Proof and Induction

classmate
Date
Page

Logic :-

Logic is the tool or language for reasoning about the truth and false of statement.

-> Logic helps us to reason about the mathematical model for solving ~~or~~ the problems of computer science.

-> Simply, logic is the generation of idea for solving problem.

-> Logic is the study or the process of reasoning

- Main reason behind the development of logic

(i) To explore the depth upto which the statement explains.
(ii) To direct the nature of truth.

Types of logic

(i) Propositional logic
(ii) Predicate logic
(iii) Fuzzy logic

Propositions

Propositions are the statement that are either true or false but not both.

→ In mathematical modelling, propositions are denoted by alphabets like p, q, r, s and so on.

→ E.g :-

Pokhara lies in Lalitpur district (F)

$2 + 10 = 12$ (T)

$x + 4 = 7$ (not proposition)

→ The truth value statement is denoted by 'T' and the false value is denoted by 'F'.

Propositional logic

→ logic that deals with propositions are called propositional logic.

→ Proposional logic are sometime called as propositional calculus.

Types of propositions

(a) Simple propositions

(b) Compound propositions

classmate

Date
Page

→ Simple propositions are those which cannot be break down into atomic sentences.

→ When two or more proposition are combined using some operator thus the resulting proposition are called compound propositions

→ Logical connectives are those operator that combines the simple proposition so form compound propositions.

Different connectives are:-

(i) Negation ($\neg$)
(ii) Conjunction ($\wedge$)
(iii) Disjunction ($\vee$)
(iv) Implication ($\rightarrow$)
(v) Double implication ($\leftrightarrow$)

Basic connectives along with their their truth table

(a) Negation ($\neg$)

→ If p is the proposition then the negation of p is denoted as $\neg p$ and read as "not p"

→ for e.g. P: It is raining

Then the negation can be

It is not raining

or

It is not the case that it is raining

Disjunction

→ Let p and q be two proposition then disjunction of p and q is denoted by p∨q and read as "p or q".

→ The truth ~~table~~ value of disjunction is true if any one of constituent proposition is ~~false~~ true and is false if all the proposition is false.

Truth table

p	q	p∨q
T	T	T
T	F	T
F	T	T
F	F	F

Conjunction

→ The conjunction of two proposition p & q is denoted by p∧q and read "p and q".

→ The truth value of conjunction is true if all constituent proposition is true otherwise false.

P.T.O

Truth table

p	q	p∧q
T	T	T
T	F	F
F	T	F
F	F	F

Implification

→ Implification is a conditional statement.

→ If p and q are two proposition then the implification is denoted by "p→q" as read as:

* if 'p' then 'q'
* 'p' inplies 'q'
* if 'p', 'q'.
* 'p is sufficient for q"

→ The implification statement follows the if then rules where if part is said to be hypothesis and then part is the consequence or cónclusion.

→ The basic idea of implification is that the true hypothesis always lead to the true conclusion, but a wrong hyporhesis never lead to correct conclusion.

Truth table

p	q	p → q
T	T	T
T	F	F
F	T	T
F	F	T

p → hypothesis, q → conclusion

Double implification

→ The double implification of p and q is denoted by "$p \Leftrightarrow q$" and read as "p if and only if q".

→ The truth value of double implification is true when all the constituent proposition have same value otherwise false.

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

→ The double implification statement is true if both hypothesis and conclusion is true.

Types of compound proposition

1. Tautology
2. contradiction
3. contingency

Tautology

→ The compound proposition is said to be tautology if all the values of the truth table is true, whatever the constituent proposition may holds.

→ E.g:- $p \to (p \vee q)$

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

show that

(a) $[p \wedge (p \to q)] \to q$ is a tautology

p	q	$p \to q$	$p \wedge (p \to q)$	$\{p \wedge (p \to q)\} \to q$
T	T	T	T	T
T	F	F	F	T
F	T	T	F	T
F	F	T	F	T

(b) $[(p \to q) \wedge (q \to r)] \to (p \to r)$

~~p~~	q
T	T
T	F
F	T
F	F

p	q	r	$p \to q$	$q \to r$	$(p \to q) \wedge (q \to r)$	$p \to r$ ~~$[(p \to q) \wedge (q \to r)] \to$~~	Result
T	T	T	T	T	T	T	T
T	T	F	T	F	F	F	T
T	F	T	F	T	F	T	T
T	F	F	F	T	F	F	T
F	T	T	T	T	T	T	T
F	T	F	T	F	F	T	T
F	F	T	T	T	T	T	T
F	F	F	T	T	T	T	T

Contradiction

→ The negation of the tautology is contradiction

→ If all the interpretation of a compound proposition results a false value then the compound statement is said to be contradiction

E.g.: $\neg(p \to (p \vee q))$ is a contradiction

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

Contingency

→ If the intreperation of truth table contains the combination of true and ~~table~~ false value then the compound proposition is said to be contingency

→ E.g $p \leftrightarrow q$ is a contingency.

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

Converse, Inverse and contrapositive

→ We know that, the conditional statement consist of hypothesis and conclusion.

→ When the position of hypothesis is changed, negated or both a new compound statement is formed.

Converse

→ If $p \to q$ is a implification statement the its converse is denoted by $q \to p$.

Inverse

→ For $p \to q$, the inverse is written as $\neg p \to \neg q$.

Contrapositive

→ If the implication of p and q is denoted by $p \to q$ then its contrapositive is denoted by

$$\neg q \to \neg p$$

If she smiles, she is happy is a implifation statement, then what are converse, inverse and contrapositive?

Converse

If she is happy, she smiles

Inverse

If she does not smile, she is not happy

Contrapositive

If she is not happy, she does not smile

Logically equivalent

→ If p and q are two compound proposition, then the equivalent of p and q is denoted by

$$P \equiv Q$$

→ If the both compound proposition p and q have the identical values in the truth table, then the preposition are said to be logically equivalent.

show that implification and its contrapositive is logically equivalent.

Check for the logical equivalence for the following preposition.

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

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

So, from above touth table, the propositions are idential, hence, the given proposition are ~~equa~~ logically equivalance

Translating the sentences into the prepositional logic

(i) Identify all the individual sentences and assign alphabets for all the atomic sentences

(ii) Identify all the connectives used in the sentences

(iii) write the propositional logic using those alphabets and connectives used in the sentences

E.g:-

You can access the college internet only if you are a computer science student or you are not a fresher.

p: You can access the college internet

q: You are a computer science student

r: You are a fresher

<u>connectives</u>

only if (implification) $\rightarrow$

or (disjunction) $\vee$

not (negation) $\neg$

$\cancel{(p \rightarrow q) \vee (\neg r)}$ $\quad p \rightarrow (q \vee \neg r)$

p	q	r	$p \rightarrow q$	$\neg r$	$q \vee \neg r$	$\cancel{(p \rightarrow q) \vee (\neg r)}$	$p \rightarrow (q \vee \neg r)$
T	T	T	T	F	T	T	T
T	T	F	F	T	T	T	T
T	F	T	T	F	F	T	F
T	F	F	F	T	T	T	T
F	T	T	T	F	T	T	T
F	T	F	T	T	T	T	T
F	F	T	T	F	F	T	T
F	F	F	T	T	T	T	T

$p \to (q \wedge \neg r)$

classmate

Date

Page

Hiking is safe along the trail if and only if ~~se~~ Berries are ripe along the trail and bears have not been seen along the area

p: Hiking is safe along the trail

q: Berries are ripe along the trail

r: bears have seen along the area

Connectives

if and only if (double implification) $\leftrightarrow$

and (conjunction) $\wedge$

not (negation) $\neg$

$$p \leftrightarrow (q \wedge \neg r)$$

p q r

Predicate logic (First order propositional logic)

→ Predicate logic are those which are defined using some predicate.

→ The truth and false value of predicate statement are not declarative. For find the truth and false value we need to define the propositional function.

→ Eg :- $x > 5$ is the statement whose truth value cannot be declared easily.

→ The predicate of the given statement can be written as :

$P(x) : x > 5$ where 'p' is the predicate and $P(x)$ is a propositional function.

→ when the variable of the propositional function is substituted by only value, then the predicate becomes proposition.

→ $P(x) : x > 5$

$P(2) : 2 > 5$ is a proposition

$P(7) : 7 > 5$ is a proposition

Quantifier & Quantification

→ Quantifier are the tools that makes the propositional function into propositions.

→ construction of proposition from the predicate using the quantifier is called Quantification.

→ For this we need to define the domain the propositional function

→ simply, quantification is the process of finding the propositional function into some predefined domain values.

→ If $p(x)$ is a propositional function, where 'x' is a variable we need to substitude the value of x from some domain.

→ If 'D' is any set for $p(x)$, then we can say that 'p' is a predicate with respect to 'D' and for each value of 'D' $p(x)$ is a proposition.

→ The set of value 'D' is known as Discourse for p.

Types of Qualifier
(i) Universal Quantifier ($\forall$)
(ii) Existential quantifier ($\exists$)

(i) Universal quantifier ($\forall$) :-

→ The universal quantifier for a predicate $P(x)$ is denoted by ~~$\forall x$~~ $\forall_x P(x)$ and read as "for all x $P(x)$" or "for each x $P(x)$ holds"

→ The universal quantifier are used for universal quantification.

→ The universally quantified statement is true if all the values in the universal set holds true.

→ Let $P(x)$ be the predicate then truth value of $\forall_x P(x)$ is true if $P(x)$ is true for all the values in the given domain and is False if $P(x)$ is false for any value in the domain.

→ Let $P(x_1, x_2, \ldots, x_n)$ be a predicate such that $P(x_1), P(x_2), \ldots, P(x_n)$ are its different instances
Then, $\forall_x P(x_1, x_2, \ldots, x_n)$ is ~~the~~ true if all the above instances are true and false if at least one instance is false

i.e.

$$\forall_x P(x_1, x_2, \ldots x_n) \equiv P(x_1) \land P(x_2) \land \ldots \land P(x_n)$$

Eg:-

$x^2 - 1 > 0$ for $x \in R$

$P(x)$: $x^2 - 1 > 0$, $x^2 - 1 > 0$

$x = 1$, $0 > 0$ (false)

$x = 2$ $x^2 - 1 > 0$

$3 > 0$ (True)

$x = 3$, $x^2 - 1 > 0$

$8 > 0$ (True)

The given statement is not true for all the values i.e. $x = 1$ it is false.

so, $\forall_x P(x)$ is false.

→ Here, $x = 1$ is a counter example that makes the universally quantified statement false.

Existential Quantifier

→ If $P(x)$ is a predicate, then the existential quantification of $P(x)$ is denoted by $\exists_x P(x)$ and read as "for some x $P(x)$" or "there exist x $P(x)$".

→ The truth value of $\exists_x P(x)$ is true if

Scanned by CamScanner

$P(x)$ is true for at least one value of the given domain is true and is false if $P(x)$ is false for each value of the given domain.

→ Let $P(x_1, x_2, x_3, \ldots, x_n)$ be a predicate such that $P(x_1), P(x_2), P(x_3), \ldots, P(x_n)$ are the different instances then

$\exists_n P(x_1, x_2, \ldots, x_n)$ is true if at least one of the above instance is true and is false if all the above instances are false.

$$\exists_x P(x_1, x_2, \ldots x_n) \equiv P(x_1) \vee P(x_2) \ldots \vee P(x_n)$$

Translating the sentences into statement of predicate logic.

a) Every ~~element~~ student ~~the~~ in this class studies MFCS

b) some student in this class likes MFCS

c) All King are men

d) Some lions are dangerous.

a) Ans

$s(x)$: x is student of this class

$M(x)$: x studies MFCS

$$\forall x\,[s(x) \rightarrow M(x)]$$

b)

$s(x)$: x is student of this class

$M(x)$: ~~x~~ student likes MFCS

$$\exists x\,[s(x) \wedge M(x)]$$

c)

$K(x)$: x is King

$M(x)$: x is men

$$\forall x\,[K(x) \rightarrow M(x)]$$

d)

$L(x)$: x is ~~a~~ lion

$M(x)$: x lions are dangorous

$$\exists x\,[L(x) \wedge M(x)]$$

Nested Quantification

→ Quantification within a quantification.

→ Let $P(x,y)$ be a predicate then

(i) $\forall x \forall y\, P(x,y)$ is true if $P(x,y)$ is true for every value of x and ~~at least~~ y ~~one value of "y"~~ otherwise false.

(ii) $\forall x \exists y\, P(x,y)$ is true for every value of x and at least one value of 'y' otherwise false.

(iii) $\exists x \forall y\, P(x,y)$ is true if $P(x,y)$ is true for at least one value of x and for every value of y otherwise false.

(iv) $\exists x \exists y\, P(x,y)$ is true if $P(x,y)$ is true for some value of x and y otherwise false.

E.g:-

If $P(x,y): x+y=0 \quad x,y \in R$

(i) $\forall x \forall y\, P(x,y) =$ false

(ii) $\forall x \exists y\, P(x,y) =$ True

(iii) $\exists x \forall y\, P(x,y) =$ False

(iv) $\exists x \exists y\, P(x,y) =$ True

Elementory stepwise induction and complete induction

→ It is also called well ordering principles.
→ sometimes also called as proof by mathematical induction.

→ Let $P(n)$ be a statement. Now our concern is to show that $P(n)$ is ~~time~~ true using mathematical induction

→ For this we first show that $P(n)$ is true for some initial value like $n = 0, 1, 2, \ldots$. This step is called basis step.

→ Then we assume that $P(n)$ is true for any arbitary value 'K' i.e. $P(K)$ is true and show that $P(n)$ is true for '$K+1$' i.e $P(K+1)$ is true. This step is called inductive step.

Thus mathematical induction can be defined as

$$\underbrace{P(1)}_{\text{Basis step}} \wedge \underbrace{[P(K) \rightarrow P(K+1)]}_{\text{Inductive step}} \rightarrow P(n)$$

Using mathematical induction show that

$$1 + 2 + 3 + \dots + n = \frac{n(n+1)}{2}, \quad n \geq 1$$

sol^n

Basis step, for $n = 1$

$$P(n) = 1 + 2 + 3 \dots + n = \frac{n(n+1)}{2}$$

$$\frac{n(n+1)}{2}$$

$$\frac{1(1+1)}{2} = 2/2 = 1$$

$\therefore P(1)$ is true.

For inductive step

we assume that the $p(n)$ is true for some arbitary k i.e.

$p(k)$ is true

i.e $\frac{k(k+1)}{2}$

Now, we try to proves for $k+1$ i.e

$$P(k+1) = 1 + 2 + 3 + 4 + \dots + k + k + 1 = \frac{(k+1)\{(k+1)+1\}}{2}$$

$$= \frac{k(k+1)}{2} + (k+1)$$

$$= (k+1)\left\{\frac{k}{2} + 1\right\}$$

$$= (k+1)\left\{\frac{k+2}{2}\right\}$$

$$= (k+1)\left\{\frac{(k+1)+1}{2}\right\}$$

Therefore, the given formula is valid for mathematical induction.

Show that

$2+2^2+2^3+\cdots+2^n = 2^{n+1}-2$ using mathematical induction.

Soln

Basis step. for $n=1$

$$P(n) = 2+2^2+2^3+\cdots+2^n = 2^{n+1}-2$$

$$2^{n+1}-2$$

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

$$4-2 = 2$$

$P(1)$ is true

For Inductive step

We assume that $P(k)$ is true for any arbitrary k

$P(k)$ is true

i.e $2^{k+1}-2$

Now, we try to proves for $k+1$

$P(k+1) = 2+2^2+2^3+\dots+2^k+2^{k+1} = 2^{(k+1)+1}-2$

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

$\Rightarrow 2^{k+1}(1+1)-2$

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

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

Therefore, $2+2^2+2^3+\dots+2^n = 2^{n+1}-2$ is valid for mathematical induction.

show that 8^n-3^n is divisible by 5 for $n\geq 1$ using mathematical induction.

Sol^n

Basis step: for $n=1$

$P(n) = 8^n-3^n$

$8^1-3^1 = 5$

$P(1)$ is true

For Inductive step

We assume that $P(k)$ is true for any arbitary k.

$P(k)$ is true

i.e 8^k-3^k

Now, we try to proves for $k+1$.

$$P(k+1) = 8^{k+1} - 3^{k+1}$$
$$= 8^k \cdot 8^1 - 3^k \cdot 3$$
$$= 8^k (5+3) - 3^k \cdot 3$$
$$= 8^k \cdot 5 + 3 \cdot 8^k - 3^k \cdot 3$$
$$= 8^k \cdot 5 + 3(8^k - 3^k)$$

Here, $5 \cdot 8^k$ is multiple of 5, so it is divisible by 5 and we assumed that $8^k - 3^k$ is true and it product by any constant is ~~also~~ also divisible by 5.
Hence, $8^n - 3^n$ is divisible by 5 is valid for mathematical induction.

or.

since $8^k \cdot 5$ is divisible by 5, from our assumption $(8^k - 3^k)$ is also divisible by 5. Two individual no. divisible by 5 when added is also divisible by 5.
So, we can say that $p(n) = 8^n - 3^n$ is divisible by 5 using mathematical induction.

Unit 3

classmate

Date
Page

Introduction to Mathematical Reasoning

→ Any of the mathematical statements are supported by arguments that makes it correct.

→ For this we need to know different techniques and rules that can be applied in the mathematical statements so we can prove that correctness of given mathematical statements.

→ This method of understanding the the correctness by sequence of statement forming an argument is a proof of statement.

Axioms :

→ Axioms are the assumption about the mathematical instructions.

→ They are the hypothesis of the theorem to be proved and previously proved theorem.

Rules of Inference

→ To draw the conclusion from the given statements we must be able to supply well defined statements that helps reaching the conclusion.

→ The steps for reaching the conclusion are provided by rules of inference.

Inference

→ The process of drawing a conclusion from the given facts using certain valid rules is called inference.

V.V.I

Rules of Inference in propositional logic

(1) Modus Ponens :-

→ whenever the proposition p and p→q are true, then we confirm that q is true.

i.e

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

(ii) Modus Tollens :-

→ when two proposition p→q and ¬q is true then ¬p is also true.

i.e.

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

(iii) Hypothetical syllogism

$$\begin{array}{c} p \to q \\ q \to r \\ \hline \therefore p \to r \end{array}$$

(iv) Disjunctive syllogism

$$\begin{array}{c} p \lor q \\ \neg p \\ \hline \therefore q \end{array}$$

(v) Additive rule

$$\begin{array}{c} p \\ \hline \therefore p \lor q \end{array}$$

(vi) simplification

$$\begin{array}{c} p \land q \\ \hline \therefore p, q \end{array}$$

(vii) conjunction

$$\begin{array}{c} p \\ q \\ \hline p \land q \end{array}$$

Date
Page

(viii) Resolution

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

Using rules of inference show that the following hypothesis

(a) It is not sunny this afternoon and it is colder than yesterday

b) We will go swimming only if it is sunny

c) If we didn't go for swimming, then we will go for a canoe trip.

(d) If we go for a canoe trip, then we will be home by sunset.

Leads to the conclusion,
we will be home by sunset

sol^n

Identifying the individual sentences

a: It is sunny this ~~af~~ afternoon
b: It is cooler than yesterday

c: we will go swimming.
d: we will go for a canoe trip.
e: we will be home by sunset

writing the given statements into propositional logic statements.

Hypothesis

(i) $\neg a \wedge b$
(ii) $c \rightarrow a$
(iii) $\neg c \rightarrow d$
(iv) $d \rightarrow e$

conclusion

e

Proof:

steps	Reasons
(i) $\neg a \wedge b$	(i) Given hypothesis
(ii) $\neg a$	(ii) using simplification on hypothesis (i)
(iii) $c \rightarrow a$	(iii) Given hypothesis
(iv) $\neg c$	(iv) using modus tollens on hypothesis (ii) & (iii)
(v) ~~Giv~~ $\neg c \rightarrow d$	(v) Given hypothesis

vi) d — (vi) using modus ponens on hypothesis (iv) & (v)

vii) $d \to e$ — (vii) Given hypothesis

viii) e — (viii) using modus ponens on (vi) & (vii)

Using rules of inference show that the following hypothesis

a) If you send me an e-mail message, then I will finish writing the program.

b) If you don't send be an email message, then I will go to sleep early.

c) If I go to sleep early then I will wake up feeling refreshed.

Leads to conclusion
If I don't finish writing the program then I will wake up feeling refreshed.

sol^n

Identifying the individual sentences

a: you send me an email message.

b: I will finish writing the program.
c: I will go to sleep early
d: I will wake up feeling refreshed

writing the given statement into propositional logic statement

hypothesis

(i) $a \rightarrow b$

(ii) $\neg a \rightarrow c$

(iii) $c \rightarrow d$

conclusion

$\neg b \rightarrow d$

Proof

steps	Reasons
i) $a \to b$	i) Given hypothesis
ii) $\neg b \to \neg a$	ii) Applying contraposition on hypothesis (i)
iii) $\neg a \to c$	iii) Given hypothesis
iv) $\neg b \to c$	iv) using ~~modus ponens~~ hypothetical syllogism on hypothesis (ii) & (iii)
v) $c \to d$	v) Given hypothesis
vi) $\neg b \to d$	vi) Applying hypothetical syllogism on hypothesis (iv) & (v)

Q. show that the following hypothesis

a) If today is tuesday, I have a test in mathematics or economics

b) If my economic professor is sick, I will not have a test in economics.

c) Today is tuesday and my economics professor is sick

Leads to the conclusion

I have a test in mathematics

son

Identifying the individual sentences

a: today is tuesday

b: I have a test in mathematics

c: I have a test in economics

d: my economics professor is sick

writing the given statement int propositional logic statement

Hypothesis

(i) $a \to [b \vee c]$

(ii) $d \to \neg c$

(iii) $a \wedge d$

conclusion

b

$a \to T$

$d \to T$

$b \vee c$

$\neg c$

b

Proof:-

steps

~~(i) $a \to [b \vee c)$~~

~~(ii) $\neg a \vee (b \vee c)$~~

Reasons

~~(i) Given hypothesis~~

(ii) logically equivallent

steps	Reasons
(i) $a \wedge d$	(i) Given hypothesis
(ii) a, d	(ii) using symplification on (i)
(iii) $d \to \neg c$	(iii) Given
(iv) $\neg c$	(iv) Using modus ponens in (ii) and (iii)
(v) $a \to [b \vee c]$	(v) Given
(vi) $b \vee c$	(vi) using modus ponen in (ii) and (v)
(vii) b	(vii) Using disjunctive syllogism (vi) and (iv)

Rules of inference for quantified statement

① **Universal Insantiation :-**

$\forall_x P(x)$
$\therefore P(d)$
where 'd' is the value from universe of discourse.

Eg:- All birds can fly

Assuming
$B(x)$: x is a bird
$F(x)$: x can fly
Then $\forall_x B(x) \rightarrow F(x)$
From this we can draw a conclusion
$B(sparrow) \rightarrow F(sparrow)$

(ii) **Universal Generalization**
· $P(d)$ is true where 'd' is the universe of discourse, then,
$\forall_x P(x)$ is true.

(iii) **Existential Instantiation**
$\exists x\, P(x)$
$\therefore P(d)$ is true

(iv) **Existential Generalization**
$P(d) \in D$, D is universe of discourse
$\therefore \exists x\, P(x)$

Proof in quantified statement

→ proof in quantified statements are complex than propositional ~~log~~ one.

→ we cannot apply the rules of inference for quantified statement directly.

→ The steps to apply the rules for inference for quantified statements are:

(i) Apply the rules of instantiation for quantified statement, so that sentences becomes proposition.

(ii) Apply the rules of inferences of propositional logic.

(iii) Finally apply the generalization rule to convert the propositional logic back to quantified statement.

predicate (quantified statement)

↓

Rules of Instantiation

↓

proposition

↓

Apply rules of proposition

↓

Rules of Generalization

↓

predicate (quantified statement)

2015 spring

2(a) We are given a hypothesis

(a) Everyone loves either Microsoft or Apple.

(b) ~~Do~~ Lynn does not love Microsoft.

show that the conclusion

Lynn loves Apple

sol^n

Defining prediction,

$M(x)$: x loves Microsoft

$A(x)$: x loves Apple.

Hypothesis

(i) $\forall x\,[M(x) \cup A(x)]$

(ii) $\neg M(\text{Lynn})$

conclusion

$A(\text{Lynn})$

Proof

steps	Reasons
(i) $\forall x\,[M(x) \vee A(x)]$	Given hypothesis
(ii) $M(\text{Lynn}) \cup A(\text{Lynn})$	Using universal instantiation hypothesis (i) $P(d) \in D$, d: lynn
(iii) $\neg M(\text{Lynn})$	Given hypothesis
(iv) $A(\text{Lynn})$	Applying disjunctive syllogism on hypothesis (ii) & (iii)

Introduction to proof

→ An argument used to establish the truth of mathematical statement is called proof.

→ while establishing the truth, different rules and already proven facts are used.

→ Proof can be
- (i) Formal proof
- (ii) Informal proof
  - → Direct proof
  - → Indirect proof

→ Formal proof is a technique where predefined rules and steps are used to show that given statement is true.

→ In informal proof such pre-defined rules and steps may not be used.

→ Further classification of Informal proof
- (a) Direct proof
- (b) indirect proof

(a) Direct proof :-

IF $p \Rightarrow q$ be an implification, in direct proof we assume that hypothesis is true i.e 'p' is true. Then by using different theorem and already proven facts, we conclude that conclusion is also true i.e. q is true.

→ The idea behind the direct proof is that true hypothesis leads to true conclusion.

Using direct proof, show that if 'n' is odd then n^2 is odd.

Sol^n

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

p: 'n' is odd

q: n^2 is odd

$p \to q$

In direct proof, we assume that hypothesis is true i.e. we assume that 'n' is odd.

Now, by definition of odd number

$n = 2k+1 \quad [k = 0, 1, 2, \ldots]$

Squaring both side

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

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

Here,

$4k^2 + 4k + 1$

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

$= 2k' + 1 \quad [k' = 2k^2 + 2k]$

for any value the above value is odd

$\therefore n^2$ is also odd.

Hence, we can say that the conclusion n^2 is odd is true and the assumption of 'n' is odd is also true. so whenever 'n' is odd n^2 is odd is proved using direct proof.

The sum of two rational number is rational: using direct proof.

Sol

Let p and q be any two rational numbers and let $M = p+q$

Then their exists integers a, b, c and d such that $p = a/b$ and $q = c/d$.

so, $M = a/b + c/d$

$$M = \frac{ad+bc}{bd}$$

since a, b, c, and d are integers, (ad+bc) and (bd) are integers. since b and d are non-zero

$\therefore M$ is a rational number.

so, the sum of any two rational numbers is also a rational number.

Prove by direct proof, show that if $(3n+2)$ is odd then n is odd

<u>sol</u>

If $3n+2$ is odd, then ~~$3n+2$~~ n is odd

p: $(3n+2)$ is odd

q: n is odd

$$p \rightarrow q$$

In direct proof, we assume that hypothesis is true i.e. we assume that $3n+2$ is odd by definition of odd number,

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

$\boxed{n = \frac{2k-1}{3}}$ ← dead end of proof.

Indirect proof :-

→ for certain case, the direct proof may not be appropriate

→ for e.g :- if (3n+2) is odd, then 'n' is odd.

→ Here, using the direct proof, we may not reach the conclusion such problem is considered as "dead end" of proof.

→ To overcome such problem, we can use indirect proof.

Two types of indirect proof are :-
(a) proof by contradiction
(b) proof by contrapositive

$p \to q \equiv \neg q \to \neg p$

Proof by contrapositive

→ Let $p \to q$ is an implication, then in proof by contrapositive, we assume that the negation of conclusion is true
i.e $\neg q$ is true.

→ Then by using different theorem and already proven facts we conclude that negation of hypothesis is also true i.e $\neg p$ is true.

→ The idea behind the proof by contrapositive is the negation of conclusion leads to negation of hypothesis and implication is also true.

→ It is because, the implication and its contrapositive are logically equivalent.

Using proof by contrapositive shows that if $(3n+2)$ is odd then 'n' is odd.

soln

If $3n+2$ is odd then n is odd

p : $3n+2$ is odd

q : n is odd

$$p \to q \equiv \neg q \to \neg p$$

we assume that negation of conclusion is true
i.e. n is even.
~~Now, by definition of even.~~

$\therefore \quad n = 2k$

$3n+2 = 2k+2$

$3n+2 = n+2$

Now, we try to prove ¬p is also true
i.e. $(3n+2)$ is even

By definition of even number
$n = 2k$

Now,
$3(2k)+2$
$(6k+2)$ which is even.

$\therefore$ $(3n+2)$ is even.
i.e. negation of hypothesis is also true.
Here, the negation of conclusion leads to negation of hypothesis. So we conclude that when $(3n+2)$ is odd then 'n' is odd by using proof by contraposition.

Proof by contradiction

→ It is also the process of proof by indirect method.

→ For a proof by contradiction, there may arise following cases.

(a) The given statement is an implification.

→ For implification statement we assume the negation of conclusion and hypothesis is true i.e $\neg q \wedge p$

→ Then by using different theorem and already proven facts we drive that negation of hypothesis is true i.e. $\neg p$ is true.

→ This is contradiction to our assumption.

Hence, we can say that our assumption was wrong and given statement is true.

(b) Given statement is not an implification

→ for this we assume that negation of the given statement is true. Then after processing we reach to the point that contradicts our assumption.

→ Hence, we say that our assumption was wrong and given statement is true.

This implies q^2 is also even.
Thus p and q are divisible by 2, so, the p and q cannot form a rational number. Thus our assumption of $\sqrt{2}$ is rational is false i.e. ~~con~~ contradicts to our assumption.

→ so our assumption was wrong i.e. $\sqrt{2}$ is not rational.

Unit 4

Recurrence Relation

→ Let $a_0, a_1, a_2, \ldots, a_n$ be the 'n' terms of a sequence a_n.

→ Then a_n is said to be recurrence relation if a_n can be expressed by an equation in terms of it's previous elements.

For e.g:- $a_n = 5a_{n-1} + 6a_{n-2}$ is a recurrence relation.

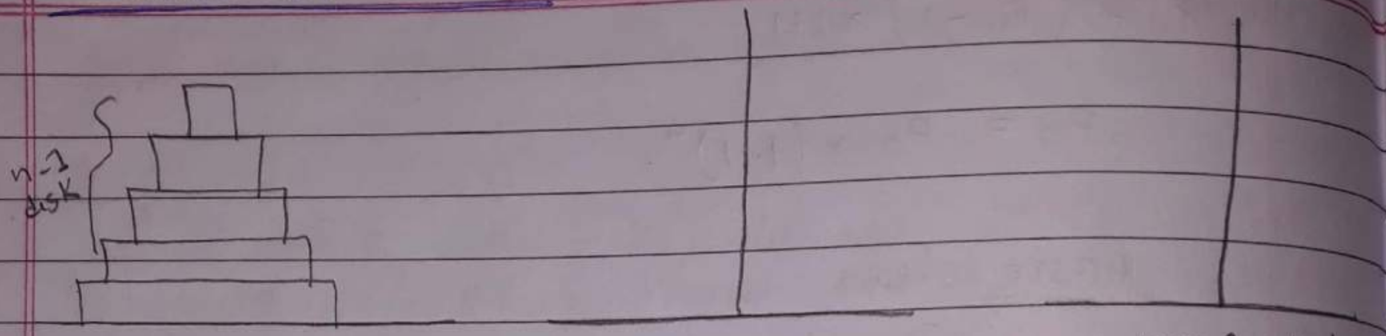
→ consider a sequence
$5, 8, 11, 14, \ldots$
Here, the first term of sequence $a_1 = 5$ and we can define the sequence as
$a_n = a_{n-1} + 3$ for $n \geq 2$.

→ This equation is the recurrence definition because it defines the term of sequence in reference to its previous time.

→ For every recurrence relation there must be some initial condition.
e.g: $a_0 = 5$, $a_1 = 7$, etc.

disk odd - destination
even - temporary

Tower of Hanoi



peg 3 : destination

page 1 : Source

peg 2 :
Temporary

-> Let H_n be the moves required to move 'n' disks from source (~~pag~~ peg 1) to destination (peg 3).

-> first of all with the help of peg 2 and peg 3, (n-1) disks from source is arranged to temporary (peg 2). This requires H_{n-1} moves

-> Then the largest disk from peg-1 is moved to peg 3, which requires '1' move.

-> Finally (n-1) disk from peg 2 are moved to peg 3, with the help of peg 1 & peg 2, which further requires H_{n-1} moves.

-> Hence we can define the recursive relation as:

$$H_n = H_{n-1} + 1 + H_{n-1}$$

$$\boxed{H_n = 2H_{n-1} + 1}$$

which is required recurrence relation.

P.T.O

Now,

$H_n = 2H_{n-1} + 1$

$2(2H_{n-2} + 1) + 1$

$= 2(2H_{n-2} + 1) + 1$

$= 2(2(2H_{n-3} + 1) + 1) + 1$

$= 2(4H_{n-3} + 2) + 2 + 1$

$= 2(4(2H_{n-4} + 1) + 2) + 1$

$= 2(8H_{n-4} + 4 + 2) + 1$

$= 2(2^3 H_{n-4} + 2^2 + 2^1 + 2^0$

For n disks

$2(2^{n-1}.1 + 2^{n-2} + \cdots + 2^3 + 2^2 + 2^1 + 2^0)$

common ratio $= \frac{2^1}{2^0} = 2$

This relation satisfies the geometric series with common ratio 2.

sum of G.S $= \frac{a(r^n - 1)}{r - 1}$

$= \frac{1(2^n - 1)}{2 - 1}$

$= 2^n - 1$

Types of Recurrence Relation

(i) Linear Homogenous recurrence relation

(ii) Linear non-homogenous recurrence relation.

(i) Linear Homogenous recurrence relation:-

→ A recurrence relation of the form

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \cdots c_k a_{n-k}$$

is called linear homogenous relation of degree k' where $c_1, c_2, \ldots c_k$ are constant and $c_k \neq 0$.

Eg:-

$a_n = 5a_{n-1} - 6a_{n-2}$ is linear homogenous recurrence relation of degree 2.

Solution of linear homogenous recurrence relation

Let $a_n = c_1 a_{n-1} + c_2 a_{n-2} + \cdots c_k a_{n-k}$ — (i)

is linear homogenous recurrence relation of degree 'k'.

→ A sequence $a_n = r^n$ is said to be it's solution, if it is satisfies the given recurrence relation.

$$r^n = c_1 r^{n-1} + c_2 r^{n-2} + \cdots + c_k r^{n-k} \quad \text{(ii)}$$

Dividing both side of eq^n (ii) by r^{n-k}

$$\frac{r^n}{r^{n-k}} = \frac{c_1 \cdot r^{n-1} + c_2 r^{n-2} + \cdots + c_k \cdot r^{n-k}}{r^{n-k}}$$

or, $r^k = c_1 r^{k-1} + c_2 \cdot r^{k-2} + \cdots + c_k$

or, $r^k - c_1 r^{k-1} - c_2 r^{k-2} + \cdots \; c_k = 0$ — (iii)

eq^n (iii) is called characteristics equation of degree k.

The roots of this equation are called characteristics root.

Eg.

$a_n = 6a_{n-1} + 10a_{n-2}$

$$\boxed{a_n = r^n}$$

$$\frac{r^n}{r^{n-2}} = \frac{6r^{n-1} + 10r^{n-2}}{r^{n-2}}$$

$$r^2 = 6r + 10$$

$$r^2 - 6r - 10 = 0 \text{ — (a)}$$

The roots of the characteristics equation can be same or ~~me~~ different.

Theorem

If the characteristic eqn of the linear homogenous recurrence relation of the form

$a_n = c_1 a_{n-1} + c_2 a_{n-2}$ of degree 2

i.e $r^2 - c_1 r - c_2 = 0$ and roots are distinct i.e r_1 & r_2 then, solution will be of form

$$\boxed{a_n = \alpha_1 r_1^n + \alpha_2 r_2^n}$$

→ If roots are same then, solution will be of form

$$\boxed{a_n = \alpha_1 r_1^n + n\alpha_2 r_1^n}$$

Derive the solution for recursive relation

$a_n = 5a_{n-1} - 6a_{n-2}$ with $a_0 = 3$ & $a_1 = 5$

→

$a_n = 5a_{n-1} - 6a_{n-2}$

The corresponding characteristics equation is

$r^2 = 5r - 6$

$r^2 - 5r + 6 = 0$ — (I)

The roots of eqⁿ (I) is

$r^2 - 2r - 3r + 6 = 0$

$r(r-2) - 3(r-2) = 0$

$(r-2)(r-3) = 0$

$r = 2$

$r = 3$

Here roots are distinct as the solution is of the form

$a_n = \alpha_1 r_1^n + \alpha_2 r_2^n$

$a_n = \alpha_1 2^n + \alpha_2 3^n$ — (II)

when $n=0$

$a_0 = \alpha_1 2^0 + \alpha_2 3^0$

$3 = \alpha_1 + \alpha_2$ — (III)

when $n=1$

$a_1 = \alpha_1 2^1 + \alpha_2 3^1$

$5 = 2\alpha_1 + 3\alpha_2$ — (IV)

$\alpha_1 = 4$

$\alpha_2 = -1$

Scanned by CamScanner

Substituting the value of α_1 & α_2 in eqn (ii)

$\boxed{a_n = 4 \cdot 2^n - 3^n}$ is the required soln

Q] Derive the soln for recurrence relation

$a_n = 6a_{n-1} - 9a_{n-2}$ with $a_0 = 5$ & $a_1 = 7$

$$a_n = 6a_{n-1} - 9a_{n-2}$$

The corresponding characteristics equation is

$$r^2 = 6r - 9$$

$$r^2 - 6r + 9 = 0$$

$$(r-3)(r-3) = 0$$

$$r = (3, 3) \quad (1)$$

Here the roots are same so soln in the form

$$\boxed{a_n = \alpha_1 3^n + n\alpha_2 3^n}$$

when $n = 0$

$$a_0 = \alpha_1 3^0 + 0 \times \alpha_2 \times 3^0$$

$$5 = \alpha_1$$

when $n = 1$

$$a_1 = \alpha_1 \cdot 3^1 + 1 \times \alpha_2 \times 3^1$$

$$7 = \alpha_1 \cdot 3 + \alpha_2 \cdot 3$$

$$7 = 5 \times 3 + \alpha_2 \times 3$$

$$7 = 15 + \alpha_2 \times 3$$

$$7 - 15 = \alpha_2 \times 3$$

$$-\frac{8}{3} = \alpha_2$$

$$\boxed{a_n = 5 \cdot 3^n - \frac{8}{3} n \cdot 3^n}$$

(3) Derive the explicit formula for the fibonacci series

$$fib(n) = \begin{cases} n & \text{if } n<2 \\ fib(n-1) + fib(n-2) \end{cases}$$

The recurrence relation of the fibonacci series

$$a_n = a_{n-1} + a_{n-2} \quad \text{with } a_0 = 0 \quad a_1 = 1$$

The corresponding characteristics equation is

$$r^2 - r - 1 = 0$$

On solving,

$$r = \frac{1+\sqrt{5}}{2} \qquad r = \frac{1-\sqrt{5}}{2}$$

when roots are not same the soln in the root

$$a_n = \alpha_1 \left(\frac{1+\sqrt{5}}{2}\right)^n + \alpha_2 \left(\frac{1-\sqrt{5}}{2}\right)^n \quad \text{— (A)}$$

When $n=0$

$$a_0 = \alpha_1 \left(\frac{1+\sqrt{5}}{2}\right) + \alpha_2 \left(\frac{1-\sqrt{5}}{2}\right)$$

$$0 = \alpha_1 + \alpha_2 \quad \text{— (1)}$$

$$\alpha_1 = -\alpha_2$$

when $n=1$

Scanned by CamScanner

$$a_1 = \alpha_1 \left(\frac{1+\sqrt{5}}{2}\right)^1 + \alpha_2 \left(\frac{1-\sqrt{5}}{2}\right)$$

$$1 = \alpha_1 \left(\frac{1+\sqrt{5}}{2}\right) + \alpha_2 \left(\frac{1-\sqrt{5}}{2}\right)$$

$$2 = \alpha_1 + \alpha_1\sqrt{5} + \alpha_2 - \sqrt{5}\alpha_2$$

$$2 = (1+\sqrt{5})\alpha_1 + (1-\sqrt{5})\alpha_2 \quad \cancel{(ii)}$$

$$2 = (1+\sqrt{5}) - \alpha_2 + (1-\sqrt{5})\alpha_2$$

$$2 = -\alpha_2 - \sqrt{5}\alpha_2 + \alpha_2 - \sqrt{5}\alpha_2$$

$$2 = -2\sqrt{5}\alpha_2$$

$$\alpha_2 = -\frac{1}{\sqrt{5}}$$

$$\alpha_1 = \frac{1}{\sqrt{5}}$$

putting the value in (A)

$$a_n = \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^n - \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^n$$

Linear non-homogenous Recurrence relation

→ A recurrence relation of the form

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_k a_{n-k} + f(n)$$

is said to be linear non-homogenous recurrence relation

→ Here $c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_k a_{n-k}$ is called associated part of $f(n)$ is called non-homogenous part.

→ the soln of associated part is called general soln / G.S & solution of non-homogenous part is particular solution (P.S)

→ The overall soln of linear non homogenous recurrence relation is the sum of general soln & particular soln.

$a_n = 5a_{n-1} - 6a_{n-2} + 7^n$ is a linear non-homogenous recurrence relation

Q) find the soln of following recurrence relation

$$a_n = 5a_{n-1} - 6a_{n-2} + 7^n$$

$$a_n = 5a_{n-1} - 6a_{n-2}$$

$$r^2 = 5r - 6$$

$$r^2 - 5r + 6 = 0$$

$$r_1 = 2 \ \& \ r_2 = 3$$

now since the root are distinct the soln is in the form

$$\boxed{G.S = \alpha_1 \cdot 2^n + \alpha_2 3^n}$$

The non-homogeneous part is

$$a_n = 7^n$$

The soln of non-homogeneous part is of the form

$$a_n = c\cdot 7^n \text{ where c is constant}$$

$$c\cdot 7^n = 5c\cdot 7^{n-1} - 6\cdot c\,7^{n-2} + 7^n$$

Dividing both sides by 7^n

$$\frac{c\cdot 7^n}{7^n} = \frac{5c\cdot 7^{n-1}}{7^n} - \frac{6\cdot c\,7^{n-2}}{7^n} + \frac{7^n}{7^n}$$

$$c = 5c\cdot 7^{-1} - 6\cdot c\cdot 7^{-2} + 1$$

$$c = 5c\cdot\frac{1}{7} - 6c\,\frac{1}{49} + 1$$

$$c = \frac{35c - 6c + 49}{49}$$

$$49c = 29c + 49$$

$$49c - 29c = 49$$

$$c = \frac{49}{20}$$

The particular soln is

$$a_n = \frac{49}{20}7^n$$

The overall soln is

H.S + P.S

$$\boxed{a_n = \alpha_1 2^n + \alpha_2 3^n + \frac{49}{20}7^n}$$

Ques. Find solution of

$$2a_n = 3a_{n-1} - a_{n-2} + 2^n$$

with $a_0 = 2$ & $a_1 = 3$

$$2a_n = 3a_{n-1} - a_{n-2}$$

$$2r^2 = 3r - 1$$

$$2r^2 - 3r + 1 = 0$$

$$(r-1)(2r-1) = 0$$

$$r = 1 \quad r = \frac{1}{2}$$

Now since the roots are distinct the $soln$ is in the form

$$a_{h.s} = \alpha_1 \cdot 1^n + \alpha_2 \cdot \frac{1}{2}^n$$

The non-homogenous eq^n is

$$a_n = 2^n$$

the $soln$ of non-homogenous part is of

$a_n = C \cdot 2^n$ where c is constant

$$2 \cdot c2^n = 3c2^{n-1} - c \cdot 2^{n-2} + 2^n$$

dividing by 2^n

$$2c = 3c2^{-1} - c \cdot 2^{-2} + 1$$

$$2c = \frac{3c}{2} - \frac{2c}{4} + 1$$

$$2c = \frac{6c - c + 4}{4}$$

$$8c - 5c = 4$$

$$c = \frac{4}{3}$$

The particular soln is

$$2a_n = \alpha_1 + \alpha_2 \frac{1}{2}^n + \frac{4}{3}2^n$$

$$a_n = \alpha_1 + \alpha_2 \frac{1}{2}^n + \frac{4}{3}2^n$$

$$a_0 = \alpha_1 + \alpha_2 \frac{1}{2}^0 + \frac{4}{3}2^0$$

$$a_0.\ 2 = \alpha_1 + \alpha_2 + \frac{4}{3}$$

$$\frac{2}{3} = \alpha_1 + \alpha_2 \quad (1)$$

$a_n = 3$

$$a_1 = \alpha_1 + \alpha_2 \times \frac{1}{2}^1 + \frac{4}{3}2^1$$

$$3 = \alpha_1 + \frac{\alpha_2}{2} + \frac{8}{3}$$

$$18 = 6\alpha_1 + 3\alpha_2 + 16$$

$$2 = 6\alpha_1 + 3\alpha_2$$

$$\alpha_1 = 0$$

$$\alpha_2 = \frac{2}{3}$$

The overall soln is

$$a_n = \frac{2}{3} \cdot \frac{1}{2}^n + \frac{4}{3}2^n$$

$$= \frac{2^{1-n}}{3} + \frac{4}{3}2^n$$

2015 - spring

3. a) Solve the following recurrence relation
$a_n = 7a_{n-1} - 10a_{n-2} + 16n$

sol^n

Associated part.

$a_n = 7a_{n-1} - 10a_{n-2}$

The characteristics eq^n is

$r^2 - 7r + 10 = 0$

or, $r^2 - 5r - 2r + 10 = 0$

$(r-5)(r-2) = 0$

$r = 2, 5$

The general sol^n for the distinct root is of the form

$a_n = \alpha_1 . 2^n + \alpha_2 . 5^n$ — (a)

The non-homogeneous part is

$a_n = 16n$

The particular solution is of the form

$a_n = Cn + D$

$Cn + D = 7(n-1) - 6(n-2) + 16n$

$\Rightarrow \cancel{16n = Cn + D} = 7n - 7 - 6n + 12 + 16n$

$\cancel{C = 16}$ $Cn + D = 13n + 13$

$\cancel{D = 0}$ $C = 13$

$D = 13$

Now,

$a_n \cancel{= \alpha_1 . 2^n} + \alpha_2 . 5^n + 16n$

$a_n = 13n + 13$

$a_n = \alpha_1 2^n + \alpha_2 5^n + 13n + 13$

Scanned by CamScanner

2016 (Spring)

~~Sto~~

5 (a) $a_n = 5a_{n-1} - 6a_{n-2} + 42\cdot 4^n$

$a_1 = 56 \qquad a_2 = 278$

Sol^n

Associated part:

$a_n = 5a_{n-1} - 6a_{n-2}$

The characteristic eq^n is

$r^2 - 5r + 6 = 0$

$r^2 - 2r - 3r + 6 = 0$

$r(r-2) - 3(r-2) = 0$

$(r-2)(r-3) = 0$

$r = 2, 3$

The general sol^n for distinct root is of the form

$a_n = \alpha_1 \cdot 2^n + \alpha_2 \cdot 3^n$ — (a)

The non-homogeneous part is

$a_n = 42\cdot 4^n$

The particular sol^n is of the form

$a_n = C\cdot 4^n$

$\cancel{C4^n + D} = 5(n-1) - 6(n-2) + 42\cdot 4^n$

$C\cdot\cancel{4^n + D} = 5n - 5 - 6n + 12 + 42\cdot 4^n$

$\cancel{C\cdot 4^n + D} =$

$C\cdot 4^n + D = 5(C\cdot 4^{n-1}) - 6$

$$\boxed{\text{ifn} - Cn + D}$$

$$\boxed{2^n = C \cdot 2^n}$$

classmate

Date

Page

$$c \cdot 4^n = 5c \cdot 4^{n-1} - 6c \cdot 4^{n-2} + 42 \cdot 4^n$$

or $c = \frac{5c}{4} - \frac{6c}{16} + 42$

$$c = \frac{20c - 6c + 672}{16}$$

$$16c = \cancel{2}\,14c + 672$$

$$2c = 672$$

$$c = 336$$

$\therefore\ a_n = 336 \cdot 4^n$

$\therefore\ a_n = \alpha_1 \cdot 2^n + \alpha_2 \cdot 3^n + 336 \cdot 4^n$

$a_1 = 56$

$\therefore\ 56 = \alpha_1 \cdot 2^1 + \alpha_2 \cdot 3^1 + 336 \cdot 4^1$

$56 = 2\alpha_1 + 3\alpha_2 + 1344$

$2\alpha_1 + 3\alpha_2 = -1288$

$a_2 = 276$

$276 = \alpha_1 \cdot 2^2 + \alpha_2 \cdot 3^2 + 336 \cdot 4^2$

$276 = 4\alpha_1 + 9\alpha_2 + 5376$

$4\alpha_1 + 9\alpha_2 = -5100$

$\alpha_1 = -618$

$\alpha_2 = 841.3$

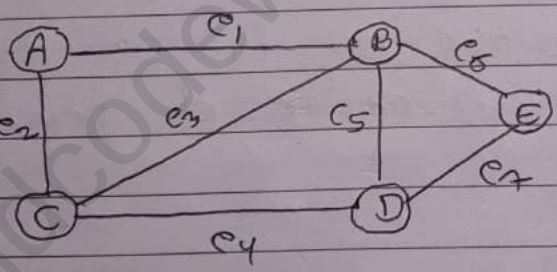
Unit 1

Graph Theory

→ Many situation that occur in computer science, physical science, chemical science economics and many other area can be analyzed big using techniques found in a relatively new area of mathematics called as graph theory.

→ Graph is a discrete structure consisting of vertices and edges.

Graph is defined by $G = (V, E)$
where $V = \{v_1, v_2, \ldots, v_n\}$ called set of vertices.
$E = \{e_1, e_2, \ldots, e_n\}$ called set of edges or arcs or links.



Here, $V = \{A, B, C, D, E\}$

$E = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7\}$

→ In a graph each vertex is represented by a circle or dot and each edge is represented by line or can arrow.

→ Further, each edge is represented by pair of vertices.
Let v_i and v_j are two vertices connected by edge e_i.
Then we can write,

$$e_i = (v_i, v_j) \text{ or } (v_j, v_i)$$

eg:- $e_1 = (A, B)$ or (B, A)

→ These pairs of vertices are either ordered pair or un-ordered pair.

→ If the direction is provided for the edge containing pair of vertices then such pair is called ordered pair.

→ If no such direction is provided then such pair is un-ordered pair.

e.g:-

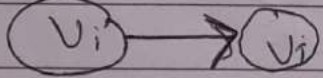


fig:- ordered pair

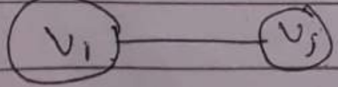


fig:- un-ordered pair

In case of ordered pair

$$(v_i, v_j) \neq (v_j, v_i)$$

For un-ordered pairs

$$(v_i, v_j) = (v_j, v_i)$$

→ Based on these pair of vertices a graph can be divided into two types

(i) Directed graph
(ii) Un-directed graph

(a) Directed graph

→ ~~The~~ If every vertices pair of a graph in ordered one i.e. direction is provided then such graph are called directed graph.

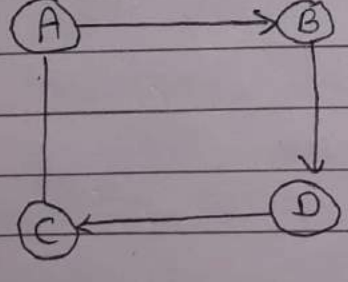


Fig :- Directed graph

(b) Undirected graph :
→ If every pair of vertices of a graph is an un-ordered then such graph are ~~called~~ undirected one.

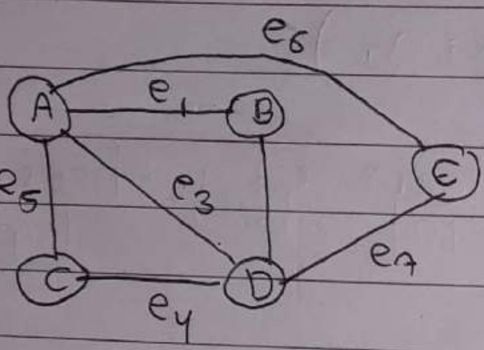


fig:- Undirected graph

Types of graph on the basis of edges

(a) simple graph :-
→ simple graph consists of non-empty set of vertices and edges having neither loops or parallel edges

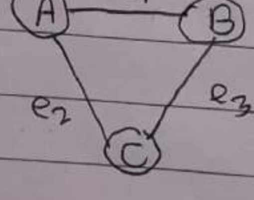


fig:- simple graph

(b) Multigraph

→ A graph $G(V, E)$ is said to be multigraph such that some of the edges are parallel.

→ Two or more than two edges having same end points are called parallel edges.

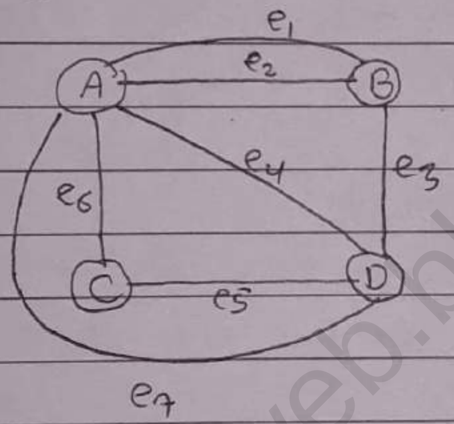


fig :- Multigraph

(c) Pseudo graph

→ A graph $G(V, E)$ is said to be pseudo graph if G has both loops and multi edges or loops only

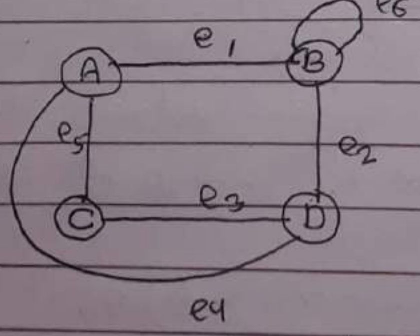


fig :- Pseudo graph

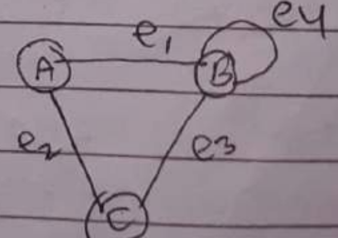


fig :- Pseudo graph

Regular graph (platonic graph)

→ If every vertex of a graph consists same degree, then such graph are called regular graph

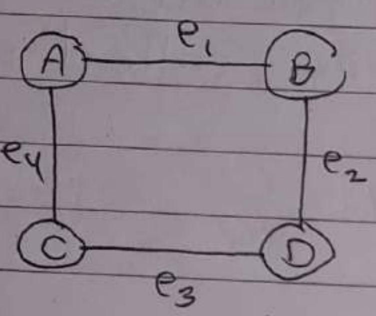


Fig :- Regular graph

Degree of vertex

→ Total number of edges incident on a particular vertex is called it's degree.

→ Let 'v' be the vertex then its degree is denoted by deg(v).

→ In case of directed graph degree is defined as sum of in degree and out degree.

→ Total number of incoming edges towards a particular vertex is called in degree.

→ Total number of outgoing edges from a particular vertex is called out degree.

Adjacency and Incidence

Relationship bet^n vertices of a graph is called adjacency.

→ Nodes that are directly connected by an edges are said to be adjacent nodes.

→ In case of ordered pairs (v_i, v_j) node v_j is adjacent to node v_i ~~be~~ but not vice versa.

→ In case of unordered pair (v_i, v_j), v_j is adjacent to node v_i and v_i is adjacent to v_j.

→ The relationship vertices and edges of a graph is called incidence.

→ An edge is said to be incident on both its end point.

e.g.: $e_i = (v_i, v_j)$
then e_i is incident to both v_i & v_j.

Graph Representation techniques

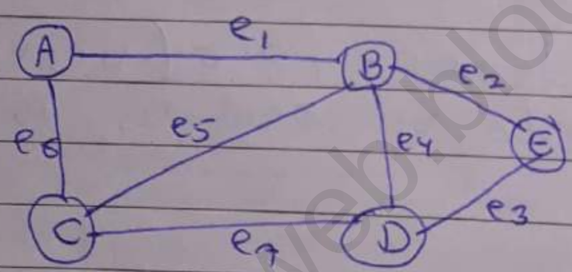
a) Adjacency matrix
b) Incidence matrix
c) Adjacency list

(a) Adjacency matrix

A matrix formed with the help of vertices of a graph is called Adjacency matrix.

Let A be the adjacency matrix of order $(n \times n)$ then each element is represented as::

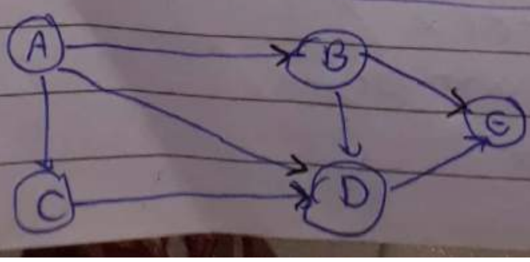
$$a_{ij} = \begin{cases} 1 & \text{if there exist an edge between vertices} \\ 0 & \text{otherwise} \end{cases}$$



Graph G

$$A = \begin{array}{c|ccccc} & A & B & C & D & E \\ \hline A & 0 & 1 & 1 & 0 & 0 \\ B & 1 & 0 & 1 & 1 & 1 \\ C & 1 & 1 & 0 & 1 & 0 \\ D & 0 & 1 & 1 & 0 & 1 \\ E & 0 & 1 & 0 & 1 & 0 \end{array}$$

For Directed graph



~~A — B — C — D~~ E

A = ~~A~~ 0

B 0

C 1

D 0

E 0

	A	B	C	D	E
A	0	1	1	1	0
B	0	0	0	1	1
C	0	0	0	1	0
D	0	0	0	0	1
E	0	0	0	0	0

Incidence matrix

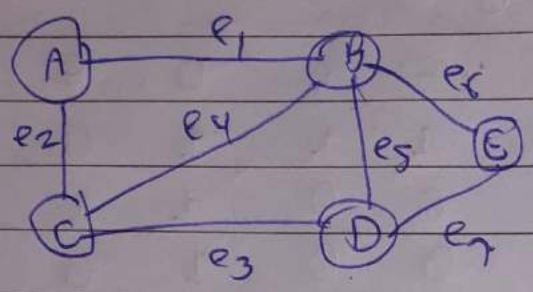
A matrix formed with vertices and edges of a graph is called incidence matrix.

Let 'I' be the incidence matrix of order $(m \times n)$ then each element of matrix is defined as

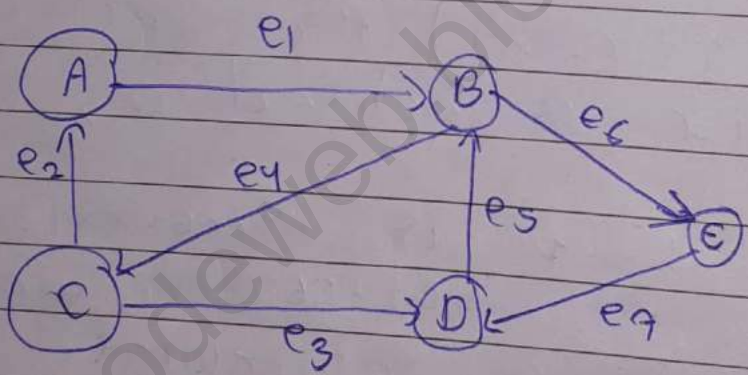
$$a_{ij} = \begin{cases} 1 & \text{if there exist an edge between } v_i \text{ and } v_j \\ 0 & \text{otherwise} \end{cases}$$

For directed graph

$$a_{ij} = \begin{cases} 1 & \text{if there exist an edge directed away from } v_i. \\ 1 & \text{if there exist an edge directed towards} \\ 0 & \text{otherwise} \\ 0 & \text{otherwise} \end{cases}$$



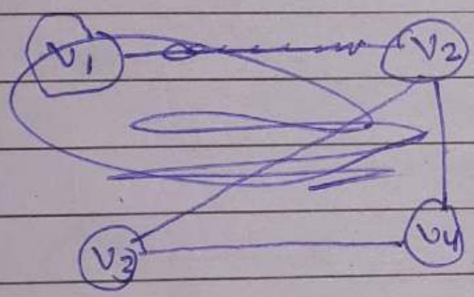
	e_1	e_2	e_3	e_4	e_5	e_6	e_7
A	1	1	0	0	0	0	0
B	1	0	0	1	1	1	0
C	0	1	1	1	0	0	0
D	0	0	1	0	1	0	1
E	0	0	0	0	0	1	1

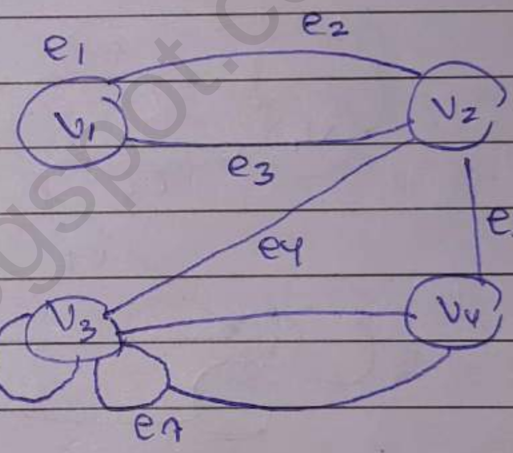


	e_1	e_2	e_3	e_4	e_5	e_6	e_7
A	1	-1	0	0	0	0	0
B	-1	0	0	1	-1	1	0
C	0	1	1	-1	0	0	0
D	0	0	-1	0	1	0	-1
E	0	0	0	0	0	-1	1

Draw undirected graph for following adjacency matrix.

	v_1	v_2	v_3	v_4
v_1	1	2	0	0
v_2	2	0	1	1
v_3	0	1	2	2
v_4	0	1	2	0

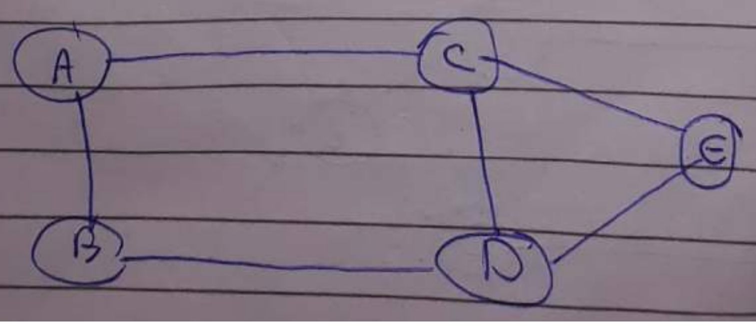




Adjacency list

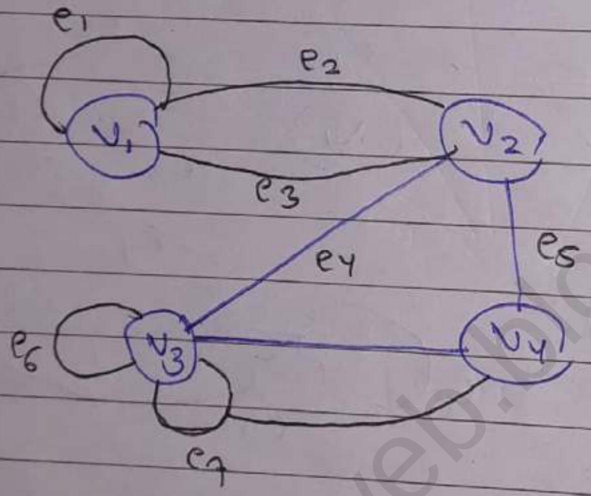
It is the dynamic representation of a graph.

→ In an adjacency list, a list of all adjacent vertex is formed and connected with each other with the help of pointer.



Draw undirected graph for following adjacency matrix.

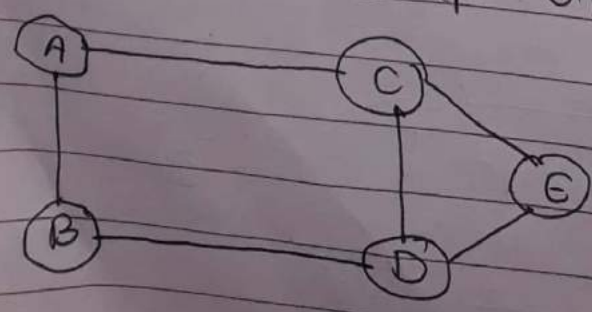
	v_1	v_2	v_3	v_4
v_1	1	2	0	0
v_2	2	0	1	1
v_3	0	1	2	2
v_4	0	1	2	0



Adjacency list

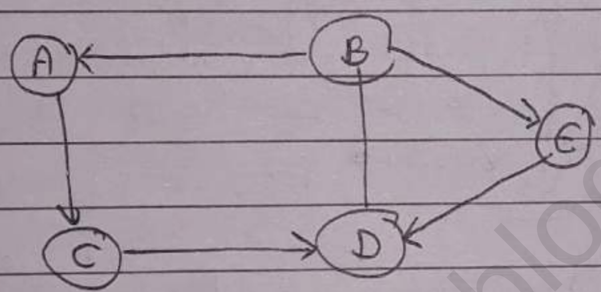
It is the dynamic representation of a graph.

→ In an adjacency list, a list of all adjacent vertex is formed and connected with each other with the help of pointer.



vertex	Adjacent list
A	B, C
B	A, D, E
C	A, D.
D	B, C, E
E	B, D

vertex	Adjacent list
A	C
B	A, E
C	D
D	B
E	D

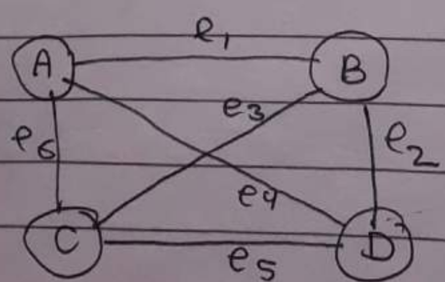


Complete Graph

A graph where each vertex are connected to each other is called complete graph.

-> A complete graph with 'n' vertices is denoted K_n

e.g K_4



Draw K_6 and write down its incidence matrix

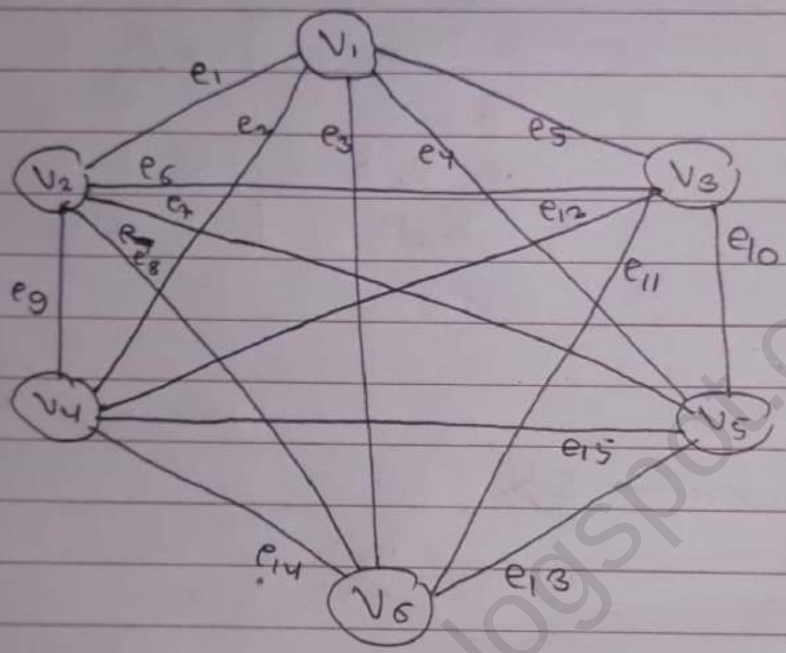


fig :- complete graph of 6 vertex

	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8	e_9	e_{10}	e_{11}	e_{12}	e_{13}	e_{14}	e_{15}
v_1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
v_2	1	0	0	0	0	1	1	1	1	0	0	0	0	0	0
v_3	0	0	0	0	1	0	0	0	0	1	1	1	0	0	0
v_4	0	1	0	0	0	0	0	0	1	0	0	1	0	1	1
v_5	0	0	0	1	0	0	1	0		1	0	0	1	0	1
v_6	0	0	0	1	0	0	0	1	0	0	1	0	1	1	0

Bi - partite Graph

A graph is said to be partite it is vertices of are divided into two parts such that the vertices of first part are connected to vertices of second part but vertices of same part are not connected.

→ If all the vertices of first part are connected to all the vertices of second part, then such graph are called complete bi-partite graph.

→ A complete bi-partite with 'm' vertices in first part and n-vertices in second part is denoted by K_{mn}

e.g. V = {A, B, C, D, E}

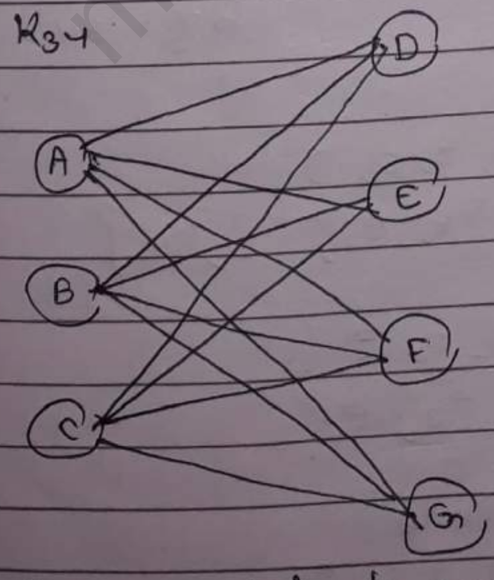


fig :- complete - bi - partite graph.

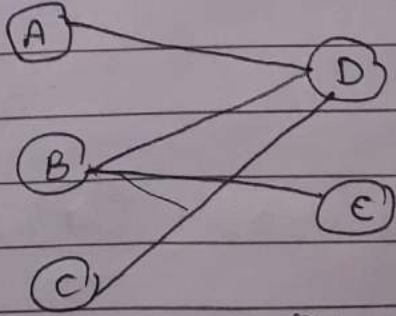


fig :- Bi-partite graph

Sub-Graph

Let $G = (V, E)$ be a graph then sub-graph of G is denoted by

$$H = (V, E)$$

where

$$V' \subset V \text{ and } E' \subset E$$

e.g:-

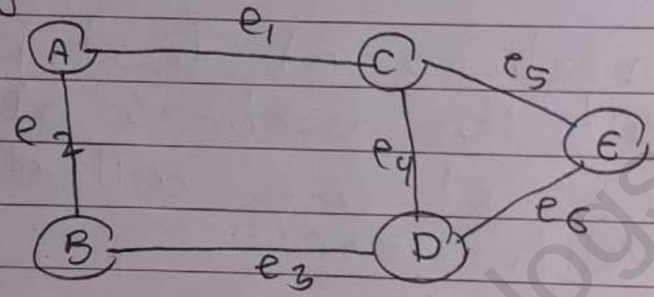


fig:- Graph G

$G(V, E)$

$V = \{A, B, C, D, E, F\}$

$E = \{e_1, e_2, e_3, e_4, e_5, e_6\}$

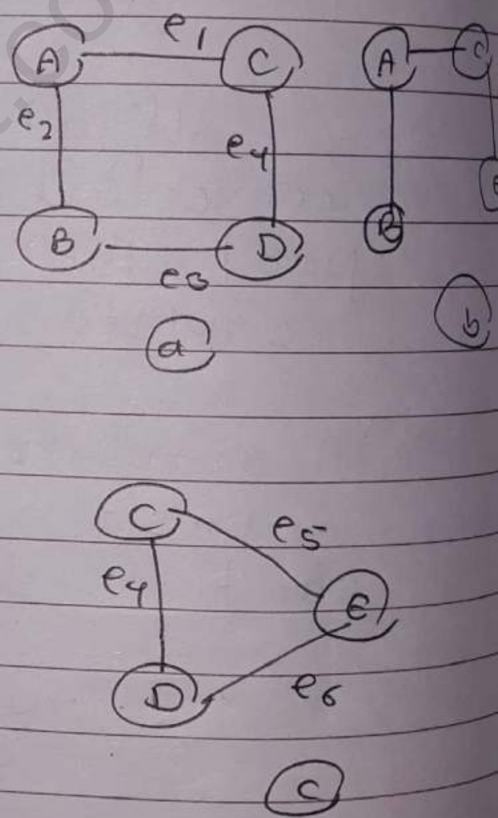


fig:- sub-graph of G

sub graph (a), $V = \{A, B, C, D\}$

$E' = \{e_1, e_2, e_3, e_4\}$

(b) $V' = \{A, B, C, E\}$

$E' = \{e_1, e_2, e_5\}$

(c) $V' = \{C, D, E\}$

$E' = \{e_4, e_5, e_6\}$

Weighted Graph

If some numerical value (weight) is assigned for the edge of a graph then such graph are called graph.

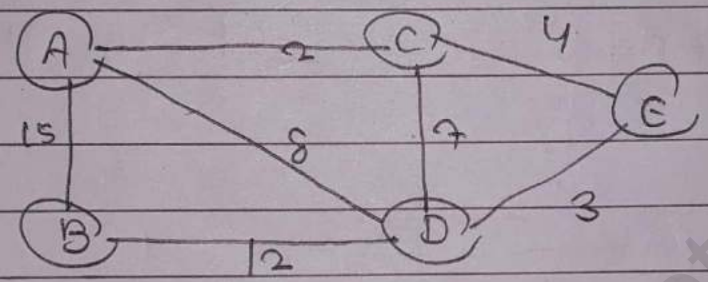


fig :- weighted Graph.

Spanning Tree

A spanning tree of a graph is a sub-graph of 'G' that contain all the vertices of 'G' and does not contain a cycle.

e.g :-

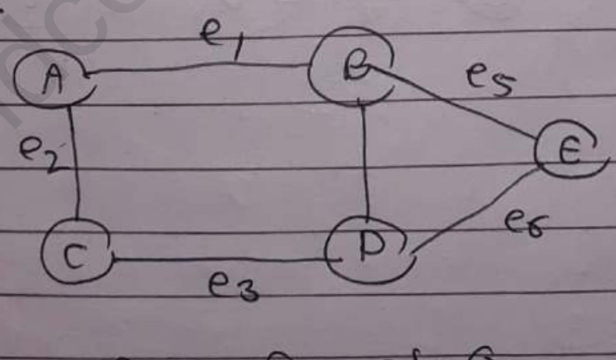
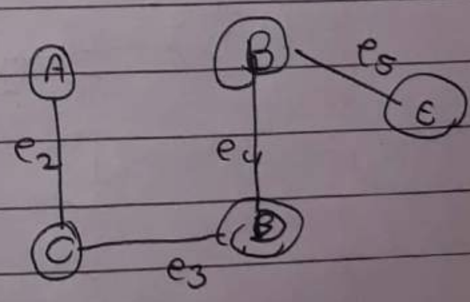
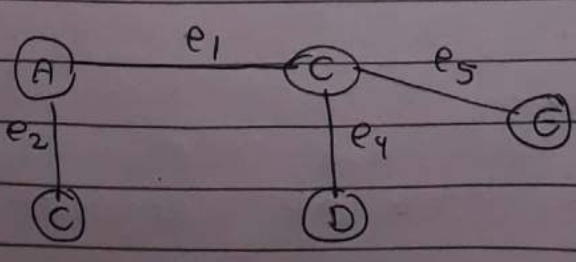


fig :- Graph G





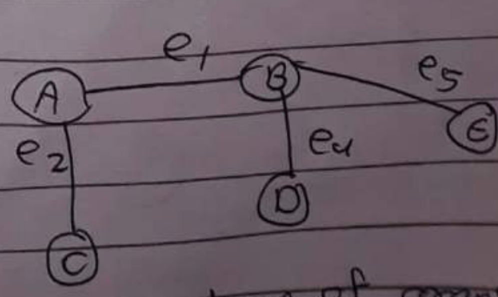


fig :- Spanning tree of graph G.

Minimum spanning tree

In minimum spanning tree, the tree with the minimum cost is constructed.

→ The cost of the tree is computed by adding all the weight of the edge included in the spanning tree.

e.g:-

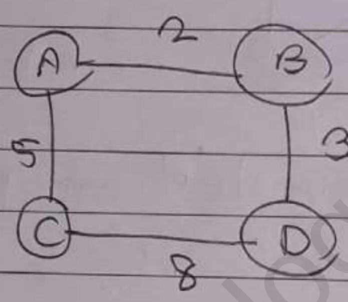
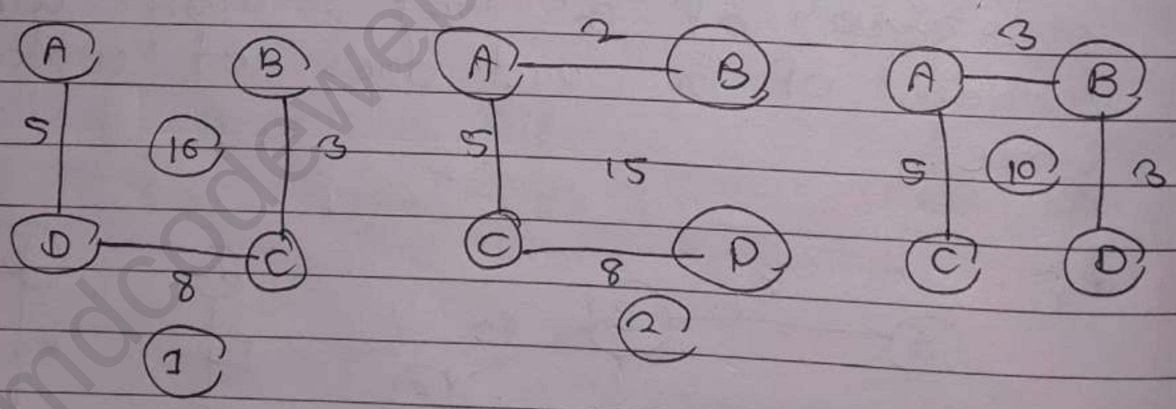


fig:- weighted graph G.



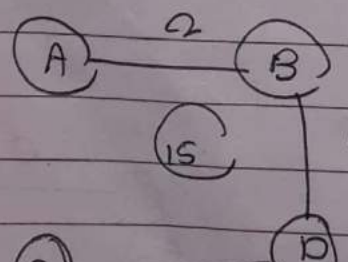


fig:- Spanning tree of weight greater

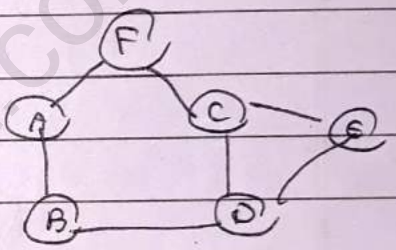
~~Algorithm to construct~~ minimum sp:

→ Here spanning tree(3) has the minimum cost, thus it is the required minimum spanning tree of the given weighted graph G.

Algorithm to construct minimum spanning tree

(a) Prism's Algorithm

(b) Kruskal's Algorithm.



(a)

(a) Prism's Algorithm

In prism's algorithm we start with any arbitary vertex for a given graph G. Let 'v' be the arbitary vertex then we find all the adjacent vertex of 'v' and form a set containing all the vertices of v.

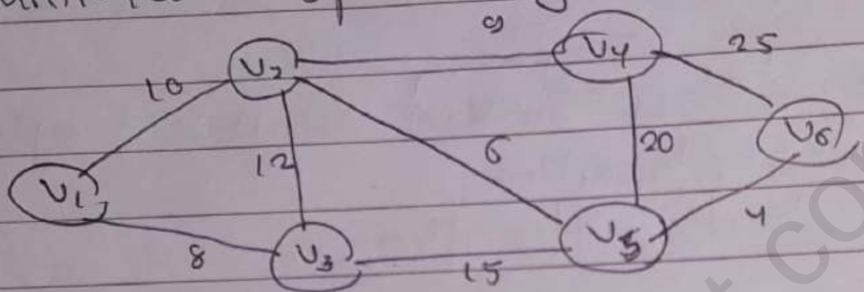
After that we select the vertex pair with least weight from the set and add it to tree being forward. Let the vertex pair be (v,w).

→ Now we find all the adjacent vertices of 'w' and remaining vertices of previous set.

→ Then we select the vertex pair with minimum weight and add that vertex pair in our tree being constructed.

→ During the process of addition of vertex pair, if any vertex with minimum weight forms a cycle it is discarded and we move to the vertex pair with next minimum weight and add it to the tree.

→ The process is continued until all the vertex pair from the set are added to the tree. when the set becomes empty, the tree obtained is the minimum spanning tree.



v_5 —— v_6 $= 4$

v_5 —— v_4 $= 20$

v_5 —— v_2 $= 9$

v_5 —— v_3 $= 15$

Using prim's Algorithm

We start with any arbitary vertex. let us choose. now, the adjacent vertex pair of v_4 are

~~ef~~ $(v_4, v_5) = 20$

$(v_4, v_6) = 25$

$(v_4, v_2) = 9$

We choose the vertex pair with minimum weight i.e. $(v_4, v_2) = 9$ add it to our tree to be constructed.

$v_2 \overset{9}{\text{——}} v_4$

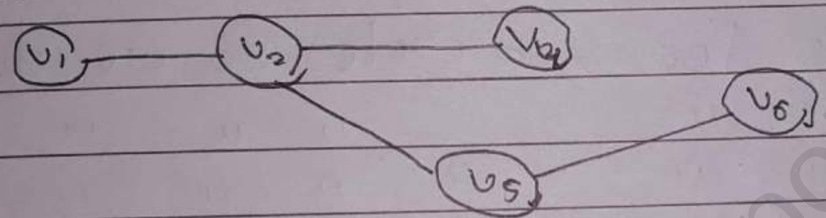
Now, the adjacent of v_2 and remaining vertex pair of previous set.

$(v_2, v_1) = 10$, $(v_4, v_5) = 20$

$(v_2, v_3) = 12$, $(v_4, v_6) = 25$

$(v_2, v_5) = 6$

→ $(v_2, v_5) = 6$ is the vertex pair with minimum weight so weight so we add it to our tree.



→ The adjacent of v_5 are

$(v_5, v_6) = 4$

$(v_5, v_3) = 13$

$(v_2, v_1) = 10$

$(v_2, v_3) = 12$

$(v_4, v_5) = 20$

$(v_4, v_6) = 23$

The adjacent of v_1 are

$(v_1, v_3) = 8$

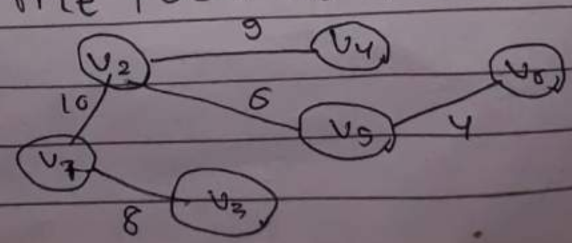
$(v_3, v_5) = 15$

$(v_2, v_3) = 12$

$(v_4, v_5) = 20$

$(v_4, v_6) = 25$

→ Add (v_1, v_3) to the tree as it has minimum weight



→ Adjacent of v_3

$(v_3, v_2) = 12$ $(v_4, v_6) = 25$

$(v_3, v_5) = 13$

$(v_4, v_5) = 20$

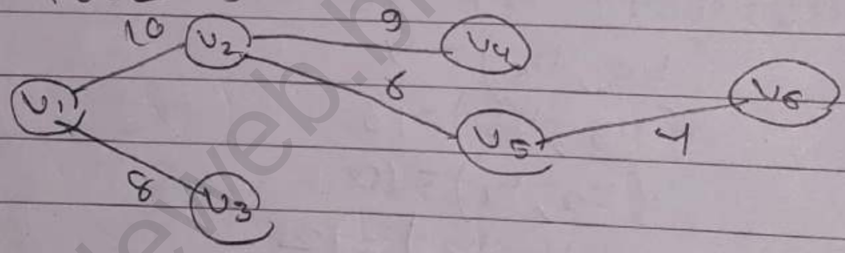
→ The vertex pair $(v_3, v_2) = 12$ with minimum cost form a cycle so we discard it and make on with the vertex pair $(v_3, v_5) = 15$

→ (v_3, v_5) also forms a cycle so we discard it

(v_4, v_5) " " " " " " " "

→ (v_4, v_6) " " " " " " " "

→ Here the list becomes empty so the final spanning tree is



The total cost is $10+8+9+6+4 = 37$

Kruskal's Algorithm

→ In Kruskal's algorithm we list all the pair of vertices of given graph in ascending order of their weight i.e. vertex pair with least weight is the first pair of the list, the pair with the next minimum is the second pair and so on.

→ Then, we choose the vertex pair with least weight and add it to the tree being ~~forward~~ a formed after that the vertex pair with next minimum weight from the list is selected and added to the tree and so on.

→ During the process of adding vertex pair if any vertex pair with minimum weight forms a cycle, we discard that vertex pair.

→ This process is continued until the list becomes when list becomes empty the tree obtained is the required minimum spanning tree.

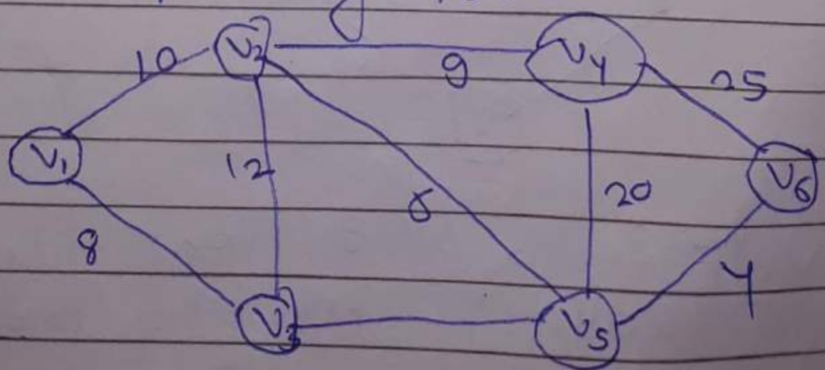


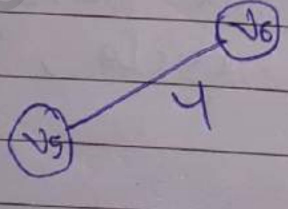
fig:- weighted graph.

solution

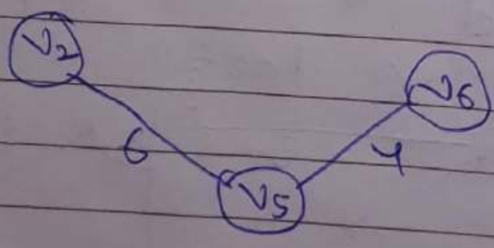
Arranging the vertex pair according to their weight in ascending order

$(v_5, v_6) = 4$
$(v_2, v_5) = 6$
$(v_1, v_3) = 8$
$(v_2, v_4) = 9$
$(v_1, v_2) = 10$
$(v_2, v_3) = 12$
$(v_3, v_5) = 15$
$(v_4, v_5) = 20$
$(v_4, v_6) = 25$

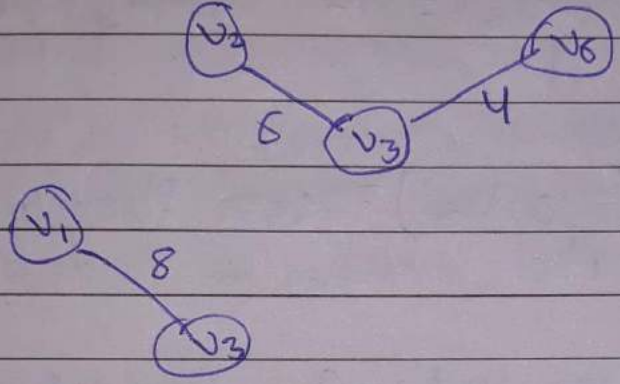
Here, $(v_5, v_6) = 4$ is the least weight vertex pair, so we add it to our tree being constructed.



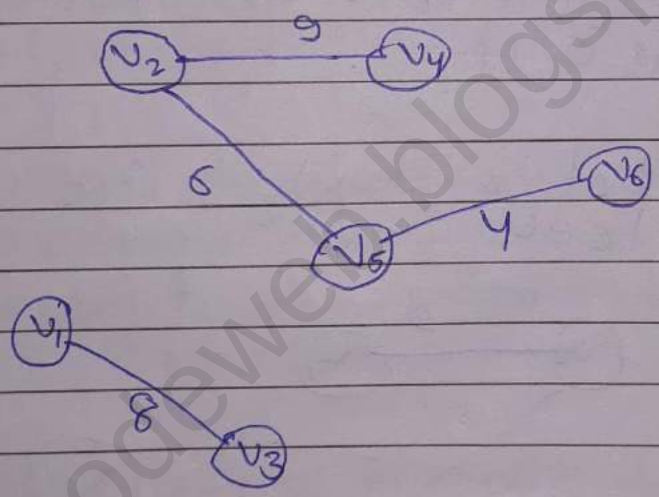
→ Add $(v_2, v_5) = 6$ in our tree



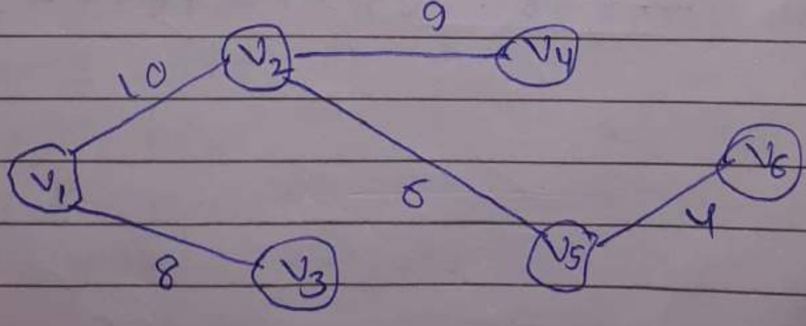
→ $(v_1, v_3) = 8$ is the next minimum weight so add it to the tree



→ $(v_2, v_4) = 9$ is the next candidate vertex pair to be added in the tree.



→ Add $(v_1, v_2) = 10$ in our tree being constructed.



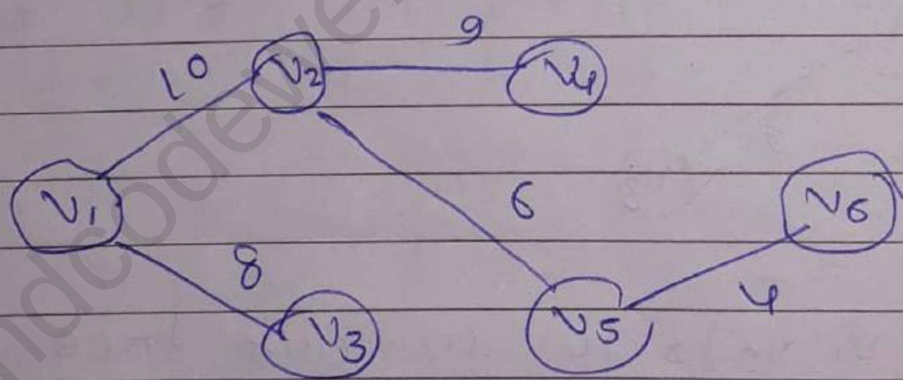
→ vertex pair $(v_2, v_3) = 12$ forms a cycle so we discard it.

→ vertex pair $(v_3, v_5) = 15$ forms a cycle so discard it.

→ vertex pair $(v_4, v_5) = 20$ forms a cycle so discard it.

→ vertex pair $(v_4, v_6) = 25$ forms a cycle so discard it.

since the list is empty, the final minimum spanning tree



Total cost is $10 + 9 + 8 + 6 + 4 = 37$

Shortest Path Algorithm (Dijkstra's Algorithm)

-> Dijkstra's Algorithm is used to find the shortest distance between source and destination node.

For this we first assign the source node distance by zero and other nodes by infinity. So that to indicate distance from source to other nodes yet to be calculated.

-> Then we calculate the distance of all the adjacent nodes of source node and select the adjacent node with least distance.

-> Let this adjacent node be 'v' such that its distance from source node is L(v).

-> After that we find the adjacent node of 'v' and choose the node with the least distance. Let that node be 'w' with distance from 'v' as L(v, w). The distance of 'w' from starting node is
L(v) + L(v, w).

-> For the node 'w' there may be previously calculated distance. Let that distance be L(w).

→ Now we compare the previously calculated distance with newly calculated distance and if the new calculated distance is smallest one, we discard the previously calculated distance.

i.e.

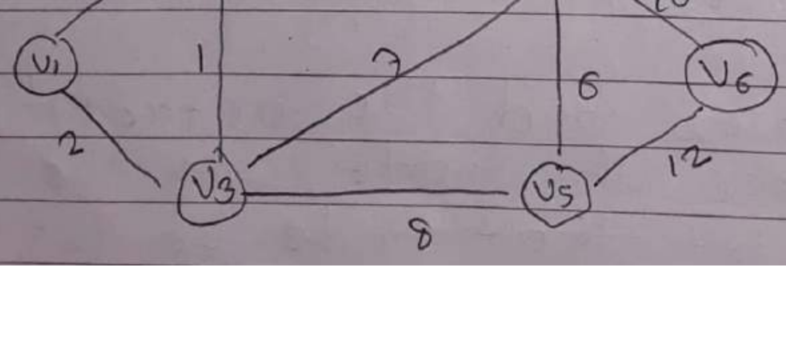
$L(w) > L(v) + L(v, w)$

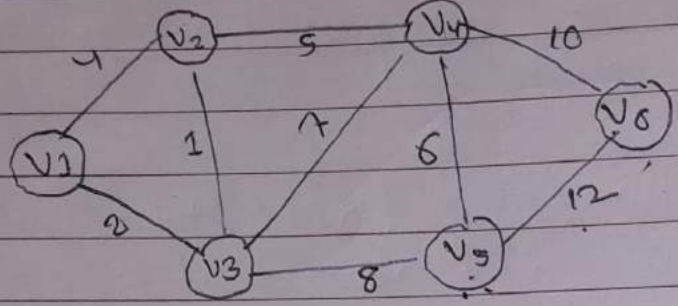
then

$L(w) = L(v) + L(v, w)$

→ This process is continued until all paths for the destination nodes are calculated. Then final given path given by the algorithm is the shortest path.

E.g.:





Initially

vertex	v_1	v_2	v_3	v_4	v_5	v_6
label	0	∞	∞	∞	∞	∞

The adjacent of v_1 ~~and~~ are v_2 and v_3. Then their distance from v_1 is

$$L(v_2) = L(v_1) + L(v_1, v_2)$$
$$= 0 + 4 = 4$$

$$L(v_3) = L(v_1) + L(v_1, v_3)$$
$$= 0 + 2 = 2$$

vertex	v_1	v_2	v_3	v_4	v_5	v_6
label	0	4	2	∞	∞	∞

Here length of v_3 is small, so we expand the adjacent nodes of v_3 i.e. v_2, v_4 and v_5

$$L(v_2) = L(v_3) + L(v_3, v_2)$$
$$= 2 + 1$$
$$= 3$$

$L(v_4) = L(v_3) + L(v_3, v_4)$
$= 2 + 7 = 9$

$L(v_5) = L(v_3) + L(v_3, v_5)$
$= 2 + 8 = 10$

Here, the length L(2) is small than the previously one so we replace the old path cost by new one.

vertex	v_1	v_2	v_3	v_4	v_5	v_6
label	0	3	2	9	10	∞

Here, length of v_2 is small so, we expand the adjacent node of v_2 i.e v_4

$L(v_4) = L(v_2) + L(v_2, v_4)$
$= 3 + 5 = 8$

Here, $L(v_4)$ is small that previously calculated one so we replace the old path cost by new path cost

vertex	v_1	v_2	v_3	v_4	v_5	v_6
Label	0	3	2	8	10	∞

Here, length of v_4 is small, so, we expand the adjacent node of v_4 i.e v_5 or v_6

$L(v_6) = L(v_4) + L(v_4, v_6)$
$= 8 + 10 = 18$

vertex	v_1	v_2	v_3	v_4	v_5	v_6
label	0	3	2	8	10	18

Here, ~~if~~ $l(v_5)$ is small ~~that previously~~ calculated one so we ~~replace~~ ~~th~~ expand the adjacent node of v_5 i.e. v_6

expanding adjacent of v_5 or as it has minimum path cost i.e. v_5

$$l(v_6) = l(v_5) + l(v_5, v_6)$$
$$= 10 + 12 = 22$$

Here, $l(v_6)$ is greater that previously calculated so, we do not replace the old path cost by new path cost

~~vertex~~

~~la~~

Planar Graph

→ A graph drawn in the plane where no edges intersects to each other is called planar graph.

→ A planar graph is divided into number of region called faces.

→ There are two types of faces, bounded and un-bounded faces.

→ Region formed by closed boundary of vertices to edges are called bounded faces.

→ A plane in which the graph is drawn is called un-bounded faces.

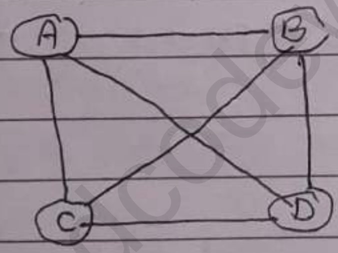


fig. Non-planar Graph

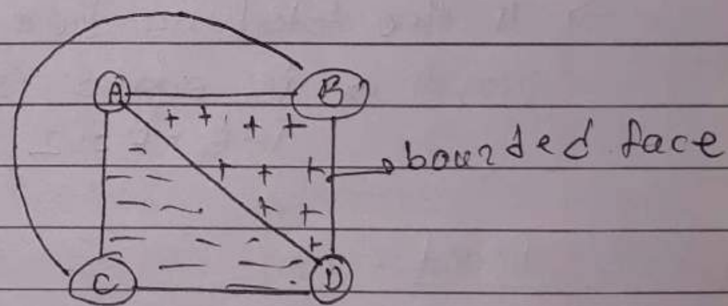
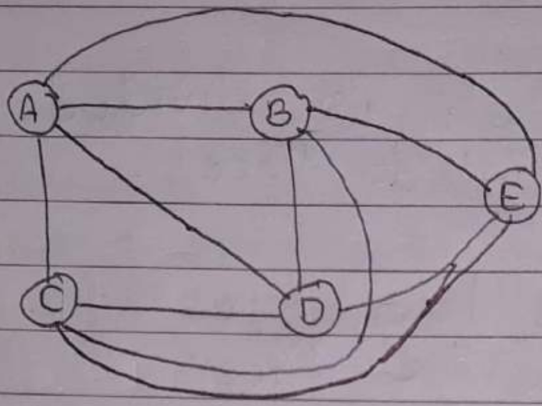


fig:- Planar Graph

Q. Draw a complete planar graph for 5 vertices



do not intersect

It doesn't exist for 5 vertices

vvr

Euler's formula for planar graph

→ If the total number of vertices of a planar graph is 'V' edges is 'E' and faces is F, then,

$$V - E + F = 2$$

Proof:-

→ We use the mathematical induction to prove the formula

we have, $V - E + F = 2$

Basic step

$E = 0$

$V = 1 \quad E = 0 \quad F = 1$

So,

$V - E + F$

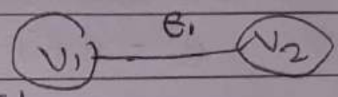
$= 1 - 0 + 1$

$= 2$

which is true.

when $E = 1$

case 1



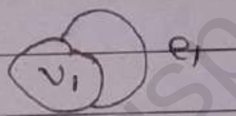
$V = 2 \quad E = 1 \quad F = 1$

$V - E + F$

$2 - 1 + 1$

$= 2$

case II



$V = 1, \ E = 1, \ F = 2$

$V - E + F$

$= 1 - 1 + 2$

$= 2$

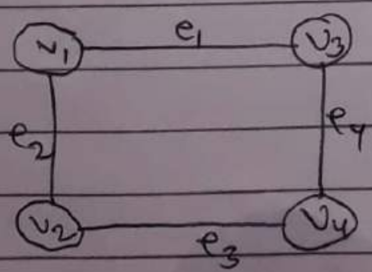
which is true

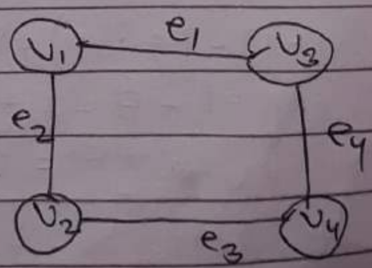
Inductive step

→ We assume that the formula is true for 'E' number of edges and try to prove it for $(E+1)$ number of edges.

~~case~~ case I

By adding the edges the number of vertices remarks the same but number of faces increases





Let V', E' and F' be the vertices, edges and faces of the newly formed graph respectively

i.e. $V' = V$, $E' = E+1$, $F' = F+1$

Now,

$$V' - E' + F' = 2$$

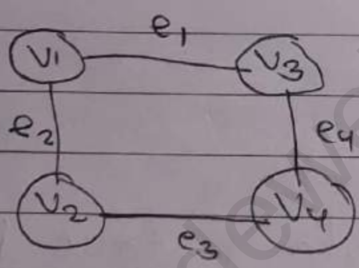
$$V - (E+1) + (F+1) = 2$$

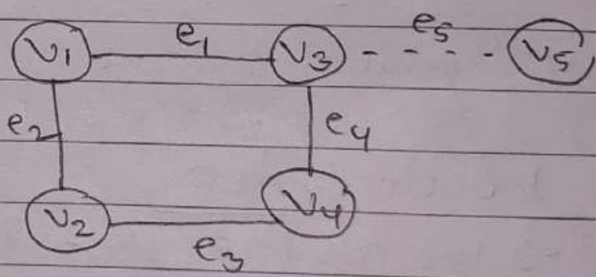
or, $V - E - 1 + F + 1 = 2$

$$\boxed{\therefore \; V - E + F = 2}$$

case II

By adding the edges the number of vertices increases but faces remains the same





Let V', E' and F' be the total number of vertices and faces of newly formed graph respectively

i.e. $V' = V+1$, $E' = E+1$, $F' = F$

Now,

$$V' - E' + F = 2$$

$$(V+1) - (E+1) + F = 2$$

$$V + 1 - E - 1 + F = 2$$

$$\boxed{V - E + F = 2}$$

which is true.

Hence, the Euler's formula for planar graph i.e. $\boxed{v - e + f = 2}$ is valid using mathematical induction.

2015 Spring Construct a graph having 9 vertices whose degree sequence is 2, 2, 2, 3, 3, 3, 4, 4 & 5. Also identify the total number of faces and edges.

→

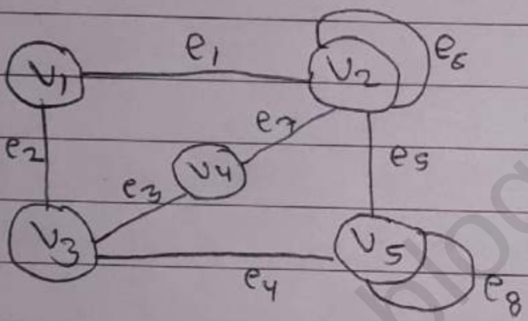
Theorem 1

The sum of degree of all the vertices of a given graph is equal to the twice the number of edges.

i.e.

$$\deg(v_1) + \deg(v_2) \ldots + \deg(v_n) = 2e$$

$$\boxed{\sum_{i=1}^{n} \deg(v_i) = 2e}$$



sum of degree $= 2+3+4+5+2$

$= 16$

or,

$2 \times$ no. of edges

$= 2 \times 8$

$= 16$

Theorem 2

For any graph there exist even number of vertices having odd degree.

we have,

$$\sum_{i=2}^{n} \deg(v_i) = 2e$$

or, $\deg(v_{even}) + \deg(v_{odd}) = 2e$

classmate

Date

Page

Q. For a planar graph with 20 vertices and degree of each vertex is 3. Find total number of region in the graph.

we have,

no. of vertices $(v) = 20$

degree of each vertex $= 3$

Total degree $=$ no. of vertices & degree of vertices

$= 20 \times 3$

$= 60$

we know,

$$\sum_{i=1}^{n} \deg(v_i) = 2e$$

$$60 = 2e$$

$$e = 60/2$$

$$\therefore e = 30$$

For planar graph using euler's formula

$$v - E + F = 2$$

$$20 - 30 + F = 2$$

$$F = 2 + 10$$

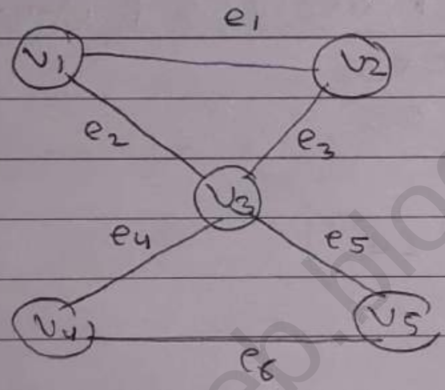
$$F = 12$$

$\therefore$ Total number of region in a graph is 12.

Euler's graph

→ A graph that contains Euler's cycle is called Euler's graph.

→ Euler's cycle is a closed path that is formed by visiting every edge of the graph exactly once and terminates in the vertex where we started to move.

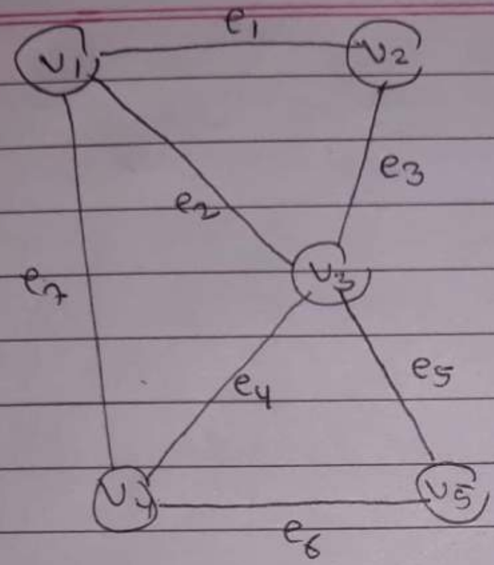


Here, $v_1 e_2 v_3 e_4 v_4 e_6 v_5 e_5 v_3 e_3 v_2 e_1 v_1$ is a Euler cycle.

Hamiltonion Graph (Hamilton Graph)

→ A graph that contains Hamilton cycle is called Hamilton graph.

→ Hamiltonion cycle is a closed path that is formed by visiting every vertex of a graph exactly once and terminates in the vertex from where we started to move.



Here, $v_1 e_7 v_4 e_6 v_5 e_5 v_1 e_1 v_2 e_1 v_1$

Q. What are the difference between euler's graph and Hamiltonian graph.

Euler's theorem

A graph 'G' consists of Euler's cycle if and only if every vertex of G has even degree.

P : A Graph 'G' consists of Euler's cycle.
Q : Every vertex of 'G' has even degree.

$P \leftrightarrow Q$.

1st part

If a graph 'G' contains a Euler cycle then every vertex of G has even degree.

2nd part

If a graph 'G' contains Euler cycle then every vertex of 'G' has even degree.

Proof:.

→ We know that Euler's cycle is a continuous path, that starts with any arbitary vertex and ends at that vertex from where we started to move. Furthermore, it is formed by visiting every incoming and one-out going edges for every vertex. This contributes degree 2 for each vertex which is even.

→ If 'a vertex is repeated or visited twice, it provides further degree 2, to that vertex which is also even.

→ It is also ~~one~~ true for starting vertex since when the continuous path starts it adds degree '1' to that vertex and when it ends to that vertex it adds further degree '1' making the degree of vertex '2' which is even

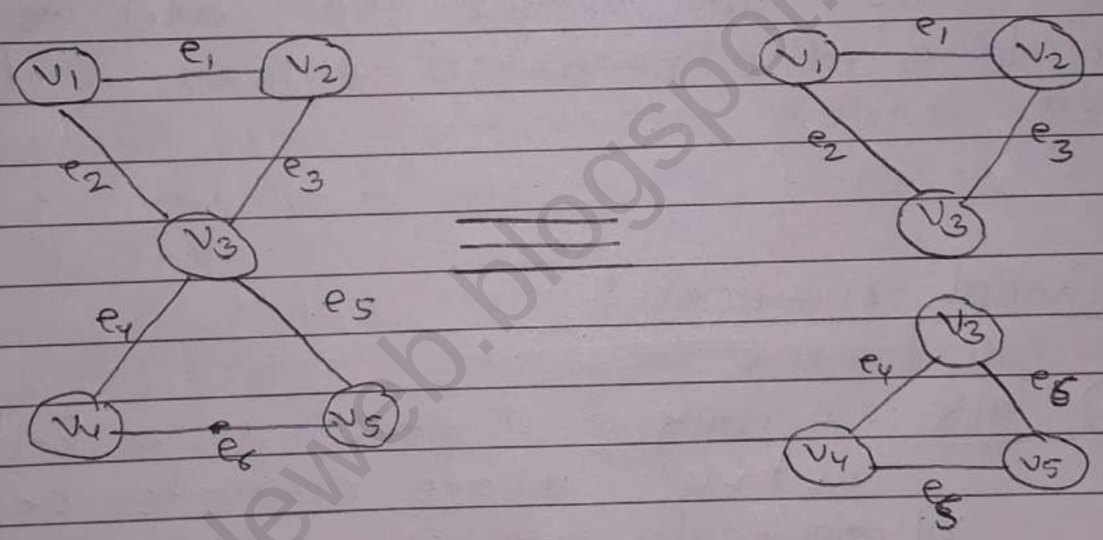
→ Hence we can say that every vertex of 'G' has even degree.

second part

1 If every vertex of 'G' has even degree then it contains euler's cycle.

Proof:-

To prove the statement we need to search for the continuous path that is formed by visiting every edges exactly once and ends in the vertex from where we started.



→ Let us start with any arbitary vertex, we know that its minimum degree is 2, we can say that there may exists an outgoing edges from this vertex to its adjacent vertex.

→ This case may exist for all the vertices of a graph.

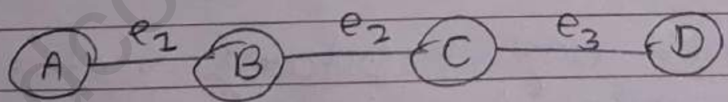
→ By processing this we find a path in which every edge is included and ends in the arbitory vertex from where we started, thus, we can say that the Euler's cycle exists.

→ But there may be the case, where cycle may form in part of the graph. In this case we split the given graph into multiple sub-graph by taking reference of common vertex.

→ Then we form different cycle in each sub-graph and merge them and resulting cycle is the required Euler's cycle.

Graph connectivity

① Walk :- A walk in a graph is a finite ordered set 'w' whose elements are alternatively vertices and edges.



$w = \{A, e_1, B, e_2, C, e_3, D\}$

→ The number of edges appearing in the sequence of walk is the length of walk.

→ If the length of walk is zero i.e. walk has no edges, it contains only one vertex and is called trivial walk.

→ A walk is closed if it starts and ends at the same vertex otherwise walk is open.

(ii) Trial

A walk w(u,v) in which all edges are distinct are called trial.

(iii) path

A walk in which all the vertex and edges are distinct are called path.

(iv) circuit

A closed trial which contains at least three edges is called circuit

Theorem :-

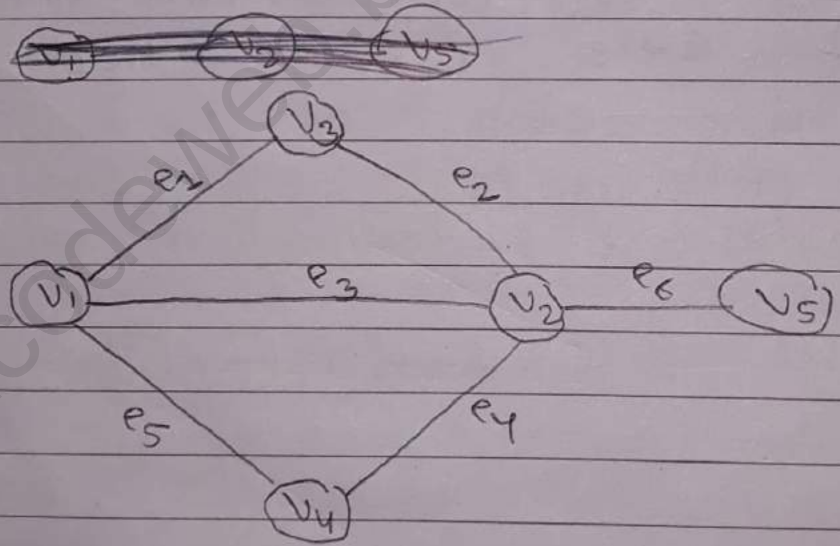
A graph 'G' contains eulerian, Trail if and only if two vertex of a G contains odd ~~de~~ edges.

=) **1st part**

If a graph contains eulerian Trail then exactly two vertex of 'G' has odd degree.

2nd part

If exactly two vertex of 'G' has odd degree, then G contains Eulerian Trial.



proof of 1st part

→ We know that Euler trail is the continuou path that starts with one orbitary vertex and ends with other arbitary vertex with edges being distinct.

→ Now, if a graph contains eulerian trail say from 'a' to 'z', then it must pass through every edge exactly once.

→ In this scenario, the first edge in the trail contributes one to the degree of vertex 'a' and at all other time when the edges passes through vertex 'a' it provides degree '2' for 'a'. Hence we can say 'a' has odd degree.

→ Similarly, the last edge in the path is coming to 'z' which contributes '1' degree to z, all the other time the edges provides degree 2, one for incoming and one for outgoing making degree of 'z' odd

→ All other vertices other than 'a' and 'z' have even degree, since the edges in these vertex enters and leaves contributing degree 2 every time.

→ Hence, in the connected graph 'G' having eulerian trail, exactly two vertices contains odd degree.

second part

If exactly two vertices of 'G' has odd degree then 'G' contains eulerian trail.

Proof:-

Let us consider two vertices 'a' and 'z' has odd degree.

→ Now consider another graph then adds an edge {a, z} to the original ~~path~~ graph, then the newly formed graph have every vertex with even degree. So there exist an Euler's cycle in a new graph.

→ The removal of the new edge gives the Eulerial trial in the original path.

→ Hence if exactly two vertices of 'G' has odd degree then G' contains Eulerian trial.

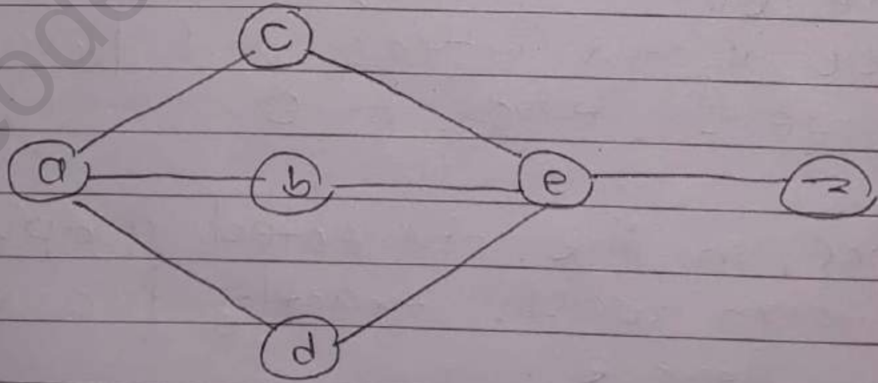


fig:- Two vertices with odd degree.

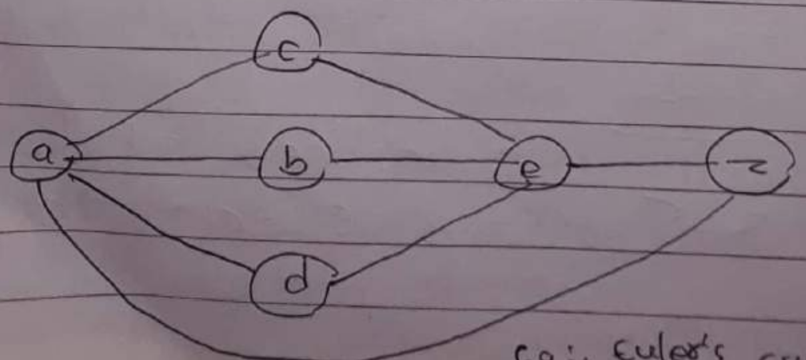
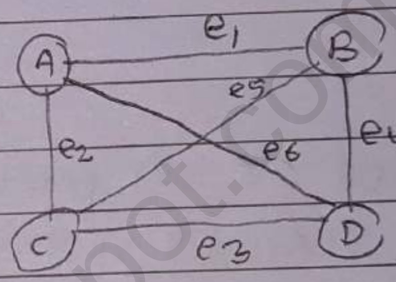


fig:- Euler's cycle (Every vertex has even

Show that for a complete graph with 'n' vertices the number of edge is given by $\frac{n(n-1)}{2}$

solution



Let us consider a graph with 'n' vertices since the graph is complete each vertex are connected to each other by distinct edges

so, the total number of degree for each vertex is $(n-1)$

The sum of all the degree of a graph is given by

$deg(v_1) + deg(v_2) + \cdots + deg(v_n)$

$(n-1) + (n-1) + \cdots + (n-1)$

for n-vertices

$n(n-1)$ is to total degree.

Again,

We know that the sum of all the degree of vertices is equal to twice the number of edges.

i.e. $\sum_{i=1}^{n} deg(v_i) = 2e$

$n(n-1) = 2e$

$e = \frac{n(n-1)}{2}$ proved//

Prove or disprove a complete graph with 5 vertices cannot be planar graph.

If 'G' is a connected planar graph with 'e' edges and 'v' vertices where $v \geq 3$ then $e \leq 3v-6$

Finite state Automata

A finite state machine (FSM) is defined mathematically by 5 tuple.

$$M = (Q, I, O, F, G)$$

where,

Q = Finite set of states
I = finite set of inputs
O = Finite set of outputs
F = transition function
G = output relation

E.g:-

Fan as an FSM

Q = {on, off}
I = {press}
O = {fan on, fan off}

F consists of
(on, press) → fan off
(off, press) → fan on

G consists of
(on, press) → (off, fan off)
(off, press) → (on, fan on)

Terminology used in FSM

① **Alphabet**:- It is the collection of input symbols.
It is denoted by Σ
Eg:- binary ~~equiv~~ alphabet = $\{0, 1\}$

(ii) **String**:- It is the combination of multiple occurance of input symbols. It is denoted by ω.

e.g:- ω = 0010, 110110 ...

(iii) **Empty string**:- No occurance of input symbol. It is denoted by epsilon. i.e. ϵ.

(iv) **Language**:- The collection of all possible string over some given alphabet. It is denoted by L.

e.g:- L = $\{0, 1, 11, 001, 00110, \ldots\}$

Design a FSM that accepts a string that starts with 01.

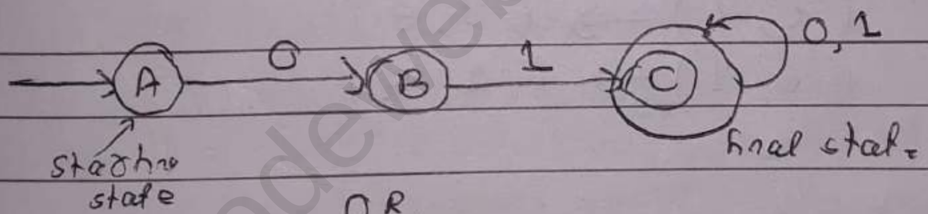
Sol^n

The finite state machine ~~co~~ M can be defined as

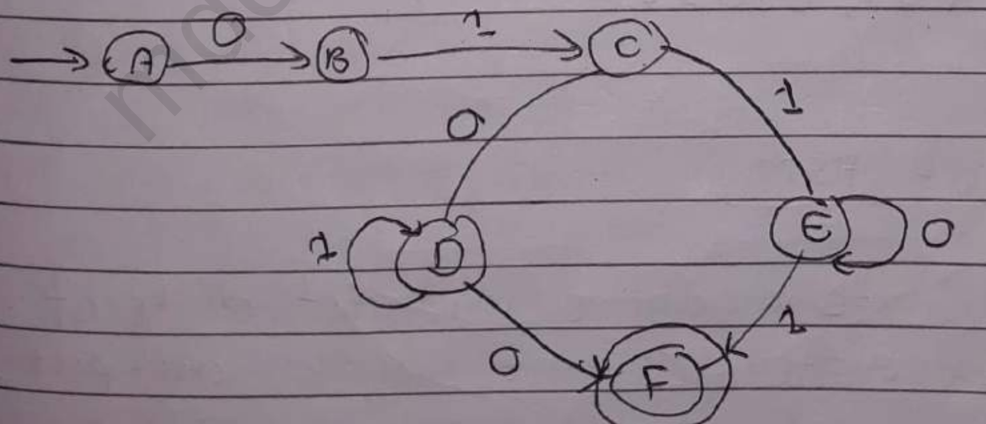
$M = \{Q, I, O, F, G\}$

$I = \{0, 1\}$

$O = \{01, 0100, \ldots\}$



OR,



$O = \{ - - - \cdot \}$

F = consists of

$\delta(A, 0) \rightarrow B$

$\delta(B, 1) \rightarrow C$

$\delta(C, 0) \rightarrow D$

$\delta(C, 1) \rightarrow E$

$\delta(D, 1) \rightarrow D$

$\delta(E, 0) \rightarrow E$

G consists of

$(A, 0) \rightarrow (0, B)$

$(B, 1) \rightarrow (1, C)$

$(C, 0) \rightarrow (0, D)$

$(D, 0) \rightarrow (0, F)$

$(C, 1) \rightarrow (1, E)$

$(E, 1) \rightarrow (1, F)$

$O = \{01, 010110 \ldots\}$

Types of FSM

(i) Finite state machine without output
(ii) Finite state machine with outputs
↳ finite state automata.

Finite state Automata

→ A finite state automata is a mathematical model used to determine whether a string is accepted or not.

→ Due to this reason it is also called language recognizer.

→ Mathematically FSA is defined as:

$$M = \{Q, \Sigma, \delta, q_0, F\}$$

where,

Q = Finite set of states

Σ = finite set of inputs

δ = transition function that takes two argument and returns an argument

The transition function is of the form

$$Q \times \Sigma \rightarrow Q$$

e.g $\delta(q_0, 1) \rightarrow q_2$

q_0 = initial state or starting state.

F = finite set of final state or accepting state.

Eg

FSA

$M = \{Q, \Sigma, \delta, q_0, F\}$

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

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

δ consists of :-

$\delta(q_1, 0) \rightarrow q_1$

$\delta(q_1, 1) \rightarrow q_2$

$\delta(q_2, 0) \rightarrow q_3$

$\delta(q_2, 1) \rightarrow q_2$

$\delta(q_3, 0) \rightarrow q_2$

$\delta(q_3, 1) \rightarrow q_2$

$q_0 = q_1$

$F = q_2$

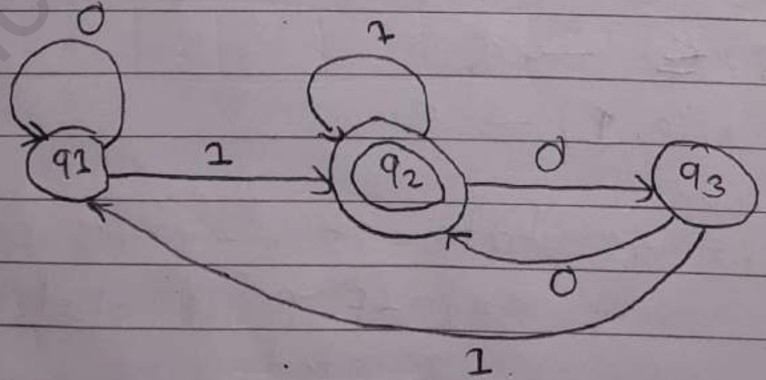


fig :- Finite state Automata

Transition Diagram and Transition Table

→ The representation of FSA can be done using transition diagram and transition table.

a) Transition Diagram

→ It is represented using the weighted directed graph where states are represented by vertices.

→ Transition from one state to another is represented directed graph.

→ value given to each edge is its input.

→ starting state are represented by single circle by pointing an arrow head and final state is represented by double circle.

b) Transition Table

→ It is a tabular representation of transition function of finite automata

→ Generally states are arranged in rows and inputs are arranged in column

→ The intersection of each row and column i.e. each cell represents the next state.

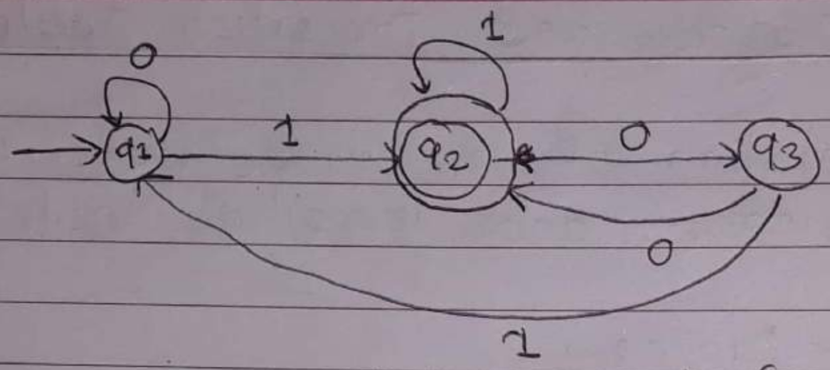


fig. finite state Automata (Transition Diagram)

Q/Σ	0	1
q_1	q_1	q_2
q_2	q_2	q_3
q_3	q_2	q_1

fig:. Transition table of a finite automata.

Processing a string by FA

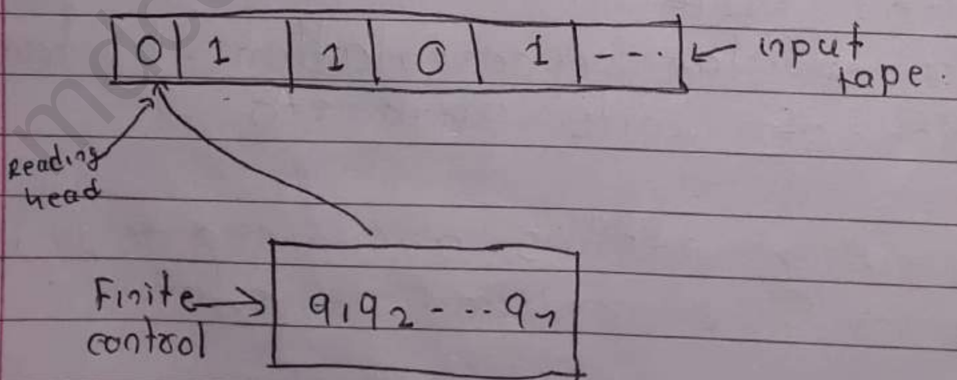


fig:- Block diagram of FA.

classmate

Date ____

Page ____

→ The processing of FA consists of
① Input tape
② Reading head
③ finite control

→ Input tape of FA (finite automata) is responsible for storing the symbol of input string. It is divided into number of square cell, each cell stores the symbol of input string.

→ Reading head scans the symbol from the input tape, each cell at a time in one direction only, either from left to right or right to left.

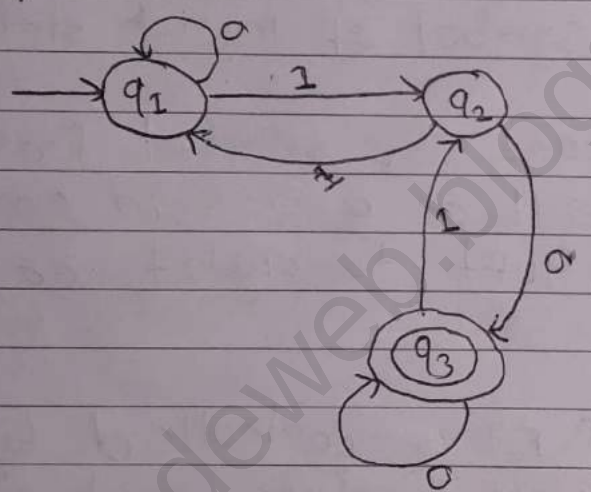
→ The processing of FA is controlled by finite control i.e. ~~the~~ current state, next state obtained after each transition function must be defined in the finite control.

→ Therefore, we say that finite control is also responsible for giving next state regularly.

→ Initially the reading head is placed at the left most cell of input tape and then it scans each cell in the defined direction

→ when the string becomes empty and if the finite control gives only of the defined ~~final~~ final states as the next state, the string is said to be accepted otherwise rejected.

Consider the following FA and check whether the string $w = 001110100$ is accepted or rejected.



solution

$\delta(q_1, 0) \rightarrow q_1$

$\delta(q_1, 1) \rightarrow q_2$

$\delta(q_2, 0) \rightarrow q_3$

$\delta(q_2, 1) \rightarrow q_1$

$\delta(q_3, 0) \rightarrow q_3$

$\delta(q_3, 1) \rightarrow q_2$

Given string

$w = 001110100$

$\delta(q_1, \cancel{01}\ \cancel{0}01110100) \rightarrow (q_1, 01110100)$

$\rightarrow (q_1, 1110100)$

$\rightarrow (q_2, 110100)$

$\rightarrow (q_1, 10100)$

$\rightarrow (q_2, 0100)$

$\rightarrow (q_3, 100)$

$\rightarrow (q_2, 00)$

$\rightarrow (q_3, 0)$

$\rightarrow (q_3, \epsilon)$

$\rightarrow \{q_3\}$

Hence, q_3 is the final state of the provided FA so the given string $w = 00110100$ is accepted by given automata.

Types of finite Automata

(i) Deterministic finite Automata (DFA)

(ii) Non-Deterministic finite Automata (NDFA)

(a) DFA

→ In DFA for each input symbol, one can determine the state to which the machine will move.

→ similar to FSA

$M = (Q, \Sigma, \delta, q_0, F)$

Q = set of finite number of states
Σ = set of finite number of inputs
δ = transition function of the form
$Q \times \Sigma \rightarrow Q$

q = starting state
F = set of final states.

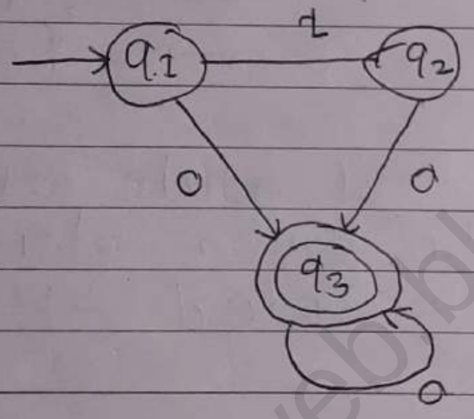


fig :- DFA.

ⓑ NDFA

In NDFA for a particular input symbol the machine can move to any combination of states in the machine

→ In other word the exact state to which the machine moves cannot be determined finite automata.

→ formal definition of NDFA
$M = (Q, \Sigma, \delta, q_0, F)$

The transition function is

$$Q \times \Sigma \to 2^Q$$

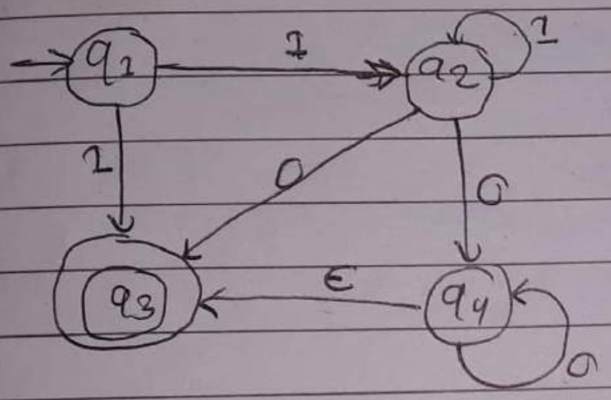


fig :- NDFA

NFA to DFA conversion

→ we use sub-set construction method to convert in NFA into DFA

steps

→ construct a transition table of given NFA
→ Identify all the new states in term of input statement.
→ find the transition for each new states in term of input symbols
→ The process is continued until transition for all the new states are identified.
→ finally draw a transition diagram by using all states obtained.

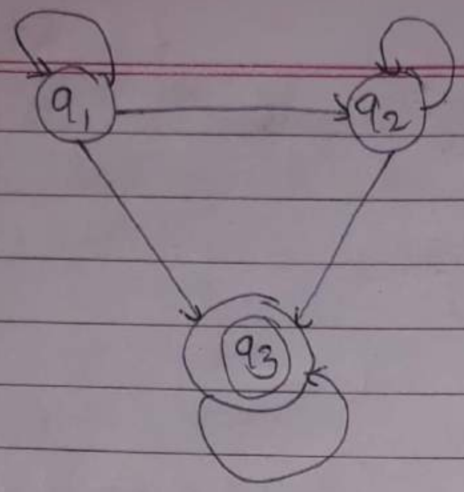


fig. NFA

Regular Expression :-

→ An expression used to generate string for a finite automata is called regular expression.

→ RE is also called language generator.

→ A Regular Expression is recursively defined as

ⓐ Empty state (ϕ), empty string (ε) and symbol of input alphabet are regular expression.

ⓑ Let R_1 and R_2 be regular expression then union of R_1 and R_2 denoted by $R_1 + R_2$ is also RE.

ⓒ Let R_1 and R_2 be regular expression then concatenation of R_1 and R_2 denoted by $R_1 . R_2$ is also RE.

→ Let R be regular expression then Kleen closure of R denoted by R^* is also regular expression.

eg:-

(i) $(0+1)^* = \{\epsilon, 0, 1, 00, 101, 0010 ----\}$

(ii) $0^* + 1^* = \{\epsilon, 1, 00, 111, 0000, ----\}$

(iii) $(01)^* = \{\epsilon, 01, 0101, 010101, ----\}$

(iv) $0^* . 1^* = \{\epsilon, 0, 1, 01, 0011, 0111, ----\}$

(v) $1.1^* = \{1, 11, 1111, ----\}$

\# Write a RE that generates the string that starts with 'a' and ends with 'bb' over {a, b}.

Soln

$RE\ (a+bb)^* = \cancel{\epsilon}, a, bb, \cancel{aa}, \cancel{ab}\ aaabb$

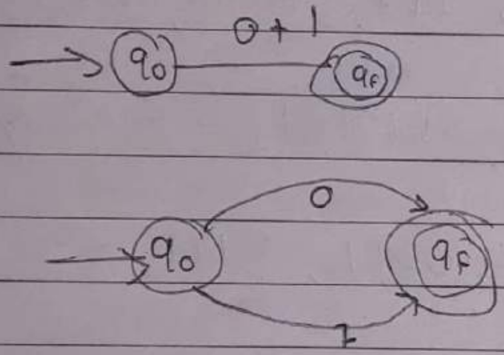
$RE = a(a+b)^* b$

RE to FA

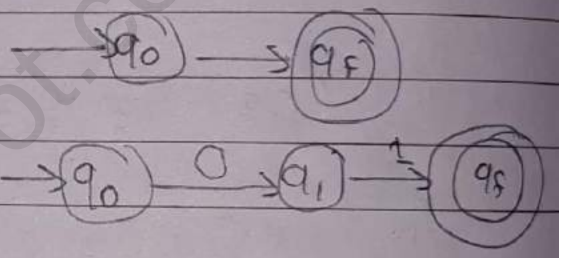
→ For every RE there exist a path from initial state to final state such that path is labelled with that regular expression.

E.g $\Sigma = \{0, 1\}$

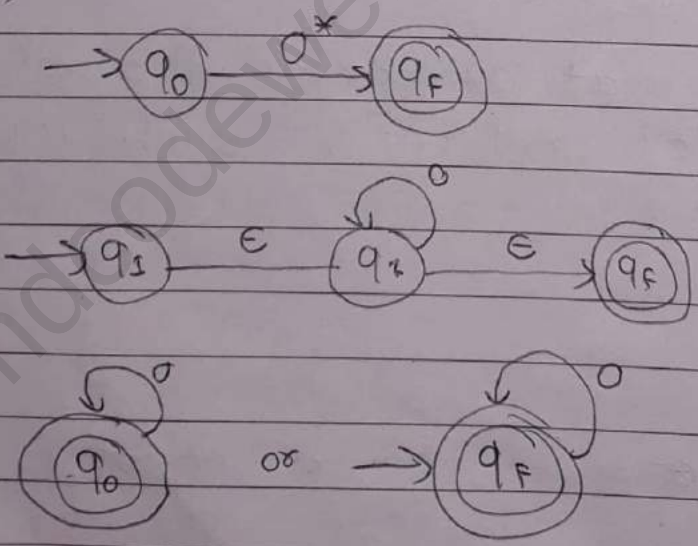
RE → 0+1



RE 0.1



iii) RE → 0*



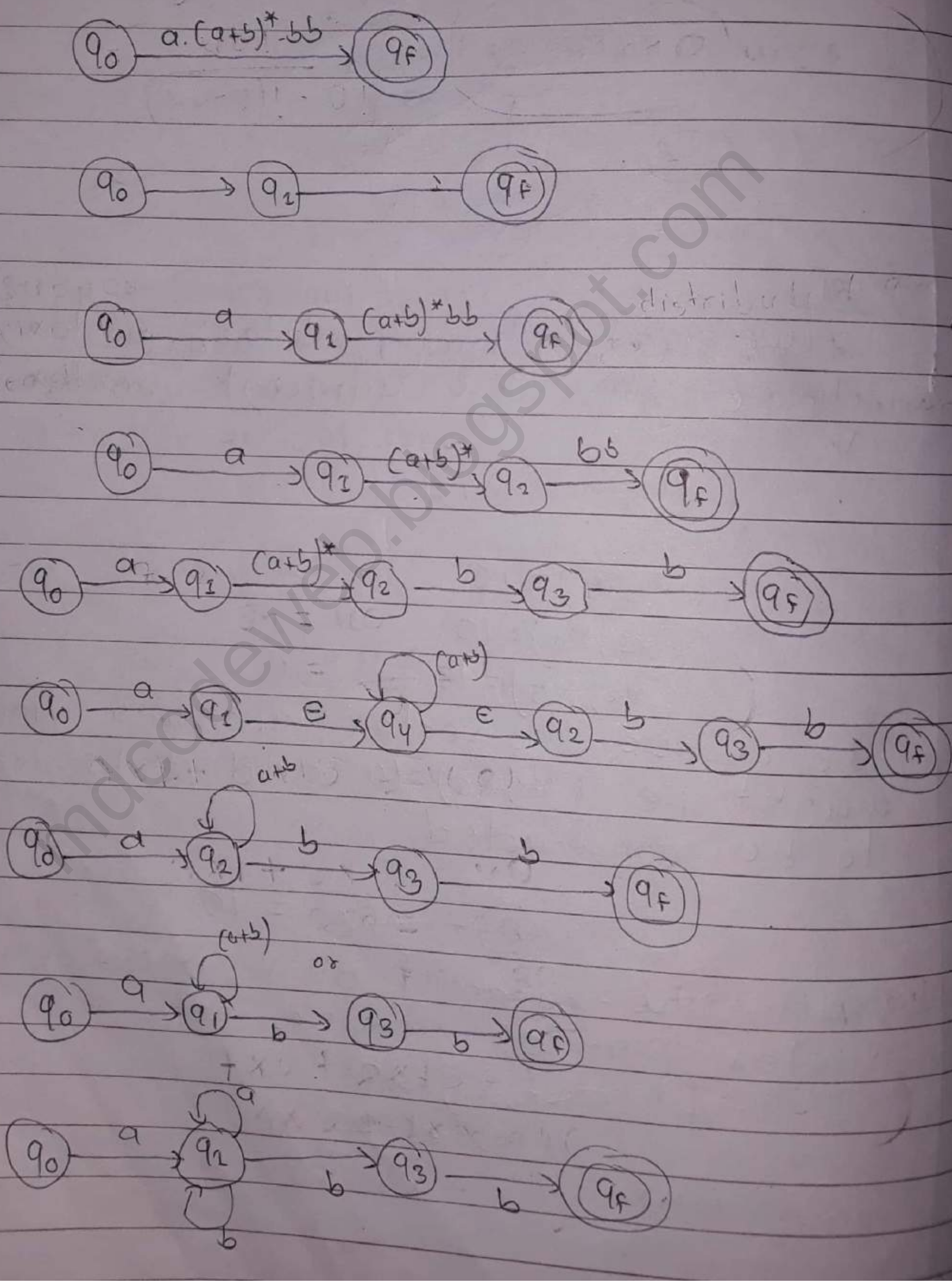
Draw FA for following RE

(a) a (a+b)* bb

(b) 01 (10+11)* 1

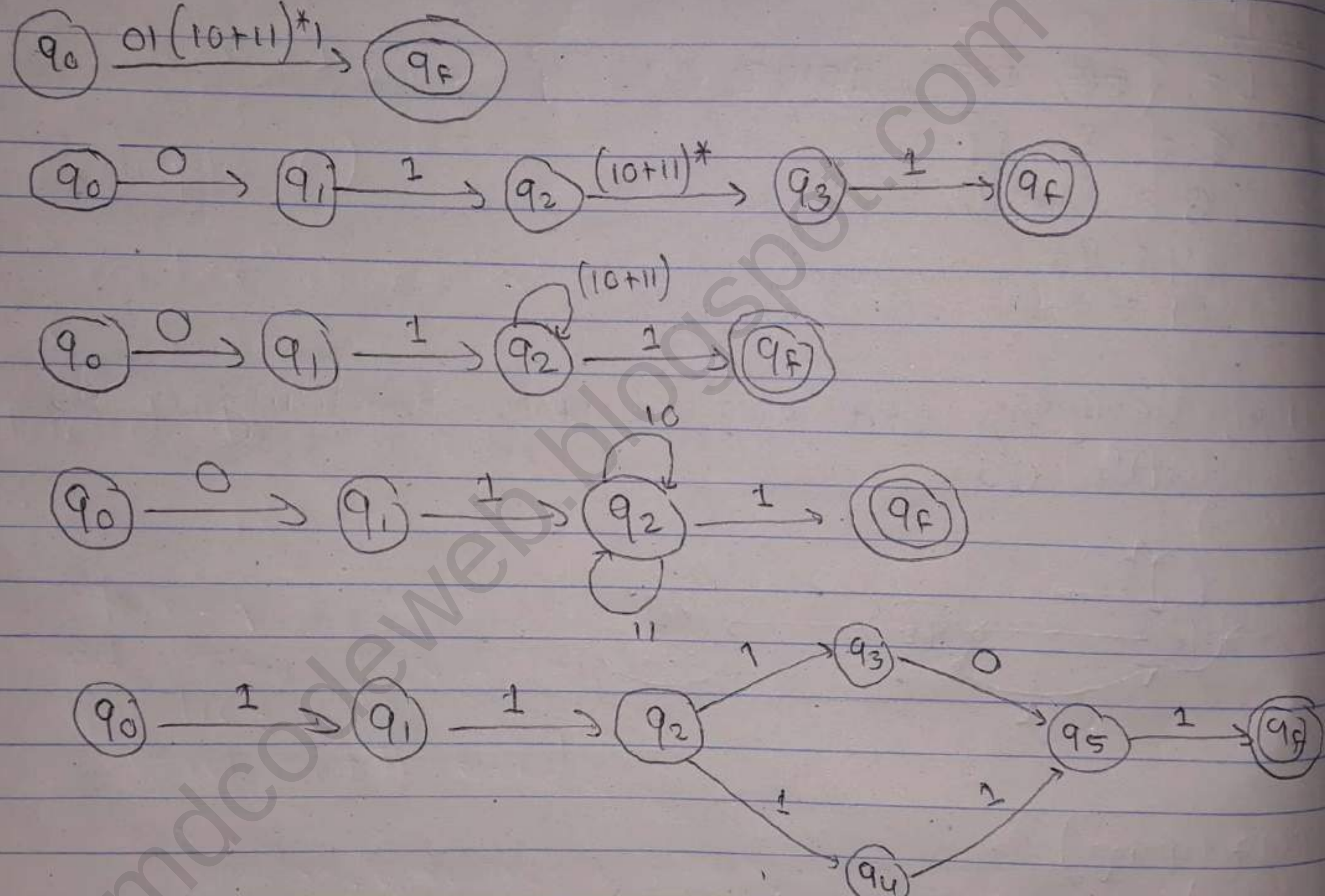
(a) a(a+b)* bb

soln



6.b

$01(10+11)^*1$



Types of Grammar (Chomsky Hierarchy)

1. Unrestricted grammar
2. context sensitive grammar
3. context free grammar
4. Regular grammar

→ If no restriction is applied to the production rule of a grammar then it is called ~~universe~~ unrestricted grammar.

→ A grammar is said to be context sensitive if it's production rule is of the form

$$w_1 \alpha w_2 \rightarrow w_1 \beta w_2$$

where w_1 and w_2 are called context of α and β and $\alpha, \beta \in V$
and w_1 and $w_2 \subset (V \cup \Sigma)^*$, string of terminals and non-terminals.

eg. $AB \rightarrow AbB$
$A \rightarrow bcA$
$B \rightarrow b$

→ A grammar is said to be context free if it's production rule is of the form

$$\alpha \rightarrow \beta$$

where $\alpha \in V$ and $\beta \in (V \cup \Sigma)^*$

eg:-

$A \rightarrow a / aB$

$B \rightarrow AC$

→ A grammar is regular if its production rule is of the form

non-terminal → exactly one terminal

OR,

Non termal → exactly one terminal follow by one non-terminal.

eg:-

$S \rightarrow a$

$S \rightarrow aA$

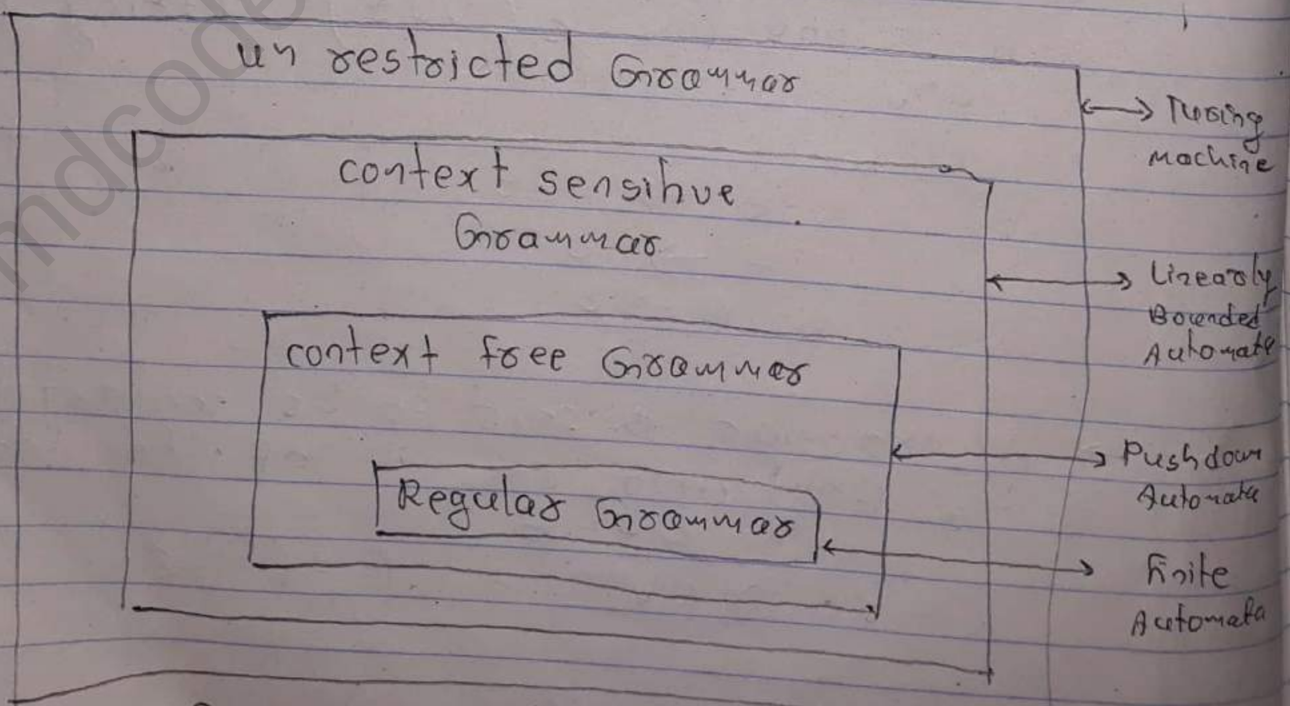


fig:- Grammar with its corresponding mathematical model.

Derivation

- The process of generating a string by using sequences of production rules is called derivation
- It is also known as Parsing

Types of Derivation

① Left most derivation
(ii) Right most derivation.

→ In left most derivation at each step, the production rules for left-most non-terminal is used, whereas in right most derivation at each step, the production rules for right most non-terminal is used.

consider a Grammar

$G = (V, \Sigma, R, S)$
$V = \{S, A, B\}$

$\Sigma = \{a, b\}$
R consists of
$S \rightarrow aAS \mid aS \mid b$
$A \rightarrow bB \mid a$
$B \rightarrow aA \mid b$

starting symbol = S
w = abaaab

LMD

$S \rightarrow aAS$ [$\because S \rightarrow aAS$]
$\rightarrow abBS$ [$\because A \rightarrow bB$]
$\rightarrow abaAS$ [$\because B \rightarrow aA$]
$\rightarrow abaaS$ [$\because A \rightarrow a$]
$\rightarrow abaaaS$ [$\because S \rightarrow aS$]
$\rightarrow abaaab$ [$\because S \rightarrow b$]

RMD

$S \rightarrow aAS$ [$\because S \rightarrow aAS$]
$\rightarrow aAaS$ [$\because S \rightarrow aS$]
$\rightarrow aAab$ [$\because S \rightarrow b$]
~~$\rightarrow aAab$~~
$\rightarrow abBab$ [$\because A \rightarrow bB$]
$\rightarrow abaAab$ [$\because B \rightarrow aA$]
$\rightarrow abaaab$ [$\because A \rightarrow a$]

Derivation Tree

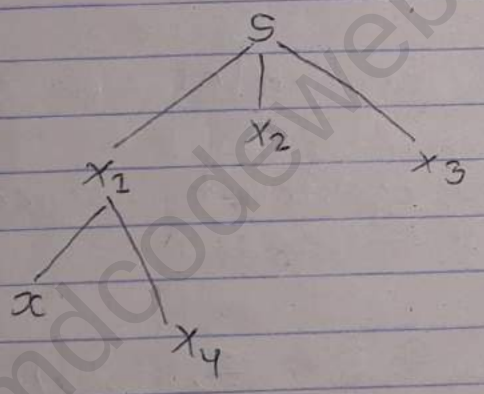
→ The hierarchical representation of derivation is called derivation tree.

→ Left hand side of a production rule is a root ~~node~~ node at each level and right hand side of a production rule is divided into multiple branches.

eg :-

$S \rightarrow X_1 X_2 X_3$

$X_2 \rightarrow x X_4$



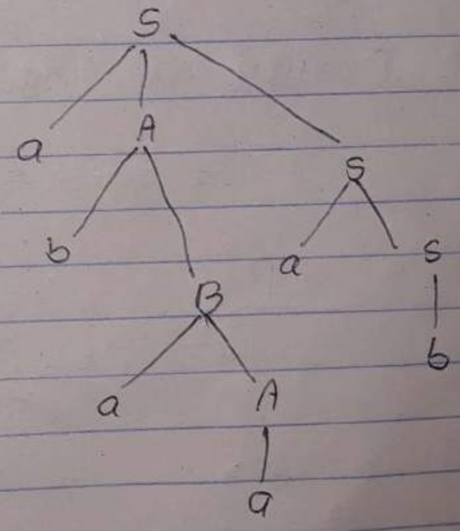


fig :- Derivation tree of LMD

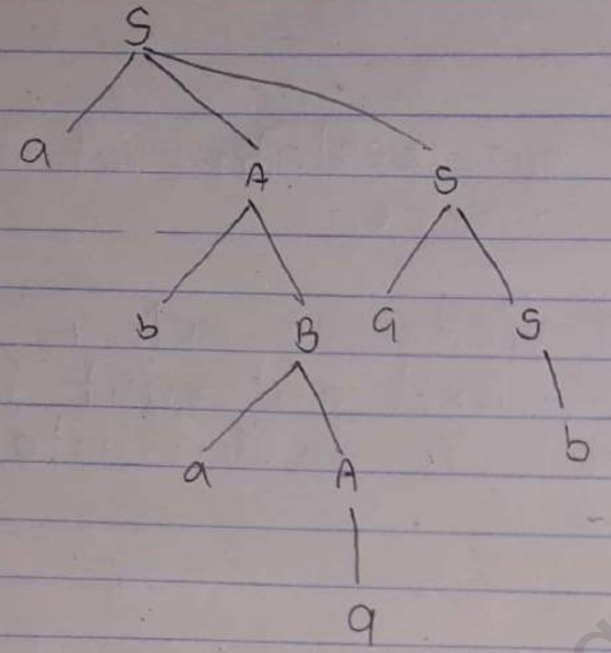


fig:- Derivation tree of RMD

Design a Grammar that generates a string.

w= a*b+b*a . Also construct a derivation tree.

sol^n

Let the grammar be G (V, Σ, R, S)

V: {S, E}

Σ = {a, b}

R consists of :

~~S→ S+S | S* S | a | b~~

S → E+E | E*E

E → a | b | s

starting symbol = S

LMD

$S \to E + E$ [$S \to E + E$]

$\to S + E$ [$E \to S$]

$\to E * E + E$ [$S \to E * E$]

$\to a * E + E$ [$\because E \to a$]

$\to a * b + E$ [$\because E \to b$]

$\to a * b + S$ [$\because E \to S$]

$\to a * b + E * E$ [$S \to E * E$]

$\to a * b + b * E$ [$E \to b$]

$\to a * b + b * a$ [$E \to a$]

LMD

$S \to E * E$ [$\because S \to E * E$]

$\to a * E$ [$\because E \to a$]

$\to a * S$ [$\because E \to S$]

$\to a * E + E$ [$\because S \to E + E$]

$\to a * b + E$ [$\because E \to b$]

$\to a * b + S$ [$\because E \to S$]

$\to a * b + E * E$ [$S \to E * E$]

$\to a * b + b * E$ [$E \to b$]

$\to a * b + b * a$ [$E \to a$]

Ambiguous Grammar

→ If there exist multiple LMD or multiple RMD for the same string then the grammar is said to be ambiguous.

(1)

Sol^n

Let G (V, Σ, R, S) be a grammar that generates palindrome string over Σ {0, 1}

$V = \{S\}$

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

R consists of :-

$S \to 0S0 \mid 1S1 \mid 1 \mid 0 \mid \epsilon$

starting symbol = S

Let G(V, Σ, R, S) be a grammar that generates palindrome string over Σ {0, 1} of even length

$V = \{S\}$

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

R consists of :-

$S \to 0S0 \mid 1S1 \mid \epsilon$ only even

starting symbol = S

$S \to 0S0 \mid 1S1 \mid 1 \mid 0$ only odd