Finite Automata:

It is a state machine that comprehensively captures all possible states and transitions that a machine can take while responding to a stream of input symbols. It is of two types

1) Deterministic finite automata (DFA) : The machine can exist in only one state at any given time.

A DFA consist of

$Q \rightarrow$ A finite set of states

$\Sigma \rightarrow$ A finite set of input symbols

$q_0 \rightarrow$ A start state

$F \rightarrow$ Set of Final states

$\delta \rightarrow$ A transition function, which is a mapping between $Q \times \Sigma \rightarrow Q$

A DFA is defined by the 5 tuple

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

The steps to use a DFA are as follows:

(i) Input a word w in Σ^*

(ii) Start at the 'start state' q_0

(iii) For every input symbol in the sequence w do compute the next state from the current state

(iv) If after all symbol in w are consumed, the current state is one of the final states

(F) Then accept w

(v) Otherwise reject w

21 Design a DFA for the string ending with 01

Sol

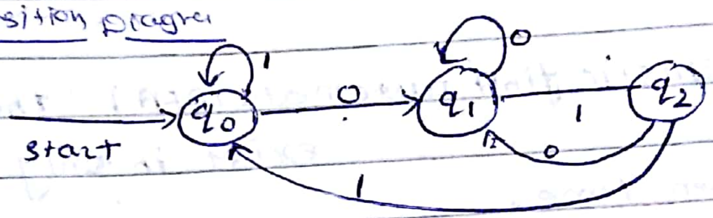
$Q = \{q_0, q_1, q_2\}$

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

start state $= q_0$

$F = \{q_2\}$

Transition Diagram



Transition table:

	δ	0	1
ε	$\rightarrow q_0^*$	q_1	q_0
0	q_1	q_1	q_2
01	q_2^*	q_1	q_0

~~ε0~~ ε1

~~φ0~~ 01

~~010~~ ~~011~~

Q2 Build a DFA for the following language
L = { w/w is a binary string that contains 01 as a substring }

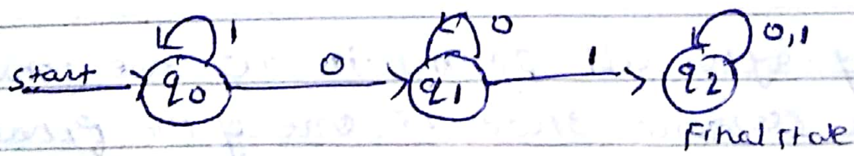
Sol

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

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

start state $= q_0$

$F = \{q_2\}$



Transition table

	δ	0	1
ε	$\rightarrow q_0$	q_1	q_0
0	q_1	q_1	q_2
01	$* q_2$	q_2	q_2

~~ε0~~ ε1

~~φ0~~ 01

~~010~~

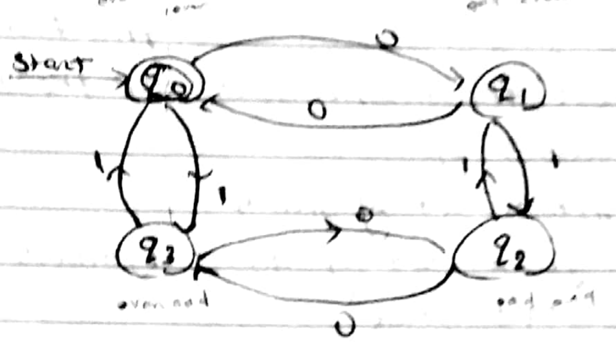
Q3 Build a DFA for even number of 0 and 1

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

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

Start state $= q_0$

$F = \{q_0\}$



Transition Table

δ	0	1
$\rightarrow q_0^*$	q_1	q_3
q_1	q_0	q_2
q_2	q_3	q_1
q_3	q_2	q_0

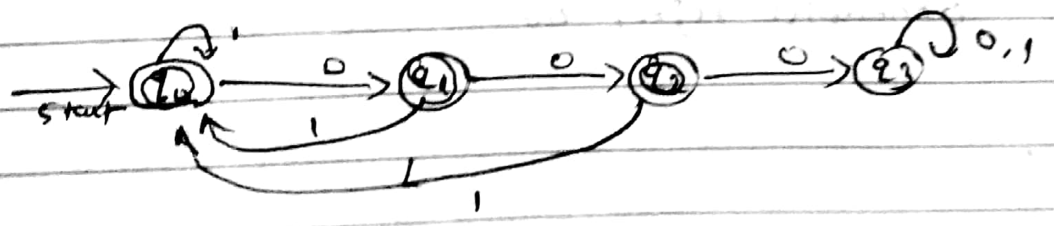
Q Build a DFA for string not having consecutive 3 zeros

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

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

Start Value $= q_0$

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



Transition table

δ	0	1
$\rightarrow q_0^*$	q_1	q_0
q_1^*	q_3	q_0
q_2^*	q_3	q_0
q_3	q_3	q_0

Scanned by CamScanner

The initial state is represented with an arrow and the final state are represented with astericks (*)

(2) Non Deterministic finite Automata :- (NDFA):

The machine can exist in more than one state at the same time.
A NDFA consists of

$Q \to$ A finite set of states

$\Sigma \to$ A finite set of input symbols

$q_0 \to$ A start state

$F \to$ A set of final states

$\delta \to$ A transition function, which is a mapping between $Q \times \Sigma \to$ subset of Q

The steps are:

(i) Input a word w in Σ^*

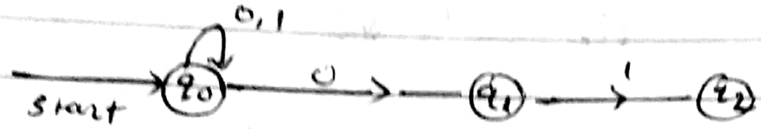
(ii) Start at the start state q_0

(iii) For every input symbol in the sequence w do determine all the possible next states from the current state,

(iv) If after all symbol in w are consumed, at least one of the current states is a final state, then accept w

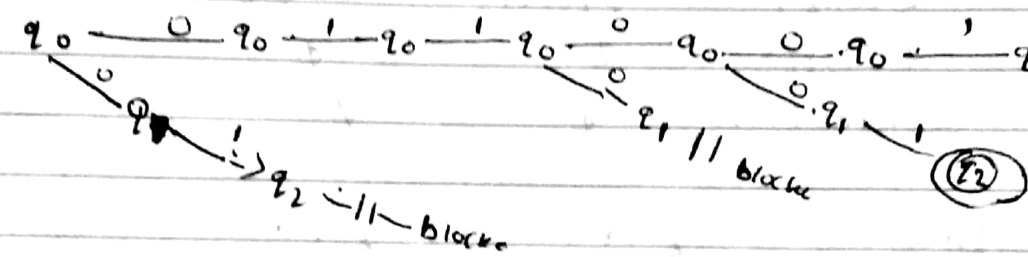
(v) otherwise reject w

Q5 Build a NDFA for string ending with 01



It is non-deterministic because if we get 0 we can loop into q_0 only or we can go to q_1 so, exact location can not be determined

for ex. 01001

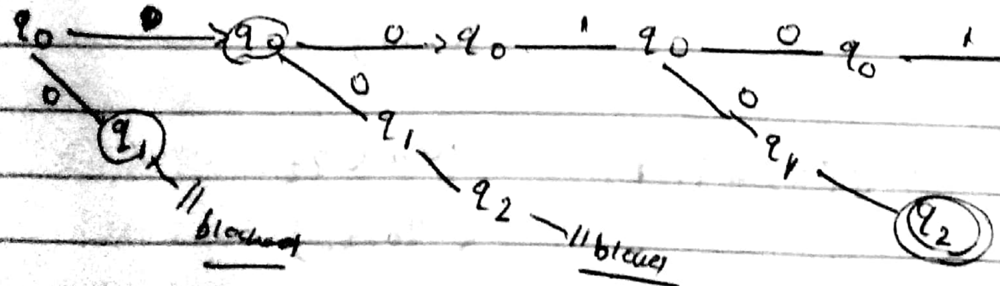


At least on string should reach the final stage

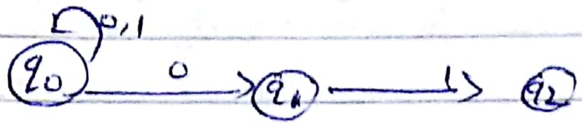
δ	0	1
$\rightarrow q_0^*$	$\{q_0, q_1\}$	$\{q_0\}$
q_1^*	ϕ	$\{q_2\}$
q_2^*	ϕ	ϕ

(*) NFA $\rightarrow$ n state
DFA $\rightarrow$ 2^n state (worst case)

ex 00101

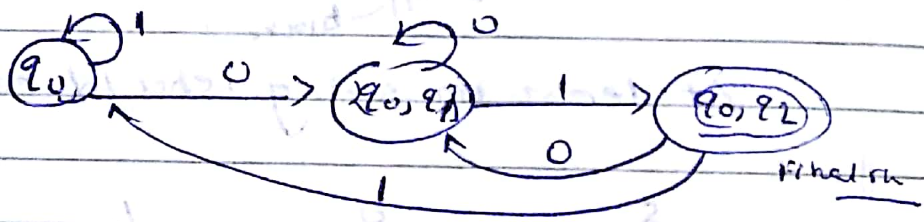


equation NFA and DFA (subset constitution,



δ	0	1
ϕ	ϕ	ϕ
q_0	$\{q_0, q_1\}$	q_0
q_1	ϕ	q_2
q_2	ϕ	ϕ
$\{q_0, q_1\}$	$\{q_0, q_1\}$	$\{q_0, q_2\}$
$\{q_0, q_2\}$	$\{q_0, q_1\}$	q_0
$\{q_1, q_2\}$	ϕ	ϕ
$\{q_0, q_1, q_2\}$	ϕ	ϕ

Diagram



$NFA = \{Q_N, \Sigma, \delta_N, q_0, F_N\}$
$DFA = \{Q_D, \Sigma, \delta_D, q_0, F_D\}$

$\delta_D(q_0, 0) = \{q_0, q_1\}$
$\delta_D(q_0, 1) = \{q_0\}$

New state

$\delta_D\{\{q_0, q_1\}; 0\} = \delta_N(q_0; 0) \cup \delta_N(q_1, 0)$
$= \{q_0, q_1\} \cup \{\phi\}$
$= \{q_0, q_1\}$

New

$\delta_D\{(q_0, q_1), 1\} = q_0 \cup q_2$
$= \{q_0, q_2\}$

Scanned by CamScanner

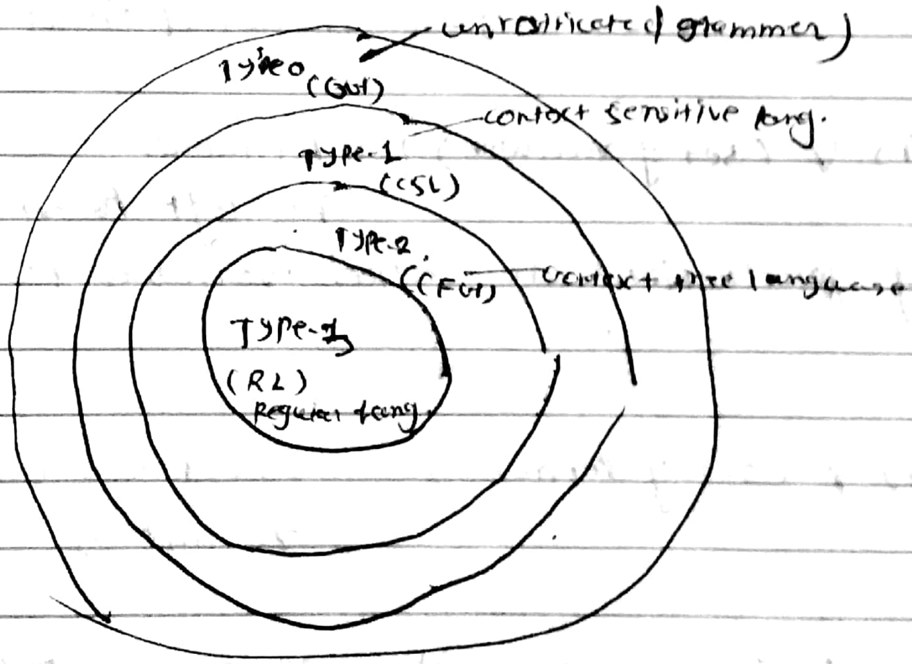
$\delta_n(\{q_0, q_2\}, 0) = \{q_0, q_1\} \cup \{\phi\}$
$= \{q_0, q_1\}$

$\delta_n(\{q_0, q_2\}, 1) = \{q_0\} \cup \phi$
$= \{q_0\}$

prob convert following NFA into DFA
Design an NFA for the string whose substring is 011 over (0,1)

pr Design DFA for mod 3 counter

language: Any language which is accepted by finite automata state is called regular



This hierarchy is called Chomsky Hierarchy
Type-3 - RG · (finite Automata)
Type 2 CFG (push down Automata)
Type-1 CSG (linear bound Automata)
Type-0 UG

Scanned by CamScanner

Context free Grammer:

Context free language.

represents whatever we are able to count

$L = \{0^n 1^n \mid n \geq 0\}$

$= 00001111$ (say)

The regular language are a part of context free language

Context free language are represented by the following

(i) Production (P):-

$S \to \epsilon$

$S \to 0$

$S \to 1$

$S \to 0S0$

$S \to 1S1$

(ii) S (start symbol):- The left side variable of $S \to \epsilon$ is called Start variable

(iii) V (set of variable) non terminal:- The left hand sides of productions are called variables (V)

(iv) T (set of terminal):- $\epsilon, 0, 1$ are called terminals

(v) A context free grammer can be represented by 4-tuples i.e (V, T, P, S)

(v) $S \to \epsilon / 0 / 1 / 0S0 / 1S1$

Scanned by CamScanner

ex 01100110

S ⇒ 0S0 [∵ S → 0S0

⇒ 01S10 [∵ S → 1S1

⇒ 011S110 [∵ S → 1S1

⇒ 0110S0110 [∵ S → 0S0]

⇒ 01100110 [∵ S → ε

Rules for consider

① Non terminal

* Italic capital. Start of the alphabet A, B, C

* Start symbol S

② Terminal :-

* lowercase symbol (a, b, c,

* digit (0, 1, 2)

* operators (+, -, * /)

* punctuation symbols (! ; , " ^ :

Generally u, v, w, x, y, z are group of terminal

ex- a, b, c, a+b

w = abc

④ X, Y, Z (grammer symbols). either terminal or non terminal

⑤ α, β, γ are string of grammer symbol ex α = XY ex $L = \{0^n 1^n \mid n > 0$

$\therefore$ $S \to \epsilon$

$S \to 0S1$

If LHS is same then we can combine them as

$S \to \epsilon \mid 0S1$

ex $a, b, 01$

$((a01 + bab) * a0)$

The production are

$I \to a \mid b \mid Ia \mid Ib \mid I0 \mid I1$

$E \to I \mid E+E \mid E*E \mid (E)$

Sol.

Variable $V = \{I, E\}$

$T = \{a, b, 0, 1, +, *,) : (\}$

$S \to E$

$E \Rightarrow (E)$

$\Rightarrow (E*E)$

$\Rightarrow ((E)*E)$

$\Rightarrow ((E+E)*E)$

$\Rightarrow ((I+E)*E)$

$\Rightarrow ((I1+E)*E)$

$\Rightarrow ((I01+E)*E)$

$\Rightarrow ((a01+E)*E)$

$\Rightarrow ((a01+I)*E)$

$\Rightarrow ((a01+Ib)*E)$

$\Rightarrow ((a01+Iab)*E)$

$\Rightarrow ((a01+bab)*E)$

$\Rightarrow ((a01+bab)*I)$

$\Rightarrow ((a01+bab)*I0)$

$\Rightarrow ((a01+bab)*a0)$

Types of derivation:

① Left most

② Right most

1) Left most:-

when we substitute the left most variable always, it is called left most derivation

2) Right most:- we substitute Right most variable always, it is called Right most derivation

Ex

Left most

$E \rightarrow E+E$

$E \rightarrow E*E$

$E \rightarrow E$

$E \rightarrow a$

$w = a+a*a$

$E \rightarrow E+E$

$\Rightarrow a+E$

$\Rightarrow a+E*E$

$\Rightarrow a+a*E$

$\Rightarrow a+a*a$

Right most:

$E \rightarrow E+E$

$\Rightarrow E+E*E$

$\Rightarrow E+E*a$

$\Rightarrow E+a*a$

$\Rightarrow a+a*a$

parse tree: (Derivation tree):

A tree where in all the variable are terminals and all the internal node are variables.

consider following ex : $a1+b0*ab$

$E \Rightarrow E+E$

$\Rightarrow I+E$

$\Rightarrow I1+E$

$\Rightarrow a1+E$

$\Rightarrow a1+E*E$

$\Rightarrow a1+I*E$

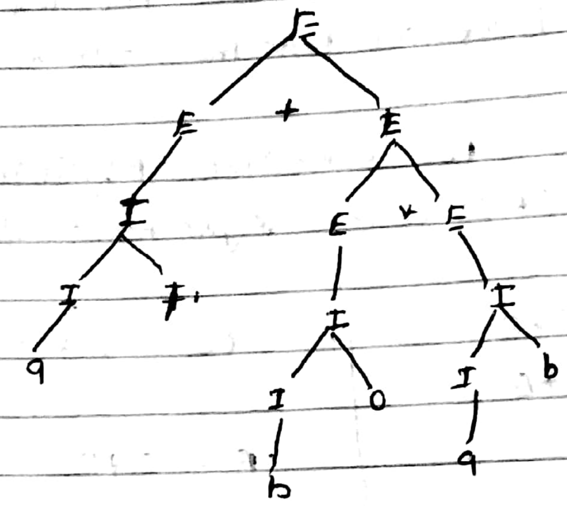
$\Rightarrow a1+I0*E$

⇒ a1 + bo * E

⇒ a1+bo * I

⇒ a1+bo * Ib

⇒ a1+bo * ab



<u>ex</u> E ⇒ E*E

⇒ E+E*E

⇒ I+E*E

⇒ I1 + E*E

⇒ a1 + E*E

⇒ a1 + I*E

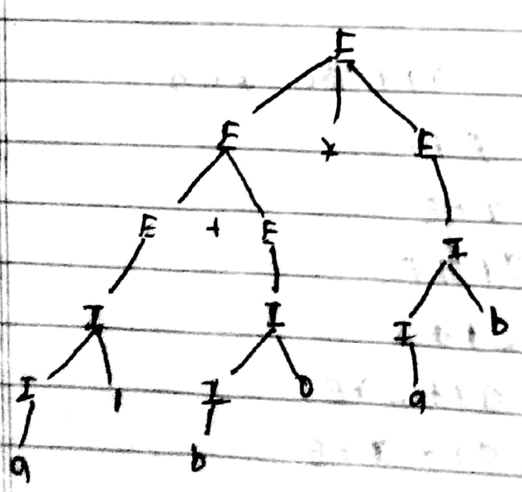
⇒ a1 + IO * E

⇒ a1 + bo * E

⇒ a1+bo * I

⇒ a1+bo * Ib

⇒ a1+bo * ab



Here we are having two different derivation and two different parse trees for the same expression such grammer are called Ambiguous grammer

The Ambiguous form of the same is

$I \rightarrow a | b | Ia | Ib | I0 | I1$

$E \rightarrow E+E | E*E | (E) | I$

Ex write a grammer form:

$L = \{ a^n b^n c^m d^m \mid m, n > 0 \}$

Sol Some of the instance can be

aaabbbcd

abcd

aabbcccddd

$A \rightarrow aAb | ab$

$B \rightarrow cBd | cd$

$S \rightarrow AB$

Ex write a CFG for even palindrom for {a, b}

$S \rightarrow aSa | bSb | \epsilon$

aa

abSba

abba

write a CFG for odd palindrom for {a, b}

$S \rightarrow aSa | bSb | a | b$

Q write a CFG for general palindrom

Useless symbols: A symbol is said to be useful if it is generating and reachable, otherwise it is called useless symbol.

Generating: A symbol is called generating if it is able to generate a string of that particular grammer

* Inductive definition:

Basis: All the terminals are generating

Induction: If $c_1, c_2 \ldots c_k$ are generating and there exists a production.

$A \to c_1 c_2 \ldots c_k$

then A is also generating

Reachable: Starting from the start variable, if we can reach a particular variable, then it is called Reachable

Consider following ex:

$S \to AB \mid a$

$A \to b$

$\because$ a, b are generating (terminals)

Now b is generating. Thus A is generating

($\because A \to b$)

Also a is generating, Thus S is also generating

$\because S \to a$

Since we cannot prove that B is also generating

Thus we have to delete B, for which we have to delete AB

$\therefore S \to a, A \to b$

it

Now starting from S, we can not reach A. Thus A is non-reachable. Thus we delete $A \to b$. Thus we are left with $S \to a$. which is the final grammer

⇒ ε-production:-

production of the form $A \to \epsilon$.

ie. A is nullable

$S \to AB$

$A \to aAA / \epsilon$

$B \to bBB / \epsilon$

Inductive Definition:

Basis:- If there exist $A \to \epsilon$ then A is nullable

Induction:-

Let there exist $B \to c_1 c_2 \ldots c_k$ and we know that all these variables are nullable. (ie after a number of steps all give ε). then $B \to \epsilon$ and hence B is also nullable

∴ we remove ε's

$B \to bBB / bB / b$

$A \to aAA / aA / a$

$S \to AB / A / B$

Unit production:

The production of the type $A \to B$ are called unit production. Here right hand side can contain only one variable.

∴ $A \to B$ is a unit production because B is a variable ~~terminal; not a variable~~

~~Inductive definition~~

$A \to b$ is not a unit production because b is a terminal not a variable

Basis: (A,A) is unit pair

Induction: If we know that (A,B) are unit pair and B→C is a production, then (A,C) is a unit pair

ex
$I \to a/b/Ia/Ib/I0/I1$
$F \to I/(E)$
$T \to F/T*F$
$E \to T/E+T$

(i) Here (E,E) is a unit pair and E→T is a production so (E,T) is unit pair

(ii) (E,T) is a unit pair and T→F is a production so (E,F) is unit pair

(iii) (E,F) is a unit pair and F→I is a production so (E,I) is a unit pair

(iv)

IMP Chomsky Normal form (CNF)

If all the production are in the form A→BC; where B and C are variable, or A→a ie LHS is always a variable and RHS must consist of a string of only 2 variable or a single terminal standing, then the grammar is said to be in CNF

The algorithm to convert any grammar into CNF is as follows

(1) Eliminate ϵ-production

(2) Eliminate unit production

(3) Eliminate useless symbols

(4) Replace all the terminals (along with variables) in the body of the production with variables

ex- $A \to aBC$

$A \to DBE$

$D \to a$

$E \to C$

ex $A \to bc$

$A \to DE$

$D \to b$

$E \to c$

(5) If the body of the production contain string of more than two variables, change it recursively to two-variable string

ex- $A \to \underbrace{C_1 C_2 \cdots C_k}_{X_1}$

$\Rightarrow A \to C_1 X_1$

$X_1 \to \underbrace{C_2 C_3 \cdots C_k}_{X_2}$

$\Rightarrow A \to C_1 X_1$

$X_1 \to C_2 X_2$

$X_2 \to \underbrace{C_3 C_4 \cdots C_k}_{X_3}$

and so on

Imp Mathematical logic

- Solve DNF or CNF

Replace

$p \rightarrow q$ by $\neg p \vee q$

$p \leftrightarrow q$ by $(p \wedge q) \vee (\neg p \wedge \neg q)$

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

make truth table of

$((p \rightarrow q) \wedge (q \rightarrow r)) \rightarrow (p \rightarrow r)$

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

Q without using truth table

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

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

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

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

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

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

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

$\equiv p \wedge q$

2. $p \to (q \to p)$

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

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

$\equiv \neg q \vee (\neg p \vee p)$ by comm

$\equiv \neg p \vee T$

$\equiv T$

Q a) $P \to (Q \to R) \Leftrightarrow (P \wedge Q) \to R$

Sol: $P \to (Q \to R)$

$= P \to (\neg Q \vee R)$ $[\because Q \to R = \neg Q \vee R$

$= \neg P \vee (\neg Q \vee R)$

$= (\neg P \vee \neg Q) \vee R$ (by associative

$\Rightarrow \neg (P \wedge Q) \vee R$ by De morgan's

$= (P \wedge Q) \to R$ by property

hence proved

Q $((p \vee q) \wedge \neg(\neg p \wedge (\neg q \vee \neg r))) \vee (\neg p \wedge \neg q) \vee (\neg p \wedge \neg r)$

is a tautology

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

De morgan

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

De morg

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

by dist

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

by idem and demorgan

$= T$ $[p \vee \neg p = T$

Convert to CNF

Q.8 $S \to bA \mid aB$

$A \to bAA \mid aS \mid a$

$B \to aBB \mid bS \mid b$

Su (i) No ϵ - production

(ii) No unit production

(iii) b is generating ⇒ B is also generating

a is generating ⇒ A is also generating

aB is generating ⇒ S is also generating

∴ All the variables are generating

Also all the variable are reachable i.e

∴ No useless symbol

(iv) $S \to CA \mid DB$

$A \to CAA \mid DS \mid a$

$B \to DBB \mid CS \mid b$

$C \to b$

$D \to a$

(v) $S \to CA \mid DB$

$A \to DS \mid a$

$A \to CX_1$

$X_1 \to AA$

$B \to CS \mid b$

$B \to DX_2$

$X_2 \to BB$

Page No
Date

Q: Convert the following grammar in CNF

$A \rightarrow bAB \mid \epsilon$

$B \rightarrow BAa \mid \epsilon$

S: (i) since both A and B are nullable

$A \rightarrow bAB \mid bA \mid bB \mid b$

$B \rightarrow BAa \mid Ba \mid Aa \mid a$

(ii) No unit production

(iii) No useless symbol

(iv) $A \rightarrow CAB \mid CA \mid CB \mid b$

$B \rightarrow BAD \mid BD \mid AD \mid a$

$C \rightarrow b$

$D \rightarrow a$

(v) $A \rightarrow CA \mid CB \mid b$

$B \rightarrow BD \mid AD \mid a$

$C \rightarrow b$

$D \rightarrow a$

$A \rightarrow CX_1$

$X_1 \rightarrow AB$

$B \rightarrow BX_2$

$X_2 \rightarrow AD$

Now the grammar in CNF

Math logic

Page No.:
Date: / /

Q3 Show that the hypothesis
1) It is not sunny this afternoon and is colder than yesterday

2) We will go swimming only if it is sunny.
3) If we do not go swimming then we will go on a trip
4) If we take a trip, then we will be home by sunset.

Lead to the conclusion "we will home by sunset

S:- A: It is sunny
B: It is colder than yesterday
C: we will go for swimming
D: we will take a trip
E: we will be home by sunsel

$$\left.\begin{array}{ll} 1) & \neg A \wedge B \\ 2. & A \to C \\ 3. & \neg C \to D \\ 4. & D \to E \end{array}\right\} \Rightarrow E \text{ is tautology}$$

Section-1

set:

A set is a collection of well defined object.

A → for set

a, b, c for element

Some standard set:

(1) N = set of all natural number

(2) W = set of all whole no.

(3) I or Z integer

(4) Q → Rational

(5) R - Real number

Equal set:

Fundamental laws of Algebra of sets.

(1) Idempotent law: - If A is any set.

(a) $A \cup A = A$ (b) $A \cap A = A$

proof:

(a) $LHS = A \cup A$

$= \{x : x \in A \cup A\}$

$= \{x : x \in A \text{ or } x \in A\}$

$= \{x : x \in A\} = A$

$= R.H.S.$

(b) $LHS = A \cap A$

$= \{x : x \in A \text{ and } x \in A$

$\{x : x \in A\} = A$

$= R.H.S.$

(ii) Identity law

(a) $A \cup \phi = A$ (b) $A \cap U = A$

(3) Commutative law:

If A and B are any two set

(a) $A \cup B = B \cup A$ (b) $A \cap B = B \cap A$

Proof: (a) LHS $= A \cup B$

$= \{x : x \in A \cup B\}$

$= \{x : x \in A \text{ or } x \in B$

$= \{x : x \in B \text{ or } x \in A$

$= \{x : x \in B \cup A\}$

$= B \cup A$

(b) LHS $= A \cap B$

$= \{x : x \in A \cap B\}$

$\{x : x \in A \text{ and } x \in B$

$= \{x : x \in B \text{ and } x \in B$

$= \{x : x \in B \cap A\}$

$= B \cap A$

$=$ RHS

(4) Associative law:

(a) $A \cup (B \cup C) = (A \cup B) \cup C$

(b) $A \cap (B \cap C) = (A \cap B) \cap C$

Proof: (a) LHS: $= A \cup (B \cup C)$

$= \{x : x \in A \text{ or } x \in B \cup C\}$

$= \{x : x \in A \text{ or } (x \in B \text{ or } x \in C)\}$

$= \{x : (x \in A \text{ or } x \in B) \text{ or } x \in C\}$

$= \{x : x \in A \cup B \text{ or } x \in C\}$

$= \{x : x \in (A \cup B) \cup C\}$

$= (A \cup B) \cup C =$ RHS

5 Distributive laws:

(a) $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$

(b) $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$

(vi) Absorption laws:

(a) $A \cup (A \cap B) = A$ (b) $A \cap (A \cup B) = A$

(vii) Complement law: $\overline{\overline{A}} = A$

(viii) De morgan's law:-

(a) $\overline{A \cup B} = \overline{A} \cap \overline{B}$

(b) $\overline{A \cap B} = \overline{A} \cup \overline{B}$

Proof (a) Let $x \in \overline{A \cup B}$ be arbitrary elem

$\Rightarrow x \notin A \cup B$

$\Rightarrow x \notin A$ and $x \notin B$

$\Rightarrow x \in \overline{A}$ and $x \in \overline{B}$

$\Rightarrow x \in \overline{A} \cap \overline{B}$

$\Rightarrow \overline{A \cup B} \subseteq \overline{A} \cap \overline{B}$ —— (*)

Let $y \in \overline{A} \cap \overline{B}$ be an arbitrary element

$\Rightarrow y \in \overline{A}$ and $y \in \overline{B}$

$\Rightarrow y \notin A$ and $y \notin B$

$\Rightarrow y \notin A \cup B$

$\Rightarrow y \in \overline{A \cup B}$

$\Rightarrow \overline{A} \cap \overline{B} \subset \overline{A \cup B}$ —— (**)

from (*) and (**) we get

$\overline{A \cup B} = \overline{A} \cap \overline{B}$

Page No
Date / /

(4) **Handshaking Theorem or Euler's Theorem.**

The sum of the degree of the vertices of a graph is even.

OR

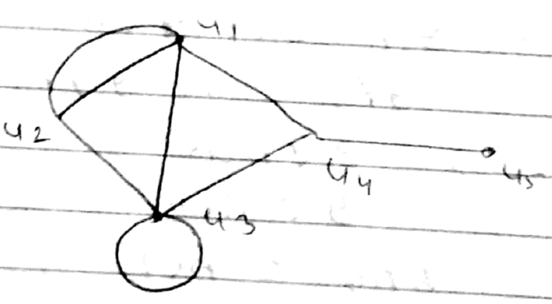
The sum of degree of the vertices in a graph G is equal to twice the number of edges

$$\sum_{v_i \in G} \deg(v_i) = 2e$$, where e is the number of edge

Proof: Let $G=(V,E)$ be a graph with 'n' vertices and 'e' edges

since, each edge is counted twice to find the degree of the vertices of a graph so the sum of the degrees is equal to the twice the number of edges

consider the following example



$\deg(u_1) = 4$

$\deg(u_2) = 3$

$\deg(u_3) = 5$

$\deg(u_4) = 3$

$\deg(u_5) = 1$

$\therefore \sum \deg(u_i) = 16$

No of edges $= 8$

$\therefore$ 2× No. of edges = sum of degree

Page No
Date: / /

(2) **Theorem:** There are even number of vertices in a graph of odd degree

or

The number of odd vertices in a graph G are even

proof let V_1 be the set of odd vertices
V_2 be the set of even vertices in a graph.
so by handshaking theorem

$$\sum_{v_i \in V} \deg(v_i) = 2e$$

$$\therefore \sum_{v_i \in V_1} \deg(v_i) + \sum_{v_i \in V_2} \deg(v_i) = 2e \Rightarrow \text{even}$$

$$\Rightarrow \sum_{v_i \in V_1} \deg(v_i) = 2e - \sum_{v_i \in V_2} \deg(v_i)$$

$$= \text{even} - \text{even}$$

$$= \text{even}$$

$\therefore$ The sum of odd number is even only when they are even in number

Theorem The number of edges in a complete graph with n vertices is $\frac{n(n-1)}{2}$

proof since every vertex in complete graph is joined with every other vertex through one edge.

$\therefore$ the degree of every vertex in a complete graph of n vertices is $n-1$. If e be the total number of edges in G, then by First theorem on graph theory, we have

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

$$n(n-1) = 2e$$

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

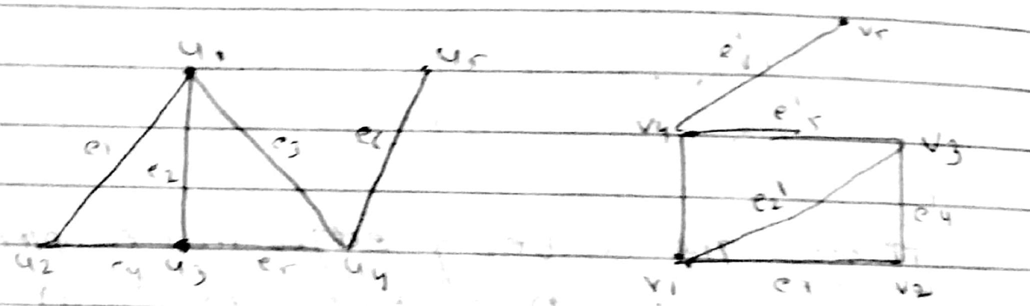
Isomorphic graph:

Let $G=(V,E)$ and $G'=(V',E')$ be two graph. Then G is isomorphic to G', written as $G \cong G'$ if there exists a bijection f, from V to V' such that $(v_i, v_j) \in E$ iff $(f(v_i), f(v_j)) \in E'$

In other words, two graphs are isomorphic if there exists a one-one correspondence between their vertices and edges such that incidence relationship is preserved.

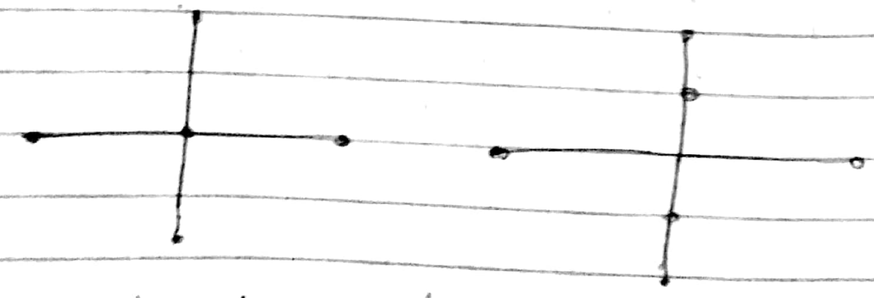
Two isomorphic graph will have

i) same number of vertices
ii) same number of edges
iii) An equal number of vertices with given degree



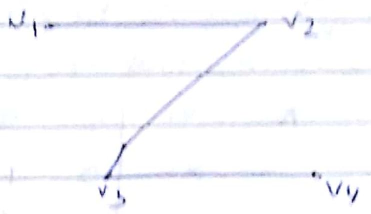
⇒ Homeomorphic graphs-

Given any graph G, obtain a new graph by dividing an edge of G with additional vertices



The above two graph are homeomorphic graph

Bipartite graph:
A graph G is said to be a bipartite graph if its vertices V can be partitioned into two subsets A and B, such that each edge of G connects a vertex of A to a vertex of B



$V = \{v_1, v_2, v_3, v_4\}$
$A = \{v_1, v_3\}$, $B = \{v_2, v_4\}$
∴ The above graph is a Bipartite graph

(2) Adjacency matrix: Let $A = a_{ij}$ be the $n \times n$ matrix defined by

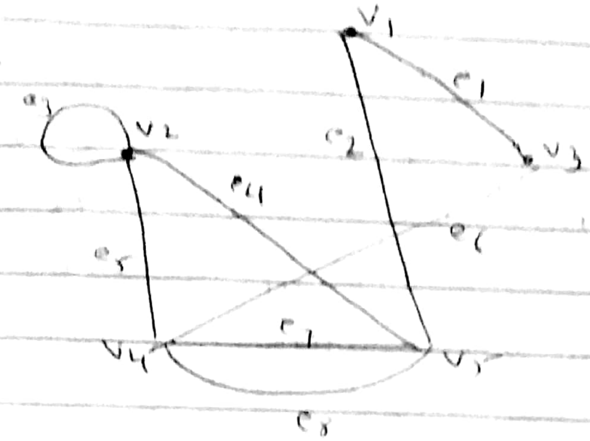
$$a_{ij} = \begin{cases} 1, & \text{if } (v_i, v_j) \text{ is an edge} \\ 0, & \text{otherwise} \end{cases}$$

Then A is called adjacency matrix of G
If we get the same adjacency matrix for two graph after applying some row or column interchanges, then the two graphs are isomorphic

⇒ Incidence matrix: - Let $M = (m_{ij})$ be the $m \times n$ matrix defined by

$$m = \begin{cases} 1, & \text{if the vertex } v_i \text{ incident on edge } e_j \\ 0 & \text{otherwise} \end{cases}$$

Then M is called the Incidence matrix of G ex



Adjacency Matrix

A

	v_1	v_2	v_3	v_4	v_5
v_1	0	0	1	0	1
v_2	0	1	0	1	1
v_3	1	0	0	1	0
v_4	0	1	1	0	2
v_5	1	1	0	1	0

Incidence Matrix

	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8
v_1	1	1	0	0	0	0	0	0
v_2	0	0	1	1	1	0	0	0
v_3	1	0	0	0	0	1	0	0
v_4	0	0	0	0	1	1	1	1
v_5	0	1	0	1	0	0	1	1

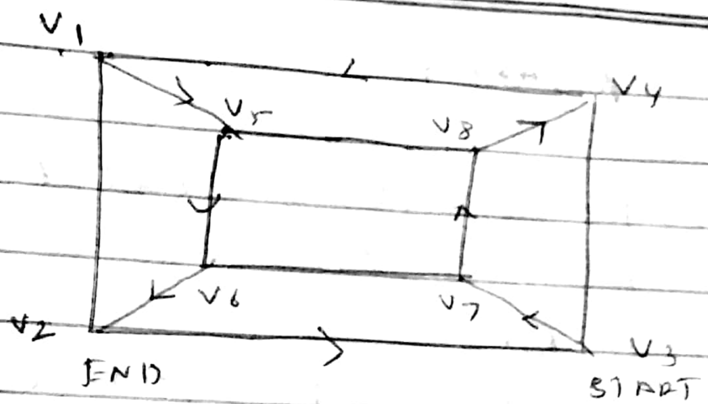
① **Hamiltonian path:-**

It is a path in a graph that contain all the vertices of the graph exactly once. The edges can be repeated

② **Hamiltonian Circuit:**

It is a cycle or a closed path in a graph that contains all vertices of the graph exactly once, except the initial vertex at which the cycle terminates.

TNPL
Page No.
Date / /



Starting at vertex V_3, if we traverse along the edges with arrows, we get hamiltonian circuit.

Relation

when A and B are sets, a subset R of the cartesian product $A \times B$ is called relation from A to B.

R is a set of ordered pair (a,b) where $a \in A$ and $b \in B$. when $(a,b) \in R$, we use notation aRb.

Reflexive: A relation R on a set A is said to be reflexive if aRa for every $a \in A$

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

reflexive relation: $R = \{(1,1), (2,2), (3,3)\}$

Symmetric, Antisymmetric

A relation R on a set A is symmetric if aRb implies bRa. if $(a,b) \in R$ $\Rightarrow$ $(b,a) \in R$. also Antisymmetric if $a=b$

Transitive:

A relation R on set A is said to be transitive if aRb and bRc then aRc

$\rightarrow$ $(a,b) \in R$, $(b,c) \in R$ $\Rightarrow$ $(a,c) \in R$

Congruence Relation:

Let $a, b, m \in Z$, where $m \geq 2$. Then a is congruent to b modulo m, denoted by $a \equiv b \pmod m$, if $a-b$ is divisible by m.

Prove to equivalence R:

1) $a \equiv a \pmod m$ (reflexive property)
2) If $a \equiv b \pmod m$; then $b \equiv a \pmod m$ (symmetric property)
3) If $a \equiv b \pmod m$ and $b \equiv c \pmod m$, then $a \equiv c \pmod m$ (transitive property)

1) $a \equiv a \pmod m$

$\Rightarrow m \mid (a-a)$

$\therefore m$ divide 0

It is universally true.

2) if $a \equiv b \pmod m$

$\Rightarrow m \mid (a-b)$

$\therefore m$ divide $(a-b)$

$\Rightarrow m \mid -(b-a)$

$\Rightarrow m$ divide $-(b-a)$

$\therefore$ So $b \equiv a \pmod m$

Also true.

$\Rightarrow m \times k_1 = a-b$

$\Rightarrow m \times k_1 = -(b-a)$

$\Rightarrow m \times k_2 = b-a$

3) Suppose $a \equiv b \pmod m$ and $b \equiv c \pmod m$

Then $m \mid (a-b)$ and $m \mid (b-c)$

consequently $a-b = mq_1$

$b-c = mq_2$

here q_1, q_2 are integer

$\Rightarrow a-c = m(q_1+q_2)$

$\Rightarrow a-c = q_3 m$

$\Rightarrow a \equiv c \pmod m$

function:

A relation f from set X to another set Y is called a function if for every $x \in X$ there is a unique $y \in Y$ such that $(x, y) \in f$

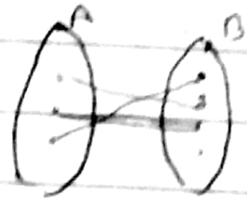
If $y = f(x)$, x is called preimage and y is called image of x under f

One-to one:

A function $f: X \to Y$ is called one-to one (1-1) or injective if distinct element of X are mapped into distinct element of Y.

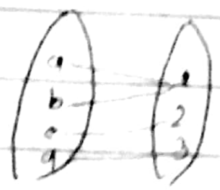
i.e. $f(x_1) \neq f(x_2)$, whenever $x_1 \neq x_2$

$f(x_1) = f(x_2)$ whenever $x_1 = x_2$



Onto:

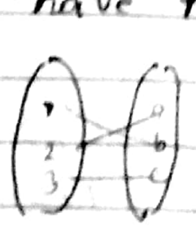
A function $f: X \to Y$ is called onto or surjective. If the range $R_f = Y$. otherwise it is called into



Bijective:

one-one and onto

If X and Y are finite such that $f: X \to Y$ is bijective, then X and Y have the same number of element



Scanned by CamScanner

Theor The necessary and sufficient condition for the function $f: A \to B$ to be invertible is that f is one-to one and onto

Proof (i) Let $f: A \to B$ is invertible.
Then there exist a unique function
$g: B \to A$ such that
$g \circ f = I_A$ and $f \circ g = I_B$

Let $a_1, a_2 \in A$ such that $f(a_1) = f(a_2)$
where $f(a_1), f(a_2) \in B$
Since $g: B \to A$ is a function

$$g\{f(a_1)\} = g\{f(a_2)\}$$
$$(g \circ f)(a_1) = (g \circ f)(a_2)$$
$$I_A(a_1) = I_A(a_2) \quad \text{by (1)}$$
$$a_1 = a_2$$

Thus whenever $f(a_1) = f(a_2)$ we have
$$a_1 = a_2$$
Hence f is one to one

(ii) Let $b \in B$ then $g(b) \in A$ since $g: B \to A$ is a function

Now $b = I_B(b) = (f \circ g)(b)$ by (1)
$= f(g(b))$

Thus corresponding to every $b \in B$ there is a element $g(b) \in A$ such that $f(g(b)) = b$
Hence f is onto
Thus the necessary of property is proved

Q Prove by mathematical induction, that $1^2+3^2+5^2+\cdots+(2n-1)^2 = \frac{1}{3}n(2n-1)(2n+1)$

S Let $S(n) = 1^2+3^2+5^2+\cdots+(2n-1)^2 = \frac{1}{3}n(2n-1)(2n+1)$

when $n=1$

$S(1) = \frac{1}{3}.1.1.3 = 1$

So $S(1)$ is true.

Let $S(n)$ be true for $n=k$

ie $1^2+3^2+5^2+\cdots+(2k-1)^2 = \frac{1}{3}k(2k-1)(2k+1)$

$= 1^2+3^2+5^2+\cdots+(2k-1)^2+(2k+1)^2 = \frac{1}{3}k(2k-1)(2k+1) + (2k+1)^2$

$= \frac{1}{3}\cdot(2k+1)\left[k(2k-1)+3(2k+1)\right]$

$= \frac{1}{3}(2k+1)(2k^2+5k+3)$

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

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

ie $S(k+1)$ is valid

Q Solve the recurrence relation $a_n - 2a_{n-1} = 3^n$, $a_1 = 5$

S The characteristic equation of the recurr. relation:

$m-2=0$

$m=2$

$\therefore a_n^{(h)} = C.2^n$

Since the Right side of the relation is 3^n,
let a particular solution of the relation be
$a_n = A.3^n$
Now using this relation, we get

$$A.3^n - 2.A.3^{n-1} = 3^n$$

$\Rightarrow$ $3A - 2A = 3$
$A = 3$
$a_n^{(p)} = 3^{n+1}$

$\therefore$ General solution is $a_n = a_n^{(h)} + a_n^{(p)}$
$a_n = C.2^n + 3^{n+1}$

Now $a_1 = 5$
$5 = 2C + 9$
$\boxed{C = -2}$

Hence the required solution is $\boxed{a_n = 3^{n+1} - 2^{n+1}}$

(2) Solve $a_n = 2a_{n-1} + 2^n$, $a_0 = 2$
characteristic eq

$m - 2 = 0$
$m = 2$

Since RS of recu rel. is 2^n and 2 is root
let $a_n = An \cdot 2^n$ be a particular integ.

$$An2^n - 2A(n-1)2^{n-1} = 2^n$$
$\Rightarrow$ $An - (n-1) = 1$
$An = n$
$A = 1$

$\therefore$ General solution $a_n = a_n^{(h)} + a_n^{(p)}$
$a_n = C2^n + n2^n$
$a_0 = 2$, $\boxed{2 = C}$

$$\boxed{a_n = (n+2)2^n}$$

Scanned by CamScanner

Q. Solve the recurrence relation. $a_{n+1} - a_n = 3n^2 - n$

$a_0 = 3$

So. The characteristic eq.

$m - 1 = 0$

$m = 1$

$\therefore a_n^{(h)} = C \cdot 1^n = C$

since R.S is $3n^2 - n$.

let particular solution is $a_n = (A_0 n^2 + A_1 n + A_2) n$

since 1 is characteristic root of the R.R, we have

$\{A_0(n+1)^3 + A_1(n+1)^2 + A_2(n+1)\} - (A_0 n^3 + A_1 n^2 + A_2 n) = 3n^2 - n$

ie $A_0(3n^2 + 3n + 1) + A_1(2n+1) + A_2 = 3n^2 - n$

Comparing

$A_0 = 1$, $3A_0 + 2A_1 = -1$

$A_0 + A_1 + A_2 = 0$

$\boxed{A_0 = 1}$, $\boxed{A_1 = -2}$ $\boxed{A_2 = 1}$

$\therefore a_n^{(p)} = n^3 - 2n^2 + n$

$n(n-1)^2$

$\therefore$ General solution

$a_n = a_n^{(h)} + a_n^{(p)}$

$a_n = C + n(n-1)^2$

Now $\boxed{a_0 = 3}$

$\boxed{3 = C}$

$\boxed{a_n = 3 + n(n-1)^2}$

Page No
Date

Q Find a formula for the general term F_n of fibonacci sequence 0, 1, 1, 2, 3, 5, 8, 13

S The recurrence relation corresponding to the fibonacci seque

$F_{n+2} = F_{n+1} + F_n \quad n \geq 0$

with initial condition $F_0 = 0, \; F_1 = 1$

Characteristic eq $m^2 - m - 1 = 0$

$m = \frac{1 \pm \sqrt{5}}{2}$

$F_n = C_1\left(\frac{1+\sqrt{5}}{2}\right)^n + C_2\left(\frac{1-\sqrt{5}}{2}\right)^n$

$F_0 = 0$ gives $C_1 + C_2 = 0 \quad \text{--- (1)}$

$F_1 = 1$ gives $C_1\left(\frac{1+\sqrt{5}}{2}\right) + C_2\left(\frac{1-\sqrt{5}}{2}\right) = 1 \quad \text{--- (2)}$

$C_1 - C_2 = \frac{2}{\sqrt{5}}$

$C_1 = \frac{1}{\sqrt{5}}, \quad C_2 = \frac{-1}{\sqrt{5}}$

∴ The general term of F_n

$F_n = \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^n - \frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^n, \; n \geq 0$

Ex solve $a_n = 2(a_{n-1} - a_{n-2})$, $n \geq 2$, $a_0 = 1$, $a_1 = 2$

So Characteristic eq of R.R is

$m^2 - 2m + 2 = 0$

$m = 1 \pm i$

The modulus amplitude form of

$1 \pm i = \sqrt{2}\left(\cos\frac{\pi}{4} \pm i\sin\frac{\pi}{4}\right)$

Hence general solution is

$a_n = (\sqrt{2})^n\left(C_1\cos\frac{n\pi}{4} + C_2\sin\frac{n\pi}{4}\right)$

Scanned by CamScanner

using condition $a_0 = 1$, $a_1 = 2$

$\boxed{1 = C_1}$

$2 = \sqrt{2}\left(\frac{1}{\sqrt{2}} + \frac{C_2}{\sqrt{2}}\right)$

$\boxed{C_2 = 1}$

Hence $a_n = (\sqrt{2})^n\left(\cos\frac{n\pi}{4} + \sin\frac{n\pi}{4}\right)$

Q. form a recurrence relation satisfied by $a_n = \sum_{k=1}^{n} k^2$ and find the value of $\sum_{k=1}^{n} k^2$ by solving it

$a_n = \sum_{k=1}^{n} k^2$, $a_{n-1} = \sum_{k=1}^{n-1} k^2$

$\therefore a_n - a_{n-1} = n^2$ clearly $a_1 = 1$

characteristic eq $m - 1 = 0$

$m = 1$

$\therefore a_n^{(h)} = C \cdot 1^n = C$

Let P.S be assumed $a_n = (A_0 n^2 + A_1 n + A_2) n$

using R.R. we have

$(A_0 n^2 + A_1 n + A_2) n - \{A_0 (n-1)^2 + A_1 (n-1) + A_2\}(n-1) = n^2$

equating

$A_0 = \frac{1}{3}$, $A_1 = \frac{1}{2}$, $A_2 = \frac{1}{6}$

Hen $a_n^{(P)} = \frac{n}{6}(2n^2 + 3n + 1)$

$= \frac{n}{6}(n+1)(2n+1)$

Hence $a_n = C + \frac{n}{6}(n+1)(2n+1)$

using $a_1 = 1$, $C = 0$

$a_n = \Sigma n^2 = \frac{n(n+1)(2n+1)}{6}$

Scanned by CamScanner

Date / /

(7) Use method of generating function to solve the recurrence relation

$a_n = 3a_{n-1} + 1, \quad n \geq 1$, given $a_0 = 1$

S Let the generating function of $\{a_n\}$ be

$G(x) = \sum_{n=0}^{\infty} a_n x^n$

The given R.R is $a_n = 3a_{n-1} + 1$ — (1)

$\therefore \sum_{n=1}^{\infty} a_n x^n = 3 \sum_{n=1}^{\infty} a_{n-1} x^n + \sum_{n=1}^{\infty} x^n$

on multiplying both side of eq(1) by x^n

$$G(x) - a_0 = 3x\, G(x) + \frac{x}{1-x}$$

$$G(x)(1-3x) = 1 + \frac{x}{1-x}$$

$$G(x) = \frac{1}{(1-x)(1-3x)}$$

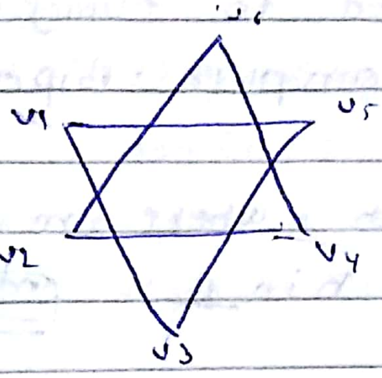
$$= \frac{-1}{2(1-x)} + \frac{3}{2(1-3x)}$$

i.e $\sum_{n=0}^{\infty} a_n x^n = -\frac{1}{2} \sum_{n=0}^{\infty} x^n + \frac{3}{2} \cdot \sum_{n=0}^{\infty} 3^n x^n$

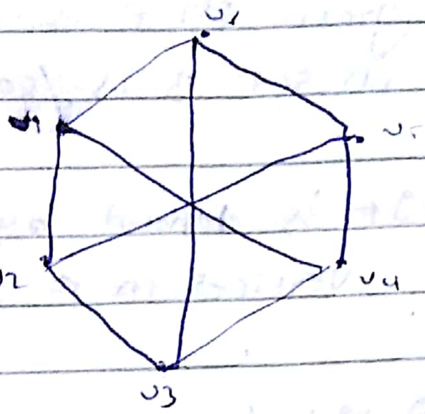
a_n = coeff of x^n in $G(x)$

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

Complement of a graph:



G



$\overline{G}$

If the degree of a vertex 'V' in a simple graph G having n vertices is K, then the degree of vertex V in $\overline{G}$ is $n-K-1$

Euler path:

It is path in a graph that contain all the edges of a graph exactly once. The vertices can be repeated.

Euler graph:

A connected graph that contain an euler circuit is called an euler graph.

A connected graph G is an euler graph iff all the vertices of G are of even degree.

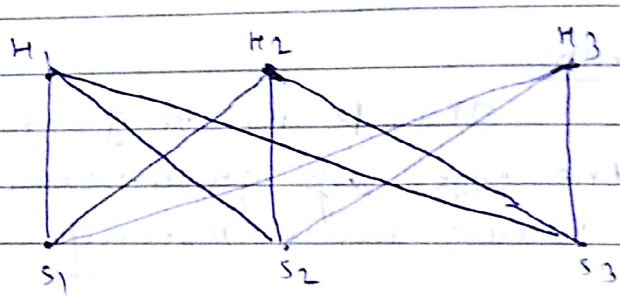
Complete Bipartite graph — A Bipartite graph with two vertex set A and B, where each vertex from set A is connected to every vertex in set B is called complete Bipartite graph.

It is denoted by $K_{m,n}$, where m is the no. of vertices in A, and n in B. $\boxed{m \leq n}$

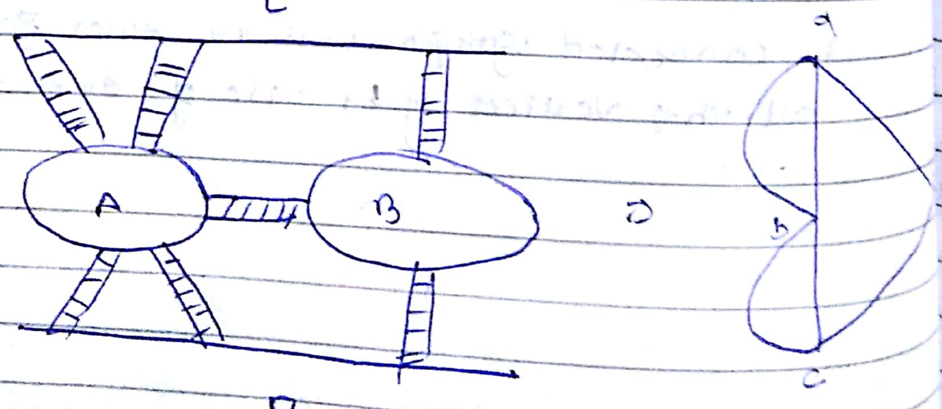
① If there are m vertices in set A and 'n' vertices in set B, then no. of edges in a complete bipartite graph is 'mn'.

K_{33} graph $G(V,E)$

$V = \{H_1, H_2, H_3, S_1, S_2, S_3\}$



→ Konigsberg Bridge Problem:



problems cross each bridge exactly once and return to starting point.

Solution

deg(a) = deg(c) = deg(d) = 3.
deg(b) = 5.

Since the 7 bridge of Konigsberg has 4 vertices of odd degree.

It can not have an euler circuit. Hence it is impossible to walk over each of 7 bridge exactly once and return to starting point.

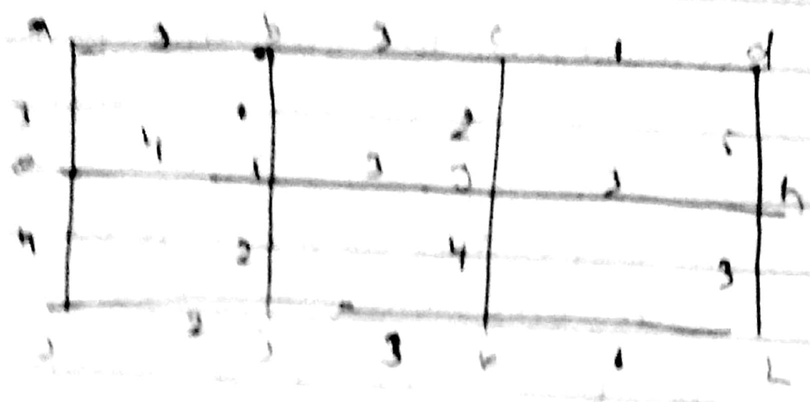
Prim's Algorithm

This algorithm is used to find the shortest spanning tree it involved the following steps

(i) Choose any vertex V_1 of G.
(ii) Choose the edge e_1 of G such that e_1 has the smallest weight among the edges of G incident of V_1

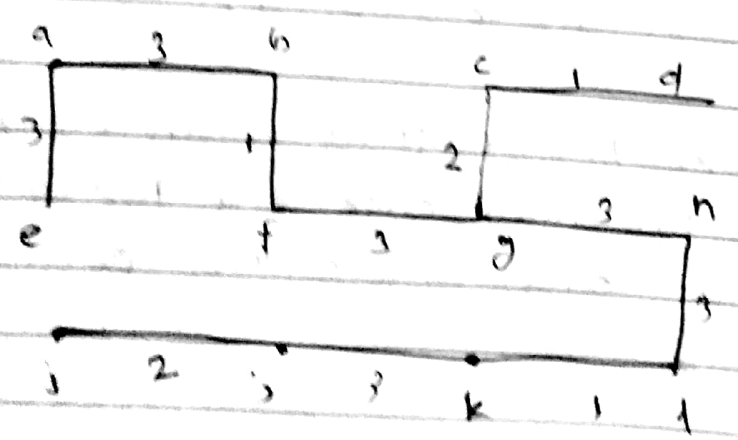
(iii) Take the edge a which is adjacent to the previous edge and having minimum weight and which does not form the cycle

(iv) continue the process till (n-1) edges are chosen e.g.



First we start with ~~vertex~~ edge KL having weight 1

edge	weight
KL	1
dh	3
gh	3
gc	2
cd	1
gf	3
fb	1
ba	3
ae	3
jk	3
ji	2

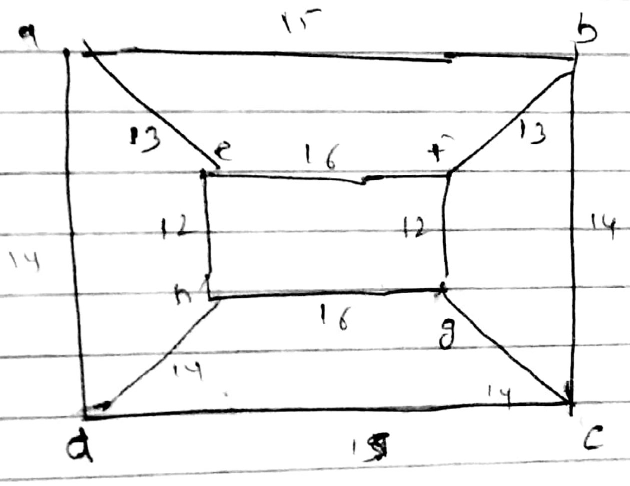


Kruskal's Algorithm

The steps are as.

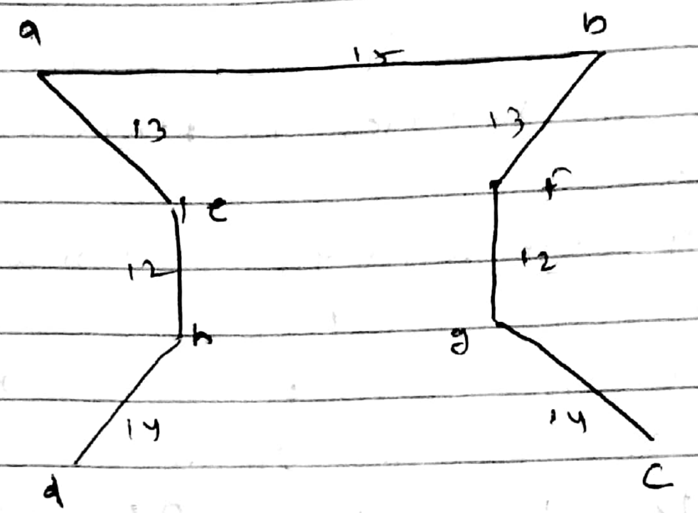
1. Arrange all the edges in increasing order of their weights
2. Add the edge with the minimum weight to the spanning tree.
3. Add the edge with the next higher weight and do not form a cycle. Repeat the process till n-1 edges are added

ex



edge	weight	Selected
eh	12	Yes
fg	12	Yes
ae	13	✓
bf	13	✓
dh	14	✓
gc	14	✓
ad	14	✗
bc	14	✗
ab	15	✓
dc	15	✗
ef	16	–
hg	16	–

Scanned by CamScanner


a
b
15
13
13
e
f
12
12
h
g
14
14
d
c

(a) f is one-to one, but not onto :>

$f(x) = e^x$

onto on $(0, \infty)$

(b) f is onto but not one-to one

$$f(x) = \begin{cases} \ln(x) & \text{if } x>0 \\ 0 & \text{if } x \le 0 \end{cases}$$

(c) neither one-one nor onto :>

$f(x) = x^2$

(d) f is one-one and onto

Algebraic form $f(x) = \underline{\underline{3x-4}}$

Total function $A \to m$, $B \to n$

$\underline{\underline{n^m}}$

$2^n - 2$

one-one fn $\quad {}^nP_m$

onto fn $\quad \sum_{k=0}^{n} (-1)^k \, {}^nC_k \, (n-k)^m$

DFA	NDFA
① All transition are deterministic	Some transition could be non deterministic
② Each transition leads to exactly one state	② A transition could lead to a subset of states
③ Accepts input if the last state visited is in F	Accepts input if one of the last state is in F
④ Backtracking possible	Not possible
⑤ harder to construct because of the no. of states	generally easier than DFA to construct
⑥ Practical implementation is feasible	practical implementation limited but emerging