is the complement of its input. If the input is high (1), the output is low (0), and vice versa. The symbol for a NOT gate is often depicted as follows:



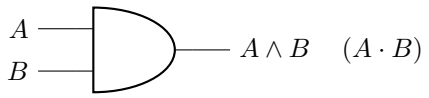
NOT gate(Inverter)

2. This gate produces a high output (1) if any of its inputs are high (1). It produces a low output (0) only if all of its inputs are low (0). The symbol for an OR gate is often depicted as follows:



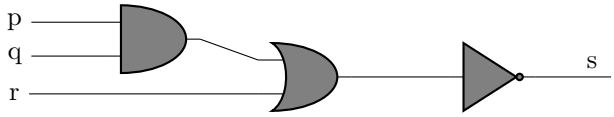
OR gate

3. This gate produces a high output (1) only if all of its inputs are high (1). Otherwise, it produces a low output (0). The symbol for an AND gate is often depicted as follows:



AND gate

Example: Find the output of the circuit given in the following figure.

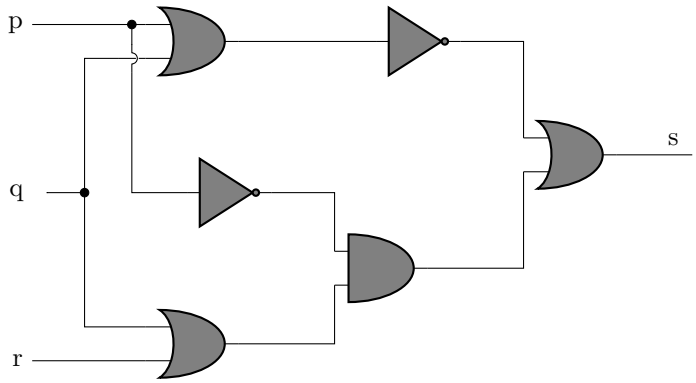


1. AND Gate :
   - Inputs: p and q
   - Output: $p \wedge q$
2. OR Gate :
   - Inputs: Output of ANDa $(p \wedge q)$ and r
   - Output: $(p \wedge q) \vee r$
3. NOT Gate :
   - Input: Output of ORa $((p \wedge q) \vee r)$
   - Output: $\neg((p \wedge q) \vee r)$

Thus, the output of the circuit s is:

$$s = \neg((p \wedge q) \vee r)$$

Example: Determine the output for the combinatorial circuit in the following figure?



Example: Consider the following logical expression: $(p \vee \neg r) \wedge (\neg p \vee (q \vee \neg r))$. Design a digital circuit that produces this output given input bits p, q, and r.

1.4 Propositional Equivalences

Definition 9. *A **tautology** is a compound proposition that is always true, regardless of the truth values of its constituent propositional variables.*

In other words, no matter what values are assigned to the variables within the proposition, the entire proposition will always evaluate to true.

Example: The proposition $p \vee \neg p$ is a tautology because it is always true, regardless of the truth value of p.

Definition 10. *: On the contrary, a **contradiction** is a compound proposition that is always false, regardless of the truth values of its constituent propositional variables.*

Regardless of the values assigned to the variables within the proposition, the entire proposition will always evaluate to false.

Example: The proposition $p \wedge \neg p$ is a contradiction because it can never be true; it asserts that p is both true and false simultaneously.

Definition 11. *A **contingency** is a compound proposition that is neither a tautology nor a contradiction. Its truth value depends on the specific truth values assigned to its constituent propositional variables.*

In other words, a contingency may be true under certain truth value assignments and false under others.

Example: The proposition $p \vee q$ is a contingency because its truth value depends on the truth values of p and q. If both p and q are true, the proposition is true; if both are false, the proposition is false.

1.4.1 Logical Equivalences

Definition 12. *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.*

Note:The symbol $\equiv$ is not a logical connective. $p \equiv q$ is not a compound proposition, but rather a statement that $p \leftrightarrow q$ is a tautology. The symbol $\Leftrightarrow$ is sometimes used instead of $\equiv$ to denote logical equivalence.

Table 1.14: De Morgan's Laws

Expression	Equivalent Expression
$\neg(p \wedge q)$	$\neg p \vee \neg q$
$\neg(p \vee q)$	$\neg p \wedge \neg q$

One way to determine whether two compound propositions are equivalent is to use a truth table. In particular, the compound propositions p and q are equivalent if and only if the columns giving their truth values agree.

Example: To show that $\neg(p \vee q)$ and $\neg p \wedge \neg q$ are logically equivalent, we can use a truth table.

p	q	$p \vee q$	$\neg(p \vee q)$	$\neg p$	$\neg p \wedge \neg q$
T	T	T	F	F	F
T	F	T	F	F	F
F	T	T	F	T	F
F	F	F	T	T	T

Table 1.15: Truth Table for $\neg(p \vee q)$ and $\neg p \wedge \neg q$

From the table, we can see that $\neg(p \vee q)$ and $\neg p \wedge \neg q$ have identical truth values in all possible cases. Hence, they are logically equivalent.

Exercise: Show that $p \to q$ and $\neg p \vee q$ are logically equivalent.

Equivalence	**Name**
$p \wedge T \equiv p$ $p \vee F \equiv p$	Identity laws
$p \vee T \equiv T$ $p \wedge F \equiv F$	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
$p \vee \neg p \equiv T$ $p \wedge \neg p \equiv F$	Negation laws

Table 1.16: Logical Equivalences

1.4.1.1 Using De Morgan's Laws

Example: Show that $\neg(p \to q)$ is logically equivalent to $p \wedge \neg q$.

$$\begin{aligned}
\neg(p \to q) &\equiv \neg(\neg p \vee q) && \text{by definition of implication} \\
&\equiv \neg(\neg p) \wedge \neg q && \text{by the second De Morgan's law} \\
&\equiv p \wedge \neg q && \text{by the double negation law}
\end{aligned}$$

Example: Show that $\neg(p \vee (\neg p \wedge q))$ and $\neg p \wedge \neg q$ are logically equivalent by developing a series of logical equivalences.

$$\begin{aligned}
\neg(p \vee (\neg p \wedge q)) &\equiv \neg p \wedge \neg(\neg p \wedge q) && \text{by the second De Morgan's law} \\
&\equiv \neg p \wedge [\neg(\neg p) \vee \neg q] && \text{by the first De Morgan's law} \\
&\equiv \neg p \wedge (p \vee \neg q) && \text{by the double negation law} \\
&\equiv (\neg p \wedge p) \vee (\neg p \wedge \neg q) && \text{by the second distributive law} \\
&\equiv F \vee (\neg p \wedge \neg q) && \text{because } \neg p \wedge p \equiv F \\
&\equiv (\neg p \wedge \neg q) \vee F && \text{by the commutative law for disjunction} \\
&\equiv \neg p \wedge \neg q && \text{by the identity law for } F
\end{aligned}$$

Example: Show that $(p \wedge q) \to (p \vee q)$ is a tautology.

$$\begin{aligned}
(p \wedge q) \to (p \vee q) &\equiv \neg(p \wedge q) \vee (p \vee q) && \text{by definition of implication} \\
&\equiv (\neg p \vee \neg q) \vee (p \vee q) && \text{by De Morgan's law} \\
&\equiv ((\neg p \vee p) \vee \neg q) \vee q && \text{by associative law} \\
&\equiv (\mathrm{T} \vee \neg q) \vee q && \text{by negation law} \\
&\equiv \mathrm{T} \vee q && \text{by domination law} \\
&\equiv \mathrm{T} && \text{by domination law}
\end{aligned}$$

1.5 Predicates and Quantifiers

1.5.1 Predicates

Statements involving variables, such as

$$x > 3 \quad , x = y + 3, \quad \text{and } x + y = z,$$

are commonly found in mathematical assertions, computer programs, and system specifications. These statements lack inherent truth values until specific values are assigned to the variables.

Consider the statement

"x is greater than 3"

It consists of two parts:

1. The variable x, representing the subject,
2. And the predicate "is greater than 3," which refers to a property that the subject can possess.

We denote this statement as $P(x)$, where

1. P represents the predicate "is greater than 3,"
2. x is the variable.

$P(x)$ is also known as the value of the propositional function P at x. Only when a specific value is assigned to x, $P(x)$ becomes a proposition with a definite truth value.

Example: Let $Q(x)$ denote the statement "x is even." What are the truth values of $Q(4)$ and $Q(3)$?

Solution: For $Q(x)$ denoting the statement "x is even," the truth values are:

$Q(4)$: True, because 4 is an even number.

$Q(3)$: False, because 3 is not an even number.

Example: Let $A(x)$ denote the statement

"Computer x is under attack by an intruder."

Suppose that of the computers on campus, only CS2 and MATH1 are currently under attack by intruders. What are the truth values of A(CS1), A(CS2), and A(MATH1)?

Solution: Let $A(x)$ denote the statement "Computer x is under attack by an intruder." Suppose that of the computers on campus, only CS2 and MATH1 are currently under attack by intruders.

The truth values are:

$A(CS1)$: False, because CS1 is not under attack by an intruder.

$A(CS2)$: True, because CS2 is under attack by an intruder.

$A(MATH1)$: True, because MATH1 is under attack by an intruder.

1.5.2 Quantifiers

When variables in a propositional function are assigned values, the resulting statement becomes a proposition with a certain truth value. However, there's another significant method, known as quantification, for creating a proposition from a propositional function. Quantification indicates the extent to which a predicate is true across a range of elements. In English, words like "all," "some," "many," "none," and "few" are utilized in quantification. We'll primarily focus on two types of quantification here:

1. Universal quantification, which indicates that a predicate holds true for every element within a specified range.
2. Existential quantification, which signifies that there exists at least one element within a specified range for which the predicate holds true.

The area of logic that deals with predicates and quantifiers is termed the predicate calculus.

Definition 13. *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****. We read $\forall x P(x)$ as "for all $xP(x)$" or "for every $xP(x)$."*

Example: Let $C(x)$ denote "x is a cat," and let $A(x)$ denote "x is an animal."
The statement "All cats are animals" can be written as:

$$\forall x(C(x) \rightarrow A(x))$$

Explanation:

- $\forall x$: This is the universal quantifier, which means "for all x."
- $C(x) \rightarrow A(x)$: This is a conditional statement that reads as "if x is a cat, then x is an animal."

This example illustrates a basic use of the universal quantifier to express that a certain property (being an animal) applies to all members of a specific group (cats).

Example: Let $P(x)$ be the statement " $x + 1 > x$." The truth value of the quantification $\forall x P(x)$, where the domain consists of all real numbers, is as follows:

Since $P(x)$ is true for all real numbers x, the quantification $\forall x P(x)$ is true.

Remark: Generally, it's implicitly assumed that all domains of discourse for quantifiers are nonempty. If the domain happens to be empty, $\forall x P(x)$ is true for any propositional function $P(x)$, as there are no elements x in the domain for which $P(x)$ is false.

Note :Universal quantification can be expressed in various ways, including "all of," "for each," "given any," "for arbitrary," "for each," and "for any." However, it's advisable to avoid using "for any x" because it can be ambiguous, as it may imply either "every" or "some." In some contexts, such as in negatives, "any" is unambiguous, as in "there is not any reason to avoid studying."

A statement $\forall x P(x)$ is false, where $P(x)$ is a propositional function, if and only if $P(x)$ is not always true when x is in the domain. One way to demonstrate that $P(x)$ is not always true when x is in the domain is by finding a counterexample to the statement $\forall x P(x)$. Note that a single counterexample is sufficient to establish that $\forall x P(x)$ is false.

Example: Let $Q(x)$ be the statement " $x < 2$." The truth value of the quantification $\forall x Q(x)$, where the domain consists of all real numbers, is as follows:

Since $Q(x)$ is not true for every real number x (e.g., $Q(3)$ is false), $x = 3$ serves as a counterexample for the statement $\forall x Q(x)$. Hence, $\forall x Q(x)$ is false.

Definition 14. *The existential quantification of $P(x)$ is the proposition*

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

We use the notation $\exists x P(x)$ for the existential quantification of $P(x)$. Here $\exists$ is called the ***existential quantifier****.*

A domain must always be specified when using a statement $\exists x P(x)$. Moreover, the interpretation of $\exists x P(x)$ varies depending on the domain. Without specifying the domain, the statement $\exists x P(x)$ is meaningless.

In addition to the phrase "there exists," existential quantification can be expressed using various other terms, such as "for some," "for at least one," or "there is." The existential quantification $\exists x P(x)$ is interpreted as:

- "There is an x such that $P(x)$,"
- "There is at least one x such that $P(x)$,"
- "For some $x, P(x)$."

Example: Let $P(x)$ be the statement " $x > 3$." What is the truth value of the quantification $\exists x P(x)$, where the domain consists of all integers?

Solution: Let $P(x)$ denote the statement " $x > 3$." The truth value of the quantification $\exists x P(x)$, where the domain consists of all integers, is as follows:

Since " $x > 3$ " is sometimes true (e.g., when $x = 4$), the existential quantification of $P(x)$, denoted as $\exists x P(x)$, is true.

Observe that the statement $\exists x P(x)$ is false if and only if there is no element x in the domain for which $P(x)$ is true. That is, $\exists x P(x)$ is false if and only if $P(x)$ is false for every element of the domain.

Example: Let $Q(x)$ denote the statement " $x = x + 1$." What is the truth value of the quantification $\exists x Q(x)$, where the domain consists of all real numbers?

Solution: Since there are no real numbers x that satisfy the equation $x = x+1$, the existential quantification $\exists x Q(x)$ is false.

Observe that the statement $\exists x Q(x)$ is false if and only if there is no element x in the domain for which $Q(x)$ is true.

Statement	When True?	When False?
$\forall x P(x)$	$P(x)$ is true for every x.	There is an x for which $P(x)$ is false.
$\exists x P(x)$	There is an x for which $P(x)$ is true.	$P(x)$ is false for every x.

Table 1.17: Quantifiers

1.5.3 Negating Quantified Expressions

We often need to consider the negation of a quantified expression. For instance, let's consider the negation of the statement

"Every student in your class has taken a course in calculus."

This statement is a universal quantification, represented as

$$\forall x P(x), \text{ where } P(x)$$

is the statement "x has taken a course in calculus," and the domain consists of the students in your class.

The negation of this statement is

"It is not the case that every student in your class has taken a course in calculus,"

which is equivalent to

"There is a student in your class who has not taken a course in calculus."

This is simply the existential quantification of the negation of the original propositional function, represented as

$$\exists x \neg P(x).$$

This example illustrates the following logical equivalence:

$$\neg \forall x P(x) \equiv \exists x \neg P(x)$$

Suppose we want to negate an existential quantification. For example, let's consider the proposition

"There is a student in this class who has taken a course in calculus."

This can be represented as the existential quantification

$$\exists x Q(x), \text{ where } Q(x)$$

is the statement

"x has taken a course in calculus."

The negation of this statement is

"It is not the case that there is a student in this class who has taken a course in calculus."

This is equivalent to

"Every student in this class has not taken calculus,"

which is the universal quantification of the negation of the original propositional function, or, phrased in the language of quantifiers,

$$\forall x \neg Q(x).$$

This example illustrates the equivalence:

$$\neg \exists x Q(x) \equiv \forall x \neg Q(x)$$

Quantifier	Equ. Statement	When is Negation True?	When is Negation False?
$\neg \forall x P(x)$	$\exists x \neg P(x)$	There exists an x for which $P(x)$ is false	Every x satisfies $P(x)$
$\neg \exists x P(x)$	$\forall x \neg P(x)$	Every x for which $P(x)$ is false	There exists an x for which $P(x)$ is true

Table 1.18: De Morgan's Laws for Quantifiers

Example: What are the negations of the statements "There is an honest politician" and "All Americans eat cheeseburgers"?

Solution: Let $H(x)$ denote "x is honest." Then the statement

"There is an honest politician"

is represented by

$$\exists x H(x),$$

where the domain consists of all politicians. The negation of this statement is

$$\neg\exists x H(x),$$

which is equivalent to

$$\forall x \neg H(x).$$

This negation can be expressed as "Every politician is dishonest."

Example: What are the negations of the statements $\forall x(x^2 > x)$ and $\exists x(x^2 = 2)$?

The negations of the statements are as follows:

1. For the statement $\forall x(x^2 > x)$, which asserts that "For all x, x^2 is greater than x":
   (a) Its negation is $\neg\forall x(x^2 > x)$, which is equivalent to $\exists x \neg(x^2 > x)$.
   (b) This can be expressed as "There exists an x such that x^2 is not greater than x."
2. For the statement $\exists x(x^2 = 2)$, which asserts that "There exists an x such that $x^2 = 2$":
   (a) Its negation is $\neg\exists x(x^2 = 2)$, which is equivalent to $\forall x \neg(x^2 = 2)$.
   (b) This can be expressed as "For all x, x^2 is not equal to 2."

The negation of the statement $\forall x(x^2 > x)$ is $\neg\forall x(x^2 > x)$, which is equivalent to $\exists x \neg(x^2 > x)$.
The negation of the statement $\exists x(x^2 = 2)$ is $\neg\exists x(x^2 = 2)$, which is equivalent to $\forall x \neg(x^2 = 2)$.