Tribhuvan University

Institute of Science and Technology

Bachelor of Science in Computer Science and Information Technology

Detailed Syllabus

Course Title: Discrete Structure

Course No.: CSC-152

Credit Hours: 3

Full Marks: 90+10

Pass Marks: 36+4

Nature of Course: Theory (3 hrs.)

Course Synopsis: This course contains the fundamental concepts of logic, reasoning and algorithms.

Goal: After completing this course, the target student will gain knowledge in discrete mathematics and finite state automata in an algorithmic approach. It helps the target student in gaining fundamental and conceptual clarity in the area of Logic, Reasoning, Algorithms, Recurrence Relation, and Graph Theory.

Course Contents

Unit 1: Logic, Induction and Reasoning (12 hrs.)

Proposition and Truth function, Propositional Logic, Expressing statements in Logic Propositional Logic, The predicate Logic, Validity, Informal Deduction in Predicate Logic, Rules of Inference and Proofs, Informal Proofs and Formal Proofs, Elementary Induction, Complete Induction, Methods of Tableaux, Consistency and Completeness of the System.

Unit 2: Finite State Automata (10 hrs.)

Sequential Circuits and Finite state Machine, Finite State Automata, Language and Grammars, Non-deterministic Finite State Automata, Language and Automata, Regular Expression.

Unit 3: Recurrence Relations (8 hrs.)

Recursive Definition of Sequences, Solution of Linear recurrence relations, Solution to Nonlinear Recurrence Relations, Application to Algorithm Analysis. Combinatory, Partial Order relation.

Unit 4: Graph Theory (15 hrs.)

Undirected and Directed Graphs, Walk Paths, Circuits, Components, Connectedness Algorithm, Shortest Path Algorithm, Bipartite Graphs, Planar Graphs, Regular Graphs, Planarity Testing Algorithms, Eulerian Graph, Hamiltonian Graph, Tree as a Directed Graph, Binary Tree, Spanning Tree, Cutsets and Cutvertices, Network Flows, Maxflow and Mincut Theorem, Data Structures Representing Trees and Graphs in Computer, Network Application of Trees and Graphs, Concept of Graph Coloring.

Text / Reference Books

1. Kenth Rosen, *Discrete Mathematical Structures with Applications to Computer Science*, WCB/ McGraw Hill.
2. G. Birkhoff, T.C. Bartee, *Modern Applied Algebra*, CBS Publishers.
3. R. Johnsonbaugh, *Discrete Mathematics*, Prentice Hall Inc. Princeton
4. G.Chartand, B.R.Oller Mann, *Applied and Algorithmic Graph Theory*, McGraw Hill.
5. Joe L. Mott, Abrahm Kandel, and Theodore P. Baker, *Discrete Mathematics for Computer Scientists and Mathematicians*, Prentice-Hall of India.

Table of Content

Logic

Mathematical Induction and Reasoning

Regular Expression, Grammar and Language 16

Recurrence Relations 8

Graph Theory 24

Scanned by CamScanner

1
Chapter

Logic

Introduction

The term "logic" came from the Greek word logos, which is sometimes translated as "discourse", "reason", etc. We might define logic as the study of the principles of correct reasoning. In artificial intelligence field, Logic is defined as the representation language of knowledge. Since logic can helps us to reason the mathematical models it needs some rules associated with logic so that we can apply those rules for mathematical reasoning. There are lots of applications of logic in the field of computer science such as designing circuits, programming, program verifications, etc.

The logic, on the basis of its representation capability are of two types: Propositional logic and predicate logic.

Propositional Logic

Propositional Logic also known as sentential logic is the branch of logic that studies way of joining or modifying propositions to form more complicated propositions as well as logical relationships. The propositional logic represents knowledge in terms of sentence in terms of propositions. In Propositional Logic, there are two types of sentences: simple sentences and compound sentences. Simple sentences express atomic propositions about the world. Compound sentences express logical relationships between the simpler sentences of which they are composed.

Propositions

Proposition is a declarative sentence that is either true or false, but not both. Example

- $2 + 2 = 5$. (False), is a proposition.
- $7 - 1 = 6$. True), is a proposition.
- odd numbers are divisible by 2 (false), is a proposition.
- Kathmandu is the capital of Nepal. (True), is a proposition.
- Open the door. Not a proposition.

The essential property of proposition is that, it is either true or false but not both.

Let us try to analyze the sentences below:

$x > 15$, go there, Who are you?

The above sentences are not propositions since we cannot say whether they are true or false.

Simple Proposition and Propositional Variables

- Any statement whose truth value does not depend on another proposition is called simple proposition. E.g. Kathmandu is capital city of Nepal.

  In Propositional Logic, these are often called propositional constants or, sometimes, logical constants. Propositions are denoted conventionally by using small letters like p, q, r, s .... The truth value of proposition is denoted by 'T' for true proposition and 'F' for false proposition. Reminder: p, q, r, s ... are not actual propositions but they are propositional variables i.e. place holders for propositions.
- Compound propositions are formed from simpler propositions and express relationships among the constituent sentences. There are six types of compound sentences, viz. negations, conjunctions, disjunctions, implications, reductions, and equivalences.

Truth Table

The table which consists of all possible truth value of any propositions (simple as well as compound). Truth tables are specially valuable in the determination of the truth values of propositions constructed from simple propositions.

Logical Operators or Connectives

Logical operators are used to construct mathematical statements having one or more Logical operators are used to construct mathematical statements having one or more propositions by combining propositions. The combined proposition is called compound Proposition. Here we present the logical operators along with their behavior in truth table:

Negation (not)

Let p be any proposition. The negative of given proposition P denoted by $\neg p$ is called negation of p. Given a proposition p, negation operator ($\neg$) is used to get negation of p denoted by $\neg p$ called "not p".

Example 1

Negation of the proposition "I love animals" is "I do not love animals" if the sentence I love animals is denoted by p then its negation is denoted by $\neg p$.

Example 2

Negation of proposition "Today is Sunday "is" It is not the case that today is Sunday" or "Today is not Sunday "or" It is not Sunday today."

Example 3

p: The summer in Tarai is very hot.

$\neg$ p:The summer in Tarai is not very hot.

Truth Table

p	$\neg p$
T	F
F	T

Conjunction (AND)

Given two propositions p and q, the proposition "p and q" denoted by $p \wedge q$ is proposition that is true whenever both the propositions p and q are true, false otherwise. The proposition that is obtained by the use of "and" operator is also called conjunction of p and q.

Example 4

If we have propositions p = "Ram is smart" and q = "Ram is intelligent."

The conjunction of p and q is Ram is smart and intelligent. This proposition is true when Ram is smart and he is intelligent also, false otherwise.

Example 5

p:Today is Saturday.

q:Today is holiday.

Then, $p \wedge q$:Today is Saturday and holiday

p∩q

Truth Table

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

Scanned by CamScanner

Disjunction (OR)

Given two propositions p and q, the proposition "p OR q" denoted by $p \vee q$ is the proposition that is false whenever both the propositions p and q are false, true otherwise. The proposition that is obtained by the use of "or" operator is also called disjunction of p and q.

Example 6

If we have propositions p = "Ram is intelligent" and q = "Ram is diligent."

The disjunction of p and q is Ram is intelligent or he is diligent. This proposition is false only when Ram is not intelligent and not diligent, true otherwise.

Example 7

Let, p: It is cold.

q: It is raining.

Then, $p \vee q$: It is cold or raining.

Truth Table

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

$p \vee q$

Exclusive OR (XOR)

Given two propositions p and q, the proposition exclusive or of p and q denoted by $p \oplus q$ is the proposition that is true whenever only one of the propositions p and q is true, false otherwise. As opposed to the disjunction above which is inclusive the general meaning of the English sentence can be used to know whether the "or" used is inclusive or exclusive.

Example 8

If we have propositions p = "Ram drinks coffee in the morning" and q = "Ram drinks tea in the morning"

The exclusive or of p and q is Ram drinks coffee or tea in the morning.

Truth Table

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

Implication (→)

Given two propositions p and q, the proposition implication $p \rightarrow q$ is the proposition that is false when p is true and q is false, true otherwise. Here p is called "hypothesis" or "antecedent" or "premise" and q is called "conclusion" or "consequence".

We come across the implication in many places in mathematical reasoning and we use different terminologies to express $p \rightarrow q$ like:

- "if p, then q"
- "q is consequence of p"
- "p is sufficient for q"
- "q if p" "q is necessary for p"
- "q follows from p"
- "if p, q"
- "p implies q"
- "p only if q"
- "q whenever p"
- "q provides p"

Example 9

p = "today is Sunday" q = "it is hot"

Then the implication can be "if today is Sunday then it is hot today"

Or "today is Sunday only if it is hot today".

Truth Table

p	q	$p \rightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T

Inverse, Converse and Contra positive

Some of the related implications formed from $p \rightarrow q$ are:

Inverse of Implication:

When we add 'not' to the hypothesis and conclusion of implication $p \rightarrow q$ then it becomes $\neg p \rightarrow \neg q$, which is known as inverse of $p \rightarrow q$.

Example 10

Implication $p \rightarrow q$	Inverse $\neg p \rightarrow \neg q$
If it is raining, then the road is muddy.	If it is not raining then the road is not muddy.

Converse of Implication

When we interchange/flip the hypothesis and conclusion of implication $p \rightarrow q$ then the result becomes $q \rightarrow p$ which is known as converse of given implication.

Example 11

Implication $p \rightarrow q$:	Converse $q \rightarrow p$:
If it is raining, then the road is muddy.	If the road is muddy then it is raining.

Contrapositive of Implication

When we Interchange/flip the hypothesis and conclusion of inverse statement of implication $p \rightarrow q$ then the resulting statement $\neg q \rightarrow \neg p$ is known as contrapositive.

Example 12

Implication $p \rightarrow q$:	Contrapositive $\neg q \rightarrow \neg p$:
If it is raining, then the road is muddy.	If the road is not muddy then it is not raining.

Example 13

Write the inverse, converse and contrapositive of the following statements:

a. If two angles are congruent, then they have same measures.

Statement($p \rightarrow q$)	If two angles are congruent, then they have same measures.
Inverse($\neg p \rightarrow \neg q$)	If two angles are not congruent, then they do not have same measures.
Converse($q \rightarrow p$)	If two angles have the same measures then they are congruent.
Contrapositive($\neg q \rightarrow \neg p$)	If two angles do not have same measures, then they are not congruent.

b. If quadrilateral is rectangle then it has two pairs of parallel sides.

Statement($p \rightarrow q$)	If quadrilateral is rectangle then it has two pairs of parallel sides.
Inverse($\neg p \rightarrow \neg q$)	If quadrilateral is not rectangle then it does not have two pairs of parallel sides.
Converse($q \rightarrow p$)	If quadrilateral has two pairs of parallel sides then it is a rectangle.
Contrapositive($\neg q \rightarrow \neg p$)	If quadrilateral does not have two pairs of parallel sides then it is not a rectangle.

Biconditional ($\leftrightarrow$)

Given propositions p and q, the biconditional $p \leftrightarrow q$ is a proposition that is true when p and q have same truth values. Alternatively $p \leftrightarrow q$ is true whenever both $q \rightarrow p$ and $q \rightarrow p$ are true.

Some of the terminologies used for biconditional are:

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

Example 14

For propositions given in above implication, the biconditional statement is "today is Sunday if and only if it is hot today".

Truth Table

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

Translating English Sentences

Translation of English sentence into propositional logic has much importance. For instances, translating English sentence into logical expression removes ambiguity, similarly after translating English sentence into logical expression, we can use these logical expression to analyze, determine their truth value and manipulate them. We can also use rule of inference on these logical expression to infer or derive new expressions.

The process of translating English sentences into logical expression consists of following steps:

1) First we break down the given complex English sentences into atomic sentences.

Example 15

If it is raining, the home team wins the game.

Then we break it into two atomic sentences as:

- "it is raining"
- "home team wins the game"

2) We will represent each atomic sentence by propositional variables.

- P="it is raining"
- q="home team wins the game"

3) Now each atomic sentence is connected with appropriate connectives.

In the above example, the sentences are connected with implication.

Hence the logical expression of above sentence is: $p \rightarrow q$

Example 16

Translate the following simple declarative sentences:

Let,

p: It is raining

q: Sita is sick.

r: Ram stayed up late last night.

t: Kathmandu is capital of Nepal.

u: Ashok is a loud mouth.

(a) **Translating Negation:**

(i) It is not raining.

$\neg p$, where p = it is raining.

(ii) It is not the case that Sita isn't sick.

Since, q: "Sita is sick"

$\neg q$: Sita isn't sick.

and

$\neg(\neg q)$: It is not the case that Sita isn't sick.

(b) **Translating conjunction**

(i) It is raining and Sita is sick

$(p \wedge q)$

(ii) Kathmandu isn't Capital of Nepal and is isn't raining.

$(\neg t \wedge \neg p)$

(iii) It is not the case that it is raining and Sita is sick.

Translation 1: It is not the case that both it is raining and Sita is sick.

$\neg(p \wedge q)$

Translation 2: Sita is sick and it is not the case that it is raining.

$(\neg p \wedge q)$

(c) Translating Disjunction

(i) Kathmandu is capital of Nepal and it is raining or Ashok is a loud mouth.

$[(t \wedge p) \vee u]$ or $[t \wedge (p \vee u)]$

(ii) Sita is sick or Sita isn't sick

$(q \vee \neg q)$

(iii) Ram stayed up at last night or Kathmandu isn't capital of epal.

$(r \vee \neg t)$

(d) Translating Implications

(i) If it is raining then Sita is sick.

$p \rightarrow q$

(ii) It is raining when Ashok is a loudmouth

$(u \rightarrow p)$

(iii) Sita is sick and it is raining implies that Ram stayed up late last night.

$(q \wedge p) \rightarrow r$

(e) Translating Biconditional

(i) It is raining if and only if sita is sick

$p \leftrightarrow q$

(ii) Kathmandu is capital of Nepal is equivalent to Ram stryed up late last night

$p \leftrightarrow r$

Example 17

Let p, q and r be the propositions defined as:

p: Ram get A on final exam.

q: Ram does every exercise in this book.

r: Ram got A in this class.

Translate the following into logical expression:

(a) Ram get A in this class but Ram do not do every exercise in this book.

Solution: $p \rightarrow \neg q$

(b) Ram get A on the final exam, Ram does every exercise in this book and Ram get A in this class.

Solution: $(p \wedge q \wedge r)$

(c) T get A in this class, it is necessary for Ram to get A on final.

Solution: We may rephrase it as:

If Ram get A on final exam then he will get A in this class.

$p \rightarrow r$

(d) Getting A on final and doing every exercise in this book is sufficient for getting an A in the class.

Solution:

$(p \wedge q) \rightarrow r$

Example 18

Translate the following paragraphs into logical expression;

You can access the internet from the campus only if you are a BIM graduate or Staff of campus.

Solution

Let p = "you can access the internet from campus"

q = "you are a BIM graduate"

s = "you are staff of campus"

Now the logical expression is:

$p \rightarrow (q \vee s)$

Example 19

Translate the following paragraphs into logical expression:

"Whenever the system software is being upgraded, users cannot access the file system. If users can access the file system then, they can save new files. If users cannot save new files then, system software is not being upgraded."

Solution

Let u = "The system software is being upgraded"

a = "Users can access the file system"

s = "Users can save new files"

Now the logical expressions are:

a) $u \rightarrow \neg a$

b) $a \rightarrow s$,

c) $\neg s \rightarrow \neg u$

OR

$u \rightarrow \neg a \wedge a \rightarrow s \wedge \neg s \rightarrow \neg u$

Example 20

Translate given logical expressions into English sentence:

a) $\neg p$ b) $r \wedge \neg p$ c) $\neg r \vee p \vee q$

When, p = "it rained last night"

q = "the sprinkles came on last night"

r = "the lawn was wet this morning"

Solution

a) $\neg p$ = " it didn't rain last night"

b) $r \wedge \neg p$ = "the lawn was wet this morning and it didn't rain last night"

c) $\neg r \vee p \vee q$ = "either the lawn was not wet this morning or it rained last night or the sprinkles came on last night"

Note :

To translate English sentences to the proposition symbolic form follow these steps: Restate the given sentence into building block sentences. Give the symbol to each sentence and substitute the symbols using connectives.

Example 21

"if it is snowing then I will go to the beach"

Restate into "it is snowing" give it symbol p and "I will go to the beach" and give it symbol q then we can write it as $p \rightarrow q$.

Tautology, Contradiction and Contingency

A compound proposition that is always true, no matter what the truth values of the atomic propositions that contain in it is called a tautology.

2^n

Example 22

Show that $p \vee \neg p$ is tautology.

Solution:

We can verify this statement with help of truth table as follows:

Truth Table

P	$\neg p$	$p \vee \neg p$
T	F	T
F	T	T

Here, the last column consists of all truth values so it is always true and is a tautology.

Example 23

Show that $\neg (p \rightarrow q) \rightarrow \neg q$ is tautology using truth table.

Solution:

We can verify this statement with the help of truth table as follows:

p	q	$p \rightarrow q$	$\neg(p \rightarrow q)$	$\neg q$	$\neg(p \rightarrow q) \rightarrow \neg q$
T	T	T	F	F	T
T	F	F	T	T	T
F	T	T	F	F	T
F	F	T	F	T	T

Contradiction

A compound proposition that is always false, no matter what the truth values of the atomic propositions that contain in it is called contradiction.

Example 24

Show that $p \wedge \neg p$ is contradiction.

Solution:

We can verify this statement with help of truth table as follows:

Truth Table

P	$\neg p$	$p \wedge \neg p$
T	F	F
F	T	F

Example 25

Show that $(p \vee q) \wedge [(\neg p) \wedge (\neg q)]$ is contradiction.

Solution:

We can verify this statement with the help of truth table as follows:

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

Contingency

A compound proposition that is neither a tautology nor a contradiction is called a contingency.

Example 26

Show that $\neg p \wedge \neg q$ is contingency

Solution:

We can verify this statement with the help of truth table as follows:

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

Propositional Equivalences

Given two propositions that differ in their syntax we may get the exactly same semantic for both the proposition. If two propositions are semantically identical then we say those two propositions are "equivalent". If two propositions $P(p, q, r, \ldots)$ and $Q(p, q, r, \ldots)$ where $p, q, r, \ldots$ are propositional variables have the same truth values in every possible case, the propositions are called logically equivalent and denoted as

$$P(p, q, r, \ldots) \equiv Q(p, q, r, \ldots)$$

To test whether two propositions P and Q are logically equivalent, the following steps are followed:

(i) Construct the truth table for P,

(ii) Construct truth table for Q using same propositional values

(iii) Check each combination of truth values of propositional variables to see whether the value of P is same as value of Q or not. If truth value of P is same as truth value of Q then P and Q are logically equivalent.

Such constructs are very useful in mathematical reasoning where we can substitute such propositions to equivalent propositions to construct mathematical arguments.

Example 1.7

Prove that: $(p \to r) \vee (q \to r) \equiv (p \wedge q) \to r$

Solution

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

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

Logical equivalences

The compound propositions p and q are logically equivalent, denoted by $p \Leftrightarrow q$ or $p \equiv q$, if proposition $p \leftrightarrow q$ is a tautology.

Identity Law	$p \wedge T \Leftrightarrow p$, $p \vee F \Leftrightarrow p$
Domination Law	$p \wedge F \Leftrightarrow F$, $p \vee T \Leftrightarrow T$
Idempotent Law	$p \wedge p \Leftrightarrow p$, $p \vee p \Leftrightarrow p$
Double Negation Law	$\neg(\neg p) \Leftrightarrow p$

Commutative Law	$p \wedge q \Leftrightarrow q \wedge p$, $p \vee q \Leftrightarrow q \vee p$
Associative Law	$(p \wedge q) \wedge r \Rightarrow p \wedge (q \wedge r)$
Distributive law	$p \wedge (q \vee r) \Leftrightarrow (p \wedge q) \vee (p \wedge r)$, $p \vee (q \wedge r) \Leftrightarrow (p \vee q) \wedge (p \vee r)$
De Morgan's Law	$\neg(p \wedge q) \Leftrightarrow \neg p \vee \neg q$, $\neg(p \vee q) \Leftrightarrow \neg p \wedge \neg q$
Absorption	$p \vee (p \wedge p) \equiv p$, $p \wedge (p \vee p) \equiv p$

1. **Identity Law**

a) $p \wedge T \Leftrightarrow p$ b) $p \vee F \Leftrightarrow p$

Verification:

p	$p \wedge T$	$p \vee F$
T	T	T
F	F	F

2. **Domination Law**

a) $p \wedge F \Leftrightarrow F$ b) $p \vee T \Leftrightarrow T$

Verification:

P	$p \wedge F \equiv F$	$p \vee T \equiv T$
T	F	T
F	F	T

Similarly, we can prove for $p \vee T \Leftrightarrow T$

3. **Idempotent Law**

a) $p \wedge p \Leftrightarrow p$

b) $p \vee p \Leftrightarrow p$

Verification:

p	p^ p ≡ p	p ∨ p ≡ p
T	T	T
F	F	F

Similarly we can prove for p ∨ p ⇔ p

4. **Double Negation Law**

¬(¬p) ⇔ p Double negation law

Verification:

p	¬p	¬(¬p) ≡ p
T	F	T
F	T	F

5. **Commutative Law**

a) p ∧ q ⇔ q ∧ p

b) p ∨ q ⇔ q ∨ p

Verification:

p	q	p^q	q^p
T	T	T	T
T	F	F	F
F	T	F	F
F	F	F	F

Similarly we can prove for p ∨ q ⇔ q ∨ p

6. **Associative Law**

a) (p ∧ q) ∧ r ⇔ p ∧ (q ∧ r)

b) (p ∨ q) ∨ r ⇔ p ∨ (q ∨ r)

Verification:

p	q	r	p^q	(p^q)^r	q^r	p^(q^r)
T	T	T	T	T	T	T
T	T	F	T	F	F	F
T	F	T	F	F	F	F
T	F	F	F	F	F	F
F	T	T	F	F	T	F
F	T	F	F	F	F	F
F	F	T	F	F	F	F
F	F	F	F	F	F	F

Similarly we can prove for (p ∨ q) ∨ r ⇔ p ∨ (q ∨ r)

7. **Distributive law**

a) p ∧ (q ∨ r) ⇔ (p ∧ q) ∨ (p ∧ r)

b) p ∨ (q ∧ r) ⇔ (p ∨ q) ∧ (p ∨ r)

Verification:

p	q	r	q∨r	p ∧ (q ∨ r)	p ∧ q	p^r	(p ∧ q) ∨ (p ∧ r)
T	T	T	T	T	T	T	T
T	T	F	T	T	T	F	T
T	F	T	T	T	F	T	T
T	F	F	F	F	F	F	F
F	T	T	T	F	F	F	F
F	T	F	T	F	F	F	F
F	F	T	T	F	F	F	F
F	F	F	F	F	F	F	F

Similarly, we can prove for p ∨ (q ∧ r) ⇔ (p ∨ q) ∧ (p ∨ r)

8. **De Morgan's Law**

a) ¬(p ∧ q) ⇔ ¬p ∨ ¬q b) ¬(p ∨ q) ⇔ ¬p ∧ ¬q

Verification:

p	q	p∧q	¬(p ∧ q)	¬p	¬q	¬p ∨ ¬q
T	T	T	F	F	F	F
T	F	F	T	F	T	T
F	T	F	T	T	F	T
F	F	F	T	T	T	T

Similarly, we can prove for $\neg(p \vee q) \Leftrightarrow \neg p \wedge \neg q$

Example 28

Show that $(p \to r) \wedge (q \to r)$ and $(p \vee q) \to r$ are logically equivalent using law of equivalent.

Solution

Here, given first expression

L.H.S. $(p \to r) \wedge (q \to r)$

$\equiv (\neg p \vee r) \wedge (\neg q \vee r)$ [Implications]

Let A represent $\neg p$, B represent $\neg$ and C represent r

Then,

$(A \vee C) \wedge (B \vee C) \equiv (C \vee A) \wedge (C \vee B)$

$\equiv C \vee (A \wedge B)$ [Distribution law]

$\equiv (A \wedge B) \vee C$ [Commutative law]

$\equiv (\neg p \wedge \neg q) \vee r$ [Substitution law]

$\equiv \neg (p \vee q) \vee r$ [Demorgan's law]

$\equiv (p \vee q) \to r$ [Implication law]

Example 29

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

Solution

To show that this statement is a tautology, we will use logical equivalences to demonstrate that it is logically equivalent to T.

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

$\equiv \neg (p \vee \neg q) \vee (p \vee q)$ [Demorgan's law]

$\equiv (\neg p \vee p) \vee (\neg q \vee q)$ [Commutative law]

$\equiv T \vee T$ [Commutative law]

$\equiv T$ [Domination law]

Logic and Bit operations

The meaning of the word bit comes from binary digit, since zeros and ones are the digits used in binary representation of numbers. Computer represents information using bits. A bit has two possible values 0 and 1 where 0 represent F (false) and 1 represent T (true). Boolean variable can be represented using a bit.

Computer bit operations correspond to the logical connectives like: ∧, ∨, and ⊕.

Bitwise AND (∧) operation

The bitwise and performs logical ANDing between two operands in the form of bit strings. The result of ANDing operation is 1 if both the bits have value of 1, otherwise it is zero.

Example 30

Let A=1001 and B=1100 are two binary numbers. Now the bitwise ANDing of A and B is

A=1001
B=1100
A∧B=1000

Bitwise OR

The bitwise OR performs logical bitwise ORing (bit by bit) between two bit strings. The result of ORing operation is 1 if either of bits have a value 1 otherwise it is zero.

Example 31

Let A=1001 and B=1101 are two binary numbers. Now the bitwise ORing of A and B is

A=1001
B=1101
A∨B=1101

Bitwise XOR

The bitwise XOR performs logical XORing between two bit strings. The result of XOR is 1 only if one of the bits have a value of 1 otherwise it is zero.

Example 32

Let A=1001 and B=1101 are two binary numbers. Now the bitwise ORing of A and B is:

A=1001
B=1101
A⊕B=0100

Example 33

Find biwise AND, OR and XOR of the given strings:

1011010 and 0110101.

Solution

The bitwise AND, OR and XOR of the given bit strings are obtained by taking, ANDing, ORing and XORing between the corresponding bits of given bit strings as:

	1	0	1	1	0	1	0
	0	1	1	0	1	0	1
Bitwise AND:	0	0	1	0	0	0	0
Bitwise OR:	1	1	1	1	1	1	1
Bitwise XOR:	1	1	0	1	1	1	1

Predicate

Any declarative statements involving variables often found in mathematical assertion and in computer programs, which are neither true nor false when the values of variables are not specified is called predicate.

Let's take a statement 5>9, this is a propositional statement because it is false. Now let's take a statement "x > 4". Is this statement a proposition? The answer is no. The statement may be either true or false depending upon the value of x (variable). We can say that any statement involving variable is not proposition.

The predicate "x > 4" has two parts. The first part, variable x is subject of statement and another is relation part ">4" called "predicate" refers to a property that the subject of statement have. We can denote the statement "x>4" by P(x) where P is predicate ">4" and x is the variable. We also call P as a propositional function where P(x) gives value of P at x. Once value is assigned to the propositional function then we can tell whether it is true or false i.e. a proposition.

For e.g. if we put the value of x as 3 and 7 then we can conclude that P(3) is false since 3 is not greater than 4 and p(7) is true since 7 is greater than 4.

We can also denote a statements with more than one variable using predicate like for the statement "x = y" we can write P(x,y) such that P is the relation "equals to". Similarly the statements with higher number of variables can be expressed.

Thus a predicate is a sentence that contains a finite number of variables and becomes a proposition when specific values are substituted for the variables.

Note:

The logic involving predicates is called Predicate Logic or Predicate calculus similar to logic involving propositions is Propositional Logic or Propositional Calculus.

Example 34

Let P(x): x+2<10, find the truth value of P(5)and P(9).

Solution

Given,

$P(x) : x + 2 < 10$

When x = 5,

$P(5) : 5 + 2 < 10$

$7 < 10$ (true)

When x = 9,

$P(9) : 9 + 2 < 10$

$11 < 10$ (false)

Example 35

Let F(x, y) : x = y + 6. Find the truth value of F(1, 5) and F(6, 0).

Solution

Given,

$F(x, y) : x = y + 6$

For F(1, 4), Set x = 1 and y = 5 we get,

$F(1, 5): 1 = 5 + 6$

$1 = 11$ (False)

Similarly,

For F(6, 0), Set x = 6, y = 0 we get,

$F(6, 0) : 6 = 0 + 6$

i.e. 6 = 6 (true)

∴ F(1, 5) is false and F(6, 0) is true for F(x, y) : x = y + 6.

Quantifiers

Quantifiers are the tools that change the propositional function into a proposition. These are the word that refers to quantities such as "some" or "all" and indicates how frequently a certain statement is true. Construction of propositions from the predicates using quantifiers is called quantification. The variables that appear in the statement can take different possible values and all the possible values that the variable can take forms a domain called "Universe of Discourse" or "Universal set".

There are two types of quantifier Universal quantifier and Existential quantifier.

Universal Quantifier

The phrase "for all" denoted by $\forall$, is called universal quantifier. The process of converting predicate into proposition using universal quantifier is called universal quantification. So, the universal quantification of P(x), denoted by $\forall x$ P(x), is a proposition where "P(x) is true for all the values of x in the universe of discourse".

We can represent the universal quantification by using the English language like: "for all x P(x) holds" or "for every x P(x) holds" or "for each x P(x) holds".

Example 36

Take universe of discourse a set of all students of Kathmandu College.

P(x) represents: x takes Discrete Mathematics class.

Here universal quantification is $\forall x$ P(x), which represent the English sentence "all students of Kathmandu college take Discrete Mathematics class", and now it is a proposition.

The universal quantification is conjunction of all the propositions that are obtained by assigning the value of the variable in the predicate. Going back to above example if universe of discourse is a set {Ram, Shyam, Hari, Sita} then the truth value of the universal quantification is given by P(ram) $\wedge$ P(Shyam) $\wedge$ P(Hari) $\wedge$ P(Sita) i.e. it is true only if all the atomic propositions are true.

Existential Quantifier

The phrase "there exist", denoted by $\exists$, is called existential quantifier. The process of converting predicate into proposition using existential quantifier is called existential quantification. The existential quantification of P(x), denoted by $\exists x$ P(x), is a proposition where "P(x) is true for some values of x in the universe of discourse". The other forms of representation include "there exists x such that P(x) is true" or "P(x) is true for at least one x".

Example 37

For the same problem given in universal quantification $\exists x$ P(x) is a proposition is represent like " some students of Kathmandu College take Mathematics class".

The existential quantification is the disjunction of all the propositions that are obtained by assigning the values of the variable from the universe of discourse. So the above example is equivalent to P(Ram) $\vee$ P(Shyam) $\vee$ P(Hari) $\vee$ P(Sita), where all the instances of variable are as in example of universal quantification. Here if at least one of the students takes graphics class then the existential quantification results true.

Example 38

Let Z, the set of integer, be the universe of discourse and consider the statements

(i) $\forall x \in Z, x^2 = x$ (ii) $\exists x \in Z, x^2 = x$

Find the truth values of each of the statements.

Solution

Let P(x) : $x^2 = x$ then

$\forall x$P(x) is false because $2 \in z$, P(2) : $2^2 = 2$ is false and $\exists x$P(x) is true because at least one proposition is true i.e. $1 \in z$, P(1) : $1^2 = 1$ is true.

Example 39

Let N = {1, 2, 3, 4, 5, 6, 7, 8, 9} be set of natural number. Determine the truth value of each of the following statements.

(a) $\exists x \in N, x + 5 = 12$

(b) $\forall x \in N, x + 4 < 15$

(c) $\forall x \in N, x + 5 \leq 10$

Solution

Given,

N = {1, 2, 3, 4, 5, 6, 7, 8, 9}

(a) $\exists x \in N, x + 5 = 12$ is true

for if x = 7 then x + 5 = 12

or, 7 + 5 = 12

i.e. 12 = 12 (true)

(b) $\forall x \in N, x + 4 < 15$ is true

for every $x \in N$ satisfies $x + 4 < 15$

(c) $\forall x \in N, x + 5 \leq 10$ is false for $6 \in N, 6 + 5 \leq 10$

$\Rightarrow 11 \leq 10$ (false)

Translating the Sentences into Logical Expression

Example 40

Translate "not every integer is even" where the universe of discourse is set of integers.

Solution

Let E(x) denotes x is even.

Then $\neg \forall x E(x)$ represents the above statement "not every integer is even"

Example 41

Translate "every man is mortal"

Let M(x) denote x is mortal, where x is from set of man (here universe of discourse is all man)

Then, $\forall x\, M(x)$ represent that "for all x, x is mortal."

Example 42

Every person is precious.

Solution

We translate this as:

For every x, if x is person then x is precious

P(x) : x is a person

Q(x) : x is precious

$\therefore \quad \forall x, P(x) \rightarrow Q(x)$

Example 43

Some student of this college passed CSIT entrance examination.

Solution:

We translate this as:

For some x, x is student of this college and x has passed CSIT entrance

Let, C(x) : x is student of this college.

E(x) : x passed CSIT entrance examination.

$\therefore \quad \exists x, C(x) \wedge E(x)$

where universe of discourse is set of all students.

Example 44

Every student in this class has studied discrete mathematics where universe of discourse is set of person.

Solution

We translate this as:

for every person x, if x is student of this class then x has studied discrete.

Let, P(x) : x is student of this class

Q(x) : x has studied discrete

$\therefore \quad \forall x, P(x) \rightarrow Q(x)$

Example 45

Let F(x): x is man M(x): x is mortal

(a) All man are mortal

$\forall x(F(x) \rightarrow M(x))$

i.e. For all x, if x is man, then x is mortal

(b) Let I(x) : x is an integer

P(x) : x is either positive or negative

"Any integer is either positive or negative".

This can be expressed as:

for all x, if x is an integer, then x is either positive or negative.

$\forall x(I(x) \rightarrow P(x))$

Example 46

Let, S(x) : x is a student

C(x): x is cleaver

M(x): x is successful

Express the following using quantifier.

(a) There exist a student

$\exists x\, S(x)$

(b) Some students are cleaver

This can be written as:

There exist an x such that x is student and x is cleaver.

$\therefore \quad \exists x(S(x) \wedge C(x))$

(c) "Some students are not successful"

This can be expressed as:

There exist an x such that x is student and x is not successful

$\therefore \quad \exists x(S(x) \wedge \neg M(x))$

Bounded and Free-variables

When a variable is assigned a value or associated with a quantifier, it is known as bounded variable.

If the variable is not bounded then it is know as free variable.

For example;

(1) $P(x, y)$ - x & y are free variable,

(2) $P(?, y)$ - y is free variable.

(3) $\forall x, P(x)$ - x is bounded variable

(4) $\forall x, P(x, y)$ - Here x is bounded and y is free variable.

Note: Any expression with no free variable is proposition and expression with at least one free variable is prodicate.

Nested Quantifier

When we use more than one quantifier in a sequence, then it is known as nested quantifier.

For example: $\forall x, \exists y\, P(x, y)$. Here quantifier $\exists$ & $\forall$ are rested in a sequences.

Example 47

Translate following sentence using quantifier.

(a) Everyone loves someone.

(b) Someone loves somebody.

(c) Everyone loves everybody.

Solution

Let $L(x, y)$: x loves y.

Then,

(a) $\forall x\, \exists y, L(x, y) \rightarrow$ for all x, there is some y such that x loves y.

(b) $\exists x\, \exists y\, L(x, y)$

(c) $\forall x\, \forall y\, L(x, y)$

Negation of Quantified Expression

Let $\forall x, P(x)$ is a quantified statement its negation is $\neg \forall x\, P(x)$, which is logically equivalent to $\exists x \neg P(x)$.

Example 48

Let x represent any girls in KTM

$L(x)$: x is lovely.

Then

$\forall x\, L(x)$: every girl in KTM is lovely.

Now, negative of above sentence is

$\forall x\, L(x) \equiv \exists x \neg L(x)$

I.e, there is some (at least, one) girl In KTM who is not lovely.

Example 49

No one has claimed every mountain in the Himalayas.

Let x represent a people.

$M(x) = x$ claimed every mountain in the Himalayas.

Then,

$\forall x\, M(x)$: everyone has claimed every mountain in the Himalays.

$\forall x \neg m(x)$: Every one has not claimed every mountain

which is logically equivalent to no one has claimed mountain.

Example 50

No girls in KTM is lovely.

Let,

$L(x)$: X is lovely.

$\forall x\, L(x)$: Every girl is lovely.

$\forall x \neg L(x)$: No girl is lovely

[Note: Equivalence involving negative]

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

b. $\neg \forall x\, p(x) \equiv \exists x \neg P(x)$

c. $\forall x \neg p(x) \equiv \neg [\exists x\; P(x)]$

d. $\exists x \neg P(x) \equiv \neg [\forall x\, P(x)]$

Find

The negation of (a) $\forall x \neg P(x)$ (b) $\exists x \neg p(x)$

Solution

(a) $\neg [\forall x, \neg P(x)]$

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

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

$\equiv \exists x\, P(x)$

(b) $\neg [\exists x, \neg P(x)]$

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

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

$\equiv \forall x, P(x)$

Scanned by CamScanner

EXERCISE

1. What do you mean by proposition? Give example to justify your answer.
2. Which of the following sentences are propositions? What are the truth values of those that are propositions?
   a. Kathmandu is capital of Nepal. b. CPU is an output device.
   c. 5+8=12 d. x+4=8
   e. What is your name? f. $x + y=y + z$ if $x = z$
3. Let p and q be propositions
   p : you do every exercise in this book
   q : You get grade A on the final exam
   Express the following into English sentence
   (a) $\neg p$ (b) $p \rightarrow q$ (c) $\neg p \wedge q$
   (d) $p \leftrightarrow q$ (e) $\neg q \rightarrow \neg p$
   (f) $\neg p \wedge (p \vee \neg q)$
4. Let p, q & r be propositions
   p : you have the flu
   q : you miss the final examination
   r : you pass the course
   Translate the following English sentence into propositions using connections:
   a. Whenever you have the flu, you miss the final examination.
   b. You miss the final exam only if you do not pass the course.
   c. If you have the flu, you do not pass the course or you miss the final examination only if you do not pass the course.
5. Determine whether the following propositions are true or false:
   a. 2+3=5 if and only if 2+1=3
   b. 1+1=3 if and only if pigs can fly.
   c. 1+!=3, then God exist.
6. State the converse, contra positive and inverse of the following implications:
   a. If it rains tonight, then I will stay at home.
   b. I go to swimming, whenever it is sunny day.
   c. when I stay up late, it is necessary that I sleep until noon.
7. Write converse, inverse and contrapositive of the following.
   a. All glitter is not gold.
   b. All hexagon are regular polygon.

   c. Fruit juice contain vitamin C.
8. Express the following sentences into logical expressions using proposition and connections:
   a. The message is scanned for viruses whenever the message was sent from an unknown system.
   b. The message was sent to from unknown system but it was not scanned for viruses.
   c. When a message is not sent from unknown system, it is not scanned for viruses.
9. Construct the truth table for each of following propositions.
   a) $(p \vee \neg q) \rightarrow q$ b) $(p \rightarrow q) \leftrightarrow (\neg q \rightarrow \neg p)$
   c) $(p \leftrightarrow q) \oplus (p \leftrightarrow \neg q)$ d) $(p \leftrightarrow q) \vee (\neg q \leftrightarrow r)$
10. Explain Tautologies, contradiction and contingences with suitable examples.
11. Show that the statements given below are tautology.
   a. $(p \vee q) \wedge (\neg p \vee q) \rightarrow (q \vee r)$ b. $(p \rightarrow q) \wedge (q \rightarrow r) \rightarrow (p \rightarrow r)$
   c. $(p \vee q) \vee [(\neg p) \wedge (\neg q)]$ d. $(p \wedge q) \rightarrow (p \rightarrow q)$
12. Show that the following pairs of proposition are logically equivalent
   a. $\neg(p \wedge q)$ and $(\neg p \vee \neg q)$ b. $(p \vee q) \rightarrow r$ and $(p \rightarrow r) \wedge (q \rightarrow r)$
13. Show that $(P \vee q) \wedge [\neg p \wedge \neg q]$ is contradiction.
14. Show that $\neg (p \vee q)$ and $\neg p \wedge \neg q$ are logically equivalent.
15. Show that the propositions $p \vee (q \wedge r)$ and $(r \vee q) \wedge (p \vee q)$ are logically equivalent.
16. Differentiate between existential and universal quantifier with suitable examples.
17. Let P(x) denote the statement "$x \leq 5$". What are the truth values?
   (a) P(1) (b) P(3) (c) P(7)
18. Let P(x) be the statement "the word x contains the letter a". What are the truth values?
   (a) P(orange) (b) P(apple)
   (c) P(monkey) (d) P(crocodile)
19. Let Q(x, y) denote the statement "x is the capital of y". What are these truth values?
   (a) Q(Kathmandu, Nepal)
   (b) Q(Delhi, India)
   (c) Q(Colombo, Japan)
   (d) Q(Kulalpur, Australia)

20. Let P(x) be the statement "x spends more than five hours every weekday in class", where the universe of discourse for x consists of all students. Express each of these quantifications in English.

(a) $\exists x\, P(x)$ (b) $\forall x\, P(x)$

(c) $\exists x \neg P(x)$ (d) $\forall x \neg P(x)$

21. Translate these statements into English, where C(x) is "x is a comedian" and F(x) is "x is funny" and the universe of discourse consists of all people.

(a) $\forall x\, (C(x) \rightarrow F(x))$ (b) $\forall x\, (C(x) \wedge F(x))$

(c) $\exists x (C(x) \rightarrow F(x))$ (d) $\exists x\, (C(x) \wedge F(x))$

22. Express the following statements using predicate and quantifier:

a. Every student in this class has studied discrete Math's.
b. Some student in this class are smart.
c. All birds can fly.
d. Some men are genius.
e. Some numbers are not rational.
f. There is student who likes discrete mathematics but not data communication.
g. There is a student at your college who can speak Japanese and who is good in C**.

□□□

2

Chapter

Mathematical Induction and Reasoning

Introduction

Any of the mathematical statement must be supported by arguments that make it correct. For this we need to know different techniques and rules that can be applied to the mathematical statements such that we can prove the correctness of the given mathematical statement.

Rules of Inference

To draw conclusion from the given premise we must be able to apply some well defined steps that help reaching the conclusion. These steps of reaching the conclusion are provided by the rules of inference. Here some of the rules of inferences are given below:

Rule 1: Modus Ponens (or Law of Detachment)

Whenever two propositions p and $p \rightarrow q$ are both true then we confirm that q is true. We write this rule as $\dfrac{p \rightarrow q,\ p}{\therefore q}$, this rule is valid rule of inference because the implication $[p \wedge (p \rightarrow q)] \rightarrow q$ is a tautology.

p v q ¬p ∴ q	Disjunctive syllogism	$[(p \vee q) \wedge \neg p] \rightarrow q$
p ∴ p v q	Addition	$p \rightarrow (p \vee q)$
p ^ q ∴ p	Simplification	$(p \wedge n) \rightarrow p$
p v q ¬ p v r ∴ q v r	$[(p \vee q) \wedge (\neg p \vee r)] \rightarrow (q \vee r)$	Resolution

Valid Argument

An argument in propositional logic is valid if the truth of all its premises implies that the conclusion is true.

Valid Argument form

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

i.e. the argument form with premises $P_1, P_2, \ldots, P_n$ and conclusion q is valid, when $(P_1 \wedge P_2 \wedge P_3 \wedge \ldots \wedge P_n) \rightarrow q$ is a tautology.

Verification of Inference Rules/Examples

Example 3

Determine whether the following argument is valid or not. If Ram is human then Ram is mortal. Ram is human. Therefore Ram is mortal.

Solution:

Let p : Ram is human.

q : Ram is mortal.

Now, the given arguments in symbolic form becomes

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

Then, the above argument is valid if $[(p \rightarrow q) \wedge p] \rightarrow q$ is tautology. To prove this, we can construct truth table as follows:

p	q	(p→q)	(p→q^p)	[(p→q)^p]→q
F	F	T	F	T
F	T	T	F	T
T	F	F	F	T
T	T	T	T	T

Since last column consists of all truth value, so $[(p \rightarrow q) \wedge p] \rightarrow q$ is tautology. Hence, the above argument is valid.

Example 4

If you invest in stock market, then you will become rich.

If you become rich then you will be happy.

Therefore, if you invest in stock market, then you will be happy.

Solution

Let p : you invest to stock markets

q : you will become rich

r : you will be happy

Now, the given arguments/sentences in symbolic form becomes

$$\begin{array}{l} p \rightarrow q \\ \underline{q \rightarrow r} \quad \text{(transitive rule)} \\ \therefore p \rightarrow r \end{array}$$

The above argument is valid iff the $(p \rightarrow q) \wedge (q \rightarrow r) \rightarrow (p \rightarrow r)$ is tautology.

Now, we can prove above tautology using truth table:

p	q	r	p→q	q→r	p→r	(p→q)^(q→r)	(p→q)^(q→r)→(p→r)
F	F	F	T	T	T	T	T
F	F	T	T	T	T	T	T
F	T	F	T	F	T	F	T
F	T	T	T	T	T	T	T
T	F	F	F	T	F	F	T
T	F	T	F	T	T	F	T
T	T	F	T	F	F	F	T
T	T	T	T	T	T	T	T

Hence, the above argument is valid.

Example 1

Ram is hard working and if Ram is hard working, then he is intelligent. By modus ponens (verify), this logically infers Ram is intelligent.

Rule 2: Hypothetical Syllogism (Transitive Rule)

Whenever two propositions $p \to q$ and $q \to r$ are both true then we confirm that implication $p \to r$ is true. We write this rule as

$$\frac{p \to q,\ q \to r}{\therefore p \to r}$$

This rule is valid rule of inference because the implication $[(p \to q) \wedge (q \to r)] \to (p \to r)$ is a tautology.

This rule can be extended to larger numbers of implications as

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

Example 2

a) If today is Sunday, then today is rainy day and if today is rainy day, then it is wet today. By transitivity rule (verify!!!), this logically infers It is wet today.

Rule 3: Addition

Due to the tautology $p \to (p \vee q)$, the rule :

$$\frac{p}{\therefore q \vee p}$$ is a valid rule of inference.

Rule 4: Simplification

Due to the tautology $(p \wedge q) \to p$, rule

$$\frac{p}{\therefore p \wedge q}$$ is a valid rule of inference.

Rule 5: Conjunction

Due to the tautology $[(p) \wedge (q)] \to (p \wedge q)$, rule

$$\frac{p\ ,\ q}{\therefore q \wedge p}$$ is a valid rule of inference

Rule 6: Modes Tollens

Due to the tautology $[\neg q \wedge (p \to q)] \to \neg p$, rule

$$\frac{\neg q,\ p \to q}{\therefore \neg\ p}$$

is a valid rule of inference

Rule 7: Modes Tollens

Due to the tautology $[\neg q \wedge (p \to q)] \to \neg p$, rule

$$\frac{\neg q,\ p \to q}{\therefore \neg p}$$ is a valid rule of inference.

Rule 8 Resolution

Due to the tautology $[(p \vee q) \wedge (\neg p \vee r)] \to (q \vee r)$, rule

$$\frac{p \vee q,\ \neg p \vee r}{\therefore q \vee r}$$ is a valid rule of inference.

Summary of Rule of Inference

Rule of Inference	Name	Tautology
p $\underline{p \to q}$ $\therefore q$	Modus ponens	$[p \wedge (p \to q)] \to q$
$\neg q$ $\underline{p \to q}$ $\therefore \neg p$	Modus Tollens	$[\neg q \wedge (p \to q)] \to \neg p$
$p \to q$ $\underline{q \to r}$ $\therefore p \to r$	Hypothetical syllogism	$[(p \to q) \wedge (q \to r)] \to (p \to r)$

Example 5

If it rains today the college will be closed. The college is not closed today. Therefore, it did not rain today

Solution

Let p : It rains today

q : The college will close.

The above argument when changed into symbolic form becomes

$p \to q$

$\underline{\neg q}$

$\therefore \neg p$

The above argument is valid if $(p \to q) \wedge (\neg q) \to \neg p$ is tautology.

This can be verified using truth table as:

p	q	¬p	¬q	p→q	(p→q)^ ¬q	(p→q)^ ¬q→¬p
F	F	T	T	T	T	T
F	T	T	F	T	F	T
T	F	F	T	F	F	T
T	T	F	F	T	F	T

Hence, the given argument is valid.

Example 6

State which rule of inference is the basis of the following argument: "it is below freezing now. Therefore, it is either below freezing or raining now.

Solution:

Let p : it below freezing now and q : it is raining now. Then the argument in above sentences when expressed in symbolic form is:

p

......

therefore $p \vee q$.

This is a valid argument since it confirms form the addition rule of inference.

Example 7

Which rule of inference is basis for the following arguments: if it is rainy then swimming pool will be closed. It is raining. Therefore the swimming pool is closed.

let p : it is rainy and q : The pool is closed. Then the argument in above sentences when expressed in symbolic form is:

p—>q

p

......

therefore q.

This is a valid argument since it confirms form the modus pones rule of inference.

Example 8

For each of given set of premises, what relevant conclusion can be drawn? Explain the rule of inference used to obtain each conclusion.

If I eat spicy food, then I have strange dreams. I have strange dreams if there is thunder while I sleep. I didn't have strange dreams.

Solution:

Let

p = "I eat spicy food.

q = " I have strnge dreams",

r = "There is thunder while I sleep"

Then the above premises are

a) $p \to q$

b) $r \to q$

c) $\neg q$

Using modus tollens in above premises b and c we have $\neg r$ i.e.

$r \to q$

$\neg q$

....

$\neg r$.............d)

Similarly using modus tollens in above premises a and c we have $\neg p$ i.e

$p \to q$

$\neg q$

....

$\neg p$.............e)

Therefore from given argument we can conclude either $\neg r$ or $\neg p$ i.e I didn't eat spicy food or There is nothunder at time of sleep.

Example 9

For the set of premises "If I play hockey, then I am sore the next day." "I use the whirlpool if I am sore." " I did not use the whirlpool". What relevant conclusion can be drawn? Explain the rules of inference used to draw the conclusion.

Solution:

Let

p = " play hockey",

q = "I am sore",

r = "I use the whirlpool"

Then the above premises are

a) $p \rightarrow q$

b) $q \rightarrow$

c) $\neg$

Using hypot tical syllogism in premises a and b we have $p \rightarrow r$ i.e. "if I play hockey, then "I use whirlpool"

Using the modus tollens in premise c and inferred proposition $p \rightarrow r$ we conclude $\neg p$ is true i.e. p is false. p is false means " I did not play hockey".

Example 10

Construct an argument using rules of inference to show that the hypotheses "If it does not rain or if it is not cloudy today, then the swimming competition will be held and the life saving dem nstration will go on,". " If the swimming competition is held, then the trophy will be awarded,' and "The trophy was not awarded" imply the conclusion " It rained".

Solution:

Let

p = "It rains",

q = "It is foggy",

r = " e sailing race is held",

s = " ife saving demonstration is done",

and t = " Trophy is awarded".

Then we have to show that the argument

$[((\neg p \vee \neg q) \rightarrow (r \wedge s)) \wedge (r \rightarrow t) \wedge \neg t] \rightarrow p$ is valid.

Now,

[1] $(r \rightarrow t)$ [Hypothesis]

[2] $\neg t$ [Hypothesis]

[3] $\neg r$ [Modus Tollens using steps 1 and 2]

[4] $((\neg p \vee \neg q) \rightarrow (r \wedge s))$ [Hypothesis]

[5] $\neg(\neg p \vee \neg q) \vee (r \wedge s)$ [Implication of Step 4]

[6] $(p \wedge q) \vee (r \wedge s)$ [De Morgan's Law in Step 5]

[7] $p \vee (r \wedge s)$ [Simplification using step 6]

[8] $p \vee r$ [Simplification using step 7]

Here our original premises changes to $(p \vee r) \wedge \neg r$ [from step 8 and 3]

[9] $r \vee p$ [Commutative law in step 8]

[10] $\neg r \vee p$ [Addition using step 3]

[11] $p \vee p$ [Resolution using steps 9 and 10]

[12] p [Idempotent law]

Hence argument is valid. With conclusion "It rained".

Fallacy

The fallacies are arguments that are convincing but not true and produce faulty infere So fallacies are contingencies rather than tautologies. There are following types of fall that may occur in logical reasoning.

Fallacy of affirming the conclusion (consequence)

This kind of fallacy has the form

q

$p \rightarrow q$

....

$\therefore p$

Hence it is a fallacy.

Example 11

If economy of Nepal is poor, then the education system in Nepal will be poor. The education system in Nepal is poor. Therefore, Economy of Nepal is poor.

In this argument above the conclusion can be false even if both the propositions "If economy of Nepal is poor, then the education system in Nepal will be poor" and "The education system in Nepal is poor" are true. Denoting with symbols we may write '$p \rightarrow q$' for first proposition and then the second proposition becomes q. this takes the form $q \wedge (p \rightarrow q) \rightarrow q$, which is not a tautology. Since the education system may not depend on the economy of the country.

i.e. $p \wedge (p \rightarrow q) \rightarrow q$. This is not a tautology

Begging the Question (Circular Reasoning)

If the statement that is used for proof is equivalent to the statement that is being proved then it is called circular reasoning.

Example 12

- The square root of 2 is irrational since it is not rational.
- Ram is black because he is black.
- Hari is student because he is a student.

Rules of Inference for Quantified Statements

If the sentence given in the argument represent the properties of more than one object then we can't represetn them by a simple propostion. we need predicate to represent suh statement. such statemnet sometimes called open proposition. For logical resoning that involed such statement needs rule of inference designed for quantified sentences.Some of the rules are:

1. Univasal Instantiation

If the proposition of the form $\forall x P(x)$ is supposed to be true then the universal quantifier can be dropped out to get P(c) is true for arbitrary c in the universe of discourse. This can be written as

In universe of discourse of all man every man is mortal implies ram is mortal where ram is a man.

2. Universal Generalization

If all the instances of c makes P(c) true, then $\forall x P(x)$ is true. This can be written as

P(c), for all c , Here the chosen c must be arbitrary, not a specific element from the universe of discourse. This rule is seldom explicitly used.

3. Existential Instantiation

If the proposition of the form $\exists x P(x)$ is supposed to be true then the there is an element c in the universe of discourse such that P(c) is true. This can be written as

$$\exists x P(x)$$

$\therefore$ *P(c), for some c.*,

Here the element c is not arbitrary, it must be specific such that P(x) is true. We generally find difficulty in finding such c.

4. Existential Generalization

If at least a element c from the universe of discourse makes P(c) true, then $\exists x P(x)$ is true.

Example 13

Reema, a student in this class, knows how to write programs in JAVA. Everyone who knows how to write programs in java can get a high paying job. Therefore, someone in this class can get a high paying job.

Solution

Let x represent a student in this class.

J(x) : x knows how to program in java.

Then,

J(Reema): Reema knows how to program in java.

Hp(x): x get high paying job.

Now, the above given argument in symbolic form becomes

(i) $\forall x, J(x) \rightarrow Hp(x)$

(ii) J (Reema)

and we have to prove $\exists x, Hp(x)$

Now, the steps are:

(1) $\forall \exists (x) \rightarrow Hp(x)$

(2) J (Reema) $\rightarrow$ H_p (Reema) - using universal instantiation in (i)

(3) J (Reema) - from given argument

(4) H_p (Reema) $\rightarrow$ Using modes process on (2) and (3)

Since we have H_p (Reema), we can now use existential generation rule as

H_p (Reema)

$\therefore$ For some c, H_p(C)

i.e. there is source 'C' who get high paying job. Hence proved.

Example 14

Everyone in descrete structure class has taken a course in computer science and Reema is a student in this class imply the conclusion Reema has taken a course in computer science.

Solution

Let D(x): x is a student in discrete structure class

C(x): x has taken a course in computer science

Then, the premise or given facts are

(a) $\forall$ I (x) $\rightarrow$ C(x)

(b) D (Reema)

and we have to prove that conclusion is C(Reema)

Now,

(1) $\forall$x, D(x) $\rightarrow$ C(x) - given

(2) D(Reema) $\rightarrow$ C(Reema) - using (vi)

(3) D(Reema) $\rightarrow$ given

(4) C(Reema) - using modus person on 2 & 3

Hence, it is concluded that Reema had taken a computer science course.

Example 15

Prove or disprove the validity of argument: "every living thing is a plant or an animal." "Hari's dog is alive and it is not a plant." "All animals have heart." Hence Hari's dog has heart.

Solution

Let,

P(x) : x is a planet

A(x) : x is an animal

L(x) : x is alive (living things)

H(x) : x has a heart

d : represent Hari's dog.

Then,

(1) $\forall$ x p(x) $\vee$ A(x) [given]

(2) L(d) $\wedge \neg$ P(d) [given]

(3) $\forall$ x, A(x) $\rightarrow$ H(x) [given]

(4) P(d) $\vee$ A(d) [universal insuntiation on (I)]

(5) $\neg$ P(d) (simplicity on rule on 2)

(6) A(d) (resolution rule)

(7) A(d) $\rightarrow$ H(d) (universal instantiation on 3)

(8) H(d) (modus persons on 6)

The conclusion "Hari's dog has heart" is valid.

Example 16

There is someone in this class who has visited Pokhara. Everyone who has been in Pokhara visits Fewalake. Therefore some one in this class has visited Fewalake.

Solution

Let P(x) : X has been in Pokhara.

F(x): X has visited Fewalake.

S(X): x is a student of this class.

Then,

(1)	$\exists$ x S(x) ^ P(x)	(given)
(2)	$\forall$ x P(x) $\therefore$ F(x)	(given)
(3)	S(C) ^ P(c)	(using existential instantion, on ID for some constant C)
(4)	P(C)	(using simple feature)
(5)	P(C) $\rightarrow$ F(C)	(Using universal instantiation on 2)
(6)	F(C)	(Using modus ponens on 4 and 5)
(7)	S(C) ^ F(x)	(using conjunction rule)
(8)	$\exists$ x S(x) ^ F(x)	(Using exestented generalization)

Methods of Proof

A theorem is a mathematical statement that can be shown to be true. A proof of given theorem is said to be well founded if its steps of mathematical statement can be present in argument that makes the theorem true.

This method of understanding correctness of statement by applying sequence of logical argument is known as proof of statement.

Problem solving or proving is not just a science so there is no hard and fast rule that is applied in problem solving. However there are some guiding methods that help us to solve different kinds of problems.

Here we will discuss different methods of proving implication p $\rightarrow$ q.

Direct Proofs

The implication p $\rightarrow$ q can be proved by showing that if p is true, then q must also be true. To carry out such a proof, we assume that hypothesis p is true and using information already available if conclusion q becomes true then argument becomes valid.

Scanned by CamScanner

Example 17

If a and b are odd integers, then a + b is an even integer.

Proof

We know the fact that if a number is even then we can represent it as 2k, where k is an integer and if the number is odd then it can be written as 2l + 1, where l is an integer. Assume that a = 2k + 1 and b = 2l + 1, for some integers k and m. then a + b = 2k + 1 + 2l + 1 = 2(k + l + 1), here (k + l + 1) is an integer. Hence a + b is even integer.

Example 18

Prove that: If n is an odd integer, then n^2 is an odd integer.

Solution:

Let $p \rightarrow q$: if n is an odd integer, then n^2 is an odd integer.

We assume that hypothesis of this implication is true i.e. suppose, n is odd then n can be expressed as n = 2k + 1

Now,

$$n^2 = (2k + 1)^2 = 4k^2 + 4k + 1$$
$$= 2(2k^2 + 2k) + 1$$

Since $2(2k^2 + 2k)$ is even but one more than even is odd

$\therefore$ n^2 is an odd integer.

Example 19

Prove that sum of two rational number is rational number.

Proof: Suppose that A and B are two rational number then by definition of rational number, it follows that there are two integers p and q where $q \neq 0$ such that $A = \frac{p}{q}$ and integers x and y with $y \neq 0$ such that $B = \frac{x}{y}$.

Now,

$$A + B = \frac{p}{q} + \frac{x}{y}$$
$$= \frac{py + ax}{ay}$$

Since: $q \neq 0$ and $y \neq 0$, it follows that $ay \neq 0$. Here, we have expressed A + B as the ratio of two integers py + qx and qy with $ay \neq 0$

$\therefore$ A + B is rational.

Example 20

Using direct proof, prove that for every positive integer n, $n^3 + n$ is even.

Proof: Case I: Suppose, n is even, then n = 2k for some k.

Now,

$$n^3 + n = (2k)^3 + 2k = 8k^3 + 2k = 2(4k^3 + k) \text{ which is even.}$$

Case II: Suppose n is odd, then it can be expressed as

n = 2k + 1 for some positive integer k.

Now,

$$n^3 + n = (2k + 1)^3 + (2k + 1)$$
$$= (8k^3 + 12k^2 + 6k + 1) + (2k + 1)$$
$$= 8k^3 + 12k^2 + 8k + 2$$
$$= 2(4k^2 + 6k^2 + 4k + 1)$$

which is even. Hence, for any positive integer n, $n^3 + n$ is even.

Indirect Proofs

We have $p \rightarrow q \equiv \neg q \rightarrow \neg p$ i.e. contra positive of implication is equivalent to the implication. So, the implication $p \rightarrow q$ can be proved by showing that its contrapositive $\neg q \rightarrow \neg p$ is true. We prove the implication $p \rightarrow q$ by assuming that the conclusion (q) is false and using the known facts we show that the hypothesis (p) is also false.

Example 21

If the product of two integers a and b is even, then either a is even or b is even.

Proof

Suppose both a and b are odd, then we have a = 2k + 1 and b = 2l + 1.

So ab = (2k +1) (2l + 1) = 4kl +2k +2l +1 = 2(2kl + k + l) + 1, i.e. ab is an odd number. Hence, both a and b being odd implies ab is also odd. This is indirect proof.

Example 22

Using indirect proof, show that if 3n + 2 is odd then n is odd.

Proof: Let $p \rightarrow q$: if 3n + 2 is odd then n is odd.

Assume that conclusion of this implication is false i.e. n is even then we can express n as:

n = 2k for some integer k

It follows that

$3n + 2 = 3 \times 2k + 2$

$= 6k + 2$

$= 2(3k + 1)$

$\therefore$ $3n + 2$ is even since $2(3k + 1)$ is multiple of 2.

Hence, if $3n + 2$ is odd then n is odd.

Trivial and Vacuous Proofs

In the implication $p \to q$, if we can show that the consequence q is true then regardless of truth values of p, the implication $p \to q$ is true. Such type of proof technique is called **trivial proof.**

In the implication $p \to q$, if we can show that the hypothesis p is false, then regardless of truth values of q, the implication $p \to q$ is true. Such kind of proof of an implication $p \to q$ is called **vacuous proof.**

Example 23

If x is an integer, then 3 is an odd integer. (Trivial) If a black is white, then pink is blue. (Vacuous)

Proofs by Contradiction

The steps in proof of implication $p \to q$ by contradiction are:

- Assume $p \wedge \neg q$ is true.
- Try to so that the above assumption $(p \wedge \neg q)$ is false
- When the assumption is found to be false then implication $p \to q$ is true
- Since $p \to q$ is equivalent to $\neg p \vee q$ and negation of $\neg p \vee q$ is $p \wedge \neg q$ (By De Morgan's Law), so if our assumption is false then its negation is true.

Alternately, Contradict the statement and show that this leads to the false conclusion; if this is true then the contradicted statement must be false (since $\neg p \to F$ is true only if $\neg p$ is false), hence the statement is true.

Example 24

If a^2 is an even number, then a is an even number.

Proof

Assume that a^2 is an even number and a is an odd number. Since a is an odd number we have

$a = 2k + 1$, for some integer k, so $a^2 = (2k + 1)^2 = 4k^2 + 4k + 1 = 2(k^2 + k) + 1$, here k^2 is some integer, say l, then $a^2 = 2l + 1$ i.e. a^2 is an odd number This contradicts our assumption that is a^2 even. Hence proved.

Example 25

Prove that if $n^3 + 5$ is odd, then n is even.

Solution:

Let: $p \to q$: if $n^3 + 5$ is odd then n is even

Suppose $n^3 + 5$ is odd and n is also odd.

$\therefore$ n can be expressed as:

$n = 2k + 1$ for some positive integer k.

$\therefore$ $n^3 + 5 = (2k + 1)^3 + 5$

$= 8k^3 + 12k^2 + 6k + 1 + 5$

$= 8k^3 + 12k^2 + 6k + 6$

$= 2(4k^3 + 6k^2 + 3k + 3)$

which is even.

This contradict our assumption that $n^3 + 5$ is odd.

Proof by Cases

The implication of the form $(p_1 \vee p_2 \vee \ldots \vee p_n) \to q$ can be prove by using the tautology $(p_1 \vee p_2 \vee \ldots \vee p_n) \to q \leftrightarrow [(p_1 \to q) \wedge (p_2 \to q) \wedge \ldots \wedge (p_n \to q)]$, i.e. we can show every implication

$(p_i \to q)$ true for $i = 1, 2, \ldots, n$.

Example 26

If $|x| > 3$, then $x^2 > 9$, where x is a real number.

Proof:

Here we have to consider two cases $-x > 3$ and $x > 3$ since $|x|$ represent an absolute value of x, we have the following relations

$|x| = x$, when $x \geq 0$ and

$-x$ when $x \leq 0$.

If $-x > 3$, then $x^2 > 9$. Similarly, if $x > 3$, then $x^2 > 9$.

Example 27

If $|x| > 3$ then $x^2 > 9$, where x is a real number.

Proof:

Here we have to consider two cases:

(a) $-x > 3$ (b) $x > 3$

as |x| is a absolute value of x.

Case 1: If $-x > 3$ then squaring both side gives us

$x^2 > 9$. Proved

Case 2: If $x > 3$ then $x^2 > 9$.

Example 28

Prove that if a and b are real numbers, where $b \neq 0$, then $\left|\frac{a}{b}\right| = \frac{|a|}{|b|}$

Proof; Assume that a and b are real numbers and $b \neq 0$, we consider the following cases:

Case 1: Assume that $a \geq 0$ and $b \geq 0$, so that $\frac{a}{b} \geq 0$, then $|a| = a$, $|b| = b$, and $\left|\frac{a}{b}\right| = \frac{a}{b} = \frac{|a|}{|b|}$

Case 2: Assume that $a \geq 0$ and $b < 0$, so that

$\frac{a}{b} \leq 0$ then $|a| = a$, $|b| = -b$ and

$\left|\frac{a}{b}\right| = -\frac{a}{b} = \frac{a}{-b} = \frac{|a|}{|b|}$

Case 3: Assume that $a < 0$ and $b > 0$, so that

$\frac{a}{b} < 0$, then $|a| = -a$, $|b| = b$, and

$\left|\frac{a}{b}\right| = -\frac{a}{b} = -\frac{a}{b} = \frac{|a|}{|b|}$

Case 4: Assume that $a < 0$ and $b < 0$, so that

$\frac{a}{b} > 0$. Then $|a| = -a$, $|b| = -b$, and

$\left|\frac{a}{b}\right| = \frac{a}{b} = \frac{-a}{-b} = \frac{|a|}{|b|}$

Thus, for all possible cases.

$\left|\frac{a}{b}\right| = \frac{|a|}{|b|}$

and $3(k^2 + k + 1)$ is also divisible by 3; is multiple of 3.

Sum of two integer divisible by 3 is also divisible by 3. Hence, $p(k + 1)$ is true.

Example 29

Prove that if n is a positive integer, then n is even if and only if $7n + 4$ is even.

Proof:

Assume n is even then we have an integer k such that $n = 2k$, so $7n + 4 = 7*2k + 4 = 2(7k + 2)$ here $7k + 4$ is an integer so that $7n + 4 = 2l$, where $l = 7k + 2$ i.e. $7n + 4$ is even. By direct proof it is proved that if n is even, then $7n + 4$ is even.

Assume n is odd then we have an integer m such that $n = 2m + 1$, then $7n + 4 = 14m + 7 + 4 = 2(7m + 5) + 1$here since $7m + 5$ is an integer $7n + 4$ is an odd number by indirect proof it is proved that if $7n + 4$ is even, then n is even.

Hence the proof.

Existence Proofs

A proof of a proposition of the form $\exists x P(x)$ is called an existence proof. There are different ways of proving a theorem of this type. Sometime some element a is found to show P(a) to be true, this is called constructive existence proof. In other method we do not provide a such that P(a) is true but prove that $\exists x P(x)$ is true in different way, this is called non constructive existence proof.

Example 30

Prove that there are 100 consecutive positive integers that are not perfect squares.

Proof:

Lets consider 2500 this is a perfect square of 50, and take 2601 this is a perfect square of 51. in between 2601 and 2500 there are 100 consecutive positive integers. Hence the proof.

Uniqueness Proofs

To prove the theorem that asserts the existence of unique element with particular property we must show that the element with this property exists and no other elements has this property. There are two parts in this uniqueness proof

Existence: here we show that the element with desire property exists

Uniqueness: we show that if $y \neq x$, then y does not have the desired property.

The above two steps can be proved if we prove the statement

$\exists x(P(x) \wedge \forall y(y \neq x \rightarrow \neg P(y)))$.

Example 31

Show that if a, b, and c are real numbers and $a \neq 0$, then there is a unique solution of the equation $ax + b = c$.

Proof:

From the equation $ax + b = c$ we get the solution as $x = (c - b)/a$ (since $a \neq 0$, it is possible). This solution is unique because there is no other value for x than $(c - b)/a$ (a real number).

Proofs By Counter Examples

To prove that the statement of the form $\forall x P(x)$ is false, we just need some value of x. So while proving for falsity we just look for counter example.

Example 32

Prove or disprove the product of two irrational numbers is irrational.

Proof:

Here we ir tantly try to get the product of the irrational to try it. Lets take both the number for produc. be $\sqrt{2}$ then we have $\sqrt{2} * \sqrt{2} = 2$ (not rational). Hence by counter example it is shown that the product of two irrational numbers is not necessarily irrational.

Solved Example

a) **Prove**

If n is an odd integer, than n^2 is also odd.

Proof: (Direct proof)

Assume n is odd. By dentition of oddness, there must exist some integer k such that

$n = 2k + 1$.

Then, $n2 = 4k^2 + 4k + 1 = 2(2k^2 + 2k) + 1$, which is odd.

Th;is, if n is odd, n^2 is also odd.b)

If n^2 is odd, then n is odd.

Proc i:

Supp ose n is even. Then, there exists integer k such that

n = [illegible].

Then $n^2 = (2k)^2 = 4k^2 = 2(2k^2)$

Thus, n^2 is also even.

c) For all n, if n is both odd and even, then $n2 = n + n$.

Proof:

The statement n is both odd and even is necessarily false, since no number can be both odd and even. So, the theorem is vacuously true.

d) (For integers n) If n is the sum of two prime numbers, then either n is odd or n is even.

Proof:

Any integer n is either odd or even. So the conclusion of the implication is true regardless of the truth of the antecedent. Thus the implication is true trivially.

Mathematical Induction

Mathematical induction is an extremely important proof technique that can be used to prove mathematical theorems or statements, to analyze the complexity of algorithms and correctness of computer programs etc. Here, induction means the method of inferring a general statement from the validity of particular cases.

Let P(n) be a statement on positive integer n then to prove P(n) is valid using induction, we need to follow following steps.

1. Basic Step:

In this step, we need to verify that P(n) is true for n = 0 or 1.

2. Induction Hypothesis:

In this step, we assume that P(n) is true for n = k i.e. P(k) is true.

3. Inductive step:

In this step, by using induction hypothesis, we prove that P(n) is true for n = k + 1

i.e. $[P(n_0) \wedge \forall k (P_k) \rightarrow P(k + 1)] \rightarrow \forall n, P(n)$

Example 33

Use mathematical induction to prove that $n^3 - n$ is divisible by 3whenever n is a positive integer.

Solution:

Let P(n) denote the proposition: $n^3 - n$ is divisible by 3.

(i) **Basis step:** Let P(n) is true for n = 1

i.e. P(1): $1^3 - 1 = 0$, is divisible by 3.

(ii) **Induction hypothesis**

Let P(n) is true for n = k

i.e. P(k): $k^3 - k$ is divisible by 3

(iii) Inductive step:

Using induction hypothesis, we try to show that P(n) is true for $n = k + 1$

Now,

$P(k + 1) = (k + 1)^3 - (k + 1)$

$= k^3 + 3k^2 + 3k + 1 - k - 1$

$P(k + 1) = (k^3 - k) + 3(k^2 + k)$

Since, both terms in this sum are divisible by 3. It follows that $(k + 1)^3 - (k + 1)$ is divisible by 3. Thus, by principle of mathematical induction, $n^3 - n$ is divisible by 3 whenever n is a positive integer.

Example 34

Prove by mathematical induction that,

$1 + 3 + 5 + \ldots\ldots + (2n - 1) = n^2$

Solution

Let $P(n) = 1 + 3 + 5 + \ldots\ldots + (2n - 1) = n^2$

1. Basis Step: For n = 1, we have

$P(1) = 1 = (1)^2$, hence P(1) is true.

2. Induction Hypothesis: Assume P(k) is true i.e.

$P(k) = 1 + 3 + 5 + \ldots\ldots + (2k - 1) = k^2$

3. Inductive Step: Now, we wish to show P(k + 1) is true. So adding (2k + 1) on both sides of P(n), then

$1 + 3 + 5 + \ldots\ldots + (2k - 1) + (2k + 1) = k^2 + (2k + 1) = (n + 1)^2$

$\therefore$ $P(n + 1) = (n + 1)^2$, is true

Thus, by mathematical induction P(n) is true for all n.

Example 35

Prove that 3 divides $n^3 + 2n$ whenever n is a nonnegative integer.

Solution

Let $P(n) = n^3 + 2n$, then

1. Basis Step: For n = 0, we have $n^3 + 2n = 0$, this is divisible by 3 hence the statement is true for n = 0.

2. Inductive Hypothesis: assume that the $P(k) = k^3 + 2k$ is divisible by 3 for all nonnegative values for $k \leq n$.

3. Inductive Step: here we are going to show that p(k+1) true. We have

$P(k+1) = (k+1)^3 + 2(k+1) = k^3 + 3k^2 + 3k + 1 + 2k + 2$

$= k^3 + 2k + 3k^2 + 3k + 3$

$= (k^3 + 2k) + 3(k^2 + k + 1)$ (since $k^3 + 2k$ is divisible by 3, by hypothesis and $3(k^2 + k + 1)$ is also divisible by 3, multiple of 3)

Hence, P(k+1) is divisible by 3.

So by mathematical induction $n^3 + 2n$ is divisible by three for all nonnegative integers n.

Example 36

Prove, for $n \geq 1$ and $a \neq 1$, that,

$$1 + a + a^2 + \ldots\ldots + a^n = \frac{a^{n+1} - 1}{a - 1}$$

Solution

Let $P(n) = 1 + a + a^2 + \ldots\ldots + a^n = \frac{a^{n+1} - 1}{a - 1}$

1. Basis Step: For n = 1, we have

$$P(1) = 1 + a = \frac{a^{1+1} - 1}{a - 1} = \frac{a^2 - 1}{a - 1} = a + 1$$

So, P(1) is true.

2. Induction Hypothesis: Suppose, P(k) is true i.e.

$$P(k) = 1 + a + a^2 + \ldots\ldots + a^k = \frac{a^{k+1} - 1}{a - 1}$$

Now, we wish to show P(k + 1) is true, for this we add a^{k+1} to both sides of P(n), then

$$1 + a + a^2 + \ldots + a^k + a^{k+1} = \frac{a^{k+1} - 1}{a - 1} + a^{k+1}$$

$$= \frac{a^{k+1} + a^{k+1}(a - 1)}{a - 1}$$

$$= \frac{a^{k+1} - 1 + a^{k+2} - a^{k+1}}{a - 1}$$

$$= \frac{a^{k+2} - 1}{a - 1}$$

$\therefore$ $P(k + 1) = \frac{a^{(k+1)+1} - 1}{a - 1}$, is true.

Thus P(n) is true for all n.

Scanned by CamScanner

Example 37

Use induction to show that $n! \geq 2^{n-1}$ for $n \geq 1$.

Solution

Let P (n) i the given statement.

1. Basis Step: Then for n = 1, $1! \geq 2^{1-1} = 1$, hence P(1) is true.

2. Induction Hypothesis: Assume that P(k) is true i.e. $k! \geq 2^{k-1}$.

3. Inductive Step: Now using induction hypothesis, we wish to show that the statement is true for n=k+1.

Now,

$$P(k+1): (k+1)! = (k+1)k!$$

$$\geq (k+1).2^{k-1} \; (\because k! \geq 2^{nk-1})$$

$$\geq 2.2^{k-1} \quad (\because k \geq 1)$$

$$= 2^{k-1+1}$$

$$= 2^k$$

$\therefore (k+1)! \geq 2^k$

Hence P(k + 1) true. Thus, by mathematical induction P(k) is true for all $n \geq 1$.

Example 38

Use mathematical induction to prove that $n < 2^n$ for all positive integer $n \geq 1$.

Solution:

Let P(n): $n < 2^n$ be a statement

1. Basis Step: For n = 1.

P(1) : $1 < 2^1$ i.e. 1 < 2 (True)

2. Induction hypothesis: Suppose P(n) is true for n = k i.e. P(k): $k < 2^k$ is true.

3. Inductive Step: By using induction hypothesis we show that P(n) is true n = k + 1.

$\therefore$ P(k + 1) : $k + 1 < 2^k + 1$

$\leq 2^k . 2^k$

$\leq 2. 2^{k+1}$

$k + 1 = 2^{k+1}$

Hence, for all positive integers n, $n < 2^n$

Example 39

Prove that: 1.1! + 2.2! + ... + n.n! = (n+1)! −1, whenever n is a positive integer.

Solution

Let P(n) = 1.1! + 2.2! + ... + n.n! = (n+1)! −1, then

1. **Basis Step:** for n = 1, we have P(1) = 1.1! = 1,

   Similarly P(1) = (1+1)! −1 = 2-1 = 1

   Hence P(1) is true.

2. **Inductive Hypothesis:** Assume that P(n) is true, i.e. 1.1! + 2.2! + ... + n.n! = (n+1)! − 1.

3. **Inductive Step:** if we are able to prove that P(n+1) is true then we are done. So we have

   P(n+1) = 1.1! + 2.2! + ... + n.n! + (n+1)(n+1)!

   = (n+1)! −1 + (n+1)(n+1)! (Using induction hypothesis)

   = (n+1)n! + (n+1)(n+1)! − 1 = (n+1) (n! + (n+1)!) −1

   = (n+1)(n! (1+ (n+1)) −1 = (n+1)n! (n+2) −1

   = (n + 2)! −1

P(n+1) is true

Hence P(n) is true for all positive integers.

Example 40

Prove that for any positive integer n, 1 + 2 + ... + n = n(n+1)/2.

Solution

Let's let P(n) be the statement "1 + 2 + ... + n = n (n+1)/2." (The idea is that P(n) should be an assertion that for any n is verifiably either true or false.)

1. Basis Step: We must verify that P(1) is True. P(1) Asserts "1 = 1(2)/2", which is clearly true.

So we are done with the initial step.

2. Induction Hypothesis: Let P(n)is true for n=k

i.e. P(k): 1+ 2+3+4+...................+k=k(k+1)/2

3. Inductive Step: Here we must prove the following assertion: "If there is a k such that P(k) is true, then (for this same k) P(k+1) is true." Thus, we assume there is a k such that 1 + 2 + ... + k = k (k+1)/2. (We call this the inductive assumption.) We must prove, for this same k, the formula 1 + 2 + ... + k + (k+1) = (k+1)(k+2)/2.

This is not too hard: 1 + 2 + ... + k + (k+1) = k(k+1)/2 + (k+1) = (k(k+1) + 2 (k+1))/2 = (k+1)(k+2)/2. The first equality is a consequence of the inductive assumption.

Strong Induction

The strong induction uses the save concept as in the mathematical induction method of proof but the only difference between these two types of induction is on the inductive step. In strong induction we as the inductive step as: P(J) is true and show that P(k + 1) must also be true.

This is also known as second principle of mathematical induction.

The formal definition of strong induction can be written as:

To prove P(n) is true for all positive integers n, we use.

(1) **Basis step:** The proposition P(1) is shown to be true.

(2) **Inductive step:** It is shown that $[P(1) \wedge P(2) \ldots\ldots P(K)] \rightarrow P(k+1)$ is true for every positive integer k.

Example 41

Show that if n is an integer greater than 1 then n can be written as products of princes.

Solution:

Let P(n) is the proposition that 'n' can be written as the product of primes.

Basis step: Here n = 2, P(2) is true, since 2 can be written as product of one prime, itself.

Inductive step: Let us suppose that P(j) is true for all positive integer $J \leq k$. Then, we have to show that P(k + 1) is true.

Now, there may be two cases:

(a) When k + 1 is prime (b) when k + 1 is composite.

If k + 1 is prime, then P(k + 1) is immediately true any prime can be written as a product of prime. Otherwise 'k+1' is composite and can be written as product of two positive integer x and y with $2 \leq x \leq y \leq k+1$.

By induction hypothesis, both x and y < k can be written as product of primes. Hence, if k + 1 is composite, it can be written as product of primes namely, the prime factors of x and prime factors of y.

Example 42

Use mathematical induction to show that '3' divides $n^3 + 2n$ whenever 'n' is a non-negative integer.

Solution

Let P(n) : $n^3 + 2n$ is divisible by 3.

Basis step: Here n = 0, then P(0) : $0^3 + 2 \times 0 = 0$, which is divisible by 3 i.e. P(0) is true.

Inductive step: Let us suppose that, $n^3 + 2n$ is divisible by 3, for all $n \leq k$.

i.e. P(k): $k^3 + 2k$ is divisible by 3.

Now, we have to prove that P(k+1) is true. For this,

$$p(k+1) : (k+1)^3 + 2(k+1)$$
$$= k^3 + 3k^2 + 3k + 1 + 2k + 2$$
$$= k^3 + 2k + 3k^2 + 3k + 3$$

Since $k^3 + 2k$ is divisible by 3 (from hypothesis).

Recursive Definition

Sometime it is difficult to define a relation directly but it may be easy to define this relation in terms of it this process is called recursion. The most common application of recursion is in mathematics and computer science, in which it refers to a method of defining functions in which the function being defined is applied within its own definition

Recursively defined functions

Let us suppose the domain of function is set of non negative numbers, then We use two steps to define a function:

1. **Basis step:** specify the value of function at 0.
2. **Recursive steps:** give a rule for finding its value at an integer from its value at smaller integers. Such a definition is called recursive definition.

Example 43

(i) Let the given sequence is 2, 5, 8, 11, 14, 17, ...

Then the n^{th} term of this sequence can be defined explicitly as

$t_n = a + (n-1)d, n \geq 1.$

And the same sequence can be defined recursively as :

(a) basis step: S(0) = 2

(b)Recursive Step: $S(n+1) = S(n) + 3, n \geq 0$

(ii) The sequence 1, 3, 9, 27,.......can be defined recursively as

(a)Basis step: S(0) = 1

(b)Recursive Step: $S(n+1) = 3\,S(n)$, for $n \geq 0$.

Example 44

Let function f is defined as

F(0)=2

F(n+1)=2f(n)+1

Then find the value of f(1),f(2) and f(3).

Solution

From the recursive definition, we have

F(1)=2f(0)+1=2*2+1=5

F(2)=2(1)+1=2*5+1=11

F(3)=2f(2)+1=2*11+1=23

Recursive Algorithms

An algorithm is called recursive If it solves a problem by reducing it into instance of the same problem with smaller input.

Example 5

An algorithm to find the factorial of given numebr

Input: an integer number n

Algorithm:

```
factorial( n) //name of algorithm
1. Start
2. If n==0
      then f=1
   Else
      f=n*factorial(n-1)
3. Stop
```

Example 46

A recursive algorithms for a^n

The power function can be defined recursively as:

a) Base case: $a^0=1$

b) Recursive definition: $a^n=a.a^{n-1}$ for $a>1$.

```
power (a,n)
1.  start
2.  if(n==0)
        return 1;
    else
    return a*power(a,n-1)
3.  End
```


EXERCISE

1. What rule of inference is used in each of these arguments?
   a. If it rains today, the college will close. The college is not closed. Therefore, it did not rain today.
   b. Ram is volleyball player and Basketball player. Therefore Ram is volleyball player.
   c. Ashma is an excellent swimmer. If Ashma is an excellent swimmer then she can work as a lifeguard. Therefore, Ashma can work as a lifeguard.
   d. If I work all night on this homework, then I can answer all the exercises. If I answer all the exercises, I will understand the material. Therefore, if I work all night on this homework, then I will understand the material.
2. What rule of inferences are used in following?
   a. All men are mortal. Socrates is a man. Therefore, Socrates is mortal.
3. For each of the following, determine whether the argument is correct or incorrect and explain why?
   a. All students in this class understand C-programming. Rina is student of this class. Therefore, Rina understand C-programming.
   b. All Parrots like fruits. My pet bird is not a parrot. Therefore, my pet bird does not like fruit.
   c. Every computer science major takes discrete structure. John is taking discrete structure. Therefore, John is computer science major.
4. Validate the following argument:
   a. If Loknath is a student then he has an internet account. Loknath doesn't have internet account. Therefore, Loknath is not a student.
   b. If I go swimming then I will stay in the sun too long. If I stay in the sun too long then I will sunburn. Therefore, if I go swimming then I will sunburn.
   c. If Bhairav play hockey too much then he will get low marks. Bhairav didn't get low marks. Therefore, Bhairav didn't play hockey.
5. Show that 'q' is a valid conclusion from the premises $p \rightarrow q, q \rightarrow r, r \rightarrow s, \neg s$ and p
6. State which rule of inference is basis of the following argument. "It is below freezing and raining now, therefore, it is below freezing now.
7. Explain the 4-rules of inference for quantified statements.
8. Give examples of addition rule and simplification rule of inference.
9. What relevant conclusion can be drawn from given premises?
   a. Every student has an internet account. Ram does not have an internet account. Hari has an internet account.

b. I am either cleaver or lucky. I am not lucky. If I am lucky then I will win the lottery.

10. Prove that square of an even number is also even.
11. Prove that sum of two odd integers is even (direct proof)
12. Prove that the product of two odd integers is odd.
13. Prove that the product of two rational number is rational.
14. Prove that if n is an integer and 3n+2 is even then n is even.
15. Prove that the square of an even number is an even number using:
    a. Direct proof
    b. Indirect proof
    c. Proof by contradiction
16. Prove that if n is an integer and $n^3 + 5$ is odd, then n is even using indirect proof and proof by contradiction.
17. Prove that if n is an integer and 3n + 2 is even, then n is even using indirect proof and proof by contradiction.
18. Prove that product of two rational number is rational.
19. Explain the 2-rule of inference for quantifies statements and give suitable example.
20. Explain the method of providing theorem by direct, indirect, contradiction and by cases.
21. Discuss the technique of proof by contradiction and by cases with suitable example.
22. Prove that sum of first n even positive integer is n(n+1)
    i.e. 2+4+6+......................2n=n(n+1)
23. Use mathematical induction to prove that
    $1+2+2^2+2^3+......................+2^n=2^{n+1}-1$
24. Use mathematical induction to prove the inequality n < 2n for all positive integers.
25. Use mathematical induction to prove that $2^n < n!$ for every positive integer n with $n \geq 4$.
26. Use mathematical induction to show that
    $1^2+2^2+3^2+......................n^2 = \frac{n(n+1)(2n+1)}{6}$, where n is positive integer
27. Use mathematical induction to prove that
    $1^2+3^2+5^2......................(2n+1)^2 = \frac{(n+1)(2n+1)(2n+3)}{3}$, where n is non- negative integer
28. Using mathematical induction prove that
    $1.2 + 2.3 + 3.4 + ... + n(n+1) = \frac{1}{3} n(n+1)(n+2)$ for all $n \in N$

• FSM – Mostly deterministic
Automata – Non-deterministic (variant)

3
Chapter

Regular Expression, Grammar and Language

Basic concepts of Automata Theory

The basic terms that pervade the theory of automata include 'alphabets", "strings", "languages", etc.

Alphabets (Represented by 'Σ')

Alphabet is a finite non-empty set of symbols. The symbols can be the letters such as {a, b, c}, bits {0, 1}, digits {0, 1, 2, 3... 9}. Common characters like S, #, etc.

{0,1} – Binary alphabets

{+, –, *} – Special symbols

Strings (Strings are denoted by lower case letters)

String is a finite sequence of symbols taken from some alphabet. E.g. 0110 is a string from binary alphabet, "automata" is a string over alphabet {a, b, c ... z}.

Empty String

It is a string with zero occurrences of symbols. It is denoted by 'ε' (epsilon).

Length of String

The length of a string w, denoted by | w |, is the number of positions for symbols in w. we have for every string s, length (s) ≥ 0.

| ε | = 0 as empty string have no symbols.

| 0110 | = 4

Power of Alphabet

The set of all strings of certain length k from an alphabet is the k^{th} power of that alphabet.
i.e. $\sum^k = \{w / |w| = k\}$

If $\sum = \{0, 1\}$ then,

$\sum^0 = \{\epsilon\}$
$\sum^1 = \{0, 1\}$
$\sum^2 = \{00, 01, 10, 11\}$
$\sum^3 = \{000, 001, 010, 011, 100, 101, 110, 111\}$

Kleen Closure

The set of all the strings over an alphabet $\sum$ is called Kleen closure of $\sum$ and is denoted by $\sum^*$. Thus, Kleen closure is set of all the strings over alphabet $\sum$ with length 0 or more.

$\therefore \sum^* = \sum^0 \cup \sum^1 \cup \sum^2 \cup \sum^3 \cup$ ....

E.g. $A = \{0\}$
$A^* = \{0^n / n = 0, 1, 2, \ldots\}$

Example

Determine Kleen closer of $A = \{0\}$, $B = \{0, 1\}$, $C = \{11\}$

Solution:

$A^* = \{\epsilon, 0, 00, 000, \ldots\}$
$B^* = \{\epsilon, 0, 1, 00, 000, 1, 11, 01, 10, \ldots\}$
$C^* = \{\epsilon, 11, 1111, 111111, \ldots\}$

Positive Closure

The set of all the strings over an alphabet $\sum$, except the empty string is called positive closure and is denoted by $\sum^+$.

$\therefore \sum^+ = \sum^1 \cup \sum^2 \cup \sum^3 \cup$ ....

Language

A language L over an alphabet $\sum$ is subset of all the strings that can be formed out of $\sum$, i.e. a language is subset of Kleen closure over an alphabet $\sum$; $L \subseteq \sum^*$. Set of strings chosen from $\sum^*$ defines language.

Example 2

- Set of all strings over $\sum = \{0, 1\}$ with equal number of 0's and 1's.
  $L = \{\epsilon, 01, 0011, 000111, \ldots\ldots\}$
- ϕ is an empty language and is a language over any alphabet.
- $\{\epsilon\}$ is a language consisting of only empty string.
- Set of strings representing binary numbers whose decimal value is a prime.
  $L = \{10, 11, 101, 111, 1011, \ldots\ldots\}$

Concatenation of Strings

Let x and y be strings then xy denotes concatenation of x and y. i.e. the string formed by making a copy of x and following it by a copy of y. More precisely, if x is the string of i symbols as $x = a_1a_2a_3 \ldots a_i$ and y is the string of j symbols as $y = b_1b_2b_3 \ldots b_j$ then xy is string of $i + j$ symbols as $xy = a_1a_2a_3 \ldots a_ib_1b_2b_3 \ldots b_j$.

For example;

$x = 000$
$y = 111$
$xy = 000111$ and
$yx = 111000$

Note: 'ϵ' is identity for concatenation; i.e. for any substring w over ϵ.

$\epsilon w = w\epsilon = w$

Suffix of a string

A string s is called a suffix of a string w if it is obtained by removing 0 or more leading symbols in w. For example;

$w = abcd$
$s = bcd$ is suffix of w.
s is proper suffix if $s \neq w$.

Prefix of a string

A string s is called a prefix of a string w if it is obtained by removing 0 or more trailing symbols of w. For example;

$w = abcd$
$s = abc$ is prefix of w.

Here, s is proper suffix i.e. s is proper suffix if $s \neq w$.

Substring

A string s is called substring of a string w if it is obtained by removing 0 or more leading or trailing symbols in w. It is proper substring of w if $s \neq w$.

If s is a string then $Substr(s, i, j)$ is substring of s beginning at i^{th} position and ending at j^{th} position both inclusive.

Example 3

Given, S = ab and ad over $\Sigma = \{a, b\}$ then substring (S, 2, 4) = baa.

Finite Automata

A finite automaton is a mathematical (model) abstract machine that has a set of "states" whose "control" moves from state to state in response to external "inputs". The control may be either "deterministic" meaning that the automation can't be in more than one state at any one time, or "non deterministic", meaning that it may be in several states at once. This distinguishes the class of automata as Deterministic Finite Automata (DFA) or Non-Deterministic Finite Automata (NFA). The DFA can't be in more than one state at any time. The NFA can be in more than one state at a time. The finite automaton may be generating output or it may not be.

The finite state machines are used in applications in computer science and data networking. For example, finite-state machines are basis for programs for spell checking, indexing, grammar checking, searching large bodies of text, recognizing speech, transforming text using markup languages such as XML and HTML, and network protocols that specify how computers communicate.

The finite state machines can be represented with state transition diagram or state transition table. A **state transition table** is a table showing what state finite state machine will move to, based on the current state and other inputs. The rows in the table indicate current states, the columns indicate input symbol, and the cells (row/column intersections) in the table contain the next state if an event happens. The state diagram is a directed graph with labeled edges. In this diagram, each state is represented by a circle. Arrows labeled with the input and/or output pair are shown for each transition in between the states represented by circle.

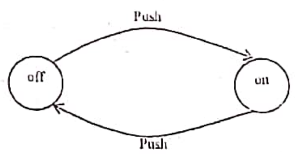


Figure above is state transition diagram showing finite automaton modeling an on/off switch

Switch	push
off	on
on	off

State transition table of automaton in figure above.

Finite State Machines in Sequential Circuits

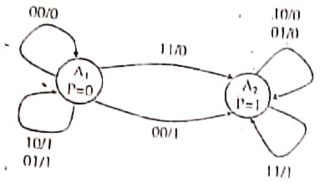
Logic block specifically are designed to sequence through specific patterns of states in a predetermined sequential manner. It is necessary to design circuits that perform specific sequences of operations. Finite State Machine has proven to be a very efficient means of modeling sequential circuits. Realizing the circuit requires memory element(s) to store state(s) and state is updated based with respect to clock.

Example: Representing Binary adder using Finite State Machine

Consider a binary adder, which adds two 2-digit binary strings. The addition is performed "bit by bit", starting from the least significant bit. Note that a carry in from the addition of previous bits should be taken into account. There are two bits at the input with possible combinations are 00, 01, 10, 11. The rules of addition are

- 0 + 0 + no carry in from previous operation results in 0 without new carry in
- 0 + 0 + carry in from previous operation results in 1 without new carry in
- 0 + 1 (or 1 + 0) + no carry in from previous operation results in 1 without new carry in
- 0 + 1 (or 1 + 0) + carry in from previous operation results in 0 with new carry in
- 1 + 1 + no carry in from previous operation results in 0 with new carry in to the more significant bit
- 1 + 1 + carry in from previous operation results in 1 with new carry in to the more significant bit

To represent this with finite state machine let us construct two internal states: A1 – an addition without carry in (p = 0) A2 – an addition with carry in (p = 1). Then the states can be illustrated as a finite state automaton as shown in following state diagram;



State diagram of a binary adder.

Finite State Machine with output

A finite state automaton with output is defined by a sextuple (6-tuple) as $(Q, \Sigma, O, \delta, f, q_0)$.

Where,

Q = Finite set of states,

Σ = Finite set of input alphabets,

O = Finite set of output alphabets,

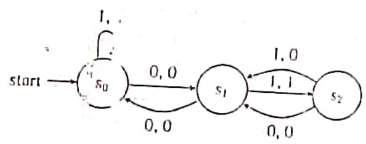
δ = A transition function that maps $Q \times \Sigma \rightarrow Q$,

f = An output function that maps $Q \times \Sigma \rightarrow O$

q_0 = A start state; $q_0 \in Q$

Example 4

Consider a finite state automaton with output as defined below;



State	δ		f	
	Input		Input	
	0	1	0	1
S_0	S_1	S_0	0	1
S_1	S_0	S_2	0	1
S_2	S_1	S_1	0	0

The output generated from the input string 01110 for the above finite-state machine is: 01010 and the state transition sequence is $S_0 \rightarrow S_1 \rightarrow S_2 \rightarrow S_1 \rightarrow S_2 \rightarrow S_1$.

Example 5

Consider a finite state automaton 'M' with following components;

$Q = \{q_0, q_1, q_2\}$

$\Sigma = \{a, b\}$

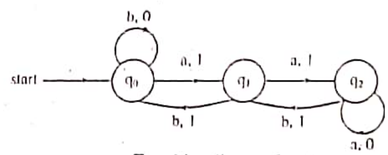
$O = \{0, 1\}$

$q_0 = q_0$

The transition function $f(Q \times \Sigma \rightarrow Q)$ and the output function $f(Q \times \Sigma \rightarrow O)$ is defined by following transition table and diagram;

state	δ		f	
	input		input	
	a	b	a	b
q_0	q_1	q_0	1	0
q_1	q_2	q_0	1	1
q_2	q_2	q_1	0	~~0~~ 1

Transition table for M



Transition diagram for M

For the input string baaa the automaton M generates the output 0110.

There are two types of finite state machines with outputs

- Mealy Machines and
- Moore Machines

Mealy machine is a finite-state machine whose output values are determined both by its current state and the current inputs. This is in contrast to a Moore machine, whose output values are determined solely by its current state. Thus, the Moore machine is a finite-state machine whose output values are determined solely by its current state.

The definition and examples under the topic **Finite State Machine with output** are the concepts of Mealy Machine.

Defining Moore Machines

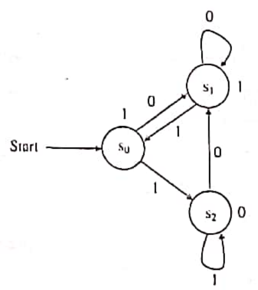
A Moore Machine is a finite state automaton with output is defined by a sextuple (6-tuple) as $(Q, \Sigma, O, \delta, f, q_0)$.

Where,

- Q = Finite set of states,
- Σ = Finite set of input alphabets,
- O = Finite set of output alphabets,
- δ = A transition function that maps $Q \times \Sigma \rightarrow Q$,
- f= An output function that maps $Q \rightarrow O$
- q_0 = A start state; $q_0 \in Q$

Example 6

A Moore machine defined with following transition diagram and table.



State	δ		f
	0	1	
S_0	S_1	S_2	1
S_1	S_1	S_0	1
S_2	S_1	S_2	0

Finite State Machines with no output

The finite state machines with no outputs are often known as finite state automata. It consists of a finite set of states, a finite input alphabet, a transition function that assigns a next state to every pair of state and input, an initial or start state, and a subset consisting of final or accepting states. The finite state automata can be either deterministic or non deterministic in nature.

Deterministic Finite Automata (DFA)

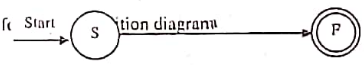
A deterministic finite automaton is defined by a quintuple (5-tuple) as $(Q, \Sigma, \delta, q_0, F)$.

Where,

- Q = Finite set of states,
- Σ = Finite set of input symbols,
- δ = A transition function that maps $Q \times \Sigma \rightarrow Q$
- q_0 = A start state; $q_0 \in Q$
- F = Set of final states; $F \subseteq Q$.

Example 7

A DFA defined with f… tion diagram



DFA having start state S and final state F, accepting a string "a".

General Notations of DFA

As discussed earlier, there are two preferred notations for describing this class of automata;

- Transition Table
- Transition Diagram

1) **Transition Table**

Transition table is a conventional, tabular representation of the transition function δ that takes the arguments from $Q \times \Sigma$ and returns a value which is one of the state of the automaton. The row of the table corresponds to the states while column corresponds to the input symbol. The starting state in the table is represented by $\rightarrow$ followed by the state i.e. $\rightarrow q$, for q being start state, whereas final state as $*q$, for q being final state. The entry for a row corresponding to state q and the column corresponding to input a, is the state $\delta(q, a)$.

For example:

1. Consider a DFA;

$Q = \{q_0, q_1, q_2, q_3\}$

$\Sigma = \{0, 1\}$

$q_0 = q_0$

$F = \{q_0\}$

$\delta = Q \times \Sigma \rightarrow Q$

Then the transition table for above DFA is as follows:

	0	1
* → q_0	q_2	q_1
q_1	q_3	q_0
q_2	q_0	q_3
q_3	q_1	q_2

This DFA accepts strings having both an even number of 0's and even number of 1's.

2. DFA accepting all strings over {0, 1} having substring 01.
   Let, Q = {q_0, q_1, q_2}, $\sum$ = {0, 1}, q_0 = q_0, F = { q_1}

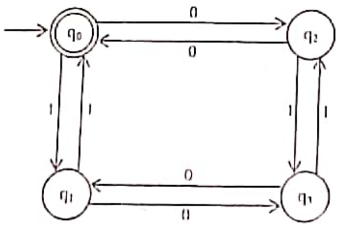
	0	1
→ q_0	q_2	q_0
*q_1	q_1	q_1
q_2	q_2	q_1

2) Transition Diagram

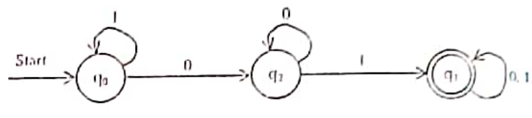
A transition diagram of a DFA is a graphical representation where; (or is a graph)

- For each state Q, there is a node represented by circle,
- For each state q in Q and each input a in $\sum$, if δ (q, a) = p then there is an arc from node q to p labeled a in the transition diagram. If more than one input symbol cause the transition from state q to p then arc from q to p is labeled by a list of those symbols.
- The start state is labeled by an arrow written with "start" on the node.
- The final or accepting state is marked by double circle.

For the example 1 considered previously, the corresponding transition diagram is:

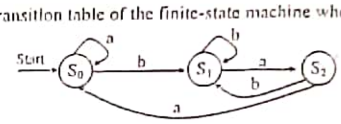


Similarly for example 2, the transition diagram is:



Example 8

Construct the state transition table of the finite-state machine whose digraph is shown.



Solution

The state transition table of above automation is

	a	b
→S_0	S_0	S_1
S_1	S_2	S_1
*S_2	S_0	S_1

How DFA process strings?

The first thing we need to understand about a DFA is how DFA decides whether or not to "accept" a sequence of input symbols. The "language" of the DFA is the set of all symbols that the DFA accepts. Suppose a_1, a_2, ... a_n is a sequence of input symbols. We start with the DFA in its start state, q_0. We consult the transition function δ also for this purpose. Say, δ (q_0, a_1) = q_1 to find the state that the DFA enters after processing the first input symbol a_1. We then process the next input symbol a_2, by evaluating δ (q_1, a_2); suppose this state be q_2. We continue in this manner, finding states q_3, q_4, ..., q_n such that δ (q_{i-1}, a_i) = q_i for each i. If q_n is a member of F, then input a_1, a_2, ..., a_n is accepted and if not then it is rejected.

Extended Transition Function of DFA

The extended transition function of DFA, denoted by $\hat{\delta}$ is a transition function that takes two arguments as input, one is the state q of Q and another is a string w$\in \sum^*$, and generates a state p $\in$ Q. This state p is that the automaton reaches when starting in state q and processing the sequence of inputs w.

i.e. $\hat{\delta}(q, w) = p$

Scanned by CamScanner

Let us define $\hat{\delta}$ by induction on length of input string as follows:

Basis step

$\hat{\delta}(q, \varepsilon) = q$ i.e. from state-q, reading no input symbol stays at the same state.

Induction

Let w be a string from $\sum^*$ such that w = xa, where x is substring of w without last symbol and *a* is the last symbol of *w*, then

$$\hat{\delta}(q, w) = \delta(\hat{\delta}(q, x), a)$$

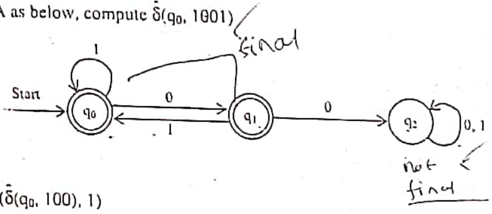
Thus, to compute $\hat{\delta}(q, w)$, we first compute $\hat{\delta}(q, x)$; the state the automaton is in after processing all but last symbol of w. let this state is p, i.e. $\hat{\delta}(q, x) = p$.

Then, $\hat{\delta}(q, w)$ is what we get by making a transition from state p on input a, the last symbol of w.

i.e. $\hat{\delta}(q, w) = \delta(p, a)$

Example 9

Given a DFA as below, compute $\hat{\delta}(q_0, 1001)$



Solution

$= \delta(\hat{\delta}(q_0, 100), 1)$

$= \delta(\delta(\hat{\delta}(q_0, 10), 0), 1)$

$= \delta(\delta(\delta(\hat{\delta}(q_0, 1), 0), 0), 1)$

$= \delta(\delta(\delta(\delta(\hat{\delta}(q_0, \varepsilon), 1), 0), 0), 1)$

$= \delta(\delta(\delta(\delta(q_0, 1), 0), 0), 1)$.

$= \delta(\delta(\delta(q_0, 0), 0), 1)$

$= \delta(\delta(q_1, 0), 1)$

$= \delta(q_2, 1)$

$= q_2$, *so accepted.*

Compute $\hat{\delta}(q_0, 101)$

$= \delta(\hat{\delta}(q_0, 10), 1)$

$= \delta(\delta(\hat{\delta}(q_0, 1), 0), 1)$

$= \delta(\delta(\delta(\hat{\delta}(q_0, \varepsilon), 1), 0), 1)$

$= \delta(\delta(\delta(q_0, 1), 0), 1)$

$= \delta(\delta(q_0, 0), 1)$

$= \delta(q_1, 1)$

$= q_0$, *so not accepted.*

String accepted by a DFA

A string x is accepted by a DFA (Q, $\sum$, δ, q_0, F) if; $\hat{\delta}(q, x) = p \in F$.

Language of DFA

The language of a DFA M = (Q, $\sum$, δ, q_0, F) denoted by L(M) is a set of strings over $\sum^*$ that are accepted by M.

i.e; $L(M) = \{w / \hat{\delta}(q_0, w) = p \in F\}$

That is, the language of a DFA is the set of all strings w that take DFA starting from start state to one of the accepting states. The language of DFA is called regular language.

Applications of DFA

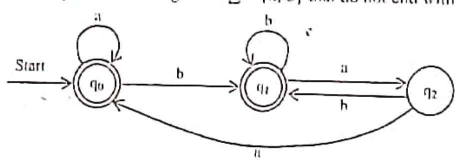
Deterministic finite automata have many practical applications as described below;

- Almost all compilers and other language-processing systems use DFA-like code to divide an input program into tokens like identifiers, constants, and keywords and to remove comments and white space.
- For many applications that accept typed commands, the command language is quite complex, almost like a little programming language. Such applications use DFA to process the input command.
- Text processors often use DFA to search a text file for strings that match a given pattern. This includes most versions of Unix tools like awk, egrep, and Procmail, along with a number of platform-independent systems such as MySQL.

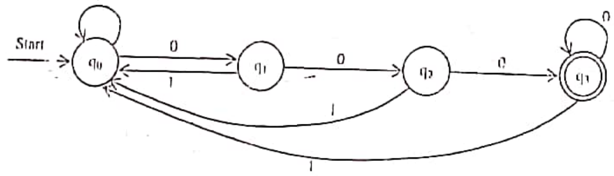
- Speech-processing and other signal-processing systems often use a DFA-like technique to transform an incoming signal.
- Controllers use DFA-like techniques to track the current state of a wide variety of finite-state systems, from industrial processes to video games. They can be implemented in hardware or in software. Sometimes, the DFA-like code in such applications is generated automatically by special tools such as lex.

Examples 10

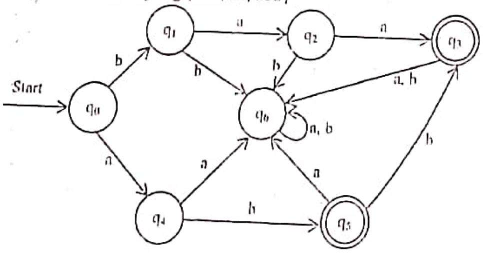
(a) DFA, that accepts all the strings over $\Sigma = \{a, b\}$ that do not end with ba.



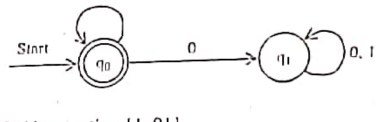
(b) DFA accepting all string over $\Sigma = \{0, 1\}$ ending with 3 consecutive 0's.



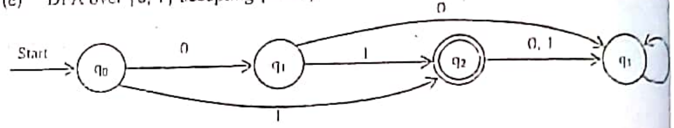
(c) DFA over {a, b} accepting {baa, ab, abb}



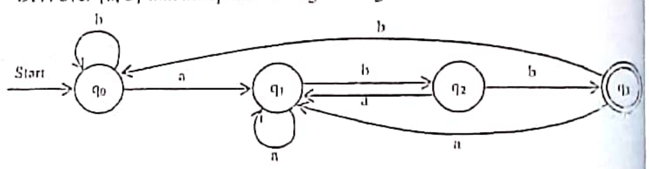
(d) DFA accepting zero or more consecutive 1's. i.e. L (M) = $\{1^n / n = 0, 1, 2, ...\}$



(e) DFA over {0, 1} accepting {1, 01}

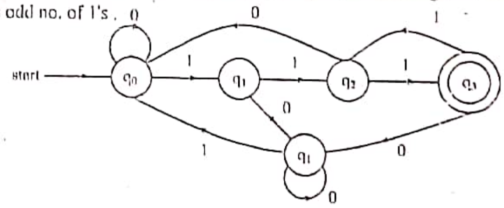


(f) DFA over {a, b} that accepts the strings ending with abb.



Example 11

Construct a DFA over $\Sigma = \{0, 1\}$ whose language is set of strings that ends with 111 and contains odd no. of 1's.



Non-Deterministic Finite Automata (NFA)

The finite automaton in which the exact state to which the machine moves cannot be determined is called a non-deterministic finite automaton.

A non-deterministic finite automaton is a mathematical model that consists of;

- A finite set of states Q
- A finite set of input symbols Σ, (alphabets)
- A transition function that maps state symbol pair to sets of states.
- A state $q_0 \in Q$, that is distinguished as a start (initial) state.
- A set of final states F distinguished as accepting (final) state. $F \subseteq Q$.

Thus, NFA can also be interpreted by a quintuple; $(Q, \Sigma, \delta, q_0, F)$, where δ is $Q \times \Sigma = 2^Q$. Here the power set of Q (2^Q) has been taken because in case of non-deterministic finite automata, from a state, transition can occur to any combination of Q states.

Unlike DFA, a transition function in NFA takes the NFA from one state to several states just with a single input. For example;

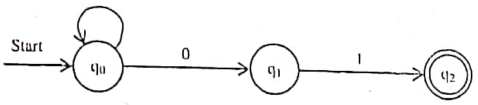
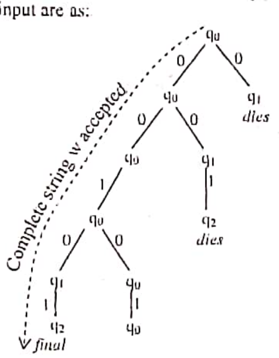


Fig NFA accepting all strings that end in 01.

Here, from state q_1, there is no any arc for input symbol 0 and no any arc out of q_2 for 0 and 1. So, we can conclude in a NFA, there may be zero number of arcs out of each state for each input symbol. While in DFA, it has exactly one arc out of each state for each input symbol. For input sequence w = 00101, the states the NFA can be in during the processing of the input are as:



$Q = \{q_0, q_1, q_2\}$

$\Sigma = \{0, 1\}$

$q_0 = \{q_0\}$

$F = \{q_2\}$

The equivalent transition table for the above mentioned NFA is

δ:	0	1
$\rightarrow q_0$	$\{q_0, q_1\}$	$\{q_0\}$
q_1	$\{\phi\}$	$\{q_2\}$
$*q_2$	$\{\phi\}$	$\{\phi\}$

|w| = 1001011 = 5

Language of NFA

The language of NFA, $M = (Q, \Sigma, \delta, q_0, F)$, denoted by L (M) is;

$L(M) = \{w / \hat{\delta}(q, w) \cap F \neq \phi\}$

i.e. L (M) is a set of strings w in Σ^* such that $\hat{\delta}(q, w)$ contains at least one accepting state.

Where $\hat{\delta}$ is the extended transition function of NFA

Example 12

When two finite state automata are equivalent?

Solution

Any two finite state automata say m_1 and m_2 are equivalent if $L(m_1) = L(m_2)$

i.e. if the strings processed by m_1 can also be processed by m_2.

In other words, for every string $w \in \Sigma$ over which m_1 and m_2 are defined, if m_1 and m_2 both leads to acceptance of w then m_1 and m_2 are equivalent.

Difference between NFA and DFA

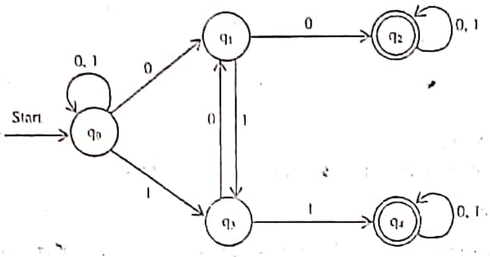
DFAs and NFAs accept exactly the same set of languages. That is, non-determinism does not make a finite automaton more powerful. Any language accepted by a NFA can also be accepted by some DFA and vice versa. Thus DFA and NFA have the same capability.

The only difference between a deterministic finite automaton and a non-deterministic finite automaton is found in the transition function. A NFA's transition function is less restrictive than a DFA's because it allows us to have several transitions from a given state to zero, one or more states for the same input symbol. On the other hand, a DFA specifies exactly one state that may be entered for a given state and input symbol combination. Thus in DFA, for a given state, on a given input, we reach to a deterministic and a unique state. On the other hand, in NFA, we may lead to more than one state for a given input. The DFA is subset of NFA.

Example 13

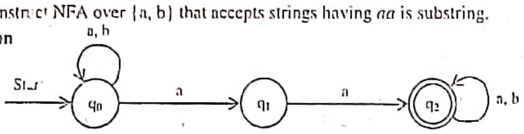
a. Construct a NFA for the language over {0, 1} that have at least two consecutive 0's or 1's.

Solution



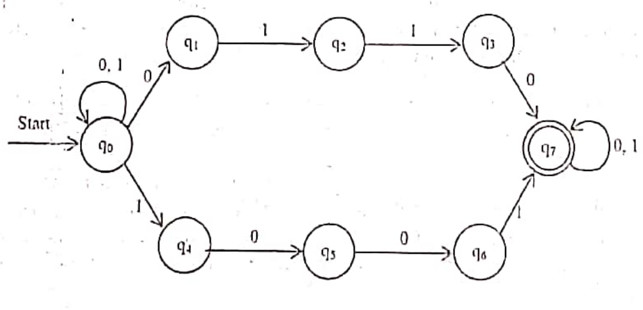
b. Construct NFA over {a, b} that accepts strings having *aa* is substring.

Solution



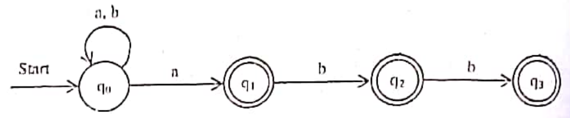
c. Construct NFA for strings over {0, 1} that contain 0110 or 1001.

Solution



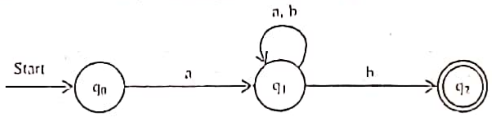
d. Construct NFA over {a, b} that have a as one of the last 3 characters.

Solution



e. Construct a NFA over {a, b} that accepts strings stating with *a* and ending with *b*.

Solution



Regular Expression

Regular Expressions are those algebraic expressions used for representing regular languages, the languages accepted by finite automaton. Regular expressions offer a declarative way to express the strings we want to accept. This is what the regular expressions offer that the automata do not.

A regular expression is built up out of simpler regular expression using a set of defining rules. Each regular expression 'r' denotes a language L(r). The defining rules specify how L(r) is formed by combining in various ways the languages denoted by the sub-expressions of 'r'.

Let Σ be an alphabet, the regular expression over the alphabet Σ are defined inductively as follows:

- Φ is a regular expression representing empty language.
- ε is a regular expression representing the language of empty strings.
- if 'a' is a symbol in Σ, then 'a' is a regular expression representing the language {a}.
- if 'r' and 's' are the regular expressions representing the language L(r) and L(s) then
  - r U s is a regular expression denoting the language L(r)UL(s).
  - r . s is a regular expression denoting the language L(r).L(s).
  - r* is a regular expression denoting the language(L(r))*.

- (r) is a regular expression denoting the language (L(r))[L(r)] [Defines parenthesis may be placed around regular expressions if we desire].

Regular operators:

Basically, there are three operators that are used to generate the languages that are regular. They are:

Union (U/|):

If L_1 and L_2 are any two regular languages then, $L_1 U L_2 = \{s/s \in L_1, \text{ or } s\ L_2\}$

e.g.

$L_1 = \{00, 11\}$, $L_2 = (\epsilon, 10)$

$L_1 U L_2 = \{\epsilon, 00, 11, 10\}$

Concatenation (.):

If L_1 and L_2 are any two regular languages then, $L_1.L_2 = \{l_1.l_2 | l_1 \in L_1 \text{ and } l_2 \in L_2\}$

e.g. $L_1 = \{00, 11\}$ and $L_2 = \{\epsilon, 10\}$

Kleen Closure (*) and Positive closure (+):

If L is any regular Language then Kleen closure of L is,

$$L^* = \bigcup_{i=0}^{\infty} = L^0 \cup L^1 \cup L^2 \cup \ldots\ldots\ldots\ldots$$

and positive closure of L is, $L^+ = \bigcup_{i=0}^{\infty} L^i = L^* - L^0$

Precedence of regular operators:

- Closure (*) has highest precedence
- Concatenation (.) has next highest precedence.
- Union (U / | / +) has lowest precedence.

Regular language:

Let Σ be an alphabet, the class of regular language over Σ is defined inductively as;

- Φ is a regular language representing empty language
- $\{\epsilon\}$ is a regular language representing language of empty strings.
- For each $a \in \Sigma$, $\{a\}$ is a regular language.
- If $L_1, L_2 \ldots\ldots\ldots L_n$ is regular languages, then so is $L_1 \cup L_2 \cup \ldots\ldots\ldots\ldots \cup L_n$.
- If $L_1, L_2, L_3, \ldots\ldots\ldots\ldots\ldots L_n$ are regular languages, then so is $L_1.L_2.L_3 \ldots\ldots\ldots L_n$
- If L is a regular language, then so is L^*

belongs to (E)

Every language can be written as an expression using the operations of union, concatenation and kleen closure. To simplify writing of these formulas, we adopt following conventions;

- Writing a symbol, say 'a' by itself is shorthand for {a}. That is, we promote a symbol to the singleton set containing it.
- The concatenation symbol . can be dropped as, in xy instead of x.y.
- Parentheses need be used only when it is necessary to override the normal precedence of * over . over U.

Each of these formulas is called as regular construction and by definition of the regular language are exactly those languages that are generated by regular construction.

Similarities in regular expressions and automatons

An automaton defines a pattern, namely the set of strings labeling paths from the initial state to some accepting state in the graph of the automaton. Regular expressions are an algebraic way to define those patterns. Regular expressions are analogous to the algebra of arithmetic expressions. Interestingly, the set of patterns that can be expressed in the regular-expression algebra is exactly the same set of patterns that can be described by automata. A regular expression is one way of describing a finite-state automaton (FSA). Both regular expressions and finite automaton are used to describe regular language.

Regular expressions and finite automata have equivalent expressive power;

- For every regular expression R, there is a corresponding Finite Automaton that accepts the set of strings generated by R.
- For every Finite Automaton A, there is a corresponding regular expression that generates the set of strings accepted by A.

Applications of Regular Expressions

- **Validation:** Determining that a string complies with a set of formatting constraints. Like email address validation, password validation etc.
- **Search and Selection:** Identifying a subset of items from a larger set on the basis of a pattern match.
- **Tokenization:** Converting a sequence of characters into words, tokens (like keywords, identifiers) for later interpretation.

Example 14

a. Consider $\Sigma = \{0, 1\}$, some regular expressions over Σ;

(i) $0^*10^* = \{w | w \text{ contains a single } 1\}$

(ii) $\Sigma^*1\Sigma^* = \{w | w \text{ contains at least one } 1\}$

- (iii) $\Sigma^*001\Sigma^*$ = {w|w contains the string 001 as substring}
- (iv) $(\Sigma\Sigma)^*$ or $((0+1)^*.(0+1)^*)$ = {w|w is string of even length}
- (v) $1^*(01^*01^*)^*$ = {w|w is string containing even number of zeros}
- (vi) $0^*10^*10^*10^*$ = {w|w is a string with exactly three 1's}
- (vii) For string that have substring either 001 or 100, the regular expression is $(1+0)^*.001.(1+0)^*+(1+0)^*.(100).(1+0)^*$
- (viii) For strings that have at most two 0's with in it, the regular expression is $1^*.(0+\epsilon).1^*.(0+\epsilon).1^*$
- (ix) For the strings ending with 11, the regular expression is $(1+0)^*.(11)^+$

b. Consider Σ = {a, b}, some regular expressions over Σ;

- (i) $(a + b)^*$ is a regular expression representing set of strings of a's and b's of any length including the null string.
- (ii) $(a + b)^*abb$ is a regular expression representing set of strings of a's and b's ending with the string abb
- (iii) $(aa)^*(bb)^*b$ is a regular expression representing set of strings consisting of even number of a's followed by odd number of b's

c. Regular expression that denotes the C identifiers:

(alphabet + _) (alphabet + digit + _)*

Language and Grammars

Natural language and Formal language

Natural languages are the languages that people speak, such as English, Nepali etc. They were not designed by people (although people try to impose some order on them); they evolved naturally. It does not seem possible to specify all rules of syntax for a natural language.

The formal language unlike a natural language is specified by a well defined set of rules of syntax. Formal languages are languages that are designed by people for specific applications. For example, the notation that mathematicians use is a formal language that is particularly good at denoting relationships among numbers and symbols. A formal language is a set of strings of symbols that may be constrained by rules that are specific to it. A formal language L over an alphabet Σ is a subset of Σ^*, that is, a set of words over that alphabet. Sometimes the sets of words are grouped into expressions, whereas rules and constraints may be formulated for the creation of 'well-formed expressions'.

Syntax and Semantics

A Syntax or Grammar for a language L is a set of rules through which the strings of may either be generated, or through which any element of A^* can be determined to be a element of S(A*). The syntax of a language determines the form of a sentence in the language.

Semantics is the structure of a language L that gives it meaning relating it to its model referents; it is an interpretation of the language L in terms of some model M.

Phrase Structured Grammar

Phrase structure grammar is a type of generative grammar, consisting of the rules determining the structure and interpretation of sentences, in which constituent structures are represented by phrase structure rules or rewrite rules. The underlying structure of a sentence or a phrase is sometimes called its phrase structure or phrase marker. Phrase-structure rules provide us with the underlying syntactic structure of sentences we both produce and comprehend.

Phrase structured grammars are used to generate the words of a language and to determine whether a word is in a language. Formal languages, which are generated by grammars, provide models for both natural languages such as, English, and for programming languages such as C, JAVA.

Formal Description of Phrase Structured Grammar:

A grammar G is defined by 4-tuples (V, T, P, S) where,

V= a vocabulary

T= set of terminal symbols

P= set of rules and productions

S= start symbol and S $\in$ V

Thus, every grammar consists of a collection of substitution rules, also called productions, with each rule being a variable and a string of variables and terminals. Each rule appears as a line in the grammar.

The symbols involved in the productions may be variable or terminal symbols. The variable symbols are after represented by capital letters. The terminals are analogous to the input alphabet and are often represented by lower case letters. One of the variables is designed as a start variable. It usually occurs on the left hand side of the topmost rule.

For example, a phrase structured grammar defined by $G = \{V, T, S, P\}$, where $V = \{a, b, A, B, S\}$, $T = \{a, b\}$, S is the start symbol, and $P = \{S \rightarrow ABa, A \rightarrow BB, B \rightarrow ab, B \rightarrow b\}$.

Example 15

What is the language of following grammar?

$S \rightarrow \varepsilon$

$S \rightarrow 0S_1$

Solution

This is a grammar defining the language of all the strings with equal no of 0's followed by equal no of 1's.

Here, the two rules define the production P,

ε, 0, 1 are the terminals defining T,

S is a variable symbol defining V

And S is start symbol from where production starts.

Example 16

Construct a grammar representing the language over $\Sigma = \{a, b\}$ which is palindrome language

Solution

$S \rightarrow \varepsilon |a| b$

$S \rightarrow a S a$

$S \rightarrow b S b$

Example 17

Let C_1 be the grammar with vocabulary $V = \{S, 0, 1\}$; set of terminals $T = \{0, 1\}$, stauting symbol S and production $P = \{S \rightarrow 11, S \rightarrow 10\}$. Determine language L(G) of this grammar.

Solution

The language of the grammar is the set of strings:

110, 11110, 1111110, ...

i.e. $L(G) = \{110, 11110, 1111110, \ldots\ldots\ldots\ldots\}$

$L(G) = \{1^{2n}0 / n >= 1\}$

Or equivalently we can say

$L(G) = \{(1)^+ 0\}$

Since,

$S \rightarrow 11S \rightarrow 110$

$S \rightarrow 11S \rightarrow 1111S \rightarrow 11110$ and so on.

Example 18

Let $G = (V, S, V_0 \rightarrow)$, where $V = \{V_0, x, y, z\}$,

$S = \{x, y, z\}$ and

$\rightarrow : v_0 \rightarrow xv_0$

$v_0 \rightarrow yv_0$

$v_0 \rightarrow z.$

Example 19

What is language L(G) of this grammar?

Solution

The language L(G) of this grammar is the set of strings over {x, y, z} which either start with x and end with z or which either start with y and end with z.

i.e. $L(G) = \{xz, xxz, xyz, xyyz, yz, yyz, xxz, \ldots\ldots\}$

or, $L(G) = \{x(x + y)^* z, y(x + y)^* z\}$

Example 20

Let $S = \{0, 1\}$. Give the regular expression corresponding to the regular set given;

(a) {00, 010, 0110, 011110, ...}

(b) {0, 001, 000, 00001, 00000, 000001, ...}

Solution

(a) 01^*0

(b) $0^* (\epsilon + 1)$

Derivation from a Grammar

Strings may be derived from other strings using the productions in a grammar. If a grammar G has a production $\alpha \rightarrow \beta$, we can say that $x \alpha y$ derives $x \beta y$ in G. This derivation is written as; $x \alpha y \Rightarrow x \beta y$.

Derivations are undertaken by beginning with the starting symbol S of a grammar *G* and applying productions from *P*. A derivation terminates when a sentential form is obtained that consists solely of words from the vocabulary T of terminal symbols. A string $\omega \in T$ is said to be a sentence generated *by G* if and only if $S \xrightarrow{*} \omega$.

Let G = (V, T, S,P) be a phrase-structure grammar. Let $w_0 = lz_0r$ and $w_1 = lz_1r$ be strings over V.

- If $z_0 \rightarrow z_1$ is a production of G, we say that w_1 is directly derivable from w_0 (denoted: $w_0 \rightarrow w_1$.).
- If $w_0, w_1, \ldots, w_n$ are strings over V such that $w_0 \rightarrow w_1, w_1 \rightarrow w_2, \ldots, w_{n-1} \rightarrow w_n$, we say that w_n is derivable from w_0 (denoted: $w_0 \xrightarrow{*} w_n$)

The sequence of all steps used to obtain w_n from w_0 is called a derivation.

Example 21

Consider a grammar defined by G = {V, T, S, P}, where V = {a, b, A, B, S}, T = {a, b}, S is the start symbol, and P = {S → Aba | AB | ab, A → BB, B → ab}. Show the derivation of string ab and ababab.

Here, the string ab is directly derivable from the grammar as we have;

$S \rightarrow ab$

For the string ababab:

We have $S \rightarrow AB$

$\rightarrow BBB$ (*using* $A \rightarrow BB$)

$\rightarrow abBB$ (*using* $B \rightarrow ab$)

$\rightarrow ababB$ (*using* $B \rightarrow ab$)

$\rightarrow ababab$ (*using* $B \rightarrow ab$)

So: $S \xrightarrow{*} ababab$. *Hence ababab* is indirectly derivable from *S*

Language of Grammar

Let G = (V, T, P and S) is a grammar. Then the language of G denoted by L (G) is the set of terminal strings that have derivation from the start symbol in G.

i.e. L (G) = $\{x \in T^* \mid S \xrightarrow{*} x\}$

Thus, the set of all strings that can be derived from a grammar is said to be the language generated from that grammar.

Example 22

Let $G = \{V, T, S, P\}$ be the grammar with $V = \{S, 0, 1\}$, $T = \{0, 1\}$, $P = \{S \rightarrow 11S, S \rightarrow 0\}$. Then language of G is, $L(G) = \{0, 110, 11110, 1111110, \ldots\}$, i.e. set of all strings that begin with an even number of 1s and end with 0.

Types of Phrase Structured Grammar

Noam Chomsky has defined the phrase structured grammar with following hierarchy ;

- Type 0 (Unrestricted Grammar)
- Type 1 (Context Sensitive Grammar)
- Type 2 (Context Free Grammar)
- Type 3 (Regular Grammar)

Type-0 (Unrestricted) Grammar

Type-0 grammars (unrestricted grammars) include all formal grammars. They generate exactly all languages that can be recognized by a Turing machine. These languages are also known as the recursively enumerable languages. An unrestricted grammar is a formal grammar on which no restrictions are made on the left and right sides of the grammar's productions.

An unrestricted grammar is a formal grammar $G = (V, \Sigma, P, S)$, where V is a set of non-terminal symbols, Σ is a set of terminal symbols, P is a set of production rules of the form $\alpha \rightarrow \beta$ where α and β are strings of terminals and non-terminals and α is not the empty string, and S is a specially designated start symbol. As the name implies, there are no real restrictions on the types of production rules that unrestricted grammars can have.

Example 23

Following productions define an unrestricted grammar;

S → AbC

Ab → c

cC → Sc | ε

The above grammar accepts empty string or strings containing c.

Type-1(Context Sensitive) Grammar

Type-1 grammars (context-sensitive grammars) generate the context-sensitive languages. A context sensitive grammar is a formal grammar $G = (V, \Sigma, P, S)$, where V is a set of non-terminal symbols, Σ is a set of terminal symbols, P is a set of production rules of the form $\alpha A\beta \rightarrow \alpha\gamma\beta$ where α and β are strings of terminals and non terminals and S is a specially designated start symbol. The strings α and β may be empty. but γ must be nonempty. The rule $S \rightarrow \lambda$ is allowed if S does not appear on the right side of any rule.

For a production $\alpha A\beta \rightarrow \alpha\gamma\beta$, the name *context-sensitive* is explained by the α and β that form the context of A and determine whether A can be replaced with γ or not. This is different from a context-free grammar where the context of a non-terminal is not taken into consideration.

The languages described by these grammars are exactly all languages that can be recognized by a linear bounded automaton (a nondeterministic Turing machine whose tape is bounded by a constant times the length of the input.)

Example 24

The grammar for $\{a^nb^nc^n : n>=1\}$ can be defined using following productions as:

S → aSBC | aBC

CB → HB

HB → HC

HC → BC

aB → ab

bB → bb

bC → bc

cC → cc

Type-2 (Context Free) Grammar

Type-2 grammars are the context-free grammars and generate the context-free languages. A context free grammar is a formal grammar $G = (V, \Sigma, P, S)$, where V is a set of non-terminal symbols, Σ is a set of terminal symbols, P is a set of production rules of the form $A \rightarrow \beta$ where β is the string of terminals and non terminals and S is a specially designated start symbol.

The languages generated by context free grammar are exactly all languages that can be recognized by a non-deterministic pushdown automaton.

Example 25

The CFG for balanced parentheses over $\Sigma = \{(\ ,\ )\}$ can be defined by following production rules:

S → SS

S → (S)

S → ()

Type-3 (Regular) Grammar

Type-3 grammars (regular grammars) generate the regular languages. Such a grammar restricts its rules to a single non-terminal on the left-hand side and a right-hand side consisting of a single terminal, possibly followed by a single non-terminal (right regular).

Alternatively, the right-hand side of the grammar can consist of a single terminal, possibly preceded by a single non-terminal (left regular). These languages are exactly all languages that can be decided by a finite state automaton. Additionally, this family of formal languages can be obtained by regular expressions. Regular languages are commonly used to define search patterns and the lexical structure of programming languages.

A right regular grammar (also called right linear grammar) is a formal grammar (V, Σ, P, S) where V is a set of non-terminal symbols, Σ is a set of terminal symbols, S is start symbol and P are productions of one of the following forms:

$B \rightarrow a$

$B \rightarrow aC$

$B \rightarrow \varepsilon$

In a left regular grammar (also called left linear grammar), all rules obey the forms

$A \rightarrow a$

$A \rightarrow Ba$

$A \rightarrow \varepsilon$

Where, A, B and C are non-terminals, a is terminal and ε is an empty string.

Example 26

An example of a right regular grammar G with V = {S, A}, Σ = {a, b, c}, P consists of the following rules

S → aS

S → bA

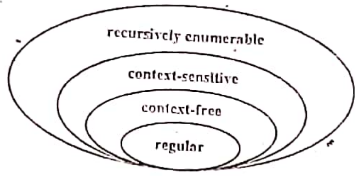
A → ε

A → cA

This grammar describes the same language as the regular expression a*bc*.

The following table summarizes each of Chomsky's four types of grammars, the class of language it generates, the type of automaton that recognizes it, and the form its rules must have.

Grammar	Languages	Automaton	Production Rules
Type-0	Recursively Enumerable	Turing machine	$\alpha \rightarrow \beta$(no restrictions)
Type-1	Context Sensitive	Linear-bounded non-deterministic Turing machine	$\alpha A\beta \rightarrow \alpha\gamma\beta$
Type-2	Context Free	Non-deterministic pushdown automaton	$A \rightarrow \gamma$
Type-3	Regular	Finite state automaton	$A \rightarrow a$ and $A \rightarrow aB$



Chomsky Hierarchy

Example 27

Let G be a grammar with vocabulary V = {s, 0, 1}, set of terminals T = {0, 1}, starting symbol s, and productions P = {s→ 11s, s → 0}. What is L(G), the language of this grammar?

Solution

Here the language of the given grammar consists of following string:

$S \rightarrow 11S \rightarrow 110$

$S \rightarrow 11S \rightarrow 1111S \rightarrow 11110$

$S \rightarrow 11S \rightarrow 1111S \rightarrow 111111S \rightarrow 111111S$

$S \rightarrow 11S \xrightarrow{*} 11111111S \rightarrow 111111110$

From above strings it can be concluded that the languae is even number of is ending with a 0.

Thus, L (G) = {110, 11110, 1111110.........}

$= \{1^{2n}0 / n \geq 1\}$

Example 28

Let G be a grammar with vocabulary V = {S, A, a, b} T = {a, b}, starting symbol S and production P = {s → aA, s → b. A → aa}, what is L(G), the language of the grammar?

Solution:

The language of grammar is L(G) = {aaa, b}

Since s → b so b is a string generated by the grammar

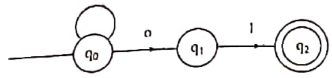
And

S → aA

→ aaa so aaa in another string generated by the grammar.

Example 29

Explain non deterministic finite automata and language of NFA with suitable example.



NFA accepting string over {0, 1} ending with 01.

EXERCISE

1. Define the term a language over vocabulary and the phrase structure grammar.
2. Distinguish between DFA and NFA.
3. Define finite state automata with output.
4. Define DFA? When are two finite state automatic equivalent? Give an example.
5. Define DFA. Construct a DFA whose language is the set of strings that ends with 111 or contains odd number of 1S.
6. Given the following grammar, what is the language accepted by this grammar?

   S → xy

   S → 2A

   S → Az

   A → y
7. Write RE to identify identifies. Mention your assumptions clearly.
8. Define the terms a language over a vocabulary and the phrase structure grammar.
9. Let G be the grammar with vocabulary V = {5. 0, 1} set of terminuls T = {0, 1}, starting symbol S and productions P = {S → 115. S → 0}. What is L(G) the language of the grammar.
10. Let a be the grammar with vocabulary V = {S. A. a, b} t = {a, b}, starting symbol S and production P = {S → aA, S → b, A → aa}. What is L(G), the language of grammar?

11. Determine the cleen closme of the sets

    A = {0}

    B = {0, 1}

    C = {11}
12. Explain non-determinant finite automata and language of NFA with example.
13. Discuss the types of phrase structure grammer sna their relations.
14. Give formal definations of regular expression over a set I.
15. What do you mean by phase structure grammar? Let G be the grammar with vocabulary V = {S, 0, 1}, set of terminals T = {0, 1}, starting symbol S and productions P = {S → 11S, S → 0}, determine the language L(G) of this grammar.
16. Construct a DFA that recognized string $(abc^{*})^{*}$ over {a, b, c}.
17. Construct a DFA over English alphabets that accepts the strings "the" and "then".
18. Construct a NFA accepting language L = $\{0^n11, 0^n01 \mid n>=0\}$ over the binary alphabets.
19. Construct a NFA accepting language L = $(ab)^{*} \cup (aba)^{*}$.
20. Write a regular expression over {0, 1} that begins with 1 and ends with 0.
21. Write a regular expression over {0, 1} that contains at least three 1's.
22. Write a regular expression over {a, b} that has length at least three and its third symbol is a.
23. Write a regular expression over {a, b} that has length at most 5.
24. Write a Context Free Grammar that defines regular expressions.
25. Write a Context Free Grammar for $\{w \mid w = 0^n1^{n+1}\}$.
26. What is the language of following grammar,

    S→ 1A | 0B

    A→ 1AA | 10S | 0

    A→ 0BB | 1AS
27. Write a grammar over {a, b}, containing at least one non-reachable and non-generating symbol, for representing strings that end with ab or ba.

4
Chapter

Recurrence Relations

Recurrence Relations

Suppose a_0, a_1, a_2,...a_n is a sequence. A recurrence relation for the n^{th} term a_n is a formula (i.e., function) giving a_n in terms of some or all previous terms (i.e., a_0; a_1; : : : ; a_n). To find the complete sequence, the rest few initial values are needed. These initial value are called the initial conditions.

If a recurrence relation with initial conditions is given, then we can write down as many terms of the sequence as we want. We will just keep applying the recurrence.

For example, $f_0 = 1$ and $f_1 = 1$; are initial condition with recurrence relation $f_n = f_{n+1} + f_{n+2}$; for n>2.

The terms of sequence are 1, 1, 2, 3, 5, 8, 13, ..., where each subsequent term is the sum of the preceding two terms. On the other hand, if you are given a sequence, you may or may not be able to determine a recurrence relation with initial conditions which describes it.

Example 1

If the given recurrence relation is $a_n = a_{n-1} - a_{n-2}$ with $a_0 = 3$ and $a_1 = 5$ find the terms a_2 and a_3.

Solution

Here,

$$a_n = a_{n-1} - a_{n-2}$$

Then, $$a_2 = a_{2-1} - a_{2-2}$$

$$a_2 = a_1 - a_0 = 5 - 3 = 2$$

and $$a_3 = a_{3-1} - a_{3-2} = a_2 - a_1 = 2 - 5 = -3$$

Example 2

Find the recurrence relation and initial conditions of the sequence 3, 8, 13, 18,....

Solution

Here, $a_1 = 3$

$a_2 = a_1 + 5 = 8$

$a_3 = a_2 + 5 = 13$

$a_4 = a_3 + 5 = 18$

$a_n = a_{n-1} + 5$

Hence, the recurrence relation is

$a_n = a_{n-1} + 5, \ n \geq 2, a_1 = 3$

Example 3

Find the recurrence relation of the sequence 2, 5, 11, 23, 47

Solution

$a_1 = 2$

$a_2 = 5 = 2 \times a_1 + 1$

$a_3 = 11 = 2 \times a_2 + 1$

$a_4 = 23 = 2 \times a_3 + 1$

$a_n = 2a_{n-1} + 1$

Hence, the recurrence relation is $a_n = 2a_{n-1} + 1$ with initial conditions $a_1 = 2$, $n \geq 2$.

Example 4

Find the recurrence relation of the sequence 1, 1, 2, 3, 5, 8, 13,.....

Solution

Here, $a_1 = 1$

$a_2 = 1$

$a_3 = 2 = a_1 + a_2$

$a_4 = 3 = a_3 + a_2$

$a_5 = 5 = a_4 + a_3$

$a_n = a_{n-1} + a_{n-2}$

Hence, the recurrence relation is

$a_n = a_{n-1} + a_{n-2}$ with initial conditions $a_1 = 1$, $a_2 = 1$ for $n \geq 3$.

This relation is also known as Fibonacci relation.

Solving Recurrence Relation

If the given recurrence relation involving sequence $a_0, a_1, a_2, \ldots\ldots, a_n$ then the solution of the recurrence relation is to find a_n explicit formula for general term a_n.

To solve the recurrence relation, we may replace each of $a_{n-1}, a_{n-2}, \ldots$ by their predecessors. This process continues until an explicit formula for n^{th} term a_n.

Example 5

Solve the recurrence relation

$a_n = 2a_{n-1}, n \geq 1$ and $a_0 = 3$

Solution

Here,

$a_0 = 3$

$a_1 = 2a_0 = 2\ (3)$

$a_2 = 2a_1 = 2\ (2 \times 3) = 2^2\ (3)$

$a_3 = 2a_2 = 2\ (2^2.3) = 2^3\ (3)$

$a_4 = 2a_3 = 2\ (2^3.3) = 2^4\ (3)$

$a_n = 2^n\ (3)$

Hence, general solution for given recurrence relation is $a_n = 2^n\ (3)$

Example 6

Solve the recurrence relation $a_n = a_{n-1} + 2$ Subject to initial condition $a_1 = 3$

Solution

$a_n = a_{n-1} + 2$	Step 0
$= (a_{n-2} + 2) + 2$	Step 1
$= (a_{n-3} + 2) + 4$	Step 2
$= (a_{n-4} + 2) + 6$	Step 3
$= (a_{n-5} + 2) + 8$	Step 4
$a_n = a_1 + 2\ (n - 1)$	Step (n − 1)

Example 7

Show that the solution of Recurrence relation $a_n = 2a_{n-1} - a_{n-2}$ is $a_n = 3n$

Solution

Here,

$$a_n = 2a_{n-1} - a_{n-2}$$
$$3n = 2\,[3\,(n-1) - 3(n-2)]$$
$$3n = 2\,(3n-3) - 3n + 6$$
$$3n = 6n - 6 - 3n + 6$$
$$3n = 3n$$
$$\text{LHS} = \text{RHS}$$

Example 8

Find the first five terms of sequence defined by $a_n = 6a_{n-1}$, $a_0 = 2$

Solution

Given, $a_n = 6a_{n-1}$

Then,

$$a_1 = 6a_0 = 6 \times 2 = 12$$
$$a_2 = 6a_1 = 6 \times 12 = 72$$
$$a_3 = 6a_2 = 6 \times 72 = 432$$
$$a_4 = 6a_3 = 6 \times 432 = 2592$$

$\therefore$ First five terms are: 2, 12, 72, 432 & 2592

Example 9

Find the first three terms of sequence defined by:

$$a_n = a_{n-1} + 3a_{n-2},\ a_0 = 2,\ a_1 = 2$$

Solution

Here,

$$a_n = a_{n-1} + 3a_{n-2}$$
$$a_2 = a_1 + 3a_0$$
$$= 2 + 3 \times 2 = 2 + 6 = 8$$
$$a_3 = a_2 + 3a_1$$
$$= 8 + 3 \times 2 = 14$$

$\therefore$ First three terms are: 2, 8, 14

Example 10

Solve the recurrence relation $a_n = a_{n-1} + 4$ subject to the initial condition $a_1 = 3$.

Solution

Here,

$$a_n = a_n - 1 + 4,\ a_1 = 2$$

$\therefore$

$$a_1 = 2$$
$$a_2 = a_1 + 4 = 3 + 4 = 7$$
$$a_3 = a_2 + 4 = 7 + 4 = 11$$
$$a_4 = a_3 + 4 = 11 + 4 = 15$$
$$a_5 = a_4 + 4 = 15 + 4 = 19$$

Introduction to Basic counting Principles

Combinatory is that branch of discrete mathematics which concerns with counting problems. Techniques for counting are important in Mathematics and computer science, especially in probability theory and in the analysis of algorithms.

For instance, counting is required to determine whether there are enough telephone numbers or internet protocol addresses to meet demand, it is also used to determine the complexity of algorithms, counting techniques are used extensively when probabilities of events are computed.

Counting problems arise throughout mathematics and computer science. For example, we need to count the number of key operations used in algorithm to study its time complexity.

There are two basic counting principles.

Sum Rule Principle

If an event E can occur in m ways and another event F can occur in n ways, and if these two events cannot occur simultaneously, then one of the two events, i.e E or F, can occur in m + n ways. More generally, suppose an event E_1 can occur in n_1 ways, a second event E_2 can occur in n_2 ways, a third E_3 can occur in n_3 ways, and so on and suppose no two of the events can occur at the same time. Then one of the events can occur in $n_1 + n_2 + n_3 + \ldots$ ways.

Example 11

If there are 24 boys and 18 girls in a class, find the number of ways of selecting one student as class representative.

Solution

Using sum rule, there are 24 + 18 = 42 ways of selecting one student as a class representative.

Example 12

Let E be the event of choosing a prime number less than 10, and F be the event of choosing an even number less than 10, find the number of ways that E or F can occur.

to old

Solution

E can occur in 4 ways {2, 3, 5, 7}, and F can occur in 4 ways {2, 4, 6, 8}. However E or F cannot occur in 4 + 4 = 8 ways. Since 2 is both prime number less than 10 and an even number less than 10. In fact, E or F can occur in only 4 + 4 – 1 = 7 ways.

Example 13

A student can choose a computer project from one of three lists. The three lists contain 23, 15, and 19 possible projects, respectively. How many possible projects are there to choose from?

Solution

The student can choose a project from the first list in 23 ways, from the second list in 15 ways, and from the third list in 19 ways. Hence, there are 23+15+19 =57 projects to choose from.

Example 14

How many ways we can get a sum of 4 or of 8 when two distinguishable dice (say one die is red and the other is white) are rolled?

Solution

Since dice are distinguishable outcome (1, 3) is different form (3, 1) so to get 4 as sum we have the pairs (1, 3), (3, 1), (2, 2), so total of 3 ways. And similarly getting 8 can be from pairs (2, 6), (6, 2), (3, 5), (5, 3), (4, 4), so total 5 ways. Hence getting sum of 4 or 8 is 3 + 5 = 8 ways

Product Rule Principle

If an event E can occur in m ways and, independent of this event, if another event F can occur in n ways, then two events can occur simultaneously in mn ways. More generally, suppose an event E_1 can occur in n_1 ways, and, following E_1, a second event E_2 can occur in n_2 ways, and following E_2, a third can occur in n_3 ways and so on. The all the events can occur simultaneously in $n_1.n_2.n_3$ .... ways.

Example 15

An office building contains 27 floors and has 37 offices on each floor. How many offices are there are in the building?

Solution

No. of floors in the office = 27

No of offices in each floor = 37

Therefore, by the product rule there are 27.37 = 999 offices in the building.

Example 16

How many different three-letter initials with none of the letters can be repeated can people have?

Solution

Here the first letter can be chosen in 26 ways, since the first letter is assigned we can choose second letter in 25 ways and in the same manner we can choose third letter in 24 ways. So by product rule number of different three-letter initials are 26.25.24 = 15600

Example 17

Three persons enter into a car, where there are 5 vacant seats. In how many ways can they take up their seats?

Solution

The first person has a choice of 5 seats and can sit in any one of those 5 seats. So there are 5 ways of occupying the first seat. The second person has a choice of 4 seats, so there are 4 ways of occupying the second seat. Similarly, there are 3 ways of occupying the third seats.

Hence, the required number of ways in which all the three persons can sit is 5 × 4 × 3 = 60

Example 18

Suppose a license plate contains two letters followed by three digits with the first digit not zero. How many different license plates can be printed?

Solution

Each letter can be printed in 26 different ways, the first digit in 9 ways and each of the other two digits in 10 ways. Hence,

26 × 26 × 9 × 10 × 10 = 608400 different plates can be printed.

Example 19

There are 32 microcomputers in a computer center. Each microcomputer has 24 ports. How many different ports to a microcomputer in the center are there?

Solution

The procedure of choosing a port consist of two tasks , first picking a microcomputer and the picking a port on this microcomputer. Since there are 32 ways to choose the microcomputer and 24 ways to choose the port no matter which microcomputer has been selected, using product rule: there are 32 . 24 = 768 ports.

Example 20

In how many ways can an organization containing 26 members elect a president, treasurer, and secretary (assuming no person is elected to more than one position)?

Solution

The president can be elected in 26 different ways, the treasurer can be elected in 25 different ways and following this, the secretary can be elected in 24 different ways. Thus, there are 26 × 25 × 24 = 15600 different ways in which the organization can elect a president, a treasurer and a secretary.

Example 21

How many strings are there of four lowercase letters that have the letter x in them?

Solution

There are total 26.26.26.26 strings of four lowercase letters, by product rule. In the same way we can say that there are 25.25.25.25 strings of four lowercase letters without x, since without x there will be a set of 25 characters only. So there are total of 26.26.26.26-25.25.25.25 = 66351 four lowercase letter strings with x in them. This is true because we are decrementing total numbers of strings with the number of strings that do not contain x in them so at least one x will be in the strings.

Example 22

How many functions are there from the set {1, 2, ..., n}, where n is a positive integer, to the set {0, 1}.

Solution

Each element from the set {1, 2, ..., n}can map the set {0, 1} in 2 ways. Since there are n elements in the first set by the product rule number of possible functions are 2.2.2.....n^{th} term i.e. 2n.

There is a set theoretical interpretation of the above two counting principles. Specifically, suppose n(A) denotes the number of elements in a set A and n(B) denotes the number of elements in set B then,

(i) **Sum Rule Principle:** If A and B are two disjoint sets, then

$n(A \cup B) = n(A) + n(B)$.

(ii) **Product Rule Principle:** If A × B be the Cartesian product of sets A and B, then

$n(A \times B) = n(A) \times n(B)$.

The Pigeonhole Principle

The pigeonhole principle states that, If k + 1 or more pigeons are placed into k pigeonholes, then there is at least one pigeonhole containing two or more of the pigeons.

Proof

We use proof by contradiction here. Suppose that k+1 or more boxes are placed into k boxes and no boxes contain more than one object in it. If there are k boxes then there must be k objects such that there are no two objects in a box. This contradicts our assumption. So there is at least one box containing two or more of the objects.

Generalized Pigeonhole Principle

If n pigeonholes are occupied by Kn + 1 or more pigeons, where K is a positive integer, then at least one pigeonhole is occupied by K + 1 or more pigeons.

Theorem

If N objects are placed into k boxes, then there is at least one box containing at least $\lceil N/k \rceil - 1$ objects.

Proof

Suppose N objects are placed into k boxes and there is no box containing more than $\lceil N/k \rceil - 1$ objects. So the total number of objects is at most $k(\lceil N/k \rceil - 1) < k((N/k + 1) - 1) = N$. This is the contradiction that N objects are placed into k boxes (since we showed that there are total number of objects less than N). Hence, the proof.

Example 23

If 9 books are to be kept in 4 shelves, there must be at least one shelf which contains at least 3 books.

Solution

The nine books can be thought of as pigeons and four shelves as oigeonholes. Then n = 4 (Pigeonholes) and Kn + 1 = 9(Pigeons)

$\therefore \quad K \times 4 + 1 = 9 \Rightarrow K = 2$

So, at least 1 pigeonhole i.e. shelf is occupied by k + 1 =2 + 1 = 3 pigeons i.e. books.

Example 24

If a class has 24 students, what is the maximum number of possible grading that must be done to ensure that there at least two students with the same grade.

Solution

There are total 24 students and the class and at least two students must have same grade. If the number of possible grades is k then by pigeonhole principle

we have ceil(24 / k) = 2.

Here the largest value that k can have is 23 since 24 = 23.1 + 1. So the maximum number of possible grading to ensure that at least two of the students have same grading is 23.

Example 25

Find the minimum number of students required in a class to be sure that three of them are born in the same month.

Solution

Here, the n = 12 months are pigeonholes and k + 1 = 3(pigeons) are born in same month. Then k + 1 = 3 gives k = 2. Thus kn + 1 = number of students (pigeons) = 2 × 12 + 1 = 25.

Example 26

Show that if any 15 people are selected, then we may choose a subset of 3 so that all 3 were born on the same day of the week.

Solution

We may assign each person (pigeon) to the day of the week on which she/he was born. So, 15 people (pigeons) are to be assigned to 7 pigeonholes (days of the week).

∴ n = 7(pigeonholes) and Kn + 1 = 15 (pigeons)

So, Kn + 1= 7 × K + 1 = 15 gives K = 2

Therefore at least 1 day is occupied by k + 1 = 2 + 1 = 3 people.

Example 27

Suppose a laundry bag contains many red, white and blue socks. Find the minimum number of socks that one needs to choose in order to get two pairs (four socks) of same colour.

Solution

Here n = 3 colours (pigeonholes) and k + 1 = 4(pigeons).So k + 1 = 4 gives k = 3. Thus kn + 1 = number of socks (pigeons) = 3 × 3 + 1 = 10

Example 28

Find the minimum number of elements that needs to take from the set A = {1, 2, ..., 9} to be sure that two of them add up to 10.

Solution

Here A = {1, 2, ..., 9}. So the pigeonholes are the five sets (1, 9), (2, 8), (3, 7), (4, 6), (5, 5). So, we have n = 5 pigeonholes and k + 1 = 2 (Pigeons). Then k = 1. Now, kn + 1 = 5 × 1 + 1 = 6. Thus any choice of six elements (pigeons) of A will guarantee that two of them add up to ten.

Example 29

How many numbers must be selected from the set {1, 3, 5, 7, 9, 11, 13, 15} to guarantee that at least one pair of these numbers add up to 16?

Solution

The pairs of numbers that sum 16 are (1, 15), (3, 13), (5, 11), (7, 9) i.e. 4 pairs of numbers are there that add to 16. If we select 5 numbers then by pigeonhole principle there are at least,

Ceil(5 / 4) = 2 numbers, that are from the set of selected 5 numbers, that constitute a pair. Hence 5 numbers must be selected.

Example 30

Find the least number of cables required to connect eight computers to four printers to guarantee that four computers can directly access four different printers. Justify your answer.

Solution

If we connect first 4 computers directly to each of the 4 printers and the other 4 computers are connected to all the printers, then the number of connection required is 4 + 4.4 = 20. To verify that 20 is the least number of cables required we have if there may be less than 20 cables then we would have 19 cables, then some printers would be connected by at most Ceil(19 / 4) = 4 cables to the computers. Then the other 3 printers would have to connect the other 4 computers here all the computers cannot simultaneously access different printer. So if we use 20 cables, then at least ceil(20 / 4) = 5 cables connects a printer to a computer directly. So the remaining 3 printers are required to connect only 3 computers. Hence the least number of cables required is 20.

Example 31

Among n + 1 different integral powers of an integer a, there are at least two of them that have same remainder when divided by the positive integer n.

Proof

Let $a_1, a_2, \dots, a_{n+1}$ be n+1 different integral powers of integer a. when these numbers are divided by n then the set of possible remainders is {0, 1, 2, ..., n − 1}. Since there are n remainders and n+1 numbers by pigeonhole principle at least 2 of the reminders must be same.

Permutation

Permutation of a set of objects means an arrangement of objects in some order. An ordered arrangement of r objects from a set of n objects is called r-permutation of n objects. It is denoted by P(n, r) or ^nP_r consider, for example, the set of letters a, b, c and d. then:

(i) bdca, abcd, cdab, etc are permutations of the four letters taking all of them at a time.

(ii) abc, acd, bcd, etc. are permutation of the four letters taking three letters at a time.

(iii) ab, bc, cd, bd, etc are permutation of the four letters taking two letters at a time.

Theorem 3.1

The total number of permutations of a set of n objects taking r objects at a time is given by

$$P(n, r) = n(n-1)(n-2) \ldots (n-r+1) = \frac{n!}{(n-r)!}(r \leq n)$$

Proof:

The number of permutations of a set of n objects taken r at a time is equivalent to the number of ways in which r positions can be filled up by those n objects. So there are n choices to fill up the first position. Now there are only n – 1 objects left to fill up the second position i.e. there are n – 1 choices to fill up the second position. Similarly there are n – 2 choices to fill up the third position and so on.

At last, to fill r^{th} position there are n – (r – 1) choices. Thus, by the fundamental principle of counting, we have,

$$P(n, r) = n(n-1)(n-2)\ldots\ldots\{n-(r-1)\}$$

$$= n(n-1)(n-2)\ldots\ldots(n-r+1)$$

$$= \frac{n(n-1)(n-2)\ldots(n-r+1)(n-r)(n-r-1)\ldots 2.1}{(n-r)(n-r-1)\ldots 2.1}$$

$$= \frac{n!}{(n-r)!}$$

Example 32

How many license plates consisting of 3 different digits can be made out of given integers 3, 4, 5, 6, 7?

Solution

This is just like arranging 3 objects out of 5 objects. So we have,

$$P(5, 3) = \frac{5!}{(5-3)!} \quad [\because {}^nP_r = \frac{n!}{(n-r)!}]$$

$$= \frac{5 \times 4 \times 3 \times 2!}{2!}$$

$$= 60$$

This problem can also be solved by product Rule.

Example 33

How many number of 3 digits can be formed from the digits 3, 4, 5, 6, 7, 8? How many of these are divisible by 5.

Solution

Here, n = 6, r = 3

$$\therefore \quad P(6, 3) = \frac{6!}{(6-3)!} = \frac{6 \times 5 \times 4 \times 3!}{3!} = 120$$

To find the number divisible by 5, we fix the digit 5 on unit place.

$\therefore$ n = 6 – 1 = 5, r = 3 – 1 = 2

$$\therefore \quad P(5, 2) = \frac{5!}{(5-2)!} = \frac{5 \times 4 \times 3!}{3!} = 20$$

Example 34

Find the number of ways in which a party of seven persons can arrange themselves: (a) in a row of seven chairs (b) around a circular table.

Solution

(a) The seven persons can arrange themselves in a row in 7.6.5.4.3.2.1 = 7! ways.

(which is same as P(7, 7))

(b) One person can sit at any place in the circular table. The other six persons can then arrange themselves in 6! ways around the table.

(which is same as, 7 persons can be arranged in a circular table is (7 – 1)! i.e. 6! Ways)

Example 35

In how many ways can 4 girls and 4 boys be arranged alternately on a round table?

Solution

First, let all the girls i.e. 4 girls can be arranged in (4 – 1)! = 3! Ways. But 4 boys between the girls can be arranged in 4! Ways, since a girl is already been fixed. Hence, 4 girls and 4 boys can be arranged alternately on a round table is 3! × 4! ways.

Example 36

In how many ways can 8 people be seated (a) in a row of 8 chairs (b) around a circular table, if two people insist on sitting next to each other?

Solution

(a) If two people insist on sitting next to each other, we consider two people as one. Then 7 people can be arranged in a row in 7! Ways. But two people can interchange their position in 2 ways.

∴ Total number of arrangements = 2 × 7! = 10080 ways.

(b) Similarly we consider two people as one. Then 7 people can be arranged in a round table in (7 – 1)! ways i.e. 6! ways. But two people can interchange their position in 2 ways.

∴ Total number of arrangements = 2 × 6! = 1440.

Permutations with Repetitions

The permutation of n objects taken all at a time, when there are P objects of one kind, q objects are of second kind, r objects are of a third kind, is $\frac{n!}{p!q!r!}$.

Example 37

How many seven – letter words can be formed using the letters of the word "BENZENE"?

Solution

There are seven objects (letters) of which there are three E_S and two N_S. Therefore, the number of permutation is $\frac{7!}{3!\,2!}$

$$=\frac{7\times 6\times 5\times 4\times 3!}{3!\times 2\times 1}=420$$

Example 38

How many six-letter words can be made using the letters of the word "SUNDAY"?

Solution

There are six-letters of which none is repeated. Therefore, the number of permutation is $\frac{6!}{0!}$

6 × 5 × 4 × 3 × 2 × 1 = 720.

(which is arrangement of 6 letters out of 6 i.e. P(6, 6))

Example 39

(a) In how many ways can the letters of the word "arrange" be arranged?

(b) In how many ways can the letters of the word "arrange" be arranged so that the two r, always come together?

(c) In how many ways the letters can be arranged so that two r, never come together?

Solution

(a) There are 7 letters in which there are two a, and two r, so

Total number of alphabet to be arrange (n) = 7

Number of repetition of a (p) = 2

Number f repetition of r (q) = 2.

∴ Total number of arrangements $=\frac{n!}{p!\,q!}$

$$=\frac{7!}{2!\,2!}$$

= 1260 ways.

(b) For the case when two r, always come together, we consider two r, as a single one. Then the word "arrange" becomes "arrange". So there are 6 letters including two a.

∴ Total number of arrangements = $\frac{6!}{2!}$ = 360 ways.

(c) The total number of arrangements of letters of the word 'arrange' is 1260 and the total number of arrangements of the word 'arrange' where two r, always come together is 360. So the total number of arrangements of the word 'arrange' where two r, never come together is (1260 – 360) = 900 ways

Example 40

(a) Find the number of Permutations that can be formed from the letters of the word ELEVEN.

(c) How many of them have the three Es together?

(d) How many of them never have the three Es together?

(e) How many of them begin with E and end with N?

Solution

(a) There are 6 letters of which three are Es.

So, total number of alphabet to be arrange (n) = 6

Number of repetition of E (p) = 3

$\therefore$ Total number of Permutations $= \frac{6!}{3!} = 120$

(b) Let's consider six boxes for six letters as

$\boxed{E}\ \boxed{}\ \boxed{}\ \boxed{}\ \boxed{}\ \boxed{E}$

If first and last box is filled by E then remaining 4 letters in 4 boxes (in a row) can be filled in 4! ways i.e. 24 ways. Thus 24 of them begin and end with E

(c) For the case when three Es always come together, we consider three Es a single E. Then there are only four letters E, L, V and N. So four letters can be arranged in a row in 4! ways i.e. in 24 ways.

(d) Since the total number of permutations of the letters of the word ELEVEN is 120 and the number of permutations of the letters of word ELEVEN if 3 Es always come together is 24 then the number of permutations of the letters of the word ELEVEN when 3 Es never come together is (120 – 24) = 96.

(e) Lets consider six boxes for six letters where first box contains E and last box contains N.

$\boxed{E}\ \boxed{}\ \boxed{}\ \boxed{}\ \boxed{}\ \boxed{N}$

Then out of six letters, only four letters are left which are L, E, V and E. Now the permutations of the 4 letters in which two are Es, is $\frac{4!}{2!} = 12$.

Example 41

Find the number of permutations of the letters in the word 'COMPUTER' taken four at a time in which

(i) two letters M and R do not occur.

(ii) two letters M and R always occur.

Solution

(i) From the word 'COMPUTER', if two letters M and R do not occur then the number of permutations = P(6, 4) = 360

(ii) If two letters M and R always occur, the number of permutations = P(6, 2) × 2 = 60.

Combination

Combination of objects means just their collection without any regard to order or arrangement. An unordered selection of r objects from a set of n objects is called a combination or r–combination. It is denoted by C(n, r) or ^nC_r and is given by the expression

$$C(n, r) = \frac{n!}{r!(n-r)!}$$

Combination	Permutations
Abc	abc, acb, bac, bca, cab, cba
abd	abd, adb, bad, bda, dab, dba
acd	acd, adc, cad, cda, dac, dca
bcd	bcd, bdc, cbd, cdb, dbc, dcb

Example 42

In how many ways can a hand of 4 cards be dealt from an ordinary pack of 52 playing cards?

Solution

We need to consider combinations, since the order in which the cards are dealt is not important.

Now, $C(52, 4) = \frac{52!}{4!\,(52-4)!}$ $\quad [\because C(n, r) = \frac{n!}{r!(n-r)!}]$

$= \frac{52!}{4!\,48!}$

$= \frac{(52) \times (51) \times (50) \times (49)}{(4)\,(3)\,(2)\,(1)}$ $\quad [\because n! = n(n-1)!]$

= 270725 ways.

Example 43

A cricket team is to be formed consisting of two wicket keepers, 4 bowlers and 5 batsman from a group of players containing 4 wicket keepers, 8 bowlers and 11 bats man. The number of ways a cricket team can be constituted.

Solution

The selection can be made as follows:

Players	Total number	Number of be selected	No. of ways
Wicket–keepers	4	2	${}^{4}c_2$
Bowlers	8	4	${}^{8}c_4$
Batsmen	11	5	${}^{11}c_5$

$\therefore$ Total number of ways in which a cricket team can be constituted

$= {}^{4}c_2 \times {}^{8}c_4 \times {}^{11}c_5$

= 1,94,040 ways

Example 44

Consider the set {a, b, c, d}. In how many ways can we select two of these letters (repetition is not allowed) when (i) order matters (ii) order does not matter?

Solution

(i) If order matters, the number of ways of selecting two letters from four letters is

$$P(4, 2) = \frac{4!}{(4-2)!} \quad [\because {}^{n}p_r = \frac{n!}{(n-r)!}]$$

$$= 12 \text{ ways.}$$

(ii) If order does not matter, the number of ways of selecting two letters from four letters is

$$C(4, 2) = \frac{4!}{2!\,(4-2)!} \quad [\because C(n, r) = \frac{n!}{r!(n-r)!}]$$

$$= \frac{4!}{2!\,2!} = 6 \text{ ways.}$$

Example 45

A committee is to be chosen from 12 men and 8 women and is to consist of 3 men and 2 women. How many different committees can be formed?

Solution

3 men can be chosen from 12 men is

$$C(12, 3) = \frac{12!}{3!(12-3)!} = 220$$

2 women can be chosen from 8 women is $C(8, 2) = \frac{8!}{2!(8-2)!} = 28$

Hence the total number of different committees possible is

$220 \times 28 = 6160.$

Example 46

A person has got 12 close friends of which 8 are girls. In how many different ways can he invite 7 friends so that 5 of them may be girls?

Solution

5 girls can be invited from 8 girls is

$$C(8, 5) = \frac{8!}{5!\,(8-5)!} = 56$$

Remaining 2can be invited from (12 – 8) is

$$C(4, 2) = \frac{4!}{2!(4-2)!} = 6$$

Thus, he can invite 7 friends in (56 × 6) ways = 336 ways.

Example 47

A bag contains six white marbles and five red marbles. Find the number of ways four marbles can be drawn from the bag if

(a) They can be any colour

(b) Two must be white and two red.

(c) They all must be of the same colour.

Solution

(a) Here, total number of marbles is 11. The four marbles of any colour can be chosen from 11 marbles in C(11, 4) ways.

$$\therefore C(11, 4) = \frac{11!}{4!\,(11-4)!}$$

$$= \frac{11 \times 10 \times 9 \times 8 \times 7!}{4! \times 7!}$$

$$= 330.$$

Solution

The selection can be made as follows:

Players	Total number	Number of be selected	No. of ways
Wicket–keepers	4	2	$^4C_2$
Bowlers	8	4	$^8C_4$
Batsmen	11	5	$^{11}C_5$

∴ Total number of ways in which a cricket team can be constituted

$= {}^4c_2 \times {}^8c_4 \times {}^{11}c_5$

= 1,94,040 ways

Example 44

Consider the set {a, b, c, d}. In how many ways can we select two of these letters (repetition is not allowed) when (i) order matters (ii) order does not matter?

Solution

(i) If order matters, the number of ways of selecting two letters from four letters is

$$P(4, 2) = \frac{4!}{(4-2)!} \qquad [\because {}^nP_r = \frac{n!}{(n-r)!}]$$

= 12 ways.

(ii) If order does not matter, the number of ways of selecting two letters from four letters is

$$C(4, 2) = \frac{4!}{2!\,(4-2)!} \qquad [\because C(n, r) = \frac{n!}{r!(n-r)!}]$$

$$= \frac{4!}{2!\,2!} = 6 \text{ ways.}$$

Example 45

A committee is to be chosen from 12 men and 8 women and is to consist of 3 men and 2 women. How many different committees can be formed?

Solution

3 men can be chosen from 12 men is

$$C(12, 3) = \frac{12!}{3!(12-3)!} = 220$$

2 women can be chosen from 8 women is $C(8, 2) = \frac{8!}{2!(8-2)!} = 28$

Hence the total number of different committees possible is

$220 \times 28 = 6160.$

Example 46

A person has got 12 close friends of which 8 are girls. In how many different ways can he invite 7 friends so that 5 of them may be girls?

Solution

5 girls can be invited from 8 girls is

$$C(8, 5) = \frac{8!}{5!\,(8-5)!} = 56$$

Remaining 2can be invited from (12 – 8) is

$$C(4, 2) = \frac{4!}{2!(4-2)!} = 6$$

Thus, he can invite 7 friends in (56 × 6) ways = 336 ways.

Example 47

A bag contains six white marbles and five red marbles. Find the number of ways four marbles can be drawn from the bag if

(a) They can be any colour

(b) Two must be white and two red.

(c) They all must be of the same colour.

Solution

(a) Here, total number of marbles is 11. The four marbles of any colour can be chosen from 11 marbles in C(11, 4) ways.

$$\therefore C(11, 4) = \frac{11!}{4!\,(11-4)!}$$

$$= \frac{11 \times 10 \times 9 \times 8 \times 7!}{4! \times 7!}$$

$= 330.$

(c) How many of them have the three Es together?

(d) How many of them never have the three Es together?

(e) How many of them begin with E and end with N?

Solution

(a) There are 6 letters of which three are Es.

So, total number of alphabet to be arrange (n) = 6

Number of repetition of E (p) = 3

$\therefore$ Total number of Permutations $= \frac{6!}{3!} = 120$

(b) Let's consider six boxes for six letters as

$\boxed{E}\ \boxed{}\ \boxed{}\ \boxed{}\ \boxed{}\ \boxed{E}$

If first and last box is filled by E then remaining 4 letters in 4 boxes (in a row) can be filled in 4! ways i.e. 24 ways. Thus 24 of them begin and end with E

(c) For the case when three Es always come together, we consider three Es a single E. Then there are only four letters E, L, V and N. So four letters can be arranged in a row in 4! ways i.e. in 24 ways.

(d) Since the total number of permutations of the letters of the word ELEVEN is 120 and the number of permutations of the letters of word ELEVEN if 3 Es always come together is 24 then the number of permutations of the letters of the word ELEVEN when 3 Es never come together is (120 – 24) = 96.

(e) Lets consider six boxes for six letters where first box contains E and last box contains N.

$\boxed{E}\ \boxed{}\ \boxed{}\ \boxed{}\ \boxed{}\ \boxed{N}$

Then out of six letters, only four letters are left which are L, E, V and E. Now the permutations of the 4 letters in which two are Es, is $\frac{4!}{2!} = 12$.

Example 41

Find the number of permutations of the letters in the word 'COMPUTER' taken four at a time in which

(i) two letters M and R do not occur.

(ii) two letters M and R always occur.

Solution

(i) From the word 'COMPUTER', if two letters M and R do not occur then the number o permutations = P(6, 4) = 360

(ii) If two letters M and R always occur, the number of permutations = P(6, 2) × 2 = 60.

Combination

Combination of objects means just their collection without any regard to order or arrangement. An unordered selection of r objects from a set of n objects is called a combination or r–combination. It is denoted by C(n, r) or ^nC_r and is given by the expression

$$C(n, r) = \frac{n!}{r!(n-r)!}$$

Combination	Permutations
Abc	abc, acb, bac, bca, cab, cba
abd	abd, adb, bad, bda, dab,dba
acd	acd, adc, cad, cda, dac, dca
bcd	bcd, bdc, cbd, cdb, dbc, dcb

Example 42

In how many ways can a hand of 4 cards be dealt from an ordinary pack of 52 playing cards?

Solution

We need to consider combinations, since the order in which the cards are dealt is not important.

Now, $C(52, 4) = \frac{52!}{4!\,(52-4)!}$ $\quad [\because C(n, r) = \frac{n!}{r!(n-r)!}]$

$= \frac{52!}{4!\,48!}$

$= \frac{(52) \times (51) \times (50) \times (49)}{(4)\,(3)\,(2)\,(1)}$ $\quad [\because n! = n(n-1)!]$

= 270725 ways.

Example 43

A cricket team is to be formed consisting of two wicket keepers, 4 bowlers and 5 batsman from a group of players containing 4 wicket keepers, 8 bowlers and 11 bats man. The number of ways a cricket team can be constituted.

(b) Two white marbles can be chosen in C(6, 2) ways and two red marbles can be chosen in C(5, 2) ways. Thus there are C(6, 2) · C(5, 2) ways of drawing two white marbles and two red marbles.

$$\therefore C(6, 2) \cdot C(5, 2) = \frac{6!}{2!\,(6-2)!} \times \frac{5!}{2!\,(5-2)!}$$

$$= \frac{6!}{2!\,4!} \times \frac{5!}{2!\,3!}$$

$$= 150$$

(c) There are C(6, 4) = 15 ways of drawing four white marbles and C(5, 4) = 5 ways of drawing four red marbles. Thus there are 15 + 5 = 20 ways of drawing four marbles of the same colour.

Example 48

In how many ways can a cricket team of eleven be chosen out of fifteen players? How many of them will always

(a) include a particular player?

(b) exclude a particular player?

Solution

The number of ways of selecting 11 players out of 15 is C(15, 11) = 1365.

(a) The number of ways in which a particular player is also included is C(14, 10) = 1001 ways

(b) The number of ways in which a particular player is always excluded is C(14, 11) = 364 ways

Example 49

A certain family consists of a Mother, a father and ten sons. They are invited to send a group of four representatives to a wedding. Evaluate the number of ways in which the group can be formed, if it must contain (i) both parents (ii) one and only one parent (iii) neither parent.

Solution

(i) If the group contains both parents, the number of ways to choose the other 2 from 10 (i.e. from 10 sons)

= C(10, 2)

= 45 ways

(ii) One parent from mother and father can be chosen in 2 ways. Now, the remaining 3 can be chosen from 10 sons

= C(10, 3)

= 120 ways

Therefore number of ways to choose the group of 4 = 2 × 120

= 240 ways

(iii) The number of ways to choose 4 from 10 = C(10, 4)

= 210 ways

Binomial Theorem

Let x and y be two variables and n be non-negative integers then the binomial theorem for positive index n states that,

$$(x+y)^n = \binom{n}{0}x^n + \binom{n}{1}x^{n-1}y + \binom{n}{2}x^{n-2}y^2 + \ldots + \binom{n}{n}y^n$$

The coefficients $\binom{n}{0}, \binom{n}{1}, \ldots \binom{n}{n}$ in the binomial expansion are called binomial coefficients, which are also denoted by $C_0, C_1, C_2, \ldots C_n$ respectively.

General Term

The $(r+1)^{th}$ term in the expansion of $(x+y)^n$ is usually called its general term, which is denoted by t_{r+1}.

In the expansion of $(x+y)^n$,

$t_1 = 1^{st}$ term $= C(n, 0)x^n$

$t_2 = 2^{nd}$ term $= C(n, 1)x^{n-1}y^1$.

$t_3 = 3^{rd}$ term $= C(n, 2)\,x^{n-2}y^2$

$t_4 = 4$th term $= C(n, 3)\,x^{n-3}y^3$

. . .

. . .

$t_{r+1} = (r+1)^{th}$ term $= C(n, r)x^{n-r}y^r$.

Thus the general term in the expansion of $(x+y)^n$ is t_{r+1} which is

$C(n, r)x^{n-r}y^r$.

Example 50

Show that:

(a) $\binom{n}{0}+\binom{n}{1}+\binom{n}{2}+\dots+\binom{n}{n}=2^n$

(b) $\binom{n}{0}-\binom{n}{1}+\binom{n}{2}-\binom{n}{3}+\dots+(-1)^n\binom{n}{n}=0$, for all natural numbers n.

Solution

(a) We have

$$(x+y)^n=\binom{n}{0}x^n+\binom{n}{1}x^{n-1}y+\binom{n}{2}x^{n-2}y^2+\dots+\binom{n}{n}y^n$$

When x = 1 and y = 1 then,

$$(1+1)^n=\binom{n}{0}1^n+\binom{n}{1}1^{n-1}1+\binom{n}{2}1^{n-2}1^2+\dots+\binom{n}{n}1^n$$

$$\therefore \quad 2^n = \binom{n}{0}+\binom{n}{1}+\binom{n}{2}+\dots+\binom{n}{n}.$$

= sum of binomial coefficient.

(b) We have

$$(x+y)^n=\binom{n}{0}x^n+\binom{n}{1}x^{n-1}y+\binom{n}{2}x^{n-2}y^2+\binom{n}{3}x^{n-3}y^3+\dots+\binom{n}{n}y^n$$

When x = 1 and y = −1, then

$$(1-1)^n=\binom{n}{0}1^n+\binom{n}{1}1^{n-1}(-1)+\binom{n}{2}1^{n-2}(-1)^2+\binom{n}{3}1^{n-3}(-1)^3+\dots+\binom{n}{n}(-1)^n$$

$$0=\binom{n}{0}-\binom{n}{1}+\binom{n}{2}-\binom{n}{3}+\dots+(-1)^n\binom{n}{n}.$$

Example 51

Expand $(2a+5)^5$ by the binomial theorem.

Solution

Here n = 5.

By using binomial theorem,

$$(2a+5)^5 = \binom{5}{0}(2a)^5+\binom{5}{1}(2a)^{5-1}(5)^1+\binom{5}{2}(2a)^{5-2}(5)^2+\binom{5}{3}(2a)^{5-3}(5)^3+\binom{5}{4}(2a)^{5-4}(5)^4+\binom{5}{5}(2a)^{5-5}(5)^5$$

$$=\frac{5!}{0!(5-0)!}.32a^5+\frac{5!}{1!(5-1)!}.(2a)^4.5+\frac{5!}{2!(5-2)!}(2a)^3.25+\frac{5!}{3!(5-3)!}(2a)^2.125+\frac{5!}{4!(5-4)!}(2a)^1.625+\frac{5!}{5!(5-5)!}(2a)^0.3125$$

$$=32a^5+400a^4+2000a^3+5000a^2+6250a+6250$$

Example 52

Write down the expansion of $(3x-5)^5$ and hence find the coefficient of x^3 in the expansion of $(x+1)(3x-5)^5$.

Solution

Comparing $(3x-5)^5$ with $(x+y)^n$ we get x = 3x and y = −5 and n = 5.

So using the coefficients from Pascal's triangle when n = 5, we have,

$$(3x-4)^5 = 1(3x)^5+5(3x)^4(-5)+10(3x)^3(-5)^2+10(3x^2)(-5)^3+5(3x)^1(-5)^4+1(-5)^5.$$

$$=243x^5-2025x^4+6750x^3-11250x^2+9375x-3125$$

Now,

$$(x+1)(3x-5)^5=(x+1)(243x^5-2025x^4+6750x^3-11250x^2+9370x-3125)$$

$$=243x^6-2025x^5+6750x^4-11250x^3+9370x^2-3125x+243x^5-2025x^4+6750x^3-11250x^2+9370x-3125$$

Therefore coefficient of $x^3 = -11250+6750$

$= -4500$

Example 53

Find the coefficient of the term, containing y^8 in the binomial expansion of $(x+3y^2)^{17}$.

Solution

The term containing y^8 in the expansion of $(x+3y^2)^{17}$ is $\binom{17}{4}x^{17-4}(3y^2)^4 = 2380x^{13}81y^8$.

The coefficient of the term containing y^8 is 2380 × 81 = 192780.

Example 54

Find the coefficients of x^{16} in the expansion of $\left(2x^2-\frac{x}{2}\right)^{12}$.

Solution

Comparing $\left(2x^2-\frac{x}{2}\right)^{12}$ with $(x+y)^n$ we get, $x = 2x^2$, $y=-\frac{x}{2}$ and n = 12

The general term in the expansion of this expression by the binomial theorem is,

$$t_{r+1} = C(12,r)(2x^2)^{12-r}\left(\frac{-x}{2}\right)^r$$

$$= C(12,r)\,2^{12-r}\left(\frac{-1}{2}\right)^r x^{2(12-r)+r}$$

$$= C(12,r)\,2^{12-2r}(-1)^r x^{24-r}$$

To get the coefficient of x^{16}, we put $24 - r = 16$, thus $r = 8$.

Then the coefficient is $C(12, 8)\ 2^{12-16}.\ (-1)^8$

$$= C(12, 8).\frac{1}{2^4}$$

$$= \frac{495}{16}$$

Example 55

Find the general term in the expansion of $\left(x^2 + \frac{a^2}{x}\right)^5$. Then find the coefficient of x.

Solution

Comparing $\left(x^2 + \frac{a^2}{x}\right)^5$ with $(x + y)^n$ we get $x = x^2$, $y = \frac{a^2}{x}$ and $n = 5$.

Since the general term in the expansion of $(x + y)^n$ is $t_{r+1} = C(n, r)x^{n-r}y^r$, the general term in the expansion of $\left(x^2 + \frac{a^2}{x}\right)^5$ is $t_{r+1} = C(5, r)\ (x^2)^{5-r}\left(\frac{a^2}{x}\right)^r$

$$= C(5, r)x^{10-2r}\frac{a^{2r}}{x^r}$$

$$= C(5, r)\ x^{10-3r}a^{2r}$$

To get the coefficient of x,

Let $10 - 3r = 1$

or $3r = 9$

$\therefore$ $r = 3$

$\therefore$ $t_{r+1} = t_{3+1} = t_4 = C(5, 3)xa^{2\times 3}$

$= C(5, 3)\ xa^6$

Thus the coefficient of x is $C(5, 3)a^6 = 10a^6$.

Example 56

Find the term independent of x in the expansion of $\left(2x + \frac{1}{x^2}\right)^9$.

Solution

Here, comparing $\left(2x + \frac{1}{x^2}\right)^9$ with $(x + y)^n$, we get $x = 2x$ and $y = \frac{1}{x^2}$ and $n = 9$

Since the general term in the expansion of $(x + y)^n$ is $t_{r+1} = C(n, r)\ x^{n-r}y^r$, then the general term in the expansion of $\left(2x + \frac{1}{x^2}\right)^9$ is,

$$t_{r+1} = C(9, r)\ (2x)^{9-r}\left(\frac{1}{x^2}\right)^r$$

$$= C(9, r)\ (2)^{9-r}\ (x)^{9-r}.\frac{1}{x^{2r}}$$

$$= C(9, r)\ (2)^{9-r}\ x^{9-3r}$$

To get the term which is independent of x,

Let $9 - 3r = 0$

$\therefore$ $r = 3$

$\therefore$ $t_{r+1} = t_{3+1} = t_4 = C(9, 3)\ (2)^{9-3}x^0 = C(9, 3)\ (2)^6 = 5376$, is the term independent of x.

Example 57

Find the 7^{th} term of $\left(x + \frac{1}{x}\right)^{10}$.

Solution

Comparing $\left(x + \frac{1}{x}\right)^{10}$ with $(x + y)^n$ we get, $x = x$ and $y \frac{1}{x}$ and $n = 10$

The general term t_{r+1} of the expression $\left(x + \frac{1}{x}\right)^{10}$ is,

$$t_{r+1} = C(10, r)\ x^{10-r}.\left(\frac{1}{a}\right)^r$$

$$= C(10, r)\ x^{10-r}.\ x^{-r}$$

$$= C(10, r)\ x^{10-2r}$$

To get t_7, we put $r = 6$ in t_{r+1}, then

$$t_{6+1} = t_7 = C(10, 6)\ x^{10-2\times 6}$$

$$= \frac{10!}{6!\ (10-6)!} \times x^{-2}$$

$$= \frac{210}{x^2}$$

Middle Term

Now let us find the middle term or terms in the expansion of $(x + y)^n$. We have to consider the case when n is an even number and when it is an odd number.

(i) When n is even:

When n is even, the number of terms in the expansion of $(x + y)^n$ is $n + 1$ which is odd. So there is exactly one middle term which is $t_{n/2+1}$

(ii) **When n is odd:**

When n is odd, the number of terms in the expansion of $(x + y)^n$ is n + 1 which is even. So there are two middle terms, which are $t_{(n+1)/2}$ and $t_{(n+1)/2+1}$

Example 58

(i) Find the middle term in the expansion of $(2a+3x)^{30}$

(ii) Find the middle term in the expansion of $\left(x-\frac{1}{x}\right)^{18}$

(iii) Find the middle term in the expansion of $(1+x/2)^{15}$

(iv) Find the middle terms in the expansion of $\left(2x+\frac{1}{2x}\right)^{15}$.

Solution:

(i) Comparing $(2a + 3x)^{30}$ with $(x + y)^n$, we get: $x = 2a$, $y = 3x$, $n = 30$

Here, n = 30 which is even so there is only one middle term.

Middle term $= (t_{30/2}) + 1 = t_{15+1}$

$= C(30,15)\,(2a)^{30-15}\,.\,(3x)^{15}$

$= C(30,15)\,(2a)^{15}\,.\,(3x)^{15}$

(ii) Here n = 18, is even. So there are (18 + 1) = 19 terms in the expansion of $\left(x-\frac{1}{x}\right)^{18}$, which is odd.

Therefore there exists exactly one middle term which is $t_{18/2+1}$ i.e. t_{10}.

$\therefore\; t_{10} = t_{9+1} = C(18, 9)\,(x)^{18-9}\left(-\frac{1}{x}\right)^{9}$

$= \frac{18!}{9!\,9!}\,.\,(x)^{9-9}(-1)^9$

$= -\frac{18!}{9!\,9!}$

(iii) Here the number of terms in the expansion is 15+1=16 which is even. So, there are two middle terms. The middle terms are $t_{(15+1)/2}$ and $t_{(15+1)/2+1}$ i.e. t_8 and t_9

$t_8 = t_{7+1} = C(15, 7)\,(1)^{15-7}\left(\frac{x}{2}\right)^7 = \frac{15!\,x^7}{7!8!\,2^7}$

$t_9 = t_{8+1} = C(15, 8)\,(1)^{15-8}\left(\frac{x}{2}\right)^8 = \frac{15!\,x^8}{7!8!\,2^8}$

(iv) Here the number of terms in the expansion is 15 + 1 = 16 which is even. So there are two middle terms, which are $t_{(15+1)/2}$ and $t_{(15+1)/2+1}$ i.e. t_8 and t_9

$t_8 = t_{7+1} = C(15, 7)\,(2x)^{15-7}\left(\frac{1}{2x}\right)^7$

$= \frac{15!}{7!\,8!}(2x)^{8-7}$

$= \frac{15!}{7!\,8!}(2x)$

$t_9 = t_{8+1} = C(15, 8)\,(2x)^{15-8}\left(\frac{1}{2x}\right)^8$

$= \frac{15!}{8!\,7!}(2x)^{7-8}$

$= \frac{15!}{8!\,7!\,(2x)}$

Pascal's Triangle

The geometrical arrangement of binomial coefficient in the expansion of $(x + y)^n$ in a triangular form is called Pascal's triangle. i.e. the coefficients of successive powers of x+y can be arranged in a triangular array. This triangular array of numbers is called Pascal's triangle. Which is shown in the following equations.

$(a + b)^0 = 1$

$(a + b)^1 = a + b$

$(a + b)^2 = a^2 + 2ab + b^2$

$(a + b)^3 = a^3 + 3a^2b + 3ab^2 + b^3$

$(a + b)^4 = a^4 + 4a^3b + 6a^2b^2 + + 4ab^3 + b^4$

$(a + b)^5 = a^5 + 5a^4b + 10a^3b^2 + 10a^2b^3 + 5ab^4 + b^5$

....

The coefficients of the successive powers of a + b can be arranged in a triangular array of numbers, called Pascal's triangle. This can be shown as,

Coefficient in $(a + b)^0$

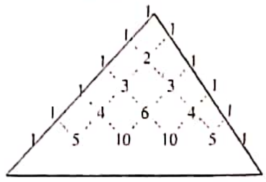
Coefficients in $(a + b)^1$

Coefficients in $(a + b)^2$

Coefficients in $(a + b)^3$

Coefficients in $(a + b)^4$

Coefficients in $(a + b)^5$



The numbers in Pascal's triangle have the following properties:

(i) The first and the last number in each row is 1.

(ii) Every other number in the array can be obtained by adding the two numbers appearing directly above it.

Example 59

Write down the expansion of $(1 + y)^6$, using Pascal's theorem.

Solution

Here n = 6, so we use Pascal's triangle up to n = 6.

When n = 0	1
When n = 1	1 1
When n = 2	1 2 1
When n = 3	1 3 3 1
When n = 4	1 4 6 4 1
When n = 5	1 5 10 10 5 1
When n = 6	1 6 15 20 15 6 1

$$\therefore \quad (1 + y)^6 = 1(1)^6 + 6(1)^5 y + 15(1)^4(y)^2 + 20(1)^3(y)^3 + 15(1)^2(y)^4 + 6(1)^1(y)^5 + 1(1)^0(y)^6$$

$$= 1 + 6y + 15y^2 + 20y^3 + 15y^4 + 6y^5 + y^6$$

Example 60

Prove that: $\binom{n+1}{r} = \binom{n}{r-1} + \binom{n}{r}$

Solution

$$\text{R.H.S.} = \binom{n}{r-1} + \binom{n}{r}$$

$$= \frac{n!}{(r-1)!\,(n-r+1)!} + \frac{n!}{r!\,(n-r)!}$$

$$= \frac{n!\,.r}{r(r-1)!\,(n-r+1)!} + \frac{n!\,.(n-r+1)}{(n-r+1)\,r!\,(n-r)!}$$

$$= \frac{n!.r}{r!\,(n-r+1)!} + \frac{n!.(n-r+1)}{r!\,(n-r+1)!}$$

$$= \frac{n!\,.r + n!.(n-r+1)}{r!\,(n-r+1)!}$$

$$= \frac{n!\,(n+1)}{r!\,(n-r+1)!}$$

$$= \frac{(n+1)!}{r!\,(n-r+1)!}$$

$$= \binom{n+1}{r} = \text{L.H.S.}$$

which is also known as Pascal's Identity.

The Inclusion–Exclusion Principle

Let A and B be any finite sets. Then $n(A \cup B) = n(A) + n(B) - n(A \cap B)$.

In other words, to find the number $n(A \cup B)$ of elements in the union $A \cup B$, we add n(A) and n(B) and then we subtract $n(A \cap B)$; that is we include n(A) and n(B), and we exclude $n(A \cap B)$. This follows from the fact that when we add n(A) and n(B), we have counted the elements of $A \cap B$ twice. This principle holds true for any number of sets. For three finite sets,

$$n(A \cup B \cup C) = n(A) + n(B) + n(C) - n(A \cap B) - n(A \cap C) - n(B \cap C) + n(A \cap B \cap C).$$

that is, we include n(A), n(B), n(C) we exclude $n(A \cap B)$, $n(A \cap C)$, $n(B \cap C)$ and we include $n(A \cap B \cap C)$. Suppose there are a finite number of finite sets, say, $A_1, A_2, A_3, \ldots A_m$. Let S_K be the sum of the cardinalities $n(A_{i1} \cap A_{i2} \cap \ldots\ldots \cap A_{iK})$ of all possible K-tuple intersections of the given m sets. Then the general inclusion –exclusion principle is

$$n(A_1 \cup A_2 \cup \ldots \cup A_m) = S_1 - S_2 + S_3 - \ldots + (-1)^{m-1} S_m$$

Example 61

Find the number of mathematics students at a college taking at least one of the languages French, German and Russian given the following data:

65 study French	20 study French and German
45 study German	25 study French and Russian
42 study Russian	15 study Germany and Russian
	8 study all three languages.

Solution

Let F, G and R denotes the sets of students studying French, German and Russian respectively.

Then by the Inclusion – exclusion principle,

$$n(F \cup G \cup R) = n(F) + n(G) + n(R) - n(F \cap G) - n(F \cap R) - n(G \cap R) + n(F \cap G \cap R).$$

$$= 65 + 45 + 42 - 20 - 25 - 15 + 8 = 100$$

Thus 100 students study at least one of the languages.

Example 62

Let A, B, C and D denote art, biology, chemistry and drama courses respectively. Find the number N of students in a dormitory given the data:

12 take A,	5 take A and B,	3 take A, B, C
20 take B,	7 take A and C,	2 take A, B, D
20 take C,	4 take A and D,	2 take B, C, D
8 take D,	16 take B and C,	3 take A, C, D
	4 take B and D,	2 take all four
	3 take C and D,	71 take none.

Solution

Let T be the number of students who take at least one course where total number of students is N. So

$T = S_1 - S_2 + S_3 - S_4$, where

$S_1 = 12 + 20 + 20 + 8 = 60$

$S_2 = 5 + 7 + 4 + 16 + 4 + 3 = 39$

$S_3 = 3 + 2 + 2 + 3 = 10$ and $S_4 = 2$

Thus $T = 29$ and then $N = 71 + T = 100$

EXERCISE (A)

1. (i) How many automobiles license plates can be made if each plate contains two different letters followed by three different digits?

   (ii) Solve the problem if the first digit cannot be zero.
2. Find the number of permutations of five different objects taken three at a time.
3. If three persons enter a bus in which there are ten vacant seats, find in how many ways they can seat.
4. If a student is offered admission to 4 different Engineering colleges and 5 different medical colleges, find the number of ways of choosing one of the above colleges.
5. Find the minimum number of students needed to guarantee that five of them belong to the same class (Freshman, Sophomore, Junior, Senior).
6. State the Pigeonhole principle. How many students must be in a class to guarantee that at least two students receive the same score on the final exam is graded on a scale from 0 to 100?

7. Find the minimum number of students in a class to guarantee that three of them belong to the same zone (There are 14 zones in Nepal).
8. From a group of 11 men and 8 women, how many committees consisting of 3 men and 2 women are possible?
9. From 6 gentlemen and 4 ladies, a committee of 5 is to be formed. In how many ways can this be done so as to include at least one lady?
10. How many committees of five people can be chosen from 20 men and 12 women?

    (i) If exactly three men must be on each committee?

    (ii) If at least four women must be on each committee?
11. In how many ways can ten adults and five children stand in a circle so that no two children are next to each other?
12. (i) In how many ways can ten boys and four girls sit in a row?

    (ii) In how many ways can they sit in a row if the boys are to sit together and the girls are to sit together too?

    (iii) In how many ways can they sit in a row if the girls are to sit together?
13. How many permutations are there of the letters of the word "Tennessee"?
14. (i) How many permutations can be made out of the letters of the word 'COMPUTER'?

    (ii) How many of these begin with R?

    (iii) How many of these begin with C?

    (iv) How many of these begin with C and end with R?

    (v) How many of these never have C and R together?
15. How many permutations can be made out of the letters of word 'SUNDAY'? How many of these

    (a) Begin with s

    (b) End with y

    (c) Begin with s and end with y

    (d) s and y always comes together
16. Find the number of permutations that can be formed from the letters of word DAUGHTER?

    (a) How many of these begin with D and end with R?

    (b) How many of these have vowels always comes together?

    (c) How many of these have not all vowels comes together?

17. In how many ways can the letters of word MONDAY be arranged?
   (i) How many of these arrangements do not begin with M?
   (ii) How many begin with M and do not end with Y?

18. For each of the following, expand either using Binomial theorem or Pascal's theorem and simplify.
   (i) $(x + y)^6$, (ii) $(2x + 3y)^6$

19. Find the coefficient x^{10} in the expansion of $(1+x^2)^{10}$

20. What is the coefficient of $x^5 y^7$ in the expansion of $(x - 2y)^{12}$?

21. Find the general term in the expansion of $\left(\frac{a}{b}+\frac{b}{a}\right)^{2n+1}$

22. Find the coefficient of x^5 in the expansion of $\left(x+\frac{1}{2x}\right)^7$

23. What is the coefficient of x^{27} in the binomial expansion of $\left(\frac{3}{x}+x^2\right)^{18}$?

24. Find the term independent of x in the expansion of
   i. $\left(2x+\frac{1}{3x^2}\right)^9$ ii. $\left(\frac{3x^2}{2}-\frac{1}{3x}\right)^9$

25. Find the term free from x in the expansion of $\left(x+\frac{1}{x}\right)^{2n}$

26. Consider the binomial expansion of $(4x - 5y)^{10}$.
   (i) How many terms are there altogether?
   (ii) Find the coefficient of x^3y^7 in the binomial expansion of $(4x - 5y)^{10}$.

27. Find the middle term or terms in the expansion of
   i. $\left(ax+\frac{1}{ax}\right)^{17}$ ii. $\left(x+\frac{1}{x}\right)^{18}$
   iii. $\left(ax-\frac{1}{ax}\right)^{2n}$

General form of Linear Homogeneous Recurrence Relation

- The homogeneous recurrence relation of degree k_1 with constant coefficient has the general form

$$a_n = c_1a_{n-1} + c_2a_{n-2} + \ldots + c_ka_{n-k} \qquad \ldots\ldots(i)$$

If $a_n = r^n$ is a solution of eq^n. (i), then, it must satisfy eq^n. (i) i.e.

$$r^n = c_1r^{n-1} + c_2r^{n-2} + \ldots + c_kr^{n-k}$$

Dividing by r^{n-k} on both side, we get

$$r^k = c_1r^{k-1} + c_2r^{k-2} + \ldots\ldots + c_k$$

or, $$r^k - c_1r^{k-1} - c_2r^{k-2} - \ldots\ldots - c_k = 0 \qquad \ldots\ldots(iii)$$

This equation is known as characteristic equation of given recurrence relation and it provides characteristics roots of recurrence relation which are used to give on explicit formula for all the solution of recurrence relation.

Solving Linear Homogeneous Recurrence Relations with Constant Coefficients

A linear homogeneous recurrence relation of degree k with constant coefficients is a recurrence relation of the form $a_n = c_1a_{n-1} + c_2a_{n-2} + \ldots + c_ka_{n-k}$, where $c_1, c_2, \ldots, c_k$ are real numbers, and $c_k \neq 0$. The above relation is linear since right hand side is a sum of the multiples of previous terms of the sequence. It is homogeneous since no term occurs without being multiple of some a_j's. All the coefficients of the terms are constants and degree k is due to the representation of a_n in terms of previous 'k' terms of the sequence.

In solving the recurrence relation of the type above, the approach is to look for the solution of the form $a_n = r^n$, where r is a constant. $a_n = r^n$ is a solution of a recurrence relation $a_n = c_1a_{n-1} + c_2a_{n-2} + \ldots + c_ka_{n-k}$ if and only if $r^n = c_1r^{n-1} + c_2r^{n-2} + \ldots + c_kr^{n-k}$.
when we divide both sides by r^{n-k} and transpose the right hand side we have

$$r^k - c_1r^{k-1} - c_2r^{k-2} - \ldots - c_k = 0.$$

Here we can say $a_n = r^n$ is a solution if and only if r is the solution if the equation $r^k - c_1r^{k-1} - c_2r^{k-2} - \ldots - c_k = 0$ (characteristic equation of the recurrence relation) and solutions to this equations are called characteristic roots of the recurrence relation.

Theorem 1: (without proof)
Let c_1 and c_2 be real numbers. Suppose that $r^2 - c_1r - c_2 = 0$ has two distinct roots r_1 and r_2. Then the sequence $\{a_n\}$ is a solution of the recurrence relation $a_n = c_1a_{n-1} + c_2a_{n-2}$ if and only if $a_n = \alpha_1r_1^n + \alpha_2r_2^n$ for n = 0, 1, 2, ..., where $\alpha 1$ and $\alpha 2$ are constants.

Example 1

Solve the recurrence relation $a_n = 5a_{n-1} - 6a_{n-2}$ for $n \geq 2$, $a_0 = 1$ and $a_1 = 0$.

Solution

The given recurrence relation is

$a_n = 5a_{n-1} - 6a_{n-2}$ ......(i)

The characteristic equation is

$r^2 - 5r + 6 = 0$

i.e. $r^2 - 3r - 2r + 6 = 0$

i.e. $r(r - 3) - 2(r - 3) = 0$

i.e. $(r - 2)(r - 3) = 0$

$r = 2, 3$ i.e. $r_1 = 2$, $r_2 = 3$

Since, two characteristics roots are (different) distinct, we use the "theorem 1" to write the general solution.

The general form of solution is

$a_n = \alpha_1 r_1 + \alpha_2 r_2^n$

$a_n = \alpha_1 2^n + \alpha_2 3^n$ ....(ii)

From initial conditions

$a_0 = \alpha_1 2^0 + \alpha_2 . 3^0$

$1 = \alpha_1 + \alpha_2$ i.e. $\alpha_1 = 1 - \alpha_2$ ....(iii)

and $a_1 = \alpha_1 2^1 + \alpha_2 3^1$

$0 = 2\alpha_1 + 3\alpha_2$ ....(iv)

i.e. $0 = 2(1 - \alpha_2) + 3\alpha_2$

or, $0 = 2 - 2\alpha_2 + 3\alpha_2$

or, $0 = 2 + \alpha_2$

or, $\alpha_2 = -2$

and $\alpha_1 = 1 - \alpha_2$

$= 1 - (-2)$

$= 1 + 2 = 3$

Therefore, the solution of given recurrence relation is

$a_n = \alpha_1 r_1^n + \alpha_2 r_2^n$

or, $a_n = 3.2^n + (-2) . 3^n$

$a_n = 3.2^n - 2.3^n$

Example 2

Solve the recurrence relation $a_n = 6a_{n-1} - 8a_{n-2}$ for $n \geq 2$, $a_0 = 4$, $a_1 = 10$.

Solution

The given recurrence relation is

$a_n = 6a_{n-1} - 8a_{n-2}$ .....(i)

The characteristic equation is

$r^2 - 6r + 8 = 0$

$r^2 - 4r - 2r + 8 = 0$

$r(r - 4) - 2(r - 4) = 0$

i.e. $(r - 4)(r - 2) = 0$

i.e. $r = 4, 2$ i.e. $r_1 = 4$, $r_2 = 2$

Since roots are distinct, the general form of solution is

$a_n = \alpha_1 r_1^n + \alpha_2 r_2^n$

$a_n = \alpha_1 4^n + \alpha_2 . 2^n$ ....(ii)

From initial conditions, we have

$a_0 = \alpha_1 4^0 + \alpha_2 . 2^0$

$4 = \alpha_1 + \alpha_2$ ......(iii)

and $a_1 = \alpha_1 .4 + \alpha_2 .2^1$

$10 = 4.\alpha_1 + 2\alpha_2$ ......(iv)

From equation (iii) and (iv), we have

$10 = 4(4 - \alpha_2) + 2\alpha_2$

$10 = 16 - 4\alpha_2 + 2\alpha_2$

$-6 = -2\alpha_2$ i.e. $\alpha_2 = 3$

and, substituting the value of α_2 in (iii), we have

$4 = \alpha_1 + \alpha_2 \Rightarrow 4 = \alpha_1 + 3 \Rightarrow \alpha_1 = 1$

Therefore, the solution of given recurrence relation is

$a_n = 1.4^n + 3.2^n$

Example 3

What is the solution of recurrence relation $a_n = a_{n-1} + 2a_{n-2}$ with $a_0 = 2$ and $a_1 = 7$?

Solution

Here the given recurrence relation is

Scanned by CamScanner

$a_n=a_{n-1}+2a_{n-2}$

And the initial conditions are $a_0=2$ and $a_1=7$.

Now, we have a characteristic equation for the above given recurrence relation as

$r^2-r-2=0$

Now solving this equations by factoring

$r^2-2r+r-2=0$

$r(r-2)+1(r-2)=0$

i.e $(r-2)(r+1)=0$ i.e either $r=2$ or $r=-1$.

Hence the roots of characteristic equation are $r_1=2$ and $r_2=-1$, both are distinct.

Hence the solution sequence is:

$a_n = \alpha_1 r_1^n + \alpha_2 r_2^n$ with $r_1 = 2$ and $r_2 = -1$

i.e. $a_n = \alpha_1 2^n + \alpha_2(-1)^n$ ....(1) for some constants α_1 and α_2

From the initial conditions, we put the value of a_0 and a_1 in (1) and get the equations as

$a_0=2=\alpha_1+\alpha_2$.....................(2)

$a_1=7=\alpha_1 2+\alpha_2(-1)$..............(3)

Solving these two equations, we have

$\alpha_1=3$ and $\alpha_2=-1$.

Hence the solution to the given recurrence relation and initial condition is the sequence $\{a_n\}$ with $a_n = 3.2^n - 1.(-1)^n$

Example 4

Solve the recurrence relation $a_n = a_{n-1} + 6a_{n-2}$ for $n > 2$, $a_0 = 3$, $a_1 = 6$.

Solution

Characteristic equation of the given relation is $r^2 - r - 6 = 0$. Its roots are $r = 3$ and $r = -2$. Since $(r - 3)(r + 2) = 0$. Hence, the sequence {an} is a solution to the recurrence relation if and only if $a_n = \alpha_1 3^n + \alpha_2(-2)^n$, for some constants α_1 and α_2. From the initial conditions we have $\alpha_0 = 3 = \alpha_1 + \alpha_2$, $\alpha_1 = 6 = 3\alpha_1 + (-2)\alpha_2$. Solving these two equations we have $\alpha_1 = 12/5$ and $\alpha_1 = 3/5$. Hence, the solution is the sequence $\{a_n\}$ with $a_n = (12.3^n + 3(-2)^n)/5$.

Example 5

Find the solution of recurrence relation $f_n = f_{n-1} + f_{n-2}$, $n > 2$, and $f_0 = 0$, $f_1 = 1$.

OR

Find the explicit formula for fibonancci sequences.

Solution

The characteristics equation of recurrence relation

$f_n = f_{n-1} + f_{n-2}$ is $r^2 - r - 1 = 0$ .....(i)

Comparing $r^2 - r - 1 = 0$ with $ax^2 + bx + c = 0$, we get

$a = 1, b = -1, c = -1$

Now,

$$\text{Roots} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} = \frac{-(-1) \pm \sqrt{(-1)^2 - 4\times1\times(-1)}}{2} = \frac{1 \pm \sqrt{1+4}}{2} = \frac{1 \pm \sqrt{5}}{2}$$

∴ Taking +ve sign, we get

$$r_1 = \frac{1+\sqrt{5}}{2}$$

Taking -ve sign, we get

$$r_2 = \frac{1-\sqrt{5}}{2}$$

Since characteristics roots are different, the general form of solution is

$f_n = \alpha_1 r_1^n + \alpha_2 r_2^n$

$$f_n = \alpha_1\left(\frac{1+\sqrt{5}}{2}\right)^n + \alpha_2\left(\frac{1-\sqrt{5}}{2}\right)^n \quad ....(ii)$$

From initial condition we have,

$f_0 = \alpha_1 + \alpha_2$

$0 = \alpha_1 + \alpha_2$ ....(iii)

and $$f_1 = \alpha_1\left(\frac{1+\sqrt{5}}{2}\right) + \alpha_2\left(\frac{1-\sqrt{5}}{2}\right) = 1 \quad ....(iv)$$

From equation (iii) and (iv) we have,

$$-\alpha_2\left(\frac{1+\sqrt{5}}{2}\right) + \alpha_2\left(\frac{1-\sqrt{5}}{2}\right) = 1$$

$$\frac{-\alpha_2 - \alpha_2.\sqrt{5} + \alpha_2 - \alpha_2.\sqrt{5}}{2} = 1$$

$$-2\alpha_2\sqrt{5} = 2$$

$$\alpha_2 = -\frac{2}{2\sqrt{5}} = -\frac{1}{\sqrt{5}}$$

and $\alpha_1 + \alpha_2 = 0$

$$\alpha_1 - \frac{1}{\sqrt{5}} = 0$$

$$\alpha_1 = \frac{1}{\sqrt{5}}$$

$\therefore$ Solution of given recurrence is

$$f_n = \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^n - \frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^n$$

Example 6

What is the solution of recurrent relation $a_n = a_{n-1} - 2a_{n-2}$ with initial conditions $a_0 = 2$, $a_1 = 7$

Solution

The given recurrence relation is:

$a_n = a_{n-1} - 2a_{n-2}$ .....(i)

The characteristics equation is

$r^2 - r - 2 = 0$

or, $r^2 - 2r + r - 2 = 0$

or, $r(r-2) + 1(r-2) = 0$

$(r-2)(r+1) = 0$

$\therefore$ $r = 2, 1$

i.e. $r_1 = 2$, $r_2 = 1$

Since roots are distinct, the general form of solution is

$a_n = \alpha_1 r_1^n + \alpha_2 r_2^n$

$a_n = \alpha_1 . 2^n + \alpha_2 . 1^n$ ....(ii)

From initial conditions we have,

$a_0 = \alpha_1 . 2^0 + \alpha_2 . 1^0$

$\therefore$ $2 = \alpha_1 + \alpha_2$

or, $\alpha_1 = \alpha_2 - 2$ ....(iii)

and $a_1 = \alpha_1 . 2^1 + \alpha_2 . 1^1$

$7 = 2\alpha_1 + \alpha_2$ ....(iv)

From equation (ii) and (iv), we have,

$7 = 2(\alpha_2 - 2) + \alpha_2$

or, $7 = 2\alpha_2 - 4 + \alpha_2$

or, $3\alpha_2 = 11$ $\therefore$ $\alpha_2 = \frac{11}{3}$

Now, substituting value of α_2 in equation (iii), we have

$\alpha_1 = \alpha_2 - 2$

$\alpha_1 = \frac{11}{3} - 2 = \frac{11-6}{3} = \frac{5}{3}$

Therefore, the solution of given recurrence relation is

$$\boxed{a_n = \frac{5}{3}2^n + \frac{11}{3}1^n}$$

Theorem 2: (without proof)

Let c1 and c2 be real numbers with $c2 \neq 0$. Suppose that $r^2 - c_1r - c2 = 0$ has only one root r_0. Then the sequence $\{a_n\}$ is a solution of the recurrence relation $a_n = c_1a_{n-1} + c_2a_{n-2}$ if and only if $a_n = \alpha_1 r_0{}^n + \alpha_2 n r_0{}^n$ for n = 0, 1, 2, ..., where α1 and α2 are constants.

Example 7

Solve the recurrence relation $a_n = 6\,a_{n-1} - 9\,a_{n-2}$ for $n \geq 2$, $a_0 = 1$, $a_1 = 6$.

Solution

Characteristic equation of the given relation is

$r^2 - 6r + 9 = 0$

or $(r-3)^2 = 0$

or r=3,3

Therefore Its only one root is r = 3.

Hence, the sequence $\{a_n\}$ is a solution to the recurrence relation if and only if

$an = \alpha_1 3^n + \alpha_2 n3^n$, for some constants α1 and α2.

From the initial conditions we have

$a_0 = 1 = \alpha_1$,

$a_1 = 6 = \alpha_1 3 + \alpha_2 .3$

Solving these two equations we have $\alpha_1 = 1$ and $\alpha_2 = 1$.

Hence, the solution is the sequence $\{a_n\}$ with

$a_n = 3^n + n3^n.z$

Example 8

Solve the recurrence relation $a_n = 2a_{n-1} - a_{n-2}$, for $n \geq 2$, $a_0 = 4$, $a_1 = 1$.

Solution

The given recurrence relation is

$$a_n = 2a_{n-1} - a_{n-2} \quad \text{......(i)}$$

The characteristic equation is

$$r^2 - 2r + 1 = 0$$

or, $(r - 1)^2 = 0$

or, $r = 1, 1$

Since, characteristic roots are same, the general form of solution is

$$a_n = \alpha_1 . r_0^n + \alpha_2 n . r_0^n$$

i.e. $a_n = \alpha_1 . 1^n + \alpha_2 . n . 1^n$ ....(ii)

From given initial conditions,

$$a_0 = \alpha_1 . 1^0 + \alpha_2 . 0 . 1^n$$

i.e. $4 = \alpha_1 . 1 \Rightarrow \alpha_1 = 4$

And $a_1 = \alpha_1 . r_0^n + \alpha_2 . n . r_0^n$

$1 = \alpha_1 . 1^1 + \alpha_2 . 1 . 1^1$

$1 = \alpha_1 + \alpha_2$ ......(iii)

Putting value of $\alpha_1 = 4$ in equation (iii), we have

$1 = 4 + \alpha_2 \Rightarrow \alpha_2 = -3$

Therefore, the solution of given recurrence relation,

$$a_n = 4.1^n + (-3).n.1^n$$

Example 9

Solve the recurrence relation $a_n = 2a_{n-1} - a_{n-2}$ for $n \geq 2$, $a0 = 3$, $a1 = 6$.

Solution

Characteristic equation of the given relation is

$$r^2 - 2r + 1 = 0.$$

i.e. $(r - 1)^2 = 0$

i.e. $r = 1, 1$

Therefore, its only one root is $r = 1$.

Hence, the sequence $\{a_n\}$ is a solution to the recurrence relation if and only if

$a_n = \alpha_1 1^n + \alpha_2 n 1^n$, for some constants α_1 and α_2.

From the initial conditions we have

$$a_0 = 3 = \alpha_1,$$

$$a_1 = 6 = \alpha_1 + \alpha_2.$$

Solving these two equations we have $\alpha_1 = 3$ and $\alpha_2 = 3$.

Hence, the solution is the sequence $\{a_n\}$ with

$$a_n = 3(1^n + n1^n).$$

Example 10

Solve: $a_n = -6a_{n-1} - 9a_{n-2}$ for $n \geq 2$, $a_0 = 5$, $a_1 = -1$

Solution

The given recurrence relation is

$$a_n = -6a_{n-1} - 9a_{n-2}$$

The characteristic equation is

$$r^2 + 6r + 9 = 0$$

$\Rightarrow$ $r^2 + 3r + 3r + 9 = 0$

$r(r + 3) + 3(r + 3) = 0$

or, $(r + 3)(r + 3) = 0$

$r = -3, -3$

Since the roots of characteristics equation are same, we have the general form of solution is

$$a_n = \alpha_1 . r_0^n + \alpha_2 . n . r_0^n$$

i.e. $a_n = \alpha_1 (-3)^n + \alpha_2 . n . (-3)^n$

From given initial condition,

$$a_0 = \alpha_1 (-3)^0 + \alpha_2 . 0 . (-3)^0$$

$5 = \alpha_1 . 1 + 0 \Rightarrow a_1 = 5$

and, $a_1 = \alpha_1 (-3)^1 + \alpha_2 . 1 . (-3)^1$

i.e. $-1 = \alpha_1 (-3) + \alpha_2 (-3)$

i.e. $-1 = -3\alpha_1 - 3\alpha_2$

i.e. $-1 = -3 \times 5 - 3\alpha_2$

$\Rightarrow$ $-1 + 15 = -3\alpha_2$

$\Rightarrow$ $\alpha_2 = 14/-3 = -\frac{14}{3}$

Therefore, the solution of given recurrence relation is

$$a_n = \alpha_1 (-3)^n + \alpha_2 n(-3)^n$$

i.e. $a_n = 5(-3)^n + (-14/3).\ n\ (-3)^n$

i.e. $a_n = 5(-3)^n - \frac{14}{3}n\ (-3)^n$

Theorem 3: (without proof)

Let $c_1, c_2, \ldots, c_k$ be real numbers. Suppose that $r^k - c_1r^{k-1} - \ldots - c_k = 0$ has k distinct roots $r_1, r_2, \ldots, r_k$. Then the sequence $\{a_n\}$ is a solution of the recurrence relation $a_n = c_1a_{n-1} + c_2a_{n-2} + \ldots + c_ka_{n-k}$ if and only if $a_n = \alpha_1r_1^n + \alpha_2r_2^n + \ldots + \alpha_kr_k^n$ for $n = 0, 1, 2, \ldots$, where $\alpha_1, \alpha_2, \ldots, \alpha_k$ are constants.

Example 11

Find the solution to $a_n = 2a_{n-1} + a_{n-2} - 2a_{n-3}$, for $n \geq 3$ with $a_0 = 3$, $a_1 = 6$ and $a_2 = 0$

Solution

The given recurrence is

$$a_n = 2a_{n-1} + a_{n-2} - 2a_{n-3} \quad \ldots\ldots(i)$$

The characteristic equation is

$$r^3 - 2r^2 - r + 2 = 0$$

or, $r^3 - r^2 - r^2 - r + 2 = 0$

or, $r^3 - r^2 - r^2 - 2r + r + 2 = 0$

or, $r^3 - r^2 - r^2 + r - 2r + 2 = 0$

or, $r^2(r - 1) - r(r - 1) - 2(r - 1) = 0$

or, $(r - 1)(r^2 - r - 2) = 0$

or, $(r - 1)(r^2 - 2r + r - 2) = 0$

or, $(r - 1)[r(r - 2) + 1(r - 2)] = 0$

or, $(r - 1)(r - 2)(r + 1) = 0$

$\therefore$ $r_1 = 1, r_2 = 2, r_3 = -1$

Since, all three roots are different, we can write the general form of solution as

$$a_n = \alpha_1 r_1^n + \alpha_2r_2^n + \alpha_3r_3^n$$

$$a_n = \alpha_1.1^n + \alpha_2.2^n + \alpha_3(-1)^n \quad \ldots\ldots(ii)$$

From given initial conditions, we have

$$a_0 = \alpha_1.\ 1^0 + \alpha_2.\ 2^0 + \alpha_3(-1)^0$$

$$3 = \alpha_1 + \alpha_2 + \alpha_3 \quad \ldots\ldots(iii)$$

Again,

$$a_1 = \alpha_1.\ 1 + \alpha_2.\ 2^1 + \alpha_3(-1)^1$$

$$6 = \alpha_1 + 2\alpha_2 - \alpha_3 \quad \ldots\ldots(iv)$$

and $a_2 = \alpha_1.\ 1^2 + \alpha_2.\ 2^2 + \alpha_3(-1)^2$

$$0 = \alpha_1 + 4\alpha_2 + \alpha_3 \quad \ldots\ldots(v)$$

From (iii) and (v)

$$3 = \alpha_1 + \alpha_2 + \alpha_3$$

$$0 = \alpha_1 + 4\alpha_2 + \alpha_3$$

$$3 = -3\alpha_2 \Rightarrow \alpha_2 = -1$$

From (iii) and (iv),

$$3 = \alpha_1 + \alpha_2 + \alpha_3$$

$$6 = \alpha_1 + 2\alpha_2 - \alpha_3$$

$$-3 = -\alpha_2 + 2\alpha_3$$

$\Rightarrow$ $2\alpha_3 = -3 + \alpha_2 = -3 - 1 = -4$

$\therefore$ $\alpha_3 = -\frac{4}{2} = -2$

From (iii),

$$\alpha_1 + \alpha_2 + \alpha_3 = 3$$

$$\alpha_1 = 3 - \alpha_2 - \alpha_3$$

$$\alpha_1 = 3 - (-1) - (-2)$$

$$= 3 + 1 + 2 = 6$$

Therefore, the solution of given recurrence relation is

$$a_n = \alpha_1.\ 1^n + \alpha_2.\ 2^n + \alpha_3(-1)^n$$

$$a_n = (-1)\ 1^n + 6.\ 2^n + (-2)(-1)^n$$

$$= -1.\ 1^n + 6.\ 2^n - 2(-1)^n$$

Example 12

Solve the recurrence relation $a_n = 2a_{n-1} + a_{n-2} - 2a_{n-3}$ for $n \geq 3$, $a_0 = 3$, $a_1 = 6$ and $a_2 = 9$.

Solution

Characteristic equation of the given relation is $r^3 - 2r^2 - r + 2 = 0$. Its roots are $r = 1, r = -1$, and $r = 2$.

Hence, the sequence $\{a_n\}$ is a solution to the recurrence relation if and only if $a_n = \alpha_1 1^n + \alpha_2(-1)^n + \alpha_3 2^n$, for some constants α_1, α_2, and α_3.

From the initial conditions we have

$a_0 = 3 = \alpha_1 + \alpha_2 + \alpha_3,$

$a_1 = 6 = \alpha_1 - \alpha_2 + 2\alpha_3,$

and $a_2 = 9 = \alpha_1 + \alpha_2 + 4\alpha_3.$

Solving these three equations we have $\alpha_1 = 3/2$, $\alpha_2 = -1/2$, and $\alpha_3 = 2$. Hence, the solution is the sequence {an} with

$a_n = (3/2)1^n - (1/2)(-1)^n + 2.2^n.$

Example 13

Find the solution of recurrence relation $a_n = 6a_{n-1} - 11a_{n-2} + 6a_{n-3}$, for $n > 3$, $a_0 = 2$, $a_1 = 5$ and $a_2 = 5$.

Solution

The given recurrence relation is

$a_n = 6a_{n-1} - 11a_{n-2} + 6a_{n-3}$

The characteristic equation is

$r^3 - 6r^2 + 11r - 6 = 0$

Since r = 1 satisfy this equation, r = 1 is a root of this equation, so we try to find the factor (r - 1) from this equation

$r^3 - r^2 - 5r^2 + 5r + 6r - 6 = 0$

$r^2(r - 1) - 5r(r - 1) + 6(r - 1) = 0$

$(r - 1)(r^2 - 5r + 6) = 0$

$(r - 1)(r^2 - 3r - 2r + 6) = 0$

$(r - 1)\{r(r - 3) - 2(r - 3)\} = 0$

$(r - 1)(r - 3)(r - 2) = 0$

$r_1 = 1, r_2 = 3, r_3 = 2$

Since all three roots are different, we have the general form of solution is

$a_n = \alpha_1 r_1^n + \alpha_2 r_2^n + \alpha_3 r_3^n$ i.e. $a_n = \alpha_1 1^n + \alpha_2 . 3^n + \alpha_3 . 2^n$

From initial condition

$a_0 = \alpha_1 . 1^0 + \alpha_2 . 3^0 + \alpha_3 . 2^0$

$2 = \alpha_1 + \alpha_2 + \alpha_3$ ......(i)

Again,

$a_1 = \alpha_1 . r_1^n + \alpha_2 . r_2^n + \alpha_3 . r_3^n$

$5 = \alpha_1 . 1^1 + \alpha_2 . 3^1 + \alpha_3 . 2^1$

$\Rightarrow$ $5 = \alpha_1 + 3\alpha_2 + 2\alpha_3$ ....(ii)

$a_2 = \alpha_1 . 1^2 + \alpha_2 . 3^2 + \alpha_3 . 2^2$

$5 = \alpha_1 + 9\alpha_2 + 4\alpha_3$ ....(iii)

Solving (i) and (ii)

$2 = \alpha_1 + \alpha_2 + \alpha_3$

$5 = \alpha_1 + 3\alpha_2 + 2\alpha_3$

$-3 = -2\alpha_2 - \alpha_3$

$\therefore$ $3 = 2\alpha_2 + \alpha_3$ ......(iv)

Solving (i) and (iii) we get

$2 = \alpha_1 + \alpha_2 + \alpha_3$

$5 = \alpha_1 + 9\alpha_2 + 4\alpha_3$

$-3 = -8\alpha_2 - 3\alpha_3$

$\therefore$ $3 = 8\alpha_2 + 3\alpha_3$ ....(v)

Multiplying eq^n. (iv) by 3 and subtracting (v), we get

$9 = 6\alpha_2 + 3\alpha_3$

$3 = 8\alpha_2 + 3\alpha_3$

$6 = -2\alpha_2$

$\therefore$ $\alpha_2 = -3$

Now, substituting value of α_2 in eq^n. (iv)

$3 = 2\alpha_2 + \alpha_3$

$3 = 2 \times (-3) + \alpha_3$

$\alpha_3 = 3 + 6$

$\alpha_3 = 9$

Again,

Substituting value of α_2 and α_3 in eq^n. (i)

$2 = \alpha_1 + \alpha_2 + \alpha_3$

or, $2 = \alpha_1 - 3 + 9$

or, $2 = \alpha_1 + 6$

or, $\alpha_1 = -4$

Therefore, solution of given recurrence relation is

$a_n = -4 . 1^n - 3 . 3^n + 9 . 2^n$

From the initial conditions we have

$a_0 = 3 = \alpha_1 + \alpha_2 + \alpha_3,$

$a_1 = 6 = \alpha_1 - \alpha_2 + 2\alpha_3,$

and $a_2 = 9 = \alpha_1 + \alpha_2 + 4\alpha_3.$

Solving these three equations we have $\alpha_1 = 3/2$, $\alpha_2 = -1/2$, and $\alpha_3 = 2$. Hence, the solution is the sequence {an} with

$a_n = (3/2)1^n - (1/2)(-1)^n + 2.2^n.$

Example 13

Find the solution of recurrence relation $a_n = 6a_{n-1} - 11a_{n-2} + 6a_{n-3}$, for $n > 3$, $a_0 = 2$, $a_1 = 5$ and $a_2 = 5$.

Solution

The given recurrence relation is

$a_n = 6a_{n-1} - 11a_{n-2} + 6a_{n-3}$

The characteristic equation is

$r^3 - 6r^2 + 11r - 6 = 0$

Since r = 1 satisfy this equation, r = 1 is a root of this equation, so we try to find the factor (r - 1) from this equation

$r^3 - r^2 - 5r^2 + 5r + 6r - 6 = 0$

$r^2(r - 1) - 5r(r - 1) + 6(r - 1) = 0$

$(r - 1)(r^2 - 5r + 6) = 0$

$(r - 1)(r^2 - 3r - 2r + 6) = 0$

$(r - 1)\{r(r - 3) - 2(r - 3)\} = 0$

$(r - 1)(r - 3)(r - 2) = 0$

$r_1 = 1, r_2 = 3, r_3 = 2$

Since all three roots are different, we have the general form of solution is

$a_n = \alpha_1 r_1^n + \alpha_2 r_2^n + \alpha_3 r_3^n$ i.e. $a_n = \alpha_1 1^n + \alpha_2 . 3^n + \alpha_3 . 2^n$

From initial condition

$a_0 = \alpha_1 . 1^0 + \alpha_2 . 3^0 + \alpha_3 . 2^0$

$2 = \alpha_1 + \alpha_2 + \alpha_3$ ......(i)

Again,

$a_1 = \alpha_1 . r_1^n + \alpha_2 . r_2^n + \alpha_3 . r_3^n$

$5 = \alpha_1 . 1^1 + \alpha_2 . 3^1 + \alpha_3 . 2^1$

$\Rightarrow$ $5 = \alpha_1 + 3\alpha_2 + 2\alpha_3$ ....(ii)

$a_2 = \alpha_1 . 1^2 + \alpha_2 . 3^2 + \alpha_3 . 2^2$

$5 = \alpha_1 + 9\alpha_2 + 4\alpha_3$ ....(iii)

Solving (i) and (ii)

$2 = \alpha_1 + \alpha_2 + \alpha_3$

$5 = \alpha_1 + 3\alpha_2 + 2\alpha_3$

$-3 = -2\alpha_2 - \alpha_3$

$\therefore$ $3 = 2\alpha_2 + \alpha_3$ ......(iv)

Solving (i) and (iii) we get

$2 = \alpha_1 + \alpha_2 + \alpha_3$

$5 = \alpha_1 + 9\alpha_2 + 4\alpha_3$

$-3 = -8\alpha_2 - 3\alpha_3$

$\therefore$ $3 = 8\alpha_2 + 3\alpha_3$ ....(v)

Multiplying eq^n. (iv) by 3 and subtracting (v), we get

$9 = 6\alpha_2 + 3\alpha_3$

$3 = 8\alpha_2 + 3\alpha_3$

$6 = -2\alpha_2$

$\therefore$ $\alpha_2 = -3$

Now, substituting value of α_2 in eq^n. (iv)

$3 = 2\alpha_2 + \alpha_3$

$3 = 2 \times (-3) + \alpha_3$

$\alpha_3 = 3 + 6$

$\alpha_3 = 9$

Again,

Substituting value of α_2 and α_3 in eq^n. (i)

$2 = \alpha_1 + \alpha_2 + \alpha_3$

or, $2 = \alpha_1 - 3 + 9$

or, $2 = \alpha_1 + 6$

or, $\alpha_1 = -4$

Therefore, solution of given recurrence relation is

$a_n = -4 . 1^n - 3 . 3^n + 9 . 2^n$

Solving (iv) and (v),

$\alpha_2 + 3\alpha_3 = 19$

$\alpha_2 + \alpha_3 = 1$

$2\alpha_3 = 18$

$\Rightarrow \quad \alpha_3 = 9$

and, $\quad \alpha_2 + 3\alpha_3 = 19$

$\Rightarrow \quad \alpha_2 = 19 - 3\alpha_3$

$= 19 - 3 \times 9$

$\alpha_2 = 19 - 27 = -8$

Therefore, the solution of given recurrence relation is

$a_n = (-5 + (-8) . n + 9.n^2) * (-1)^n$

Example 15

Solve the recurrence relation $a_n = 2a_{n-1}$, $n \geq 1$, with initial condition $q_0 = 3$.

Solve:

The characteristics equation of recurrence relation $a_n = 2a_{n-1}$ is:

$r^1 - 2 = 0$

$r = 2$

Since r=2 with multiplicity m=1, the general form of solution is

$a_n = \alpha r^n$ .....(i)

i.e. $\quad a_n = \alpha 2^n$

From initial condition,

$a_0 = 3$, we have

$a_1 = \alpha 2^n$

i.e. $\quad a_0 = \alpha 2^0$

or, $\quad 3 = \alpha 2^0 \Rightarrow \alpha = 3$

Hence, the solution of given R.R. is

$a_n = 3.2^n$ .....(i)

Solving Linear Non-homogeneous Recurrence Relations with Constant Coefficients

The recurrence relation of the form

$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \ldots + c_k a_{n-k} + F(n)$,

where c1, c2, ..., ck are real numbers and F(n) is a function depending upon n. The recurrence relation preceding F(n) is called **associated homogeneous recurrence relation.**

For example an =7an-1 + 3an-2 +6n is a linear non-homogeneous recurrence relation with constant coefficients.

Theorem 5: (without proof)

If $\{a_n(p)\}$ is a particular solution of the non-homogeneous linear recurrence relation with constant coefficients $a_n = c_1 a_{n-1} + c_2 a_{n-2} + \ldots + c_k a_{n-k} + F(n)$, then every solution of the form

$\{a_n(p) + a_n(h)\}$,

where $a_n(h)$ is a solution of the associated homogeneous recurrence relation $a_n = c_1 a_{n-1} + c_2 a_{n-2} + \ldots + c_k a_{n-k}$.

Example 16

Find all the solutions of the recurrence relation $a_n = 4a_{n-1} + n^2$. Also find the solution of the relation with initial condition $a_1 = 1$.

Solution

We have associated linear homogeneous recurrence relation as $a_n = 4a_{n-1}$. The root is 4, so the solutions are $a_n(h) = \alpha 4^n$, where α is a constant. Since $F(n) = n^2$ is a polynomial of degree 2, a trial solution is a quadratic function in n, say, $p_n = an^2 + bn + c$, where a, b, and c are constants.

To determine whether there are any solution of this form, suppose that

$p_n = an^2 + bn + c$ is such solution.

Then the equation $a_n = 4a_{n-1} + n^2$ becomes

$an^2 + bn + c = 4(a(n-1)^2 + b(n-1) + c) + n^2.$

$= 4a\, n^2 - 8an + 4a + 4bn - 4b + 4c + n^2$

$= (4a + 1)n^2 + (-8a + 4b)n + (4a - 4b + 4c)$

Here $an^2 + bn + c$ is the solution if and only if

$4a + 1 = a$ i.e. $a = -1/3$;

$-8a + 4b = b$ i.e. $b = -8/9$;

$4a - 4b + 4c = c$ i.e. $c = -28/27$.

So $a_n(p) = -1/3(n^2 + 8/3.n + 28/9)$ is a particular solution and all solutions are

$a^n = \{a_n(p) + a_n(h)\} = 1/3(n^2 + 8/3.n + 28/9) + \alpha 4^n$, where α is a constant.

For solution with $a_1 = 1$, we have

$a_1 = 1 = -1/3(1 + 8/3 + 28/9) + \alpha 4$ i.e. $\alpha = 22/27$.

Then the solution is $an = -1/3(1 + 8/3 + 28/9) + 22/27\ 4^n$.

Example 17

Find all the solution of recurrence relation $a_n = 2a_{n-1} + 3^n$ and a solution with initial condition $a_1 = 5$.

Solution

The given recurrence relation is $a_n = 2a_{n-1} + 3^n$ ....(i)

The linear homogeneous part of above eq^n. is

$a_n = 2a_{n-1}$ .....(ii)

Then, characteristic equation is

$r - 2 = 0 \Rightarrow r = 2$

Now, the solution of equation (ii) is

$a_n = \alpha_1 r_1^n$

$= \alpha_1 2^n$ .....(iii)

To find the particular solution, we write

$F(n) = 3^n$ .....(iv)

The trial solution of (iv) is

$a(p) = c.3^n$ .....(v) where c is constant

Substituting terms of this sequence into recurrence relation, we have

$a_n = 2a_{n-1} + 3^n$ [Put $a_n^{(p)} = c_3^n$ in both side]

i.e. $c.3^n = 2.c.3^{n-1} + 3^n$

$$c.3^n = 2.c.\frac{3^n}{3} + 3^n$$

$$c.3^n = 3^n\left(\frac{2c}{3} + 1\right)$$

Comparing the coefficient of 3^n,

$$c = \frac{2c}{3} + 1 \Rightarrow 3c = 2c + 3,\ c = 3$$

Therefore, the particular solution is

$a_n^{(p)} = 3.3^n$

and the all solution of given R.R. is

$a_n = a_n^{(h)} + a_n^{(p)}$

$a_n = \alpha_1.2^n + 3.3^n = \alpha_1.2^n + 3^{n+1}$

From given conditions,

$a_1 = \alpha_1.2^1 + 3^{1+1}$

$5 = 2\alpha_1 + 9$

$$5 - 9 = 2\alpha_1 \Rightarrow \alpha_1 = -\frac{4}{2} = -2$$

Hence, the solution of given R.R. is

$a_n = -2.2^n + 3^{n+1}$

Theorem 6: (without proof)

Suppose that (an) satisfies the linear non-homogeneous recurrence relation $a_n = c_1a_{n-1} + c_2a_{n-2} + \dots + c_ka_{n-k} + F(n)$, where c1, c2, ..., ck are real numbers and $F(n) = (b_tn^t + b_{t-1}n^{t-1} + \dots + b_1n + b_0)\, s^n$, where $b_0, b_1, \dots, b_t$ and s are real numbers.

When s is not a root of the characteristic equation of the associated linear homogeneous recurrence relation, there is a particular solution of the form $(p_tn^t + p_{t-1}n^{t-1} + \dots + p_1n + p_0)s^n$.

When s is a root of the characteristic equation and its multiplicity is m, there is a particular solution of the form

$n^m(p_tn^t + p_{t-1}n^{t-1} + \dots + p_1n + p_0)s^n$.

Example 18

Find the solution of the recurrence relation $a_n = 2a_{n-1} + n.2^n$.

Solution

We have the associated linear homogeneous recurrence relation is

$a_n(h) = 2a_{n-1}$.

The characteristic equation for this would be $r-2 = 0$, so the root is 2 and hence the solution is $a_n(h) = \alpha_1 2^n$, where α is a constant.

We have $F(n) = n.2^n$. (Of the form $n.s^n$) where s is the root of the characteristic equation and the multiplicity of 2 is 1 so, the particular solution has the form

$c = n.(p_1n)2^n$

or $a_n(p) = p_1n^2 2^n$.

The all solution is $a_n = \alpha_1 2^n + p_1n^2 2^n$.

Recurrences Applications

One of the application areas of recurrence relations is analysis of divide and conquers algorithms.

Divide and Conquer Algorithms

Divide and conquer algorithms divide a problem of larger size to the problem of smaller size so continually such that the problem of the smallest size that has trivial solution is obtained. If f(n) represents the number of operations required to solve the problem of size n, then follows the recurrence relation f(n) = af(n/b) + g(n), called divide and conquer recurrence relation. In the relation above the problem of size n is partitioned into a parts of the problem of the size n/b and g(n) is the operations required to conquer the solutions. In this section no algorithms are presented but their recurrence relations are tried to achieve.

Example 19: Fibonacci Numbers

We know that the Fibonacci numbers are generated by the formula fn = fn-1 + fn-2. Here nth Fibonacci number is the sum of $(n-1)^{th}$ and $(n-2)^{nd}$ Fibonacci numbers. Here for the initial conditions are f0 = 0, and f1 = 1. Use of the above relation does not exactly produce the recurrence relation mentioned above, however this is an example of divide and conquer algorithm since each time the problem is changed into two problems of smaller size.

Example 20: Merge Sort

In merge sorting the input sequence of n items is broken down into 2 halves (here there may be difference in 1 item between two parts). Since the list of size n need more comparisons than list of size n/2, the problem here is simplified. This process continues until all the comparisons are trivial. This problem satisfies the divide and conquer recurrence relation

M(n) = 2M(n/2) + O(1).

EXERCISE (B)

1. State pigeon hole principle? Find the minimum number of student in a class to be sure that four of them are born in the same day of week.
2. Define linear homogeneous recurrence relation with example.
3. Find an explicit formula for the Fibonacci numbers with recurrence relation $f_n = f_{n-1} + f_{n-2}$ with $f_1 = 1$ and $f_2 = 2$.
4. Define linear homogeneous recurrence relation with degree k with constant coefficient. What is the solution of recurrence relation $a_n = 4a_{n-1} + 4a_{n-2}$ with $a_0 = 3$ and $a_1 = 5$?
5. Let $\{a_n\}$ be a sequences that satisfies the recursion relation $a_n = a_{n-1} - a_{n-2}$ for $n \geq 2$ and suppose that $a_0 = 3$ and $a_1 = 5$. Find the value of a_2 and a_3.
6. Describe linear homogeneous and non-linear homogeneous recurrence relation with examples.

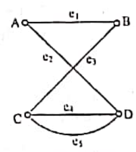
5
Chapter

Graph Theory

Introduction

Many situations that occur in computer science, physical science, communication science, Economics and many others areas can be analyzed by using techniques found in a relatively new area of mathematics called graph theory. Graphs can be used to represent almost any problem involving discrete arrangements of objects, where concern lies not with the internal properties of these objects but with relationships among them.

Graph is a discrete structure consisting of vertices and edges connecting the vertices. A graph G=(V,E) is a mathematical model consist of two non empty sets : V = $\{v_1, v_2, v_3, v_4, \ldots\ldots v_n\}$ called set of vertices and E=$\{e_1, e_2, e_3, e_4, \ldots\ldots e_n\}$ of ordered or un-ordered pairs of distinct vertices called edges. We usually write G=(V,E) and say V is the vertex set and E is the edge-set of G.



G = (V, E)

In above figure, graph G = (V, E) consist of set of vertices V = {A, B, C, D} and set of edges E = $\{e_1, e_2, e_3, e_4, e_5\}$ where e_1 = {A, B}, e_2 = {A, D}, e_3 {B, C}, e_4 = {C, D}, e_5 = {D, C}.

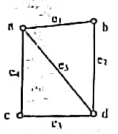
Applications of Graph

Graphs are used to solve problems in many fields. Some of them are as follows:

- Graphs are used to model the geographic maps of cities in which each place in city can be represented by the node and the road connecting such places are represented by an arc (edge).
- Graphs are used to model the computer network in which each node is a machine (computer, hub, router, switch etc) and the link between them represents the edge.
- They are used to analyze the electrical circuits, project planning, genetics etc.
- Any structured problem can be modeled by graphs. Then can help to solve typical problems those concerned with finding shortest path or most economical route between two vertices, or the smallest set of edges which connect all the vertices in a graph.
- Graphs are used to study the structure of WWW.
- It is used to find the number of different combination of flight between two cities in an airline network.
- It can be used to distinguish between two chemical compounds with the same molecular formula but different structure.
- It is also used to assign channels to television stations.

Simple Graph

We define a simple graph as 2 – tuple consists of a non empty set of vertices V and a set of unordered pairs of distinct elements of vertices called edges, having neither loop nor parallel edges. A graph G = (V,E) is said to be simple if G has no lops and no parallel edges.

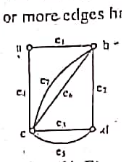


G = (V, E)

Here, V(G) = {a, b, c, d}
E(G) = {e_1, e_2, e_3, e_4, e_5}
Where,
e_1 = {a, b}, e_2 = {b, d},
e_3 = {d, c}, e_4 = {c, a}, e_5 = {a, d}
In this graph, none of the edges are parallel and does not contain loop so, it is simple graph.

Multi-graph

A computer network may contain multiple links between data centers, to model such networks we need graphs that have more than one edge connecting the same pair of vertices. Two or more edges between same pair of vertices are called parallel edges or two or more edges having same end points are called parallel edges.



G = (V, E)

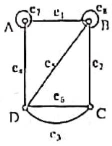
Here, V(G) = {a, b, c, d}
E(G) = {e_1, e_2, e_3, e_4, e_5, e_6, e_7}
Where,
e_1 = {a, b}, e_2 = {b, d}, e_3 = {d, c}, e_4 = {c, a},
e_5 = {d, c}, e_6 = {c, b}, e_7 = {c, b}
In this graph, the edges e_3 and e_5 are drawn between same pair of vertices. Similarly, e_6 and e_7 also so such edges are called parallel and graph is multi-graph.

A graph G = (V, E) consists of a set of vertices V, a set of edges E such that some of edges are parallel edges is called multigraph.

Pseudograph

Sometimes a communication link connects a data center with itself, perhaps a feedback loop for diagnostic purposes. to model such a network we need to include edges that connects a vertex to itself such a edge is called loop.

A graph G =(V, E) consists of a set of vertices V, a set of edges E is said to be pseudo graph if G has both loops and multiple edges or loops only.



G = (V, E)

Here, V(G) = {A, B, C, D}
E(G) = {e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8}
Where,
e_1 = {A, B}, e_2 = {B, C}, e_3 = {C, D}, e_4 = {D, A},
e_5 = {B, D}, e_6 = {D, C}, e_7 = {A, A}, e_8 = {B, B}
In this graph, the edge e_7 is drawn from vertex A to A and e_8 is drawn from vertex B to B. So, such edges are loop and graph is called pseudo-graph.

Order and size of Graph

If G=(V,E) be a finite graph then the number of vertices in graph G is called order of graph G and the number of edges in G is called size of G.

Degree of vertex

Let G = (V, E) be a graph and v be a vertex of G. The degree (or valency) of v is denoted by d(v) is the number of edges incident on v.

A vertex v in graph G is said to be even vertex if its degree is even and a vertex v in G is said to be odd vertex if its degree is odd.

Isolated vertex and pendant vertex

In a graph G=(V,E), a vertex having degree zero(0) is called isolated vertex and vertex having degree one(1) called pendant vertex.

Degree Sequence of a Graph

If $v_1, v_2, \ldots\ldots v_n$ are n vertices of G_1 then the sequence $(d_1, d_2, \ldots.., d_n)$ where $d_i = \deg(v_i)$ is the degree sequence of G. In general, we order the vertices so that the degree sequence is monotonically increasing i.e.

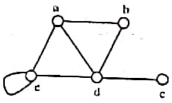
$$\delta(G) = d_1 \le d_2 \le \ldots\ldots \le d_n = \Delta(G),$$

Where,

$$\delta(G) = \min\{\deg v : \in V(G)\}$$

$$\Delta(G) = \max\{\deg v : \in V(G)\}$$

Example 1



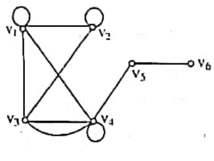
G = (V, E)

In this graph, deg(a) = 3, deg(b) = 2, deg(c) = 1, deg(d) = 4, deg(e) = 4.
Therefore, degree sequence of G is :
(1, 2, 3, 4, 4)

Adjacent vertices and adjacent edges

Two vertices u and v in an undirected graph are called adjacent if there is an edge between u and v.

Two edges e_1 and e_2 in an undirected graph are said to be adjacent if they share a common vertex.



G = (V, E)

In above graph G, Order of graph = no. of vertices = 6, Size of graph= no of edges=9
$\deg(v_1)=5$, $\deg(v_2)=4$, $\deg(v_3)=4$, $\deg(v_4)=6$, $\deg(v_5)=1$ $\deg(v_6)=0$
Vertex v_5 has degree 1 so it is pendent vertex and v_6 has degree 0 so it is isolated vertex.
Degree sequence of G = (0, 1, 4, 4, 5, 6)
Adjacent vertices of v_1 are v_2, v_3 and v_4 but vertices v_5 and v_6 are non-adjacent to v_1.
Adjacent edges of edge $\{v_1,v_2\}$ are $\{v_1,v_1\}$, $\{v_1,v_3\}$, $\{v_1,v_4\}$, $\{v_2,v_3\}$ and $\{v_2,v_2\}$

Representation of Graph

Graph can be represented in many ways. One of the representation method of graph is Matrix representation, based on adjacency of vertices, called adjacency matrix and another is based on incidence of vertices and edges called incidence matrix. The other representation of graph is adjacency list, based on the list of adjacent vertices.

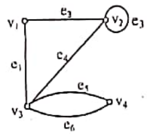
Adjacency Matrix

Let G=(V,E) be a graph with n vertices: $v_1, v_2, v_3, \ldots\ldots v_n$. The adjacency matrix of G with respect to given ordered list of vertices is a n×n matrix denoted by $A(G) = (a_{ij})_{n\times n}$ such that

$$a_{ij} = \begin{cases} 0 \text{ if there is no edge between the vertices } v_i \text{ an } v_j \\ 1 \text{ if there is an edge between the vertices } v_i \text{ an } v_j \\ K \text{ If there are } K(\ge 2) \text{ edges between the vertices } v_i \text{ an } v_j \end{cases}$$

Example 2

Find the adjacency matrix to represent the graph shown in figure given below.



Solution

Since G has four vertices, adjacence matrix A(G) will be a 4 × 4 matrix

$$A(G) = \begin{array}{c c} & \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\ \begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} & \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 2 \\ 0 & 0 & 2 & 0 \end{bmatrix} \end{array}$$

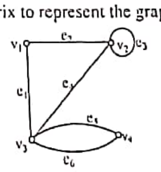
Incidence Matrix

Let G be a graph with vertices $v_1, v_2, \ldots, v_m$ and edges $e_1, e_2, \ldots e_n$. The incidence matrix I(G) of graph G is a m×n matrix with $I(G) = (m_{ij})_{m\times n}$, where

$$m_{ij} = \begin{cases} 1 \text{ if } e_j \text{ is incident with } v_i \\ 0 \text{ if } e_j \text{ is not incident with } v_i \\ 2 \text{ if } v_i \text{ is the end of the loop } e_j \end{cases}$$

Example 3

Find the incidence matrix to represent the graph shown in figure below



Solution

Since G ahs four vertices and six edges, incidence matrix I(G) will be a 4 × 6 matrix.

$$I(G) = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \begin{array}{c} \begin{array}{cccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \end{array} \\ \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \end{array}$$

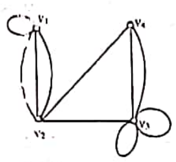
Example 4

Draw the graph G corresponding to the following adjacency matrix.

$$A = \begin{bmatrix} 1 & 3 & 0 & 0 \\ 3 & 0 & 1 & 1 \\ 0 & 1 & 2 & 2 \\ 0 & 1 & 2 & 0 \end{bmatrix}$$

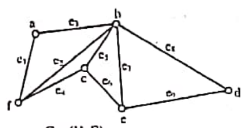
Solution

Since A is a 4 – square matrix, G has four vertices say v_1, v_2, v_3 and v_4. Then G is



Example 5

Find the adjacency matrix and incidence matrix of the graph given below.



G = (V, E)

Solution

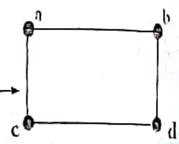
Let the order of the vertices be a, b, c, d, e, f and edges order be e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9

$$A(G) = \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \\ f \end{array} \begin{array}{c} \begin{array}{cccccc} a & b & c & d & e & f \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix} \end{array} \quad I(G) = \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \\ f \end{array} \begin{array}{c} \begin{array}{ccccccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 & e_9 \end{array} \\ \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

Adjacency List

One of the ways of representing a graph without multiple edges is by listing its edges. This type of representation is suitable for the undirected graphs without multiple edges, and directed graphs. This representation looks as in the tables below.

Edge List for Simple Graph	
vertex	Adjacent Vertices
a	b,c
b	a,d
c	a,d
d	b,c



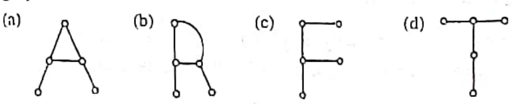
Isomorphism of Graphs

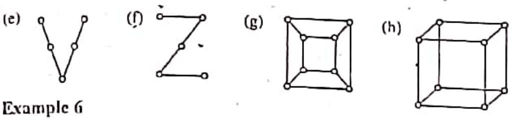
Two graphs $G_1(V_1, E_1)$ and $G_2(V_2, E_2)$ are isomorphic if there exists a bijective mapping ϕ between $V_1(G_1)$ and $V_2(G_2)$ such that $\{u, v\}$ is in E_1 if and only if $\{\phi(u), \phi(v)\}$ is in E_2. The function ϕ is called isomorphism.

if two graph G_1 and G_2 are isomorphic then they must have:

i. same number of vertices i.e. $|V_1|=|V_2|$
ii. same number of edges i.e. $|E_1|=|E_2|$
iii. if $\{u,v\}$ in $E_1 \rightarrow \{\phi(u), \phi(v)\}$ in E_2
iv. if $\deg(u)=k$ in G_1 then $\deg(\phi(u)) = k$ in G_2
v. $A(G_1)=A(G_2)$ with respect to orders of vertices $v_1, v_2, v_3, \ldots, v_n$ and $\phi(v_1), \phi(v_2), \phi(v_3), \ldots\ldots\ldots \phi(v_n)$.

In figure below, graphs in (a) and (b), (c) and (d), (e) and (f), and (g) and (h) are isomorphic graphs.





Example 6

Show that the two graphs shown below are isomorphic

Solution

Here,

$V(G_1) = \{1, 2, 3, 4\}$, $V(G_2) = \{a, b, c, d\}$

$E(G_1) = \{(1, 2), (2, 3), (3, 4)\}$

$E(G_2) = \{(a, b), (b, d), (c, d)\}$

Define a function f: $V(G_1) \rightarrow V(G_2)$ as

$f(1) = a$, $f(2) = b$, $f(3) = d$, $f(4) = c$

Then f is one-to-one onto and

$(1, 2) \in E(G_1)$ and $\{f(1), f(2)\} = (a, b) \in E(G_2)$.

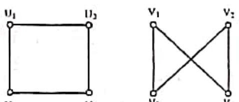
$(2, 3) \in E(G_1)$ and $\{f(2), f(3)\} = (b, d) \in E(G_2)$.

$(3, 4) \in E(G_1)$ and $\{f(3), f(4)\} = (d, c) \in E(G_2)$.

Thus G_1 and G_2 are isomorphic.

Example 7

Show that the graphs G_1 and G_2 are isomorphic.



Solution

The isomorphic invariants for two graphs are:

No. of vertices in G_1 $|V(G_1)| = 4$

No. of edges in G_1 $|E(G_1)| = 4$

No. of vertices in G_2 $|V(G_2)| = 4$

No. of edges in G_2 $|E(G_2)| = 4$

In graph G_1, there are four vertices each of degree 2 i.e. (2, 2, 2, 2) similar is true in graph G_2 also.

Since both graphs agree so many isomorphic invariants so, it is reasonable to find an isomorphison ϕ.

Let ϕ: $V(G_1) \rightarrow V(G_2)$ defined by

$\phi(U_1) = V_1$, $\phi(U_2) = V_4$, $\phi(U_3) = V_3$ and $\phi(U_4) = V_2$

Now, adjacency matrices with respect to the ordering of vertices (U_1, U_2, U_3, U_4) and $(\phi(U_1), \phi(U_2), \phi(U_3), \phi(U_4))$ are

$$A(G_1) = \begin{array}{c} \\ U_1 \\ U_2 \\ U_3 \\ U_4 \end{array} \begin{array}{c} \begin{array}{cccc} U_1 & U_2 & U_3 & U_4 \end{array} \\ \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

$$A(G_2) = \begin{array}{c} \\ V_1 \\ V_2 \\ V_3 \\ V_4 \end{array} \begin{array}{c} \begin{array}{cccc} V_1 & V_2 & V_3 & V_4 \end{array} \\ \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

Since: $A(G_1) = A(G_2)$ with respect to ordering of vertices so, ϕ is isomorphism and G_1 and G_2 are isomorphic.

Example 8

Show that graphs G_1 and G_2 given below are not isomorphic.

Solution

The isomorphic invariants of two graphs are

$|V(G_1)| = |V(G_2)| = 6$

$|E(G_1)| = |E(G_2)| = 9$

Degree sequence of G_1: (3, 3, 3, 3, 3, 3)

Degree sequence of G_2: (3, 3, 3, 3, 3, 3)

Since both graphs agree so many invariants so, it is reasonable to find an isomorphism ϕ.

Let $\phi : V(G_1) \rightarrow V(G_2)$ defined by

$\phi(1) = V_1$, $\phi(2) = V_2$, $\phi(3) = V_3$, $\phi(4) = V_4$, $\phi(5) = V_6$, $\phi(6) = V_5$

Now,

Adjacency matrices with respect to ordering of vertices in ϕ are

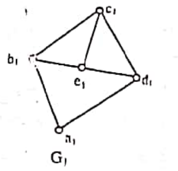
$$A(G_1) = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array}\begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix} \end{array}$$

$$A(G_2) = \begin{array}{c} \\ V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_6 \\ V_5 \end{array}\begin{array}{c} \begin{array}{cccccc} V_1 & V_2 & V_3 & V_4 & V_6 & V_5 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 \end{bmatrix} \end{array}$$

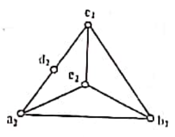
Since: $A(G_1) \neq A(G_2)$ with respect to ordering of vertices so, ϕ is not isomorphism and G_1 and G_2 are not isomorphic.

Example 9

Show that the following graphs G_1 and G_2 are isomorphic by showing their corresponding adjacency matrices are equal.



G_1



G_2

Solution

Consider the map $\phi: G_1 \rightarrow G_2$ defined as $\phi(a_1) = d_2$, $\phi(b_1) = a_2$, $\phi(c_1) = b_2$, $\phi(d_1) = c_2$ and $\phi(e_1) = e_2$. The adjacency matrix of G_1 for the ordering a_1, b_1, c_1, d_1 and e_1 is

$$A(G_1) = \begin{array}{c} \\ a_1 \\ b_1 \\ c_1 \\ d_1 \\ e_1 \end{array}\begin{array}{c} \begin{array}{ccccc} a_1 & b_1 & c_1 & d_1 & e_1 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

The adjacency matrix of G_2 for the ordering d_2, a_2, b_2, c_2 and e_2 is;

$$A(G_2) = \begin{array}{c} \\ d_2 \\ a_2 \\ b_2 \\ c_2 \\ e_2 \end{array}\begin{array}{c} \begin{array}{ccccc} d_2 & a_2 & b_2 & c_2 & e_2 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

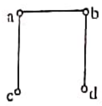
$\therefore$ G_1 and G_2 are isomorphic.

Self-complementary Graph

A graph $G = (V, E)$ is said to be self-complementary if G is isomorphic to its complement graph.

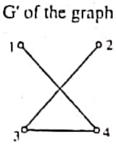
Example 10

Show that the following graph is self-complementary.



Solution

As we know, a graph is self-complementary if it is isomorphic to its complement, we now find a complement G' of the graph G(given)



Here $V(G) = \{a, b, c, d\}$, $E(G) = \{a, b\}, \{a, c\}, \{b, d\}$

$V(G') = \{1, 2, 3, 4\}$, $E(G') = \{(1, 4), (2, 3), (3, 4)\}$

Define a function $\phi: V(G) \rightarrow V(G')$ as

$\phi(c) = 1$, $\phi(d) = 2$, $\phi(a) = 4$, $\phi(b) = 3$

Then ϕ is one-to-one onto and

$(a, c) \in E(G)$ we have $\{\phi(a), \phi(c)\} = (4, 1) \in E(G')$.

$(a, b) \in E(G)$ we have $\{\phi(a), \phi(b)\} = (4, 3) \in E(G')$.

$(b, d) \in E(G)$ we have $\{\phi(b), \phi(d)\} = (3, 2) \in E(G')$.

Since G is isomorphic to its complement G', it is self complementary.

Theorem: (The Handshaking theorem)

The sum of the degrees of the vertices of a graph is equal to twice the number of edges.

Proof: Consider a graph G=(V,E) with n vertices $v_1, v_2, \ldots, v_n$ and |E|, the number of edges in G. Since each edge contributes a degree of 2, the sum of the degrees of all the vertices in G is twice the number of edges in G. That is $\sum_{i=1}^{n} d(v_i) = 2|E|$.

Theorem

The number of odd vertices in a graph is always even.

Proof:

Let G be a graph. If G contains no odd vertices, then the result follows immediately. Suppose that G contains K number of odd vertices which are $v_1, v_2, \ldots v_k$ and n number of even vertices which are $u_1, u_2, \ldots u_n$. Clearly, $d(u_1) + d(u_2) + \ldots + d(u_n) = \text{even}$

By Handshaking theorem,

$$[d(v_1) + d(v_2) + \ldots + d(v_k)] + [d(u_1) + d(u_2) + \ldots + d(u_n) + \ldots + d(u_n)] = 2|E|.$$

Where |E| is the total number of edges in G.

Then,

$$\begin{aligned} d(v_1) + d(v_2) + \ldots + d(v_k) &= 2|E| - [d(u_1) + d(u_2) + \ldots + d(u_n)] \\ &= 2|E| - \text{even} \\ &= \text{even} \end{aligned}$$

Here, sum of the degrees of odd vertices is even, to hold this condition K must be even i.e. G has an even number of odd vertices. If G has no even vertices then we have

$$d(v_1) + d(v_2) + \ldots + d(v_k) = 2|E|, \text{ is even}$$

From which we again conclude that K is even. This completes the proof of the theorem.

Example 11

Find the sum m of the degrees of the vertices of the graph G(V, E) where V(G) = {v_1, v_2, v_3, v_4, v_5} and E(G) = {(v_1, v_2), (v_1, v_4), (v_1, v_5), (v_2, v_5), (v_2, v_3), (v_3, v_4)}

Solution

Number of edges in G = |E| = 6

Sum of degrees of vertices in G = $\sum_{i=1}^{5} d(v_i) = 2|E| = 2 \times 6 = 12$

Example 12

How many vertices do the following graphs have if they contain

(a) 16 edges and all vertices of degree 2.

(b) 21 edges, 3 vertices of degree 4 and all other vertices of degree 3.

Solution

Let S be the sum of degrees of vertices, |E| be the number of edges and n be the number of vertices in G. Then,

(a) |E| = 16, let there be n vertices of degrees 2.

Since S = 2|E|, then

Number of vertices × equal degree of each vertex = 2|E|

or $n \times 2 = 2 \times 16$

$\therefore$ n = 16

(b) Since S = 2|E|, then

$3 \times 4 + (n-3) \times 3 = 2 \times 21$

or $12 + 3n - 9 = 42$

or $3n = 39$

$\therefore$ n = 13

Example 13

Show that the maximum numbers of edges in a simple graph with n vertices is $\frac{1}{2}n(n-1)$.

Solution

By the handshaking theorem,

$$\sum_{i=1}^{n} d(v_i) = 2|E|,$$

Where |E| is the number of edges in G with n vertices.

Or $d(v_1) + d(v_2) + \ldots + d(v_n) = 2|E|$

Since maximum degree of each vertex in the graph G can be (n − 1), then

$(n-1) + (n-1) + \ldots$ to n terms = 2|E|

or $n(n-1) = 2|E|$

$\therefore$ $|E| = \frac{1}{2}n(n-1)$.

Special types of Graphs

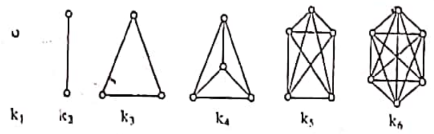
Some important types of graphs are introduced here. These graphs are often used as examples and arise in many applications.

Trivial Graph

A graph with one vertex and no edges is called a trivial graph

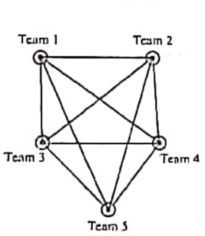
Complete Graph

A graph G is said to be complete if there exists exactly one edge between any pair of vertices in G. The complete graph with n vertices is denoted by K_n.



Round Robin Tournament

A game tournament where each team plays with each other team exactly once, is called round-Robin tournament. Such tournaments can be modelled using complete graph where there exist exactly one edge between each possible parts of vertices except itself. The following figure shows the tournament between 5 teams in Round-Robin fashion. Here, in the graph edge between vertices represent the matches between the teams and vertices represent teams.

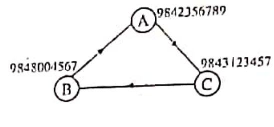


Collaboration Graph

A simple graph that can be used to model joint authorship of academic papers, where vertices represent people and edges between two vertices- represent joint authorship if they have jointly written a paper.

Call Graph

Graph can be used to model telephone calls in a telecommunication network. In such graph, each telephone number is represented by a vertex and each telephone call is represented by edge. The source vertex represent a start call (sender) and the vertex where edge ends represents end of call (receiver).



Regular Graph

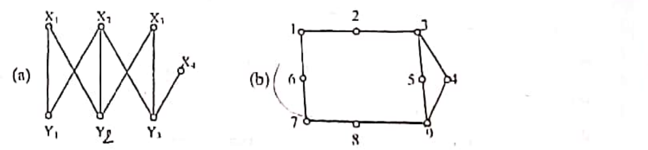
A graph in which all vertices are of same (equal) degree is called a regular graph. If the degree of each vertex is r then the graph is a r-regular graph. In fig. k_1, k_2, k_3, k_4, k_5, and k_6 are 0-regular, 1-regular, 2-regular, 3-regular, 4-regular and 5-regular respectively.

Bipartite Graph

A graph G = (V,E) is said to be bipartite if V is the union of two non-empty disjoint subsets V_1 and V_2 of V such that each edge in E is incident on one vertex in V_1 and one vertex in V_2. or

A graph G=(V,E) is said to be bipartite if its vertex set V can be partitioned into two subsets V_1 and V_2 such that each edge of G connect the vertex of V_1 to vertex of V_2 so that no edge in G connect two vertices in V_1 or two vertices in V_2.

The graphs shown in figure below are bipartite.



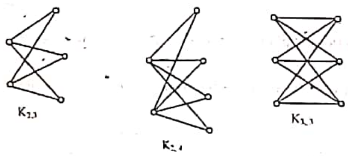
In fig (a), bipartite sets are $\{X_1, X_2, X_3, X_4\}$ and $\{Y_1, Y_2, Y_3\}$.

In fig (b), bipartite sets are $\{1, 3, 7, 9\}$ and $\{2, 4, 5, 6, 8\}$.

Complete Bipartite Graph

A bipartite graph G is said to be complete if each vertex of first bipartite set is connected to every vertices of another bipartite set. A complete bipartite graph with m number of vertices in first bipartite set and n number of vertices in second bipartite set is denoted by $K_{m,n}$

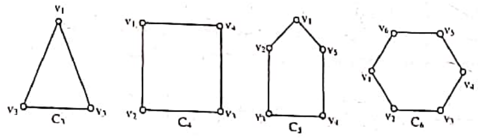
Scanned by CamScanner



Cycle Graph

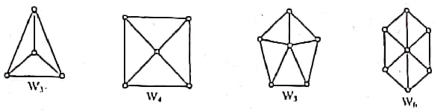
A Graph G=(V,E) with vertex $n \geq 3$in which every vertex is connected with two other vertices one on either side of it and last vertex is connected with first one to form a closed path, is called cycle graph

The cycle graph C_n of lengths n consists of n vertices $v_1, v_2, \ldots, v_n$ and edges $\{v_1, v_2\}, \{v_2, v_3\}, \{v_3, v_4\}, \ldots \{v_{n-1}, v_n\}$ and $\{v_n, v_1\}$.



Wheel Graph

The wheel Wn, for $n \geq 3$, is an union of Cn and additional vertex where the new vertex is connected by each vertex of the cycle. i.e. A graph which is obtained by adding an additional vertex to the cycle graph Cn for n>=3 and connect new vertex to each of n-vertices in Cn by new edge is called wheel graph.



n-cube graph

The n-dimensional cube, or n-cube denoted by Q_n, is the graph that has vertices representing the 2nd bit strings of length n. Here, two vertices are adjacent if and only if the bit string that they represent differ in exactly one bet-positions.

Example 14

What are Q_1, Q_2 and Q_3?

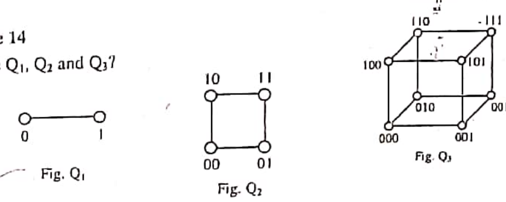


Fig. Q_1 Fig. Q_2 Fig. Q_3

Application of Graph on Local Area Network

In a local area network, there are many computer connected to each other, which form a special geometrical structure, called network topology. To model local area network in the form of topology, graph can be used-in-which computers or devices are represented by vertices and connections between these device are represented by edges.

For example:

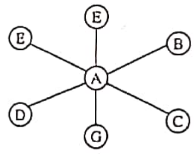


Fig: Stont network.

This topology can be modelled using complete bipontite graph $k_{1,6}$.

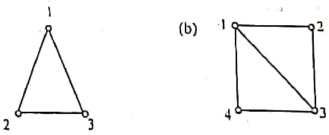
Platonic Graph

The graph formed by the vertices and edges of five regular (platonic) solids – the tetrahedron, octahedron, cube, dodecahedron and icosahedron are called the platonic graphs.

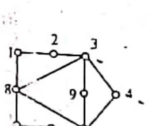
Example 15

Determine whether or not each of the following graphs is bipartite. In each case, give the partition sets or explain why the graph is not bipartite.

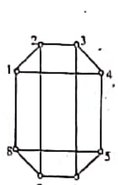
(a) (b)



(c)



(d)



Solution

(a) In this triangle, at least two of the three vertices must lie in one of the bipartite sets and these two are joined by an edge. Hence this graph can not be bipartite.

(b) The graph is not bipartite because it contains two triangles.

(c) The graph is bipartite with the partition sets M = {1, 3, 5, 7} and N = {2, 4, 6, 8, 9}.

(d) The graph is bipartite with the partition sets M = {1, 3, 5, 7} and N = {2, 4, 6, 8}.

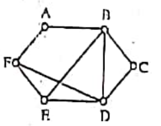
Subgraphs

Let G = (V, E) be a graph with vertex set V(G) and edge set E(G) and H = (V, E) be a graph with vertex set V(H) and edge set E(H).then H is said to be sub-graph of G written as $H \subseteq G$ if $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$ If H is a sub-graph of G, then

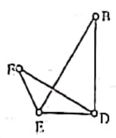
(i) All the vertices of H are in G.

(ii) All the edges of H are in G.

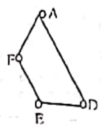
(iii) Each edge of H has the same end points in H as in G.



G_1



G_2



G_3

In above figures G_2 is a subgraph of G_1 but G_3 is not a subgraph of G_1.

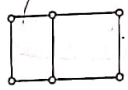
Spanning subgraph

The subgraph H of graph G is said to be spanning subgraph if V(H) = V(G). i.e. H and G has exactly same vertex set.

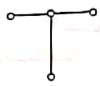
Induced sub-graph

If G is a graph with vertex set V and U is a subset of V, then the subgraph G(U) with the vertex set U and with edge set consisting of those edges of G that have both ends in U, is called the induced subgraph of G induced by the vertices in U.

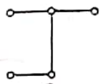
For example, consider three graphs which are shown in figure below as



G



G_1



G_2

Here G_1 is an induced subgraph of G but G_2 is not an induced subgraph of G.

Vertex deleted sub-graph

Let G=(V,E) be a graph and S be a non empty subset of V. The induced subgraph denoted by G-S is a subgraph obtained by deleting vertices in S. In this method, two form a subgraph:

1) Remove all vertices from V (G) which are in S.

2) Remove all the edges which are incident on vertices in S.

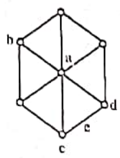
Edge deleted sub graph

If a subset F of E is deleted from G then G – F denotes the subgraph of G with vertex V and edge E – F, then G – F is called an edge deleted subgraph.

Example 16

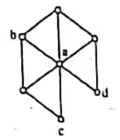
For the graph G shown below, draw the subgraphs

(a) G – e (b) G – a (c) G – b.

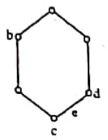


Solution

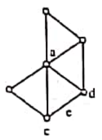
(a) After deleting the edge e = (c, d) from G, we get a subgraph G – e which is



(b) After deleting the vertex a from G and all the edges incident on it, we get a subgraph G – a which is



(c) The subgraph G – b is



Union and Intersection of Graphs

Union

Given two graphs G_1 and G_2, their union will be a graph such that

$$V(G_1 \cup G_2) = V(G_1) \cup V(G_2)$$

and $$E(G_1 \cup G_2) = E(G_1) \cup E(G_2)$$

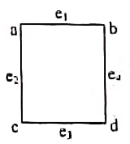
Intersections

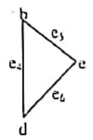
Given two graphs G_1 and G_2, their intersection will be a graph such that

$$V(G_1 \cap G_2) = V(G_1) \cap V(G_2)$$

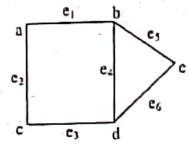
$$E(G_1 \cap G_2) = E(G_1) \cap E(G_2)$$

Consider two graphs G_1 and G_2 as

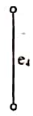




Then $G_1 \cup G_2$



and $G_1 \cap G_2$ is



Graph Connectivity

Walk

A walk in a graph G is a finite ordered set $W = (v_0, e_1, v_1, e_2, v_2, \ldots v_{k-1}, e_k, v_k)$ whose elements are alternately vertices and edges such that $1 \le i \le k$, the edge e_i has ends v_{i-1} and v_i.

This walk W is a $V_0 - V_K$ walk or walk from V_0 to V_K. The number of edges appearing in the sequence of the path is called its length. If the length of the walk is zero i.e. the walk has no edges, it contains only a single vertex and is called a trivial walk. A walk is closed if it starts and ends at the same point, otherwise the walk is open.

Trail

A walk W(U, V) in which all the edges are distinct, is called a trail.

Path

A walk in which all the vertices and edges are distinct, is called a path.

Circuit

A closed trail which contains at least three edges is called a circuit.

Cycle

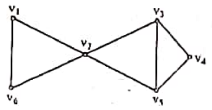
A circuit which does not repeat any vertices (except the initial and final vertex) is called a cycle. Thus, a cycle is a non-intersecting circuit and must have length three or more. A cycle of length K is called K-cycle. It should be noted that while every cycle is a circuit, the converse is not always true. A circuit may have repeated vertices other then the end vertices, but in a cycle the only repeated vertices are the first and last.

These definitions are summarized in the following table.

Term	Repeated edge	Repeated vertex
Path	No	No
Trail	No	Allowed
Walk	Allowed	Allowed
Circuit	No	Allowed
Cycle	No	First and last only

Example 17

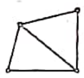
Consider the graph G in figure below, the closed trail ($v_1, v_2, v_3, v_4, v_5, v_2, v_6, v_1$) is a circuit but not a cycle, while the closed trail (v_2, v_3, v_4, v_5, v_2) is a cycle as well as a circuit.



Connected graph

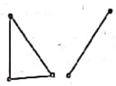
A graph G is said to be connected if there is a path between each possible pair of its vertices; otherwise G is disconnected.

(i)



Connected graph

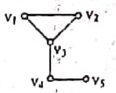
(ii)



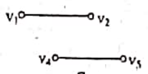
Disconnected graph

Cut-vertex

A vertex v of a connected graph G is called a cut-vertex (or cut-point) if G − v is disconnected. Any vertex in a graph is said to be cut vertex if graph becomes disconnected after removal of this vertex form the graph.



G = (V, E)

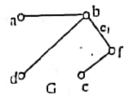


$G - v_3$

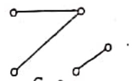
In above figure, after removable of vertex v_3, graph G becomes disconnected. So, v_3 is cut-vertex. Similarly, vertex v_4 also.

Cut-edge or bridge

Any edge in graph after whose removal graphs becomes disconnected, is called bridge or cut edge. An edge e is a bridge for G if G − e is disconnected.



G



G − e, where e = {b, f}

In above figure, after removable of an edge e = {b, f}, the connected graph G becomes disconnected. So, e = {b, f} is cut-edge.

Connected Component

A graph that is not connected is the union of two or more connected subgraphs, each pair of which has no vertex in common. These disjoint connected subgraphs are called the connected components of G. In other word, a connected component H of an undirected graph G is maximal connected subgraph. By 'maximal' we mean that G contains no other subgraph that is both connected and properly contains H.

Theorem 4:

There is a simple path between every pair of distinct vertices of a connected undirected graph.

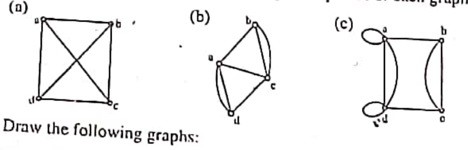
Proof:

Suppose a and b are two distinct vertices of the connected undirected graph G = (V, E). We know that G is connected so by definition there is at least one path between a and b. Let $x_0, x_1, \ldots, x_n$, where $x_0 = a$ and $x_n = b$, be the vertex sequence of a path of a least length. Now if this path of the least length is not simple then we have $x_i = x_j$, for some i and j with $0 \le i < j$. This implies that there is a path from a to b of shorter length with the vertex sequence $x_0, x_1, \ldots, x_i, \ldots, x_{j+1}, \ldots, x_n$ obtained by removing the edges corresponding to the vertex sequence $x_{i+1}, \ldots, x_j$. This shows that there is a simple path.

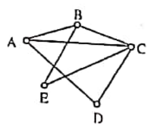
Scanned by CamScanner

EXERCISE (A)

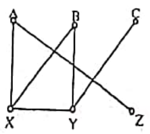
1. Considering the following graphs, determine
   (i) whether each of the graphs shown is a simple graph, a multigraph, a Pseudograph
   (ii) vertex set (iii) edge set
   (iv) degree of each vertex (v) degree sequence of each graph



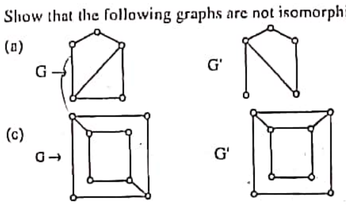
2. Draw the following graphs:
   (i) two 3-regular graphs with six vertices.
   (ii) two 3-regular graphs with eight vertices.
   (iii) if possible, a 3-regular graph with nine vertices. If it is not possible, explain why?
   (iv) the complete bipartite graph $K_{3,5}$.
3. Consider a graph given below. Find the degree of each vertex and verify the Handshaking theorem.



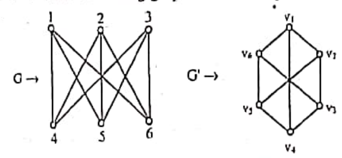
4. Draw a graph with five vertices v_1, v_2, v_3, v_4, v_5, such that deg $(v_1) = 3$, v_2 is an odd vertex, deg $(v_3) = 2$, and v_4 and v_5 are adjacent.
5. Consider the graph G in question no.3 and find
   (a) all simple paths from A to C.
   (b) all cycles
6. Consider the graph given below and find the following
   (a) Subgraph H of G induced by V(H) = {B, C, X, Y}
   (b) G – Y (c) all cut-vertices
   (d) all bridges



7. Suppose a graph has vertices of degree 0, 2, 2, 3 and 9. How many edges does the graph has?
8. Show that the following graphs are not isomorphic.



9. Show that the following graphs are isomorphic.



10. Draw a graph with given adjacency matrices

(a) $\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$ (b) $\begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$

(c) $\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$ (d) $\begin{bmatrix} 1 & 2 & 3 \\ 2 & 0 & 4 \\ 3 & 4 & 0 \end{bmatrix}$

(e) $\begin{bmatrix} 1 & 2 & 0 & 1 \\ 2 & 0 & 3 & 0 \\ 0 & 3 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$ (f) $\begin{bmatrix} 0 & 1 & 3 & 0 & 4 \\ 1 & 2 & 1 & 3 & 0 \\ 3 & 1 & 1 & 0 & 1 \\ 0 & 3 & 0 & 0 & 2 \\ 4 & 0 & 1 & 2 & 3 \end{bmatrix}$

(g) $\begin{bmatrix} 1 & 2 & 1 \\ 2 & 0 & 2 \\ 1 & 2 & 2 \end{bmatrix}$ (h) $\begin{bmatrix} 0 & 2 & 3 & 0 \\ 2 & 2 & 2 & 1 \\ 3 & 2 & 1 & 0 \\ 0 & 1 & 0 & 2 \end{bmatrix}$

11. Are the simple graphs with the following adjacency matrices isomorphic?

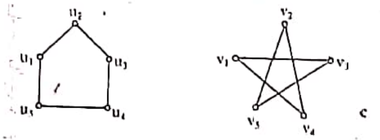
(a) $\begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$ (b) $\begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$

12. Determine whether the graphs with these incidence matrices are isomorphic.

$$\begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{bmatrix}.$$

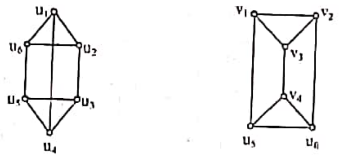
13. Determine whether the given pair of graph is isomorphic.

(i)

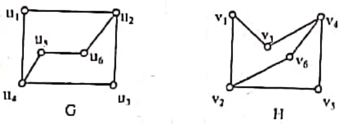


(ii)

(iii)

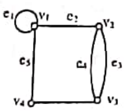


(iv)



G H

14. Find the incidence matrix and adjacency matrix of the graph given below.



15. Draw the graph G corresponding to the following adjacency matrix.

$$A = \begin{bmatrix} 1 & 1 & 1 & 2 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 2 \\ 2 & 0 & 2 & 2 \end{bmatrix}$$

Euler and Hamiltonian Graphs

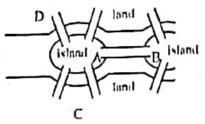
Eulerian Graphs

The development of the concept of the Eulerian graph is due to the solution of the famous Konigsberg bridge problem by the Swiss Mathematician Leonard Euler (1707 – 1783) in 1736.

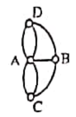
The Konigsberg Bridge Problem

The eighteen century city of Konigsberg included two islands and seven bridges crossing the Pregel river with land areas, denoted by A, B, C and D as shown in figure (a). It was asked whether it would be possible to walk around the city by crossing each of the bridges exactly once. Euler proved in 1736 that such a walk is impossible. He replaced the land areas and islands by points and the bridges by curves, as shown in figure (b).

(a)



(b)



Eulerian Trail

A trail in a graph G is called an Eulerial trail if it includes every edge of G and then G is called a traversable graph.

Eulerian Circuit

A circuit (closed trail) containing all the edges of a graph G is called an Eulerian circuit.

Eulerian Graph

A graph containing an Eulerian circuit is called an Eulerian graph or simply Eulerian.

We summarize the above definitions using a table as

Term	Initial and terminal vertices the same	Must include every edge	Repeated vertices allowed
Eulerian circuit	Yes	Yes	Yes
Eulerian Trail	No	Yes	Yes

The following two theorems give a criterion for determining which graphs and multi graphs are Eulerian.

Theorem

A connected graph (multi graph) G is Eulerian if and only if each vertex has even degree.

Proof:

Take a connected multigraph G =(V, E) where V and E are finite. We can prove the theorem in two parts. First we prove that if a connected multigraph has an Euler circuit, then all the vertices have even degree. For this, take a vertex v, where the Euler circuit begins. There is some edge that is incident to v and some other vertex say u then we have an edge {v, u}. This edge {v, u} contributes one to the degree of v and u both. Again there must be some edge other than {v,u} that is incident to u and some other vertex. In this case the total degree of the vertex u becomes even, so whenever in the circuit the vertex is met the degree of that vertex is even since every time entering and leaving the vertex gives even degree to all the vertices other than the initial vertex. However since the circuit must terminate in the vertex v and the edge that is terminating the circuit contributes one to the degree of the initial vertex v the total degree of the vertex v is also even. Now we have every time the vertex is entered and left it gives even degree and the initial vertex also gives even degree, we can conclude that if a graph has Euler circuit, then all the vertices have even degree.

Now we try to prove that if all the vertices in the connected multigraph have even degree, then there exist Euler circuit. For this, take a connected multigraph G with all the vertices having even degree. To make a circuit start at arbitrary vertex, say a of G. now start from the vertex $a = x_0$ and arbitrarily choose other vertex x1 to form and edge $\{x_0, x_1\}$. Continue building the simple path $\{x_0, x_1\}, \{x_1, x_2\}, \ldots, \{x_{n-1}, x_n\}$. This path terminates since it has a finite number of edges. It begins at a with an edge {a, x} and terminate at a with some edge {y, a}. This is correct since every vertex has even degree in the graph we are considering, if an edge left some vertex then there must be an edge entering that vertex to make its degree even. Now we have shown that there exists simple circuit in the graph with all the vertices of even degree. If this circuit has all the edges of the graph in it, then the simple circuit is itself an Euler circuit. If all the edges are not in the circuit, then we have next possibility. Now, consider the subgraph, say H that is formed by removing all the edges that are already in the simple circuit formed above and by removing the isolated vertices after edges are removed. Since the original graph G is connected, there must be at least one vertex of H that is common with the circuit we have formed. Let w be such a vertex. Every vertex in H has even degree since it is a subgraph of original graph. In case of w, while forming the circuit pairs of incident edges are used up. So the degree of w is again even. Beginning at w we can build a simple circuit as described above. We can continue this process until all edges have been used. Now if we combine the formed circuit in a way that it makes use of common

vertex to make a circuit then we can say that the circuit is an Euler circuit. Hence if every vertices of a connected graph has an even degree then it has an Euler circuit.

Theorem

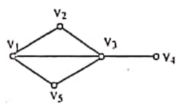
A connected graph G has an Eulerian trail if and only if it has exactly two odd vertices.

Proof:

This fact can be proved if we can prove that first, if the connected multigraph has Euler path exactly two vertices have odd degree and second if the connected multigraph has exactly two vertices of odd degree, then it has Euler path. Now, if the graph (in this proof graph means connected multigraph) has an Euler path say from a to z but not Euler circuit, then it must pass through every edge exactly once. In this scenario the first edge in the path contributes one to the degree of vertex a, and at all other time when other edges pass through vertex a it contributes twice to the degree of a, hence we can say that degree of a is odd. Similarly the last edge in the path coming to z contributes one to the degree of z, all the other edges contributes two one for entering and one for leaving. Here also the degree of last vertex, z is odd. All the other vertices other than a and z must have even degree since the edges in those vertices enter and leave the vertex contributing two to the degree every time the vertices are met. Hence if there is an Euler path but not an Euler circuit, exactly two vertices of the graph have odd degree. Secondly, if exactly two vertices of a graph have odd degree and lets consider they are a and z. Now, consider another graph that adds an edge {a, z} to the original graph, then the newly formed graph will have every vertices of even degree. So there exists Euler circuit in the new graph and the removal of the new edge gives us the Euler path in the original graph. Hence if exactly two vertices of the graph have an odd degree, then the graph has an Euler path but not Euler circuit.

Example 1

Show that the graph in figure contains an Eulerian trail but no Eulerian circuit.



Solution

Here,

G is connected and

$Deg(V_1) = 3, deg(V_3) = 4$

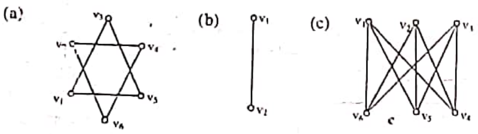
$Deg(V_2) = deg(V_5) = 2$

and $deg(V_4) = 1$

Since the graph G has exactly two odd vertices, it contains an Eulerian trail but no Eulerian circuit.

Example 2

Show that the graphs in figure contain no Eulerian circuit.



Solution

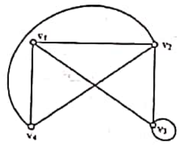
(a) It is not connected, so it does not contain an Eulerian circuit.

(b) It is connected but each degree is odd, so it does not contain an Eulerian circuit.

(c) It is connected but each degree is odd, so it does not contain an Eulerian circuit.

Example 3

Show the graph shown in figure has no Eulerian circuit but has an Eulerian trail.



Solution

Here,

G is connected and

$\deg(V_1) = \deg(A_4) = 3$ and

$\deg(V_2) = \deg(V_3) = 4$

Since it has exactly two odd vertices, it has an Eulerian trail but no Eulerian circuit. Where Eulerian trail can be described as $(V_1, V_2, V_4, V_1, V_3, V_3, V_2, V_4)$.

Hamiltonian Graphs

Hamiltonian graphs are named after sir Willian Hamilton, an Irish mathematician who introduced the problem of finding a circuit in which all the vertices of a graph appear exactly once.

A cycle that contains every vertex of a graph G exactly once is called Hamiltonian cycle (Hamiltonian circuit).

A graph G is Hamiltonian of it has a Hamiltonian cycle.

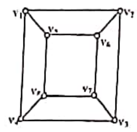
A Hamiltonian path is a simple path that contains all vertices of G.

Now, we summarize two basic concepts of this chapter in the following table:

Term	Initial and terminal vertices the same	Must include every edge	Repeated vertices followed
Eulerian circuit	Yes	Yes	Yes
Hamiltonian cycle	Yes	No	No

Example 4

Is the graph (cube) given in figure is Hamiltonian?



Q3

Solution

The graph in figure is Hamiltonian because it contains a cycle covering all the vertices, is $(v_5, v_1, v_2, v_6, v_7, v_3, v_4, v_8, v_5)$.

Theorem (without proof)

A simple connected graph with $n \geq 2$ vertices is Hamiltonian if degree of every vertex is at least n/2.

(This theorem is known as Dirac's Theorem)

Theorem (without proof)

A connected graph with n vertices is Hamiltonian if for any two non-adjacent vertices u and v, $\deg(u) + \deg(v) \geq n$.

(This theorem is known as Ore's Theorem).

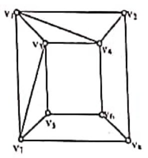
It may be noted that the theorems stated above provide only sufficient criterion. There are no known simple necessary and sufficient criteria for the existence of Hamiltonian cycle nor is there even an efficient algorithm. For finding such a cycle many theorems exist that establish either necessary or sufficient conditions for a connected graph to have Hamiltonian cycle or path.

A few helpful hints for trying to find a Hamiltonian cycle in a graph are given below.

1. If G has a Hamiltonian cycle, then for all $u \in V$, $deg(u) \geq 2$.
2. If $v \in V$ and deg $(v) = 2$ then two edges incident with vertex v must appear in every Hamiltonian circuit for G.
3. If $v \in V$ and deg $(v) > 2$, then we try to build a Hamiltonian cycle, once we pass through vertex v, any unused edges incident with v are deleted from further consideration.
4. In building a Hamiltonian circuit for G, we cannot obtain a circuit for a subgraph of G unless it contains all the vertices of G.

Example 5

Determine, in graph of figure, if there is an Eulerian circuit and/or a Hamiltonian cycle.



Solution

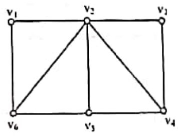
Clearly there is no Eulerian circuit since there are some odd vertices, but the graph has Hamiltonian cycle $(v_1, v_2, v_4, v_6, v_8, v_7, v_5, v_3, v_1)$.

Example 6

Give an example of a graph with six vertices which is Hamiltonian but not Eulerian and vice-versa.

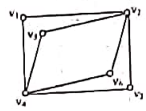
Solution

Here, we need to be very clever since for a graph to not be Eulerian means that not all the vertices are of even degree. So we try to build some odd degrees in that graph as shown in figure.



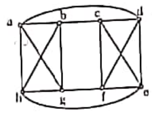
The graph above contains a Hamiltonian circuit $(v_1, v_2, v_3, v_4, v_5, v_6, v_1)$ but no Eulerian circuit.

The graph shown in figure is Eulerian but not Hamiltonian.



Example 7

Is the graph in figure is Hamiltonian?



Solution

The graph shown in figure above is Hamiltonian (by Dirac's theorem) because it has 8 vertices such that each vertex has degree $\frac{8}{2} = 4$. So the Hamiltonian cycle (circuit) is (a, b, g, h, e, f, c, d, a).

Weighted Graph or Labelled Graph

A weighted graph is a graph G, in which each edge, e, is assigned a non-negative real number, w(e), called the weight of e. The weight of a subgraph H of G is the sum of the weights of the edges of the subgraph H.

Now we discuss two famous network problems-the shortest path problem and the Chinese postman problem.

The Shortest Path Algorithm

Consider a weighted graph G. The length of a path in a weighted graph is the sum of the weights of the edges of this path and the shortest path between the two vertices is the minimum length of the path. There are several different algorithms to find the shortest path between two vertices in a weighted graph. We discuss here one discovered by E.W. Dijkstra.

Dijkstra's Algorithm

Step 1 : Label the initial vertex of the graph with weight zero.

Step 2 : Calculate the weights of all vertices adjacent to the initial vertex corresponding to the weights of the edges incident on the initial vertex.

Step 3 : Label these vertices with smallest possible value of their weights.

Step 4 : Calculate the weights of all those vertices which are adjacent to the vertices with minimum weights determined in step 3.

Step 5 : Label these vertices with minimum weight.

Step 6 : Continue this process until all the vertices of weighted graph are labeled.

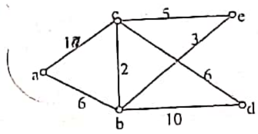
Step 7 : Trace the path of cumulative minimum weight from the initial vertex to desire vertex.

Pseudo code for Dijkstra's algorithm

```
DijkstraSP(G: Weighted connected simple graph with all weights positive)
{
    for i=1 to n
        L(vi)=∞
    L(a)=0
    S=φ //the labels are now initialized so that the label of a is 0
            and all other labels are ∞ and S is empty set)
    while Z ∉ S
    begin
        u=a vertex not in S with L(u) minimal
        S=S∪{u}
        for all vertices v not in S
        if L(u)+w(u,v)<L(v) then L(v)=L(u)+w(u,v)
    // this ads a vertex     in S with minimal label and updates the label of
       vertices not in S
    end // L(z)=length of a shortest path from a to z.
}
```

Example 8

Apply Dijkstra's algorithm to find the shortest path from the vertex a to each of the other vertices of the following weighted graph in figure.



Solution

Since we have to find the shortest path from the vertex a to each of the other vertices, then we assign weight 0 to vertex a and ∞ to all remaining vertices.

Vertex	(a)	b	c	d	e
Label	0	∞	∞	∞	∞

Now, The vertices adjacent a are b & c, we calculate weights of b & c and label them with minimum weight.

wt (b) = wt (a) + wt (a,b), = 0 + 6 = 6

wt (c) = wt (a) + wt (a,c) = 0 + 1 = 1

Since, c has minimum weight so, we select it

Vertex	(a)	b	c	d	e
Label	0	6	1	∞	∞

Now the vertices adjacent to c are b, e & d

wt (b) = wt (c) + wt (c,b) = 1+2 = 3

wt (e) = wt (c) + wt(c,e) = 1+5 = 6

wt (d) = wt (c) + wt(c,d) = 1+6 = 7

Since, b has smallest weight we select it & replace previous wt of b by new weight.

Vertex	(a)	(b)	(c)	d	e
Label	0	3	1	7	6

Now, the vertices adjacent 'b' are 'e' and 'd'

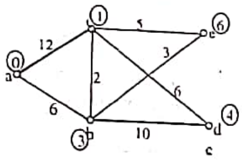
So,

wt (e) = wt (b) + wt(b,e) = 3+3 = 6

wt (d) = wt(b) + wt(b,d) = 3+1 = 4

Since, 'd' has minimum weight so, we select 'd' and replace it's pervious weight by new.

Vertex	(a)	(b)	(c)	(d)	(e)
Label	0	3	1	4	6



∴ The shortest path from a to c is a — c length = 1

The shortest path from a to b is :

a — c — b, length = 3

The length of shortest path from a to d is :

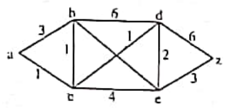
a — c — b — d length = 4

The length of shortest path from a to e is :

a — c — e length = 6

Example 9

Find the shortest path between a to z in the given weighted graphs.



Solution

Since we have to find the shortest path from vertex a to z. So, we assign weight 0 to a and ∞ to all remaining vertices.

Vertex	(a)	b	c	d	e	z
Label	0	∞	∞	∞	∞	∞

The vertices adjacent to a are b and c, we calculate weights of b and c and label them with minimum weight.

Wt(b) = min(wt(b), wt(a) + wt(a, b))

= min(∞, 0 + 3) = min(∞, 3) = 3

Wt(c) = min (wt(c), wt(a) + w + (a, c))

= min (∞, 0 + 1) = min (∞, 1) = 1

Since c has minimum weight so we select it

Vertex	(a)	b	(c)	d	e	z
Label	0	3	1{a, c}	∞	∞	∞

The vertices adjacent to c are b, d, and e

∴ wt(b) = min(wt (b), wt(c) + wt(c, b)

= min (3, 1 + 1) = min (3, 2) = 2

wt(d) = min (wt(d), wt(c) + t(c, d))

= min(∞, 1 + 1) = min(∞, 2) = 2

wt(e) = min(wt(e), wt(c) + wt(c, e)

= min(∞, 1 + 4) = min(∞, 5) = 5

Since b and d has equal wt so we randomly select b.

Vertex	(a)	(b)	(c)	d	e	z
Label	0	2{a,c,b}	1{a, c}	2	5	∞

The vertex adj. to b still unmarked is d.

So, wt(d) = min(wt(d), wt(b) + wt(b, d))

= min(2, 2 + 6) = min(2, 8) = 2

Now we select d

Vertex	(a)	(b)	(c)	(d)	e	z
Label	0	2{a,c,b}	1{a, c}	2{a,c,d}	5	∞

The vertices adj. to d still unmarked are: e & z

∴ wt(e) = min(wt(e), wt(d) + wt(d, e)

= min(5, 2 + 2) = min (5, 4) = 4

wt(z) = min (wt(z), wt(d) + wt(d, z))

= min (∞, 2 + 6) = min(∞, 8) = 8

Since e has minimum weight so we select it.

Vertex	(a)	(b)	(c)	(d)	(e)	z
Label	0	2(a,c,b)	1{a, c}	2{a,c,d}	4{a,c,d,e}	8

The vertex adjacent to e is z

So, wt(z) = min(wt(z), wt(e) + wt(e, z))

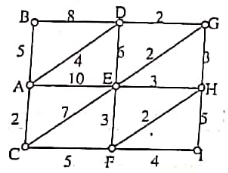
= min(8, 4 + 3) = min(8, 7) = 7

Vertex	(a)	(b)	(c)	(d)	(e)	(z)
Label	0	2{a,c,b}	1{a,c}	2{a,c,d}	4{a,c,d,e}	7{a,c,d,e,z}

∴ Shortest path from a to z is a - c - d - e - z and weight is 7.

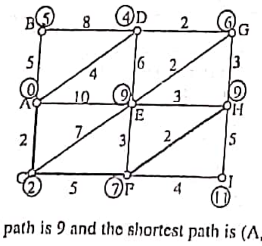
Example 10

Apply Dijkstra's Algorithm to find the shortest path from the vertex A to the vertex H in figure.



Solution

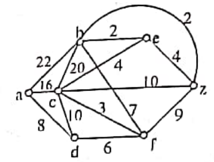
In the graph of figure, we have to find the shortest path from the vertex A to H, so we label initial vertex A with weight zero. Then we calculate the weights of all vertices adjacent to the vertex A and label them with minimum weight. We continue this process until all the vertices of the graph are labelled with minimum weight and then we trace the shortest path of cumulative minimum weight from the vertex A to H, which is shown in figure below.



The length of the shortest path is 9 and the shortest path is (A, C, F, H).

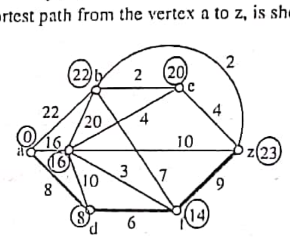
Example 11

Determine the shortest path from the vertex a to z in the graph of figure below by using Dijkstra's algorithm.



Solution

To find the shortest path from the vertex a to z, we label the vertex a with zero. Then we calculate the weights of all adjacent vertices to the vertex a and label them with minimum weight. We continue this process until all the vertices are labeled with minimum weight. Then we find the shortest path from the vertex a to z, is shown in figure below.



Example 12

The complete graph K_n is Hamiltonian for all n > 2.

Solution

Dirac's theorem tells us that a connected graph with n(>2) vertices is Hamiltonian if d(v) ≥ $\frac{n}{2}$, for every vertex in G.

Since K_n is a complete graph with n(> 2) vertices and d(v) = n – 1, for every vertex in K_n. Therefore to show K_n is Hamiltonian, we show

$d(v) \geq \frac{n}{2}$, for every vertex $v \in K_n$.

Since n > 2 (given)

or $\frac{n}{2} > 1$

or $\frac{n}{2} - 1 > 0$

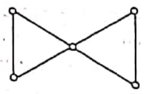
or $n - \frac{n}{2} - 1 > 0$

or $n - 1 > \frac{n}{2}$

Thus, K_n is Hamiltonian.

Example 13

Is the graph given below Eulerian? Is it Hamiltonian?



Solution

Yes, the graph is Eulerian but not Hamiltonian.

Traveling Sales Man Problem (TSP)

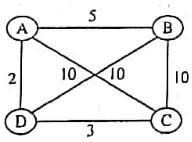
This problem was first formulated in 1930 and is one of the most intensively studied problem in graph theory and optimization. The general - TSP problem can be formulated as follows:

given a list of cities (represented by vertex of graph) and distance between each pair of cities. the problem is to find out the shortest possible route or path that usual each city exactly once and returns to the origin or starting point (city).

The name "traveling salesman problem" is given as it is analogous to the problem like a salesman want to visit each of n-city exactly once to sale his business and finally returns to his horse at the end of journey.

Example 14

Let a travelling salesman want to visit places. A, B, C, D, as shown in figure below. Now the problem is: in which order should be visit these places to travel minimum total distance.



To solve this problem, let us assume that salesman start as A then we can examine all possible way for him to visit other places and then return to A.

Route	Total distance
A → B → C → D → A	5 + 10 + 3 + 2 = 20
A → C → B → D → A	10 + 10 + 10 + 2 = 32
A → D → B → C → A	2 + 10 + 10 + 10 = 32

From the table above, the minimum distance a travel salesman has to travel to cover all the fan cities and return to his home 'A' is equals to 20.

The most straightforward solution to TSP is to examine all possible Hamilton circuit and select the circuit with minimum length. Now, the issue is how many such circuit exist with vertex 'n'?

There are (n - 1)! such different hamilton circuit as there are (n - 1) choice for selection first vertex (n - 2) for second and so on.

Since hamilton circuit can be travelled in reverse order, we only need to examine (n - 1)!/2 Hamilton circuit.

For example,

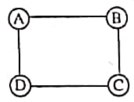
In the previous figure, there are four vertices. So n = 4 and the total possible route = (n - 1)!/2 = (4 - 1)!/2 = 6/2 = 3.

The complexity (n - 1)!/2 grows rapidly it is impossible to solve TSP when n is large. For example, when n = 25 then total number of possible path 24!/2 which is approximately 3.1×10^{23} = path. It will take approximately many million year to find a minimum length path.

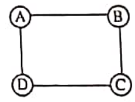
Planer Graph

A graph G = (V, E) is said to be planer graph if it can be drawn in plane such that no-intersection of edges exist at a point other then their common end point i.e. a graph is planer if it is drawn without crossing the edge.

Examples 15

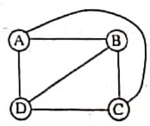


Planer graph



Planer graph

The planner representation of k_4 is



Planer k_4

The cube graph Q_3 is -

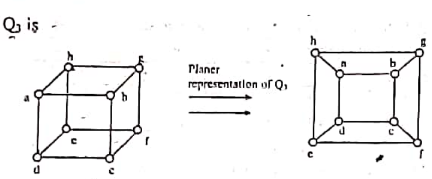


Fig. Q_3

Euler's Formula

Theorem:

Let G = (V, E) be a connected planer simple graph with V vertices, e edges and r be number of regions in planar representation then $r = e - V + 2$.

Example 16

Suppose that a connected planer simple graph has 20 vertices, each of degree 4. Into how many regions does a representation of this planner graph split the plane?

Solution

The graph has 20 vertices, each of degree 4.

Therefore, V = 20

Sum of degree of vertices = 4×20 = 80

Since, sum of degree of vertices is equal to twice the number of edges.

$\therefore \quad 80 = 2\times e$

or, $\quad e = \frac{80}{2} = 40$

Now,

Using Euler's formula

$r = e - V + 2$

$= 40 - 20 + 2$

$= 22$

Therefore, number of region is 22.

Example 17

Suppose that a connected planner graph has 30 edge. If a planner representation of this graph divides the plane into 20 regions, how many vertices does this graph have?

Solution

Here, the number of edge is the graph e = 30

The number of region in the planer graph r = 20

Now, we know that the euler's theorem state the relation

$r = e - v + 2$

$v = e - r + 2$

$= 30 - 20 + 2$

$v = 12$

$\therefore$ The total number of vertices (v) = 12.

Example 18

Suppose that a connected planar graph has 30 edges. If a planar representation of this graph divides the plane into 20 regions, how many vertices does this graph have?

Solution

We have, r = 20, e = 30, so by Euler's formula we have $v = e - r + 2 = 30 - 20 + 2 = 12$

So the number of vertices is 12.

Corollary 1: If G is a connected planar simple graph with e edges and v vertices where $v \geq 3$, then $e \leq 3v - 6$.

Corollary 2: If G is a connected planar simple graph, then G has a vertex of degree not exceeding five.

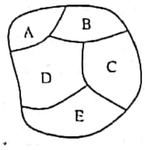
Corollary 3: If a connected planar simple graph has e edges and v vertices with $v \geq 3$ and no circuits of length three, then $e \leq 2v - 4$.

Graph Coloring

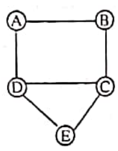
The map of the world consist of many region. To designers one region with its adjacent another region, different colors are used. One way to ensure that two adjacent regions never have the same color is to use a different on maps with many regions it would be hard to distinguish similar colors. Instead, a small number of color should be used whenever possible. For this, we some method to determine least no. of color for coloring map.

Each map in the plane can be represented by a graph. Each region of the map is represented by a vertex. Edge connects two vertices if the region represented by vertices have a common order. The graph made in this way is known as dual graph.

A coloring of a simple graph is the assignment of a color to each vertex of the graph so that no two adjacent vertices are assigned the same color.



Map

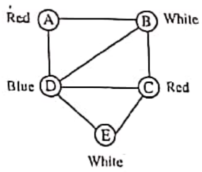


Equivalent graph of map.

A graph can be coloured by assigning different color to each vertex but it is inefficient of the number of vertex one very large. We can color graph with less number of color than number of vertex.

The least number of colors needed for coloring a graph is known as chromatic number.

Example 19

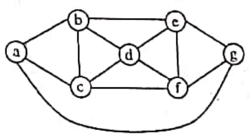


Here, chromatic number of this graph = 3.

Theorem: (Four color Theorem): The chromatic number of a planner graph is no greater than four.

Example 20

What is the chromatic no. of given graph?



Solution

Here, let us choose first vertex 'a' and colour / with 'red' i.e. col(a) = Red.

Now, adjacent vertex of 'a' can't have red color so we need different color for vertex 'b'. So let

col (b) = blue

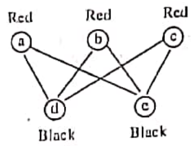
Example 21

What is the chromatic number of $k_{m,n}$?

Solution

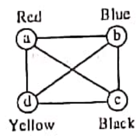
In complete bipatrate graph $k_{m,n}$, every verties of first position set are adjacent to each 'n' vertices of another - partition set. Each vertices of first position set can be colored with single color as the vertices in the same position are upon adjacent to each other. Similarly, all vertices of second position can be colored with another single color. Hence, chromatic number of $k_{m,n}$ is equals to 2.

i.e. $|k_{m,n}| = 2$



Here, chromatic number = 2,

Fig: $k_{3,2}$



Here, chromatic number = 4

Fig.: k_4

Since 'c' is adjacent vertex of both a and b, so it requires different color then a and b. Let

col (c) = black

Next let us choose vertex 'd' since 'd' is adjacent to previous (processed) vertices b and c but non-adjacent to a. It need different color then the color of vertex b and c but similarly to a. Therefore,

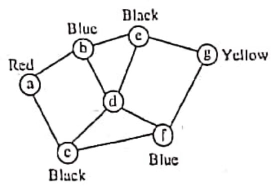
col (d) = red

Similarly, we can process other vertices and get the following graph with color.

col (e) = black

col (f) = blue

col (g) = yellow



Graph with color

Example 22

What is the chromatic number of k_n?

Solution

In complete graph k_n, there exist an edge between each possible pair of vertices, so no two vertices can be assigned the same color. Hence, chromatic number of $k_n = n$ (each vertex has different color).

Application of graph coloring

The problems involving scheduling assignments of resour, can be solved by using concept of graph coloring.

(a) Scheduling Final Exam

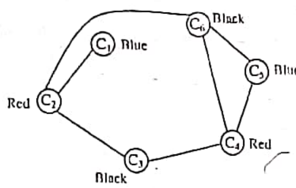
Problem:

How can the final exams at a university be scheduled so that no students has two exam at the same time?

Solution

This problem can be solved using a graph model in which comses represents vertices and there is an edge between two vertices if a common students in the course represented by these vertices. Each time slot for final exam is represented by different color.

For example, let there are six finals course to be scheduled and course are numbered from C_1 to C_6 suppose that the course number C_1 & C_2. C_1 and C_3, C_2 and C_3, C_3 and C_4, C_4 and C_5, C_4 and C_6, C_5 and C_6, C_2 and C_6 has common students for exam. Then the graph becomes



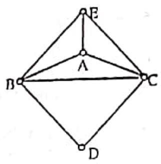
Now, scheduling of exam is determined by coloring this graph.

Therefore, the exam schedule is determined with save color as time period with respective course.

Time period	Course
I	C_1, C_5 (Blue color)
II	C_3, C_6 (Black color)
III	C_2, C_4 (Red color)

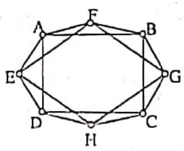
EXERCISE – (B)

1. Test each of the following graphs for Eulerian circuits and Hamiltonian cycles
   (i) K_4 (ii) K_5 (iii) $K_{3,5}$
2. From the graph in fig below, state that
   (i) Is the graph Hamiltonian?
   (ii) Is there a Hamiltonian path?
   (iii) Is it Eulerian?
   (iv) Is there an Eulerian trail?

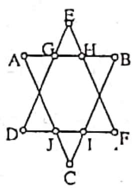


3. In each case, explain why the graph is Eulerian and find an Eulerian circuit.

(i)

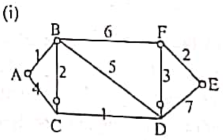


(ii)

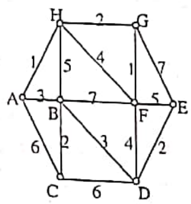


4. (a) How many edges must a Hamiltonian cycle in K_n contain?
   (b) How many Hamiltonian cycles does K_n have?
5. Use Dijkstra's algorithm to find the shortest path from A to E in the following weighted graphs.

(i)

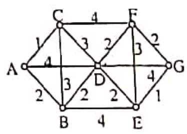


(ii)

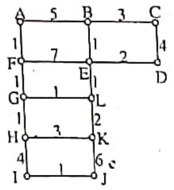


6. Solve the shortest path problem for following graphs (Choose source and destination vertex as your choice).

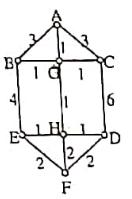
(i)



(ii)



(iii)



7. Suppose that a connected planner simple graph has 20 verties, each of degree 3. Into how may regions does a representation of this planar split a plane?

8. What is Euler's formula for planar graph? How can Euler's formula for planar graph can be used to show that simple graph is non-planar.

9. Which of the undirected graph in the following figure have an Euler circuit? Explain.

10. What is chromatic number of $k_{m,n}$ and k_n.

Directed Graph

The theory of graphs studied so far is concerned with undirected graphs. Most of the concepts and terminology of undirected graphs are similar to directed graphs. Such graphs are frequently more useful in various dynamical systems such as digital computers or flow systems.

This section gives the basic definitions and properties of directed graphs. Many of the definitions will be similar to those in the preceding section on undirected graphs.

A directed graph or digraph D = (V, E) consists of two things:

(i) A set V = V(D) whose elements are called vertices, points or nodes of D.

(ii) A set E = E(D) of ordered pairs (u, v) of vertices called arcs or directed edges or simply edges.

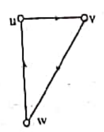
A picture of directed graph D is the representation of D in the plane. That is, each vertex v of D is represented by a dot or small circle and each directed edge e = (u, v) is represented by an arrow or directed curve from the initial vertex u to terminal vertex v.

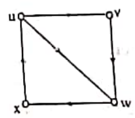
If edges and /or vertices of a directed graph D are labeled with some type of data, then D is called a labeled directed graph.

A directed graph D=(V, E) is said to be finite if its set V of vertices and its set E of edges are finite.

Simple digraphs

A directed graph D=(V, E) which has neither loops nor multiple edges is called simple directed graph

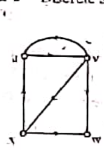




Multiple digraphs

In any diagraph, two or more edges are said to be parallel if they have same initial and terminal vertex with same orientation.

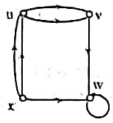
A digraph D=(V,E) which contains some multiple edges is called a multiple directed graph.



Pseudo digraph

In any digraph, an arc or edge with same initial and terminal vertex, is called self loop.

A digraph D = (V, E) which contains self loop is called a pseudo digraph.



Subgraphs

Let H be a directed graph with vertex set V(H) and edge set E(H). Similarly let G be a directed graph with vertex set V(G) and edge set E(G). Then H is said to be subgraph of G, written as $H \subseteq G$, if $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$.

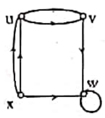
Adjacent Vertices

The vertices u and v are said to be adjacent vertices if there is a directed edge e = (u, v) or e = (v, u). Note that the edges (u, v) and (v, u) are different in a directed graph.

The edge e is said to be incident on each of its end points u and v.

In degree and out degree of vertex

Suppose D = (V, E) is a directed graph. The in-degree of v, written indeg(v), is the number of edges incident towards v and the out degree of a vertex v of D, written as outdeg(v), is the number of edges directed away from v.

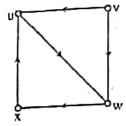


Here,

Vertex	u	v	w	x
In degree	2	2	3	0
Out degree	2	1	1	3

Source and Sink vertex

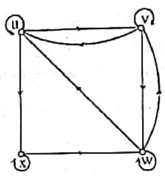
In a digraph D = (V, E), a vertex v with zero indegree is called a source vertex, and a vertex v with zero outdegree is called a sink vertex. For example, we consider a directed graph D of four vertices u, v, w and x in figure below.



In figure, the vertex v has indegree 0, so it is a source and there are no vertices with outdegree zero, so there are no sinks. The indegree sequence of the directed graph D in figure is (2, 2, 1, 0) and outdegree sequence of D is (2, 1, 1, 1).

Example 1

Determine whether the directed graph D in figure is a reflexive, symmetric and transitive digraph.



Solution

Here in directed graph D,

V(D) = {u, v, w, x} and

E(D) = {(u, u), (u, v), (v, u), (v, v), (v, w), (w, v), (w, w), (w, u), (x, w), (x, x), (u, x)}

Digraph D is reflexive since for all vertex $a \in V(D)$, we have $(a, a) \in D$.

Digraph D is not symmetric since there is no directed edge (a, b) for all directed edge (b, a) in D. Digraph D is not transitive since for (a, b) and (b, c) in D, there is no directed edge (a, c) in D. for example, (u, v) and (v, w) does not imply (u, w) in D.

Paths

Let D be a directed graph. The concepts of paths, simple paths, trials and cycles carry over from non-directed graphs G except that the directions of the edges must agree with the direction of the path. Specifically:

(i) A (directed) path in D is an alternating sequence of vertices and directed edges, say,

$P = (v_0, e_1, v_1, e_2, v_2, \ldots, v_n)$

such that each edge e_i begins at v_{i-1} and ends at v_i.

(ii) The length of the path P is n, its number of edges.

(iii) A simple path is a path with distinct vertices.

(iv) A trial is a path with distinct edges.

(v) A closed path has the same first and last vertices.

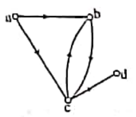
(vi) A spanning path contains all the vertices of D.

(vii) A cycle is a closed path with distinct vertices except first and last.

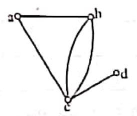
(viii) A circuit is a closed path containing at least three edges.

Underlying Graph

Given a digraph D, we can obtain a graph G from D by removing all the arrows or directions from the edges. If the graph G has the same vertex set as that of the digraph D and corresponding to each directed edge in D associated with the ordered pair of vertices (a, b), there is an edge in G associated with the pair (a, b) then G is called the underlying graph of D. For example, figure (b) is the underlying graph of directed graph D in figure (a).



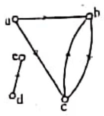
(a)



(b)

Weakly Connected

A digraph D is said to be weakly connected if its underlying graph is connected. For example, the digraph D in figure is weakly connected but the digraph D in figure is not weakly connected.



Strongly Connected

A directed graph or digraph D is said to be strongly connected if for any pair of vertices v_i and v_j in D, there is a directed path form v_i to v_j as well as a directed path from v_j to v_i.

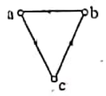
In case, if there is a directed path from v_i to v_j or from v_j to v_i (not necessarily both) for any pair of vertices v_i and v_j, then D is said to be unilaterally connected.

Clearly, a digraph which is strongly connected is also unilaterally connected.

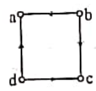
Example 2

Determine which of the digraphs in figure is strongly connected and unilaterally connected?

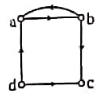
(i)



(ii)



(iii)



Solution

The digraph D in figure (i) is strongly connected, so it is unilaterally connected also.

The digraph D in figure (ii) is not strongly connected since there is no path from a to b. This digraph D is not unilaterally connected as well, since there is no directed path from b to d or from the vertex d to b.

The digraph D in figure (iii) is not strongly connected since there is not a directed path from the vertex c to b but this digraph D is unilaterally connected since there is a directed path between any two vertices.

Theorem

The sum of the outdegrees of vertices, the sum of indegrees of vertices and the number of edges in a directed graph are equal to each other.

OR,

If D is a directed graph with vertices $v_1, v_2, \ldots, v_n$ and q is the number of directed edges in D, then

$$\sum_{i=1}^{n} \text{in-deg}(v_i) = \sum_{i=1}^{n} \text{out-deg}(v_i) = q$$

Proof: When the in-degrees of the vertices are summed, each edge is counted exactly once since every edge goes to exactly one vertex. Thus

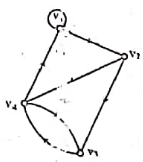
$$\sum_{i=1}^{n} \text{in-deg}(v_i) = q$$

Similarly, when the outdegrees are summed, each edge is counted exactly once every edge goes out of exactly one vertex. Then

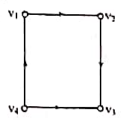
$$\sum_{i=1}^{n} \text{out-deg}(v_i) = q$$

Thus, $\sum_{i=1}^{n} \text{in-deg}(v_i) = \sum_{i=1}^{n} \text{out-deg}(v_i) = q$

(i)

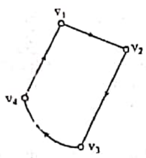


(ii)

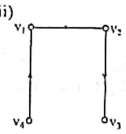


In figure (i), we can clearly get a closed spanning path, which is shown in figure, so it is strongly connected. Since (i) is strongly connected it is unilaterally connected as well. The digraph (ii) has no closed spanning paths, hence it is not strongly connected but it has a spanning path (v_4, v_1, v_2, v_3), so it is unilaterally connected. Figure shows this spanning path.

(i)



(ii)



Representation of Digraph

(i) Adjacency Matrix:

Let D = (V, E) be a digraph with vertices $v_1, v_2, \ldots, v_n$. Then the adjacency matrix of D with respect to given order of vertices is n×n matrix A(D) = $[a_{ij}]_{n\times n}$ is defined as follows:

$$a_{ij} = \begin{cases} 1 \text{ if there is an edge } (v_i, v_j) \\ 0 \text{ if there is an edge } (v_i, v_j) \\ k \text{ if there are k number of arc from } v_i \text{ to } v_j \end{cases}$$

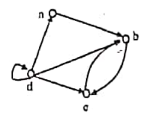
(ii) Incidence Matrix:

Let D =(V,E) be a digraph with vertices $v_1, v_2, \ldots, v_m$ and directed edges $e_1, e_2, \ldots, e_n$. Then incidence matrix of D, $[m_{ij}]_{m\times n}$, is defined as follows:

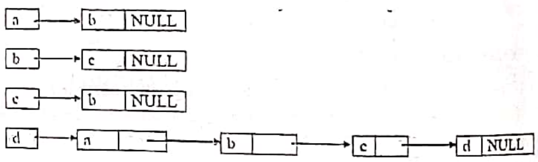
$$m_{ij} = \begin{cases} 1 \text{ of edge } e_j \text{ is directed away from a vertex } v_i \\ -1 \text{ if edge } e_j \text{ is directed towards vertex } v_i \\ 0 \text{ otherwise} \end{cases}$$

(iii) Adjacency List (Linked list)

Edge List for Simple Graph	
Vertex	Adjacent Vertices
a	b
b	c
c	b
d	a, b, c, d

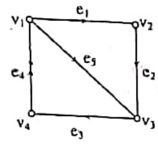


And for each node a linked list of its adjacent nodes is given as;



Example 3

Find the incidence matrix to represent the graph shown in figure



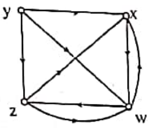
Solution

The incidence matrix of the graph of figure is

$$[m_{ij}] = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \begin{array}{c} \begin{array}{ccccc} e_1 & e_2 & e_3 & e_4 & e_5 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & -1 & 1 \\ -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & -1 \\ 0 & 0 & -1 & 1 & 0 \end{bmatrix}_{4 \times 5} \end{array}$$

Example 4

Find the adjacency matrix to represent the directed graph shown in figure, where vertices are ordered as $v_1 = x$, $v_2 = y$, $v_3 = z$ and $v_4 = w$.



Solution

The adjacency matrix of the directed graph of fig. 6.64 is

$$A = \begin{array}{c} \\ x \\ y \\ z \\ w \end{array} \begin{array}{c} \begin{array}{cccc} x & y & z & w \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} \end{array}$$

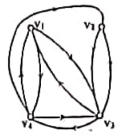
Example 5

Draw the digraph D corresponding to the adjacency matrix A, where A is given as,

$$A = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

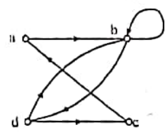
Solution

Since the given matrix is a square matrix of ordered 4, the digraph D has 4 vertices, say v_1, v_2, v_3 and v_4. Draw a directed edge from v_i to v_j where $a_{ij} = 1$. The required digraph D is shown in figure below.

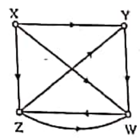


EXERCISE (C)

1. Draw the diagraph D given that $V = \{v_1, v_2, v_3, v_4, v_5\}$ and $E = \{(v_1, v_2), (v_1, v_3), (v_2, v_3), (v_2, v_4), (v_2, v_4), (v_3, v_3), (v_4, v_2)\}$
2. Find the relation determined by the given diagraph.



3. Draw the diagraph D of the relation R on $V = \{1, 2, 3, 4\}$ defined by $R = \{(1, 2), (2, 2), (2, 4), (3, 2), (3, 4), (4, 1), (4, 3)\}$.
4. Consider a directed graph D, which is given below
   (a) Find all simple paths from X to Z.
   (b) Find all cycles in D.
   (c) Find all the indegrees and outdegrees of each vertex of D.
   (d) Are there any sources or sinks?
   (e) Find the subgraphs H of D determined by the vertex set $V(H) = \{X, Y, Z\}$.
   (f) Is D unilaterally connected? Strongly connected?

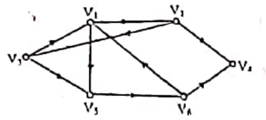


5. Consider the directed graph D in following figure.

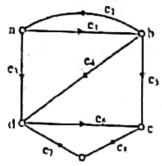
   (a) Find the simple paths from V_1 to V_6.

   (b) Find four cycles in D which includes V_3.

   (c) Is D unilaterally connected? Strongly connected?



6. Find the incidence matrix and adjacency matrix of the diagraph given below.



6

Chapter

Trees

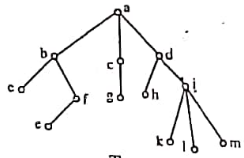
Introduction

The word tree suggests branching out from a root and never completing a cycle. Trees form one of the most widely used subclasses of graphs. This is due to the fact that many of the applications of graph theory, directly or indirectly, involves trees. A tree occurs in situations where many elements are to be organized into some short of hierarchy. In computer science, trees are useful in organizing and storing data in a database, used to construct efficient algorithm for locating items in a list. They can also be used in algorithms, such as Huffman coding, to study games such as checkers and chess and can help determine winning strategies for playing those games.

A tree is a non-linear data structure in which items are arranged in a sequence. It is used to represent hierarchical relationship existing among several data items.

A tree is defined as a finite set of one or more data items (nodes) such that there is a special data item called the root of the tree and its remaining data items are partitioned into a number of mutually exclusive (i.e. disjoint) subsets, each of which itself is a tree (called sub trees).Tree data structure grows downwards from top to bottom.

The tree consisting of a single vertex with no edges is called the trivial tree or the degenerate tree.

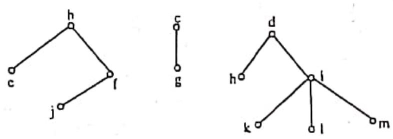


Tree

Tree Terminologies

- **Node:** Each data item in a tree is called a node. It is the basic structure in a tree. It specifies the data and links (branches) to other data items. There are 13 nodes in the above tree.
- **Root:** It is the first data item (or node) in the hierarchical arrangement of data items. In the above tree, a is the root item.
- **Degree of a node:** It is the number of sub trees (the no of trees which have node n as root node) of a node n in a given tree. For example, in the above tree,
    - The degree of node a is 3
    - The degree of node b is 2
    - The degree of node c is 1
    - The degree of node h is 0
    - The degree of node i is 3
- **Degree of a tree:** It is the maximum degree of nodes in a given tree. In the above tree, the degree of node a is 3 and another node i also have degree 4. In the whole tree, this value is the maximum. So, the degree of the above tree is 4.
- **Terminal node(s):** A node with out degree zero is called a **terminal node** or **leaf node.** In the above tree, there are 7 terminal nodes: e, j, g, h, k, l and m.
- **Non-terminal node(s):** Any node (except the root node) whose out degree is not zero is called **non-terminal node.** Non-terminal nodes are the intermediate nodes in traversing the given tree from its root node to the terminal nodes (leaves). In the above tree, there are 5 **non-terminal nodes:** b, c, d, f and i.
- **Siblings:** The children nodes of a given parent node are called siblings. They are also called **brothers.** In the above tree,
    - e and f are siblings.
    - k, l and m are siblings of parent node i

- **Level/Depth:** The number of edges that need to be followed while traversing a tree from root to that node is called depth of node. The entire tree structure is leveled in such a way that the root node is always at level 0. Then its immediate children are at level 1 and their immediate children are at level 2 and so on up to the terminal nodes. In general, if a node is at level n, then its children are at level n+1. In the above tree, there are four levels from level 0 to level 3.
- **Edge:** It is a connecting line of two nodes or relation between two nodes of tree.
- **Path:** Path in a tree is the sequence of consecutive edges from the source to the destination node of tree. In the above tree, the path between a and j is given by the node pairs: (a, b), (b, f) and (f, j).
- **Depth:** It is the maximum level of any leaf in a given tree. This equals the length of the longest path from root to the terminal nodes (leaves). The term **height** is also used to denote the depth. The depth of the above tree is 3.
- **Forest:** It is a set of disjoint trees. In a given tree, if we remove its root, then it becomes a forest. In the above tree, there is forest with three trees if we remove the root node.



Forest

m-ary tree: A rooted tree T = (V, E) in which every non-terminal (internal) vertex has no more then m-children, is called m-ary tree.

Full m-ary tree: A m-ary tree T = (V, E) where every internal vertex has exactly m children, is called full m-ary tree.

An m-ary tree with m=2 is called binary tree.

Example 1

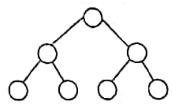


Fig.: Full 2-ary tree

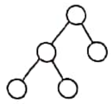


Fig.: 2-ary tree

Theorem:

A full m-ary tree with 'i' internal vertices contain n = mi + 1 vertices.

Proof: Let $T = (V, E)$ is a full m-ary tree with 'i' internal vertices except the root are child of internal vertices.

Since, by defination of full-m-ary tree each internal vertex has exactly on-children the total vertices except root is equal to $m \times i$.

Hence, A full m-ary tree T has total number of vertex $(n) = m \times i + 1$, including root.

Theorem:

A full m-ary tree with

(i) n vertices has $i = (n - 1)/m$ internal vertices and $l = [(m - 1) \times n + 1)]/m$ leaves.

(ii) i internal vertices has $n = mi + 1$ vertices and $l = (m - 1) \times i + 1$ leaves

(iii) l leaves has $n = (ml - 1)/(m - 1)$ vertices and $i = (l - 1)/(m - 1)$ internal vertices.

Proof:

Let $T = (V, E)$ be a full m-ary tree with n-number of vertices, i be number of internal vertices and l-be number of leaves.

Since,

A full m-ary tree with i internal vertices has

$n = mi + 1$ vertices ....(i)

and

$n = i + l$ [∵ total vertices = no. of internal vertices + leaf vertices)

From eq^n. (i)

$n = mi + 1$

or, $mi = n - 1$

or, $i = (n - 1)/m$ Proved.

Again, putting value of i in equation (ii) we get,

$n = (n - 1)/m + l$

or, $n - \frac{(n - 1)}{m} = l$

or, $\frac{m * n - n + 1}{m} = l$

or, $\frac{n(m - 1) + 1}{m} = l$

i.e. $l = \frac{n(m - 1) + 1}{m}$

(ii) We know that

$n = mi + 1$ .....(i)

and $n = i + l$

Now, putting value of n from (i) to (ii), we get

$mi + 1 = i + l$

$mi + 1 - i = l$

i.e. $l = i(m - 1) + 1$

(iii) We know that

$n = mi + 1$ ....(1)

and $n = l + i$ .....(2)

Solving (1) and (2)

$mi + 1 = l + i$

$mi - i = l - 1$

$i(m - 1) = l - 1$

$i = \frac{(l - 1)}{(m - 1)}$ Proved.

Again, putting value of (i) in (1)

$n = \frac{m(l - 1)}{m - 1} + 1$

(iv) $n = \frac{m(l - 1) + m.1}{m - 1}$

$n = \frac{ml - m + m - 1}{m - 1}$

$n = \frac{ml - 1}{m - 1}$ Proved.

Theorem

There are at most m^h leaves in an m-ary tree of height h.

Proof

Let $T = (V, E)$ is a m-ary tree with height h. Now, we prove the above statement using method of induction on the height h. Let P(h) : m-ary tree with hight in has at most m^h leaves.

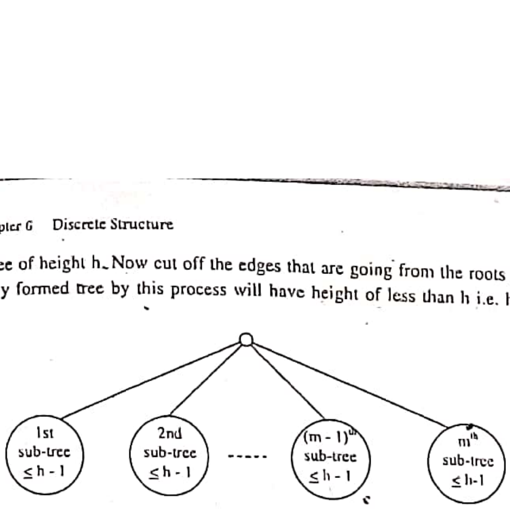
Basis step

When height $h = 1$, then P(1) tree consist of a root with no more than m-children, each of these children are leaf. Hence, there are no more than $m^n = m^1 = m$ leaves in an m-ary tree of height 1 i.e. P(1) is true.

Inductive hypothesis: Let us suppose that the statement is true for all m-ary tree with hight less than h i.e. h - 1.

Inductivestep: Now, we have to prove that the statement is true for h.

Take a tree of height h. Now cut off the edges that are going from the roots to its children. The newly formed tree by this process will have height of less than h i.e. h - 1 as shown below:



From inductive hypothesis, we know that the a m-ary tree with height h - 1 has at most m^{h-1} leaves.

Here, there are at most 'm' such sub-tree of height h - 1 as the original tree T is as m-ary tree (i.e. m-ary tree has at most m-children).

So the total number of leaves on the original tree of hight h has at most

$$= m * m^{h-1}$$

$$= m * \frac{m^h}{m}$$

$$= m^h$$

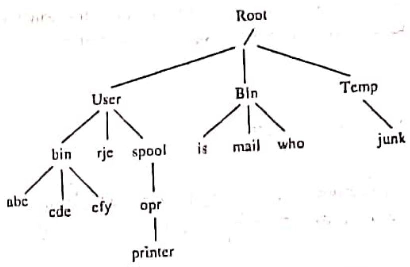
This proves the theorem.

Tree as Models

Tree are used for modelling different problems related to diverse areas such as computer science, chemistry, botany and zoology etc.

Tree as models in computer file system

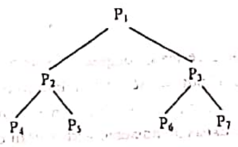
Files in computer memory can be organized into directories or folder in which folder contains both files and sub-folders. In file system, root directory or folder contains files in the system. So a file system can be represented by a rooted tree, where the root represents main folder, internal vertices represents sub-folders and leaf nodes represents ordinary files or empty folders.



Part of linux file system.

Tree as a model for parallel processing

A parallel processing system consists of many processing devices (CPU). These processor are interconnected with each other to complete a task in parallel computing fashion. Tree can be used to represent the connection between these processor. Here, a processor is represented by vertex and the edges between these vertices are used to represent connection between them. A tree connected network for seven processors is as shown in figure.



A tree connected network.

Theorem

Let G(V, E) be a loop-free undirected graph. Then G is a tree if there is a unique path between any two vertices of G.

Proof

Let G=(V, E) be a Graph. Since there is a path between each pair of vertices u and v, G must be connected. Thus, to show G is a tree it remains to show that G has no cycles. Since G is loop-free, it has no cycles of length 1. Suppose that G has a cycle of length greater than one, say

$$C = (v_1, v_2, \ldots\ldots, v_n, v_1).$$

Then any two distinct vertices of the cycle C are joined by two paths, which contradicts the fact that there is an unique path between any two vertices of G. Hence, G has no cycles and so it is a tree.

Theorem

A tree with n vertices has exactly $n - 1$ edges.

Proof

Let $G=(V,E)$ be a tree with n vertices. We use induction method to prove it. Let $n = 1$ i.e. G has one vertex, since it has no loops, the number of edges in G is 0. This establishes that it is true for $n = 1$.

Let it be true for n. Now we wish to show it is true for $n + 1$. Let G be a tree with $n + 1$ vertices and let u be a vertex of degree 1. If we remove such a vertex and the edge incident on it, then sub graph $G - u$ is still connected and has no cycles. Hence $G - u$ is a tree. However, $G - u$ has n vertices, so by induction it has $n - 1$ edges. Since $G - u$ has exactly one edge less than that of G, it follows that G has n edges. So assuming $n + 1$ vertices of G we got n edges. This completes the proof.

Theorem

In any tree G, there are at least two pendant vertices.

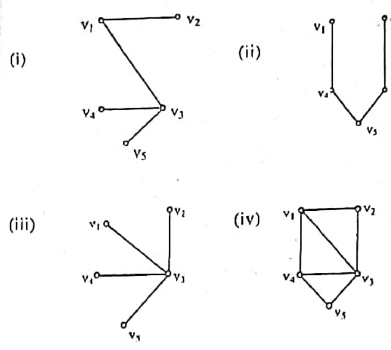
Proof

Let $G=(V,E)$ be a tree and v_1 be a vertex of degree 1. If there is a vertex v_2 adjacent to v_1 with degree 1 then we are done. If not, we select v_3 adjacent to v_2. Continuing in this way, we get a sequence of vertices $v_1, v_2, \ldots\ldots v_k$. Since a tree has no cycle, this sequence terminates for some K i.e. there exists an adjacent vertex with degree 1. Thus v_1 and v_k are our desired vertices each of degree 1.

Spanning Tree

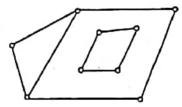
A graph has many subgraphs and some of these subgraphs are trees as well. Such trees lead to the definition of spanning trees.

Let $G = (V,E)$ be any connected graph then any subgraph $T=(V,E')$ of given graph G consist of all the vertices in G, which is still connected and acyclic is called spanning tree of given graph. In figure (i), (ii) and (iii) are the spanning tree of (iv).



Example 2

Determine whether the graph shown below has an spanning tree.

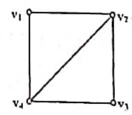


Solution

Since the graph shown above is not a connected graph, it has no spanning tree.

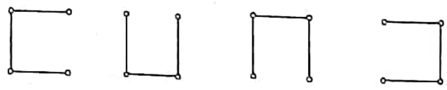
Example 3

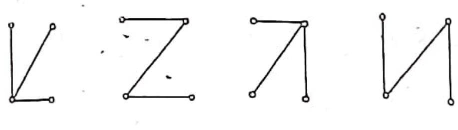
Write all the spanning trees of the graph G shown in figure below.



Solution

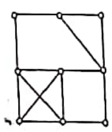
Since the graph in figure below is connected, so it has some spanning trees which are shown below:





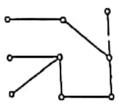
Example 4

Draw two spanning trees for the graph given in figure below.

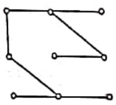


Solution

(i)



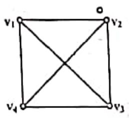
(ii)



Note: According to English mathematician Arthur Cayley, for a complete graph K_n with n vertices there are n^{n-2} different ways of joining them to form a tree.

Example 5

How many different spanning trees can be formed from K_4, shown in figure?



K_4 or 3 – regular graph

Solution

Since, K_4 consists of n = 4 vertices, there are $(4)^{4-2} = 16$ different ways of joining them to form a spanning tree. This means 16 different trees can be formed.

Theorem

A simple graph is connected if and only if it has a spanning tree.

Proof

If a graph G has a spanning tree than by definition of spanning tree it is connected.

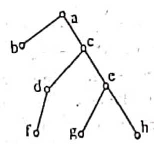
Conversely, let G be a connected graph, if G has no cycles, then G is itself a tree. Hence, in the case, G itself is a spanning tree. Now suppose that G has at least one cycle. If we delete an edge of the cycle of G, the resulting graph G_1 is connected and has the same set of vertices as G. If G_1 has a cycle then we delete an edge of a cycle of G_1 and so on. In this way, we ultimately arrive at a connected graph H which has no cycles and has the same set of vertices of G. Hence, by definitions H is a spanning tree of G which complete the proof of theorem.

Rooted Trees

A rooted tree is a tree in which a particular vertex is distinguished from the others and that particular vertex is called the root of the tree. Since, there is a unique simple path from the root R to any other vertex v in T, which determines a direction to the edges of T. Thus, T may be viewed as a directed graph. We note that any tree may be made into a rooted tree by simply selecting one of the vertices as the root.

Example 6

Consider the rooted tree in figure below.



(a) What is the root of T?

(b) Find the leaves and internal vertices of T.

(c) What are the levels of c and e?

(d) Find the children of c and e.

(e) Find the descendents of the vertices a and c.

Solution

(a) Vertex a is distinguished as the only vertex located at the top of the tree, so vertex a is the root.

(b) The leaves are those vertices that have no children. Thus, b, f, g and h are the leaves and vertices c, d and e are the internal vertices.

(c) The levels of c and e are 1 and 2 respectively.

(d) The children of c are d and e; and g and h are those of e.

(e) The descendents of a are b, c, d, e, f, g and h; and the descendents of c are d, e, f, g and h.

Algorithms for Constructing Spanning Trees

We have already discussed that from a given connected graph G to construct a spanning tree we delete one or more edges. Instead of constructing spanning trees by removing edges, spanning trees can be built by successively adding edges. There are two algorithms based on this principle for finding a spanning tree and they are Breadth-first search (BFS) and Depth-first search (DFS).

(a) BFS Algorithm

In this algorithm a rooted tree will be constructed, and the underlying undirected graphs of this rooted tree forms the spanning tree. The idea of BFS is to visit all vertices on a given level before going into the next level.

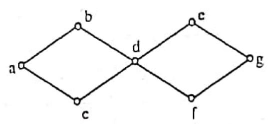
Procedure: Arbitrarily choose a vertex and designate it as the root. Then add all the edges incident to this vertex such that the addition of edges does not produce any cycle. The new vertices added at this stage become the vertices at level 1 in the spanning tree. Next, for each vertex at level 1, visited in order, add each edge incident to this vertex to the tree as long as it does not produce any cycle. Arbitrarily order the children of each vertex at level 1. This produces the vertices at the level 2 in the tree. Continue this process until all the vertices in the tree have been added. The procedure will end since, there are only a finite number of edges in the graph. A spanning tree is produced since we have produced a tree without a cycle but containing every vertex of the graph.

Pseudocode

```
BFS (G,s)        /*s is start vertex*/
{
    T={s};
    L=ϕ;             /*an empty queue*/
    Enqueue (L,s);
While (L!= ϕ)
{
    V = Dequeue(L);
    For each neighbor w to v
    {
        If(w ∉ L and w ∉ T)
        {
            Enqueue(L,w);
            T=T U {w}     /*put edge {v,w} Also*/
        }
    }
}
}
```


Example 7

Use BFS algorithm to find a spanning tree of graph G of figure.



Solution

(i) Choose the vertex a to be root.

(ii) Add edges from the vertex a to all adjacent vertices on it. Then the edges {a, b}, {a, c} are added.

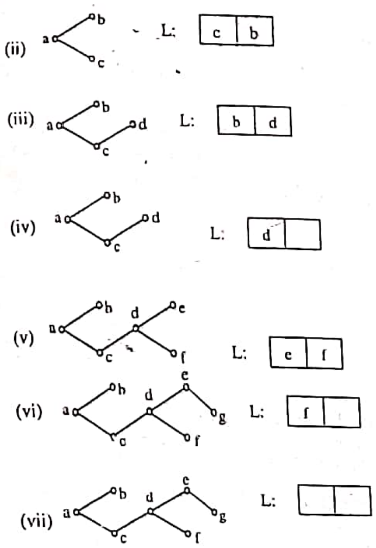
(iii) Add edges from these vertices at level 1 to adjacent vertices which are not already joined in the tree. Hence the edge {c, d} is added. The vertex d is at level 2.

(iv) Add edges from d in level 2 to adjacent vertices not already in the tree. The edges {d, e} and {d, f} are added. Hence e and g are at level 3.

(v) Add edge from e at level 3 to adjacent vertices which are not already in the tree and hence {e, g} is added.

The steps of BFS are shown in figure below

(i) a L: a



(b) DFS Algorithm

An alternative to BFS is DFS which proceeds to successive levels in a tree at the earliest possible opportunity. DFS is also called back tracking.

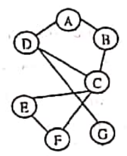
Procedure: Arbitrarily choose a vertex and designate it as the root. Form a path starting from the root 'a' by successively adding edges as long as possible so that the edges do not produce any cycle. If the path goes through all the vertices of the given graph, the tree with this path is a spanning tree. If the path does not go through all the vertices of the given graph, we move back to the next to last vertex in the path, and, if possible, form a new path starting at this vertex which passes through those vertices which were not already visited. If this cannot be done, we move back to another vertex and repeat the same thing. We repeat the procedure until all the vertices are visited.

Pseudocode

```
DFS(G,s)
{
    T={s};
    Traverse(s);
}
Traverse(v)
{
    for each w adjacent to v and not yet in T
    {
        T= T U {w};   /*put edge {v,w} also*/
        Traverse(w);
    }
}
```

Illustration of above algorithms

A	B	D		
B	A	C		
C	B	D	E	F
D	A	C	G	
E	C	F		
F	C	E		
G	D			



Edge table (adjacent table)

In order to do a DFS is to follow the chain of edges as far as we can into graph before we back track to pick u other vertices. As we visit each vertex we cross it off everywhere it appears in the edge table. Since we are starting with A, cross it off every where it appears.

~~A~~	B	D		
B	~~A~~	C		
C	B	D	E	F
D	~~A~~	C	G	
E	C	F		
F	C	E		
G	D			

A ⟶ B ⟶ C

A ⟶ B ⟶ C ⟶ D ⟶ G

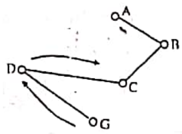
Hence we have seen that while traversing we start at A and got B and moved to vertex B, then start traversing and found C then to vertex D. from first row. In that row A and C are already visited and next found G which send us to 7th row from where there is no any other vertex to traverse. Now we have not visited E and F, to visit E and F we back track

G → D → C.

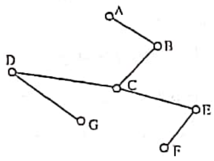
Hence, we found vertex E not visited thus we visit E and then F from 6th row Hence final traverse list will be

A → B → C → D → G → E → F → Q.

Therefore the spanning tree of given graph is as shown below:



backtracking→



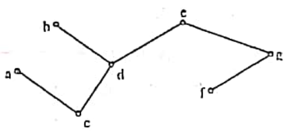
spanning tree

Example 8

Find a spanning tree of the graph of figure using DFS algorithm.

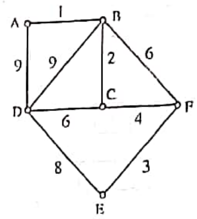
Solution

Choose the vertex a to be the root. Form a path starting from the vertex a by successively adding edges incident with vertices which are not already in the path as long as possible. This produces the path a – c – d – e – g – f. since this path does not include the vertex b, so we move back to g. There is no path beginning at g containing those vertices which are not already visited. Similarly we move back to e, there is no new path. So we move back to d and form the path d – b. This produces the required spanning tree which is shown in figure.



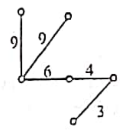
Minimal Spanning Trees

Let G be a connected weighted graph. The weight of a spanning tree of G is the sum of the weights of the edges which are included on that spanning tree. Thus, a minimal spanning tree of G is a spanning tree of G with minimum possible weight among all possible spanning trees of that graph. Consider figure which shows six cities and cost of lying railway links between certain pair of cities. We want to set up railway links between the cities at minimum costs.

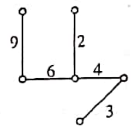


Now, we consider some weighted spanning trees of the weighted graph of figure given below.

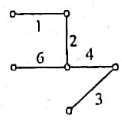
(i)



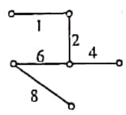
(ii)



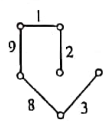
(iii)



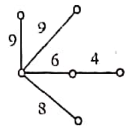
(iv)

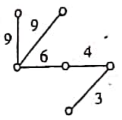


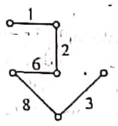
(v)



(vi)



(vii) 

(viii) 

Among the spanning trees in figure, spanning tree (iii) is defined as the minimal spanning tree since its weight is minimum as compared to the weights of the other spanning trees.

It is unnecessary to find the weights of all the possible spanning trees of a given weighted graph to determine its minimal spanning tree. There are several algorithms for determining the minimal spanning tree of the given weighed graphs. However we shall discuss only two algorithms here.

Kruskal's Algorithm

The algorithm involves the following steps:

Step 1: List all the edges of G with non–decreasing order of their weights.

Step 2: Select an edge of minimum weight (if there are more then one edge of minimum weight, arbitrarily choose one of them). This will be the first edge of T.

Step 3: At each stage, select an edge of minimum weight from all the remaining edges of G if it does not form a cycle with the previously selected edges in T. Then add the edge to T.

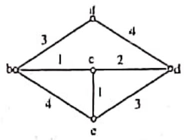
Step 4: Repeat step 3 until n –1 edges have been selected.

Pseudo-code for kurskal's Algorithm

```
KruskalMST(G)
{
    T={V}// Set of vertices
    E= Set of edges sorted in non-decreasing order of their weight
    while(|T| <n-1 and E!=NULL)
    {
        select(u,v) from the E in order
        remove (u,v) from E
        if(u,v) does not create cycle in T
            T=TU{(u,v)
    }
}
```


Example 9

Show step by step, how Kruskal's algorithm can be used to find a minimal spanning tree for G of figure.



Solution

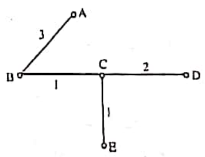
Step 1: List the edges with non–decreasing order of their weights.

Edge	(b, c)	(c, e)	(c, d)	(a, b)	(d, e)	(a, d)	(b, e)
Weight	1	1	2	3	3	4	4

Step 2: The edge (b, c) has the smallest weight, so include it in T.

Step 3: An edge with next smallest weight is (c, e), so include it in T.

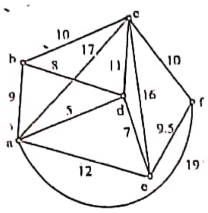
Step 4: Similarly next smallest weight including edge is (c, d). Since it does not form a cycle with the existing edges in T, include it in T. Similarly we follow this process until T has 5 – 1 = 4 edges which are shown in figure and weight is: 1+1+2+3=7



T = (V, E)

Example 10

Find a minimal spanning tree of weighted graph G in figure, using Kruskal's algorithm.

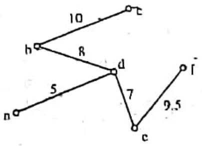


Solution

The graph G has six vertices. Hence, any spanning tree of G will have (6 – 1) = 5 edges.

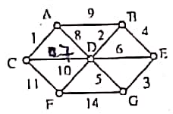
Edges	(a,d)	(d, c)	(b,d)	(a, b)	(c, f)	(b, c)	(c, f)	(d, c)	(a, e)	(c,c)	(a, c)	(a, f)
Weight	5	7	8	9	9.5	10	10	~~11~~	12	16	17	19
Select	Yes	Yes	Yes	No	Yes	Yes	No	No	No	No	No	No

Thus, the minimal spanning tree of G contains the edges {(a, d), (d, c), (b, d), (c, f), (b, c)}. This minimal spanning tree of weight 39.5 is shown in figure.



Example 11

Find a minimal spanning tree of the graph G in figure, using Kruskal's algorithm.

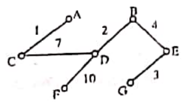


Solution

The graph G has seven vertices. Hence, any spanning tree of G will have (7 – 1) = 6 edges. By applying Kruskal's algorithm, we have the following data:

Edge	AC	BD	EG	BE	DG	DE	DC	AD	AB	DF	CF	FG
Weight	1	2	3	4	5	6	7	8	9	10	11	12
Select	Yes	Yes	Yes	Yes	No	No	Yes	No	No	Yes	No	No

Thus, the minimal spanning tree of G contains the edges (AC, BD, EG, BE, DC, DF). This minimal spanning tree of weight 27 is shown in figure.



T = (V, E)

Prim's Algorithm

This algorithm involves the following steps:

Step 1: Select any vertex in G. Among all the edges which are incident to it, choose an edge of minimum weight. Include it in T.

Step 2: At each stage, choose an edge of smallest weight joining a vertex which is already included in T and a vertex which is not included yet so that it does not form a circuit. Then include it in T.

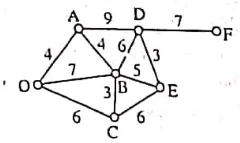
Step 3: Repeat until all the vertices of G are included with n – 1 edges.

The pseudo-code for prim's algorithm

```
primsMST(G)
{
    T={ }    //set of edges of MST
    S={s}    //s is the randomly chosen vertex
    while(S!=V)
    {
      e={u,v} //an edge of miimum weight incident to vertices in s and not forming a simple cycle in T.
      T=T U{u,v}
      S=SU{v}
    }
}
```

Example 12

Find the minimal spanning tree of the weighted graph G of figure using Prim's algorithm.



Solution

Step 1: We choose a vertex O. Now edge with smallest weight incident on O is OA. So we include OA in T.

Step 2: Now w(OB) = 7, w(OC) = 6, w(AB) = 4 and w(AD) = 9. We choose the edge AB since it is minimum. Then we include it in T.

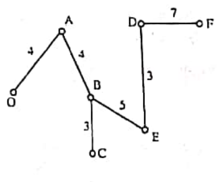
Step 3: Now w(BD) = 6, w(BE) = 5, w(BC) = 3, w(AD) = 9, w(OC) = 6. So we choose BC and include it in T.

Step 4: Again w(OC) = 6, w(CE) = 6, w(BE) = 5. So we choose BE and include it in T.

Step 5: Since w(ED) = 3, we include it in T.

Step 6: Now w(DF) = 7, and further we have no choice since if we choose any remaining edge whose weight is less or equal to seven, it forms a circuit. So we include DF in T.

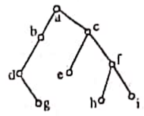
So we have (7 – 1) = 6 edges in a minimal spanning tree, which is shown in figure and weight is: 3+3+4+4+5+7=26



T = (V, E)

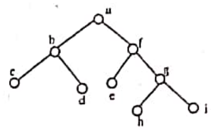
Binary Tree

A binary tree is a finite set of elements that is either empty or partitioned into three disjoint subsets. the first subset contains the single elements called root of the tree, the other subsets are called left and right sub tree of original tree. The binary tree is so named because each node can have at most two descendants.



Strictly Binary Tree

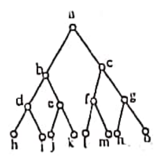
A binary tree is called strictly binary tree if every non-leaf node in binary tree has non-empty left and right sub-tree.



Strictly binary Tree

Complete (Full) Binary Tree

A strictly binary tree in which all the leaf nodes lies on same level is called complete binary tree.



Complete Binary Tree

Tree Traversal

Tree traversal is one of the most common operations performed on tree data structures. Traversing a binary tree is a way in which each node in the tree is visited exactly once in a systematic manner. We know that the order of traversal of nodes in a linear list is from first to last, however there is no such "natural" linear order for the nodes of a tree.

We have the following three orderings or methods for traversing a non-empty binary tree:

- Preorder (Depth-first order) Traversal
- Inorder (Symmetric order) Traversal
- Postorder Traversal

All theses traversal techniques are defined recursively.

Preorder Traversal

In this technique, first of all we process the root R of the binary tree T. Then, we traverse the left subtree T_1 of R in preorder (which means that we traverse root of subtree T_1 first and then its left subtree). After visiting left subtree of R, then we take over right subtree T_2 of R and process all the nodes in preorder.

Steps

1. If root=NULL, return.
2. Visit the root.
3. Traverse the left subtree in preorder.
4. Traverse the right subtree in preorder.

Inorder Traversal

In this traversal technique, first of all we process the left subtree T_1 of the root R in inorder, then we process the root and at last the right subtree T_2 of R.

Steps

1. If root=NULL, return.
2. Traverse the left subtree in inorder.
3. Visit the root.
4. Traverse the right subtree in inorder.

Postorder Traversal

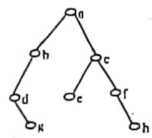
In this technique, first of all we process the left subtree T_1 of R in postorder, then right subtree T_2 of R in postorder and at the last, the root R.

Algorithm for postorder traversal

1. If root=NULL, return.
2. Traverse the left subtree in postorder.
3. Traverse the right subtree in postorder.
4. Visit the root.

Example 13

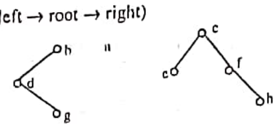
Find the preorder, in-order and post-order traversal of following tree.



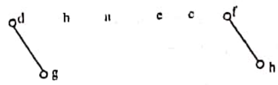
Solution

In order (left → root → right)

Step 1:



Step 2:

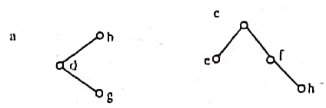


Step 3:

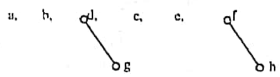
d, g, b, a , e, c, f, h

Pre-order (root → left → right)

Step 1:



Step 2:

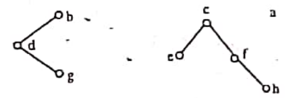


Step 3:

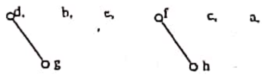
a, b, d. g, c, e, f, h

Post order (left → right → root)

Step 1:



Step 2:



Step 3:

d, g, b, e, h, f, c, a

Therefore,

Preorder	a	b	d	g	c	e	f	h
In oder	d	g	b	a	e	c	f	h
Postorder	g	d	b	e	h	f	c	a

Applications of Tree

Prefix codes

A prefix code is a special type of coding system in which each code satisfy a "prefix property".

Prefix property means there is no code were in the system that is prefix of any other code word in the system.

Example 14

A code with codings (code word) {0, 10, 11} is a prefix code. But the coding system with code words {0, 01, 011} is not a prifix code since 0 is prefix of second code word (01) and 01 is prefix of third code word.

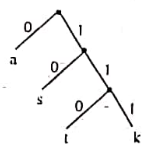
A prefix code can be represented using binary tree where the symbols are leaf node and edges are labeled with code words.

Character
e
a
t
n
s

The edges of the tree are labeled so that an edge leading to a left child is assigned a '0' and an edge leading to a might child is assigned a bit '1'. Now the bit string used to encode a character is the sequence of edges in the unique path from the root to that leaf node with specific character.

Example 15

Let us draw a tree as follows (by maintaining rule of prefix tree as explained below):



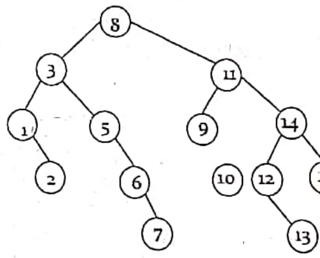
Now, the prefix for the character a, s, t and R can be written as

Character	Code
a	0
s	10
t	110
k	111

Binary Search Trees

A binary search tree is a binary tree that is either empty or in which each node possesses a key that satisfies the following three conditions:

- For every node X in the tree, the values of all the keys in its left subtree are smaller than the key value in X.
- Keys in its right subtree are greater than the key value in X.
- The left and right subtrees of the root are again binary search trees.



A Binary Search Tree

Example 16

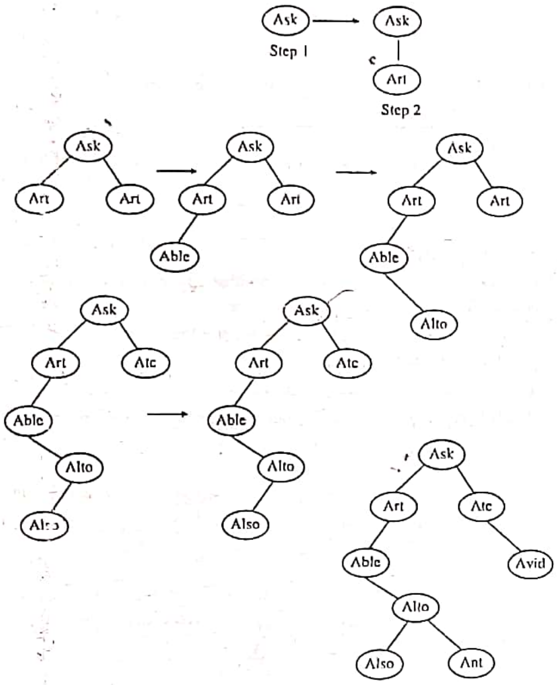
Draw a BST for the flowing words: ask, art, ate, able, alto, also, avid, ant.

Solution

Given list of key is

ask, art, ate, able, alto, also, avid, ant

The finite key in the list act as root key for node in the tree.



[Note: Since alphabetically art comes before the Ask, it is placed on the left of node "Ask". Similarly, "Ate" curve after the word "Ask" so it is placed on the right of node "Ask" and so on].

Algebraic Expression Tree

Binary trees are used to represent algebraic expressions. An expression tree is a binary tree used to represent mathematical expression in which all the leaf node contains operands such as constant or variables and non leaf node contains mathematical operators. Operations such as +, −, ×, ÷, ↑, /, etc. can only be assigned to the internal nodes. This particular tree happens to be binary because all of the operations are binary.

Example 17

Use a binary tree to represent the expressions

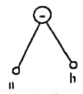
(i) $a - b$ (ii) $(a + b) * c$

(iii) $(a + b) * (c - d)$ (iv) $((a + b) * c) + (d / e)$

(v) $(a + b) * (c + (d / e))$

Solution

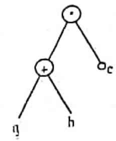
(i)



Polish Notation: -ab

Reverse Polish Notation: ab-

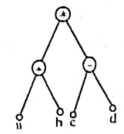
(ii)



Polish notation: *+abc

Reverse Polish notation: ab+c*

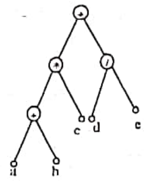
(iii)



Polish Notation: *+ab-cd

reverse Polish Notation: ab+cd-*

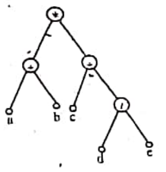
(iv)



Polish Notation: +*+abc/de

Reverse Polish Notation: ab+c*de/+

)



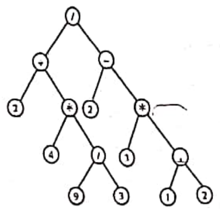
Polish Notation: *+ab+c/de   Reverse Polish Notation: ab+cde/+*

Example 18

Construct an expression tree for given algebraic expression:

(2 + (4 ∗ 9/3)) / (2 – (3 ∗ (1 + 2)).

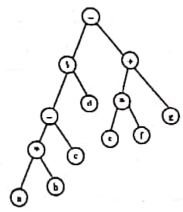
Solution



Example 18

Draw the binary tree to represent the expression (((a ∗ b) – c)$d) – ((e ∗ f) + g) and find its polish and reverse polish notation.

Solution



Now the polish notation for the above expression tree is:

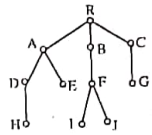
-$-*abcd+*efg

and the reverse polish notation is:

ab*c-d$ef*g+-

EXERCISE

1. Consider the tree with root R, shown below:



(a) Identify the path from the root R to each of the following vertices and find the level number of the vertex:

(i) D (ii) J (iii) G

(b) Find the leaves of T (tree).

(c) Assuming T is an ordered rooted tree, find the universal address of each leaf of the tree.

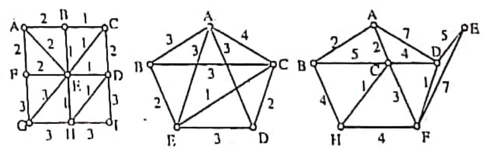
2. Obtain the rooted tree with 'a' as the root from the graph G(V, E) where V = {a, b, c, d, e, f, g, h, I, j, k, l, m} and E = {(c, d), (b, c), (a, b), (a, g), (g, h), (h, k), (a, l), (c, e), (g, i), (g, j), (j, m), (j, f)} . Also find i) ancestors of e, ii) siblings of h, iii) height of tree.

3. If D(V, A) be a directed graph, where the vertices set V = { a, b, c, d, e, f, g, h, i} and the directed edges set A = {(b, c), (b, a) , (d, e), (d, f), (e, h), (f, g), (d, b), (g, i), (g, j)}. Show that D (V, A) is a rooted tree and identify the root and leaves. Also find the height of the tree.

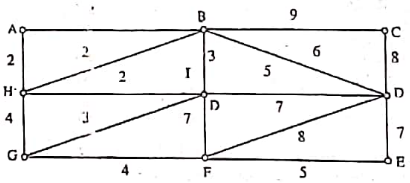
5. Obtain the rooted tree with 'a' as the root from the graph G(v, E), where v={a, b, c, d, e, f, g, h, i, j, k, l, m} and E={(c, d), (b, c), (a, b), (a, g), (g, h), (h, k), (a, l), (c, e), (g, i), (g, j), (j, m), (j, l). Also, find (i) level of each vertices (ii) siblings of h (iii) ancestors of e (iv) height of tree.

6. Find a minimal spanning tree in the weighted graphs shown below by using both Kruskal's algorithm and Prim's algorithm.

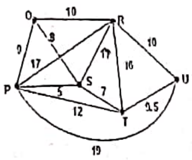
(a) (b) (c)



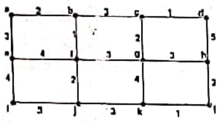
7. Define minimal spanning tree. Find the minimal spanning tree of the following weighted graph.



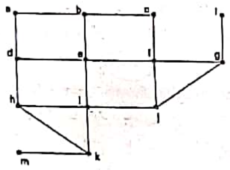
8. Find a minimal spanning tree of the given graph by using Kruskal' algorithm.



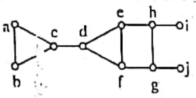
9. Use kruskal's algorithm to find a minimum spanning tree of the graph shown below.



10. Use breadth first search to find a spanning tree for the graph.



11. Find a spanning tree in the following graphs using depth first search and breadth-first search. Choose 'd' as the root of the spanning tree.



7 Chapter

Network Flow

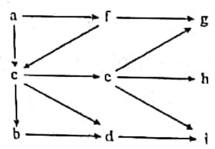
Definition of Transport network

In graph theory, a flow network (also called transport network) is a directed graph where each edge has a capacity and each edge receives a flow. Here, capacity implies the maximum rate at which something flows through the edge. The amount of flow on an edge cannot exceed the capacity of the edge. All vertices except the source denoted by S and sink denoted by D are called intermediate vertices. A flow must satisfy the restriction that the amount of flow into an intermediate vertex equals the amount of flow out of it, unless it is a source S, which has only outgoing flow, or sink D, which has only incoming flow. A flow network can be used to model traffic in a road system, fluids in pipes, currents in an electrical circuit, or anything similar in which something travels through a network of nodes.

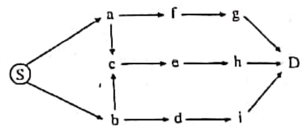
Note: In most case, there is single source and single destination but sometimes there are multiple sources and multiple destinations. In such case, we can create the equivalent transport network by combining there multiple source auto single source with additional edge.

Example 1

In the following figure, there are two sources a and b and three destinations g, h, and i.



Then, the equivalent transportation network with single source s and single destination D is as shown in figure.



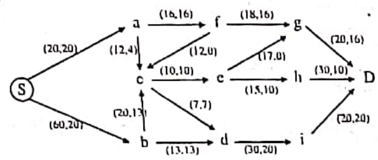
Flows: The materials (or oil) transport along any edge is known as flow.

Let us label edges with two no. first label represents the capacity of edge and second represent the amount of flow through that edge. The two criteria are:

(i) The amount of flow can't exceed the capacity of that edge.

(ii) For every node (except S & D), the amount of flow flowing into vertex v (incoming flow) must be equal to the account of flow flouring out of vertex v.

Example 2



Σincoming flow at C = 17

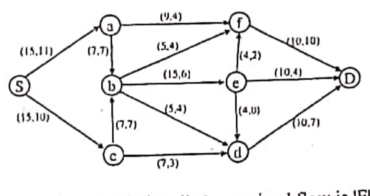
Σoutgoing flow at C = 17

[Note: Total flow leaving at S = Total flow arriving at D]

Network Flow Problem

Given a transport graph G = (V, E) where each edge e is associated with its capacity C(e) > 0 and the two special node: source node S and sink or destination node D then the network flow problem states that what is the maximum total amount of flow is possible to carry on from source node S to D.

Example 3



A flow F in a network (G, k) is called a maximal flow is |F| ≥ |F'| for every flow F' in (G, k).

Computing Max Flow: Concept

- Identify the augmenting path
- Increase flow along that path
- Repeat the above two steps till the maximum flow is obtained.

Augmenting path

- An augmenting path P is simple path from S to D with unsaturated edge.
- We build augmented path in two ways:
  (1) Augmenting path with only forward unsaturated edge.
  (2) Augmenting path with same backward edge.

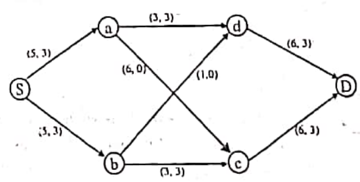
In backward edge A.P. the backward edge must have positive flow and of course, the forward edge must be unsaturated.

Two Basic Ways to Increase the Value of Flows

(1) If an edge is not being used to capacity, try to send more flow through it.

(2) If an edge is working against us by sending some flow back toward the source, we could try to reduce the flow along this edge and redirect in a more practical direction.

Example 4



Here, we increase flow along path S - a - d - D.

Now,

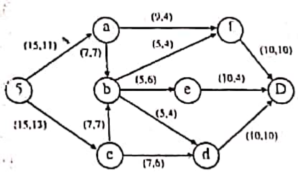
k(S, a) - F(S, a) = 5 - 3 = 2

k(a, d) - F(a, d) = 6 - 0 = 6

k(d, D) - F(d, D) = 6 - 3 = 3

∵ Min. {2, 6, 3} = 2, increase flow by 2.

Example 5



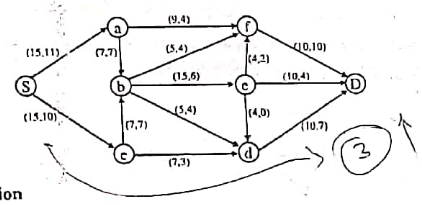
There is no augmenting path with only forward edge. So we have to find (if any) augmenting path with some backward edge. One of such path is:

S - c - d - b - e - D

This is an augmenting path since- we can increase flow by 1 along the path. Note- in backward edge, we will subtract the flow by 1, to maintain the rule of flow conservation)

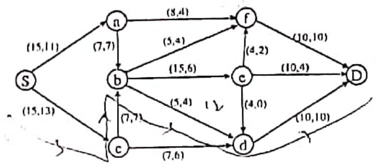
Example 6

Find a maximal flow for the network below:



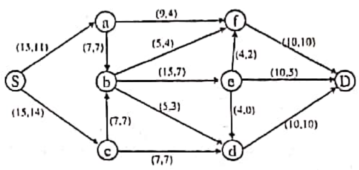
Solution

Here, first we look for an augmenting path where, if possible, all edges are forward edges (and they have to be unsaturated). The path P_1 : s - c - d - D is such a path. The minimum slack of the edges along P_1 is 3. By increasing the flow by 3 with along the edges of P_1, we obtain the flow F_1 as shown below.

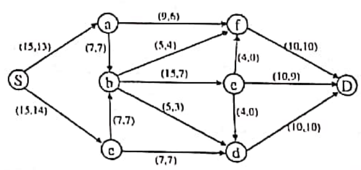


Now, there are no directed S-D paths with unsaturated edges. Thus, we look for augmenting paths with some backward edges. Note that the backward edges must have positive flow and the forward edges must be unsaturated.

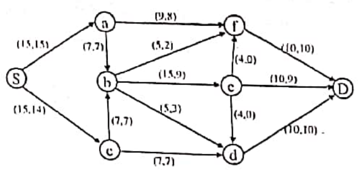
Now, we augment the flow by 1 along the path S - c - d - b - d - D to obtain the flow as in figure below. Note that we have decreased the flow by 1 unit along the edge (b, d) and increased the flow by 1 unit along other edges in the path S - c - d - b - e- D.



Now, we augment the flow by 2 along the path S - a - f - e - D to obtain the flow as in figure below:



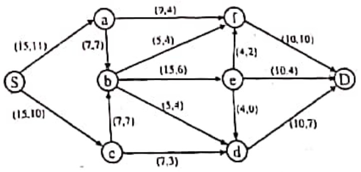
Next, we augment the flow by 2 along the path S - a - f - b - e - D to obtain the flow as in figure below.



There are no more augmenting paths from S to D because (S, a) in saturated and both edges out of c are saturated. Thus the maximal flow is 29.

Example 7

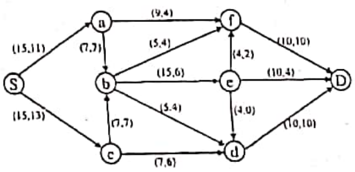
Find a maximal flow for the network shown below:



Solution

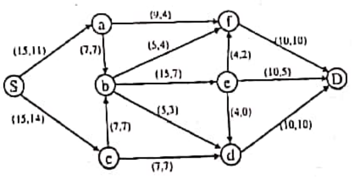
First we try to find out the augmenting path with all forward edge if possible the path: P_1 = S → c → d → D is such path.

The minimum slack (the value that can be augmented /added) along P_1 is 3. After increasing by 3, we have the flow network as:

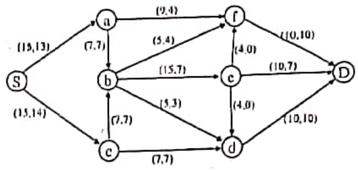


Now, there is no directed augmenting path. So we look for augmenting path with some backward edge.

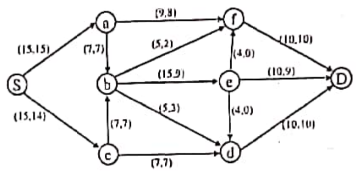
The path: P_2: S - c - d - b - e - D is such path with minimum slack. So after augmenting this path with flow 1, the network is:



Now, another augmenting path with backward edge is P_3: S → a → f → e → D, where the minimum slack is 2 (since we can't make backward edge -ve), the flow network is



Now, another augmenting path with backward edge is P_4 : S - a - f - b - e - D, where the minimum slack is 2, the flow network augmenting this path.



S - D Cut or Cut

A cut of transport network is a set of edge whose removal will divide the network into two halves X and $\bar{X}$ where

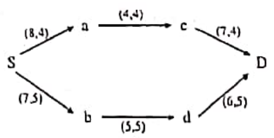
- Source vertex D∈X
- Sink vertex D∈$\bar{X}$

It is denoted by (X, $\bar{X}$)

Capacity of Cut

The capacity of S-D cut (X, $\bar{X}$) is defined as sum of all capacitors of edges from X to $\bar{X}$. It is denoted by K(X, $\bar{X}$).

Example 8



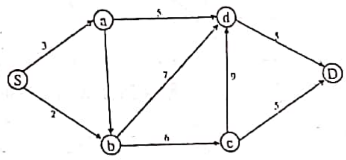
The SD cuts are:

X	$\bar{X}$	Cuts $(X, \bar{X})$	$K(X, \bar{X})$
S	{a, b, c, D}	{(S, a), (s, b)}	15
{S, a}	{b, c, d, D}	{(S, b), (a, c)}	11
{S, a, c}	{b, d, D}	{(S, b), (c, D)}	14
{S, a, b}	{C, d, D}	{(a, c), (b, d)}	9
{S, b, d}	{a, c, D}	{(s, a), (d, D)}	14
{S, a, c, b}	{d, D}	{(c, d), (b, d)}	13
{S, a, b, d}	{c, D}	{(a, c), (d, D)}	10
{S, a, c, b, d}	{D}	{(c, d), (d, D)}	13

The capacity of minimal cut is 9.

Example 9

Calculate the capacity of all possible S-D cuts in the following flow network.



The possible S-D- cuts are:

X	$\bar{X}$	Capacity of $(X, \bar{X})$
{S}	{a, b, c, d, D}	3 + 2 = 5
{S, a}	{b, c, d, D}	2 + 4 + 5 = 11
{S, b}	{a, c, d, D}	3 + 7 = 10
{S, c}	{a, b, d, D}	3 + 2 + 6 + 5 = 16
{S, d}	{a, b, c, D}	3 + 2 + 9 + 5 = 19
{S, a, b}	{c, d, D}	5 + 7 = 12
...	...	...
{S, a, b, c, d}	{D}	5 + 5 = 10

∴ Minimal cut = 5

Maximal flow and minimal cuts

Suppose (G, k) in a flow nework with source S and sink D, where G represents the graph and k represents the capacity of the edges. Suppose that X in a set of vertices such that S ∈ X, but D ∉ X. Let $\bar{X}$ denotes the complement of X in V(4). Then the set $(X, \bar{X})$ of all edges from a vertex in X to a vertex in $\bar{X}$ is called an S - D cut.

If C is any set of edges in a transport network (G, k), then the capacity of C is the sum of the capacities of the edges of C. Thus, the capacity k(C) in defined by :

$$k(C) = \sum_{c \in C} k(c)$$

The capacity of an S-D cut, $k(X, \bar{X})$ in the sum of all capacities of edges from X to $\bar{X}$. There may be edges from $\bar{X}$ to X but are not exerted into the computation of $k(X, \bar{X})$. We call an S-D cut $(X, \bar{X})$ a minimal cut if there b no S-D cut $(Y, \bar{Y})$ such that

$$k(Y, \bar{Y}) < k(X, \bar{X})$$

How to compute minimal cut from max flow

Let V_S be the set of verties reached by augmenting path from the source S, and v_D is the set of remaining vertices, then the cut (v_S, v_D) is the minimal cut.

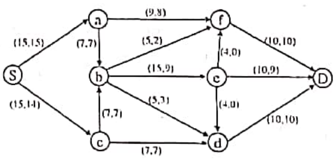
One very simple but inefficient way to find the minimum cut is to simply list out all possible cuts and select the smallest. However, the number of possible cuts is extremely large, it is impossible to list all possible cuts in a network.

A better approach is to make use of max flow min-cut theorem.

The minimum cut is actually simple to find after max flow is cut by Ford-Folkerson algo.

Simply marks the edges that one carrying a flow equal to their capacity and look for a cut that consists only of marked edge and no other edges.

Example 10



Theorem (Max. flow min cut)

Statement: In any flow (transport) network, the value of any maximal flow is equal to the capacity of a minimal cut.

Proof: We know that for a given flow network with flow F and capacity of cut K, we have

value of F $\le$ capacity of k

Now, optimizing this flow network such that there are no F-augmenting paths, we have

value of F = capacity of cut k

Let F^* be maximum flow and k^* be minimum cut for the network. Then for some flow F and cut k, we have

capacity of k = value of F $\le$ capacity of k^*

But no cut can have capacity less than minimum cut, we have

value of F = capacity of k^*

Also, we have

value of $F^* \le$ capacity of k^* = value of F

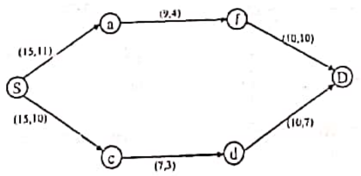
But no flow can be greater than the maximum flow.

$\therefore$ Value of F^* = Capacity of k^*

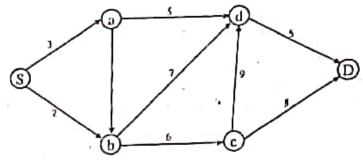
Hence proved.

EXERCISE

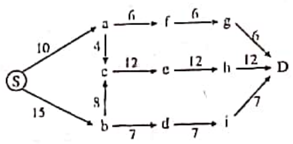
1. Define cut and capacity of cut.
2. Define maximum flow for the transport network.
3. What is augmenting path?
4. What is maximum flow problem? Illustrate it with suitable example.
5.
5. Find all possible S-D cut for the following network.



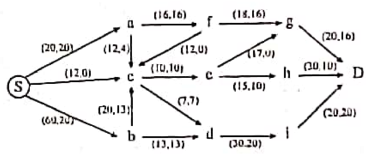
6. Find the maximum flow in the given figure.



7. Find the maximum flow for the given railway network.



8. Find the maximum flow for the given network.



■